

# **Entwurf und Implementierung des Simulationsszenarios 'Micro Finance' in EMIL-S**

Bachelorarbeit

zur Erlangung des Grades eines Bachelor of Science im Studiengang  
Informationsmanagement

vorgelegt von

Manuel Pauli

Erstgutachter: Prof. Dr. Klaus G. Troitzsch

Zweitgutachter: Dipl.-Inform. Ulf Lotzmann

Koblenz, im September 2009

# **Erklärung**

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

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

Koblenz, September 2009

---

## Inhaltsverzeichnis

1 Einleitung.....	1
1.1 Problemstellung.....	1
1.2 Aufbau der Arbeit.....	2
2 Arbeitsgrundlagen.....	3
2.1 Micro Finance.....	3
2.1.1 Die Grundlagen von Micro Finance.....	3
2.1.2 Die Micro Finance Studien von Pablo Lucas.....	4
2.2 Das EMIL Projekt.....	6
2.2.1 Die Bedeutung von Simulationen und die Einordnung von EMIL.....	6
2.2.2 Der EMIL Ansatz.....	9
2.2.3 Die Arbeitsschritte des EMIL Projekts.....	10
2.2.3.1 EMIL-M.....	10
2.2.3.2 EMIL-A.....	10
2.2.3.3 EMIL-S.....	12
2.2.3.4 EMIL-T.....	13
2.2.4 Teilnehmer des Projekts.....	14
2.2.5 Die EMIL Simulation Suite.....	14
2.2.6 Der Aufbau von EMIL-S.....	15
2.2.7 Das EMIL-S GUI .....	19
2.2.8 XML als Modellierungssprache für EMIL-S.....	21
2.3 RePast als Simulationsumgebung.....	23
3 Durchführung.....	25
3.1 Modellierung des EMIL-S Micro-Finance Szenarios .....	25
3.1.1 Anfangsversion.....	25
3.1.2 Endversion.....	29
3.2 Die Simulationsumgebung oder RePast Ebene .....	33
3.2.1 Die Schnittstelle RePast / EMIL-S.....	33
3.2.2 Programmierung der Modellumgebung.....	35

---

3.2.2.1 Die Grundzüge.....	35
3.2.2.2 Eine beispielhafte Kommunikation mit EMIL-S.....	36
3.2.2.3 Ereigniseintritt basierend auf gespeicherten Nachrichten.....	37
3.2.2.4 Die Berechnung von Gruppenentscheidungen.....	39
3.2.2.5 Lernen durch Auslösung von Sanktionen.....	40
3.3 Simulationsläufe und Verhalten des Szenarios.....	41
3.3.1 Vorbemerkungen zur Analyse.....	41
3.3.2 Simulationsläufe.....	43
4 Zusammenfassung und Ausblick.....	53

## Abbildungsverzeichnis

Abbildung 1: Eine Frage aus Pablo Lucas' Fragebogen an MF-Gruppenmitglieder, Quelle: [PLR.08]	5
Abbildung 2: Verhalten in MF-Gruppen, Quelle: Pablo Lucas, [PLR.08]	5
Abbildung 3: Gruppenabläufe in Pablo Lucas Modell, Quelle: Pablo Lucas, [PLC.09]	6
Abbildung 4: "The logic of simulation as a method", Quelle: [SSS.05], Seite 17	7
Abbildung 5: "A comparison of social science simulation techniques", Quelle: [SSS.05], Seite 13	8
Abbildung 6: Quelle: EMIL-S Projekt , Deliverable 1.2, Seite 7, [D12.07]	12
Abbildung 7: Die Emil Simulation Suite, Quelle: [SNB.09]	15
Abbildung 8: EMIL-S Baum-Modell	16
Abbildung 9: Event Board, Quelle: [SNB.09]	17
Abbildung 10: Normative Frame, Quelle: [SNB.09]	18
Abbildung 11: Norm Adoption, Quelle: [SNF.08]	19
Abbildung 12: EMIL-S GUI (1/2)	20
Abbildung 13: EMIL-S GUI (2/2)	20
Abbildung 14: EMIL-S XML-Tester	21
Abbildung 15: Micro Finance XML Regelsatz: Ereignisdefinitionen	22
Abbildung 16: Micro Finance XML Regelsatz: Aktionsdefinitionen	22
Abbildung 17: Micro Finance XML Regelsatz: Regeldefinitionen	22
Abbildung 18: Micro Finance XML Regelsatz: Ereignisdefinitionen und Normative Board	23
Abbildung 19: Micro Finance UML Modell - Anfangsversion	26
Abbildung 20: EMIL-S Micro Finance Baummodell - Anfangsversion	27
Abbildung 21: Micro Finance UML Modell - Endversion	29
Abbildung 22: EMIL-S Micro Finance Baummodell - Endversion (1/2)	31
Abbildung 23: EMIL-S Micro Finance Baummodell - Endversion (2/2)	32
Abbildung 24: Klasse MicroFinanceGeneral: Aufruf des XML Regelsatzes	33
Abbildung 25: Klasse MicroFinanceGroupMember: zu importierende Klassen	33
Abbildung 26: Klasse MicroFinanceGroupMember: IEMILAgentWrapper	33
Abbildung 27: Klasse MicroFinanceGroupMember: Senden einer Nachricht an EMIL-S, die ein Ereignis bedeutet	34
Abbildung 28: Klasse MicroFinanceGroupMember: Empfangen von Nachrichten	34
Abbildung 29: Klasse MicroFinancialInstitution (Bank): Konstruktor	34
Abbildung 30: Die Schnittstelle EMIL-S/RePast, Quelle: Ulf Lotzmann/Michael Möhring, Konferenzvortrag Universität Koblenz, 29.01.2009	35
Abbildung 31: Klasse MicroFinanceGeneral: Quartal-einleitende Methode	36
Abbildung 32: Klasse MicroFinanceGeneral: Information der Agenteninstanzen über ein neues Ereignis	36
Abbildung 33: Klasse MicroFinanceGroupMember: Der RePast/JAVA Agent informiert seine EMIL-S Ebene über ein Ereignis	36
Abbildung 34: Klasse MicroFinanceGroupMember: Der RePast/JAVA Agent erhält die Reaktion der EMIL-S Ebene zu einem Ereignis	37
Abbildung 35: Klasse OriginalMessage: stellt ein Format bereit in dem Nachrichten gespeichert werden	38
Abbildung 36: Klasse MicroFinanceGeneral: Ereignisse werden abhängig von Nachrichten der Vergangenheit ausgelöst	38
Abbildung 37: Klasse MicroFinanceGeneral: Die Gruppenmitglieder werden für eine Gruppenentscheidung befragt	39
Abbildung 38: Klasse MicroFinanceGeneral: Die Gruppenentscheidung wird gewichtet und	

---

ausgeföhrt.....	40
Abbildung 39: alle Agentenklassen: explizite Sanktion wird der EMIL-S Ebene gemeldet.....	41
Abbildung 40: Legende für Simulationsgraphen.....	42
Abbildung 41: Simulationslauf 1.....	45
Abbildung 42: RePast Konsolenausgabe: Bestrafung eines Nichtzahlers (Quartalsrate).....	46
Abbildung 43: EmilS Log-Datei: Bestrafung eines Nichtzahlers (Quartalsrate).....	46
Abbildung 44: RePast Konsolenausgabe: Bestrafung der Gruppe bei unvollständiger Zahlung an die Bank.....	47
Abbildung 45: EmilS Log-Datei: Bestrafung der Gruppe bei unvollständiger Zahlung an die Bank (Fragment).....	47
Abbildung 46: Bestrafungsstärken bei Nichtzahlen der Quartalsrate; Simulationslauf 1.....	48
Abbildung 47: Bestrafungsstärken bei Nichtzahlen der Quartalsrate; Simulationsläufe 2+3.....	48
Abbildung 48: Simulationslauf 2.....	49
Abbildung 49: Simulationslauf 3.....	50
Abbildung 50: RePast Konsolenausgabe: Sanktion durch die Bank für alle Gruppenmitglieder, Sanktionsziel durch EmilS gewählt (Beispiel 1).....	51
Abbildung 51: EmilS Log-Datei: Sanktion durch die Bank für alle Gruppenmitglieder, Sanktionsziel durch EmilS gewählt (Beispiel 1).....	51
Abbildung 52: RePast Konsolenausgabe: Sanktion durch die Bank für alle Gruppenmitglieder, Sanktionsziel durch EmilS gewählt (Beispiel 2).....	51
Abbildung 53: EmilS Log-Datei: Sanktion durch die Bank für alle Gruppenmitglieder, Sanktionsziel durch EmilS gewählt (Beispiel 2).....	52

## Abkürzungsverzeichnis

MFI :	Microfinanzinstitut, engl.: Micro Financial Institution
MFD:	Microfinanzdienstleistungen
EMIL:	Emergence In The Loop
IRB:	Initial Rule Base
NGO:	Non Governmental Organisation
XML:	Extended Markup Language
UML:	Unified Modeling Language
GUI:	Graphical User Inteface
RePast:	Recursive Porous Agent Simulation Toolkit
MEME:	Model Exploration Module





# 1 Einleitung

## 1.1 Problemstellung

Normen entstehen in jeder Gemeinschaft. Sie sind in der Lage, abseits von Gesetzbüchern und Vertragswerken, für ein rücksichtsvolles Zusammenleben zu sorgen. Dabei haben sie die Fähigkeit zur Anpassung und sind das Durchgangsprodukt eines sich ewig erneuernden Widerstreits zwischen erfahrungsbasierten Erwartungen, die das Handeln bestimmen und Rückmeldungen der Außenwelt, die zu neuen Erfahrungen werden. Gesetzestexte dagegen, die ursprünglich den Versuch darstellten, solche Normen mit der weitesten Verbreitung, festzuhalten und somit das Leben des Einzelnen zu vereinfachen, besitzen durch ihre Statik eine viel geringere Rückkopplung; werden darüber hinaus, sobald der Urzusammenhang mit Normen weitgehend vergessen ist, zum Spielball von Machtinteressen und entfremden sich so vom Normempfinden des Einzelmenschen. Für Normen gilt das "Panta rhei", welches voraussetzt, daß lebendige Einzelwesen ihre Träger sind. Der Versuch die Norm abzutrennen, bspw. durch Niederschrift muss also immer ein Versuch bleiben, der höchstens angenähert zum Erfolg führt. Nicht zuletzt wegen der Unschärfe die dadurch entsteht, daß obwohl wir von einer Norm sprechen, in jedem Einzelnen dem sie bekannt ist, doch wieder eine Abwandlung der generellen Norm existiert und bei ihm, auch nur auf den Zeitpunkt der Erhebung beschränkt, existiert.

Somit ist die Simulation von Normentwicklung sicherlich eine schwierige Aufgabe, die allerdings bei Erfolg wichtige Aufschlüsse über das Menschenwesen in der Gesellschaft bringen kann. Wenn das Verständnis über Selbstregulierung dadurch verbessert werden kann, besteht die Hoffnung einen Impuls dafür zu geben, Selbstregulierung auch häufiger den Vortritt vor Regulierung zu lassen. Die Aufgabe, Normentwicklung zu simulieren, stellt sich das EMIL Projekt: "The main technological aim of the project is to construct a simulator for exploring and experimenting norm-innovation. The forecasted impact of the project is to contribute to the regulation of e-communities by handing out a simulator for the emergence of new norms in complex social systems."<sup>1</sup> "While the simulator will be designed as a general-purpose tool, some specific study cases will be selected as so to provide the necessary grounding parameters."<sup>2</sup>

Eine agentenbasierte Simulation, die für jeden Agenten über ein komplexes Gedächtnis von bereits gehaltenen Normen verfügt und Lernalgorithmen, die nach Interaktion einen Normenwandel

---

<sup>1</sup> Quelle: [AND.07], Seite 2

<sup>2</sup> Quelle: Projektseite EMIL - Objectives, [PSO.09]

einleiten, sollen das Ziel erreichbar machen.

Die vorliegende Arbeit ist eine der ersten Implementierungen mit EMIL und soll Impulse für eine Verbesserung der Simulationssoftware geben. Das hier vorgestellte Szenario basiert auf dem Verhalten von Mikrofinanzgruppen.

## **1.2 Aufbau der Arbeit**

Im Kapitel 2, nach einer kurzen Einführung in das Thema Mikrofinanz im Allgemeinen und die Forschungen von Pablo Lucas in Mexiko, die dem Modell dieser Arbeit zugrunde liegen, wird das EMIL Projekt noch einmal kurz vorgestellt und im Anschluß das Werkzeug beschrieben, das zur Simulation im EMIL Projekt benötigt wird. Es werden sowohl die EMIL-eigenen Werkzeuge, wie auch externe Software aufgeführt und erklärt.

Im Kapitel 3 geht es um die konkrete Modellierung mit EMIL anhand der Mikrofinanzstudie und die Schwierigkeiten die dabei auftraten. Danach wird das entstandene EMIL Modell implementiert und Simulationsläufe durchgeführt. Das Lernverhalten in der Simulation wird beschrieben.

Das 4. Kapitel fragt nach der Bedeutung der Simulationsergebnisse, sowie welche Veränderungen für weitere Simulationen sinnvoll erscheinen.

