



UNIVERSITÄT  
KOBLENZ · LANDAU

Fachbereich 4: Informatik

# Simulation ethnischer Konflikte

## Diplomarbeit

zur Erlangung des Grades eines  
Diplom-Informatikers  
im Studiengang Wirtschaftsinformatik

vorgelegt von

**Suvad Markisic und Sinan Kilic**

Betreuer: Prof. Dr. Klaus G. Troitzsch,  
Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich 4  
Erstgutachter: Prof. Dr. Klaus G. Troitzsch,  
Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich 4  
Zweitgutachter: Dipl. Inf. Ulf Lotzmann,  
Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich 4  
Drittgutachter: Dr. Martin Neumann,  
Institut für Soziologie, RWTH Aachen

Koblenz, im Januar 2012

## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

	<b>Ja</b>	<b>Nein</b>
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	<input type="radio"/>	<input type="radio"/>
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	<input type="radio"/>	<input type="radio"/>

.....  
(Ort, Datum)

.....  
(Unterschrift)

## Vorwort und Danksagung

“Read! In the Name of your Lord, Who has created (all that exists), has created man from a clot. Read! And your Lord is the Most Generous, Who has taught (the writing) by the pen, Has taught man that which he knew not.” (96, 1-5)

---

Durch diese Diplomarbeit wird ein großes Kapitel in meinem Leben beendet. Hiermit möchte ich mich bei den Personen, die mich während meiner Studienzeit und der Entstehung dieser Diplomarbeit unterstützt haben, bedanken.

Ich danke Herrn Suvad Markisic für die angenehme Zusammenarbeit und seine Freundschaft während meiner ganzen Studienzeit.

Ich bedanke mich bei Hüseyin Tatar, weil er sich seit meiner Kindheit um meine Ausbildung gekümmert hat.

Ferner bedanke ich mich bei meinem Bruder, der mich zum Studieren motiviert hat.

Mein ganz besonderer Dank gilt abschließend Songür Kilic. Da sie mir meinen Studienabschluss ermöglicht hat und mir immer helfend zur Seite stand, möchte ich ihr diese Diplomarbeit widmen.

Sinan Kilic

---

Ich möchte diese Gelegenheit nutzen und mich den Menschen bedanken, die zum Erfolg dieser Arbeit und meines Studiums beigetragen haben.

Ich bedanke mich vor allem meinen Eltern und meiner Schwester, die immer hinter mir standen und mich in allen Situationen unterstützt haben.

Ein besonderer Dank gilt Hafiz Vehbija Secerovic, der mich immer wieder motivierte, weiterzumachen und besonders in schwierigen Abschnitten meines Lebens für mich da war.

Ich danke auch meiner lieben Frau, die mir immer mit einem Lächeln entgegenkam und viel Verständnis mir gegenüber zeigte.

Ich bedanke mich auch bei meinem guten Freund, Herrn Sinan Kilic, mit dem ich seit dem ersten Semester immer in einer fairen und freundschaftlichen Atmosphäre zusammengearbeitet habe.

Suvad Markisic

---

Wir bedanken uns auch ganz herzlich bei unseren drei Betreuern, Prof. Dr. Klaus G. Troitzsch, Dipl. Inf. Ulf Lotzmann und Dr. Martin Neumann für die kompetente Betreuung und für ihre hilfreichen Anregungen bei der Erstellung dieser Arbeit.

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielsetzung . . . . .	2
1.3	Aufbau der Arbeit . . . . .	2
<b>2</b>	<b>Einführung und Begriffsabgrenzung</b>	<b>4</b>
2.1	Modellierung realweltlicher Systeme . . . . .	4
2.2	Agentenbasierte Simulation . . . . .	4
2.2.1	Agenten . . . . .	4
2.2.2	Umgebung . . . . .	5
2.3	Ethnie und Ethnischer Konflikt . . . . .	5
2.4	Ursachen ethnischer Konflikte . . . . .	6
2.4.1	Soziale und ökonomische Benachteiligung . . . . .	6
2.4.2	Bedrohung kollektiver Identität . . . . .	6
2.4.3	Die Rolle von politischen Eliten . . . . .	7
2.4.4	Fazit . . . . .	7
2.5	Einige Charakteristika des Verlaufs ethnischer Konflikte . . . . .	7
<b>3</b>	<b>Hintergründe</b>	<b>9</b>
3.1	Jugoslawien . . . . .	9
3.2	Ethnische Zusammensetzungen der Teilrepubliken vor 1991 . . . . .	10
3.3	Desintegration und Staatszerfall . . . . .	12
3.4	Jugoslawienkriege . . . . .	12
<b>4</b>	<b>Anforderungen und Konzept</b>	<b>14</b>
4.1	Umgebung . . . . .	14
4.2	Agenten . . . . .	14
4.2.1	Politiker . . . . .	14
4.2.2	Bürger . . . . .	17
<b>5</b>	<b>Modellierungs- und Simulationswerkzeuge</b>	<b>21</b>
5.1	ArcGIS . . . . .	21
5.1.1	Einführung in ArcMap . . . . .	23
5.1.2	Darstellung-von-GIS-Daten . . . . .	24
5.1.3	Editieren von Grafik-Daten (ArcMap) . . . . .	28
5.1.4	ArcCatalog . . . . .	32
5.2	EmIL . . . . .	33
5.3	Repast . . . . .	33
5.3.1	Kontext . . . . .	34
5.3.2	Projektionen . . . . .	34
<b>6</b>	<b>Implementierung</b>	<b>35</b>
6.1	ArcGIS . . . . .	35
6.1.1	Aufbau der Karte . . . . .	35
6.1.2	Erstellung der Shapefiles mit ArcCatalog . . . . .	37
6.1.3	Zeichnung der Landkarte mit ArcMap . . . . .	37
6.1.4	Erstellung und Bearbeitung der Attributentabelle mit empirischen Daten . . . . .	39
6.1.5	Behebung von Referenzfehlern für Layer . . . . .	41

<b>6.2</b>	<b>EmIL</b> . . . . .	<b>42</b>
<b>6.3</b>	<b>Repast</b> . . . . .	<b>43</b>
	<b>6.3.1</b> <b>Parameter</b> . . . . .	<b>44</b>
	<b>6.3.2</b> <b>Kontext</b> . . . . .	<b>48</b>
	<b>6.3.3</b> <b>Kontexte der Umgebung</b> . . . . .	<b>49</b>
	<b>6.3.4</b> <b>Kontexte der Agenten</b> . . . . .	<b>51</b>
	<b>6.3.5</b> <b>Granularität der Zeiteinheit</b> . . . . .	<b>54</b>
	<b>6.3.6</b> <b>Politiker</b> . . . . .	<b>54</b>
	<b>6.3.7</b> <b>Bürger</b> . . . . .	<b>59</b>
	<b>6.3.8</b> <b>Watcher</b> . . . . .	<b>65</b>
	<b>6.3.9</b> <b>Demonstration</b> . . . . .	<b>66</b>
	<b>6.3.10</b> <b>Miliz</b> . . . . .	<b>67</b>
<b>7</b>	<b>Simulationslauf und Ergebnisanalyse</b>	<b>70</b>
	<b>7.1</b> <b>Zusammenhang zwischen Parametern</b> . . . . .	<b>70</b>
	<b>7.2</b> <b>Simulationsläufe</b> . . . . .	<b>73</b>
	<b>7.2.1</b> <b>1. Simulationslauf</b> . . . . .	<b>76</b>
	<b>7.2.2</b> <b>2. Simulationslauf</b> . . . . .	<b>83</b>
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>89</b>
	<b>Literatur</b>	<b>91</b>

## Abbildungsverzeichnis

1.1	Wechselwirkung der Agenten [23]	2
3.1	Ethnische Gruppen in Ex-Jugoslawien[8]	11
4.1	Use Case Diagramm - Evaluationsprozess	15
4.2	Use Case Diagramm - Entscheidungsprozess	17
4.3	Use Case Diagramm - Mobilisierungsprozess	19
5.1	Modellierungs- und Simulationswerkzeuge	21
5.2	ArcGIS - Desktop-Produkte [12]	22
5.3	ArcGIS - Applikationen [12]	22
5.4	ArcMap - Oberfläche	23
5.5	ArcMap - Features-Punkte [3]	24
5.6	ArcMap - Features-Linien [3]	25
5.7	ArcMap - Features-Polygone [3]	25
5.8	ArcMap - Shapefiles [3]	26
5.9	ArcMap - Attributentabelle	26
5.10	ArcMap - Layer [3]	28
5.11	ArcMap - Das Auswählen eines Shapefiles [3]	29
5.12	ArcMap - Fenster zur Features-Erstellung [3]	30
5.13	ArcMap - Werkzeugsatz [3]	31
5.14	ArcMap - Stützen und Segmente [3]	31
5.15	ArcMap - Beenden einer Zeichnung [3]	31
5.16	ArcCatalog	32
5.17	EmIL Event Board	33
6.1	Staatsgrenzen von Serbien, Bosnien und Kroatien	36
6.2	Verwaltungskreise von Serbien, Bosnien und Kroatien	37
6.3	ArcMap - Shapefile Hinzufügen	38
6.4	ArcMap - Auto Complete Polygon	39
6.5	ArcMap - Eintrag in die Attributentabelle	40
6.6	Attributentabelle - Choroplethenkarte	40
6.7	Einwohnerzahlen nach Ethnizitäten [24]	41
6.8	ArcMap - Referenzfehler	42
6.9	Event-Action-Tree	42
6.10	Aufbau von parameters.xml	43
6.11	Einbindung ins Repast	43
6.12	Abbildung der Agenten auf der EmIL-Ebene	43
6.13	Aufbau von parameters.xml	44
6.14	GUI Parameter View	44
6.15	Zugriff auf die Parameter in parameters.xml	47
6.16	Aufbau von context.xml	48
6.17	Kontexthierarchie des Konfliktmodells	48
6.18	Scenario Tree	49
6.19	Metamodell - Kontextaufbau der Umgebung	49
6.20	BosniaContext.java - Projektion	50
6.21	BosniaContext.java - ShapefileLoader	50
6.22	StateContext.java - Hinzufügen der Sub-Kontexte	50
6.23	Kontextaufbau	51
6.24	Klassendiagramm - Zugehörigkeit der Politiker	51
6.25	PoliticianContext.java - Koordinaten	52
6.26	PoliticianContext.java - Generierung der Agenten	52

6.27	PoliticianContext.java - Synchronous Speeches . . . . .	52
6.28	Choropleth.java - Aufbau der Klasse . . . . .	53
6.29	CitizenContext.java - Iteration der Choropleth Objekte . . . . .	53
6.30	CitizenContext.java - Berechnung der initialen Koordinaten . . . . .	54
6.31	Klassendiagramm - Zugehörigkeit der Politiker . . . . .	55
6.32	Politician.java - Weitere Attribute des Politikers . . . . .	55
6.33	Politician.java - ID Vergabe . . . . .	56
6.34	Kommunikationsweise von EmIL- und Repast-Agenten . . . . .	56
6.35	Politician.java - Senden eines Events E1 an den EmIL-Agenten . . . . .	56
6.36	Politician.java - Erstellung einer EmilMessage . . . . .	56
6.37	Politician.java - Scheduler-Methode . . . . .	57
6.38	Politician.java - speechProcess() . . . . .	57
6.39	Politician.java - Ausführen einer Aktion . . . . .	58
6.40	Speech.java - Die Struktur der Speech Klasse . . . . .	58
6.41	Klassendiagramm - Bürger . . . . .	59
6.42	Identitätsausprägung der Bürger . . . . .	59
6.43	Citizen.java - Nachbarsuche . . . . .	60
6.44	Suchprozess der Nachbarsuche . . . . .	60
6.45	Citizen.java - Bestimmung der Fluchtkoordinaten . . . . .	61
6.46	Suchprozess der Fluchtkoordinaten . . . . .	61
6.47	Citizen.java - step() - Methode . . . . .	62
6.48	Citizen.java - Rate und Discuss Queues . . . . .	62
6.49	Bewertung der Reden . . . . .	62
6.50	Aktivitätsdiagramm - Demonstrationsteilnahme . . . . .	64
6.51	Citizen.java - Beweglichkeit des Agenten . . . . .	65
6.52	Citizen.java - agentStep() Methode . . . . .	65
6.53	Watcher.java - Annotation . . . . .	66
6.54	Watcher.java - MilitiaBuilder Listen . . . . .	67
6.55	Watcher.java - MilitiaBuilder . . . . .	68
6.56	Watcher.java - Erstellung von Miliz-Objekten . . . . .	68
6.57	Angriffsziele der Miliz . . . . .	69
6.58	Citizen.java - Vertreibung und Radikalisierung . . . . .	69
7.1	Simulation - Serbische Identitätsausprägungen 1. Lauf . . . . .	70
7.2	Simulation - Serbische Identitätsausprägungen 2. Lauf . . . . .	71
7.3	Simulation - Serbische Identitätsausprägungen 3. Lauf . . . . .	71
7.4	Simulation - Serbische Identitätsausprägungen 4. Lauf . . . . .	72
7.5	Simulation - Serbische Identitätsausprägungen 5. Lauf . . . . .	72
7.6	Simulationsläufe - Bevölkerungsübersicht - Tick 1 . . . . .	74
7.7	Simulationsläufe - Choroplethenkarte Bosniaken . . . . .	75
7.8	Simulationsläufe - Choroplethenkarte Kroaten . . . . .	75
7.9	Simulationsläufe - Choroplethenkarte Serben . . . . .	76
7.10	1. Simulationslauf - Bevölkerungsübersicht - Tick 1820 . . . . .	77
7.11	1. Simulationslauf - Bevölkerungszahlen . . . . .	77
7.12	1. Simulationslauf - Choroplethenkarte - Bosniaken . . . . .	78
7.13	1. Simulationslauf - Choroplethenkarte - Kroaten . . . . .	78
7.14	1. Simulationslauf - Choroplethenkarte - Serben . . . . .	79
7.15	1. Simulationslauf - Identitätsausprägung der Bosniaken . . . . .	79
7.16	1. Simulationslauf - Identitätsausprägung der Kroaten . . . . .	80
7.17	1. Simulationslauf - Identitätsausprägung der Serben . . . . .	80
7.18	1. Simulationslauf - Politiker in Bosnien . . . . .	81

7.19	1. Simulationslauf - Politiker in Kroatien . . . . .	81
7.20	1. Simulationslauf - Politiker in Serbien . . . . .	82
7.21	2. Simulationslauf - Bevölkerungsübersicht - Tick 1820 . . . . .	83
7.22	2. Simulationslauf - Bevölkerungszahlen . . . . .	84
7.23	2. Simulationslauf - Choroplethenkarte - Tick 1820 . . . . .	84
7.24	2. Simulationslauf - Choroplethenkarte - Tick 1820 . . . . .	85
7.25	2. Simulationslauf - Choroplethenkarte - Tick 1820 . . . . .	85
7.26	2. Simulationslauf - Identitätsausprägung . . . . .	86
7.27	2. Simulationslauf - Identitätsausprägung . . . . .	86
7.28	2. Simulationslauf - Identitätsausprägung . . . . .	87
7.29	2. Simulationslauf - Politiker in Bosnien . . . . .	87
7.30	2. Simulationslauf - Politiker in Kroatien . . . . .	88
7.31	2. Simulationslauf - Politiker in Serbien . . . . .	88

## Tabellenverzeichnis

1	Bevölkerungsverteilung von Bosnien und Herzegowina im Jahr 1991[2] . . . . .	9
2	Bevölkerungsverteilung von Kroatien im Jahr 1991[10] . . . . .	10
3	Bevölkerungsverteilung von Serbien im Jahr 1991[24] . . . . .	10
4	Simulationsparameter . . . . .	45
5	Koordinatenpunkte der Redetypen . . . . .	63
6	Identitätsausprägung nach einer Demonstration . . . . .	66
7	Demonstrationsgrößen . . . . .	67
8	Anzahl der Bürger-Agenten in der Simulation . . . . .	73
9	Anzahl der Politiker-Agenten in der Simulation . . . . .	73
10	Farbendarstellung der Agenten . . . . .	74

## 1 Einleitung

Nach dem Ende des kalten Krieges gab es Hoffnungen, dass “die grundsätzliche Spaltung der Welt nun endlich überwunden” [16] war und die Welt friedlicher und zivilisierter sein wird. Am 11. September 1990 sprach der US-Präsident George H. W. Bush von einer neuen Weltordnung, die nach dem kalten Krieg auf der ganzen Welt herrschen würde. Mit dieser neuen Weltordnung sollten “sich Konflikte auf vernünftige und institutionalisierte Weise auf der Basis allgemein anerkannter Werte regeln” [16] lassen. Trotz dieser Hoffnungen hat die Entwicklung der Weltpolitik leider gezeigt, “dass weder die immer weiter verregelten internationalen Beziehungen friedlich, noch die vermeintlich demokratisierten Gesellschaften gewaltfrei geworden sind” [16]. Die Demokratisierung führt auch nicht zwangsläufig zu einer besseren wirtschaftlichen Entwicklung [16]. Die Ereignisse der Jahre 1989 bis 1993, wie beispielsweise der Fall der Berliner Mauer, die Auflösung der Vielvölkerstaaten, wie der Sowjetunion oder der Sozialistischen Föderativen Republik Jugoslawien, der zweite Golfkrieg oder der berühmte Beitrag von Samuel Huntington über den “clash of civilizations”, haben dazu beigetragen, dass die sogenannte neue Ära durch ethnische Konflikte bzw. Konflikte der Kulturen geprägt wurde [16].

Wegen der oben genannten Ereignisse haben ethnische Konflikte die Aufmerksamkeit der Öffentlichkeit und der Wissenschaft geweckt [23]. Diese Aufmerksamkeit lässt sich durch die geographische Nähe der Sowjetunion und Jugoslawien zu westeuropäischen Ländern und die Folgen, die von diesen Ländern stark spürbar waren, beispielsweise durch die Flüchtlinge erklären. Dabei versucht die Wissenschaft die wahren Hintergründe ethnischer Konflikte herauszufinden. Wissenschaftliche Untersuchungen zeigen, dass ethnische Konflikte im Zeitraum zwischen 1985 und 1994 einen Anteil von 64,7% an der Gesamtzahl aller Konflikte haben [22, 13]. Über 50% dieser Konflikte sind zwischen verschiedenen Ethnien (13%) oder zwischen einer Ethnie und dem Staat (44%) [22, 13]. Ethnische Konflikte sind nicht nur regional zu betrachten, sondern sie bedrohen den Weltfrieden [14].

In den Medien wird oft berichtet, dass die Ethnizität die “Ursache politischer Fragmentierung sei” [27]. Hierbei tauchen Fragen auf, ob die Ethnisierung für bestimmte Interessen der herrschenden Eliten funktional eingesetzt wird [16, 29] oder, wie die Kriege, die extrem nationalistisch und hasserfüllt waren, in den Ländern des ehemaligen Jugoslawiens innerhalb kürzester Zeit zustande kommen konnten [28].

In Anbetracht dessen beschäftigt sich diese Diplomarbeit mit der Untersuchung von Ursachen ethnischer Konflikte mit Hilfe der agentenbasierten Simulation. Dabei fokussiert sie sich besonders auf die politischen Akteure und ihrer Rolle bei der Entstehung solcher Konflikte.

### 1.1 Motivation

Der jugoslawische Bürgerkrieg ist der erste Krieg in Europa nach dem Zweiten Weltkrieg. Dieser mitten in Europa stattgefundenen Krieg zeigte die “Hilfs- und Konzeptlosigkeit der Staatsgemeinschaft” [22, 12]. Westliche Staaten befürchten, dass ethnische Konflikte zu einem Charakteristikum der neuen Weltordnung werden können. Je nach Definition existieren 2500-8000 Ethnien, die als potentielle Konfliktursachen betrachtet werden. Angesichts dieser Tatsachen

sind wissenschaftliche Untersuchungen über die Dynamiken ethnischer Konflikte enorm wichtig. Zu diesem Zweck bietet die agentenbasierte Simulation, die mit empirischen Daten untermauert werden kann, wichtige Ansatzpunkte zur Analyse solcher Konflikte.

Bei dieser Untersuchung dient Ex-Jugoslawien als Modellbeispiel, da unterschiedliche Ethnien im damaligen Jugoslawien vorhanden waren, und die Entstehung und der Ablauf des Konflikts sehr gut dokumentiert sind. Ferner sind interne Prozessmechanismen, die zu einer Eskalation führen, gut beobachtbar.

## 1.2 Zielsetzung

Das Ziel der Diplomarbeit ist die Untersuchung von Dynamiken der Konflikteskalation mit Hilfe von agentenbasierter Simulation. Das Modell unterscheidet dabei zwischen zwei Arten von Agenten:

- **Politische Elite:** halten Reden und streben nach Macht und Einfluss und gehören zu einer Ethnie. Deren Anzahl wird durch die Anzahl der Bürger gleicher Ethnie bestimmt.
- **Lokale Bevölkerung:** sind einfache Bürger mit ethnischer Zugehörigkeit. Die Anzahl soll repräsentativ sein und damalige Verhältnisse von Jugoslawien möglichst gut darstellen.

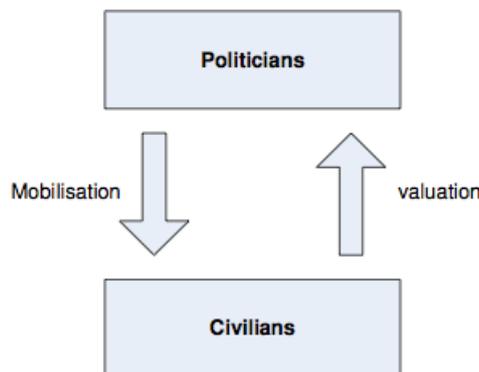


Abbildung 1.1: Wechselwirkung der Agenten [23]

Beide Agenten stehen in Wechselwirkung (siehe Abbildung 1.1) und haben Entscheidungs- und Handlungsmöglichkeiten, wodurch sie im Laufe der Simulation ein normatives Verhalten entwickeln können. Bürger werden von Politikern beeinflusst bzw. mobilisiert und Politiker werden von Bürgern validiert. Hierbei soll untersucht werden, inwiefern die politische Elite Einfluss auf die Ethnisierung der Bevölkerung durch Reden hat. Ferner soll die zu erstellende Simulation nicht im Sinne eines Kriegsmodells betrachtet werden.

## 1.3 Aufbau der Arbeit

Im weiteren Verlauf dieser Arbeit werden in Kapitel zwei die wichtigsten Begriffe der Thematik, Ursachen und Charakteristika ethnischer Konflikte definiert. Nach der Definition beschreibt Kapitel drei die historische Entwicklung von

Jugoslawien und die Hintergründe des Bürgerkriegs. Kapitel vier stellt die Konzeption und den Aufbau der Simulation vor. Anschließend wird in Kapitel fünf das Modellierungs- und Simulationswerkzeug vorgestellt, welches für die in Kapitel sechs beschriebene Implementierung eingesetzt wurde. Im Anschluss der Arbeit folgen die Ergebnisanalyse und der Ausblick.

## 2 Einführung und Begriffsabgrenzung

In diesem Kapitel werden die Begrifflichkeiten der Thematik definiert. Der erste Teil dieses Kapitels behandelt dabei die Begriffe der Modellierung und der Simulation. Im zweiten Teil werden die Begriffe Ethnie und ethnische Konflikte, sowie deren Ursachen untersucht.

### 2.1 Modellierung realweltlicher Systeme

Als ein Modell wird eine vereinfachte Beschreibung eines realen Systems bezeichnet. Dabei soll das Modell die Realität durch Abstraktion gezielt auf die problemrelevanten Aspekte vereinfachen und zu vergleichbaren Aussagen oder Beobachtungen führen wie das reale System. Genau durch diese Auslassung unwichtiger Dinge erlaubt das Modell den Modellierern, sich auf die essentielle Problematik eines komplexen Systems zu fokussieren [15].

Die Simulation kann als eine besondere Art der Modellierung betrachtet werden. Simulationen bestehen aus einer Menge von Eingaben bzw. Einstellungen, die für die Initialisierung des Modells notwendig sind. Das Resultat einer durchgeführten Simulation ist eine Menge von Ausgaben, welche das Verhalten und den Zustand des Modells nach einer bestimmten Zeit darstellt. Dieses Resultat kann mit echten bzw. realen Daten verglichen werden, um zu prüfen, inwieweit das Modell das reale System überdeckt bzw. reproduziert [15].

In diesem Sinne kann ein solches Modell, welches die Dynamik eines realen Systems zuverlässig und getreu reproduzieren kann, als ein Vorhersage-Modell genutzt werden. Die Simulation würde somit zeigen, wie sich das beobachtete System bzw. dessen Verhalten in der Zukunft entwickeln würde [15].

Simulationen eignen sich deshalb besonders für die Theoriebildung. Jeder Theorie liegen mehrere beschreibende oder erklärende Annahmen zugrunde, die bestätigt werden müssen. Simulationen geben Aufschlüsse, inwieweit die vermuteten Annahmen einer Theorie zutreffend sind und die Theorie angepasst werden muss, um zu einer validen Aussage zu führen [15].

### 2.2 Agentenbasierte Simulation

Die agentenbasierte Simulation beschäftigt sich mit der Analyse und Interpretation komplexer Systeme, welche aus diskreten Einheiten bestehen. Diese Einheiten oder Agenten haben freie Entscheidungs- und Handlungsmöglichkeiten, verfügen aber nur über Informationen aus ihrer Umgebung. Agentenbasierte Modelle sind somit dezentrale Systeme, deren Verhalten vom kollektiven Verhalten der Agenten im System abhängt. Ein agentenbasiertes Modell besteht aus einer endlichen Menge von Agenten, die eine Reihe von Beziehungen zueinander haben und die miteinander interagieren [20].

#### 2.2.1 Agenten

Es gibt keine generell akzeptierte Definition, mit welcher ein Agent definiert werden kann. Vielmehr aber werden Begriffe eingesetzt, welche die Eigenschaften von Agenten charakterisieren [15].

Die wichtigste Eigenschaft eines Agenten ist die, dass er die Fähigkeit hat, autonom und ohne externe Einflüsse zu agieren. Ein Agent ist ein eindeutig identifizierbares Individuum, welches bestimmte Merkmale besitzt, die durch Attribute beschrieben sind. Diese Attribute können entweder statisch oder dynamisch sein. Statische Attribute sind Attribute, die während der Simulation bzw. Lebendigkeit des Agenten ihren Wert nicht ändern, wie z. B. die ID oder der Name des Agenten. Dynamische Attribute ändern sich während der Simulation. Ein Beispiel dafür wäre ein Agent mit limitierten Ressourcen oder einem ständig ändernden Nachbarschafts-Netzwerk. Solche Unterschiede machen die Agenten in einem System heterogen, was konkret heißt, dass sie unterschiedliche Reaktionen auf bestimmte Ereignisse aus ihrer Umgebung zeigen können. Diese Eigenschaft wird als die Reaktivität bezeichnet. Analog dazu kann der Agent als eine aktive Einheit, bedingt durch sein Verhalten und seinen Zustand, selbstständig zielgerichtete Ereignisse bzw. Aktionen auslösen. Dies wird als die Proaktivität bezeichnet [15].

Agenten können ähnlich wie die Menschen, jedoch nur auf einem sehr abstrakten Niveau, miteinander kommunizieren, in Beziehung treten oder Nachbarschaftsnetzwerke aufbauen. Sie können aus ihrer Umgebung lernen oder können von ihrer Umgebung beeinflusst werden.

### 2.2.2 Umgebung

Die Art der Umgebung, in der sich die Agenten befinden, hängt von dem jeweiligen Modell ab. Stellen die Agenten bestimmte Personen oder Personengruppen der Realität dar, so sollten diese in einem raumbezogenen Kontext dargestellt werden [1]. Sie können sich in einer solchen Umgebung frei bewegen und diese explizit wahrnehmen. Diese Umgebung kann lokale oder globale Eigenschaften besitzen: Ein Beispiel für eine lokale Eigenschaft wäre ein begrenzter Ressourcenvorrat an einer bestimmten Stelle. Eine globale Eigenschaft wäre eine überall gleich geltende Bedingung. Um den Realitätsgrad zu verstärken, können dem Modell GIS Daten hinzugefügt werden. Eine solche Integration erlaubt eine detailgetreue Abbildung einer realen Umgebung und die Einschränkung des Agenten in seiner Beweglichkeit. In Anbetracht auf das große Potential und die enormen Möglichkeiten, eignet sich die agentenbasierte Modellierung besonders für die Simulation komplexer sozialer Systeme.

## 2.3 Ethnie und Ethnischer Konflikt

Die Definition der Ethnie ist sehr vielfältig und es existieren verschiedene Interpretationen. Häufig wird die Ethnie als ein Stamm und ethnische Konflikte als Stammeskonflikte definiert. Leider reichen diese Definitionen wegen der Komplexität des Begriffs bzw. der Problematik nicht aus. Anstatt den Begriff zu definieren, werden dessen Eigenschaften wie folgt aufgezählt. "Eine Gruppe ist als Ethnie zu verstehen, wenn sie

1. als eine historisch gewachsene oder wiederentdeckte Gemeinschaft von Menschen verstanden wird, welche sich größtenteils selbst reproduziert.
2. über eine spezifische, andersartige gemeinsame Kultur, insbesondere eine eigene Sprache verfügt,

3. (an) ein kollektives Gedächtnis oder eine geschichtliche Erinnerung anknüpft
4. dadurch über Solidarität und Wir-Gefühl verfügt.“ [22, 14]

Wegen der Bestimmung einer exakten Anzahl ethnischer Gruppen in einem Land, ist das Verständnis des Begriffs besonders wichtig. Wenn die Eigenschaften genauer betrachtet werden, hängt die Definition auch von “dem Zeitpunkt der Erhebung” ab. Dies zeigt, dass “ethnische Gruppen variabel, flexibel und mit der Zeit veränderlich” [22, 14] sind. Ferner wird eine fünfte Eigenschaft zu den oben genannten hinzugefügt. Sie besagt, “dass die Mitglieder einer Ethnie untereinander häufigere und engere Kontakte als mit Mitgliedern anderer Ethnien haben” [22, 14].

Ethnische Konflikte werden als “diejenigen Konflikte bezeichnet, in denen mindestens eine der Konfliktparteien eine ethnische Gruppe ist und in denen die Unterscheidung von “Freund” und “Feind” anhand ethnischer Zugehörigkeit vorgenommen wird” [22, 15]. Hierzu werden drei Typen von ethnischen Konflikten unterschieden:

1. “interethnische Konflikte zwischen verschiedenen Ethnien,
2. Konflikte zwischen einzelnen Ethnien und dem Staat,
3. intraethnische Konflikte zwischen Angehörigen einer Ethnie.” [22, 16]

Hierbei steht es fest, dass die Zugehörigkeit zu einer Ethnie bzw. zu unterschiedlichen Ethnien kein Grund für einen Konflikt ist. Er ist als “mehrdimensional” [22, 16] zu betrachten. Ein Konflikt entsteht erst, wenn es “Interessenunterschiede” [22, 16] zwischen den ethnischen Gruppen gibt.

## 2.4 Ursachen ethnischer Konflikte

In diesem Kapitel werden verschiedene Ursachen ethnischer Konflikte und deren Zusammenhänge erläutert.