## 2 Arbeitsgrundlagen

### 2.1 Micro Finance

#### 2.1.1 Die Grundlagen von Micro Finance

Mikrofinanzierung (engl. Micro Finance) ist ein vor allem in der 2. und 3. Welt verwendetes Werkzeug, mittels welchem kleinen und kleinsten Unternehmern Kapital zugänglich wird. Dabei gehören die Geber solcher Kredite nicht zu einer homogenen Gruppe; Geber und Verteiler können beispielhaft wie folgt sein: "The institutions that provide micro finance and the methods used to deliver microfinance products take a variety of forms—cooperatives, village banks, and lending to solidarity groups or individuals... . As there can be a variety of lending approaches, a range of institutional models are also found for MFIs. These include unregulated NGOs, credit unions or cooperatives (which are often regulated), registered banking institutions (either banks or non-bank financial institutions) and government organizations."<sup>3</sup> Oft werden jedoch alle Geber in der Bezeichnung Mikrofinanzinstitute (MFI) zusammengefasst.

Dabei werden MFI seltener als gewinnorientierte Unternehmungen, vielmehr oft als politische Werkzeuge verstanden, welche zur Armutsbekämpfung eingesetzt werden, indem Kreditnehmer gefunden werden, die mit dem zu Verfügung gestellten Kapital Wertschöpfung betreiben und somit bei sich selbst, wie auch ihrem Umfeld, marktwirtschaftliche Prinzipien einführen.

"There has been an extensive debate, which we do not touch on here, on the financial sustainability of MFIs. We would simply make the point that just because an institution needs a subsidy to cover its costs in itself is not a reason for not supporting the institution. The issue should be, what benefits in terms of income gains for the poor can be achieved with the subsidy and how does the ratio of subsidy to benefits compare with that for other interventions."<sup>4</sup>

Der Erfolg eines solchen Angebots ist nicht immer sicher. So zeigen Studien, daß die Umsetzbarkeit von Mikrofinanzierung stark vom jeweiligen Kulturkreis abhängt, oder anders gesagt für unterschiedliche Kulturkreise verschiedene Ansätze nötig sind:

"Die wirtschaftliche Leistungsbilanz afrikanischer Mikrofinanzinstitute (MFI) bleibt weit hinter der anderer Regionen wie Asien und Lateinamerika zurück. ... Problematisch ist

---

<sup>3</sup> Quelle: ADB Institute Research Paper Series No. 53, Seite 3, [RPS.03]

<sup>4</sup> Quelle: ADB Institute Research Paper Series No. 53, Seite 16, [RPS.03]

zugleich, dass Mikrokredite in Afrika statt zu Investitionen oft zum Konsumausgleich von Haushalten verwendet werden. ... Erwartungen, Armut durch Mikrofinanzierung rasch und direkt zu mindern, scheinen dagegen überzogen."<sup>5</sup>

Auch innerhalb eines erfolgreichen Mikrofinanzmarktes wird das Ziel einer wirtschaftlichen Verbesserung für die Kreditnehmer nicht immer erreicht:

"In fact, it is one of the most interesting generalizations to emerge from the micro finance and poverty literature that the poorest of the chronic poor (the core poor) will borrow essentially for protectional purposes given both the low and irregular nature of their income. This group it is suggested will be too risk averse to borrow for promotional measures (that is investment in the future) and will therefore be only a very limited beneficiary of microfinance schemes (Hulme and Mosley 1996: 132)."<sup>6</sup>

Somit muß der Erfolg von Mikrofinanzdienstleistungen (MFD) stark davon abhängen, daß sie auf den Kulturkreis und sogar auf subkulturelle Verhaltensweisen angepasst sind:

"Amongst practitioners there is widespread acceptance of the view that it is both necessary to diversify the products of micro finance and adapt them to local circumstances. Any simple replication of formulae successful elsewhere is rightly treated with suspicion."<sup>7</sup>

Also scheint es sinnvoll sich bei der Untersuchung von Mikrofinanz auf einen geographisch abgegrenzten Bereich zu beschränken und danach vorsichtig zu prüfen welche Verallgemeinerungen möglich sind.

Pablo Lucas dos Anjos, der die Grundlegenden Erhebungen und Beobachtungen für das folgend implementierte Modell gemacht hat, wählte für seine Studien den Mikrofinanzplatz Mexiko (Chiapas) aus.

### **2.1.2 Die Micro Finance Studien von Pablo Lucas**

Pablo Lucas von der Manchester Metropolitan University, England, führt zur Zeit (Mitte 2009) Untersuchungen über die internen sozialen Normen in mexikanischen Gruppen welche Mikrofinanzkredite in Anspruch nehmen durch.

Die Basisdaten dafür versucht Pablo Lucas durch Befragungen der Teilnehmer zu erheben:

"Aiming to study effects of local social conventions on micro finance groups promoted by the MFI, two types of questionnaires were designed: one to better understand the seasoned experience of credit advisers on their fieldwork tasks, plus another oriented to uncover social behaviour and

---

<sup>5</sup> Quelle: German Institute of Global and Area Studies, Publikation: Focus, Ausgabe 09/2008, Seite 1, [GIG.08]

<sup>6</sup> Quelle: ADB Institute Research Paper Series No. 53, Seite 4, [RPS.03]

<sup>7</sup> Quelle: ADB Institute Research Paper Series No. 53, Seite 16, [RPS.03]

structural composition of groups. All advisors were questioned twice using online forms and, additionally, semi-structured verbal interviews were recorded during our fieldwork visit."<sup>8</sup>

Eine typische Frage des Bogens ist in Abbildung 1 gezeigt.

With who would you participate in a business activity?

Name \_\_\_\_\_

Is the person in your group? Yes \_\_\_ No \_\_\_

This person is your: family member\_\_ neighbour\_\_ friend \_\_\_ business partner \_\_\_

You ask this person: always \_\_\_\_\_ sometimes \_\_\_\_\_

Where this person is living? ( ) in the same area\_\_ ( ) Which another area? \_\_\_\_\_

Abbildung 1: Eine Frage aus Pablo Lucas' Fragebogen an MF-Gruppenmitglieder, Quelle: [PLR.08]

Pablo Lucas konnte mit Hilfe dieser Bögen Aussagen über Grundhaltungen, Ereignisse und spezielle Verfahrensweisen in Mikrofinanzierungsgruppen treffen wie aus der Aufbereitung in Abbildung 2 ersichtlich.

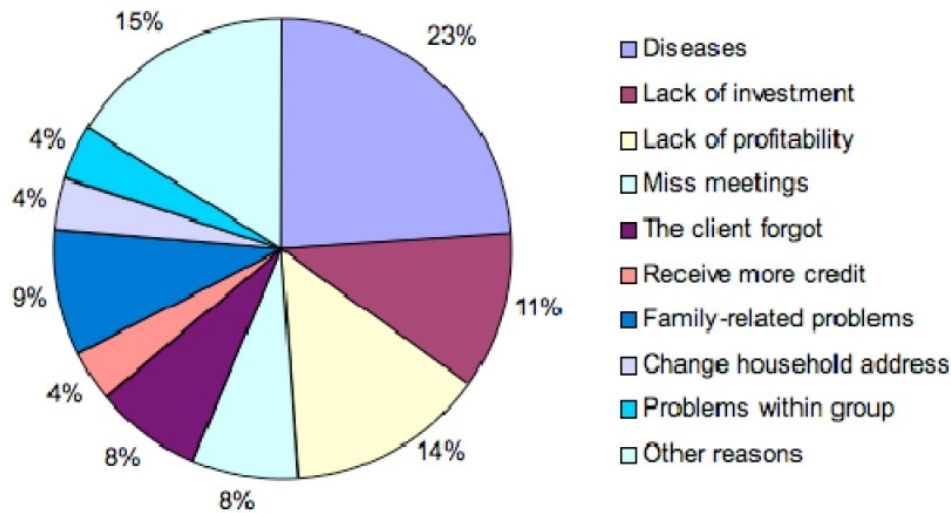


Figure 5. Why clients miss payments?

Abbildung 2: Verhalten in MF-Gruppen, Quelle: Pablo Lucas, [PLR.08]

Daraus ergeben sich Gruppenabläufe die als Basis für ein Modell dienen, Abbildung 3 zeigt solche Zusammenhänge.

<sup>8</sup> Quelle: Pablo Lucas, Dynamics of social conventions in micro finance groups: an experience in chiapas, Mexico, Seite 1,[PLR.08]

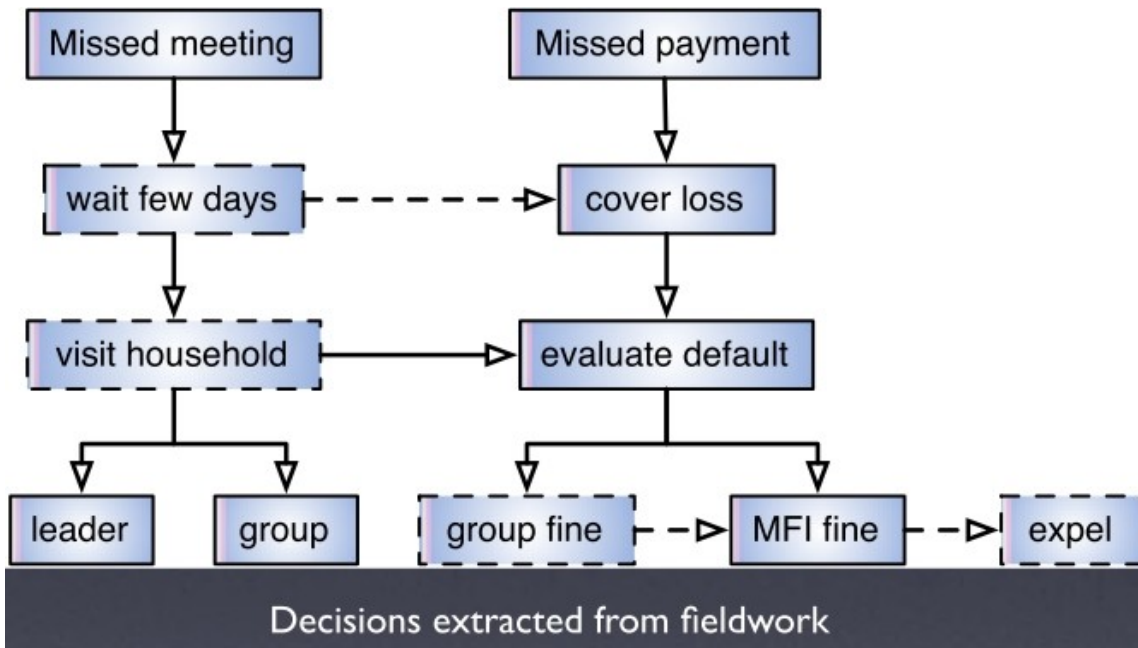


Abbildung 3: Gruppenabläufe in Pablo Lucas Modell, Quelle: Pablo Lucas, [PLC.09]

Wie sich der obigen Grafik entnehmen lässt sind wesentliche Ereignisse im Gruppenleben das Nichterscheinen zu einem Treffen und das Nichtbezahlen einer fälligen Ratenzahlung. Reaktionen darauf können sein, das betreffende Gruppenmitglied zu besuchen, seine Rate mitzubezahlen, ihm eine Strafzahlung aufzuerlegen, ihn gar Auszustoßen aus der Gruppe oder einfach nichts zu tun. Wie eine Gruppe genau funktioniert, wird im Modellierungskapitel 3.1 vorgestellt.

Trotz der Schwierigkeit Simulationsergebnisse auf ihre Aussagekraft überprüfen zu können, möchte Pablo Lucas, wenn möglich auch durch Modellierung dieser Gruppen, Hilfestellung für konkrete Entscheidungen von MFI geben. Dazu arbeitet er, parallel zu dieser Simulation, an eigenen EMIL-unabhängigen Implementationen.

Bei einem Vortrag durch Pablo Lucas an der Universität Koblenz Ende Januar 2009, sowie durch persönliche Gespräche, wurde ein auf diesen Erkenntnissen basierendes vereinfachtes Modell erstellt, welches für eine erste Implementierung mit EMIL-S gedacht ist. Eine Erweiterung des Modells, um Details die es an die Lucas'schen Erhebungen weiter annähern, sollte von da an problemlos möglich sein, was allerdings nicht Gegenstand dieser Arbeit ist.

## 2.2 Das EMIL Projekt

### 2.2.1 Die Bedeutung von Simulationen und die Einordnung von EMIL

Die Hilfestellung die Simulationen der Sozialwissenschaft geben kann ist vielfältig. Verhaltensweisen können simuliert und besser verstanden werden, Vorhersagen können sich daraus ableiten. Erweist sich eine solche Simulation als aussagekräftig, können Ergebnisse sogar an die Stelle menschlicher Entscheidungen und Bewertungen treten. Damit können Folgen abgeschätzt und, als Lernhilfe, die Reaktionen auf eigene menschliche Entscheidungen ohne Risiko erfahrbar gemacht werden. Ob als Unternehmensplanspiel oder Videospiele, der Mensch der die Simulation nutzt leitet Verhaltensregeln aus ihnen ab, wobei dies bei ersterem auch tatsächlich das erklärte Ziel ist.<sup>9</sup>

Grundlage für eine gute Simulation, ist ein gutes Modell, d.h. eine schlanke aber entsprechende, formalisierte Darstellung der gesammelten Beobachtungen. Häufig wird das Modell auf Teilaspekte des Untersuchungsgegenstandes beschränkt, die dem Sozialwissenschaftler als die entscheidenden erscheinen. Vereinfachung macht Modelle verständlich und Simulationen auswertbar, bildet selbstverständlich aber auch eine Fehlerquelle. "A model is a simplification - smaller, less detailed, less complex, or all of these together - of some structure or system"<sup>10</sup>

Da der Formalisierungsprozeß, also die Übersetzung in ein computerlauffähiges Programm, allerdings eine ordnende Tätigkeit ist, kann sie einen Theoriegewinn durch eine neue Sicht auf den Untersuchungsgegenstand bedeuten. "The Programm will be much more precise than the textual form of the procedure and is therefore helpful in refining one's theory. Simulation can thus be used as a *method of theory development*."<sup>11</sup>

Einen guten Überblick über den Zusammenhang zwischen Untersuchungsgegenstand, Modellierung, empirischen Studien und Simulation gibt Abbildung 4.

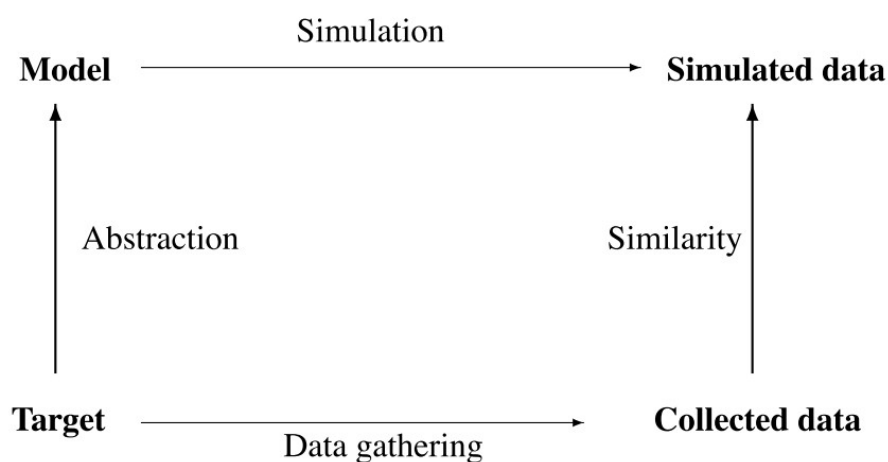


Abbildung 4: "The logic of simulation as a method", Quelle: [SSS.05], Seite 17

<sup>9</sup>Quelle: *Simulation for the Social Scientist*, S.4+5, [SSS.05]

<sup>10</sup>Quelle: *Simulation for the Social Scientist*, S. 2, [SSS.05]

<sup>11</sup>Quelle: *Simulation for the Social Scientist*, S.3, [SSS.05]

Darüber hinaus hat die Geschichte der Simulation verschiedenste Simulationstechniken hervorgebracht. Dabei bestimmt der Untersuchungsgegenstand welche Simulationstechnik die jeweils geeignetste ist und jede Technik erfordert eine abweichende Form der Modellierung. "A technique capable of modelling two or more levels is required to investigate emergent phenomena. Some techniques allow the modelling of communication (for example, the passing of messages) between agents and so are appropriate for modelling language and interaction; others do not."<sup>12</sup>

Welche Anforderungen stellt nun EMIL, "EMergence In the Loop", an die zugrunde liegende Technik? Normen die bei der Interaktion von Agenten entstehen (Emergence), sollen wieder einfließen in neue Agentenentscheidungen (Immergence). Die EMIL Projektseite fasst die Ziele wie folgt zusammen<sup>13</sup>:

- understand and manage complexity in social systems with autonomous agents
- understand how new conventions and norms emerge and spread in these systems
- study of norm innovation by means of agent-based simulation

Nimmt man Abbildung 5 zu Hilfe können sofort verschiedene Techniken ausgeschlossen werden. Zum Ersten, da für die Simulation von "emergent phenomena" mindestens zwei Ebenen, nämlich neben der für Einzelwesen eine weitere für die Gesellschaft benötigt wird. Zum Zweiten, da beliebig komplexe Einzelagenten benötigt werden. Zum Dritten, da sich Agenten untereinander austauschen können sollen.

	<i>Number of levels</i>	<i>Communication between agents</i>	<i>Complexity of agents</i>	<i>Number of agents</i>
System dynamics	1	No	Low	1
Microsimulation	2	No	High	Many
Queuing models	1	No	Low	Many
Multilevel simulation	2+	Maybe	Low	Many
Cellular automata	2	Yes	Low	Many
Multi-agent models	2+	Yes	High	Few
Learning models	2+	Maybe	High	Many

Abbildung 5: "A comparison of social science simulation techniques", Quelle: [SSS.05], Seite 13

Nach Vergleich mit der erwähnten Graphik, bleiben Multi-Agenten-Modelle und Learning-Modelle in der Auswahl. Da das EMIL-Konzept das Lernen, wie im Verlauf der Arbeit gezeigt, in eine eigene Komponente namens EMIL-S ausgliedert, muss der Modellierer einer solchen Simulation ein Multi-Agenten-Modell und seine Umgebung schaffen. "In all multi-agent simulations, the

<sup>12</sup> Quelle: *Simulation for the Social Scientist*, S.13, [SSS.05]

<sup>13</sup> Quelle: *Simulating the two way dynamics of norm innovation*, Seite 2, [AND.07]



agents are located in an environment. What constitutes an environment depends on what is being modelled, but if the agents are individual people, rather than organizations, one of the main functions of the environment will be to provide a spatial context."<sup>14</sup> Da für das Modell dieser Arbeit keine physische Bewegung im Gelände, wie sie bspw. ein Zellularer Automat bietet, für Umgebungsänderungen verantwortlich ist, sondern ausschließlich durch Variablenänderungen zustande kommt, kann die Modellumgebung und ihre Schnittstelle zu EMIL-S von Grund auf in JAVA programmiert werden.

### 2.2.2 Der EMIL Ansatz

Wie bereits in der Einleitung erwähnt, ist es das Ziel des Projekts, mehr über die Entstehung von Normen zu erfahren. Normen sind sehr anpassungsfähig. Auch wenn die Entstehung von rechtlichen Normen bereits gut untersucht wurde, ist es noch unklar wie gesellschaftliche Normen entstehen. (vgl.[CAM.08])

Die Antwort des EMIL Projekts auf diese Frage lautet, daß sich Normen auf die Normvorstellungen (normative beliefs) der Einzelwesen gründen, die wiederum von globalen Normen induziert werden. Das Wiedereinfließen von Normen und die entstehende Wechselwirkung ist der Ansatz des Projekts. "Whereas the emergence of conventions has received considerable attention, the way-down in the circuit of emergence has been accounted for, if only, in purely behavioural terms. Hence, the process of immergence has being generally overlooked, and with it, the emergence of norms as something in connection with, but distinct from, mere conventions and coercion."<sup>15</sup>

Im Gegensatz zu Konventionen, die einen mehr statischen Charakter haben, unterliegen Normen einem dynamischen Innovationsprozeß. "As the core of the matter is the way norms not only emerge, but also are innovated, it is clear that norms cannot be considered as conventions, i.e., mere behavioural regularities cannot be innovated."<sup>16</sup>

Die Überzeugung des Agenten einer Norm gefolgt zu sein, wird dieses Verhalten für ihn zukünftig attraktiver machen, und ihn auch dazu bringen dieses Verhalten anderen Agenten als "richtig" zu vermitteln. Die Agenten tun dies durch Aufforderungen (Deontic) und Bewertungen (explizit normative valuations).<sup>17</sup>

---

<sup>14</sup> Quelle: *Simulation for the Social Scientist*, S.184, [SSS.05]

<sup>15</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 10

<sup>16</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 2

<sup>17</sup> Quelle: *Normal = Normative? The Role of Intelligent Agents in Norm Innovation*, [CAM.08]

Die Arbeit des EMIL Projekts wurde dabei in mehrere Unteraufgaben zerlegt, die im Kapitel 2.2.3 vorgestellt werden. Dazu gehören EMIL-A und die Folgenden: "EMIL is articulated in three main building tasks, EMIL M, EMIL S, EMIL T, and in two testing tasks aimed at collecting real data and simulation findings[...]"<sup>18</sup>

## 2.2.3 Die Arbeitsschritte des EMIL Projekts

### 2.2.3.1 EMIL-M

Die Ideen und Ziele die das EMIL Projekt ausmachen werden genau in EMIL-M festgehalten. Was sind die Unterschiede zwischen Normen und Konventionen? Wie hängen beide zusammen? Was sind die optimalen Bedingungen für das Ausbilden von Normen? Wie und wann entstehen Normen?<sup>19</sup>

Um diese Fragen zu beantworten benötigt das Projekt eine klar abgegrenzte Fachsprache und wissenschaftlich eng definierte Probleme. Beides liefert EMIL-M.<sup>20</sup>

EMIL-M definiert so bspw. welche Abläufe zwischen Agenten möglich und nötig sind sowie welche Fähigkeiten ein Agent selbst besitzen muss. Hauptschwerpunkte sind Emmergence, Immergence, Diffusion und Interaction bei der Normentstehung.<sup>21</sup>

Die wissenschaftlichen Fragen die EMIL-M stellt, sollen durch EMIL-S und EMIL-T beantwortet werden.

### 2.2.3.2 EMIL-A

EMIL-A bietet die theoretische Architektur, die einer späteren Simulationssoftware zugrunde liegt und setzt dabei die Kapitel 2.2.2 genannten Ziele um. Zur Erreichung des Ziels wird das Agentenverhalten in verschiedene Entscheidungsphasen zerlegt und die Variablen definiert die auf das Agentenverhalten Einfluß nehmen. "Our normative architecture (EMIL-A) consists of mechanisms and mental representations allowing norms to affect the behaviors of autonomous intelligent agents. EMIL-A is meant to show that norms not only regulate the behavior but also act on different aspects of the mind: recognition, adoption, planning, and decision-making."<sup>22</sup>

<sup>18</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 13

<sup>19</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 13

<sup>20</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 13+14

<sup>21</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 14+15

<sup>22</sup> Quelle: *Normal = Normative? The Role of Intelligent Agents in Norm Innovation*, [CAM.08]

So sollen Agenten die Fähigkeit erhalten Normen selbsttätig in einer anfänglichen Entscheidungsphase zu erkennen. Der Einzelagent erhält vier Module, die ihm die Fähigkeiten geben Normen zu erkennen, sie für sich zu übernehmen, eigene Absichten nach Eintritt von konkreten Ereignisse zu formulieren und Mittel zur Förderungen seiner Absichten auszuwählen.<sup>23</sup>

Vier Schritte werden bei jedem Agenten durchlaufen:

- Norm Recognition
- Norm Adoption
- Decision Making
- Normative Action Planning

Die Einflüsse die zu einem Entschluß führen, werden in ein komplexes System von Variablen zerlegt.<sup>24</sup>

Das Gedächtnis des Agenten im Zusammenhang mit Normempfinden umfasst folgende Sammlungen von Einträgen.

- Normative Beliefs (N-Beliefs)
- Normative Goals (N-Goals)
- Normative Intentions (N-Intentions)
- Meta-norms
- Normative Board

Der Ablauf im Normempfinden des Einzelagenten wird in Abbildung 6 übersichtlich dargestellt.

---

<sup>23</sup> Quelle: EMIL-S Projekt , Deliverable 1.2, Seite 7, [D12.07]

<sup>24</sup> Quelle: EMIL-S Projekt , Deliverable 1.2, Seite 7, [D12.07]

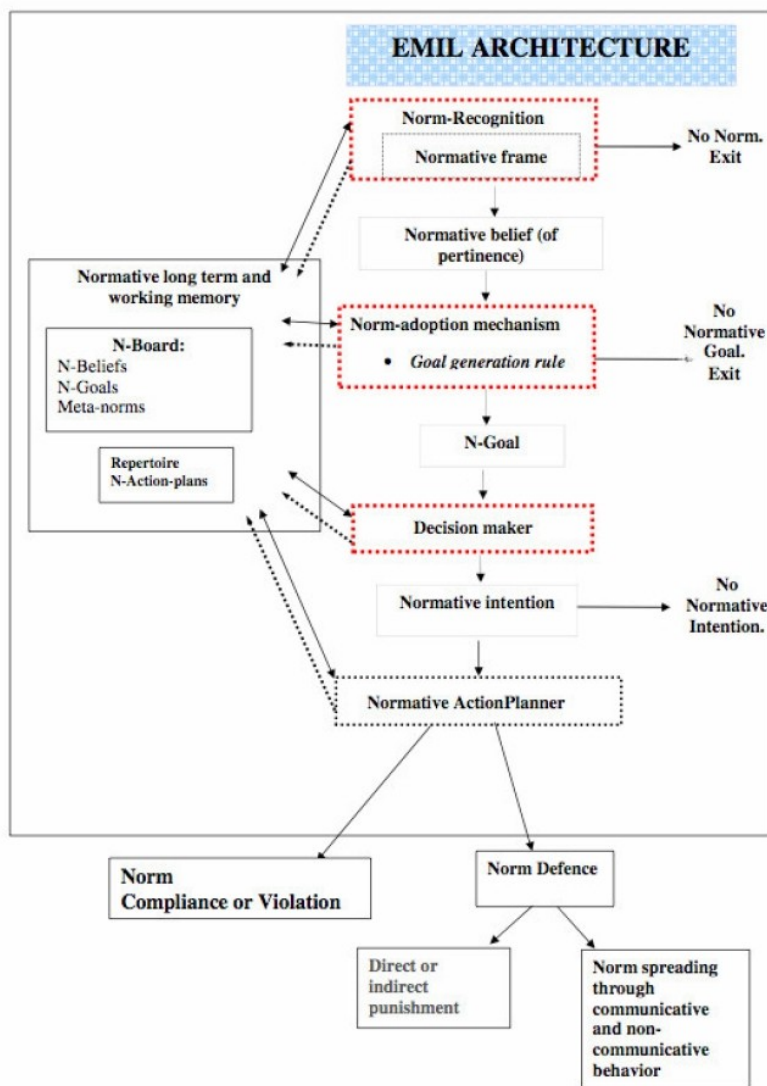


Abbildung 6: Quelle: EMIL-S Projekt , Deliverable 1.2, Seite 7, [D12.07]

### 2.2.3.3 EMIL-S

EMIL-S setzt EMIL-M und EMIL-A zu einer Simulationssoftware um. "A simulation platform for experiments, EMIL-S, will be built up with the purpose of testing EMIL-M. EMIL-S will be applied to execute simulations and compare results with available data[...]"<sup>25</sup>

Das umfassende Gedächtnis der Einzelagenten über alle Vorkommnisse, wie in den voran gegangenen Kapiteln beschrieben, sowie globale Variablen (Gesellschaftsebene) müssen implementiert werden. "Agents will be able to keep memories of encounters with other agents and will also generate (incomplete) models of other agents and of their environments.