### 2.4.1 Soziale und ökonomische Benachteiligung

In manchen Fällen der Wirtschaftskrisen kann es vorkommen, dass der Staat nicht mehr die Grundfunktionen, wie Gesundheit, Bildung, Nahrungsmittel und Infrastruktur, erfüllen kann [22, 24]. Die sich verschlechternden Lebensbedingungen führen oft zu einer Konkurrenz innerhalb der Bevölkerung und zwischen einzelnen Gruppen. Dies kann wiederum einen enormen Stress in der Gesellschaft aufbauen, wodurch die Wahrscheinlichkeit für einen ethnischen Konflikt erhöht wird. Laut Schmidt, kann allerdings der wirtschaftliche Niedergang und die besondere Benachteiligung bestimmter ethnischer Gruppen allein einen Konflikt nicht auslösen [22, 24].

### 2.4.2 Bedrohung kollektiver Identität

Die Untersuchungen des Krieges im ehemaligen Jugoslawien zeigen, dass ethnische Konflikte auf sozialpsychologische Faktoren beruhen können [22, 24]. Hierbei spielen zwei Erinnerungsmerkmale eine wesentliche Rolle. Die kollektiven Negativerinnerungen der Vergangenheit einer ethnischen Gruppe werden als

“chosen traumas” und “die kollektiven Erinnerungen an historische “Großtaten”, die ein Überlegenheitsgefühl einer ethnischen Gruppe hervorbringen”, werden als “chosen glories” bezeichnet [22, 25]. Chosen traumas wie “die Niederlage des serbischen Heeres gegen die Türken auf dem Amselfeld” [22, 25] wurden von dem damaligen Präsidenten Slobodan Milosovic gezielt zur ethnischen und nationalistischen Mobilisierung der Bevölkerung eingesetzt. Chosen traumas und chosen glories sind für die politische Manipulation und Mobilisierung der Gruppen sehr geeignet, da sie tief verwurzelt sind und “über eine Langzeit konsistente Feindbilder” [22, 25] hervorrufen können. Diese “manipulativ eingesetzten chosen traumas werden als eine Bedrohung der kulturellen oder religiösen Identität” [22, 25] betrachtet.

### 2.4.3 Die Rolle von politischen Eliten

Politische Eliten spielen bei der Mobilisierung ethnischer Gruppen und der Entstehung von ethnischen Konflikten eine Schlüsselrolle. “Ethnischer Hass und ethnische Gewalt sind das Ergebnis erfolgreicher Mobilisierung von oben” [22, 27]. Politiker können durch die Ausnutzung von ökonomischer und politischer Krisen ethnische Gruppen mobilisieren. “Versagt also die Regelungskompetenz des betreffenden Staates und treten neue ethnopolitische Führer, ethnische Ideologen und Unternehmer auf, so kann die kalkulierte ethnische Gewalt zu unkontrollierbaren Gewaltkreisläufen, am Ende zum Bürgerkrieg eskalieren ” [22, 27]. Dabei verwandeln Politiker die negative soziale, ökonomische, und politische Lage in ethnische Konflikte. “Eliten werden zu den eigentlichen Initiatoren ethnischer Konflikte” [22, 27]. Sie streben nach politischer Macht und individuellem ökonomischem Wohlstand [22, 27]. Die Untersuchung der Rollen politischer Eliten ist der Kernpunkt dieser Ausarbeitung.

### 2.4.4 Fazit

Ethnische Konflikte sind moderne Phänomene [22, 34] und haben unterschiedliche auslösende Faktoren, die von Eliten gezielt und manipulativ zur Mobilisierung bestimmter Bevölkerungsgruppen eingesetzt werden [22, 34]. Der Ethnologe Schlee besagt in einem Interview, obwohl ethnische Konflikte “häufig eine ethnische oder religiöse Ausdrucksform haben, haben die eigentlichen Konflikt- oder Kriegsursachen damit sehr wenig zu tun” [27]. Laut Schlee ist “der Zugang zu materiellen Ressourcen etwa Öl, Wasser, Weideland, Diamanten die eigentliche Konfliktursache” [27] und “diejenigen, die die strategischen Entscheidungen fällen, müssen sich überlegen, wie sie ihre eigene Gruppe oder Allianz stark genug machen, um die umkämpften Ressourcen für sich zu gewinnen” [27]. Hierbei werden Bündnispartner unter Menschen gesucht, mit denen sie Gemeinsamkeiten haben, wobei Identifikation über die Staats-, Volks-, Kultur- oder Religionszugehörigkeit in Frage kommt [27]. Ethnische Konflikte sind eine der größten Herausforderungen im heutigen Zeitalter und die Untersuchung dieser Phänomene für die Zukunft der internationalen Politik enorm wichtig [22, 34].

## 2.5 Einige Charakteristika des Verlaufs ethnischer Konflikte

Beim Verlauf ethnischer Konflikte werden folgende Charakteristika beobachtet:

1. Sie sind oft ernsthafte Kriege mit ethnischen Säuberungen, wobei Zivilisten Täter und Opfer sein können. Ethnische Konflikte unterscheiden sich von Kriegen, die zwischenstaatlich sind, dadurch dass keine Unterscheidung zwischen der Zivilbevölkerung und den Streitkräften seitens der Kriegsparteien gemacht wird [22, 32].
2. Kämpfende Akteure sind meistens paramilitärische Kräfte und sind hochmotiviert [22, 32]. Sie können in manchen Fällen "materielle Interessen verfolgen und kriminell agieren" [22, 32].
3. Da ethnische Gruppen ständig in Bewegung sind, "besitzen ethnische Konflikte in vielen Fällen einen grenzüberschreitenden Charakter" [22, 33]. Dies kann "die Ausweitung von Konflikten auf Nachbarstaaten und ganze Regionen" [22, 33] verursachen.
4. Kämpfe werden zur Schaffung eines ethnisch reinen Territoriums ausgeübt. Mit der steigenden Dauer des Konfliktes können "ethnisch homogene" [22, 33] Territorien entstehen.

## 3 Hintergründe

In diesem Kapitel wird ein Überblick über die damalige Lage in Jugoslawien verschaffen und die Chronologie wichtiger Ereignisse im Jugoslawienkrieg erläutert.

### 3.1 Jugoslawien

Jugoslawien bestand aus sechs Teilrepubliken (Bosnien und Herzegowina, Kroatien, Mazedonien, Montenegro, Serbien, Slowenien) und zwei autonomen Provinzen (Kosovo, Vojvodina). Der nach dem Zweiten Weltkrieg gegründete Staat wurde von 1945 bis 1963 als die Föderative Volksrepublik Jugoslawien und von 1963 bis 1992 als Sozialistische Föderative Republik (SFR) Jugoslawien bezeichnet. Die offizielle Aberkennung der Sozialistischen Föderativen Republik Jugoslawien hat durch Mehrheitsbeschluss der Vereinten Nationen im Jahr 1992 stattgefunden.

#### Bosnien und Herzegowina

Bosnien und Herzegowina waren seit dem Zweiten Weltkrieg bis in die frühen Neunziger eine multikulturelle Teilrepublik der SFR Jugoslawien. Im Zensus von 1991 wurden folgende statistische Daten erfasst (siehe Tabelle 1):

Gesamt	Muslime	Serben	Kroaten	Jugoslawier	Andere
4.377.033	1.902.956	1.366.104	760.852	242.682	104.439

Tabelle 1: Bevölkerungsverteilung von Bosnien und Herzegowina im Jahr 1991[2]

In Bosnien und Herzegowina ist beobachtbar, dass sich die jeweiligen ethnischen Gruppen nicht unbedingt mit dem Land, in dem sie leben, identifizieren. So ist z.B. eine Verhaltensweise feststellbar, dass sich ein in Bosnien lebender Serbe eher zu Serbien hingezogen fühlt und sich mit Serbien als Nation identifiziert, auch wenn ihn nichts mit Serbien im Sinne der Herkunft etc. verbindet oder er sogar nie in Serbien gewesen ist. Analog findet sich auch eine ähnliche Verhaltensweise bei den Kroaten wieder. Dieses Verhalten ist zugleich auf der politischen Ebene feststellbar. Politiker in Bosnien und Herzegowina erhalten immer direkt oder indirekt Unterstützung durch das Land, mit dem sie sich verbunden fühlen. Hierbei kann als Beispiel Radovan Karadzic betrachtet werden. Als Vorsitzender der Serbischen Demokratischen Partei (SDS - Srpska demokratska stranka) in Bosnien und Herzegowina wurde er direkt durch Slobodan Milosevic und zudem durch einen großen Teil der serbischen Bevölkerung unterstützt [5], während des Krieges dann auch durch die noch damalige Jugoslawische Volksarmee und serbische paramilitärische Truppen.

#### Kroatien

Die Sozialistische Republik Kroatien hatte eine deutlich homogenere Bevölkerung als z. B. Bosnien und Herzegowina. Laut amtlicher Statistik betrug die Gesamtbevölkerung im Jahr 1991 4.784.265 (siehe Tabelle 2). Die Bevölkerung bestand aus den folgenden ethnischen Gruppen: Kroaten (78%), Serben (12%),

Muslime (ca. 1%), Jugoslawen (ca. 2%) und andere ethnische Gruppen (ca. %7). Kroatien war mit Slowenien unter anderem die wirtschaftlich stärkste Teilrepublik der SFR Jugoslawien[7].

Gesamt	Kroaten	Serben	Muslime	Jugoslawier	Andere
4.784.265	3.736.356	581.663	43.469	106.041	433.004

Tabelle 2: Bevölkerungsverteilung von Kroatien im Jahr 1991[10]

### Serbien

Die Sozialistische Republik Serbien war die größte Teilrepublik der SFR Jugoslawien, deren Hauptstadt Belgrad auch die Hauptstadt der SFR Jugoslawien war. Laut amtlicher Statistik betrug die Gesamtbevölkerung im Jahr 1991 mehr als zehn Millionen (siehe Tabelle 3). Die Bevölkerung setzte sich aus Serben (ca. 62,5%), Jugoslawen (ca. 3,3%), Muslimen (ca. 3,2%) und den restlichen ethnischen Gruppen (ca. 30%) zusammen. Obwohl Kroaten in dieser Statistik nicht vorkommen, betragen sie ca. 2% der Gesamtbevölkerung. Innerhalb der Sozialistischen Republik Serbien gab es zwei autonome Provinzen, Vojvodina und Kosovo. Die Region zwischen diesen Provinzen wurde Zentralserbien genannt und deren administrative Verwaltung von der Sozialistischen Republik Serbien geführt.

Gesamt	Serben	Muslime	Kroaten	Jugoslawier	Andere
10.394.026	6.504.048	336.025	-	349.784	3.204.169

Tabelle 3: Bevölkerungsverteilung von Serbien im Jahr 1991[24]

### 3.2 Ethnische Zusammensetzungen der Teilrepubliken vor 1991

Als ein multiethnisches Land besaß Jugoslawien viele Gebiete die sogar über 90% nur von einer Ethnie besiedelt wurden. Abbildung 3.1 stellt diese Tatsache visuell dar:



Abbildung 3.1: Ethnische Gruppen in Ex-Jugoslawien[8]

Als Beispiel zur Verdeutlichung führt das statistische Bundesamt in Kroatien im Zensus von 1991 auf, dass in der damaligen Teilrepublik Kroatien 78 % der Bevölkerung Kroaten ausmachen [10]. Nur in wenigen Gebieten konnten die 12% Serben, die 1991 in Kroatien lebten, eine Mehrheit bilden. Während des Kroatienkrieges wurde in diesen Gebieten der de-facto-Staat "Republik serbische Krajna" ausgerufen. Im weiteren Kriegsverlauf eroberte Kroatien diese Gebiete zurück und sicherte sich somit seine territoriale Integrität.

Ähnlich wie in Kroatien, gab es auch in Serbien eine absolute Mehrheit der Serben gegenüber anderen ethnischen Gruppen. Die Muslime machten mit 2-3% der Gesamtbevölkerung die zweitgrößte ethnische Gruppe in Serbien aus. Die einzige Teilrepublik, in der keine absolute Mehrheit einer ethnischen Gruppe vertreten war, stellte Bosnien und Herzegowina dar. Mit ca. 44% waren die Muslime die größte ethnische Gruppe, gefolgt von den Serben mit 32% und Kroaten mit 17%.

Tatsächlich werden aber die in Bosnien und Herzegowina, teilweise in Serbien und Montenegro lebenden Muslime seit dem Mittelalter als Bosniaken bezeichnet. Die Bezeichnung Muslime als eine eigenständige ethnische Gruppe wurde jedoch nach dem zweiten Weltkrieg impliziert. In der Volkszählung von 1948 hatten Muslime des damaligen Jugoslawiens drei Optionen, wie sie ihre Zugehörigkeit erklären konnten. Sie hatten die Wahl zwischen muslimische Serben,

muslimische Kroaten oder einfach nur Muslime [17]. Ziel des kommunistischen Regimes war es, die Muslime als eine Volksgruppe und nicht als religiöse Zugehörigkeit darzustellen, um so ihre republikanische Identität zu nehmen. In den nachfolgenden 40 Jahren wurde diese Idee ständig vorangetrieben und unter der Brüderlichkeit und Einigkeit des Jugoslawismus verschleiert. Erst 1993 wird im Rahmen der Bosniakischen Versammlung die Bezeichnung Bosniake durch Abstimmung als Bezeichnung der Nation wieder eingeführt[17].

### 3.3 Desintegration und Staatszerfall

Wegen des Todes von Staats- und Parteiführer Josip Broz Tito und der schlechten Wirtschaftslage, gab es Unstimmigkeiten im Präsidium der Republik, die aus den Vertretern der Teilrepubliken und autonomen Provinzen bestand und die Republik regierte [7]. Dazu wurde die Autonomie von Kosovo im März 1989 vom serbischen Parlament aufgehoben. Dies löste Großdemonstrationen wie in Ljubljana aus, die von slowenischen intellektuellen Verbänden zur Unterstützung von Kosovaren aufgerufen wurde. In Belgrad kam es infolge dessen zu Unruhen. “Innerhalb von 24 Stunden versammelte sich eine Million Menschen vor dem Bundesparlament in Belgrad”[7] um gegen Slowenien zu protestieren. Solche soziale und wirtschaftliche Probleme trugen dazu bei, dass es in allen Teilrepubliken der Sozialistischen Föderativen Republik Jugoslawien, außer in Serbien, Wahlen zur staatlichen Souveränität gab. Die Aufteilung der Zustimmung der Souveränität in Teilrepubliken sieht wie folgt aus:

- **Slowenien:** 88,5 Prozent [32, 19]
- **Kroatien:** 94,2 Prozent [32, 19]
- **Bosnien und Herzegowina:** 99,4 Prozent[32, 19]
- **Mazedonien:** 74,1 Prozent [32, 19]
- **Montenegro:** 95,9 Prozent stimmten für ein souveränes Montenegro als Teil Jugoslawiens ab. Das Referendum wurde durch die nichtserbische Bevölkerung boykottiert [32, 19].

“Unmittelbar nach den Wahlen begannen in Slowenien, Kroatien und im Kosovo die Vorbereitungen für die Unabhängigkeit” [7]. Am 2. Juli 1990 wurden Souveränitätserklärungen von dem slowenischen und dem kroatischen Parlament und einer Gruppe von kosovarischen Abgeordneten veröffentlicht. “Die schrittweise Auflösung Jugoslawiens” [7] verursachte den ersten Krieg bzw. die ersten Kriege nach dem zweiten Weltkrieg in Europa.

### 3.4 Jugoslawienkriege

- **Zehn-Tage-Krieg:** nach der Unabhängigkeitserklärung von Slowenien im Jahr 1991 “kam es zu bewaffneten Auseinandersetzungen” zwischen Slowenen und der jugoslawischen Volksarmee. Da beide Parteien nach zehn Tagen zu einem Waffenstillstand kamen, wird dieser Krieg als der Zehn-Tage-Krieg genannt [7].

- **Kroatienkrieg:** hat von 1991 bis 1995 zwischen der kroatischen Armee und der jugoslawischen Volksarmee stattgefunden und wurde mit der Unabhängigkeit von Kroatien beendet [7].
- **Bosnienkrieg:** der Kroatienkrieg hat zugleich auch Konflikte zwischen den Ethnien in Bosnien und Herzegowina ausgelöst. Der Bosnienkrieg hat zwischen der bosnischen, kroatischen, jugoslawischen bzw. serbischen Armee sowie verschiedener serbischer paramilitärischer Truppen von 1992 bis 1995 stattgefunden. Er wurde durch den Dayton-Vertrag beendet [7].
- **Kosovokrieg:** hat zwischen der kosovarischen Unabhängigkeitsarmee und den serbischen Streitkräften stattgefunden. Der von 1998 bis 1999 gedauerte Krieg wurde durch die Luftangriffe der NATO beendet [7].

## 4 Anforderungen und Konzept

Dieses Kapitel beschreibt das Konzept und den Aufbau Konfliktmodells. Es werden zuerst die Anforderungen an die Umgebung gestellt und anschließend die Eigenschaften und Funktionalitäten der Agenten herausgearbeitet. Dieses Kapitel bildet somit die Verständnisbrücke zu Kapitel 6, in dem die hier erläuterten Konzepte implementiert werden.

### 4.1 Umgebung

Das Konfliktmodell beschränkt sich auf die drei großen Teilrepubliken des ehemaligen Jugoslawiens, nämlich Bosnien und Herzegowina, Kroatien und Serbien. Dabei soll das Modell eine realitätsnahe Bevölkerungsaufteilung nach Ethnizitäten von 1991 besitzen. Um dies realisieren zu können, soll das Modell mit realen statistischen Daten der statistischen Ämter der Teilrepubliken aufgebaut werden. Die Granularität der Bevölkerungsaufteilung soll dabei auf der Ebene der Verwaltungskreise bzw. Gespanschaften liegen. Hierfür ist eine zweistufige realitätsnahe Modellierung der Gebietsgrenzen notwendig, welche mit ArcGIS erreicht werden kann. Die erste Stufe bilden die territorialen Grenzen der Teilrepubliken und die zweite Stufe die Grenzen der Verwaltungskreise. Die zweite Stufe ist vor allem wegen der Bevölkerungsaufteilung von großer Bedeutung. Sie soll Informationen beinhalten, wie viele Einwohner im jeweiligen Verwaltungskreis leben und wie sie ethnisch aufgeteilt sind. Diese Gebietsgrenzen sollen in einem zweidimensionalen Koordinatensystem dargestellt werden. Die topologischen Gegebenheiten werden wegen der Komplexität vernachlässigt.

### 4.2 Agenten

Im Konfliktmodell kommen konkret zwei Typen von Agenten vor: die Politiker und die Bürger. Dabei können beide Typen von Agenten einer der drei größten ethnischen Gruppen angehören: Den Bosniaken, Kroaten und Serben. In erster Linie streben die Politiker nach Macht. Hierfür ist eine breite Unterstützung durch die Bürger notwendig. Dies versuchen sie zu erreichen, in dem sie mit verschiedenen Redetypen an die Bürger appellieren [23]. Der Politiker ist der komplexere Agententyp, denn im Unterschied zum Bürger, soll er über eine gewisse Intelligenz verfügen, welche ihm die Fähigkeit gibt, aus seinen gemachten Erfahrungen zu lernen.

#### 4.2.1 Politiker

Der Politiker bildet zusammen mit anderen Politikern das Parlament in der Teilrepublik, in der er aktiv ist. Unabhängig von seiner Ethnizität kann er sich durch Reden mit nationalistischen, national-zivilen oder zivilen Werten an die Bevölkerung wenden. Dabei ist der Politiker charakterisiert durch seine Fähigkeit die jeweiligen Massen seiner Ethnizität mobilisieren zu können. Als Beispiel hierzu kann die Amselfeld-Rede von Slobodan Milosevic betrachtet werden. Dies stärkt die politische Macht des jeweiligen Politikers und erhöht somit die Wahrscheinlichkeit an diesem Redetyp festzuhalten [23]. Die Abbildung 4.1 verdeutlicht die Anforderungen an den Politiker.

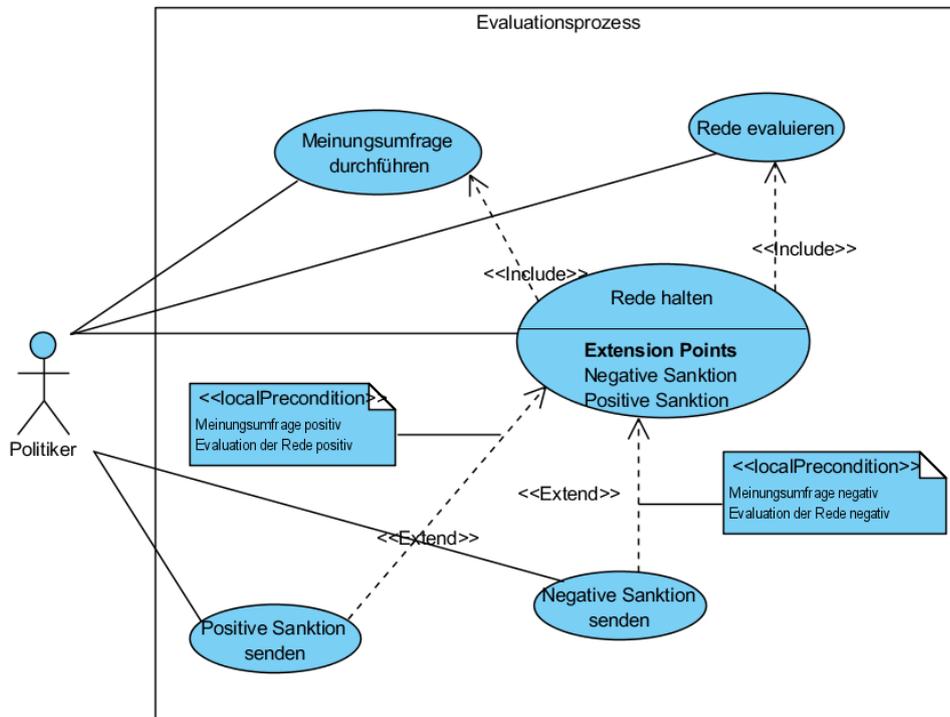


Abbildung 4.1: Use Case Diagramm - Evaluationsprozess

### Evaluation

Hat ein Politiker eine Rede gehalten, so finden zwei Evaluationsprozesse statt. Diese sollen die gehaltene Rede evaluieren und einen Feedback an den Politiker liefern. Der erste Prozess ist die Befragung durch ein Meinungsforschungsinstitut. Dabei werden zufällig 10-20% der Bürger ausgewählt, die nach ihrer Meinung zu dieser Rede befragt werden. Der zweite Evaluationsprozesses tritt nur ein, wenn eine Rede eine Demonstration ausgelöst hat. Wurde eine Demonstration in einer bestimmten Republik ausgelöst, so soll die Größe der Demonstration und die Anzahl der Demonstrationsteilnehmer von außerhalb dieser Republik in die Bewertung hineinfließen[23].

### Normatives Verhalten und Sanktionen

Je nach dem wie die Ergebnisse ausfallen, wird eine negative oder positive Sanktion ausgeführt. Eine positive Sanktion führt dazu, dass der Politiker an diesem Redetyp mit einer größeren Wahrscheinlichkeit festhält. Analog dazu führt eine negative Sanktion zu einer Abschwächung der Wahrscheinlichkeit diesen Redetyp wieder halten. Dies ist zugleich der Lernprozess der zu einem normativen Verhalten des Politikers führt. In Abhängigkeit von bereits gemachten Erfahrungen, wählt er den Redetyp aus, für den er die besten Erfolgsaussichten zur Steigerung seiner Macht hat.

### Wahlen

Ferner hat jeder Politiker eine Beliebtheitsskala, welche aus dem Durchschnitt der Redebewertungen und der Redeanzahl berechnet wird. Um die Wahlperioden zu simulieren, stellt die Beliebtheitsskala zugleich auch den Erfolg der Politiker dar und die Grundlage für die Ersetzung der unbeliebtesten Politiker nach Ablauf ihres Mandats [23].

### Glaubwürdigkeit

Die Glaubwürdigkeit ist eine weitere Eigenschaft eines Politikers. Ein Politiker, der konstant an einem Redetyp festhält, soll eine größere Glaubwürdigkeit besitzen als ein Politiker, der ständig den Redetyp ändert [23]. Auch wenn ein bestimmter Redetyp bei einem Teil der Bürger sehr gut ankommen würde, so würde die schlechte Glaubwürdigkeit eines Politikers zu einem schlechten Evaluationsergebnis führen.

### Mobilität

Zur Laufzeit der Simulation sollen die Politiker eindeutig der Republik zugeordnet werden können, in der sie aktiv sind. Dies wird erreicht indem die jeweiligen Politiker mit festen Koordinaten in die Umgebung projiziert werden. Im Unterabschnitt 4.1 wurde erläutert, dass die Umgebung unter anderem durch territoriale Grenzen der Teilrepubliken unterteilt werden soll. Während der Simulation ist dann immer feststellbar, wenn ein Politiker eine Masse von Bürgern in einer bestimmten Republik mobilisiert, welcher dieser Demonstranten seine Herkunft außerhalb dieser Republik hat. Hierbei soll vor allem die Unterstützung der Demonstrationen von außerhalb der jeweiligen Republik modelliert werden. In der Mengendarstellung gilt dann Folgendes:

Seien  $A$  und  $B$  Mengen von Bürgern  $x$  die Republiken darstellen und  $p$  ein Politiker aus  $A$ , sowie  $A_1 \subset A$  und  $B_1 \subset B$ . Findet eine Demonstration  $C$  statt, so ist:  $C \subseteq A_1 \cup B_1$ , und es gilt:

$$\forall x \in C : x \in A \Rightarrow x \text{ gleiche Republik} \vee x \notin A \Rightarrow x \text{ ausserhalb}$$

### Konfliktalarm

Der Konfliktalarm entsteht, wenn außerhalb der Republik in der die Demonstration stattfindet, eine breite Unterstützung der dortigen Bevölkerung existiert. Dabei sollen nur national orientierte Reden der Politiker als Ursache für den Konfliktalarm verantwortlich sein. Der Konfliktalarm ist die Voraussetzung, dass sich Bürger zu Milizen zusammenschließen können um gemischt ethnische Territorien anzugreifen. Dabei sollen die Vorzeichen eines anbahnenden Konflikts erfasst werden, so dass rechtzeitig Bemühungen gestartet werden können, um die folgende Konflikteskalation zu verhindern. Um diese Vorzeichen erfassen zu können, muss der Konfliktalarm zwei Grenzwerte besitzen. Beim Überschreiten des ersten Grenzwertes, sollen Maßnahmen bzw. Bemühungen zur Vermeidung des Konflikts durchgeführt werden. In diesem Fall soll eine negative Sanktion an den Politiker gesendet werden, dessen Rede die erste Stufe des Konfliktalarms ausgelöst hat. Dies ist empirisch mit der Einberufung des Botschafters

zu Konsultation vergleichbar, was die zweit-schärfste Sanktion im diplomatischen Verkehr darstellt. Wird der zweite Grenzwert überschritten, so ist die Voraussetzung für die Bildung von Milizen gegeben.

#### 4.2.2 Bürger

Der Bürger hat eine nationale und zivile Wertorientierung, die unterschiedlich ausgeprägt sein kann. Diese Wertorientierung soll indizieren, in wie weit ein bestimmter Bürger die zivile Werte bzw. den Jugoslawismus vertritt und in welchem Maße er sich seiner nationalen Zugehörigkeit verbunden fühlt [23]. Der Bürger hat primär die Aufgabe die Reden der Politiker aufzunehmen und zu bewerten. Dieser Bewertung folgt ein Entscheidungsprozess, welcher in Abbildung 4.2 verdeutlicht wird.

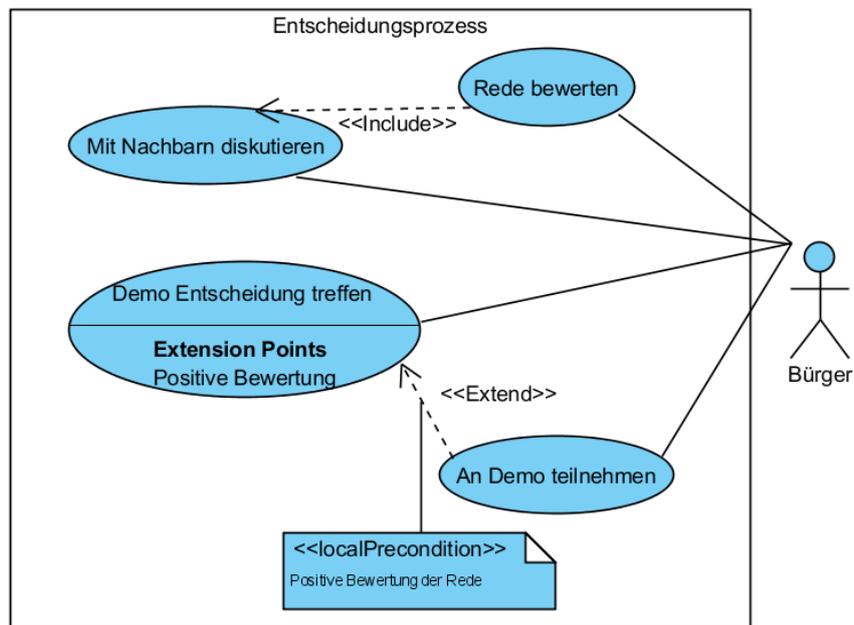


Abbildung 4.2: Use Case Diagramm - Entscheidungsprozess

#### Nachbarschaftsnetzwerk

Als Teil eines sozialen Systems hat der Bürger eine bestimmte Anzahl von Nachbarn mit denen er kommuniziert, darüber hinaus aber auch über die gehaltenen Reden diskutiert. Als Nachbarn können dabei, genauso wie in der Realität, Bürger unterschiedlicher Ethnizität vorkommen.

#### Bekanntschafnetzwerk

Neben der Nachbarschaft die lokal begrenzt ist, kann ein Bürger ein Bekantshafnetzwerk haben. Das Bekantshafnetzwerk ist in diesem Falle ein globales Netzwerk welches alle Teilrepubliken bedecken kann. Dieses repräsentiert

Familienangehörige, Bekanntschaften etc. welches von Bürger zu Bürger unterschiedlich sein kann.

### **Bewertung**

Im Allgemeinen sollen sich die Bürger alle Reden anhören können. Dies heißt auch zugleich, dass sie sich in Abhängigkeit der eigenen Ethnizität und der Wertausprägung, eine Meinung über diese Rede bilden und diese bewerten können.

### **Diskussion**

Nach der Eigenbewertung durch den Bürger selbst, folgt die Diskussion mit seiner Nachbarschaft. Prinzipiell werden alle fremd-ethnischen nationalen Reden als negativ bewertet, so dass der jeweilige Bürger solche Reden nur mit seiner gleich-ethnischen Nachbarschaft ausdiskutiert. Bei den zivilen oder national-zivilen Reden werden alle Nachbarn unabhängig von der Ethnizität mit in die Diskussion einbezogen. In der Diskussion soll zugleich auch die kollektive Meinungsbildung stattfinden, d.h. dass die kollektive Bewertung der Nachbarschaft in einem gewisse Maße auf die Eigenbewertung Einfluss nimmt und den Bürger entweder zu einer Demonstrationsteilnahme stärkt oder auch abschwächt [23].

### **Demonstration**

Wenn gewisse Bürger eine Rede sehr gut bewerten und zu gleich ihre Nachbarschaft ähnlicher Meinung ist, so nehmen sie an einer Demonstration teil. Eine Demonstration hat die Wirkung, dass sie in Abhängigkeit der Größe, die Wertausprägung der Demonstranten weiter kräftigt bzw. ansteigen lässt. Analog können auch Anti-Demonstrationen entstehen, bei denen die negative Bewertung einer Rede der Auslöser ist.