<sup>25</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 19

The simulator will provide these technical characteristics of agents and their relations, as well as a protocol for the communication among agents;"<sup>26</sup>

Die Umsetzung erfolgte in vier Phasen die im Juli 2009 entweder abgeschlossen oder nahezu abgeschlossen sind:

- Die erste Aufgabe war die Anforderungen für eine solche Software zu ermitteln, sowie welche Simulationswerkzeuge für den Bau der Modellumgebung zur Verfügung stehen.
- Die zweite Aufgabe bestand in der eigentlichen Implementation von EMIL-S sowie in der Koordination mit der Steuerungs- und Auswertungssoftware MEME, die von den ungarischen Kollegen des Projekts geliefert wird. Außerdem wird ein Hilfswerkzeug bereitgestellt, welches den Bau von EMIL-S Modellen erleichtert
- Im dritten Schritt wird die Software alle notwendigen Schnittstellen erhalten um plattformunabhängig eingesetzt werden zu können.
- Als vierte Aufgabe soll die Entwicklungsgeschichte nochmals genau dokumentiert, sowie die üblichen Verfahren zur Qualitätssicherung durchgeführt werden.(vgl.[AND.07])

Mit der fertigen Simulationssoftware kann nun mit einer Vielzahl von Durchläufen und idealerweise auch Modellen erforscht werden, welche Möglichkeiten der gewählte Ansatz hat.<sup>27</sup>

Da die erforderliche Rechenleistung für solch komplexe Agenten, die über lange Erinnerung und mehrere Entscheidungsfindungsphasen verfügen, wobei immer eine Vielzahl Vergleiche nötig werden, bei zunehmender Simulationsdauer sehr hoch werden kann, wurde im Projekt über die Notwendigkeit einer verteilten Ausführung nachgedacht.

#### 2.2.3.4 EMIL-T

Bei EMIL-T handelt es sich um die Aufgabe die zugrunde liegende Theorie, nach den Erkenntnissen welche die Simulation gebracht haben, zu überarbeiten und zu verbessern. Dazu sind folgende Schritte vorgesehen:

- Vergleich der Simulationsergebnisse mit eventuell vorhandenen empirischen Daten
- Erstellung einer überarbeiteten Theorie zur Norminnovation und Abschätzung der Anwendbarkeit für die Sozialwissenschaft. Ein neues EMIL-T Modell wird erstellt

---

<sup>26</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 19

<sup>27</sup> Quelle: *Emergence In the Loop: Simulating the two way dynamics of norm innovation*, [AND.07], Seite 21

- Die bisher erstellten Modelle sollen mit der überarbeiteten EMIL-T Version neu implementiert werden. Die Unterschiede in der Simulation von verschiedenen EMIL-T Versionen wird ausgewertet. Daraus soll auf die Aussagekraft der Ergebnisse geschlossen werden können. (vgl.[AND.07], Seite 25)

## 2.2.4 Teilnehmer des Projekts

Die verschiedenen Aufgaben dieses von der EU für drei Jahre geförderten Projekts wurden auf mehrere Projektgruppen in ganz Europa verteilt.

Außer der Universität Koblenz-Landau sind die weiteren Projektteilnehmer<sup>28</sup>:

- Institute of Cognitive Science and Technology, National Research Council CNR-ISTC Italy
- University of Bayreuth, Dept. of Philosophy UBT Germany
- University of Surrey, Centre for Research on Social Simulation UNIS United Kingdom
- Manchester Metropolitan University, Centre for Policy Modelling MMU United Kingdom
- AITIA International Informatics Inc. AITIA Hungary

Die in Kapitel 2.2.3 besprochenen Teilaufgaben werden nicht von allen Projektteilnehmern gleich stark bearbeitet, vielmehr sind die Aufgaben verteilt, wobei die Koordination wichtig ist. So wird die Entwicklung von EMIL-S von der Universität Koblenz-Landau ausgeführt. EMIL-M wurde zum Großteil bei den italienischen Kollegen erdacht. Das Unternehmen AITIA (Ungarn) liefert die Software MEME, die zur Auswertung und Steuerung der EMIL-S Komponente eingesetzt werden soll.

## 2.2.5 Die EMIL Simulation Suite

Mittels der entwickelten Softwarekomponente EMIL-S können Agenten, wie folgend dargestellt, normenbasiert handeln. EMIL-S ist als Software allerdings aus zwei Gründen nicht eigenständig nutzbar.

Erstens, da Ereignisse welche erst Bewertungen durch EMIL-S hervorrufen, nicht selbsttätig von EMIL-S erzeugt werden können. Vielmehr benötigen wir dazu eine Simulationsumgebung (z. B.

---

<sup>28</sup> Quelle: EMIL Projektseite Startseite, [EPS.09]

RePast, siehe Kapitel 2.3), die Entscheidungen von EMIL-S Agenten verarbeiten kann und daraus, in Kombination mit externen Modellparametern, neue Ereignisse generiert.

Zweitens ist EMIL-S nicht auf ansprechende Weise in der Lage dem Nutzer über die während der Simulation stattfindenden Veränderungen Auskunft zu geben. Dazu ist eine Steuerungs- und Auswertungssoftware (MEME) als Komponente wünschenswert.

So bildet idealerweise EMIL-S mit diesen zwei zusätzlichen Komponenten eine funktionsfähige Einheit. Dabei ist die MEME-Komponente nicht zwingend für die Modellimplementation notwendig, sondern ein hilfreiches Werkzeug für die wissenschaftliche Nutzung der entstandenen Simulation.<sup>29</sup> Abbildung 7 zeigt die Simulation Suite im Überblick.

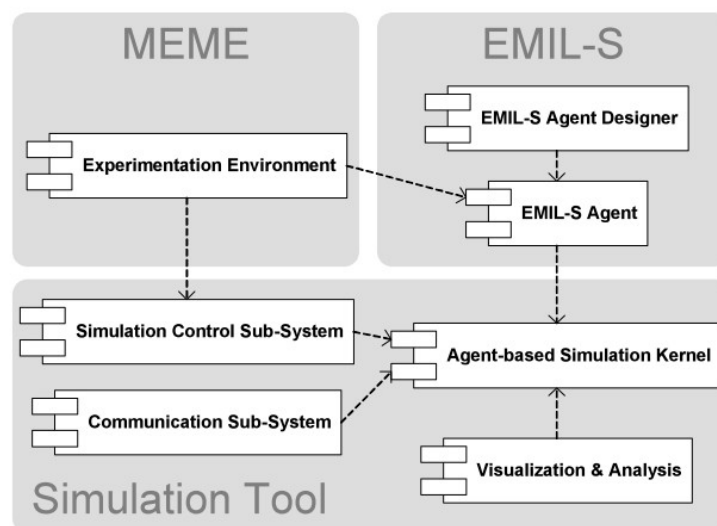


Abbildung 7: Die Emil Simulation Suite, Quelle: [SNB.09]

## 2.2.6 Der Aufbau von EMIL-S

EMIL-S ist, wie bereits erwähnt, der Kern der Simulation in dem Normentstehung agentenbasiert stattfindet. Wie werden nun Agenten- entscheidungen und kommunikation, Normen und Normentstehung in EMIL-S abgebildet? Die Basis für Agentenentscheidungen in EMIL-S bildet ein Regelsatz der sich in Entscheidungsbäumen (Event-Action-Tree) abbilden läßt. Ein solcher Regelsatz wird zu Simulationsbeginn als "Initial Rule Base" (IRB) den jeweiligen Agententypen mitgeliefert. Wie der Name schon sagt ist dieser Regelsatz nicht unveränderlich sondern der

<sup>29</sup> Quelle: Möhring, Michael; Lotzmann, Ulf (2009): *Simulating Normative Behaviour and Norm Formation Processes*, [SNB.09]

Ausgangspunkt für neue oder veränderte Regeln, die wiederum Teil des Regelsatzes werden. Eine EMIL-S Regel besitzt die Form von Abbildung 8 (Eine semantisch ausführlichere Darstellung von EMIL-S Bäumen ist mit der Markupsprache XML möglich, auf deren Vorteile später eingegangen wird).

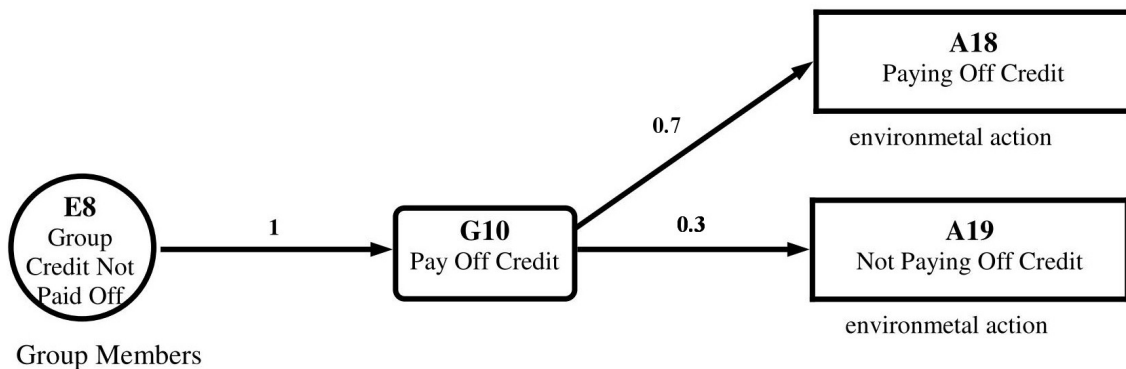


Abbildung 8: EMIL-S Baum-Modell

Ein solcher Baum besitzt drei Glieder: Ereignisse, Aktionsgruppen und Aktionen.

- Ereignisse: Sie werden von der Simulationsumgebung aufgerufen.

Das weitere Durchlaufen des Entscheidungsbaumes, von dem das Ereignis die Wurzel darstellt, führt zu Agentenentscheidungen, also zu einer Reaktion:

- Aktionsgruppen: Aktionsgruppen werden Aktionen zusammengefasst die sich gegenseitig ausschließen. Sie werden je nach ihrer Wahrscheinlichkeit, die an der eingehenden Kante ablesbar ist, im Entscheidungsbaum aufgerufen. Dabei kann die Wahrscheinlichkeit für einen Aufruf der Gruppe beliebige Werte annehmen, also auch Null und Eins.
- Aktionen: Sie stellen die Agentenentscheidung dar und können, wie Ereignisse, entweder von der Art "Environmental" oder "Norm-Invocation" sein. Dabei werden Aktionen vom ersten Typ an die Simulationsumgebung ausgegeben um den Einfluß auf die Umgebung zu berechnen und sind vom Modal B (Behaviour). Dagegen sind "Norm Invocations" Aktionen die innerhalb von EMIL-S zu Neubewertungen führen. Die Modale sind ebenfalls V (Valuation), D (Deontic)



oder S (Sanction).<sup>30</sup>

Ein weiteres EMIL Konzept ist die Verwendung von Rollen, die jeder Regel zugeordnet werden können. Dabei können Regeln von Agenten entweder als "Actor", d.h. unmittelbar Handelnder durchlaufen werden, oder als "Observer", der keinen direkten Einfluß auf die Umgebung ausüben kann, sondern für sich Bewertungen (Norm-Invocations) aufgrund der Beobachtung vornimmt.

Mit diesen Regelsätzen und ihrem Aufruf alleine wäre allerdings kein Prozeß der Normentstehung zu beobachten, wenn nicht zusätzlich EMIL-S auch die Funktionalität zur Speicherung und internen Auswertung von Aktionen und Reaktionen über einen längeren Zeitraum besitzen würde.

Diese Anforderung ist so umgesetzt, daß die Nachricht (Event Message) eines stattgefundenen Ereignisses, mit dem folgend ausgeführten Baum (Action Tree), dem Agentenzustand (Environmental State) und einer Zeitangabe (Time Stamp), zusammen als ein Ereignis in einer Ereigniskartei (Event Board) abgelegt wird.

Aus chronologisch geordneten Abschnitten dieses Event Boards werden zusätzlich Schlüsseleinträge (Classifier) errechnet und in das Event Board mitaufgenommen. Diese Classifier ermöglichen EMIL-S im Verlauf der Simulation einen Vergleich von verschiedenen Abschnitten des Event Boards zur Normgenerierung.<sup>31</sup> Abbildung 9 gibt einen Überblick über das Event Board.

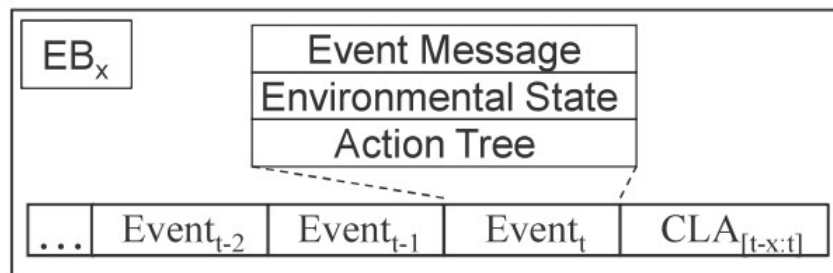


Abbildung 9: Event Board, Quelle: [SNB.09]

Aus diesem Gedächtnis kann EMIL-S nun, durch Vergleich, Normen ableiten. Die entdeckten Normen werden nun ebenfalls in einer Kartei als neue Regel (Merged Action Tree) abgelegt, dem "Normative Frame". Um die entstandenen Regeln wiederum vergleichbar zu machen wird der Classifier des zugehörigen Event Board Abschnitts mitsamt seinen Ereigniseinträgen übernommen. Komplettiert wird der Eintrag im Normative Frame durch eine "Valuation History" die Buch führt, wie oft diese Regel bereits als richtig erkannt worden ist. Dabei ist das Normative Frame

<sup>30</sup> Quelle: EMIL-S Projekt, Deliverable 3.3, Seiten 21-20, [D33.08]

<sup>31</sup> Quelle: Möhring, Michael; Lotzmann, Ulf (2009): *Simulating Normative Behaviour and Norm Formation Processes*, [SNB.09]

(Abbildung 10) ein individueller Agentenregelsatz für Regeln die Einzelagenten als Norm betrachten. Populäre Individualregeln können allerdings in das "Normative Board" aufsteigen und bilden dann tatsächlich eine Art gesellschaftlicher Norm.<sup>32</sup>

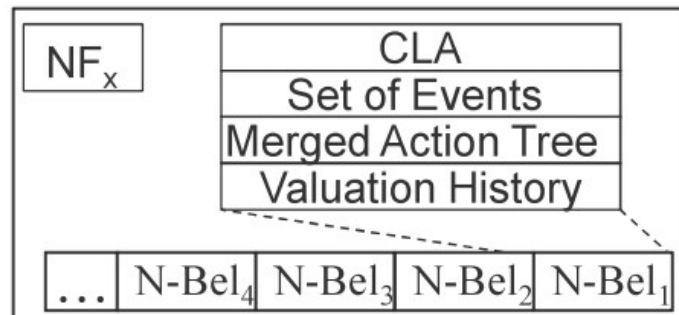


Abbildung 10: Normative Frame, Quelle: [SNB.09]

Das UML Aktivitätsdiagramm in Abbildung 11 zeigt den Vergleichsprozess von aktuellem Ereignis mit zugehörigem Event Board Classifier und vorhandenem Regelsatz. Wird eine Regel die zu dem Classifier passt gefunden, kann der Agent entscheiden die gefundene Regel anzuwenden. Abbildung 11 ist ein an dieser Stelle nur exemplarisch aufgeführter Teil des EMIL-A Prozesses.

<sup>32</sup>Quelle: Möhring, Michael; Lotzmann, Ulf (2009): *Simulating Normative Behaviour and Norm Formation Processes*, [SNB.09]

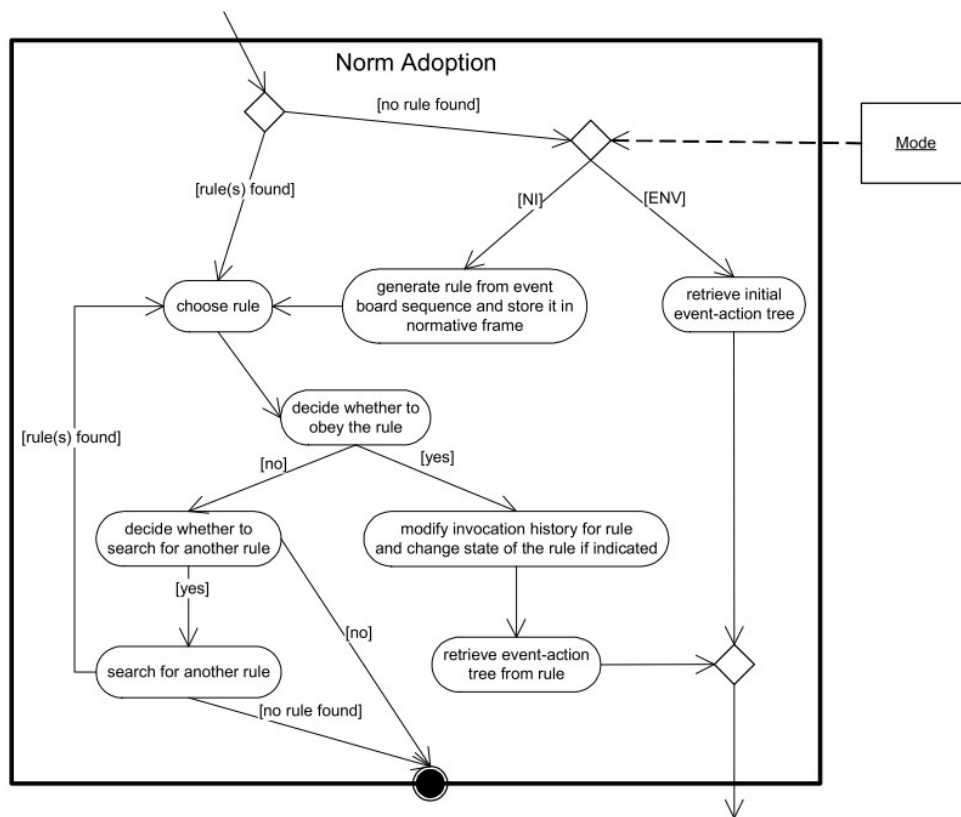


Abbildung 11: Norm Adoption, Quelle: [SNF.08]

## 2.2.7 Das EMIL-S GUI

Ereignisse, Aktionen und Regeln welche die IRB beschreiben, werden in XML (Extended Markup Language) notiert. Somit ist diese Sprache die Schnittstelle zwischen Modellierung und EMIL-S. Da man den zukünftigen EMIL-S Nutzern das Schreiben von XML-Code ersparen wollte, programmierte Ulf Lotzmann, der auch für die Implementierung von EMIL-S verantwortlich war, ein Hilfswerkzeug mit GUI, welches den Modellbau anhand einer Baumstruktur, ähnlich Abbildung 5, ermöglicht. An weiterer Funktionalität der Software wird derzeit gearbeitet.

Wer also bereits über ein Baummodell seiner IRB verfügt, wird mit der EMIL-S GUI wie folgt vorgehen:

1. Definition von Ereignissen und Aktionen. Dazu stehen im Reiter "Global Definitions" (Abbildung 12) entsprechende Frames zur Verfügung, in denen lediglich Name, Beschreibung und Art eingetragen werden müssen.

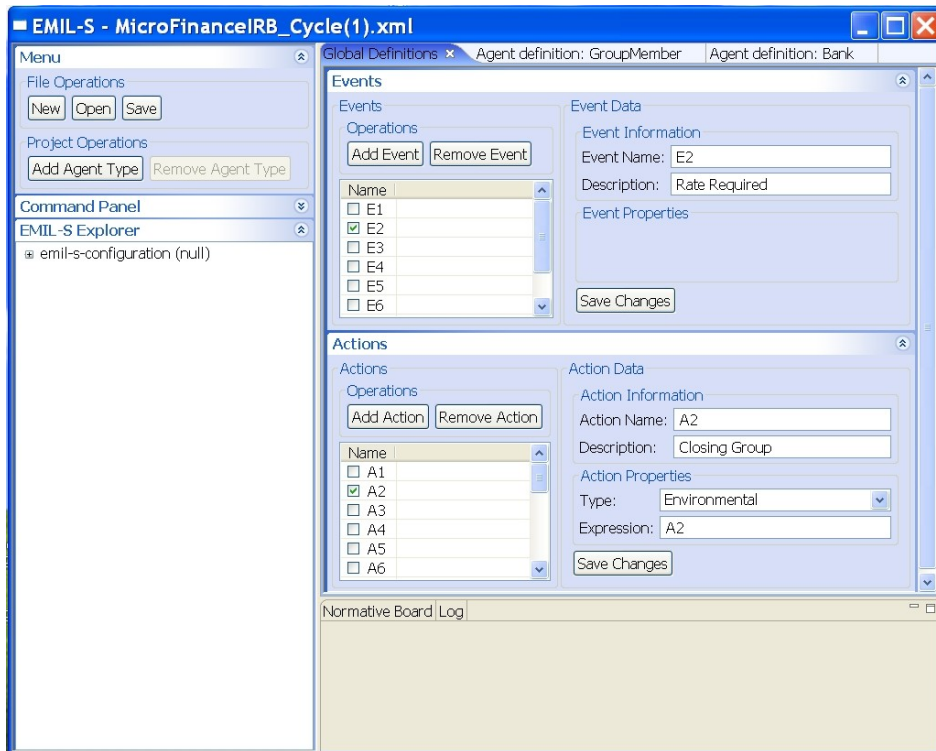


Abbildung 12: EMIL-S GUI (1/2)

2. Aus den so geschaffenen Elementen werden für jeden Agententyp im Reiter Agentendefinitionen Regelbäume zusammengestellt. Dabei können die Aktionen die auf ein Ereignis folgen beliebigen Aktionsgruppen zugeordnet werden (Abbildung 13).

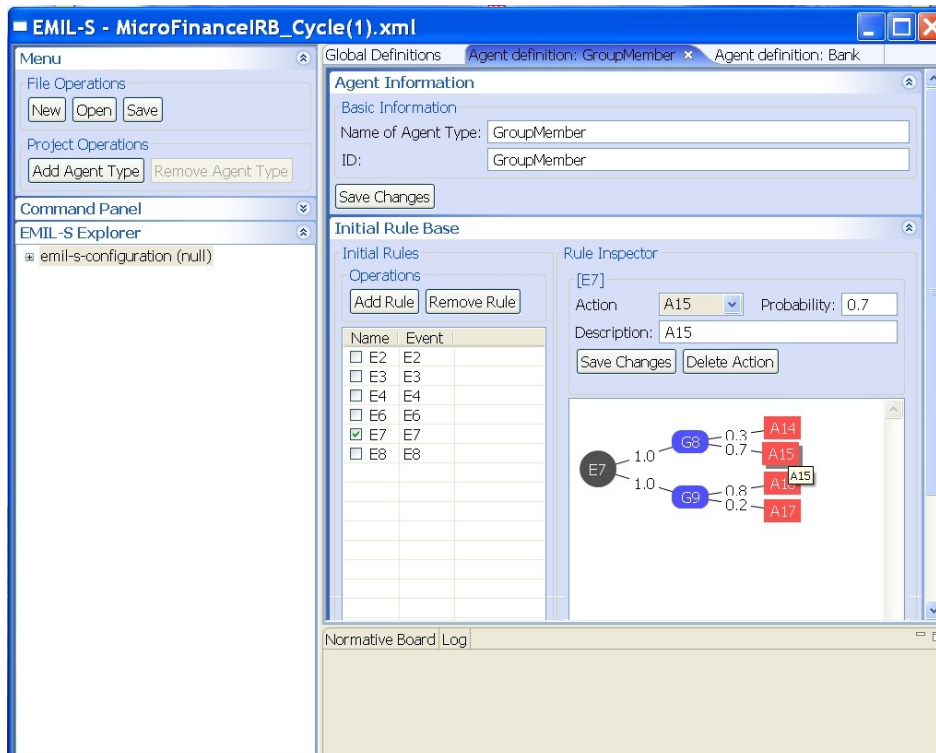


Abbildung 13: EMIL-S GUI (2/2)

Falls ein XML Modell ohne EMIL-S GUI erstellt wurde, steht noch ein weiteres Werkzeug zur Verfügung mit dem die EMIL-S Antworten auf ausgelöste Ereignisse getestet werden können. Ist also eine XML IRB korrekt konfiguriert liefert das Werkzeug aus Abbildung 14 sinnvolle Rückgabewerte in Form von Aktionen. Die Reihenfolge bei der Bedienung ist wie folgt:

1. Laden der zu testenden Konfiguration
2. Auswahl eines Agententypen
3. Auswahl eines Ereignisses und Senden des Ereignisses