### **Konfliktalarm**

Damit es zu einem Konfliktalarm kommen kann, muss es, wie in Unterabschnitt 4.2.1 erläutert, eine hohe Unterstützung einer Demonstration von außerhalb geben. Damit es aber zu einer Demonstration kommen kann, muss es Bürger geben, die sich leicht zu einer Masse mobilisieren lassen. Dies kann nur geschehen, wenn die Wertausprägung der Bürger ein gewisses Niveau erreicht. Folglich hängt die Konflikteskalation mit der Wertausprägung der Bürger und deren Anzahl zusammen. Abbildung 4.3 verdeutlicht die Anforderungen, die im Falle eines Konfliktalarms an den Bürger gestellt werden.

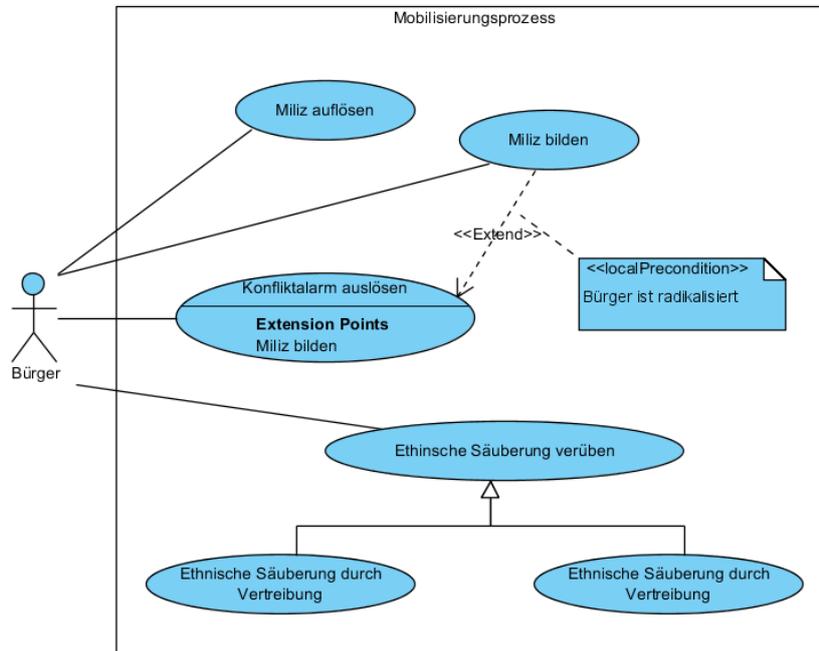


Abbildung 4.3: Use Case Diagramm - Mobilisierungsprozess

### Milizbildung

Wird der zweite Grenzwert des Konfliktalarms, wie in Unterabschnitt 4.2.1 erläutert, überschritten, so kann es zu einer Konflikteskalation kommen. Hierbei haben die Bürger, deren nationale Wertausprägung nach oben und deren zivile Wertausprägung nach unten grenzwertig sind, das Potential sich zu Milizen zusammenzuschließen. Diese können auch als radikalisierte Nationalisten bezeichnet werden [23]. Eine Miliz kann nur in der Republik entstehen, in der auch die Demonstration, welche den Konfliktalarm ausgelöst hat, stattgefunden hat. Der Miliz können sich aber, auch Bürger aus Republik-übergreifenden Regionen anschließen. Tatsächlich gab es im Bosnien-Krieg unter anderem paramilitärische Einheiten, die oft aus Wochenend-Kriegern aus Serbien und aus dem Ausland zusammengesetzt waren [29].

### Ethnische Säuberungen

Die ethnische Säuberung ist das Resultat der Milizbildung. Diese soll unter der lokalen anders-ethnischen Bevölkerung verübt werden. Dabei sucht sich die jeweilige Miliz zuerst willkürlich anders-ethnische Bürger in der direkten Umgebung als Ziel aus. Steht das Ziel fest, kann die ethnische Säuberung entweder durch Mord oder Vertreibung durchgeführt werden [23]. Wird ein Bürger vertrieben, so hat er die Möglichkeit zu Bürgern aus seinem Bekanntschaftsnetzwerk zu flüchten. Demzufolge wird dieser Bürger radikalisiert, was bedeutet, dass sich dieser auch einer Miliz anschließen kann [23].

**Mobilität**

Anders als der Politiker, der an einem Koordinatenpunkt fixiert ist, soll sich der Bürger in seiner Umgebung frei bewegen können. Dabei soll die Bewegung zu einer Demonstration bzw. die kollektive Bewegung einer Miliz zu einem Zielpunkt visualisiert werden. Im Falle einer Vertreibung, muss das Nachbarschaftsnetzwerk wegen des Lokalitätswechsels neu aufgebaut werden. Das Bekanntschaftsnetzwerk aber, bleibt während der Lebendigkeit des Bürgers konstant bestehen.

## 5 Modellierungs- und Simulationswerkzeuge

Dieses Kapitel gibt einen Überblick über die Werkzeuge, die für die Implementierung des Modells verwendet werden. Zuerst wird auf ArcGIS eingegangen, das die statischen Elemente des Modells liefern soll. Danach werden die beiden Werkzeuge EmIL und Repast erläutert die den dynamischen Teil des Modells realisieren sollen. Repast bildet unter anderem die zentrale Schnittstelle des Modells, in der die gesamte Konzeption vereint wird. Hierbei veranschaulicht Abbildung 5.1 die eingesetzten Werkzeuge.

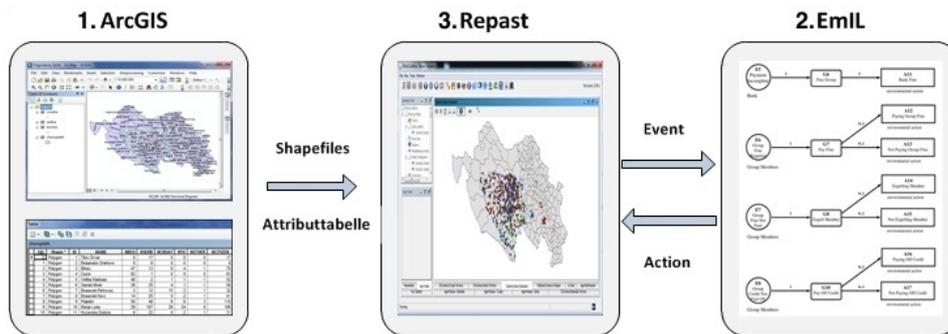


Abbildung 5.1: Modellierungs- und Simulationswerkzeuge

### 5.1 ArcGIS

ArcGIS ermöglicht die Erzeugung raumbezogener Daten und besteht aus einer Reihe von Softwareprodukten, die aus dem Hause ESRI entstanden sind. ESRI bietet unterschiedliche Client- und Serverprodukte an, wobei für die Realisierung des Modells nur die Clientprodukte von Bedeutung sind. Die ArcGIS-Clients, die auch als GIS-Desktop-Produkte bezeichnet werden, werden ArcView, ArcEditor und ArcInfo genannt. Obwohl sie auf derselben Technologie basieren und über dieselben Applikationen verfügen, unterscheiden sie sich in ihrem jeweiligen Funktionsumfang. Die Abbildung 5.2 zeigt die Charakterisierung der Produkte, wobei das höherwertige Produkt den Funktionsumfang der anderen beinhaltet.

ArcGIS-Desktop	Charakteristische Funktionalität
ArcView	Datenzugriff Darstellung Abfrage (Einfaches) Editieren (Einfache) Datenanalyse (Einfaches) Datenmanagement
ArcEditor	+ Komplexeres Editieren
ArcInfo	+ Komplexere Datenverarbeitung + Komplexeres Datenmanagement

↓ steigende Funktionalität

Abbildung 5.2: ArcGIS - Desktop-Produkte [12]

Jedes Desktop Produkt wird mit drei Applikationen ArcMap, ArcToolbox und ArcCatalog zur Verfügung gestellt (siehe Abbildung 5.3). Wenn beispielsweise ArcView zur Erstellung einer Karte benutzt wird, werden diese drei Applikationen verwendet. Es gibt also keine Applikation, die ArcView heißt. Da das Interface bei jedem Desktop-Produkt das gleiche ist, nimmt der Benutzbarkeitseffekt automatisch zu. Für die Erstellung der Karte werden ArcMap und ArcCatalog eingesetzt.

Applikation	Aufgaben
ArcMap	Dient der Visualisierung, Analyse, Editierung und Ausgabe von Geodaten.
ArcCatalog	Ist ein Geodaten-Browser wie Windows-Explorer, dient dem Erzeugen und Anzeigen von Metadaten und der Datenverwaltung.
ArcToolbox	Beinhaltet Werkzeuge zur Datenkonvertierung, Koordinatensysteme und Projektionen.

Abbildung 5.3: ArcGIS - Applikationen [12]

### 5.1.1 Einführung in ArcMap

ArcMap ist eine ArcGIS-Applikation, mit der sich Geodaten erstellen, anschauen, abfragen, editieren und analysieren lassen [12]. Bevor die Erstellung der Karten beschrieben wird, werden die Elemente und die Vokabeln von ArcMap erläutert. Die Abbildung 5.4 zeigt die Benutzeroberfläche von ArcMap mit den zugehörigen Elementen.

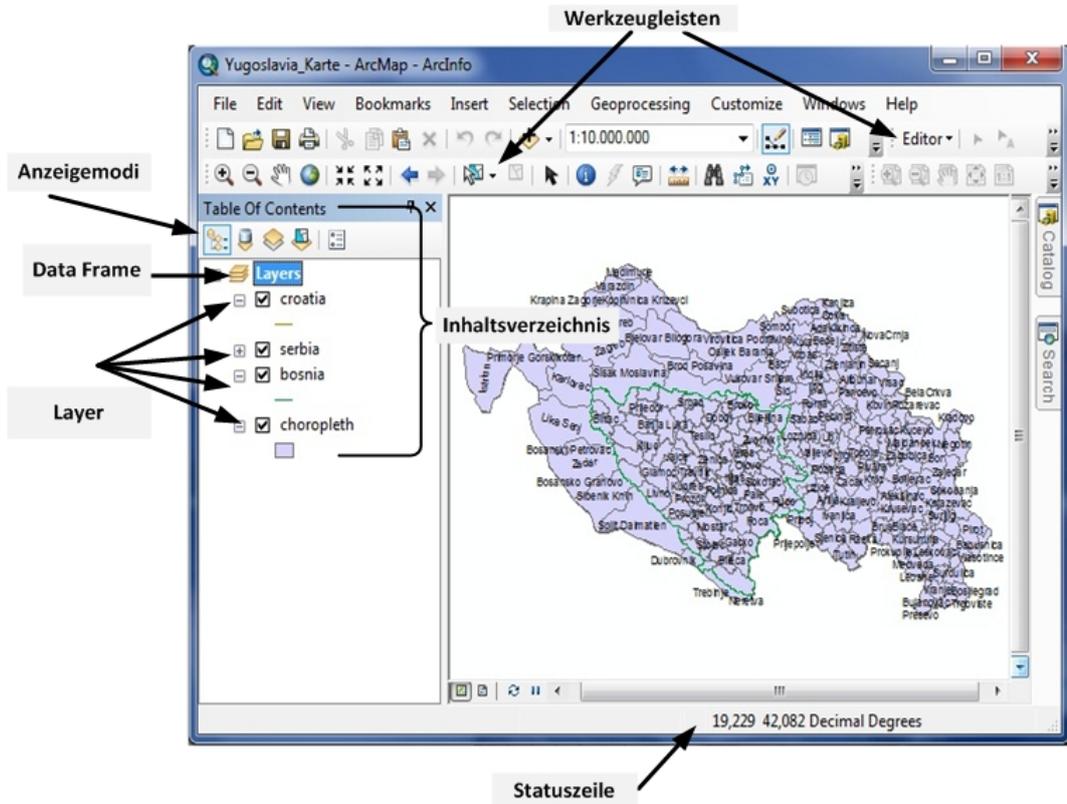


Abbildung 5.4: ArcMap - Oberfläche

#### Elemente der ArcMap-Oberfläche

- **Werkzeuggesteigen:** Sie bilden die gesamte Funktionalität von ArcMap ab. Dabei ist die wichtigste Funktion der Editor, mit dem komplette Karten nachgezeichnet werden können. Die Erstellung der Karten und der Editor werden explizit behandelt.
- **Inhaltsverzeichnis:** Es beinhaltet alle Elemente (Data Frames und Layers) des ArcMap-Projektes. Diese Elemente können mit unterschiedlichen Anzeigemodi aufgelistet werden. Layer können durch das Klicken vom Kontrollkästchen aktiviert bzw. deaktiviert werden. Nach der Aktivierung eines Layers, wird dieser auf der Editieroberfläche angezeigt. Dabei überdeckt der Layer alle Anderen, der sich ganz oben im Inhaltsverzeichnis befindet.

- **Anzeigemodi:** Layer können im Inhaltsverzeichnis in unterschiedlichen Modi dargestellt werden. Sie können beispielsweise nach ihrer Sichtbarkeit oder ihrer Quelle angezeigt werden.
- **Layer:** Jede ArcMap-Karte repräsentiert eine grafische Darstellung geographischer Daten, wobei diese als Layer angezeigt werden. Ein Layer ist als eine Verlinkung auf ein Shapefile (5.1.2) zu sehen. Darin werden die tatsächlichen geographischen Daten nicht gespeichert. Jeder Layer hat einen bestimmten Feature-Typ, beispielsweise eine Staatsgrenze oder einen Fluss (siehe Kapitel 5.1.2).
- **Data Frame:** Layer werden zur besseren Übersicht in Datenrahmen zusammengefasst bzw. gruppiert. Die Gruppierung kann beispielsweise nach dem Autor der Karte oder Zusammengehörigkeit der Inhalte erfolgen.
- **Statuszeile:** zeigt die X- und Y-Koordinate des Mauszeigers in Dezimalgrad.

### 5.1.2 Darstellung-von-GIS-Daten

Ein geographisches Informationssystem (GIS) ist ein System, womit reelle geographische Informationen mit charakteristischen Eigenschaften zur Visualisierungs- und Analysezielen modelliert werden können. Jedes GIS besteht aus folgenden drei GIS-Datenstrukturen:

- Feature-Classes
- Shapefiles
- Attributentabellen

**Feature-Classes:** reelle Objekte werden in ArcMap mit geographischen Features repräsentiert. Dabei können sie als Gebietsunterteilungen (z.B. Länder, politische Bezirke), Bauten (z.B. Straßen, Gebäuden) oder auch als Objekte natürlicher Entstehung (z.B. Flüsse) vorkommen. Die oben genannten Objekte werden mit unterschiedlichen Features dargestellt.

- **Punkte** werden benutzt, wenn Objekte zu klein sind um als Linien oder Polygone dargestellt zu werden. Sie zeigen die Position des Objektes (z.B. Brunnen, Berggipfel, GPS-Koordinaten, siehe Abbildung 5.5).

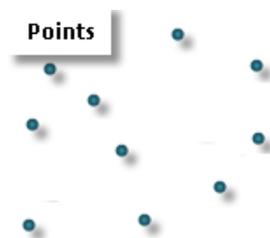


Abbildung 5.5: ArcMap - Features-Punkte [3]

- **Linien** werden dazu benutzt um den Verlauf eines geographischen Objektes darzustellen. (z.B. Staatsgrenzen, Flüsse, Verwaltungsgrenzen, siehe Abbildung 5.6).

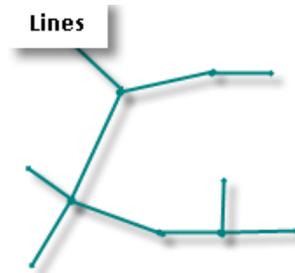


Abbildung 5.6: ArcMap - Features-Linien [3]

- **Polygone** repräsentieren geschlossene Flächen (siehe Abbildung 5.7). Sie werden bei der Darstellung der Form und Position von Regionen oder Verwaltungskreisen eingesetzt.

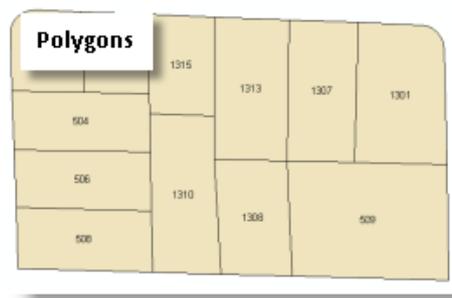


Abbildung 5.7: ArcMap - Features-Polygone [3]

**Shapefiles** beinhalten die tatsächlichen geographischen Informationen. Die Positionen geographischer Objekte werden in einem Shapefile durch Feature-Classes (Punkte, Linien, Polygone) und Attributentabellen dargestellt und beschrieben. Die Attributentabellen werden im gleichen Workspace aber explizit im dBASE-Format gespeichert (siehe Abbildung 5.8), wobei sie mit den Features des Shapefiles zusammengeführt werden. Ein Shapefile kann für die Definition von Features unterschiedliche Koordinatensysteme benutzen. Wenn es nicht vorgegeben ist und “die Koordinaten der Eckpunkte von Features im Bereich -180 bis 180 in X-Richtung sowie -90 bis 90 in Y-Richtung liegen, geht ArcGIS davon aus, dass es sich um geographische Daten handelt und das Datum NAD27 (Bezugssystem für die Landesvermessung) ist.” Die Abbildung 5.8 zeigt einige Shapefiles mit unterschiedlichen Feature-Classes-Symbolen. Shapefiles eignen sich sehr gut für einfache Darstellungen von Attributentabellen und Features. Komplexe Projekte, wo beispielsweise die Topologiebeziehungen, die Attributbeziehungen oder die Koordinatengenauigkeit eine große Rolle spielen, werden von Shapefiles nicht unterstützt. Hierfür müssen Geodatabases verwendet werden.

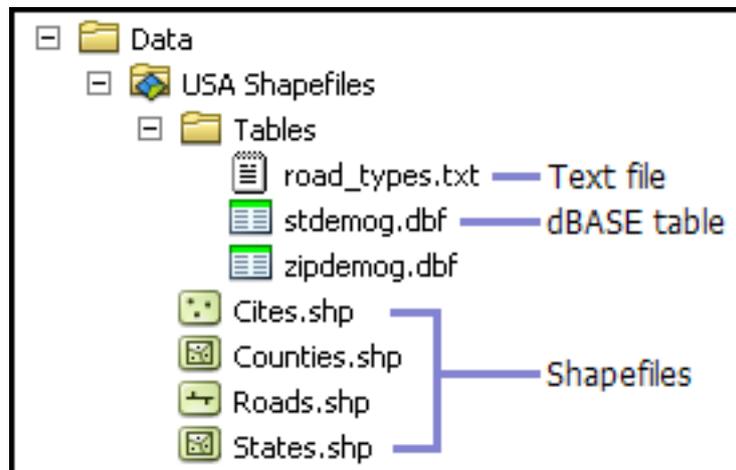


Abbildung 5.8: ArcMap - Shapefiles [3]

FID	Shape *	ID	NAME	NBOS	NSERB	NCROAT	NYU	NOTHER	NCITIZEN	BOSPROC	SERPROC	CROPROC
0	Polygon	1	Třov Drvar	0	17	0	0	0	17	0	100	0
1	Polygon	2	Bosansko Grahovo	0	8	0	0	0	8	0	100	0
2	Polygon	3	Bihac	47	13	6	4	1	70	67	19	9
3	Polygon	4	Cazin	62	1	0	0	0	63	98	2	0
4	Polygon	5	Velika Kladusa	48	2	1	1	1	52	92	4	2
5	Polygon	6	Sanski Most	28	25	4	1	1	58	48	43	7
6	Polygon	7	Bosanski Petrovac	3	12	15	2	1	32	9	38	47
7	Polygon	8	Bosanski Novi	14	25	0	2	1	41	34	61	0
8	Polygon	9	Prijedor	50	48	6	6	3	110	45	44	5
9	Polygon	10	Banja Luka	29	107	29	24	8	189	15	57	15
10	Polygon	11	Kozarska Dubica	6	22	0	2	1	31	69	20	0
11	Polygon	12	Bosanska Gradiska	16	36	3	3	2	58	28	62	5
12	Polygon	13	Laktasi	0	24	3	2	1	29	0	83	10
13	Polygon	14	Srbac	1	20	0	1	1	22	5	91	0
14	Polygon	15	Prnjavor	7	34	2	2	3	45	16	76	4
15	Polygon	16	Derвента	7	23	22	3	1	55	13	42	40
16	Polygon	17	Kljuc	18	19	0	1	0	38	47	50	0
17	Polygon	18	Mrkonjic Grad	3	21	2	1	0	27	11	78	7
18	Polygon	19	Livno	6	4	29	1	0	40	15	10	72
19	Polygon	20	Tomislavgrad	3	1	26	0	0	30	10	3	87

Abbildung 5.9: ArcMap - Attributentabelle

**Attributentabellen** werden zum Speichern und zum Beschreiben von Attributen geographischer Objekte benutzt. Die Zeilen und Spalten der Attributentabelle werden auch Datensätze und Felder genannt. Ein Feld kann immer nur ein bestimmten Datentyp speichern, z.B. eine Zahl, ein Datum oder Text. ArcMap fügt bestimmte Felder wie die eindeutige Kennnummer (Objekt ID) oder Shapes automatisch in die Tabelle hinzu. Attributentabellen spielen bei der Analyse-, Darstellungs- und Datenmanagement-Funktionen eine wesentliche Rolle. Bei-

spielsweise können Gebiete bzw. Städte mit großen Bevölkerungszahlen durch ein Symbol dargestellt werden. In diesem Modell wird mit Hilfe der Attributentabelle die Anzahl der Einwohner getrennt nach Ethnie in einer bestimmten Region verwaltet (siehe Abbildung 5.9). Die Tabelleninformationen können unterschiedliche Quellen haben. Sie können entweder manuell eingegeben werden oder auch aus unterschiedlichen Datenformaten importiert werden. Dabei stehen folgende Formate zur Verfügung:

- **Feature-Layer-Attributentabellen:** werden bei der Erstellung von Feature-Classes automatisch erstellt.
- **dBase-Tabellen:** können zum Erstellen, Lesen, Löschen und Schreiben benutzt werden.
- **INFO:** können zum Erstellen, Lesen, Löschen und Schreiben benutzt werden, wobei dafür ArcInfo oder ArcEditor erforderlich ist.
- **Textdateien:** können nur zum Lesen benutzt werden.
- Viele andere Arten von Tabellen wie beispielsweise Microsoft Excel (nur Lesen).

Darüber hinaus verfügen Attributentabellen mathematische Funktionen, mit denen sich bestimmte Werte berechnen lassen. Es ist auch möglich, die Attributentabellen zur Weiterbearbeitung in unterschiedliche Datenformate zu exportieren.

Die oben genannten GIS-Datenstrukturen bilden zusammen einen Layer, wobei jeder Layer ein bestimmtes Thema für ein gemeinsames geographisches Gebiet enthält. Jedes Thema bzw. jeder Layer kann von anderen unabhängig bearbeitet werden. Layer können beliebig überlagert und zur gemeinsamen Kartenanzeige kombiniert werden (siehe Abbildung 5.10). Dabei können Daten-Layer beispielsweise folgendes beinhalten:

- Staatsgrenzen
- Verwaltungsbereiche
- Gewässer und Flüsse
- Als Mittelachsen dargestellte Straßen, Landnutzungsflächen mit Vegetation, Wohngebieten, Gewerbegebieten usw.
- Flurstückpolygone, die Grundbesitz darstellen
- Oberfläche zur Darstellung von Höhe und Terrain
- Luftaufnahme oder Satellitenbild eines Interessensbereichs

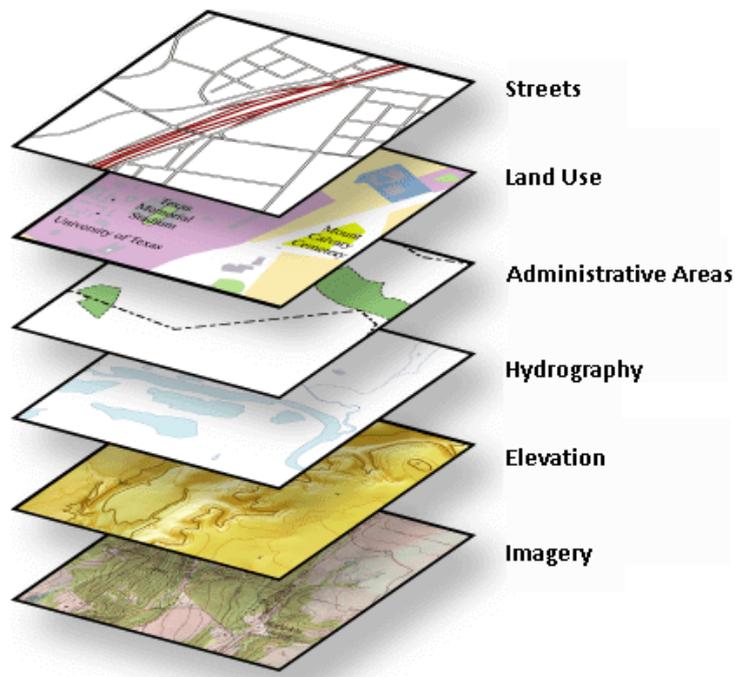


Abbildung 5.10: ArcMap - Layer [3]

### 5.1.3 Editieren von Grafik-Daten (ArcMap)

In ArcMap können die oben genannten Feature-Classes, die in Shapefiles abgespeichert sind, erstellt und bearbeitet werden. Hierbei werden Editierwerkzeuge zur Digitalisierung von geographischen Daten eingesetzt. Dadurch werden Feature-Classes in ein digitales Format konvertiert. Zur Digitalisierung von Geodaten bietet ArcMap unterschiedliche Möglichkeiten. Die gängigste Methode ist die interaktive Digitalisierung am Bildschirm. Hierbei dient eine Karte als Grundkarte, über die die Grenzen nachgezeichnet werden. Als Erweiterung bietet ArcGIS ArcScan an, womit Daten automatisch in Vektor-Features konvertiert werden.

**Starten einer Editiersitzung** Die Vektor-Features oder Attributentabellen können während einer Editiersitzung erstellt oder geändert werden. Dabei kann immer nur ein Datenrahmen in einem Workspace (Ordner mit mehreren Shapefiles) bearbeitet werden. Eine Editiersitzung wird durch das Klicken auf "Editor" gestartet. Da Änderungen einer Editiersitzung Temporär sind, müssen die Änderungen in der Editiersitzung gespeichert werden. Hierbei muss beachtet werden, dass die Speicherung von Kartendokumenten allein nicht ausreicht.

**Erstellen und Bearbeiten von Feature-Classes und Attributentabellen** Nachdem eine Editiersitzung gestartet wurde, wird der Ziel-Layer ausgewählt (siehe Abbildung 5.11), der bearbeitet bzw. geändert werden soll.

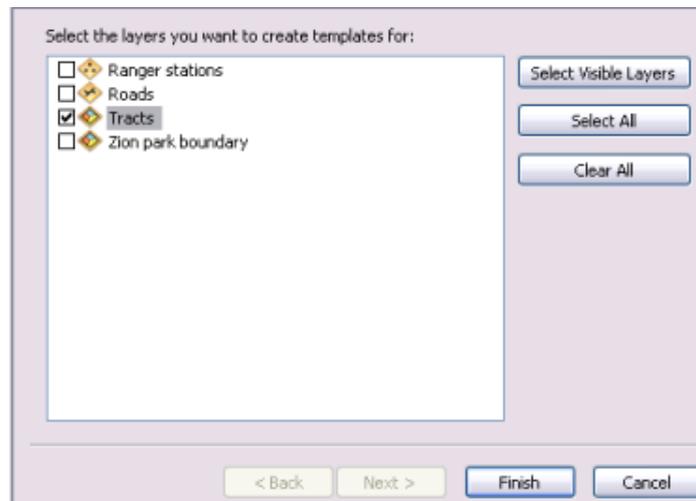


Abbildung 5.11: ArcMap - Das Auswählen eines Shapefiles [3]

Nach der Auswahl des Layers wird das "Features erstellen" Fenster und die Werkzeuge zum Konstruieren von Features-Classes aktiviert. Im oberen Bereich des Fensters befinden sich Vorlagen, Visualisierungseigenschaften und der Layername. Im unteren Bereich werden die Werkzeuge angezeigt (siehe Abbildung 5.12). Hierbei hängt die Verfügbarkeit von Werkzeugen vom Typ der Vorlage ab. Wenn beispielsweise die Linien-Vorlage aktiv ist, wird die Werkzeugleiste mit Linien-Features zur Verfügung gestellt.

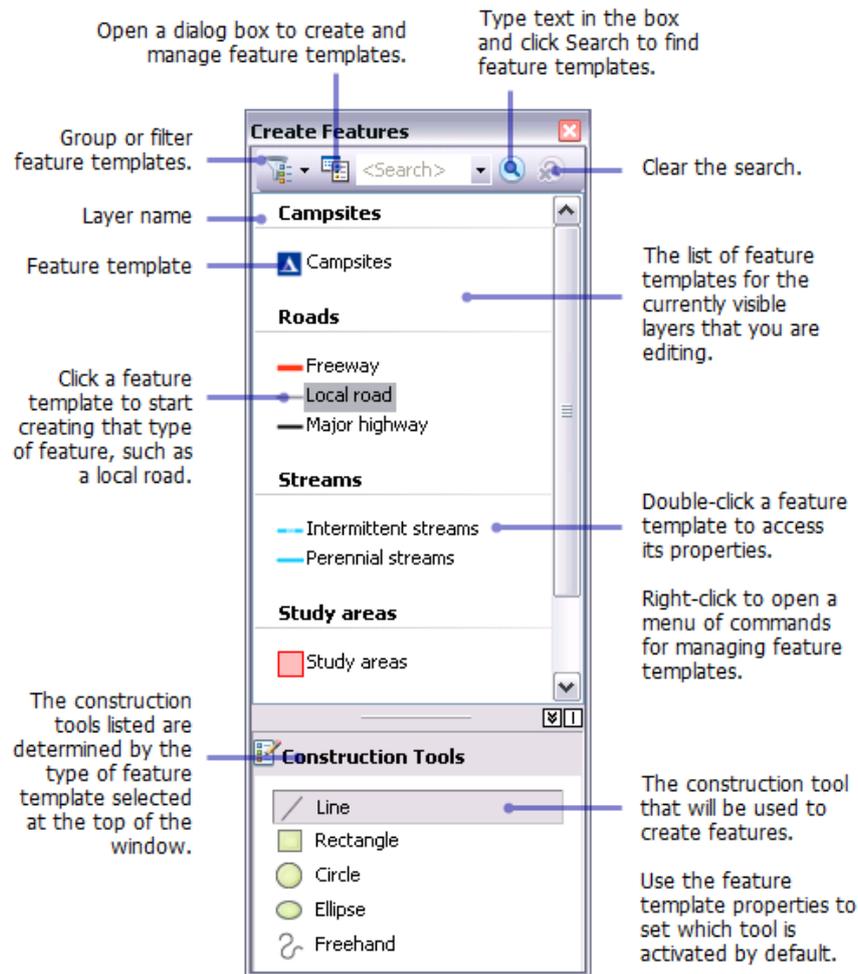


Abbildung 5.12: ArcMap - Fenster zur Features-Erstellung [3]

**Erstellen von Punkten, Linien und Polygonen** Nachdem die Punkt-Vorlage im Fenster ausgewählt wurde, wird das "Punkt" Werkzeug automatisch aktiviert. Danach kann ein Punkt auf der Karte in einer beliebigen Stelle gezeichnet werden. Für die Erstellung von Linien und Polygonen wird die Linien- bzw. Polygon-Vorlage ausgewählt, wobei sich beide ähnlich verhalten. Wenn eine Vorlage ausgewählt ist, wird der Werkzeugsatz automatisch mit den passenden Funktionen ausgestattet (siehe Abbildung 5.13).

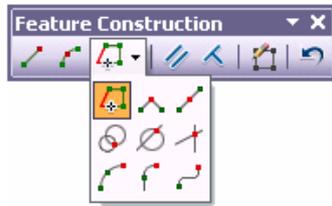


Abbildung 5.13: ArcMap - Werkzeugsatz [3]

Eine Zeichnung wird durch Segmente und Stützpunkte dargestellt. Segmente sind Linien, die Stützpunkte verbinden. An den Stützpunkten ändern die Linien ihre Richtung, wobei sie während der Zeichnung mit roten und grünen Punkten angezeigt werden (siehe Abbildung 5.14).

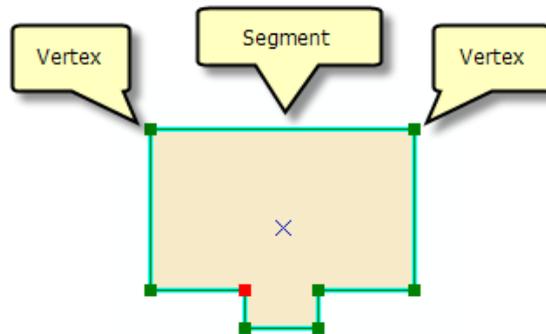


Abbildung 5.14: ArcMap - Stützen und Segmente [3]

Nach der Zeichnung eines Objekts, muss die Zeichnung beendet werden, damit diese als Feature übernommen werden kann. Dafür muss der rote Punkt mit dem Mauszeiger doppelklickt werden (siehe Abbildung 5.15).

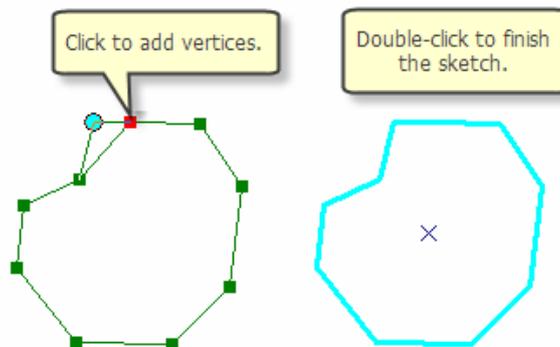


Abbildung 5.15: ArcMap - Beenden einer Zeichnung [3]

**Auto-Complete Polygon (Polygon Automatisch Schließen)** Die Linien bzw. Polygone mit denen Grenzen und Verwaltungsgebiete dargestellt werden, grenzen sich aneinander. Um Überlappungen oder Lücken zwischen den Grenzen zu vermeiden, stellt ArcMap das Werkzeug “Polygon Automatisch Schließen” zur Verfügung. Wenn beispielsweise die Grenzen eines Gebietes schon gezeichnet worden sind, kann der Mauszeiger nach der Aktivierung dieser Funktion über diese Linien bewegt werden und die neuen Linien werden an die Vorhandenen automatisch angegrenzt.

#### 5.1.4 ArcCatalog

ArcCatalog wird von ArcGIS zur Verwaltung und Organisierung von unterschiedlichen Typen geographischer Informationen zur Verfügung gestellt. Die Anwendung ähnelt dem Windows-Explorer (siehe Abbildung 5.16), wobei nur die Daten angezeigt werden, die für ArcGIS-Relevant sind. Im Katalog Fenster werden Geodatabases und GIS-Server verwaltet und GIS-Inhalte organisiert. Ferner können neue Shapefiles, Layer, dBASE Tabellen durch ArcCatalog zur ArcMap-Karte hinzugefügt werden. Nachdem ein Shapefile beispielsweise im Kontextmenü hinzugefügt wurde, wird der Name vergeben und eine Feature-Class ausgewählt (Punkt, Linie, Polygon). Dieses Shapefile kann dann in ArcMap bearbeitet werden.

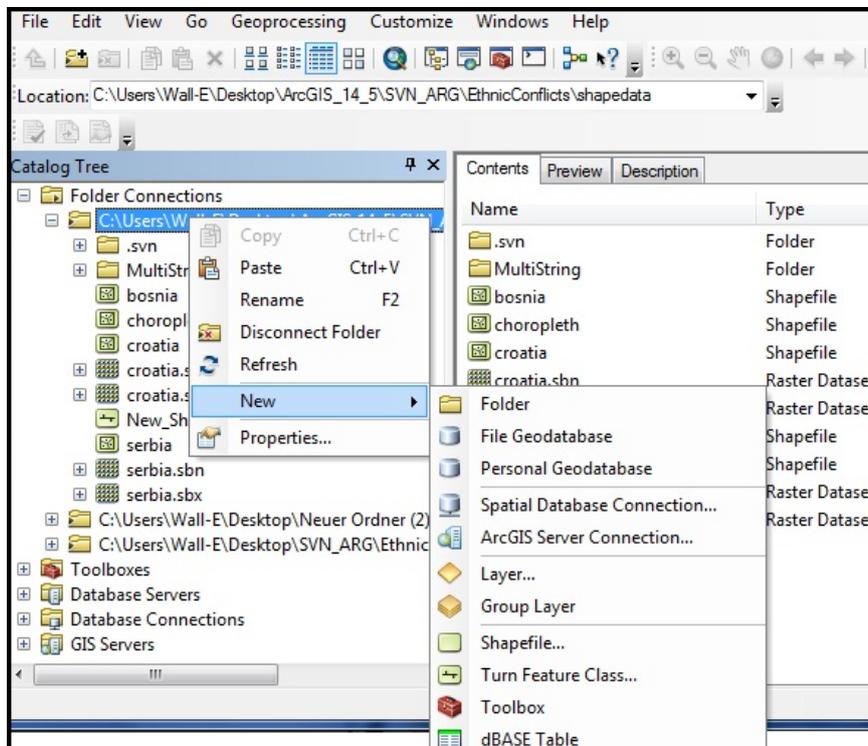


Abbildung 5.16: ArcCatalog

Zusammenfassend ist ArcCatalog zur Strukturansicht bzw. Strukturschaffung eines ArcMap-Projektes eine wichtige Anwendung, womit auch ein GIS-Element ausgewählt, mit dessen Eigenschaften angezeigt und bearbeitet werden kann.

## 5.2 EmIL

Der Kern des EmIL (Emergence in the loop) Frameworks ist das normative Verhalten der Agenten. Die Erfahrungen eines Agenten während der Interaktion mit anderen Agenten sollen in die anschließende Entscheidungsfindung wieder einfließen [13]. Hierfür wird die Softwarekomponente EmIL-S benötigt welche nicht eigenständig benutzt werden kann, sondern eine Simulationsumgebung benötigt. Diese Simulationsumgebung muss die jeweiligen Ereignisse zuerst aufrufen, damit EmIL diese anschließend bewerten kann [25].

Die Basis für die Entscheidungen von Agenten bildet ein Regelsatz IRB (Initial Rule Base), welcher sich in Form von Event-Action-Trees visualisieren lässt. Ein solcher Regelsatz besitzt Ereignisse, Aktionsgruppen und Aktionen die durch Kanten verbunden sind und zu Beginn initiale Wahrscheinlichkeiten besitzen. Ereignisse werden von der Simulationsumgebung aufgerufen. Aktionsgruppen fassen definierte Aktionen zusammen, die die jeweiligen Entscheidungsmöglichkeiten darstellen. Diese Aktionen werden an die Simulationsumgebung zurückgeliefert und von ihr ausgeführt.

Alle aufgerufenen Events werden von EmIL in einem "Event Board" gespeichert. In diesen Event Board wird die Event Message mit dem Action Tree, dem Zustand des Agenten und einem TimeStamp abgespeichert. Zusätzlich werden Schlüsseleinträge berechnet die während der Simulationen einen Vergleich von verschiedenen Abschnitten des Event Boards ermöglichen.

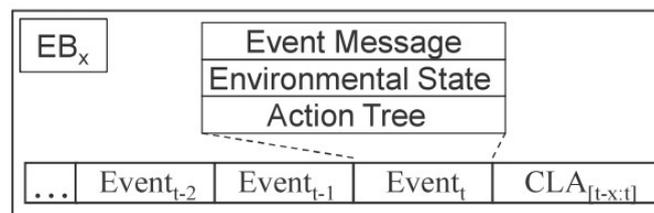


Abbildung 5.17: EmIL Event Board

Durch diese Abspeicherung im Event Board können Normen abgeleitet werden und so anstehende Entscheidungen in Abhängigkeit von bereits gemachten Erfahrungen berechnet werden [25].

## 5.3 Repast

Das Rekursive Porous Agent Simulation Toolkit ist ein open-source agentenbasiertes Modellierungs- und Simulationswerkzeug. Das Repast wurde in ver-

schiedene Sprachen implementiert, unter anderem gibt es ein C++ basiertes System welches speziell für große Cluster und Supercomputer entwickelt wurde [4]. Für das Konflikt-Modell wird die Java-basierte Version mit der Bezeichnung Repast Symphony eingesetzt. Repast eignet sich besonders für die Anwendung im soziologischen Bereich wie z. B. bei der Simulation von Verhaltensweisen verschiedener Akteure. Verglichen mit anderen agentenbasierten Modellierungstools wie NETLOGO, Starlogo, Swarm etc., ist RepastS ein Modellierungstool mit der höchsten Modellierungsstärke [20].

### 5.3.1 Kontext

Das Kernkonzept von Repast ist der Begriff des Kontext. Ein Kontext ist eine Datenstruktur, welche als Sammelbecken für Objekte eines Modells dient. Dabei können Kontexte hierarchisch aufgebaut sein, so dass Agenten aus dem untergeordneten Kontext sich auch im übergeordneten Kontext wiederfinden. Durch Kontexte können bestimmte Objekte gruppiert werden, aber auch von anderen Objekten gezielt getrennt werden. Ferner unterstützen Kontexte auch Projektionen. [30]

### 5.3.2 Projektionen

Projektionen realisieren eine Struktur zwischen den Agenten. Diese Struktur erlaubt es den Agenten, in einer räumlichen Beziehung mit anderen Agenten zu treten [9]. Diese Projektionen werden an die Kontexte angebracht, so dass jeder Agent bzw. jedes Objekt aus dem Kontext in der Projektion sichtbar ist. Repast hat vier Arten von Projektionen die zu verschiedenen Zwecken eingesetzt werden können. Dazu zählen kontinuierliche räumliche Projektionen, GIS Projektionen, Zellulare Projektionen und Netzwerk Projektionen [1]. Für das Konflikt-Modell wurden konkret die GIS Projektionen eingesetzt. Diese geographischen Projektionen sind Räume, in denen Agenten durch eine räumliche Geometrie verbunden sind. Typischer Weise bestehen diese GIS Projektionen aus mehreren Layern auf denen eine bestimmte Anzahl von Elementen eingebunden ist. Diese Elemente besitzen neben den geographischen Koordinaten auch Eigenschaften bzw. Attribute mit denen sie assoziiert werden können [11]. Diese GIS Daten können auf die Repast Ebene transferiert werden, d.h. Elemente die mit bestimmten Eigenschaften auf den Layern abgebildet sind, werden zu Objekten mit Eigenschaften auf der Repast Ebene. Für die topologischen Kalkulationen im GIS Raum wird die JTS (Java Topology Suite) eingesetzt. Die JTS ist eine API mit einer Reihe von geometrischen Algorithmen die für die agentenbasierte Modellierung in GIS Räumen unverzichtbar ist und sehr viel Zeit erspart.

## 6 Implementierung

Dieses Kapitel behandelt die Implementierung des in Kapitel vier vorgestellten Konzepts mit Hilfe des in Kapitel fünf beschriebenen Modellierungs- und Simulationswerkzeugs. Zuerst wird die Erstellung der geospatialen Daten mit ArcGIS beschrieben. Danach wird auf die Realisierung des normativen Verhaltens durch EMIL eingegangen. Schließlich bildet die Implementierung des Rumpfes den abschließenden Teil dieses Kapitels.

### 6.1 ArcGIS

Für die Implementierung werden aus der ArcGIS Umgebung nur die folgenden zwei Tools benötigt:

- ArcCatalog zur Erstellung und Verwaltung der Layer.
- ArcMap zur Zeichnung von Shapefiles und Bearbeitung der Attributentabelle.

In Kapitel fünf wurden die obengenannten Anwendungen ausführlich erklärt. In diesem Unterabschnitt werden die Implementierungsschritte der Landkarte, die in der Simulation verwendet wird, konkret beschrieben.

#### 6.1.1 Aufbau der Karte

Die Karte besteht aus zwei Ebenen, wobei die Erste die territorialen Grenzen der Republiken bzw. der aktuellen Staaten (siehe Abbildung 6.1) und die Zweite die Verwaltungskreise der jeweiligen Republiken darstellen. Dieses Zwei-Ebenen-Modell wird für folgende Zwecke benötigt:

- **Staatsgrenzen:** Staatsgrenzen werden für die eindeutige geographische Zuordnung der Politiker zu einem Staat benötigt.

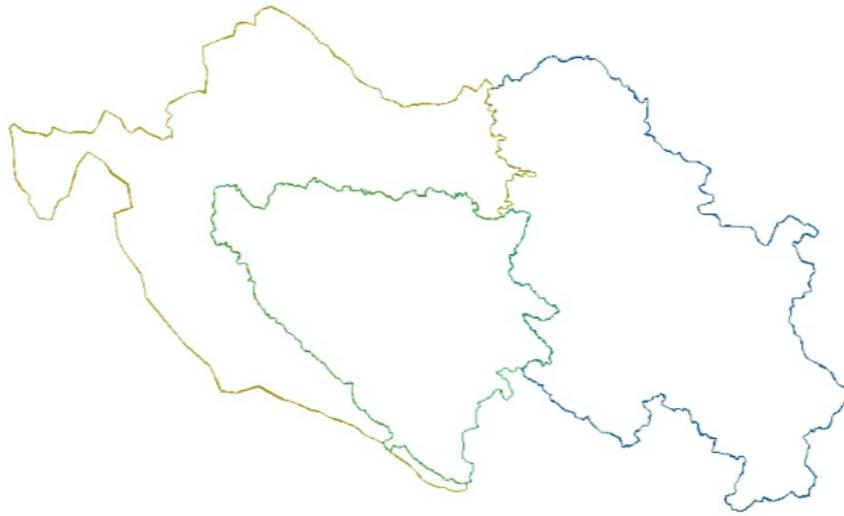


Abbildung 6.1: Staatsgrenzen von Serbien, Bosnien und Kroatien

- **Verwaltungskreise:** Da statistische Daten über die Verteilung der Ethnien nach Verwaltungskreisen im damaligen Jugoslawien vorliegen, können diese eingesetzt werden, um eine genaue Repräsentation der Bevölkerung zu realisieren. Für Bosnien und Herzegowina und Serbien wurden die Einwohnerdaten von der Ebene der Kreise (Opstinas) benutzt. Für Kroatien wurden die Daten der nächsthöheren Ebene benutzt, nämlich die der Gespanschaften (Zupanije). Auf der realisierten Karte befinden sich 174 Kreise für Serbien, 105 Kreise für Bosnien und 21 Gespanschaften für Kroatien (siehe Abbildung 6.2).



Abbildung 6.2: Verwaltungskreise von Serbien, Bosnien und Kroatien

### 6.1.2 Erstellung der Shapefiles mit ArcCatalog

Die Karte besteht aus vier Shapefiles welche zwei Ebenen bilden. Bevor die Karte mit entsprechenden Features gezeichnet werden kann, müssen die Shapefiles mit Hilfe des ArcCatalog generiert werden. Nach der Generierung werden sie im Inhaltsverzeichnis von ArcMap angezeigt und können dann bearbeitet werden. Die Shapefiles Serbien, Bosnien und Kroatien bilden die erste Ebene und beinhalten die Staatsgrenzen. Die Choroplethenkarte bildet die zweite Ebene und beinhaltet die oben genannten Verwaltungskreise.

### 6.1.3 Zeichnung der Landkarte mit ArcMap

Um ein Shapefile in einem ArcMap-Projekt bearbeiten zu können, muss es im Inhaltsverzeichnis hinzugefügt werden (siehe Abbildung 6.3). Die Grenzen wurden so gezeichnet, in dem die jeweiligen Karten der Republiken in Form von Grafiken in den Hintergrund gelegt und die Grenzen nachgezeichnet wurden.

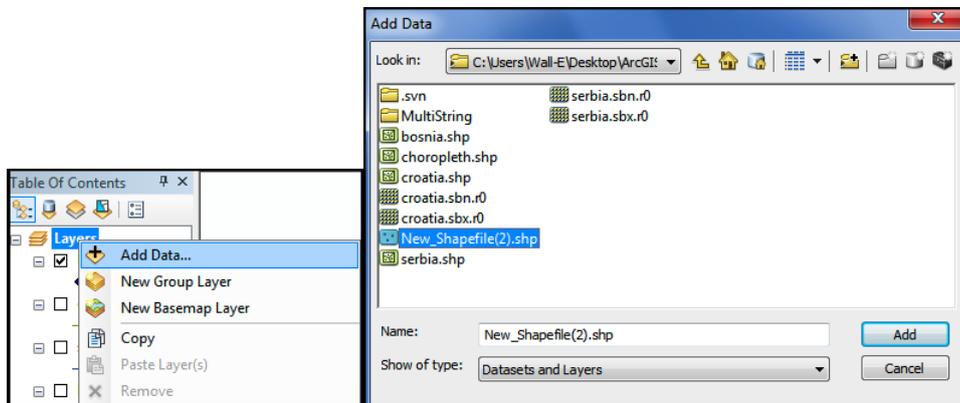


Abbildung 6.3: ArcMap - Shapefile Hinzufügen

Für das Erstellen der Karten muss die Editier-Sitzung aktiviert werden. In dieser Sitzung kann auch die Attributentabelle aktualisiert werden. Nachdem die Grenzen eines Staates nachgezeichnet wurden, kann z. B. die “Auto Complete Polygon”-Funktion verwendet werden (siehe Abbildung 6.4), um die gezeichnete Grenze zum angrenzenden Staat zu übernehmen. Das “Auto Complete Polygon” stellt zwei wichtige Funktionen zur Verfügung:

- **Straight Segment:** dadurch werden einfache gerade Linien gezeichnet. Die Grenzen bzw. Verwaltungskreise, deren angrenzenden Kreise noch nicht nachgezeichnet wurden, werden mit dieser Funktion realisiert (siehe Kapitel 5).
- **Trace:** wird verwendet, wenn ein Polygon bzw. Verwaltungskreis angrenzende Verwaltungskreise hat, die bereits nachgezeichnet wurden. Hiermit werden die bereits gezeichneten Grenzen für das neue Polygon einfach übernommen. Diese Funktion ist in Abbildung 6.4 als eine dunkelgrüne Linien zu sehen.

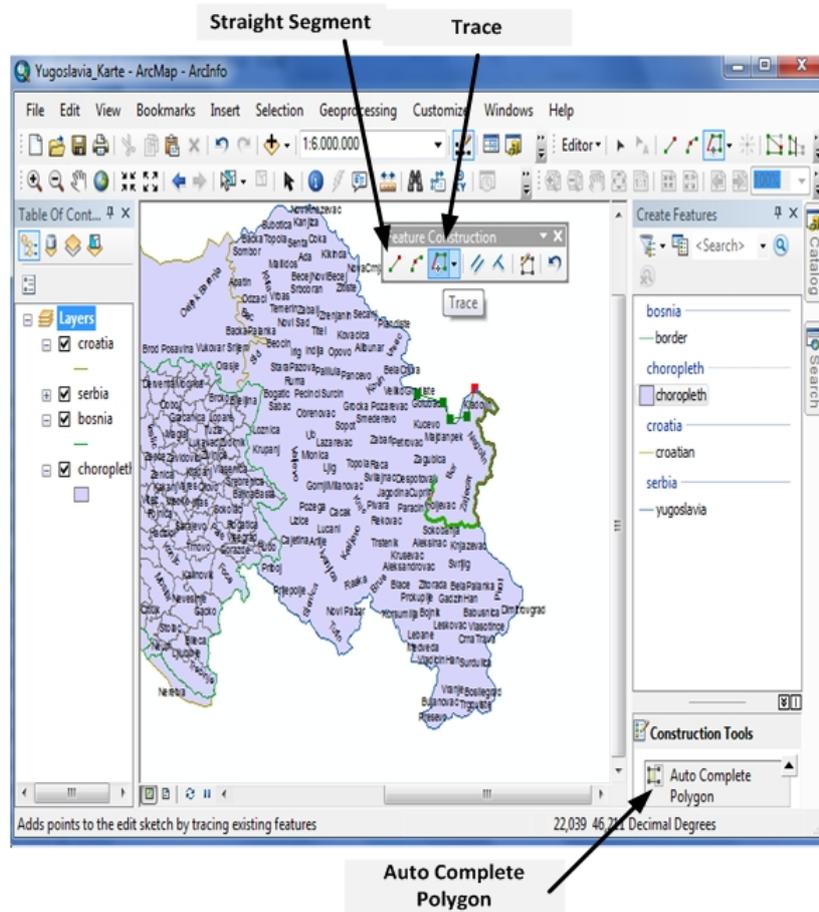


Abbildung 6.4: ArcMap - Auto Complete Polygon

#### 6.1.4 Erstellung und Bearbeitung der Attributentabelle mit empirischen Daten

Immer wenn ein Polygon fertig gezeichnet wird, erfolgt automatisch ein Eintrag in die Attributentabelle. (siehe Abbildung 6.5). In diesem Tabelleneintrag können bestimmte Eigenschaften zu dem gezeichneten Polygon hinterlegt werden. Abbildung 6.5 zeigt einen solchen Eintrag.

	FID	Shape *	Id	I
▶	0	Polygon	0	0

Abbildung 6.5: ArcMap - Eintrag in die Attributentabelle

Attributentabellen können mit beliebig vielen Spalten erweitert werden. Dadurch können z. B. den Polygonen bestimmte Werte zugewiesen werden. Die Choroplethenkarte in welcher die Verwaltungskreise nachgezeichnet wurden, beinhaltet die Attributentabelle (siehe Abbildung 6.6) die für die repräsentative Bevölkerungsverteilung benötigt wird. Die Attributentabelle der Choroplethenkarte besitzt folgende Spalten:

- NAME: Bezeichnung des Verwaltungskreises.
- NBOS: Anzahl der Bosniaken im jeweiligen Verwaltungskreis.
- NSERB: Anzahl der Serben im jeweiligen Verwaltungskreis.
- NCROAT: Anzahl der Kroaten im jeweiligen Verwaltungskreis .
- NYU: Anzahl Jugoslawen im jeweiligen Verwaltungskreis.
- NOTHER: Anzahl anderer ethnischer Gruppen im jeweiligen Verwaltungskreis.
- NCITIZEN: Gesamtzahl der Einwohner im jeweiligen Verwaltungskreis.

FID	Shape *	ID	NAME	NBOS	NSERB	NCROAT	NYU	NOTHER	NCITIZEN
0	Polygon	1	Titov Drvar	0	17	0	0	0	17
1	Polygon	2	Bosansko Grahovo	0	8	0	0	0	8
2	Polygon	3	Bihac	47	13	6	4	1	70
3	Polygon	4	Cazin	62	1	0	0	0	63
4	Polygon	5	Velika Kladusa	48	2	1	1	1	52
5	Polygon	6	Sanski Most	28	25	4	1	1	58
6	Polygon	7	Bosanski Petrovac	3	12	15	2	1	32
7	Polygon	8	Bosanski Novi	14	25	0	2	1	41
8	Polygon	9	Prijedor	50	48	6	6	3	110
9	Polygon	10	Banja Luka	29	107	29	24	8	189
10	Polygon	11	Kozarska Dubica	6	22	0	2	1	31

Abbildung 6.6: Attributentabelle - Choroplethenkarte

Die statistischen Daten, die von den staatlichen Statistikämtern zu Verfügung gestellt wurden, stammen aus dem Jahr 1991. Im folgenden Beispiel ist ein Auszug aus den statistischen Daten von Serbien aufgeführt (Abbildung 6.7). Pfeil eins zeigt auf die Summe der Einwohner im jeweiligen Verwaltungskreis (Opština) an. Ferner zeigt Pfeil zwei auf die Anzahl der Serben, Pfeil drei auf die Anzahl der Jugoslawen und Pfeil vier auf die Anzahl der Muslime. Wie in Unterabschnitt 3.2 aufgeführt und erklärt, wurden hier die Bosniaken als Muslime aufgeführt. Letztlich zeigt Pfeil fünf auf den Namen des Verwaltungskreises.

БОР	← 5	59900	49391	597	1834	11	-	37	1103	205
ГРАДСКА		40668	34153	538	1544	11	-	34	1013	190
ОСТАЈА		19232	15238	59	290	-	-	3	90	15
БОР	Г	40668	34153	538	1544	11	-	34	1013	190
БРЕСТОВАЦ		3140	2737	38	62	-	-	2	38	7
БУЧЈЕ		789	659	1	1	-	-	-	1	-
										36

703. СТАНОВНИШТВО ПРЕМА НАЦИОНАЛНОЈ ПРИПАДНОСТИ

ДЕО 1 СРЕБИЈА - ЦЕНТРАЛНА СРЕБИЈА

	1	2	3	4	9	10	11	12	13
	УКУПНО	СРБИ	ЦРНО-ГОРЦИ	ЈУГО-СЛОВЕНИ	ГРЦИ	ЈЕВ-РЕЈИ	МАЂАРИ	МАКЕ-ДОНЦИ	МУСЛИ-МАНИ
ГОРЂАНЕ	1446	1261	-	13	-	-	-	5	-
Д. БЕЛА РЕКА	1049	977	1	4	-	-	-	-	-
ЗЛОТ	4280	3479	13	61	-	-	1	8	-
КРИВЕЉ	1586	900	1	27	-	-	-	2	1
ЛУКА	702	312	-	7	-	-	-	-	-

Abbildung 6.7: Einwohnerzahlen nach Ethnizitäten [24]

Im Bezug auf die Einwohnerzahlen würde es sehr viel Rechenleistung erfordern, wenn pro Einwohner ein Bürger-Agent realisiert werden würde. Das Konflikt-Modell soll jedoch ein vereinfachtes Modell darstellen, so dass für die Darstellung eine repräsentative Menge von Agenten gewählt wurde. Dafür wurde Folgendes festgelegt:

- Pro 1.000 Einwohner ein Bürger-Agent
- Pro 200.000 Einwohner ein Politiker-Agent

### 6.1.5 Behebung von Referenzfehlern für Layer

Die Layer referenzieren Shapefiles. Wenn der Name des Shapefiles geändert wird oder der Ordner, in dem sich das Shapefile befindet, verschoben wird, kann das Map-Dokument nicht vollständig geöffnet werden. Dies wird im Inhaltsverzeichnis mit einem roten Ausrufezeichen neben dem Layernamen angezeigt (siehe Abbildung 6.8). Wenn dies der Fall ist, muss der neue Ordner bzw. Name explizit angegeben werden. Dazu wird der Layer, für den die Datenquelle angegeben werden muss, mit der rechten Maustaste geklickt [[12],25]. Danach werden die Layereinstellungen und die Registerkarte Source ausgewählt [[12],25]. Zum Schluss wird unter "Set Data Source" die Datenquelle angegeben (siehe Abbildung 6.8).

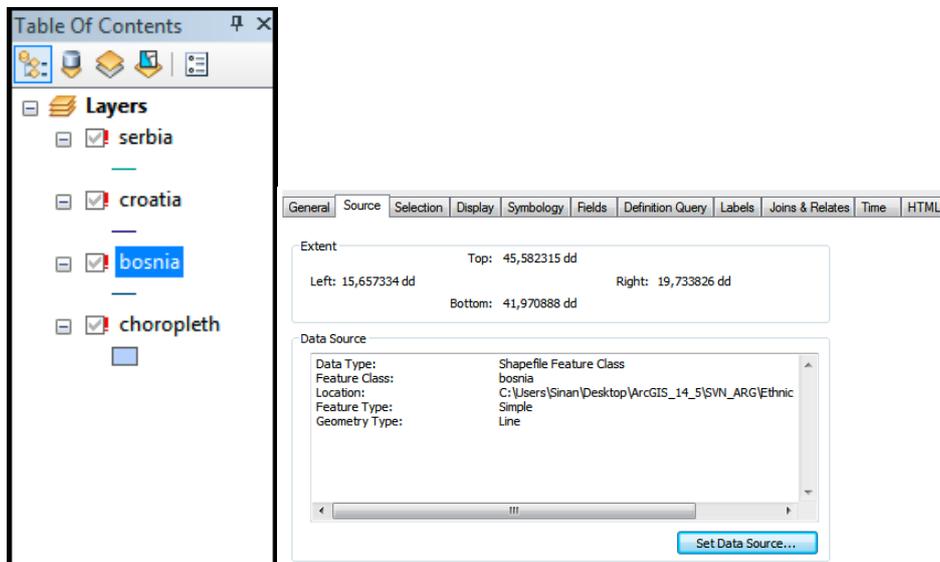


Abbildung 6.8: ArcMap - Referenzfehler

## 6.2 EmIL

Wie in Unterabschnitt 4.1 beschrieben, soll der Politiker ein normatives Verhalten bei der Auswahl seiner Reden zeigen. Der Politiker kann hierbei zwischen drei unterschiedlichen Reden wählen, die sich als Regelsatz in einem Event-Action-Tree abbilden lassen (Abbildung 6.9).

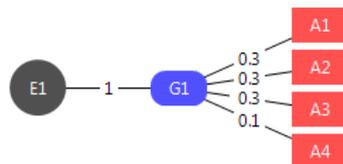


Abbildung 6.9: Event-Action-Tree

Das Ereignis E1 steht hierfür, dass ein bestimmter Politiker eine Rede halten will. Die Aktionsgruppe G1 fasst dabei vier mögliche Aktionen zusammen:

- A1 - nationale Rede halten
- A2 - zivile Rede halten
- A3 - national-zivile Rede halten
- A4 - keine Rede halten

Die Wahrscheinlichkeiten zwischen den drei Rede-Typen sind gleich verteilt. Dies ist der initiale Regelsatz, der sich während der Simulation in Abhängigkeit der Bewertungen ändern kann. Für die Verarbeitung durch das EmIL Framework wird der oben definierte Event-Action-Tree in ein XML File abgebildet:

```

<agent name="Politician" id="Politician">
<initial-rule-base>
  <event-action-tree name="E1" description="E1">
    <event name="E1"/>
    <action-group name="G1" probability="1" description="G1">
      <action name="A1" probability="0.3" description="A1"/>
      <action name="A2" probability="0.3" description="A2"/>
      <action name="A3" probability="0.3" description="A3"/>
      <action name="A4" probability="0.1" description="A4"/>
    </action-group>
  </event-action-tree>
</initial-rule-base>

```

Abbildung 6.10: Aufbau von parameters.xml

Die Anbindung von EMIL wird in Repast in der Erzeugerklassen der Kontexte eingebunden. Hierfür wird der EmIL-Controller mit der jeweiligen IRB aufgerufen.

```

Controller.initializeController(new File("../EthnicConflicts/emil/config/
EthnicConflicts.xml"));

```

Abbildung 6.11: Einbindung ins Repast

Des Weiteren ist es dann nur noch erforderlich, die Objekte der Repast-Ebene mit den Agenten der EmIL-Ebene zu referenzieren. Hierfür wird bei der Erzeugung der Politiker-Agenten auf der Repast-Ebene die Methode *setID(int id)* aufgerufen, die den Agenten eine ID vergibt. Zugleich aber wird auch das jeweilige Abbild des Politiker-Agenten der EmIL-Ebene hinzugefügt.

```

public void setID(int id){
  this.ID=id;
  emilLayer = emil.Controller.getInstance().addAgent("Politician",
  this.ID, this);
}

```

Abbildung 6.12: Abbildung der Agenten auf der EmIL-Ebene

### 6.3 Repast

Repast bildet das Bindeglied zwischen ArcGIS und EmIL. Zum Einen nutzt es das EmIL Framework, um das normative Verhalten von Politikern umzusetzen und zum anderen integriert es die in ArcGIS erstellte Karte in die jeweilige Projektion des Modells. Repast verwaltet unter anderem auch zwei Files die besonders wichtig sind:

- parameters.xml
- context.xml

Dabei handelt es sich um zwei XML-Files welche die Struktur des jeweiligen Anwendungsbereiches verwalten.