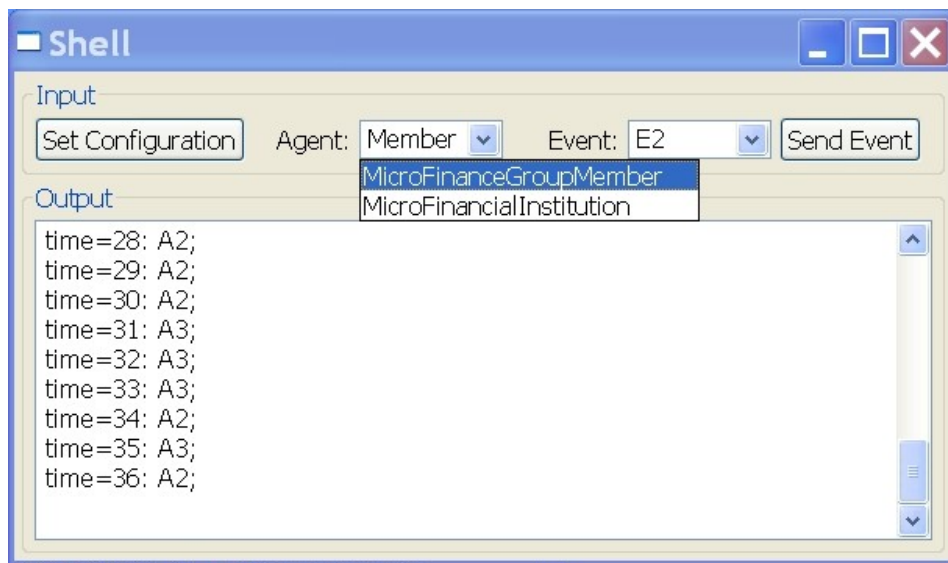


Abbildung 14: EMIL-S XML-Tester

## 2.2.8 XML als Modellierungssprache für EMIL-S

Die durch Eingabe des Baummodells in die EMIL-S GUI erhaltene XML IRB wird folgend abgebildet, um ihre Gliederung zu zeigen. Dazu wird vorgehend das fertige Micro Finance Modell verwendet. Wie sofort erkennbar, finden sich die Elemente die wir im Reiter Global Definitions erstellt haben, also zuerst alle möglichen Ereignisse, dann alle Aktionen, ganz zu Beginn des XML Dokumentes, eingerahmt von den Tags für Global Definitions (siehe Abbildung 15).

```

<emil-s-configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema
  <global-definitions>
    <rule-elements>
      <event name="E1" description="New Cycle"/>
      <event name="E2" description="Rate Required"/>
      <event name="E3" description="Group Fine?"/>
      <event name="E4" description="Cover Payment?"/>
      <event name="E5" description="Payment Incomplete"/>
      <event name="E6" description="Group Fine Required"/>
      <event name="E7" description="Group Fine Not Paid"/>
      <event name="E8" description="Group Credit Not Paid Off"/>

```

Abbildung 15: Micro Finance XML Regelsatz: Ereignisdefinitionen

Die Definition von Aktionen mit Angabe des Typs erfolgt wie in Abbildung 16.

```

<action name="A1" description="Approving Demand">
  <type>Environmental</type>
  <expression>A1</expression>
</action>
<action name="A2" description="Closing Group">
  <type>Environmental</type>
  <expression>A2</expression>
</action>
<action name="A3" description="Sending Warning">
  <type>Environmental</type>
  <expression>A3</expression>
</action>
<action name="A4" description="Sending Approval">
  <type>Environmental</type>
  <expression>A4</expression>
</action>

```

Abbildung 16: Micro Finance XML Regelsatz: Aktionsdefinitionen

Auf die "global-definitions" folgen die "agent-definitions", die im Modell der Arbeit geteilt sind in zwei verschiedene Agententypen. Innerhalb dieser Typen findet sich die zugehörigen IRBs mit allen Regeln wieder. Abbildung 17 zeigt den Baum zu Ereignis E2 des Agenten Group Member.

```

</rule-elements>
<observation-parameters mode="ALL"/>
</global-definitions>
<agent-definitions>
  <agent name="GroupMember" id="GroupMember">
    <initial-rule-base>
      <event-action-tree name="E2" description="E2">
        <event name="E2"/>
        <action-group name="G3" probability="1.0" description="G3">
          <action name="A5" probability="0.7" description="A5"/>
          <action name="A6" probability="0.3" description="A6"/>
        </action-group>
      </event-action-tree>

```

Abbildung 17: Micro Finance XML Regelsatz: Regeldefinitionen

Am Ende des XML Dokumentes (Abbildung 18) befindet sich der Platz für das Normative Board, welches zu Beginn der Simulation noch keine Einträge enthält.

```
</agent>
<agent name="Bank" id="Bank">
  <initial-rule-base>
    <event-action-tree name="E1" description="E1">
      <event name="E1"/>
      <action-group name="G1" probability="1.0" description="G1">
        <action name="A1" probability="1.0" description="A1"/>
        <action name="A2" probability="0.0" description="A2"/>
      </action-group>
      <action-group name="G2" probability="1.0" description="G2">
        <action name="A3" probability="0.5" description="A3"/>
        <action name="A4" probability="0.5" description="A4"/>
      </action-group>
    </event-action-tree>
    <event-action-tree name="E5" description="E5">
      <event name="E5"/>
      <action-group name="G6" probability="1.0" description="G6">
        <action name="A11" probability="1.0" description="A11"/>
      </action-group>
    </event-action-tree>
  </initial-rule-base>
  <eventboard/>
</agent>
</agent-definitions>
<agent-instances/>
<environment-definitions>
  <normative-board/>
</environment-definitions>
</emil-s-configuration>
```

Abbildung 18: Micro Finance XML Regelsatz: Ereignisdefinitionen und Normative Board

Eine solche XML Datei ist hervorragend für Maschinen zu lesen und zu beschreiben. Auswertungen können also durch Auslesen des bei der Simulation veränderten Dokumentes durchgeführt werden.

## 2.3 RePast als Simulationsumgebung

RePast ist ein Open Source Simulationswerkzeug, speziell für agentenbasierte Modelle. Die Programmiersprache ist Java. Obwohl RePast in einer neuen Version, genannt "Symphony" vorliegt, wurde für diese Arbeit RePast 3 verwendet, da die zusätzlichen Funktionalitäten von Symphony ohne Bedeutung für die Lösung der Aufgabe waren. Wie die Schnittstelle zwischen RePast und EMIL-S funktioniert wird in Kapitel 3.2.1 erklärt.

RePast bietet vorgefertigte Agentenrumpfe die für jeden Agententypen durch eine JAVA Klasse repräsentiert werden und vom EMIL-S Modellierer zur Simulationsumgebung ausgebaut werden können.

Viele Funktionen, die im EMIL Projekt von MEME für EMIL-S bereitgestellt werden sollen, sind

von RePast für die RePast Ebene bereits verfügbar. So können Simulationsläufe geplant, parallel ausgeführt, Parameter justiert, Veränderungen von Variablen ausgewertet werden.<sup>33</sup>

---

<sup>33</sup> Quelle: RePast Projektseite, RePast3, [REP.09]



## 3 Durchführung

### 3.1 Modellierung des EMIL-S Micro-Finance Szenarios

#### 3.1.1 Anfangsversion

Das Micro Finance EMIL-S Modell unterscheidet sich von vorher in EMIL-S implementierten Modellen dadurch, daß Individualentscheidungen nicht immer ausreichen um neue Ereignisse herbeizuführen. Vielmehr müssen erst mehrere solcher Entscheidungen durch die Simulationsumgebung gebündelt und zu einer Gruppenentscheidung transformiert werden. Diese Gruppenentscheidung ist dann in der Lage ein neues Ereignis in EMIL-S auszulösen. Die Umsetzung dieser Anforderung erforderte mehrere Versionen bis ein EMIL-S fähiges, dem Micro Finance entsprechendes, Modell gefunden war.

Basierend auf dem Vortrag Pablo Lucas am 29.01.2009 in Koblenz wurde ein erstes Modell erstellt, welches, da keine Rücksprache stattfand, nicht der ursprünglichen Idee Pablo Lucas entsprach. Trotzdem stellte Pablo Lucas später fest, das auch dieses Modell in der Realität vorkommt und somit für eine weitere Simulation in Frage kommen würde.

Generell stehen sich beim Micro Finance zwei Spieler gegenüber, zum einen das MFI (Bank), zum anderen Kreditnehmer die sich als Gruppe organisieren und nur als Gruppe mit dem MFI in Kontakt treten, gleich ob es sich um Abzahlungen oder Kreditwünsche handelt. Eine Gruppe ist als Ganzes für ihren Vertrag mit dem MFI verantwortlich und muss Ausfälle von Einzelmitgliedern ausgleichen. Deshalb kann die Gruppe immer erst dann mit der Bank in Kontakt treten, wenn alle Einzelmitglieder ihre Entscheidungen gefällt haben. Ein erster Gedankengang war, zur Sammlung der Einzelentscheidungen und Kommunikation mit der Bank einen Gruppenagenten einzuführen, der keine eigene Entscheidungsgewalt besitzt. Von dieser Idee wurde nach Erstellung eines Probemodells wieder Abstand genommen, da alle Aufgaben eines solchen Agenten auch von der Modellumgebung ohne entsprechenden EMIL-S Vertreter ausgeführt werden können.

Das folgende UML Aktivitätsdiagramm erklärt die Anfangsversion. Die Zeitrechnung dieses Modells erfolgt in Quartalen. In einem Quartal müssen die Gruppenagenten ihre Ratenzahlung vornehmen, ggf. mehr Kredit verlangen, über Nichtzahler urteilen und dies ggf. mehrfach. Das

Urteil variiert dabei von Ermahnung bis zu Ausstoß aus der Gruppe. Ferner kann die Gruppe entscheiden die Rate des Nichtzahlers zu übernehmen. Der Kreditwunsch ist unabhängig von der eigenen Zahlung und wird parallel ausgeführt. Sind die Agenten der Gruppe zu Entschlüssen gekommen, wird die Bank informiert, die nun reagiert. Wurde die Gruppenrate nicht komplett gezahlt kann die Bank die Kreditwürdigkeit der Gruppe herabsetzen, was sich in der Folge darauf auswirkt ob noch neuer Kredit gewährt wird und somit über das Weiterbestehen der Gruppe entscheidet. Nach den Bankentscheidungen beginnt ein neues Quartal. Ein UML Diagramm für dieses erste Modell findet sich in Abbildung 19.

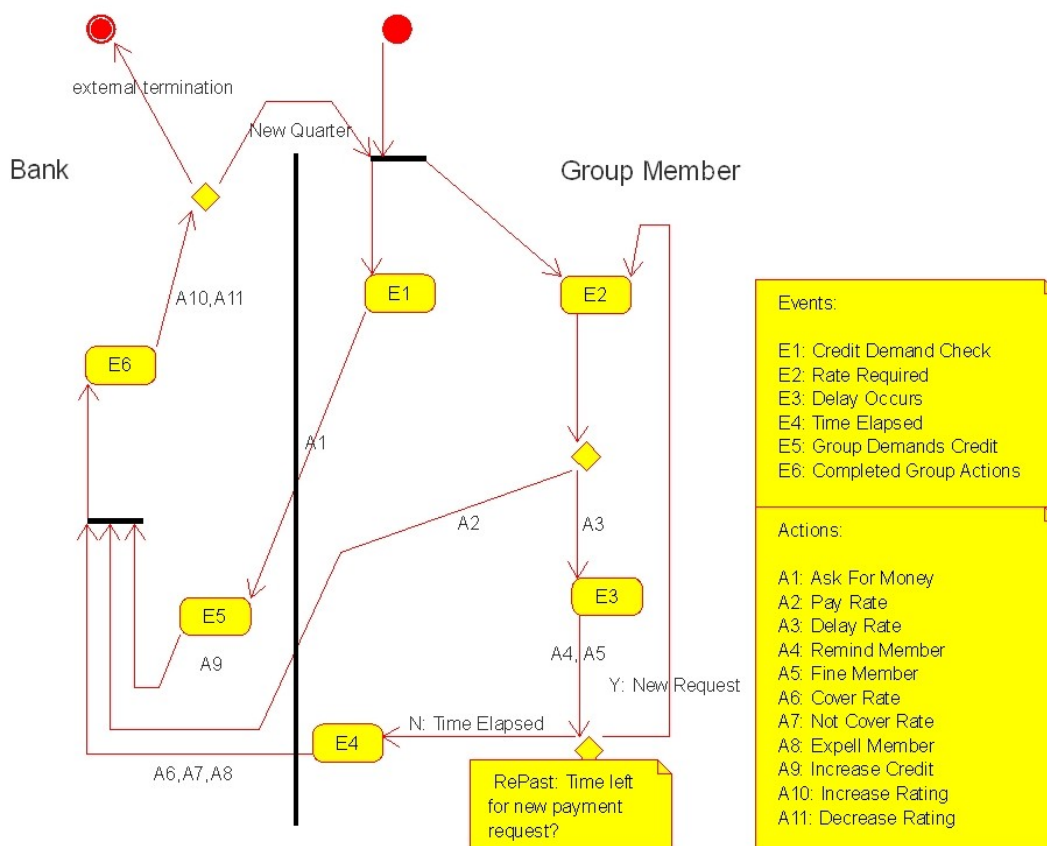


Abbildung 19: Micro Finance UML Modell - Anfangsversion

Außer der Modellierung dieser Aktivitäten, die bei der Simulation EMIL-S Entscheidungen erfordern, sind zusätzlich Variablen nötig die ausschließlich in RePast implementiert werden und eine Modellumgebung herstellen, die ggf. mitverantwortlich für das Auslösen von Ereignissen ist. Dazu gehört hier eine Kontoführung mit Konten für Einzelagenten.

Abbildung 20 stellt die Übersetzung des Aktivitätsdiagramms in ein EMIL-S IRB-fähiges Baummodell dar.