### 6.3.1 Parameter

Bei dem File *parameters.xml* handelt es sich um die Simulationsparameter, die beim Starten der Simulation in dem "Parameter View" angezeigt werden. Diese Parameter können benutzerspezifisch direkt in der GUI angepasst werden, ohne dabei den Quellcode ändern zu müssen. Dies gibt dem jeweiligen Nutzer die Möglichkeit, besonders die Zusammenhänge und Auswirkungen zwischen den einzelnen Parametern zu erkennen. Letztendlich wird dadurch das Verhalten des Systems beeinflusst, so dass die Parameter auch zur empirischen Kalibrierung des Modells benutzt werden können. Die Struktur von *parameters.xml* ist folgendermaßen aufgebaut:

```
<parameters>
  <parameter name="agentRadius" displayName="Agent_Radius" type="double"
    defaultValue="0.01" isReadOnly="false"
    converter="StringConverterFactory$DoubleConverter" />
</parameters>
```

Abbildung 6.13: Aufbau von parameters.xml

Abbildung 6.13 stellt den Aufbau des XML Files dar. Das File besteht aus dem Wurzelement "<parameters>" und mehreren Parameter-Definitionen "<parameter>", die durch Attribute beschrieben werden. Abbildung 6.14 illustriert ein Ausschnitt aus der GUI mit einigen Simulationsparametern.

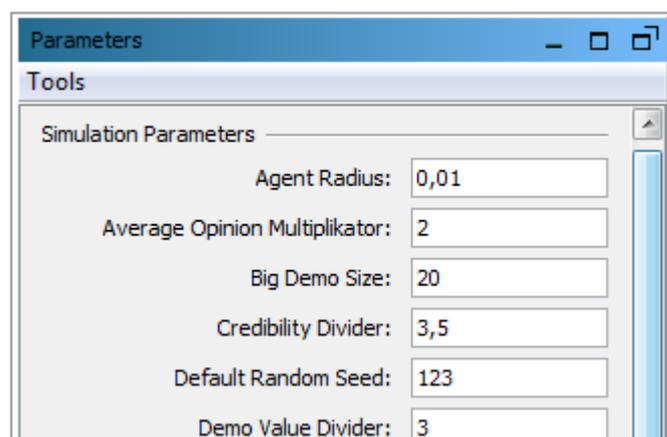


Abbildung 6.14: GUI Parameter View

Unter anderem sollte ein Modell konzipiert werden, welches eine gewisse Flexibilität besitzt. Diese Flexibilität wird genau durch die Vielzahl an Anpassungsmöglichkeiten durch die Simulationsparameter gegeben. Tabelle 4 listet diese Parameter mit ihrem Standardwert auf:

Parameter	Standardwert
Default Random Seed	123
Agent Radius	0,15
Moore Neighborhood Range	3
Average Opinion Multiplikator	3
Credibility Divider	3,5
Synchronous Speeches	10
Opinion Research Strength Divider	2,5
Demo Value Divider	7
Threshold For Demo Decission	2,1
Small Demo Size	40
Middle Demo Size	30
Big Demo Size	20
Mass Demo Size	10
Threshold Conflict Alarm Stage1	10
Sanction Strength For Ambassador	-0,1
Threshold Conflict Alarm Stage2	25
Max Militia Size	5
Militia Threshold	1
Neighbor Expulsion	0,3

Tabelle 4: Simulationsparameter

**Default Random Seed**

Der Default Random Seed gibt den Seed für den Zufallszahlengenerator vor. Für die Reproduzierbarkeit einer Simulation sollte dieser Wert beibehalten werden.

**Agent Radius**

Der Agent Radius ist der Radius eines Bürgers bei der Suche nach seinen Nachbarn. Alle anderen Bürger, die sich innerhalb dieses Radius befinden, gelten als seine Nachbarn.

**Moore Neighborhood Range**

Anders als bei den zellulären Automaten, dient hierbei die Moore Nachbarschaft nur als Grenzwert für die Anzahl der Nachbarn eines Agenten. Der Wert des Parameters besagt, dass es sich um die x-fache Moore Nachbarschaft handelt und das beim Erreichen dieses Wertes die Suche nach den Nachbarn abbricht.

**Average Opinion Multiplikator**

Der Average Opinion Multiplikator ist der Gewichtungsfaktor der Summe der Bewertungen einer Rede der Nachbarn eines Agenten. Hiermit kann die Gewichtung der kollektiven Meinung gegenüber der eigenen Meinung verstärkt oder vermindert werden.

**Credibility Divider**

Der Credibility Divider ist der Abschwächungsfaktor einer Sanktion, die auf Grund der Unglaubwürdigkeit einer Rede an den jeweiligen Politiker gesendet wird.

**Synchronous Speeches**

Dieser Parameter begrenzt die Anzahl der parallelen Reden auf dem Niveau der Republiken. Dies heißt konkret, dass zu einem Zeitpunkt pro Republik nur eine Anzahl von Politikern eine Rede halten können die kleiner ist als "Synchronous Speeches".

**Opinion Research Strength Divider**

Hierbei handelt es sich um den Abschwächungsfaktor der Sanktion, die resultierend aus der Meinungsumfrage an den Politiker gesendet wird.

**Demo Value Divider**

Der Demo Value Divider ist ein Faktor welcher die Funktion hat, die durchschnittliche Werteausprägung der Teilnehmer einer Demonstration abzuschwächen, damit dann die Werteausprägung jedes einzelnen Teilnehmer der Demonstration, in Abhängigkeit dieser abgeschwächten durchschnittlichen Werteausprägung, neu berechnet wird.

**Threshold For Demo Decission**

Überschreitet die Bewertung eines Agenten einer Rede diesen Grenzwert, so nimmt er an der Demonstration teil, die durch diese Rede ausgelöst wurde.

**Small Demo Size**

In Abhängigkeit der Demonstrationsgröße, sollen die Werteausprägungen der Agenten entweder erhöht oder verkleinert werden. Small Demo Size definiert die "kleine Demonstration" durch eine Integer Zahl.

**Middle Demo Size**

Analog zu Small Demo Size. Die Middle Demo Size bildet die nächsthöhere Stufe in der Demonstrationshierarchie.

**Big Demo Size**

Analog zu Middle Demo Size. Die Big Demo Size ist die zweithöchste Stufe in der Demonstrationshierarchie.

**Mass Demo Size**

Mass Demo Size bildet die höchste Demonstrationsstufe. Wegen  $1/x$  gilt:

$$SmallDemoSize > MiddleDemoSize > BigDemoSize > MassDemoSize$$

**Threshold Conflict Alarm Stage1**

Dieser Parameter bildet den Grenzwert für die erste Stufe des Konfliktalarms. Wird dieser überschritten, so wird die Einberufung des jeweiligen Botschafters simuliert.

**Sanction Strength For Ambassador**

Wird der Botschafter einberufen, so wird eine negative Sanktion, die durch Sanction Strength For Ambassador definiert ist, an den Politiker gesendet, der den Konfliktalarm ausgelöst hat.

**Threshold Conflict Alarm Stage2**

Dies ist der zweite Grenzwert des Konfliktalarms. Wird dieser überschritten, so kommt es zu einer Konflikteskalation und es können sich Milizen bilden.

**Max Milita Size**

Dieser Parameter definiert die maximale Größe einer Miliz.

**Militia Threshold**

Militia Threshold ist ein Grenzwert der definiert, ob der jeweilige Demonstrant so stark radikalisiert ist, dass er einer Miliz beitreten kann.

**Neighbor Expulsion**

Wird ein Agent durch eine Miliz vertreiben, so flüchtet dieser in ein anderes Gebiet. In diesem Gebiet muss er zuerst seine neuen Nachbarn identifizieren, zugleich aber "informiert" er seine neuen gleich-ethnischen Nachbarn über die Vertreibung wodurch die nationale Wertausprägung dieser Nachbarn um den Faktor Neighbor Expulsion steigt.

Der Zugriff auf die Parameter in *parameters.xml* wird in den implementierenden Klassen folgendermaßen realisiert:

```
Parameters p = RunEnvironment.getInstance().getParameters();
private int syncSp =(Integer) p.getValue("syncSpeeches");
```

Abbildung 6.15: Zugriff auf die Parameter in *parameters.xml*

In diesem Beispiel wird ein Objekt *p* vom Typ *Parameters* instanziiert und der Wert von "syncSpeeches" durch den Funktionsaufruf *p.getValue()* der Variablen *syncSp* zugewiesen. Ist der jeweilige Parameter in *parameters.xml* nicht als ReadOnly attribuiert, so kann der Wert in *parameters.xml* während der Simulation verändert werden.

### 6.3.2 Kontext

Die Kontexthierarchie sowie die dazugehörigen Projektionen, werden in *context.xml* definiert. Dieses beschreibt die Komponenten des Modells. Hierbei bildet “<context>” das Wurzelement, welches wiederum in den unteren Hierarchien als ein Sub-Kontext vorkommen kann. Einem Kontext kann eine “<projektion>” in Form einer Geographie oder eines Netzwerks assoziiert werden. Abbildung 6.16 zeigt einen Ausschnitt aus der *context.xml*:

```
<context id="EthnicConflicts" class="EthnicConflicts.ContextCreator">
  <context id="StateContext">
    <context id="BosniaContext" class="contextState.BosniaContext">
      <projektion type="geography" id="BosniaGeography"></projektion>
    </context>
  </context>
</context>
```

Abbildung 6.16: Aufbau von context.xml

Die Kontexthierarchie des Konfliktmodells ist folgendermaßen aufgebaut (Abbildung 6.17):

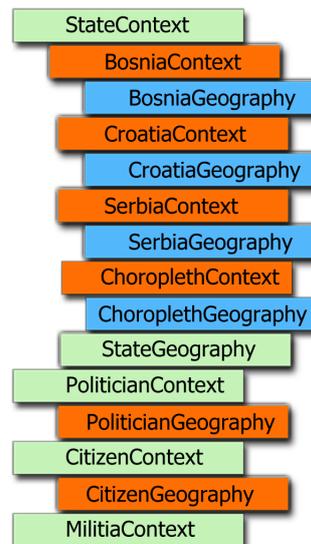


Abbildung 6.17: Kontexthierarchie des Konfliktmodells

Im Staat-Kontext befinden sich vier Sub-Kontexte, nämlich die drei Teilrepubliken von Ex-Jugoslawien und die Choroplethenkarte mit ihren geographischen Projektionen. Parallel zum Staat-Kontext existiert je ein Kontext für die Politiker, Bürger und die Miliz.

Alle Kontexte findet sich im Scenario Tree des GUI (Abbildung 6.18) wieder, in dem die Displays für den jeweiligen Kontext mit zugehöriger Projektion erstellt werden können.

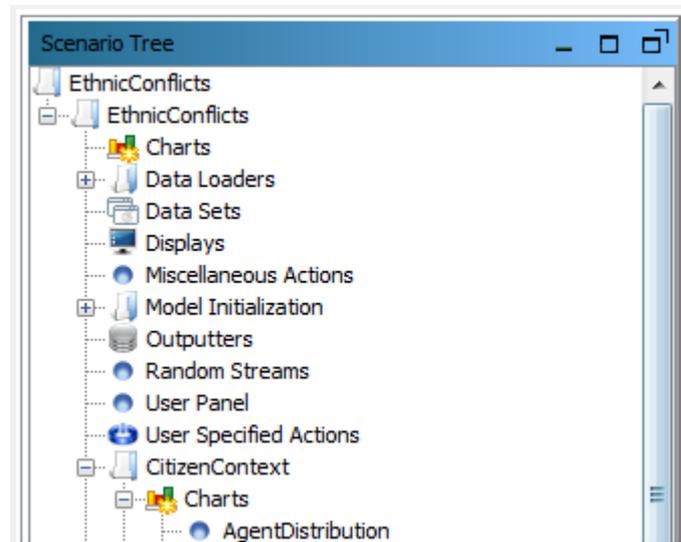


Abbildung 6.18: Scenario Tree

### 6.3.3 Kontexte der Umgebung

Um die Grenzen bzw. Verwaltungskreise darstellen zu können, müssen die von ArcGIS bereitgestellten geospatialen Daten in die jeweiligen Kontexte geladen werden. Zu jedem Kontext gibt es eine Shapefile dessen Einträge bzw. Zeilen aus der Attributentabelle als Objekte dargestellt werden. Diese Objekte werden in den Kontext geladen und in der zugehörigen geographischen Projektion abgebildet. Abbildung 6.19 zeigt ein Metamodell welches den Aufbau verdeutlicht:

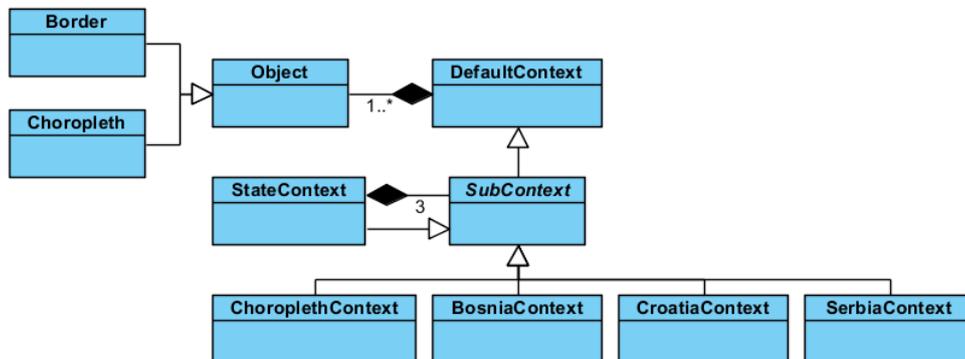


Abbildung 6.19: Metamodell - Kontextaufbau der Umgebung

*StateContext*, *BosniaContext*, *CroatiaContext*, *SerbiaContext* und *ChoroplethContext* sind Spezialisierungen von *DefaultContext*. *SubContext* bildet dabei eine abstrakte Klasse welche die Zugehörigkeit verdeutlichen soll. Dabei sind *BosniaContext*, *CroatiaContext* und *SerbiaContext* von *StateContext* existenzabhängig. Der *DefaultContext* selbst hat einen oder beliebig viele *Choropleth*- oder *Border*-Objekte.

Die Kontexte der drei Teilrepubliken und der Choroplethenkarte werden auf die gleiche Art implementiert. Der einzige Unterschied ist der Type Parameter (Border und Choropleth). In den jeweiligen Kontext-Klassen wird zuerst die Projektion erstellt:

```

GeographyParameters<Border> geoParams = new GeographyParameters<Border>();
Geography<Border> bGeography = GeographyFactoryFinder.createGeographyFactory
(null).createGeography("BosniaGeography", this, geoParams);

```

Abbildung 6.20: BosniaContext.java - Projektion

Im weiteren Schritt wird das bereitgestellte Shapefile in den Kontext geladen:

```

ShapefileLoader<Border> borderLoader = null;
try {
    borderFile = new File(GlobalVariables.BOSNIA_SHAPE);
    borderLoader = new ShapefileLoader<Border>(Border.class,
borderFile.toURL(), bGeography, this);
    while (borderLoader.hasNext()) {
        Border border = borderLoader.next();
    }
}

```

Abbildung 6.21: BosniaContext.java - ShapefileLoader

Schließlich werden die vier Kontexte als Subkontexte im *StateContext* verbunden.

```

public void createSubContexts() {
    this.addSubContext(new BosniaContext());
    this.addSubContext(new CroatiaContext());
    this.addSubContext(new SerbiaContext());
    this.addSubContext(new ChoroplethContext());
}

```

Abbildung 6.22: StateContext.java - Hinzufügen der Sub-Kontexte

Die komplette Struktur der Umgebungskontexte lässt sich somit in Abbildung 6.23 darstellen. Die Subkontexte bilden Teilmengen des *StateContext*.

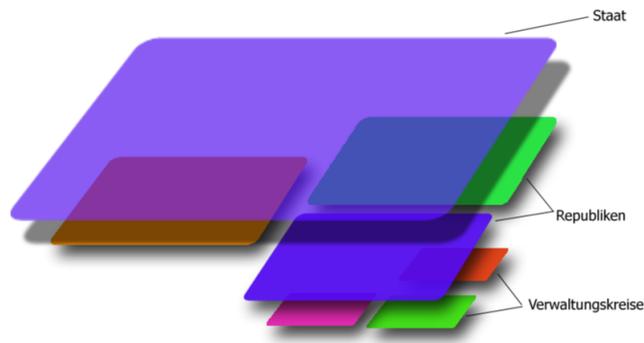


Abbildung 6.23: Kontextaufbau

### 6.3.4 Kontexte der Agenten

Die geographischen Projektionen der Agenten-Kontexte werden analog wie die der Umgebungs-Kontexte realisiert. Für die Agenten-Kontexte stehen jedoch keine Shapefiles zu Verfügung. Die Objekte der Kontexte werden im *PoliticianContext* und *CitizenContext* dynamisch erstellt. Abbildung 6.24 zeigt den Aufbau der Agenten-Kontexte:

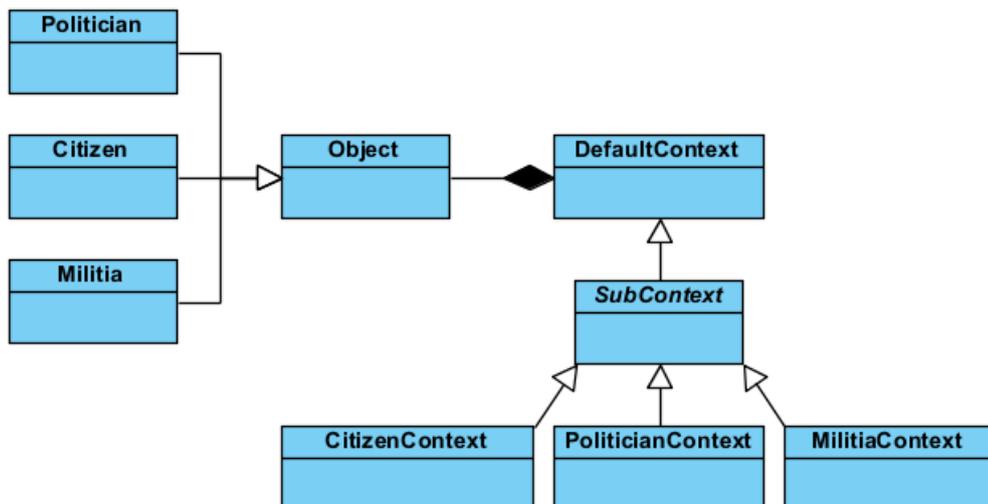


Abbildung 6.24: Klassendiagramm - Zugehörigkeit der Politiker

#### Kontext der Politiker

Die in Unterabschnitt 6.3.3 erstellten Objekte Borders und Choropleths, werden wie beschrieben, in einer geographischen Projektion abgebildet. Des Weiteren sind Agenten in einem geographischen Raum wie in 5.3.3 erläutert, durch eine Geometrie verbunden. Dies heißt konkret, dass jeder einzelne Agent in diesem Raum bestimmte Koordinaten besitzt. In Bezug auf die Anforderung aus 4.2.1, dass die Politiker eindeutig einer Republik zugeordnet werden müssen, wurde

jedem Politiker ein festes Koordinatenpaar zugeordnet. Dafür wurden Koordinaten ausgewählt, die sich innerhalb der Teilrepubliken befinden.

```
public static Coordinate cSarajevo = new Coordinate(18.3321,43.9147);
public static Coordinate cZagreb = new Coordinate(15.8948, 46.5794);
public static Coordinate cBelgrade = new Coordinate(20.4879, 45.1512);
```

Abbildung 6.25: PoliticianContext.java - Koordinaten

Die anschließende Generierung der Agenten wird wie in Abbildung 6.26 realisiert:

```
generatePoliticians(19, "Bosniak", polGeography, "Bosnia");
generatePoliticians(18, "Croat", polGeography, "Bosnia");
generatePoliticians(14, "Serb", polGeography, "Bosnia");
```

Abbildung 6.26: PoliticianContext.java - Generierung der Agenten

In der ersten Zeile der Abbildung 6.26 werden durch die Funktion *generatePoliticians()* 19 bosniakische Politiker instanziiert. In Bezug auf die Republik werden automatisch auch die Koordinaten gesetzt. Ebenso werden die Politiker dem *PoliticianContext* hinzugefügt.

Der *PoliticianContext* verwaltet auch zugleich die parallelen Reden pro Teilrepublik.

```
private static int [] speechSlots = {0,0,0}; //0-Bosnia //1-Croatia //2-Serbia
...
public int getSpeechSlotState(int x){
    return this.speechSlots[x];
}
public void setSpeechSlotState(int i, int t){
    this.speechSlots[i]+=t;
}
}
```

Abbildung 6.27: PoliticianContext.java - Synchronous Speeches

Das Array *speechSlots* dient als Speicher für die Anzahl der Reden, die zu einem Zeitpunkt gehalten werden. Der Index steht dabei für die jeweilige Teilrepublik. Die Anzahl der Reden, die pro Zeiteinheit parallel in den Teilrepubliken gehalten werden, wird durch den Parameter *Synchronous Speeches* aus *parameters.xml* beschränkt. Diese Überprüfung wird explizit im nächsten Unterabschnitt behandelt.

### Kontext der Bürger

Anders als bei den Politikern wird die Anzahl der zu erstellenden Bürgern aus dem Choropleth Shapefile abgeleitet. Im Unterabschnitt 6.3.3 wurde erläutert,

dass die Einträge bzw. die Zeilen der Attributentabelle der Shapefiles als Objekte behandelt werden. Eine Zeile im Shapefile kann aber auch weitere Spalten besitzen, die im Objekt selbst als Attribute behandelt werden können. Ausgehend vom Unterabschnitt 6.1.4 in dem die Erstellung und das Füllen der Attributentabelle mit einwohner-relevanten Informationen behandelt wurde, werden diese Informationen für eine realitätsnahe Aufteilung der Bürger-Agenten eingesetzt. Die *Choropleth* Klasse ist folgendermaßen aufgebaut:

```
public class Choropleth {
    private int ID;
    private int nBos;
    private int nSerb;
    private int nCroat;
    private int nYu;
    private int nCitizen;
}
```

Abbildung 6.28: Choropleth.java - Aufbau der Klasse

Hierbei werden die Parameter bei der Instanziierung im *ChoroplethContext* mit den Werten aus der Attributentabelle der jeweiligen Zeile belegt. Weil sich die Choropleths als Objekte im *ChoroplethContext* befinden bzw. in der *choropleth Geography* abgebildet sind, kann über sie iteriert werden:

```
for(Choropleth choroplethElem: cholt){
    for(int j=0;j<choroplethElem.getnBos();j++){
        Bosniak b=new Bosniak();
        b.setID(this.generateAgentID());
        this.add(b); //Dem Kontext hinzufügen
        int ci=Functions.uniformDistribution(0,10); //ziv. Identität
        int ni=Functions.uniformDistribution(0,10); //nat. Identität
        b.setNatIdentity(ni);
        b.setCivIdentity(ci);
    }
}
```

Abbildung 6.29: CitizenContext.java - Iteration der Choropleth Objekte

In Abbildung 6.29 ist der Instanzierungsprozess der bosniakischen Bürgern dargestellt. Für jedes Choropleth aus dem ChoroplethContext wird die Anzahl der Bosniaken betrachtet. In jeder inneren Iteration wird ein Bosniake erstellt. Diesem Bosniaken werden dann eine ID und die nationale und zivile Werteausprägung zugewiesen. Die Werteausprägung ist gleich verteilt.

Letztendlich müssen den Bürgern initiale Koordinaten in den jeweiligen Choropleths zugewiesen werden. Ein Choropleth bzw. Verwaltungskreis stellt ein unregelmäßiges Polygon dar. Somit kann nicht unbedingt durch Betrachtung der kleinsten und größten Punkte des Polygons auf der x- bzw. y-Achse ein Punkt z innerhalb dieses Polygons berechnet werden. Hierbei muss zusätzlich nach der Berechnung des Punktes z geprüft werden, ob dieser Punkt auch tatsächlich im Polygon liegt (Abbildung 6.30).

```
do{
  c.x = RandomHelper.nextDouble() * env.getWidth() + env.getMinX();
  c.y = RandomHelper.nextDouble() * env.getHeight() + env.getMinY();
  testGeom = new GeometryFactory().
  createPointFromInternalCoord(c, choroplethGeo);
  if(choroplethGeo.contains(testGeom)
      within=true);
}while(within==false);
```

Abbildung 6.30: CitizenContext.java - Berechnung der initialen Koordinaten

Mit Hilfe einer do-while Schleife werden per Zufall aus der Breite, Höhe, MinX und MinY des Polygons Koordinaten erstellt und anschließend geprüft, ob sie sich im jeweiligen Polygon befinden. Die Schleife wird erst dann abgebrochen, wenn die Koordinate auch tatsächlich im Polygon liegt. Dies ist dann auch der initiale Startpunkt eines Bürgers.

### 6.3.5 Granularität der Zeiteinheit

Die Ticks bzw. die Zeiteinheiten von Repast wurden für das Konflikt-Modell so normiert, dass sieben Ticks einer Woche entsprechen. Im Konflikt-Modell wird davon ausgegangen, dass eine bestimmte Zahl von Politikern jede Woche jeweils eine Rede halten können. Dies heißt, dass die Politiker alle sieben Ticks Aktionen ausführen. Die Bürger sind jedoch auch zwischen den sieben-Ticks Intervallen aktiv. Die Simulation sollte eine mehrjährige Zeitspanne abdecken, nämlich die Zeit vor dem Zerfall von ex-Jugoslawien und die folgende Kriegszeit.

### 6.3.6 Politiker

Die Klasse *Politician.java* implementiert den Politiker. Dabei wird die ethnische Zugehörigkeit eines Politikers anhand der Attribute *ethnicity* definiert. Die Zugehörigkeit zu einer Teilrepublik wird durch die Attribute *republic* festgelegt. Des Weiteren hat ein Politiker ein Attribut *popularity* welches seine Popularität festhält.

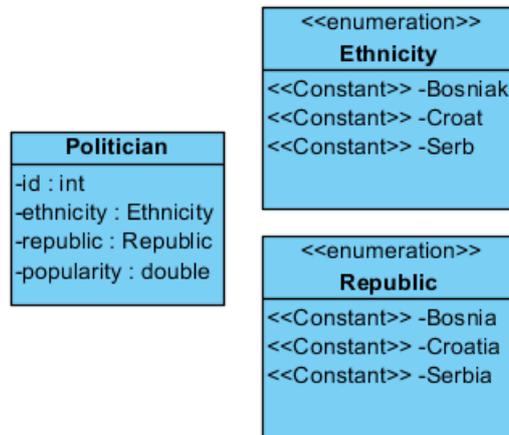


Abbildung 6.31: Klassendiagramm - Zugehörigkeit der Politiker

Neben den in Abbildung 6.31 aufgelisteten Attributen, besitzt der Politiker weitere Attribute, die wegen der Übersichtlichkeit getrennt behandelt werden:

```

1. private Queue<EmilMessage> emilActionList = new LinkedList<EmilMessage>();
2. private IEMILAgent emilLayer;
3. private Queue<Speech> mySpeeches = new LinkedList<Speech>();
4. private Speech actualSpeech;
5. public enum ACTIONS {A1, A2, A3, A4}
6. private double opinionResearchStrengthDivider;
7. private double credibilityDivider;
8. private int syncSpeeches;
  
```

Abbildung 6.32: Politician.java - Weitere Attribute des Politikers

### Kommunikation mit der EmIL Ebene

Für die Realisierung des normativen Verhaltens eines Politikers, wurde eine ähnliche Implementationsweise wie die des Micro-Finance Modells gewählt [25]. Die Zeilen 1-2 in Abbildung 6.32 werden explizit für die Umsetzung des normativen Verhaltens benötigt. Die Queue `emilActionList` ist eine Datenstruktur, welche die vom EmIL-Agenten erhaltenen Actions zwischenspeichert. Diese Actions werden dann vom Repast-Agenten ausgeführt. Bei den EmIL- und Repast-Agenten handelt es sich um ein und denselben Agenten jedoch auf zwei unterschiedlichen Ebenen. Die Identifizierung erfolgt anhand der *ID*. Diese *ID* wird während der Instanziierung der Agenten im *PoliticianContext.java* vergeben.

```

public void setID(int id){
    this.ID=id; //Repast Ebene
    //EmIL Ebene
    emilLayer = emil.Controller.getInstance().addAgent("Politician",
    this.ID, this);
}

```

Abbildung 6.33: Politician.java - ID Vergabe

Abbildung 6.34 illustriert die Kommunikationsweise des EmIL- und Repast-Agenten. Wird der Repast-Agent aktiv, so ruft er ein bestimmtes Ereignis auf und sendet dieses an den EmIL-Agenten (Abbildung 6.35).

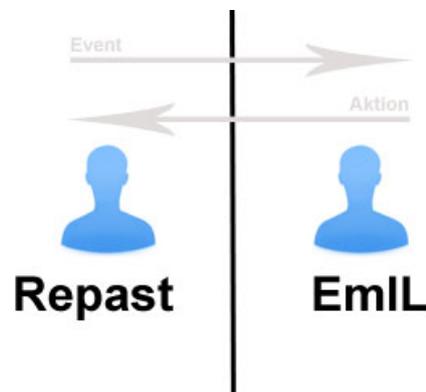


Abbildung 6.34: Kommunikationsweise von EmIL- und Repast-Agenten

```

emilLayer.processMessage{
    new EMILMessage(ID, ID, Modal.ASSERTION,
    new emil.message.ENVContent(RepastEssentials.GetTickCount(),
    "E1", ID, 0));
}

```

Abbildung 6.35: Politician.java - Senden eines Events E1 an den EmIL-Agenten

Dieses Ereignis wird dann vom EmIL-Agenten verarbeitet und die auszuführende Aktion anschließend an den Repast-Agenten geliefert. Hierfür wird ein neues Objekt der Klasse EmilMessage erstellt. Diese enthält den TimeStamp, die OriginalNachricht von EmIL und die Aktion.

```

EmilMessage a = new EmilMessage((int) arg0.getContent().getTimestamp(), arg0,
    (String) arg0.getContent().getContent());
this.emilActionList.add(a); //Einfügen in die emilActionList
this.actualMessage = a; // Aktuelle Nachricht auf a setzen

```

Abbildung 6.36: Politician.java - Erstellung einer EmilMessage

### Der Redeprozess

Der Redeprozess wird durch eine Scheduler-Methode ausgelöst. Eine solche Methode besitzt eine Annotation, welche dem System zusätzliche Informationen gibt. In diesem Fall gibt die Scheduler-Annotation über einer bestimmten Methode dem System Informationen, wann diese ausgeführt werden soll [31].

```
@ScheduledMethod(start = 1, pick = 50, interval = 5, shuffle=true)
public void step() {
    PoliticianContext p = ContextCreator.getPoliticianContext();
    if(p.getSpeechSlotState(this.getSpeechSlotId())<syncSpeeches){
        speechProcess();
    }
}
```

Abbildung 6.37: Politician.java - Scheduler-Methode

Die Abbildung 6.37 zeigt eine Scheduler-Annotation über der Methode `step()`. Ihre Argumente definieren, dass sie angefangen vom ersten Tick, in einem Intervall von je fünf Ticks ausgeführt wird. Die beiden Argumente `pick` und `shuffle` definieren hierbei, dass in jedem Intervall 50 Politiker-Agenten rein zufällig ausgewählt werden und deren `step()`-Methode aufgerufen wird.

Wird eine `step()`-Methode aktiv, dann wird zuerst geprüft, ob es noch freie “SpeechSlots” gibt. Der Methode `getSpeechSlotState(int slotID)` welche im Unterabschnitt 6.2.4 erläutert wurde, muss eine `slotID` übergeben werden. Diese `slotID` ist der Index des Arrays `SpeechSlots` aus `PoliticianContext.java` und steht wie erwähnt, für die jeweilige Teilrepublik im Array. In Abhängigkeit der Zugehörigkeit eines Politikers zu einer Teilrepublik, wird der Wert aus `republic`, welcher als String vorliegt, durch eine Integer Zahl von 0 bis 2 dargestellt.

Somit kann konkret geprüft werden, ob z. B. zu einem Zeitpunkt in Kroatien, die maximale Anzahl der Reden gehalten wird. Wenn dies nicht der Fall ist, so können weitere kroatische Politiker einen Platz in den “SpeechSlots” einnehmen und sich an die Bürger mit eine Rede wenden.

Ist die Bedingung erfüllt, so wird die Funktion `speechProcess()` aufgerufen.

```
if(actualSpeech==null){
    this.eventSwitcher(1);
    actionListExecuter();
}
...
if(actualSpeech!=null && actualSpeech.getOpinionState()==true){
    this.actualSpeech=null;
}
```

Abbildung 6.38: Politician.java - speechProcess()

Falls eine aktuelle Rede existiert, wird sie erst dann dereferenziert, wenn die erste Evaluierungsstufe bzw. die Meinungsumfrage stattgefunden hat. Existiert keine

aktuelle Rede, so wird die Funktion `eventSwitcher()` aufgerufen. Diese sendet ein Ereignis an den EmIL-Agenten (Abbildung 6.35). In diesem Fall handelt es sich um das Ereignis “Politiker will eine Rede halten”. Als Antwort auf das Ereignis sendet der EmIL-Agent die zu ausführende Aktion zurück. Zu gleich ruft der Repast-Agent die Funktion `actionListExecuter()` auf, die prüft, ob sich neue Aktionen in der `emilActionList` befinden die dann iterativ abgearbeitet werden.

```
Speech s=new Speech(this,"A1");
mySpeeches.add(s);
this.actualSpeech=s;
s.setCredibility(this.credibilityCalculation("A1"));
deploySpeechToAgents(s);
```

Abbildung 6.39: Politician.java - Ausführen einer Aktion

Abbildung 6.39 zeigt ein Beispiel, in dem eine “A1” Aktion verarbeitet wird. Diese steht wie im Unterabschnitt 6.2 erläutert, für das Halten einer nationalen Rede. Hierfür wird eine neues Redeobjekt vom Typ `Speech` erstellt, die Glaubwürdigkeit berechnet und dann an alle Bürger verteilt.

$$Credibility = \frac{\frac{\sum TypeX}{\sum allSpeeches}}{credibilityDivider}$$

Die Glaubwürdigkeit berechnet sich aus der Summe eines bestimmten Rede-Typs durch die Summe aller gehaltenen Reden. Dieser Faktor wird zusätzlich durch den `credibilityDivider` abgeschwächt.

### Die Rede

Die Rede wird als ein Rede-Objekt dargestellt. Dieses beinhaltet alle relevanten Information über die Rede selbst und über ihre Bewertung.

```
public Speech(Politician p, String i){
    this.p = p;
    this.typeOfSpeech = i;
    this.allRatings = new double[this.numCitizen];
    this.ciRated = 0;
    ...
    this.pro = new Demonstration();
    this.anti = new Demonstration();
}
```

Abbildung 6.40: Speech.java - Die Struktur der Speech Klasse

Das Array `allRatings` dient hierbei als Speicher für die Bewertungen der Bürger. Dabei wird die Array Länge auf die Anzahl aller Bürger initialisiert. Weil jeder Bürger eine aufsteigende ID hat, kann diese ID als der Index dieses Array betrachtet werden. Unter diesem Index, kann dann die Bewertung des Bürgers ins Array abgelegt werden. Unter anderem kann eine Rede auch eine Pro-

und Anti Demonstration auslösen. Diese werden bei der Instanziierung eines Rede-Objekts auch mit instanziiert, um so Informationen über die eventuell auftretende Demonstrationen aufzunehmen.

### 6.3.7 Bürger

Anders als beim Politiker wird die ethnische Zugehörigkeit im Fall der Bürger durch Instanzen der jeweiligen Klassen unterschieden. Abbildung 6.41 zeigt die Klassenhierarchie des Bürgers.

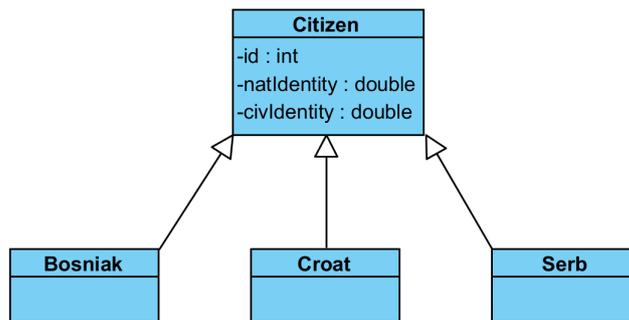


Abbildung 6.41: Klassendiagramm - Bürger

Die Identitätsausprägung des Bürgers wird durch die beiden Attribute *natIdentity* und *civIdentity* definiert ist. Diese beschreiben einen Punkt in einem zweidimensionalen Koordinatensystem. Beide Attribute können einen Wert zwischen 0.0 bis 10.00 annehmen, wobei 0.0 als schwach und 10.00 als stark interpretiert wird. (Abbildung 6.42)

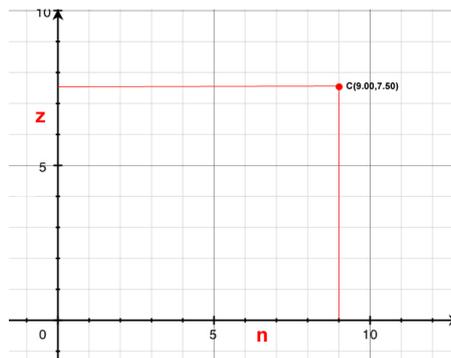


Abbildung 6.42: Identitätsausprägung der Bürger

Zum Zeitpunkt des Instanzierung, wie im Unterabschnitt 6.3.4 beschrieben, wird die nationale und zivile Identitätsausprägung eines Bürgers gesetzt. Unter anderem bekommen die Bürger auch einen initialen Koordinatenpunkt. Dieser Punkt ist sein Startpunkt in der Simulation.

## Netzwerke

Vor dem Beginn der Simulation, müssen zuerst die Netzwerke der Bürger aufgebaut werden. Dies passiert das erste Mal unmittelbar nachdem alle Bürger-Agenten erstellt worden sind (Unterabschnitt 6.3.4). Das erste Netzwerk das aufgebaut werden muss, ist das Nachbarschaftsnetzwerk. Als Abbruchbedingung dient hierbei die Größe der Moore-Nachbarschaft und die fünfmalige Wiederholung dieses Prozesses. Die Nachbarn werden dabei als Objekte in einer Array Liste verwaltet.

```
while ( myNeighbors.size() < Functions.mooreRangeToNeighbors(mooreRange) &&
        resizeRadius < 5) {
    getNeighborsWithinTheAgentRadius(aRadius);
    aRadius = aRadius * 1.2;
    resizeRadius++;
}
```

Abbildung 6.43: Citizen.java - Nachbarsuche

Da alle Agenten über Koordinatenpunkte besitzen, kann der Radius benutzt werden, um festzustellen, welche Agenten sich innerhalb des Radius eines bestimmten Koordinatenpunktes befinden. Befinden sich nicht genügend Nachbarn im Radius, so wird dieser mit einem Multiplikator vergrößert. Abbildung 6.44 verdeutlicht den Suchprozess.

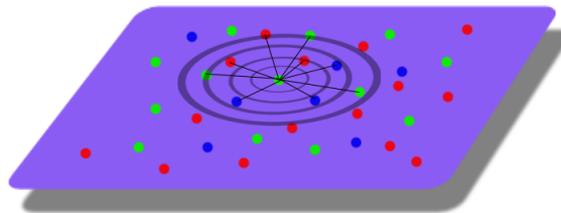


Abbildung 6.44: Suchprozess der Nachbarsuche

Das zweite Netzwerk ist das Bekanntschaftsnetzwerk. Hierbei werden zuerst alle Agenten aus dem *CitizenContext* nach der Ethnizität sortiert. Dann wählen die jeweiligen Agenten in der Initialisierungsphase rein zufällig zwischen 5 und 20 Bekannte aus. Diese können über den gesamten Staat verstreut sein. Ist das Bekanntschaftsnetzwerk aufgebaut worden, so muss im Falle einer Vertreibung eine Fluchtcoordinate vorab bekannt sein.

```

for(Citizen c:myConnections){
    temp = c.getNeighborList();
    int friend = 0;
    int enemy = 0;
    for(Citizen cc:temp){
        if(this.getClass().equals(cc.getClass()))
            friend++;
        else
            enemy++;
    }
    if(bestDifference < friend - enemy)
        bestDifference = friend - enemy;
        bestLocation = c.getGeometry();
}

```

Abbildung 6.45: Citizen.java - Bestimmung der Fluchtkoordinaten

Abbildung 6.45 zeigt den Prozess zur Bestimmung der Fluchtkoordinaten. Hierbei wird die Nachbarschaft jedes einzelnen Bürgers aus dem Bekanntschaftsnetzwerk betrachtet und bestimmt, wie viele gleich- und fremd-ethnische Nachbarn dieser hat. Die Umgebung des "Bekanntnen" mit den meisten gleich-ethnischen Bürgern wird zum Fluchort gewählt (Abbildung 6.46).

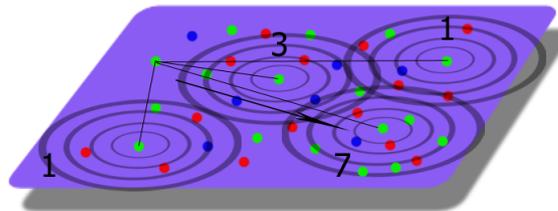


Abbildung 6.46: Suchprozess der Fluchtkoordinaten

### Initialisierungsphase

Ebenso wie der Politiker, hat der Bürger auch eine Scheduled-Method, dessen Intervall jedoch feiner ist. Wird *step()* das erste Mal ausgeführt, so wird die temporäre Geometrie initialisiert. Aus der Geometrie können zu jeder Zeit die Koordinaten des Agenten extrahiert werden. Der Agent besitzt eine temporäre und eine feste Geometrie. Nimmt er z.B. an einer Demonstration teil, so beinhaltet die temporäre Geometrie die aktuellen Koordinaten und die feste Geometrie seine Ausgangskoordinaten.

```

@ScheduledMethod(start = 1, pick = 2000, interval = 1)
public void step() {
    if (initializeLocation==false){
        this.citizenGeography = ContextCreator.getCitizenGeography();
        this.myGeometry = citizenGeography.getGeometry(this);
        this.tempGeometry = this.myGeometry;
        initializeLocation=true;
    }
    if (this.memberOfMilitia==false){
        rateSpeeches();
    }
}

```

Abbildung 6.47: Citizen.java - step() - Methode

### Bewertung

Ist der jeweilige Bürger nicht in einer Miliz, so kann der Bewertungsprozess der Reden beginnen. Der Bürger verwaltet zwei Queues, welche mit Rede-Objekten gefüllt werden können:

```

private Queue<Speech> rateSpeeches = new LinkedList<Speech>();
private Queue<Speech> discSpeeches = new LinkedList<Speech>();

```

Abbildung 6.48: Citizen.java - Rate und Discuss Queues

Erstellt der Politiker eine Rede (Unterabschnitt 6.3.7), so wird diese an die Bürger verteilt. Hierbei wird der CitizenContext iteriert, so dass jedem Bürger das erstellte Rede-Objekt in die Queue *rateSpeeches* eingefügt wird. Bei jedem Tick prüft der Bürger-Agent, ob sich neue Rede-Objekte in *rateSpeeches* befinden. Ist dies der Fall, so gibt er eine eigene Bewertung ab. Dies passiert in Abhängigkeit der nationalen und zivilen Identitätsausprägung (Abbildung 6.49).

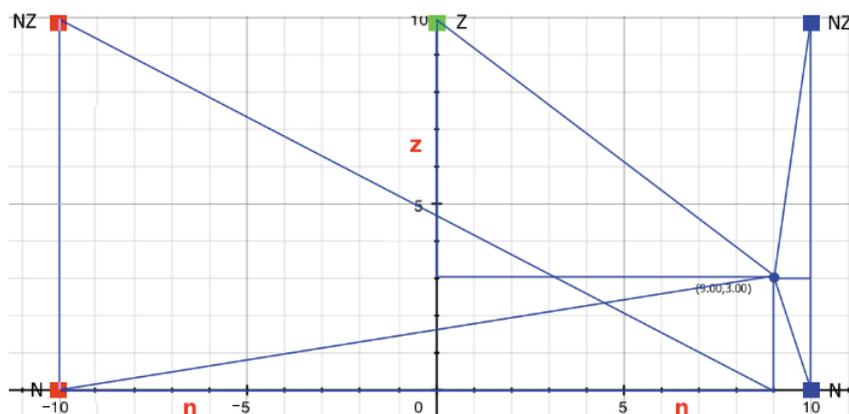


Abbildung 6.49: Bewertung der Reden

Die jeweiligen Reden bilden selbst feste Punkte im Koordinatensystem und sind

folgendermaßen aufgeteilt:

Politiker	Redetyp		
	national	national-zivil	zivil
eigene Ethnizität	(10.00, 0.00)	(10.00, 10.00)	(0.00, 10.00)
fremde Ethnizität	(-10.00, 0.00)	(-10.00, -10.00)	(0.00, 10.00)

Tabelle 5: Koordinatenpunkte der Redetypen

Die Bewertung ist somit die Distanz zwischen dem Punkt der Identitätsausprägung des Bürgers und dem Punkt der jeweiligen Rede. Hierbei bildet 0.00 die beste und 22.36 die schlechteste Bewertung. In die Bewertung fließt auch die Gewichtung der Glaubwürdigkeit. Bei maximaler Glaubwürdigkeit bleibt die Bewertung erhalten und bei maximaler Unglaubwürdigkeit verdoppelt sie sich, wobei der Wert über 22.36 abgeschnitten werden muss. Die Bewertung wird ins Bewertungs-Array (Unterabschnitt 6.3.6) unter dem Index der Bürger ID abgelegt.

### Diskussion

Ist die Bewertung abgegeben worden, so wird das Rede-Objekt aus der Queue *rateSpeeches* in die Queue *discSpeeches* verschoben. Hierbei beobachtet die Watcher Klasse welche im Unterabschnitt 6.3.8 beschrieben wird, ob alle Bürger ihre Bewertung zu einer bestimmten Rede abgegeben haben. Ist dies der Fall, so triggert die Watcher Klasse die Diskussion bzgl. dieser Rede.

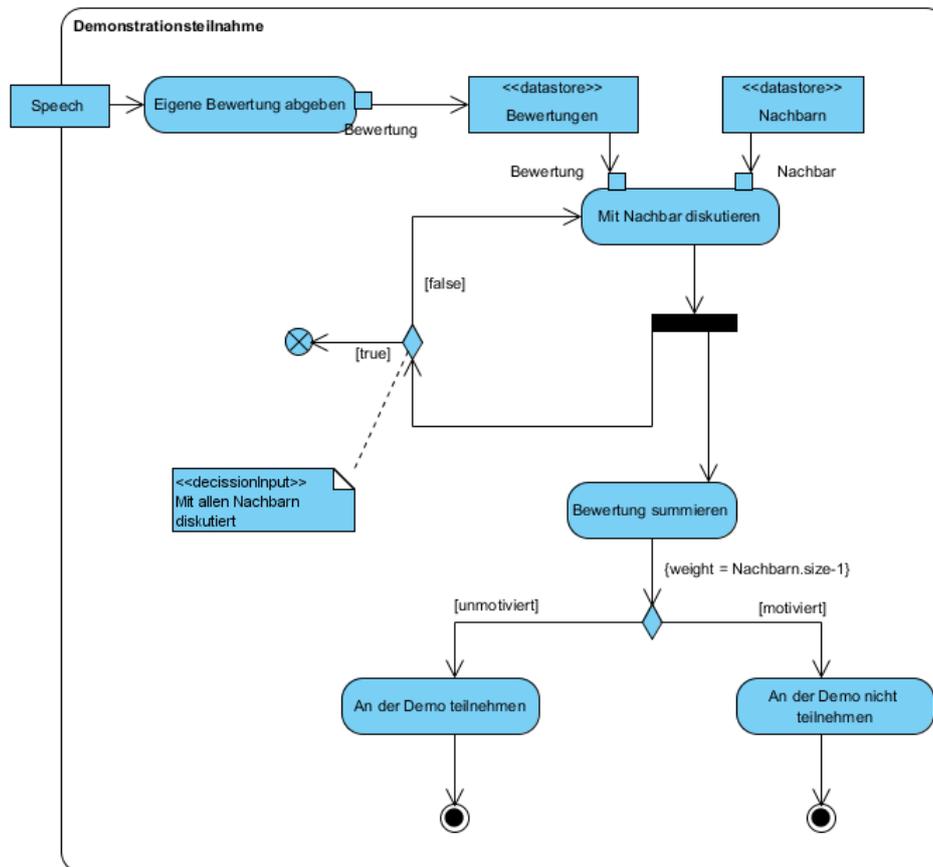


Abbildung 6.50: Aktivitätsdiagramm - Demonstrationsteilnahme

Für jeden Bürger wird die Nachbarschaftsliste iteriert und die Bewertungen aufsummiert (Abbildung 6.50). Bei nationalen Reden werden nur die Bewertungen der gleich-ethnischen Nachbarn betrachtet. Aus der Summe und Anzahl der Nachbarn wird der Durchschnittswert berechnet. Dieser wird zusammen mit der eigenen Bewertung für die Berechnung der Teilnahme an einer Pro- oder Anti-Demonstration benutzt:

$$d = avgDistance \cdot avgOpinionMultiplikator$$

$$newRatingDistance_{temp} = (ownRatingDistance + avgDistance)$$

$$newRatingDistance = newRatingDistance_{temp} / (avgOpinionMultiplikator + 1)$$

Ist das Ergebnis kleiner als der Parameter *thresholdDemoDecission*, so nimmt der Bürger-Agent an einer Pro-Demonstration teil. Analog findet die Bewertung für eine Anti-Demonstration teil, jedoch mit:

$$newRatingDistance > 22.36 - thresholdDemoDecission$$

## Mobilität

Nimmt ein Bürger an einer Demonstration teil, so muss sich dieser zu den Koordinaten des Versammlungsorts bewegen. Um zu vermeiden, dass sich alle Agenten auf einem einzigen Punkt zubewegen und sich so in einem Punkt überlagern, wird eine ähnliche Methode wie bei der Erstellung der initialen Koordinaten der Bürger eingesetzt (Unterabschnitt 6.3.4).

```
Geometry movePoint = Functions.getGeometryWithinSpecificArea(target);
while(!atDestination(movePoint))
    agentStep(movePoint);
```

Abbildung 6.51: Citizen.java - Beweglichkeit des Agenten

Abbildung 6.51 zeigt die Methode, welche die Beweglichkeit der Bürger realisiert. Der Übergabeparameter *target* ist hierbei z.B. die Geometrie welche die Koordinaten des Versammlungsorts einer Demonstration beinhaltet. Aus den Koordinaten werden Umgebungskoordinaten berechnet und die *agentStep(movePoint)* Methode solange aufgerufen, bis sich der Agent am Zielort befindet. *agentStep()* dient zu Visualisierung der schrittweisen Bewegung eines Bürgers.

```
Coordinate target = targetG.getCoordinate();
double angle = Functions.angleOfVector(target, this.tempCoordinates)+Math.PI;
double distToTarget = DistanceOp.distance(this.tempGeometry, targetG);
double movePerTurn = distToTarget/5;
double movePerTurnInMeters = Functions.convertToMeters(movePerTurn);
citizenGeography.moveByVector(this, movePerTurnInMeters, angle);
```

Abbildung 6.52: Citizen.java - agentStep() Methode

Diese wird realisiert, in dem zuerst der Winkel des Vektors und die Distanz zwischen den aktuellen Koordinaten und den Zielkoordinaten berechnet werden. Danach wird die Distanz durch fünf geteilt, was konkret fünf Schritte darstellen soll. Der Bürger überquert dann in einem Schritt und in einem bestimmten Winkel  $1/5$  der Distanz zum Ziel.

### 6.3.8 Watcher

Der *Watcher* ist eine Klasse, welche eine Reihe von Methoden implementiert, die in Abhängigkeit vom Zustands des Modells ausgeführt werden sollen. Ähnlich wie die Scheduled-Methods haben die Methoden der *Watcher* Klasse auch Annotationen, die jedoch auf die dynamischen Veränderungen des Systems reagieren. Eine Watcher Annotation ist folgendermaßen aufgebaut:

```

@Watch(watcheeClassName = "Speech", watcheeFieldNames = "ratingState",
whenToTrigger = WatcherTriggerSchedule.LATER)
    public void triggerDiscussSpeech(Speech s){
        ..
    }

```

Abbildung 6.53: Watcher.java - Annotation

Das erste Attribut definiert die zu beobachtenden Objekte einer Klasse. *watcheeFieldNames* definiert das zu beobachtende Attribut innerhalb derselben Klasse. Im letzten Attribut der Notation ist festgelegt, wann die Funktion ausgeführt werden soll. Ändert sich somit innerhalb eines Objekts der Klasse *Speech* der Wert des Attributs *ratingState*, welches definiert ob die Einzelbewertung einer Rede abgeschlossen ist, so wird der oben beschriebene Diskussionsprozess angestoßen.

### 6.3.9 Demonstration

Im Unterabschnitt 6.3.6 wurde erläutert, dass wenn ein Rede-Objekt erstellt wird, gleichzeitig auch zwei Demonstrations-Objekte erstellt werden. Die Klasse *Demonstration* ist dabei ähnlich wie die Klasse *Speech* aufgebaut. Für die Verwaltung der "Demonstranten" wird hierbei auch eine Array Liste benutzt. Um eine Demonstration auszulösen, beobachtet die Watcher Klasse, ob der Diskussionsprozess der Bürger beendet worden ist. Der Trigger ist mit einer Toleranz von ca. 1% angesetzt, d.h. dass er die Demonstration auslöst auch wenn 1% der Bürger evtl. die Diskussion nicht abgeschlossen hat. Der Versammlungsort der Demonstration ist dabei der Ort, an dem sich der Bürger mit der besten bzw. schlechtesten Bewertung zu dieser Rede befindet.

In der Demonstration finden zwei Prozesse statt. Im ersten Prozess wird die Identitätsausprägung aller Agenten in Abhängigkeit der Demonstrationsgröße und der durchschnittlichen Identitätsausprägung neu berechnet. Im zweiten Prozess wird geprüft, ob die Demonstration die Voraussetzungen für das Auslösen eines Konfliktalarms erfüllt hat. Im Hintergrund liefern diese Zahlen auch ein Feedback an den Politiker über seinen Erfolg oder Misserfolg in Form einer Sanktion zurück.

### Berechnung der Identitätsausprägung

Die Identitätsausprägung kann entweder schrumpfen oder wachsen. Dies ist Abhängig von dem Redetyp.

Identitätsausprägung	Rede-Typ			
	A1	A2	A3	A3
national	+	-	+	
zivil	-	+	+	+

Tabelle 6: Identitätsausprägung nach einer Demonstration

Wie stark die Identitätsausprägung wächst, hängt von der Demonstrationsgröße ab:

smallDemoSize	middleDemoSize	bigDemoSize	massDemoSize
$demoSize < 50$	$50 \leq demoSize < 100$	$100 \leq demoSize < 500$	$demoSize \geq 500$

Tabelle 7: Demonstrationsgrößen

Folglich wird auf Grund der Demonstrationsgröße der dazugehörige Parameter als Gewichtungsfaktor gewählt. Für die positive Identitätsausprägung gilt somit:

$$newValue = \left( \frac{1}{demoSize} \cdot avgValue \right) / demoValueDivider + myValue$$

Und für die negative Identitätsausprägung gilt:

$$newValue = myValue - \left( \frac{1}{demoSize} \cdot avgValue \right) / demoValueDivider$$

### Konfliktalarm

Damit der Konfliktalarm ausgelöst werden kann, muss zum einen die Demonstration durch eine national motivierte Rede ausgelöst worden sein und zum anderen muss es eine große Unterstützung von außerhalb der jeweiligen Republik geben. Hierfür gilt die folgende Bedingung:

$$support = (100.00 / countOwnEthicityInOtherArea) * supportFromOtherArea$$

$$(support > thresholdConflictAlarmStage1)$$

Hält z. B. ein serbischer Politiker in Bosnien und Herzegowina eine nationale Rede, so werden alle serbischen Bürger außerhalb (Serbien und Kroatien) gezählt. Danach wird bestimmt, wie viele Bürger sich aus diesen Teilrepubliken auf der Demonstration befinden. Übersteigt dieser Prozentsatz die erste Konfliktalarmstufe, so wird die Einberufung des Botschafters durch das Senden einer Sanktion an den jeweiligen Politiker simuliert. Dadurch wird versucht, die Konflikteskalation zu verhindern, indem der jeweilige Politiker zum Wechsel des Rede-Typs forciert wird. Übersteigt dieser Prozentsatz die zweite Konfliktstufe, so ist die Voraussetzung für die Miliz-Bildung erfüllt.

#### 6.3.10 Miliz

Ist der zweite Konfliktalarm ausgelöst worden, so können sich Milizen bilden. Hierzu wird die Liste der Demonstranten betrachtet, da diese am stärksten radikalisiert und sehr gewaltbereit sind.

```
ArrayList<Citizen> militiaList = new ArrayList<Citizen>();
ArrayList<Citizen> militiaPotentialLeaders = null;
```

Abbildung 6.54: Watcher.java - MilitiaBuilder Listen

In der Methode *militiaBuilder()* der *WatcherKlasse* werden zwei Array Listen verwaltet (Abbildung 6.54). Die *militiaList* beinhaltet die potentiellen Mitglieder, als auch die *militiaPotentialLeaders*, welche für die potentiellen Anführer stehen. Diese Miliz-Anführer müssen Bürger der Teilrepublik sein, in der der Konfliktalarm ausgelöst worden ist.

```
if (s.getRatingById(c.getID()) < militiaThreshold){
    militiaList.add(c);
}
```

Abbildung 6.55: Watcher.java - MilitiaBuilder

Durch Iteration der Demonstrationsliste wird geprüft, ob die Bewertung eines Bürgers kleiner (d.h. besser) ist als der Grenzwert *militiaThreshold*. Ist dies der Fall, so ist dieser Bürger ein potentielles Mitglied einer Miliz. Für die potentiellen Mitglieder wird die Anzahl der möglichen Milizen berechnet:

$$m = militiaList.size() - (militiaList.size() \% maxMilitiaSize)$$

$$numMilitia = m / maxMilitaSize$$

```
for (int i=0; i < numMilitia; i++){
    Militia m = new Militia();
    Citizen l = militiaStartLocation(militiaPotentialLeaders);
    m.setGeoemtry(l.getGeometry());
    ..
    while(m.getSlotStatus() == true && militiaList.size() > 0){
        Citizen militiaMember = closestMilitiaMember(militiaList, l);
        m.addToMilita(militiaMember);
        ..
    }
    m.setActionTrigger(true);
}
```

Abbildung 6.56: Watcher.java - Erstellung von Miliz-Objekten

Für die Anzahl der möglichen Milizen werden Miliz-Objekte der Klasse *Militia* erstellt und dem Miliz-Kontext hinzugefügt. Jedes Miliz-Objekt hat einen Anführer und eine maximal mögliche Anzahl von Mitgliedern. Die Startkoordinaten der Miliz sind die aktuellen Koordinaten des Anführers. Die Priorität bei der Füllung der Miliz ist nach der Nähe zum Anführer geordnet, d.h. potentielle Miliz-Mitglieder aus der Umgebung eines Anführers schließen sich der Miliz direkt an.

Ist eine Miliz vollständig, so wird ein weiterer Trigger, der *actionTrigger* in der *Militia* Klasse ausgelöst. Dieser Trigger wird von der *Watcher* Klasse registriert und die möglichen Aktionen einer Miliz ausgelöst. Hierbei werden aus der Nachbarschaftsliste des Miliz-Anführers die fremd-ethnischen Nachbarn extrahiert, die mögliche Angriffsziele darstellen.

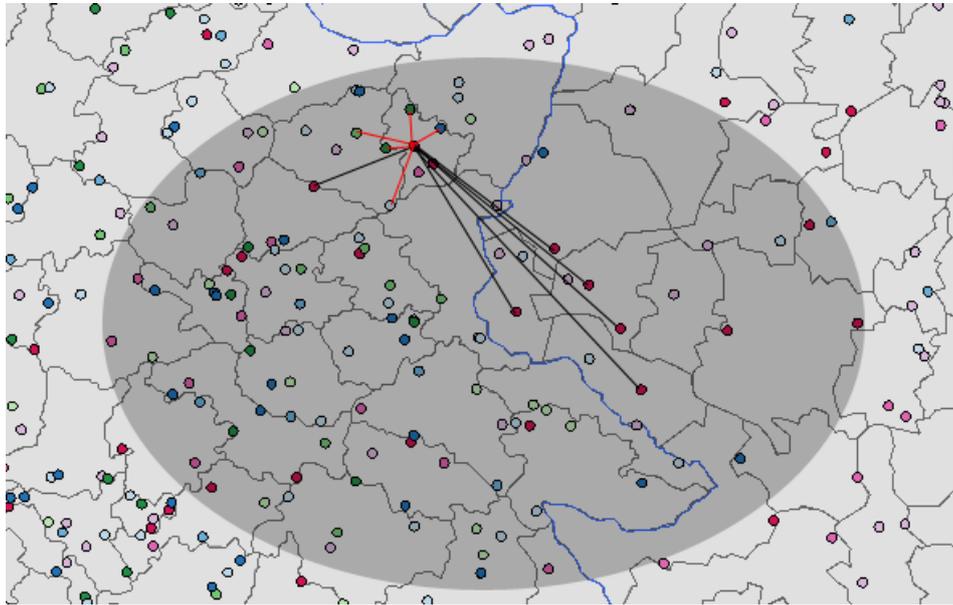


Abbildung 6.57: Angriffsziele der Miliz

Abbildung 6.57 illustriert eine mögliche Bildung einer serbischen Miliz (schwarze Kanten) und die möglichen Angriffsziele (rote Kanten). Ist ein Angriffsziel festgelegt worden, so hat die Miliz die Möglichkeit den fremd-ethnischen Bürger am Angriffsziel entweder zu töten oder ihn zu vertreiben. Die Wahrscheinlichkeit ist hierfür gleich verteilt. Nach jedem Angriff löst sich die Miliz teilweise auf, indem jeweils ein Mitglied der Miliz auf dem ethnisch sauberen Gebiet verbleibt. Im Falle einer Vertreibung flüchtet der fremd-ethnische Bürger zu seinen Fluchtkoordinaten aus dem Bekanntschaftsnetzwerk (Abbildung 6.46). Dieser wird maximal radikalisiert und informiert seine neuen gleich-ethnischen Nachbarn über die Vertreibung.

```

while (itr.hasNext()) {
    Citizen c = itr.next();
    double civIdentity = neighborExpulsion * c.getCivIdentity();
    double natIdentity = neighborExpulsion * c.getNatIdentity();

    if (c.getCivIdentity() - civIdentity > 0) {
        c.setCivIdentity(c.getCivIdentity() - civIdentity);
    } else {
        c.setCivIdentity(0);
    }
    ...
}

```

Abbildung 6.58: Citizen.java - Vertreibung und Radikalisierung

Dabei wird die zivile Identitätsausprägung der Nachbarn reduziert und die nationale Identitätsausprägung erhöht. Dies hat als Konsequenz, dass die vertriebene Ethnie schneller radikalisiert wird und somit selbst Milizen bilden kann.