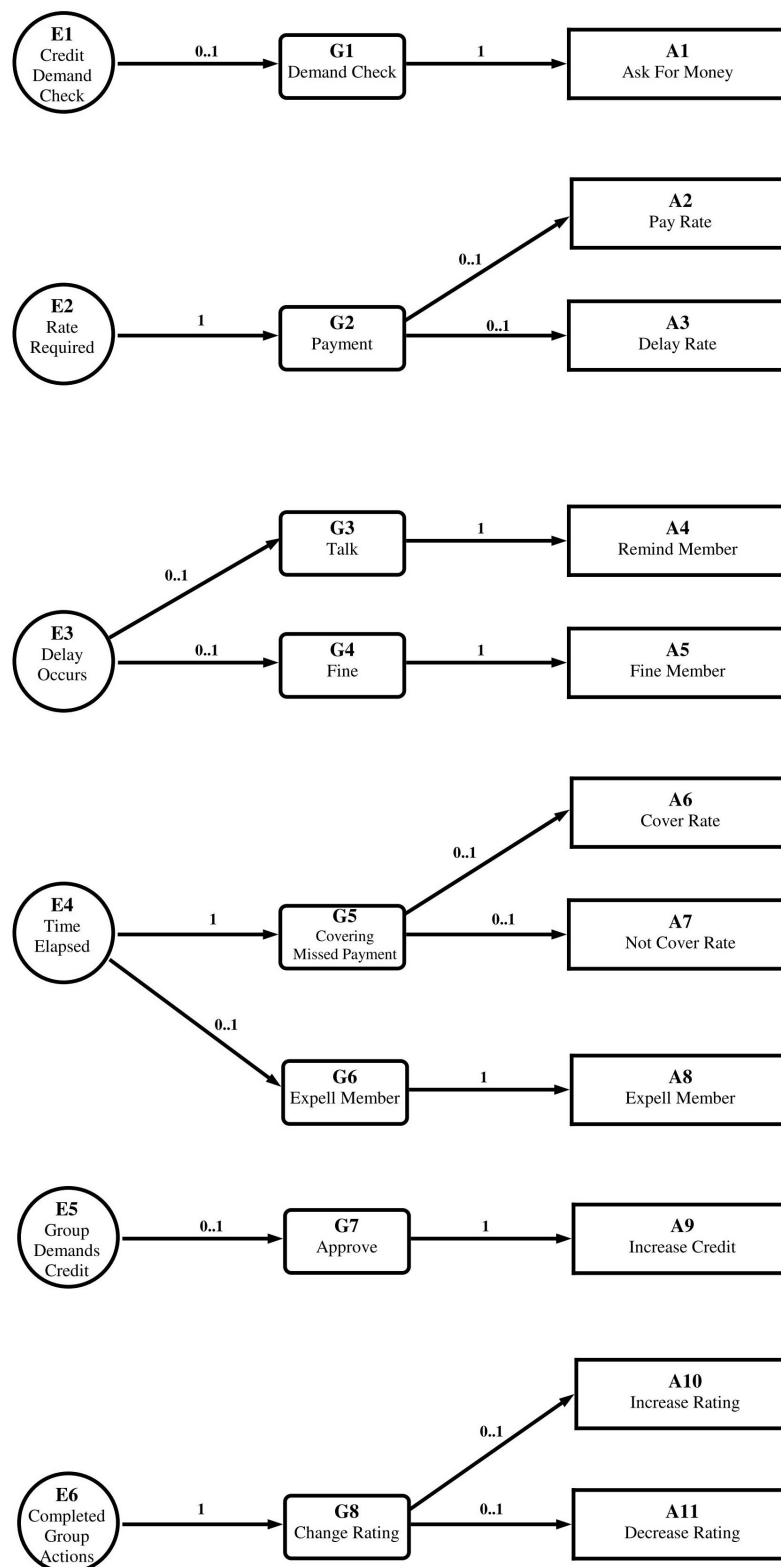


Abbildung 20: EMIL-S Micro Finance Baummodell - Anfangsversion

Dieses Modell wurde in RePast implementiert und arbeitete mit der EMIL-S Schnittstelle fehlerfrei zusammen, allerdings ohne Lernverhalten zu zeigen. Das die Lernalgorithmen noch fehlten war für

die Simulation das kleinere Problem, da sie unabhängig von der Implementierung arbeiten. Das größere Problem war es, den EMIL-S Agenten wissen zu lassen ob eine von der Gruppe ausgesprochene Strafe ihn betraf oder jemand anderen, und herauszufinden welche Rolle die einzelnen Aktions- und Ereignis-Modale dabei spielen könnten.

Die in dieser Version versuchte Lösung war wie folgt: Ausgelöste Ereignisse sollten von RePast mit einem individuellen Zeitschlüssel (Time-Stamp) versehen werden, der von EMIL-S an die Aktionen, die auf dieses Ereignis als Agentenentscheidung folgen, weitergegeben wird. Ferner speichert der EMIL-S Agent den Time-Stamp mit dem Ereignis und seiner erfolgten Entscheidung in seinem Event Board. Ist die nun erfolgte Aktion derart, das sie eine Reaktion durch andere Agenten erfordert, wird dem reagierenden Agenten der Time-Stamp des Verursacherereignisses mitgeliefert. Betrachten wir konkret ein solches reaktives Ereignis, E3 aus dem Baummodell Abb.16. Die Lösung war, die nun möglicherweise folgenden Aktionen A4, A5 als Norm-Invocations mit dem Modal Sanction zu definieren, und den Time-Stamp des Verursachers an die Verurteilung anzuhängen. Die Nachricht der erfolgten Verurteilung sollte für jeden sichtbar sein, allerdings würde nur der Verursacher diese auf sich beziehen, da er den angehängten Verursacher-Time-Stamp als seinen eigenen erkennt.

Diese Lösung besitzt Mängel, die sich bei weiterer Prüfung als nicht vereinbar mit einer akkuraten Modellierung des Szenarios zeigten. So würde bspw. bei einer Gruppe mit 10 Mitgliedern, der Verursacher neunmal eine Sanktion erhalten, in der Realität aber nur einmal. Die n:1 Beziehung wäre also stärker gewichtet als die 1:n oder 1:1 Beziehungen, die tatsächlich nur zu einer einzigen Botschaft für alle Verursacher führen würden. Außerdem wird das Auslösen von Norm Invocations nicht an RePast gemeldet, was allerdings zu einer zwingenden Änderung in EMIL-S geworden wäre, um die Entscheidungen in der Umgebungssimulation zur Berechnung weiterer Ereignisse verfügbar zu haben.

Die Arbeit an diesem Modell wurde dann letztendlich eingestellt als sich, bei einem Szenario-Workshop an dem auch Pablo Lucas teilnahm, zeigte, daß unabhängig von den erwähnten EMIL-S Schwierigkeiten, das Modell nicht dem Ablauf der von ihm untersuchten Gruppen entspricht. Trotzdem, so Lucas, sollte dieses Modell, wenn möglich, überarbeitet und ebenfalls simuliert werden, da es diese Form des Gruppenablauf in der Mikrofinanzierung ebenfalls gäbe und ein Vergleich der Ergebnisse interessant sein könnte. Diese Arbeit wird sich allerdings nur noch mit dem jetzt folgenden, nach Rücksprache entstandenen Modell befassen.

### 3.1.2 Endversion

Wesentliche Änderungen im Gruppenablauf bestehen darin, daß zu dem Quartal als bisheriges Zeitintervall ein übergeordnetes dazukommt; der Zyklus. Der Zyklus ist der Zeitraum in dem ein einmal aufgenommener Kredit abbezahlt werden muss. Die Zahlungen werden zu gleichen Teilen in den Quartalen eines Zyklus geleistet. Die Möglichkeit weiteren Kredit aufzunehmen besteht während eines Zyklus nicht. Das Aktivitätsdiagramm aus Abbildung 21 zeigt das neue Modell.

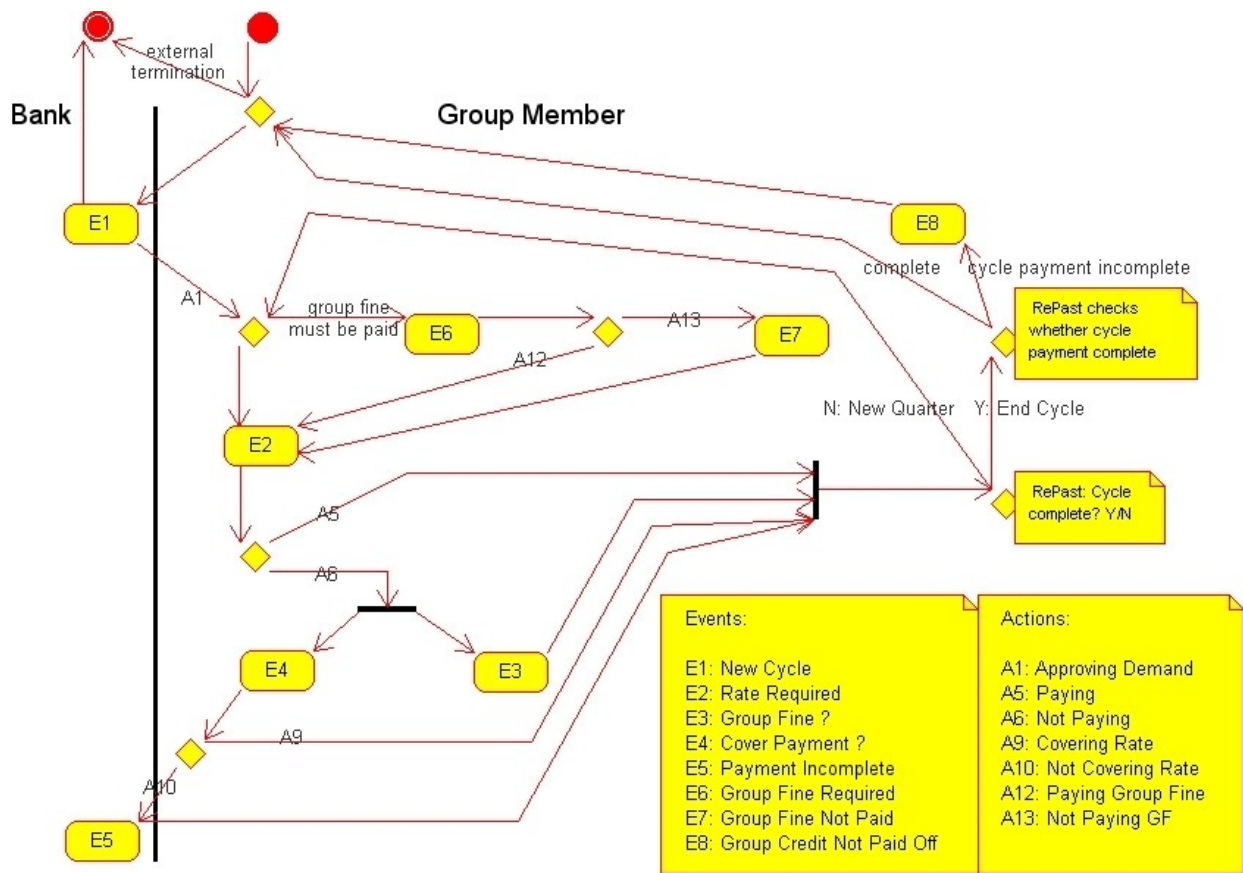


Abbildung 21: Micro Finance UML Modell - Endversion

In chronologische Reihenfolge beginnt das Modell mit dem Ereignis E1, welches zur IRB des Agententypen Bank gehört. Die Bank entscheidet ob die Gruppe kreditwürdig ist oder nicht. In ersten Fall wird Geld zugewiesen und ein neuer Zyklus beginnt mit dem ersten Quartal. Im zweiten Fall wird die Simulation durch die Bank beendet. Im ersten Fall gelangen wir für jeden Agenten des Typs Group Member zum Entscheidungsknoten mit eingehender Kante A1; die folgende Entscheidung wird von RePast getroffen und ist nicht Teil von EMIL-S: Wurde in einer vorhergehenden Runde dem Agenten ein Bußgeld durch die Gruppe auferlegt, verfolgt das Modell die zu E6 führende Kante, ist dies nicht der Fall springt es zu E2. In E6 wird der Agent aufgefordert

sein Bußgeld nun zu bezahlen; entscheidet er sich dafür, geht es über die Kante A12 jetzt erst in E2. Weigert der Agent sich und wählt die Aktion A13, folgt E7, in dem alle anderen Agenten der Gruppe über einen Ausstoß des Nichtzahlers aus der Gruppe abstimmen. Sollte der erfolgen verschwindet der Agent aus der Simulation. Entscheidet man sich gegen einen Ausstoß, wird für den betroffenen Agenten nun E2 ausgeführt. In E2 muss der Einzelagent sich entscheiden die fällige Quartalszahlung auszuführen, tut er dies ist das Quartal, ihn betreffend, abgeschlossen. Will er nicht zahlen und löst die Aktion A6 aus werden davon zwei Ereignisse parallel ausgelöst, E3 und E4. In E3 wird von den übrigen Gruppenmitgliedern entschieden, ob eine Strafzahlung an die Gruppe auferlegt wird, die wie wir gesehen haben erst am Beginn des nächsten Quartals eingefordert wird. Das Ereignis E4 wird ebenfalls für alle Übrigen ausgelöst und behandelt die Frage, ob die Ratenzahlung des Nichtzahlers durch die anderen Mitglieder übernommen werden soll. Wird sich dafür entschieden (A9: Covering Rate), erfährt die Bank nichts von dieser gruppeninternen Unregelmäßigkeit. Sollte man A10 bevorzugen, also den fehlenden Betrag nicht ausgleichen, wird die Bank nach Aufruf von E5 eine Strafzahlung für die ganze Gruppe aussprechen.