## 7 Simulationslauf und Ergebnisanalyse

In diesem Kapitel wird das aus dem Kapitel 6 implementierte Modell verschiedenen Simulationsläufen unterzogen und die Ergebnisse analysiert und interpretiert.

### 7.1 Zusammenhang zwischen Parametern

Eine Beobachtung, die während verschiedener Simulationsläufe festgestellt wurde, ist der Zusammenhang zwischen den beiden Parametern *demoValueDivider* und *avgOpinionMultiplikator*. Hierfür wurden mehrere kurze Simulationen mit ca. 100 Bürger-Agenten durchgeführt und jeweils verschiedene Werte für die beiden Parameter gewählt.



Abbildung 7.1: Simulation - Serbische Identitätsausprägungen 1. Lauf

In Abbildung 7.1 ist der zeitliche Verlauf der durchschnittlichen nationalen und zivilen Identitätsausprägung der serbischen Bürger dargestellt. Zu dieser Simulation wurden folgende Parameter-Einstellungen genutzt:

- *demoValueDivider* = 1
- *avgOpinionMultiplikator* = 1

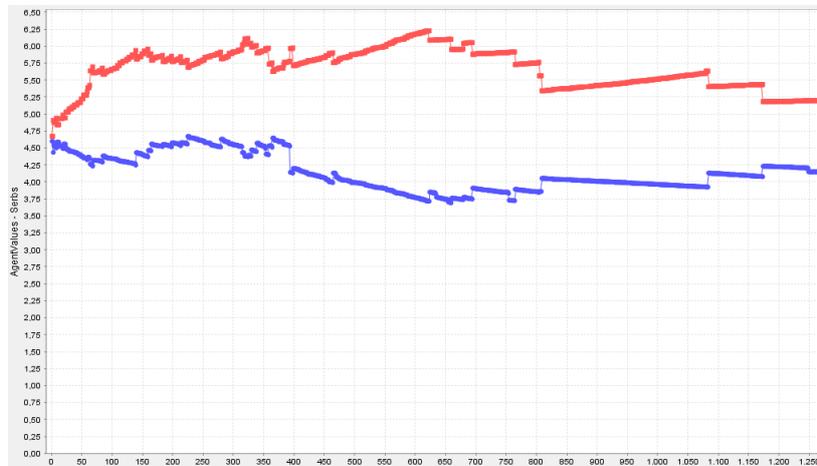


Abbildung 7.2: Simulation - Serbische Identitätsausprägungen 2. Lauf

Abbildung 7.2 illustriert dasselbe Diagramm, jedoch mit folgenden Parameter-Einstellungen:

- $demoValueDivider = 6$
- $avgOpinionMultiplikator = 2$

Auch wenn andere Einstellungen gewählt wurden, sind zum Vergleich mit dem Diagramm aus Abbildung 7.1 kaum größere Unterschiede feststellbar.

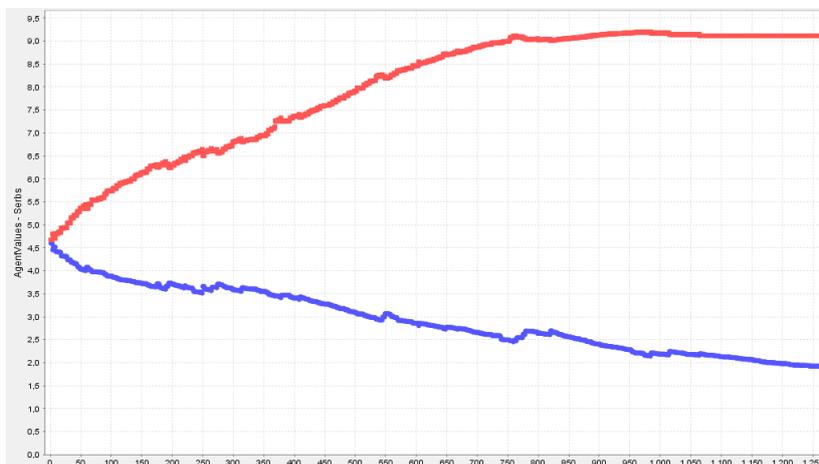


Abbildung 7.3: Simulation - Serbische Identitätsausprägungen 3. Lauf

Für diesen Simulationslauf wurden folgende Parameter-Einstellungen gewählt:

- $demoValueDivider = 5$
- $avgOpinionMultiplikator = 3$

Im Vergleich zum vorherigen Diagramm, wurde in diesem Simulationslauf der *avgOpinionMultiplikator* inkrementiert und der *demoValueDivider* dekrementiert. Hierbei ist feststellbar, dass die Distanz zwischen den beiden Kurven mit der Zeit immer größer wird.

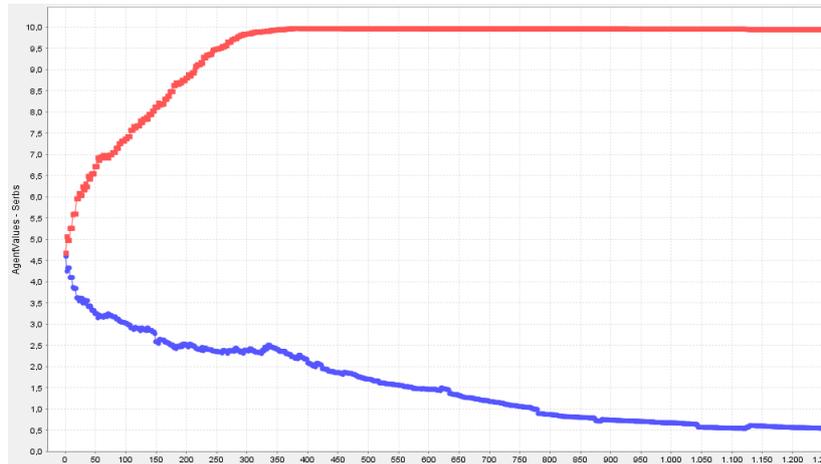


Abbildung 7.4: Simulation - Serbische Identitätsausprägungen 4. Lauf

Wird der *avgOpinionMultiplikator* noch weiter inkrementiert, so wird die Distanz der beiden Kurven noch größer. Hierfür wurden folgende Einstellungen gewählt:

- *demoValueDivider* = 5
- *avgOpinionMultiplikator* = 5

Letztendlich verdeutlicht das folgende Diagramm den Zusammenhang beider Parameter:

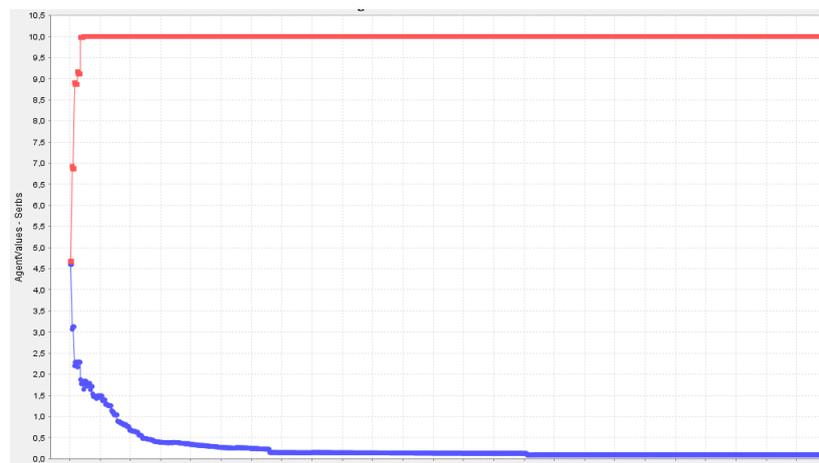


Abbildung 7.5: Simulation - Serbische Identitätsausprägungen 5. Lauf

- *demoValueDivider* = 1
- *avgOpinionMultiplikator* = 5

Wie beschrieben ist der “Average Opinion Multiplikator” der Gewichtungsfaktor der Summe der Bewertungen einer Rede der Nachbarn eines Agenten. Je größer dieser gewählt wird, desto größer ist der Unterschied zwischen den nationalen und zivilen Identitätsausprägung. Interpretieren lässt sich der Sachverhalt folgendermaßen: Je stärker die Meinung der Nachbarschaft auf den Bürger wirkt, desto stärker wird dieser auch von der Masse mitgezogen. Im Gegensatz dazu, gibt der “Demo Value Divider” nur an, wie schnell die jeweiligen Identitätsausprägungen wachsen sollen bzw. wie stark sich der jeweilige Bürger durch die Demonstrationsteilnahme radikalieren lässt. Der *demoValueDivider* ist aber ohne einen kräftigen *avgOpinionMultiplikator* unbrauchbar, d.h. ein Bürger dem die Meinung seiner Umgebung bzw. Nachbarschaft unwichtig ist, kann sich auch durch diese nicht radikalieren lassen.

## 7.2 Simulationsläufe

Die Simulationsläufe wurden auf einen 8-Kern Intel i7 2670QM Rechner mit 8GB RAM durchgeführt. Trotz dessen kam es bei Versuchen die Simulation mit ca. 15.000 Agenten durchzuführen oft nach einer bestimmten Zeit zu einem Heap Überlauf. Aus diesem Grund wurden die Simulationsläufe mit  $\frac{1}{5}$  der geplanten Agenten durchgeführt. Die Anzahl der Bürger-Agenten zum Zeitpunkt 1 ist in der folgenden Tabelle dargestellt:

Bosniaken	Kroaten	Serben
410	1021	1771

Tabelle 8: Anzahl der Bürger-Agenten in der Simulation

Die Anzahl der Politiker-Agenten zum Zeitpunkt 1 ist folgendermaßen aufgeteilt:

Republik	Bosniaken	Kroaten	Serben
Bosnien	9	4	7
Kroatien	1	30	2
Serbien	1	1	30

Tabelle 9: Anzahl der Politiker-Agenten in der Simulation

Zum Zeitpunkt 1 sieht die Bevölkerungsaufteilung folgendermaßen aus:

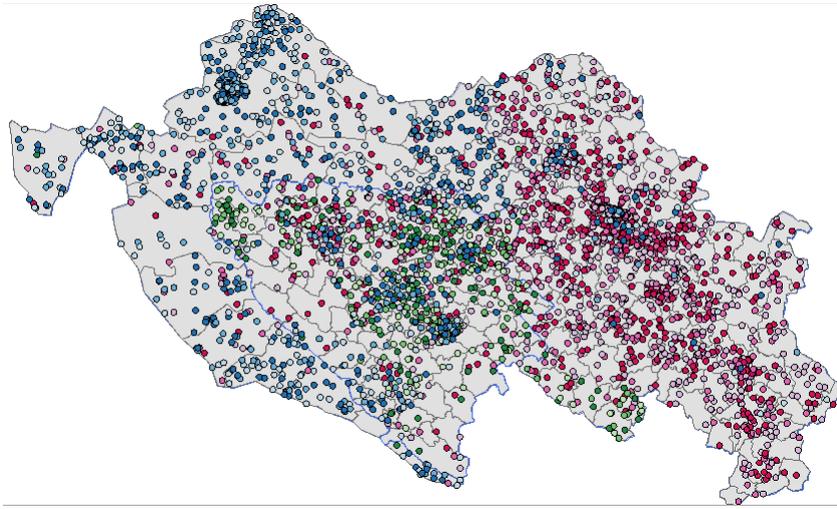


Abbildung 7.6: Simulationsläufe - Bevölkerungsübersicht - Tick 1

Die drei Bevölkerungsgruppen werden durch Farben unterschieden. Diese haben drei Intensitätsstufen, je nachdem wie stark die nationale Identität bei den Agenten ausgeprägt ist.

- **Erste Intensitätsstufe:** 0 - 3
- **Zweite Intensitätsstufe:** 4 - 8
- **Dritte Intensitätsstufe:** 9-10

Bosniaken	Kroaten	Serben
		

Tabelle 10: Farbendarstellung der Agenten

In den folgenden Abbildungen sind die Choroplethenkarten dargestellt. Sie zeigen die prozentualen Anteile der jeweiligen Ethnizität in einem bestimmten Verwaltungskreis. Diese werden auch durch drei Farbstufen unterschieden. Je stärker die Farbe, desto ethnisch reiner ist das jeweilige Gebiet. Für die drei Ethnizitäten gibt es somit je eine Choroplethenkarte.

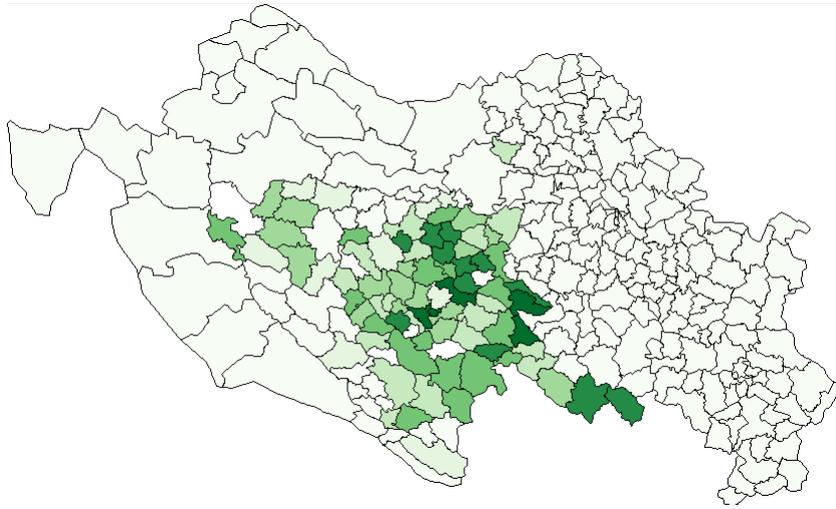


Abbildung 7.7: Simulationsläufe - Choroplethenkarte Bosniaken

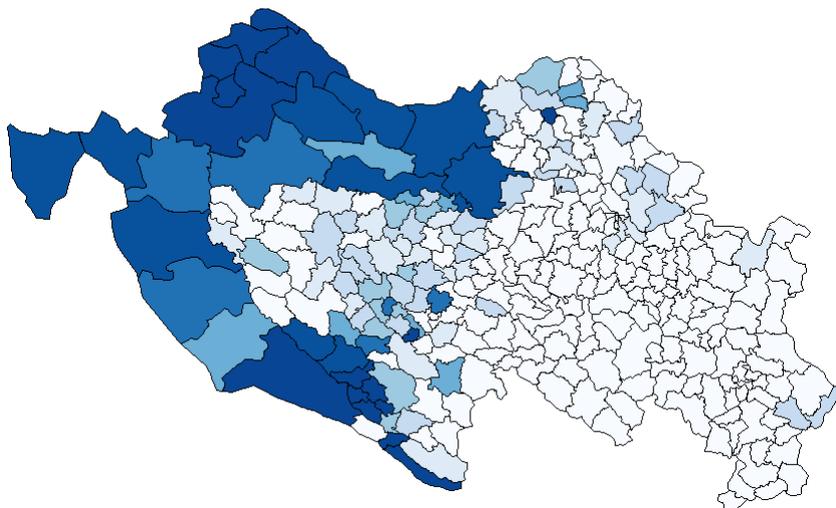


Abbildung 7.8: Simulationsläufe - Choroplethenkarte Kroaten

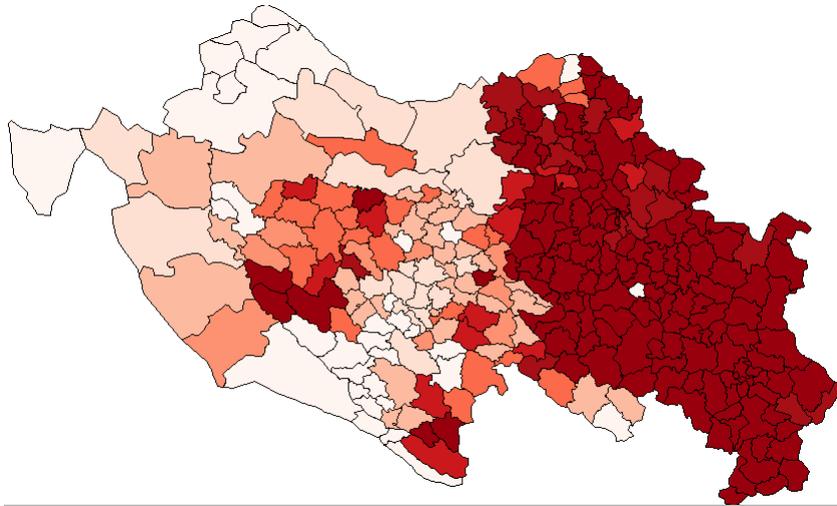


Abbildung 7.9: Simulationsläufe - Choroplethenkarte Serben

### 7.2.1 1. Simulationslauf

Für das erste Beispiel wurden bis auf die beiden Parameter *avgOpinionMultiplikator* und *demoValueDivider*, die Standardwerte für die anderen Parameter wie in Unterabschnitt 6.3.1 aufgeführt, benutzt. Die beiden Parameter wurden folgendermaßen gewählt:

- *demoValueDivider* = 7
- *avgOpinionMultiplikator* = 4

#### Bevölkerung

Nach 1820 Ticks was etwa einem Zeitraum von 5 Jahren entspricht, haben sich die Bevölkerungszahlen drastisch reduziert. Wie aus Abbildung 7.10 ersichtlich, gab es auch keine größeren Verlagerungen der Ethnien über die jeweiligen territorialen Grenzen hinaus.

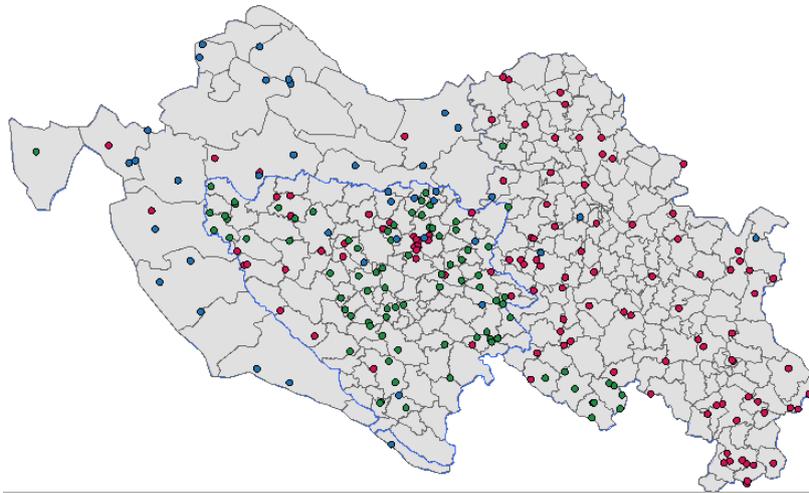


Abbildung 7.10: 1. Simulationslauf - Bevölkerungsübersicht - Tick 1820

Diese Bevölkerungsreduzierung verdeutlicht die Abbildung 7.11. Hierbei ist ersichtlich, dass die Bevölkerungszahlen der Kroaten und Serben sogar unter 10% zurückgegangen sind. Je größer eine Ethnie zum Anfangszeitpunkt vertreten war, desto stärker sind ihre Bevölkerungszahlen zurückgegangen. Offensichtlich standen die beiden größten Ethnien unter ständigem Konflikt. Interessant ist, dass die Bosniaken, auch wenn sie sich zwischen den Kroaten und Serben befunden haben, am wenigsten Schaden davongetragen haben.

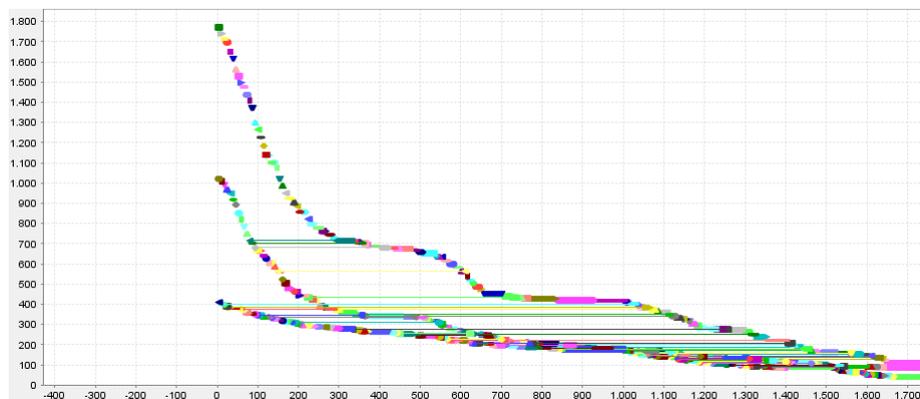


Abbildung 7.11: 1. Simulationslauf - Bevölkerungszahlen

Die folgenden Abbildungen verdeutlichen nochmal, dass die Ethnien zum größten Teil auf ihrem Territorium geblieben sind. Auf Grund der hohen Verluste, jedoch in vielen Teilen mit niedrigen prozentualen Anteilen.

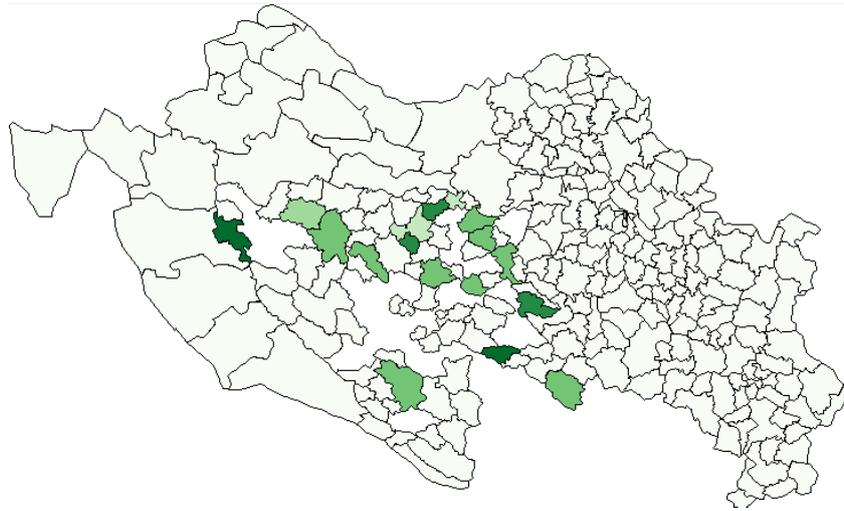


Abbildung 7.12: 1. Simulationslauf - Choroplethenkarte - Bosniaken

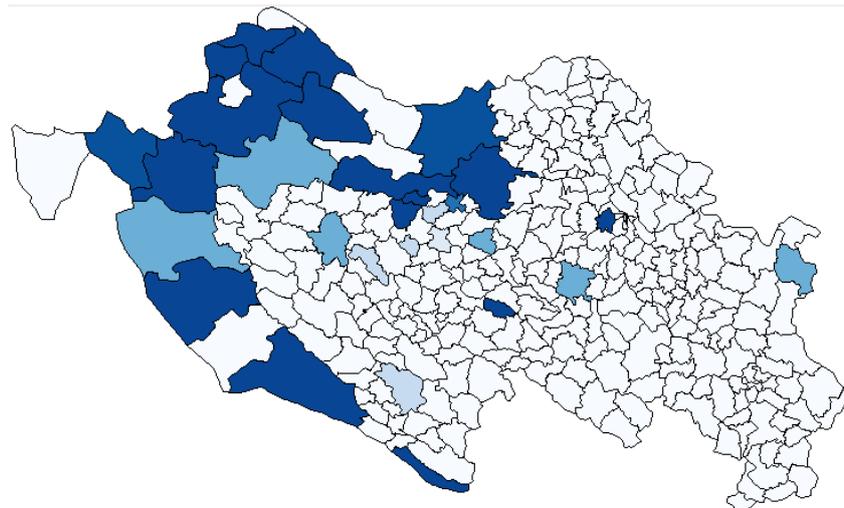


Abbildung 7.13: 1. Simulationslauf - Choroplethenkarte - Kroaten

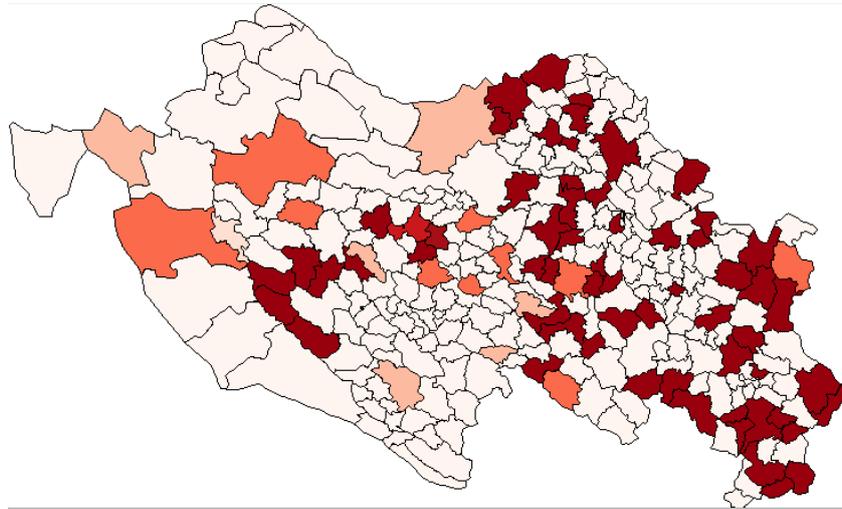


Abbildung 7.14: 1. Simulationslauf - Choroplethenkarte - Serben

Die Abbildungen 7.15, 7.16, 7.17 zeigen die nationale und zivile Identitätsausprägung der Ethnien. Die Kroaten und Serben haben schnell das durchschnittliche Maximum der nationalen Identitätsausprägung erreicht. Die zivile Identitätsausprägung hat während der Zeit langsam aber konstant nachgelassen. Die nationale Identitätsausprägung der Bosniaken<sup>1</sup> hat erst nach zwei Drittel der Zeit das Maximum erreicht. Je größer die Kluft zwischen den nationalen und zivilen Werten, desto gewaltbereiter sind die jeweiligen Agenten. Hierbei ist ersichtlich, dass die Serben die größte Kluft zwischen den Identitätsausprägungen haben und somit am radikalisierten sind.

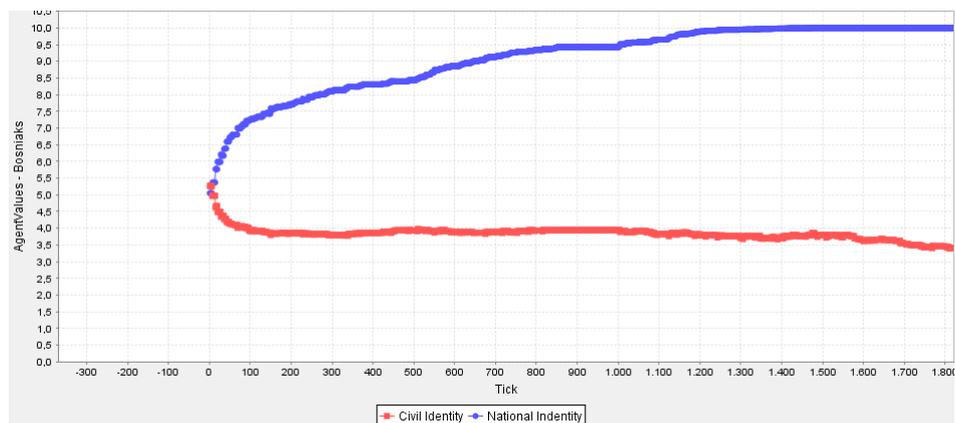


Abbildung 7.15: 1. Simulationslauf - Identitätsausprägung der Bosniaken

<sup>1</sup>Die nationale Identitätsausprägung wird im Diagramm 7.15 durch einen blauen Graphen dargestellt. Bei den anderen Diagrammen ist die nationale Identitätsausprägung durch einen roten Graphen dargestellt.

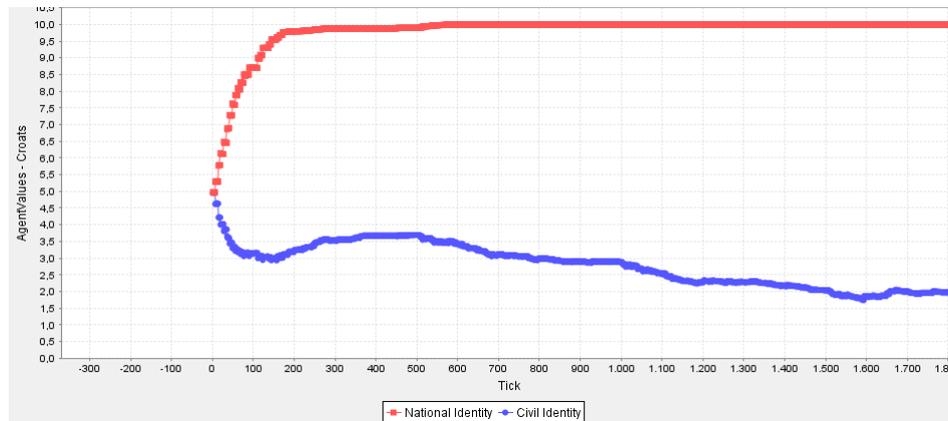


Abbildung 7.16: 1. Simulationslauf - Identitätsausprägung der Kroaten

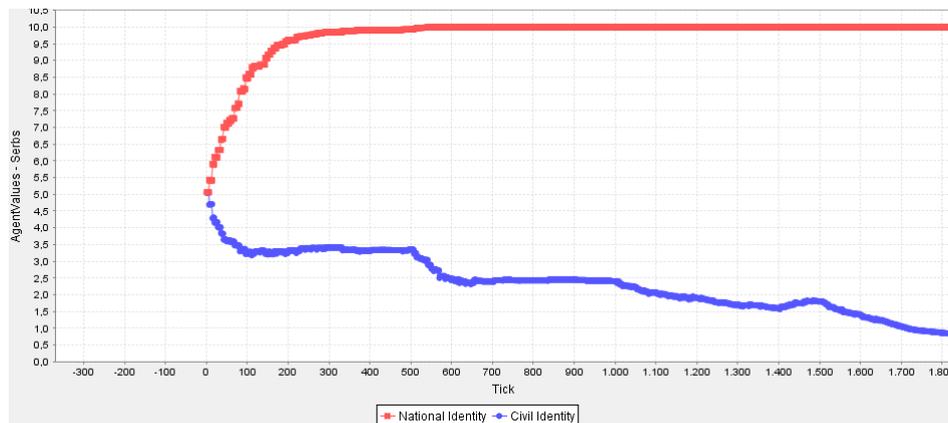


Abbildung 7.17: 1. Simulationslauf - Identitätsausprägung der Serben

### Politiker

In den folgenden Abbildungen sind die ersten 13 Politiker der jeweiligen Teilrepublik dargestellt. Was die politische Stimmung in Bosnien anbetrifft, so ist ersichtlich, dass von den 9 bosniakischen Politiker (Tabelle 9), sich nur drei deutlich mit nationalen Reden an die Bürger gewandt haben. Nur ein bosniakischer Politiker befindet sich an 12. Position der sich für die zivile Redehaltung entschieden hat. Bei den kroatischen und serbischen Politikern in Bosnien sieht es sehr ausgeglichen aus. Von diesen Politikern hat sich keiner deutlich für die nationale Reden entschieden. Dies erklärt auch das mäßige Ranking. Die erste und zweite Position liefern das Beispiel, dass nicht unbedingt die Anzahl der Reden ausschlaggebend ist, sondern dass die Glaubwürdigkeit der Reden eine sehr große Rolle spielt.

```

-----
Bosnia
-----

```

PolID	Ethnicity	NumSpeeches	STypes	Popularity	Avg
0	Bosniak	50	A1:12-A2:0-A3:3	269,00	5,38
5	Bosniak	55	A1:16-A2:6-A3:4	327,00	5,95
86	Bosniak	10	A1:3-A2:0-A3:2	63,00	6,30
8	Bosniak	59	A1:11-A2:4-A3:5	479,00	8,12
11	Croat	55	A1:9-A2:6-A3:3	459,00	8,35
15	Serb	58	A1:0-A2:0-A3:19	488,00	8,41
16	Serb	58	A1:9-A2:7-A3:5	490,00	8,45
18	Serb	55	A1:8-A2:8-A3:3	472,00	8,58
17	Serb	58	A1:1-A2:18-A3:0	509,00	8,78
13	Serb	56	A1:5-A2:4-A3:5	525,00	9,38
14	Serb	55	A1:9-A2:3-A3:4	525,00	9,55
3	Bosniak	59	A1:0-A2:18-A3:1	579,00	9,81
9	Croat	51	A1:2-A2:17-A3:1	503,00	9,86

Abbildung 7.18: 1. Simulationslauf - Politiker in Bosnien

In Kroatien ist die politische Struktur auf Grund der überwiegenden kroatischen Bevölkerungsstruktur sehr monoton. Auf den ersten 7 Stellen haben sich klar die nationalen kroatischen Politiker abgesetzt. Auch das sehr gute Ranking des ersten Politikers verdeutlicht, dass die nationale Identität bei den Kroaten sehr ausgeprägt ist, zugleich aber auch verdeutlicht es die distanzierte Haltung gegenüber den zivilen Werten.

```

-----
Croatia
-----

```

PolID	Ethnicity	NumSpeeches	STypes	Popularity	Avg
48	Croat	24	A1:8-A2:0-A3:2	73,00	3,04
88	Croat	6	A1:2-A2:0-A3:0	20,00	3,33
35	Croat	31	A1:9-A2:1-A3:0	129,00	4,16
24	Croat	30	A1:9-A2:0-A3:3	150,00	5,00
36	Croat	36	A1:6-A2:4-A3:0	234,00	6,50
45	Croat	30	A1:8-A2:5-A3:3	202,00	6,73
28	Croat	27	A1:4-A2:3-A3:0	185,00	6,85
87	Croat	6	A1:1-A2:1-A3:0	42,00	7,00
43	Croat	27	A1:4-A2:0-A3:4	212,00	7,85
52	Serb	29	A1:1-A2:10-A3:0	233,00	8,03
41	Croat	31	A1:5-A2:1-A3:3	250,00	8,06
39	Croat	30	A1:1-A2:10-A3:0	244,00	8,13
27	Croat	29	A1:3-A2:0-A3:2	237,00	8,17

Abbildung 7.19: 1. Simulationslauf - Politiker in Kroatien

Serbien liefert ein ähnliches Bild wie Kroatien ab. Die ersten Stellen sind wieder durch Politiker belegt, die sich mit nationalen Reden an die Bürger gewandt haben.

```

-----
Serbia
-----

```

PolID	Ethnicity	NumSpeeches	STypes	Popularity	Avg
76	Serb	33	A1:11-A2:1-A3:3	94,00	2,85
81	Serb	31	A1:9-A2:0-A3:2	107,00	3,45
78	Serb	29	A1:4-A2:2-A3:0	161,00	5,55
64	Serb	30	A1:9-A2:4-A3:1	178,00	5,93
84	Serb	30	A1:6-A2:0-A3:5	184,00	6,13
65	Serb	37	A1:7-A2:2-A3:2	234,00	6,32
77	Serb	32	A1:8-A2:7-A3:1	215,00	6,72
56	Serb	32	A1:7-A2:3-A3:3	236,00	7,38
71	Serb	36	A1:7-A2:4-A3:3	270,00	7,50
67	Serb	32	A1:5-A2:1-A3:0	248,00	7,75
90	Serb	7	A1:0-A2:2-A3:0	56,00	8,00
75	Serb	38	A1:7-A2:8-A3:0	307,00	8,08
73	Serb	33	A1:0-A2:0-A3:11	284,00	8,61

Abbildung 7.20: 1. Simulationslauf - Politiker in Serbien

In allen Teilrepubliken bilden die Politiker die Spitze der Liste, die mehrheitlich nationale Reden gehalten haben. Des Weiteren gibt es in jeder Teilrepublik einen Politiker, der deutlich vor den anderen Politikern abgesetzt ist. Tatsächlich gab es immer eine politische Figur, die durch ihre nationale Haltung aus der politischen Masse besonders herausstach. Diese Figur hatte zugleich auch die politische Führungsrolle.

### 7.2.2 2. Simulationslauf

Für das zweite Beispiel wurden folgende Parameter benutzt:

- *demoValueDivider* = 7
- *avgOpinionMultiplikator* = 2,3
- *mooreRange* = 2

#### Bevölkerung

Im zweiten Simulationslauf sind die Bevölkerungszahlen nach 1820 Ticks deutlich höher als im ersten Simulationslauf. Dies ist darauf zurückzuführen, dass nicht so viele Milizen gebildet wurden wie im ersten Simulationslauf. Aus Abbildung 7.21 ist anhand der Farben der Bürger ersichtlich, dass die Bevölkerung nicht so stark radikalisiert wurde.

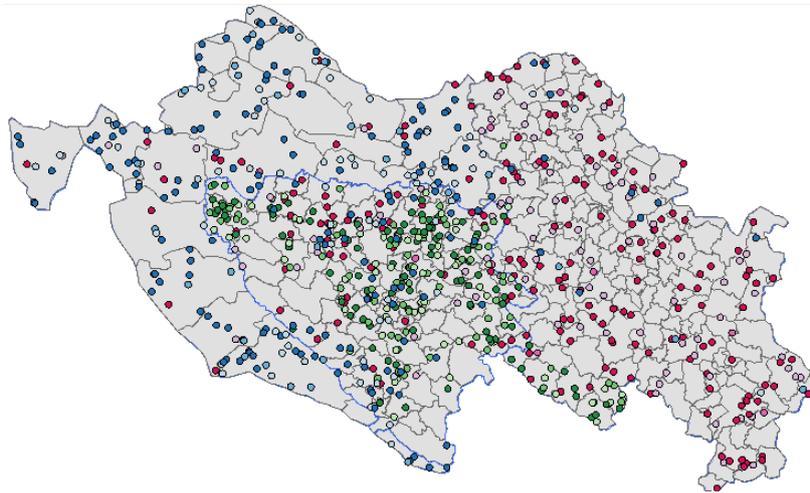


Abbildung 7.21: 2. Simulationslauf - Bevölkerungsübersicht - Tick 1820

Abbildung 7.22 zeigt den Bevölkerungsrückgang. Nach etwa einem Drittel der Zeit haben sich die Zahlen einigermaßen stabilisiert. Hierbei ist ersichtlich, dass auch hier die zwei größeren Ethnien am meisten Schaden davongetragen haben.

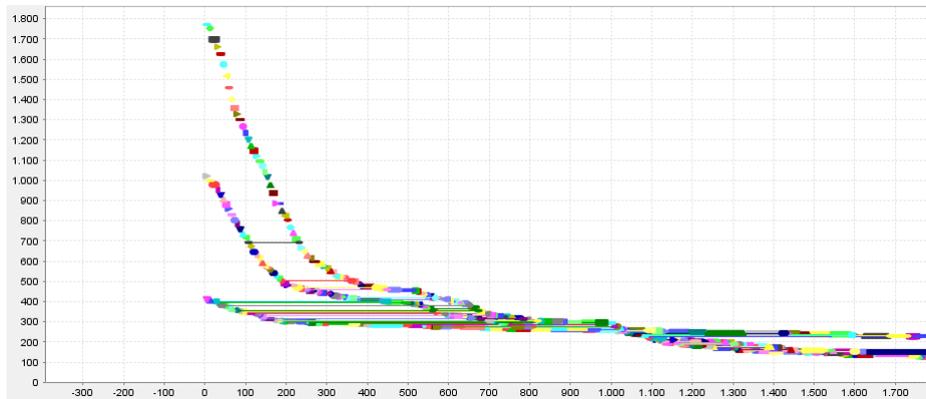


Abbildung 7.22: 2. Simulationslauf - Bevölkerungszahlen

Die Abbildungen 7.23, 7.24 und 7.25 zeigen, dass auch in diesem Simulationslauf die Ethnien auf ihren Territorien geblieben sind und dass es auch hier nur minimale Verlagerungen gab.

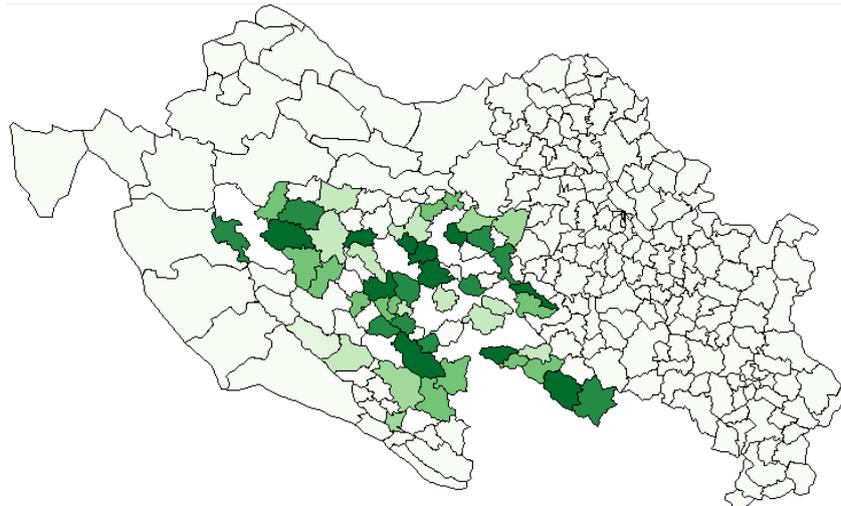


Abbildung 7.23: 2. Simulationslauf - Choroplethenkarte - Tick 1820

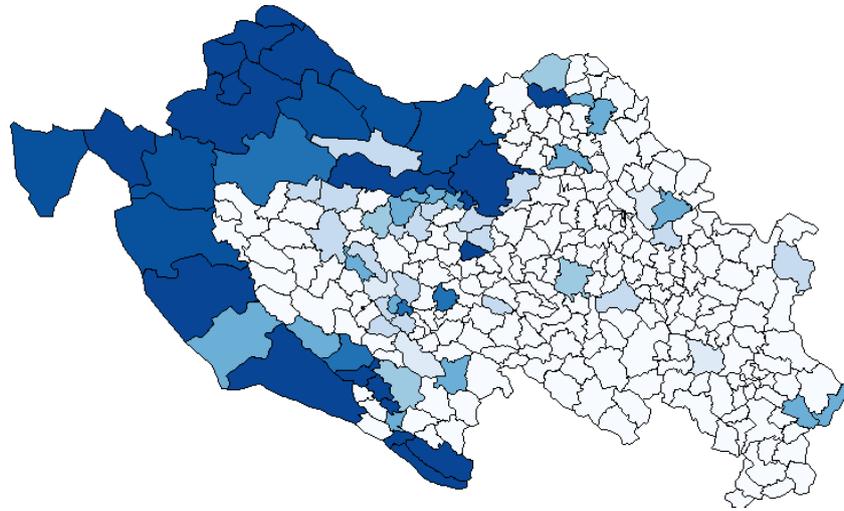


Abbildung 7.24: 2. Simulationslauf - Choroplethenkarte - Tick 1820

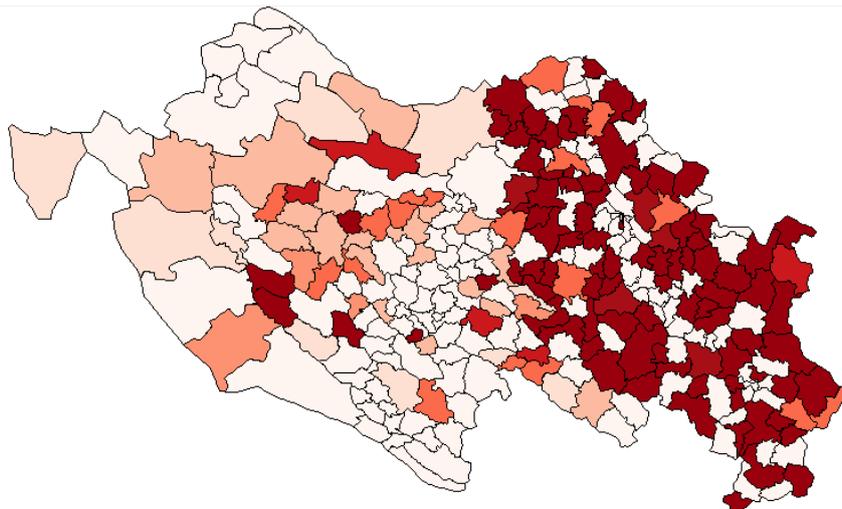


Abbildung 7.25: 2. Simulationslauf - Choroplethenkarte - Tick 1820

Die Bosniaken haben sich während der Simulation nur minimal radikalisiert. Die Kluft zwischen der nationalen und zivilen Identitätsausprägung ist einigermaßen konstant geblieben.

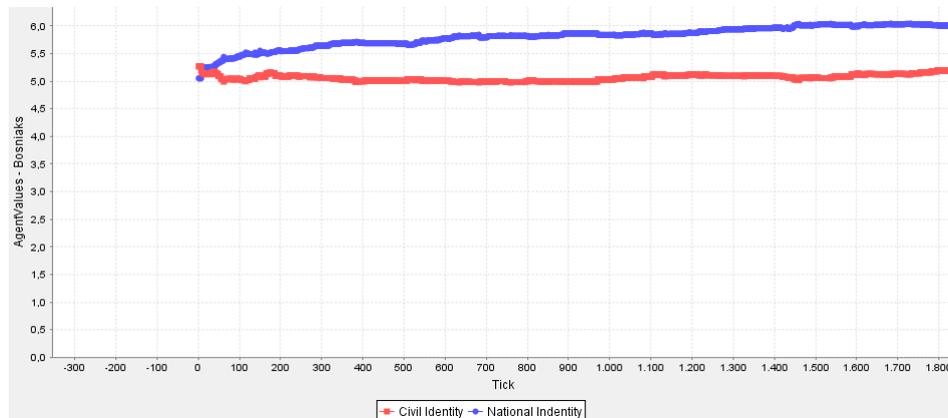


Abbildung 7.26: 2. Simulationslauf - Identitätsausprägung

Bei den Kroaten und Serben gab es in der Anfangsphase eine größere Steigerung der nationalen Identität. Diese hat nach ca. zwei Drittel der Zeit ihr Maximum erreicht. Dies ist auch der Punkt an dem sich die Bevölkerungszahlen stabilisieren. Zu gleich hat die zivile Identitätsausprägung einen stetigen Wachstum nachvollzogen, so dass sich die Kluft zwischen den beiden Ausprägungen immer weiter zugezogen hat.

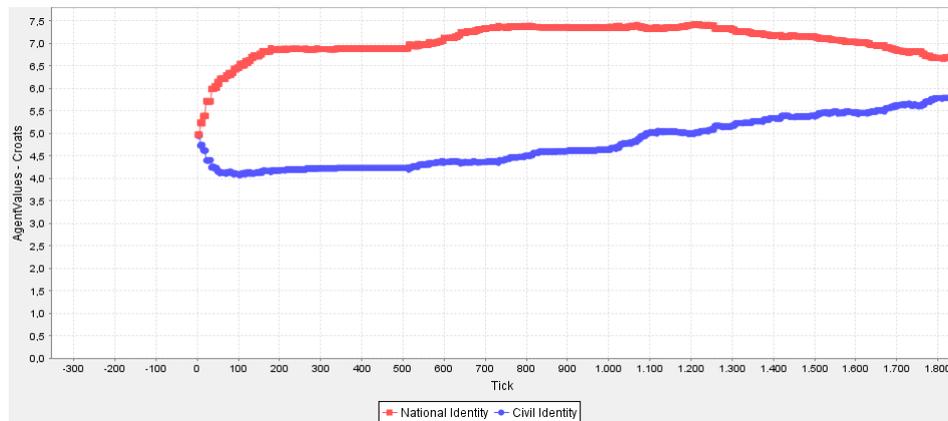


Abbildung 7.27: 2. Simulationslauf - Identitätsausprägung

Dem Trend nach, würden im weiteren Verlauf der Simulation die nationalen Werte immer weiter zurückgehen und dafür die zivilen Werte immer weiter steigen. Dies wäre zugleich die Bedingung für eine friedliche Koexistenz der drei Ethnien.

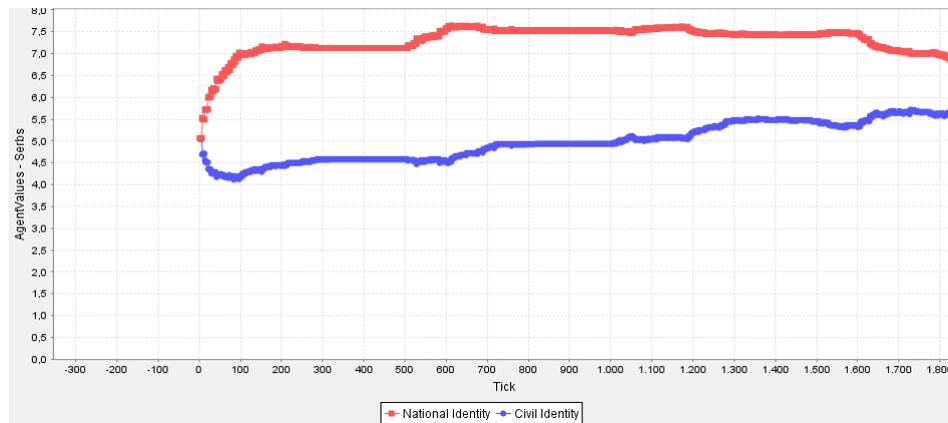


Abbildung 7.28: 2. Simulationslauf - Identitätsausprägung

### Politiker

Die politische Struktur in Bosnien ist überwiegend zivil ausgeprägt. Zwar gibt es Politiker, die sehr gute Ergebnisse mit nationalen Reden erreicht haben, jedoch reichen diese nicht aus, um die Bevölkerung noch stärker zu radikalisieren.

-----  
Bosnia  
-----

PolID	Ethnicity	NumSpeeches	STypes	Popularity	Avg
7	Bosniak	49	A1:12-A2:1-A3:3	278,00	5,67
13	Serb	54	A1:0-A2:17-A3:0	325,00	6,02
19	Serb	46	A1:1-A2:15-A3:0	277,00	6,02
17	Serb	47	A1:0-A2:15-A3:0	289,00	6,15
86	Bosniak	13	A1:1-A2:4-A3:0	86,00	6,62
15	Serb	53	A1:13-A2:2-A3:5	353,00	6,66
4	Bosniak	48	A1:0-A2:16-A3:1	331,00	6,90
5	Bosniak	51	A1:1-A2:16-A3:0	353,00	6,92
6	Bosniak	52	A1:2-A2:16-A3:1	370,00	7,12
14	Serb	54	A1:0-A2:16-A3:1	389,00	7,20
1	Bosniak	49	A1:1-A2:15-A3:0	360,00	7,35
11	Croat	45	A1:6-A2:4-A3:2	363,00	8,07
9	Croat	52	A1:1-A2:16-A3:2	448,00	8,62

Abbildung 7.29: 2. Simulationslauf - Politiker in Bosnien

Auch in Kroatien ist die politische Struktur überwiegend zivil ausgeprägt. Es ist aber ersichtlich, dass viele Politiker oft zwischen nationalen und zivilen Reden geschwankt haben.

```

-----
Croatia
-----

```

PolID	Ethnicity	NumSpeeches	STypes	Popularity	Avg
44	Croat	34	A1:0-A2:11-A3:0	178,00	5,24
40	Croat	33	A1:8-A2:0-A3:3	185,00	5,61
33	Croat	33	A1:3-A2:7-A3:0	200,00	6,06
49	Croat	29	A1:1-A2:10-A3:0	176,00	6,07
88	Serb	14	A1:0-A2:4-A3:0	87,00	6,21
24	Croat	25	A1:1-A2:7-A3:0	156,00	6,24
50	Croat	31	A1:8-A2:1-A3:2	196,00	6,32
27	Croat	25	A1:5-A2:4-A3:0	159,00	6,36
39	Croat	33	A1:6-A2:3-A3:3	214,00	6,48
25	Croat	30	A1:8-A2:1-A3:4	212,00	7,07
36	Croat	31	A1:1-A2:9-A3:0	220,00	7,10
34	Croat	35	A1:7-A2:0-A3:3	253,00	7,23
48	Croat	35	A1:3-A2:10-A3:0	264,00	7,54

Abbildung 7.30: 2. Simulationslauf - Politiker in Kroatien

In Serbien wird die Spitze der Liste von den national-orientierten Politikern geführt, jedoch in einer sehr kleinen Zahl. Die meisten Politiker in Serbien haben sich jedoch für zivile Reden entschieden.

```

-----
Serbia
-----

```

PolID	Ethnicity	NumSpeeches	STypes	Popularity	Avg
76	Serb	39	A1:9-A2:3-A3:0	180,00	4,62
73	Serb	36	A1:10-A2:1-A3:3	199,00	5,53
56	Serb	36	A1:10-A2:0-A3:3	202,00	5,61
78	Serb	34	A1:10-A2:2-A3:2	209,00	6,15
55	Serb	35	A1:0-A2:12-A3:1	223,00	6,37
82	Serb	40	A1:1-A2:13-A3:0	261,00	6,53
69	Serb	34	A1:0-A2:10-A3:0	226,00	6,65
57	Serb	38	A1:1-A2:12-A3:0	262,00	6,89
75	Serb	32	A1:2-A2:10-A3:0	224,00	7,00
81	Serb	34	A1:0-A2:10-A3:1	239,00	7,03
79	Serb	36	A1:4-A2:8-A3:1	265,00	7,36
74	Serb	36	A1:0-A2:10-A3:0	270,00	7,50
61	Serb	37	A1:1-A2:10-A3:1	279,00	7,54

Abbildung 7.31: 2. Simulationslauf - Politiker in Serbien

## 8 Zusammenfassung und Ausblick

Das Konflikt-Modell zeigt, dass ethnische Konflikte durch die politische Ebene ausgelöst werden, aber nur dann, wenn es in der jeweiligen ethnischen Gruppe auf jeden Fall Menschen gibt, die von Anfang an stark nationalisiert sind und eigene Ziele verfolgen. Zudem muss es in der jeweiligen Ethnizität Menschen geben, die sich leicht beeinflussen lassen und so für die Erreichung bestimmter Ziele instrumentalisiert werden können. Je mehr Politiker sich mit dem nationalen Redetyp an die Bevölkerung wenden, desto stärker wird auch die Radikalisierung vorangetrieben.

Zudem ist beobachtbar, dass ethnische Konflikte besonders dann entstehen, wenn eine Ethnie starke Unterstützung von außerhalb bekommt. Auch die zahlenmäßige Vertretung der Ethnizitäten spielen auch eine erhebliche Rolle. Größere Ethnizitäten werden dem Konflikt-Modell nach schneller radikalisiert und sind somit gewaltbereiter. Die kleinste Ethnizität mit minimaler Unterstützung von außerhalb ist somit dem Konflikt-Modell nach, nur mäßig zu radikalisieren. Die Gewaltbereitschaft erhöht sich erst dann deutlich, wenn es zur Vertreibung durch die Milizen kommt. Kommt es zu einer ethnischen Säuberung einer Ethnie durch eine Andere, so ist dieser Prozess von sich selbst aus nicht zu stoppen. Dies verdeutlichen die Kurvenverläufe der Bevölkerung in den beiden Simulationsläufen in Kapitel 7. Dieser Prozess kommt erst dann zum Halt, wenn entweder die Ziele der Ethnie erreicht worden sind oder durch eventuelle Einwirkung von außerhalb.

Das Konflikt-Modell verdeutlicht Zusammenhänge bestimmter Einflussgrößen, die in der Realität komplexe soziale Verhaltensweisen darstellen. Eine wichtige Frage die vom soziologischen Standpunkt genauer untersucht werden muss, ist die Frage, wie schnell sich die Menschen radikalieren bzw. mobilisieren lassen. Dieser Prozess ist oft mit dem geschichtlichen Hintergrund verbunden. Historisch betrachtet war z.B. der Bosnien-Krieg, in dem ein Genozid an der bosniakischen Bevölkerung seitens der Serben verübt wurde, nicht der erste ethnische Konflikt in diesem Gebiet. Seit dem 18. Jahrhundert gab es an der bosniakischen Bevölkerung stets ethnische Säuberungen [6]. Auch wenn die Hauptursachen solcher ethnischer Säuberungen stets ideologische und territoriale Motive waren, so bleibt es zu klären, wie eine solche Idee über Generationen hinweg erhalten bleibt. Aber auch die Frage, wie stark Menschen manipuliert werden können und wie stark ihre Umgebung auf sie einwirkt, ist ein weiterer Punkt für die soziologische Untersuchung.

Die agentenbasierte Simulation eignet sich unter anderem besonders gut für die Untersuchung von Systemen im soziologischen Bereich. Das Konflikt-Modell zeigt, dass auch ethnische Konflikte sehr gut modelliert werden können. Für das Modell könnten passende Parameter gefunden werden, die es realitätsnaher bzgl. der Bevölkerungszahlen darstellen. Es lässt sich auch beliebig erweitern. So könnte z.B. eine politische Struktur geschaffen werden, die verschiedene initiale Wahrscheinlichkeiten für die Auswahl der Reden besitzt. Hierbei ist jedoch eine Erweiterung an dem EmIL Framework notwendig. Auch könnte der Akteur Bürger erweitert werden und so durch Normen entscheiden, in wie weit er die kollektive Meinung seiner Nachbarschaft akzeptiert. Aber auch könnten die

historischen Ursachen im Modell abgebildet werden, so dass eine Ethnie stets eine bestimmte Ethnie angreift. Ein nicht zu unterschätzender Faktor ist die wirtschaftliche Lage. Diese könnte in der weiteren Entwicklung des Modells mit betrachtet werden. Es bleibt natürlich offen, welches Gebiet ethnischer Konflikte weiter untersucht werden soll. Das Konflikt-Modell bietet auf jeden Fall die Basis und die Flexibilität die Untersuchung in beliebiger Richtung fortzuführen.

## Literatur

- [1] S. A. Abdulkareem. Simulating the spread of pertussis in enschede region using-agent. 2010.
- [2] Agency for Statistics of Bosnia and Herzegovina. Census 1991, 1991.
- [3] ArcGIS Resource Center. Arcgis dektop 10.
- [4] Argonne National Laboratory. Repast for high performance computing.
- [5] H. Ball. *Genocide: A Reference Handbook*. ABC-CLIO, 2010.
- [6] Bosnjaci.net. 11 genocides on bosniaks, 11.01.2004.
- [7] M.J Calic. *Geschichte Jugoslawiens im 20. Jahrhundert*. Beck C. H, 2010.
- [8] Central Intelligence Agency. Ethnic groups in yugoslavia: Making the history of 1989.
- [9] N. Collier. Repast simphony : Projections, 12.05.2008.
- [10] Croatian Bureau of Statistics. Census 1991.
- [11] A. T. Crooks. The repast simulation / modelling system for geospatial simulation: Working working papers series. (1467-1298).
- [12] M. Czeranka. Arcgis-schulungshandbuch, 2004.
- [13] EmIL project team. Emil emergence in the loop: simulating the two way dynamics of norm innovation: Deliverable 5.1.
- [14] T. Fleiner and L.R.B Fleiner. *Allgemeine Staatslehre: über die konstitutionelle Demokratie in einer multikulturellen globalisierten Welt*. Springer, 2004.
- [15] G.N Gilbert and Troitzsch K.G. *Simulation for the social scientist*. Open University Press, 2005.
- [16] H. Hummel and B. Wehrhöfer. *Geopolitische Identitäten: Kritik der Ethnisierung einer sich regionalisierenden Welt als paradigmatische Erweiterung der Friedensforschung*. Seminar für Politikwiss. und Soziologie, TU Braunschweig, 1996.
- [17] W. Ismayr, S. Richter, and M. Soldner. *Die politischen Systeme Osteuropas*. VS Verlag für Sozialwissenschaften, 2010.
- [18] D. Levinson. *Ethnic groups worldwide: a ready reference handbook*. Oryx Press, 1998.
- [19] C. M. Macal and M.J North. Tutorial on agent-based modelling and simulation: Journal of simulation (2010) 4, 151–162, 2010.
- [20] C.M Macal and M.J North. Introduction to agent-based modeling and simulation, 2006.
- [21] N. Malleson. Repast simphony tutorial, 2008.

- [22] G. Meyer and A. Thimm. *Ethnische Konflikte in der Dritten Welt: Ursachen und Konsequenzen*. Johannes Gutenberg-Universität Mainz, 2001.
- [23] M. Neumann. *Pathways to ethnic cleansing*. Proceedings of the 3rd World Congress on Social Simulation WCSS2010 (CD-ROM), 2010.
- [24] P3C Statistical Office of the Republic of Serbia. Census 1991.
- [25] M. Pauli. Entwurf und implementierung des simulationsszenarios 'micro-finance' in emil-s. 2009.
- [26] Robin Emde. Entwurf und implementierung des simulationsszenarios 'col-laborativewriting' in emil-s.
- [27] G. Schlee. Ethnien und religion sind keine kriegsursachen.
- [28] science@Orf.at. Experte: Jugoslawien nicht national aufgeladen, 2011.
- [29] H. Sundhaussen. Der zerfall jugoslawiens und dessen folgen: Aus politik und zeitgeschichte (apuz 32/2008).
- [30] E. Tartara. Repast reference: Repast simphony : Context.
- [31] E. Tartara. Repast simphony : Working with the scheduler.
- [32] P. Ther and H. Sundhaussen. *Nationalitätenkonflikte im 20. Jahrhundert: Ursachen von inter-ethnischer Gewalt im Vergleich*. Harrassowitz, 2001.