Sind diese Vorgänge für alle Gruppenmitglieder erledigt, prüft RePast ob ein weiteres Quartal im Zyklus verbleibt, wenn ja wird dieses ohne Weiteres aufgerufen. Sollte es sich um das letzte Quartal handeln, steht durch RePast die Prüfung an, ob eine Restschuld bei der Bank verblieben ist, was laut Modell nicht der Fall sein darf. Wenn nein, wird ein neuer Zyklus durch Aufruf von E1 begonnen. Im anderen Fall werden ein letztes Mal alle Gruppenmitglieder aufgefordert die Restzahlung durchzuführen. Wie sich die Gruppenmitglieder auch entscheiden, in jedem Fall wird im Anschluß das Ereignis E1 ausgelöst. Die Bank wird allerdings das Nichtbegleichen der Restschuld als stark negatives Signal auffassen, was sich auf die Möglichkeit des Fortbestehens der Gruppe auswirkt.

Übersetzt in ein EMIL-S Baummodell ergeben sich die Abbildungen 22 und 23.

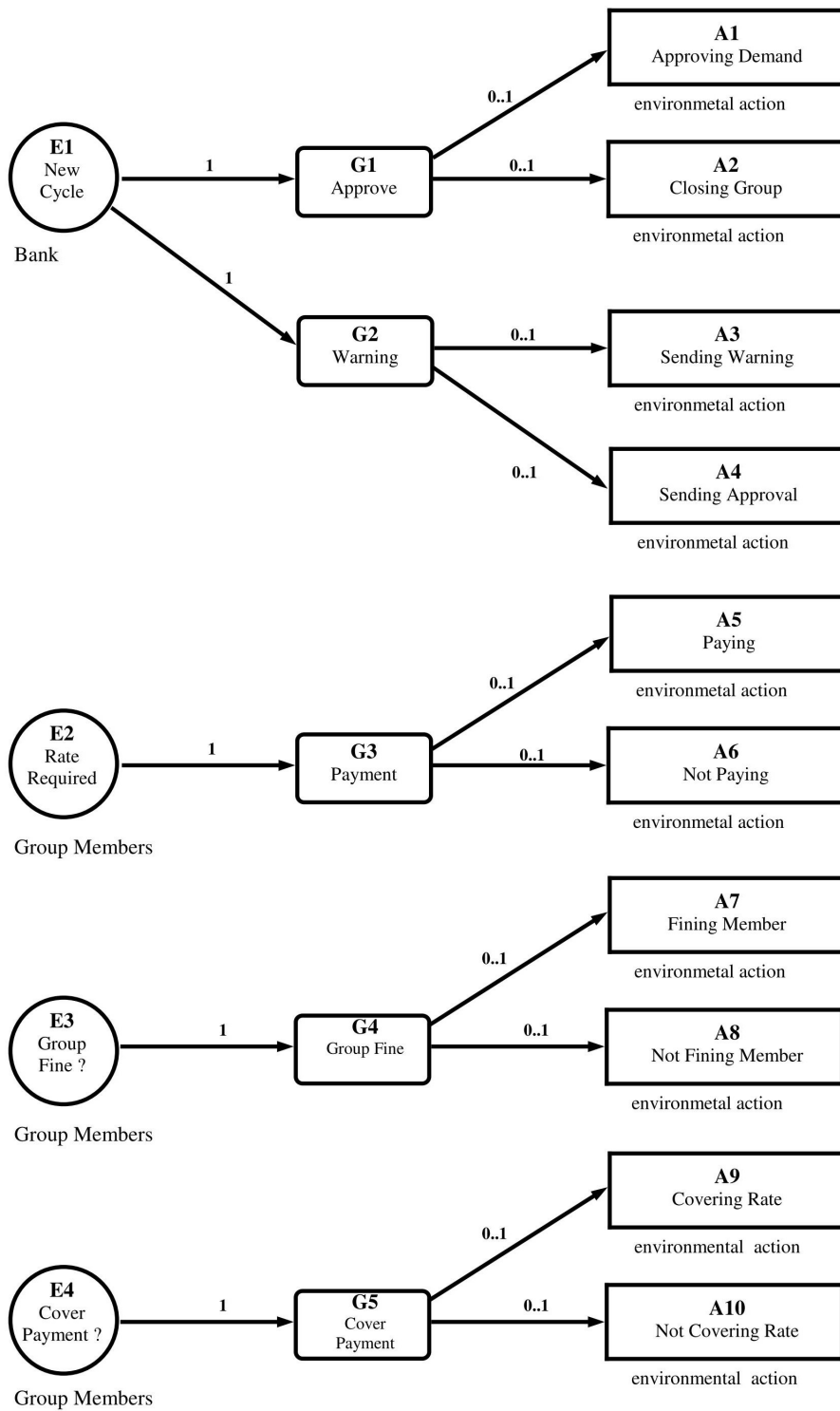


Abbildung 22: EMIL-S Micro Finance Baummodell - Endversion (1/2)

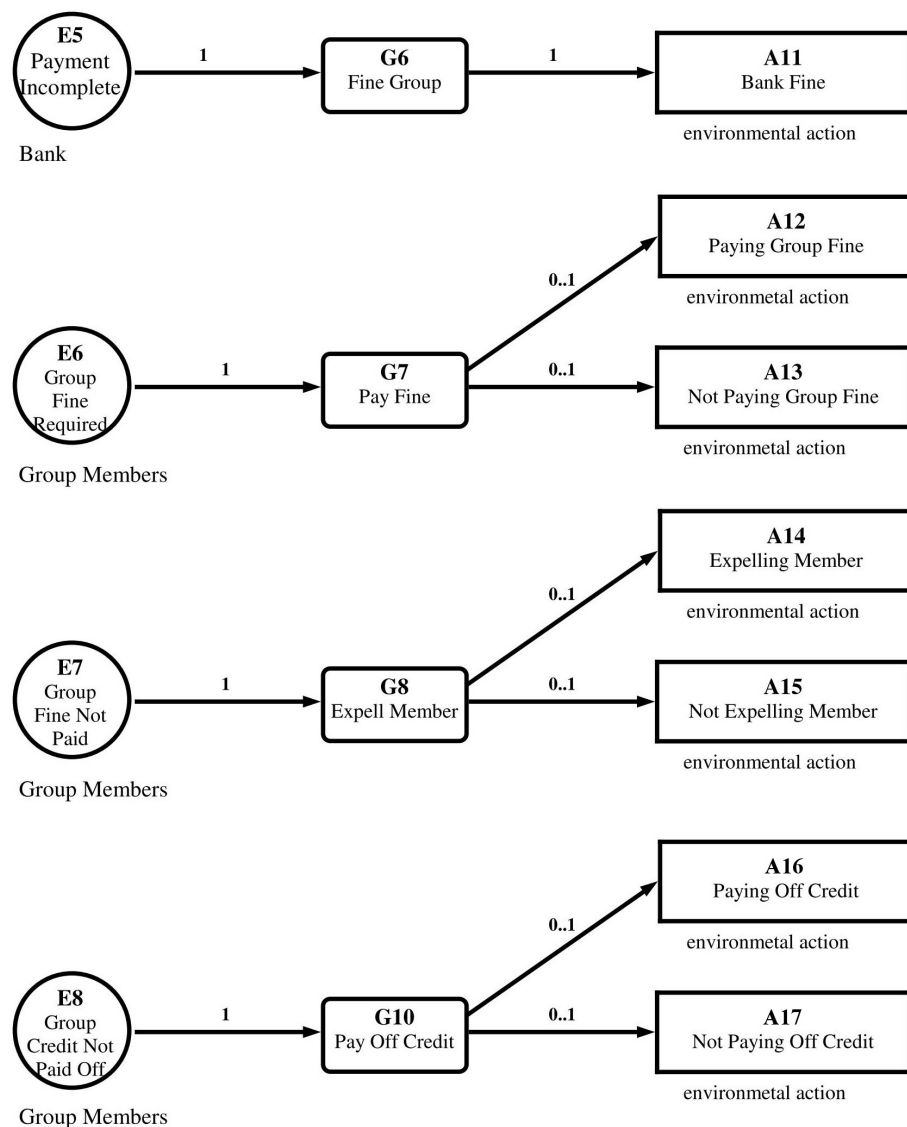


Abbildung 23: EMIL-S Micro Finance Baummodell - Endversion (2/2)

Nachdem sich im ersten Modell die umfangreiche Benutzung von Norm Invocations als unpraktikabel herausstellte, musste für dieses Modell ein neuer Weg gefunden werden, Bewertungen zwischen Agenten auszutauschen. Dazu wurden in einem ersten Schritt alle Agentenaktionen als "Environmental" definiert und dadurch von EMIL-S an RePast weitergeleitet. RePast kann nun aus kumulierten Einzelentscheidungen, Gruppenentscheidungen produzieren (bspw. durch Mehrheitsentscheidung). Die gültige Mehrheitsentscheidung wird dann sowohl im Umgebungsmodell für die Generierung neuer Ereignisse weiterverwendet, als auch mittels negativer oder positiver expliziter Norm Invocation an den verursachenden Agenten gemeldet und können so ein Lernen auslösen. Die Arbeitsweise wird in 3.2.2 noch genauer dargestellt.



## 3.2 Die Simulationsumgebung oder RePast Ebene

### 3.2.1 Die Schnittstelle RePast / EMIL-S

Damit RePast mit EMIL-S kommunizieren kann, muss das RePast Modell dem entsprechenden EMIL-S Modell zugeordnet werden. Dazu wird in der RePast "setup"-Methode (Abbildung 24) welche zur agententypunabhängigen Modellklasse gehört, der EMIL-S Controller mit Pfadangabe der richtigen XML IRB aufgerufen.

```
public void setup() {
    Controller
        .initializeController(new File(
            "C:/Program Files/RepastSimphony-1.2.0/" +
            "workspace/MicroFinance_Emils/MicroFinanceIRB(2).xml"));
}
```

Abbildung 24: Klasse *MicroFinanceGeneral*: Aufruf des XML Regelsatzes

In jeder Agententypklasse müssen diverse Klassen von EMIL-S, wie in Abbildung 25 gezeigt, importiert werden.

```
import Emil.agent.IEMILAgent;
import Emil.agent.IEMILAgentWrapper;
import Emil.message.EMILMessage;
import Emil.message.IEMILMessage;
import Emil.message.StringContent;
```

Abbildung 25: Klasse *MicroFinanceGroupMember*: zu importierende Klassen

Die Agententypklasse kann nun den "IEMILAgentWrapper" (Abbildung 26) implementieren.

```
public class MicroFinanceGroupMember implements IEMILAgentWrapper {
    // ...
}
```

Abbildung 26: Klasse *MicroFinanceGroupMember*: *IEMILAgentWrapper*

Zum Nachrichtenaustausch mit EMIL-S wird die Schnittstelle "EMILMessage" verwendet. Zur Auslösung eines Ereignisses durch RePast in EMIL-S (Abbildung 27) wird eine neue EMILMessage erzeugt, welche die Art (Modal), die aktuelle Rundenzeit und den Namen des Ereignisses enthält, und an EMIL-S gesendet.

```

public void e8GroupCreditNotPaidOff(int eventStamp) {
    emilLayer.processMessage(new EMILMessage(ID, ID, Modal.ASSERTION,
        new ENVContent(eventStamp, "E8", null, 0)));
}

```

Abbildung 27: Klasse *MicroFinanceGroupMember*: Senden einer Nachricht an EMIL-S, die ein Ereignis bedeutet

Um Nachrichten, also Aktionen von EMIL-S zu empfangen, muss die Methode `sendMessage` implementiert werden, welche selbsttätig von EMIL-S aufgerufen wird sobald es eine Nachricht an RePast senden möchte. EMIL-S übergibt eine Nachricht vom Typ `IEMILMessage`, aus der RePast die gewollte Aktion auslesen kann, in der Form wie in Abbildung 28 dargestellt.

```

public boolean sendMessage(IEMILMessage arg0) {

    int key = 0;

    if (arg0.getContent().getContent().equals("A9")) {
        key = 9;
    }

    switch (key) {

        case 9:
            model.increaseCredit();
            break;

        default:
            System.out.println("Agent " + ID + " Keine neuen Befehle.");
            break;
    }

    return false;
}

```

Abbildung 28: Klasse *MicroFinanceGroupMember*: Empfangen von Nachrichten

Schließlich können die benötigten physikalischen Agenten als Instanzen der Agententypklassen geschaffen werden, um sie dann jeweils einem EMIL-S Agent vom gleichen Typ zuzuordnen. Dazu wird der Aufruf aus Abbildung 29 an EMIL-S gesendet, der Agententyp, ID und Objektreferenz des RePast Agenten übergibt.

```

public MicroFinancialInstitution(MicroFinanceGeneral model) {
    this.model = model;
    emilLayer = emil.Controller.getInstance().addAgent(
        "MicroFinancialInstitution", ID, this);
    IDNumber++;
    ID = IDNumber;
}

```

Abbildung 29: Klasse *MicroFinancialInstitution* (Bank): Konstruktor

Die Graphik, Abbildung 30, stellt den erklärten Sachverhalt nochmals anschaulich dar.

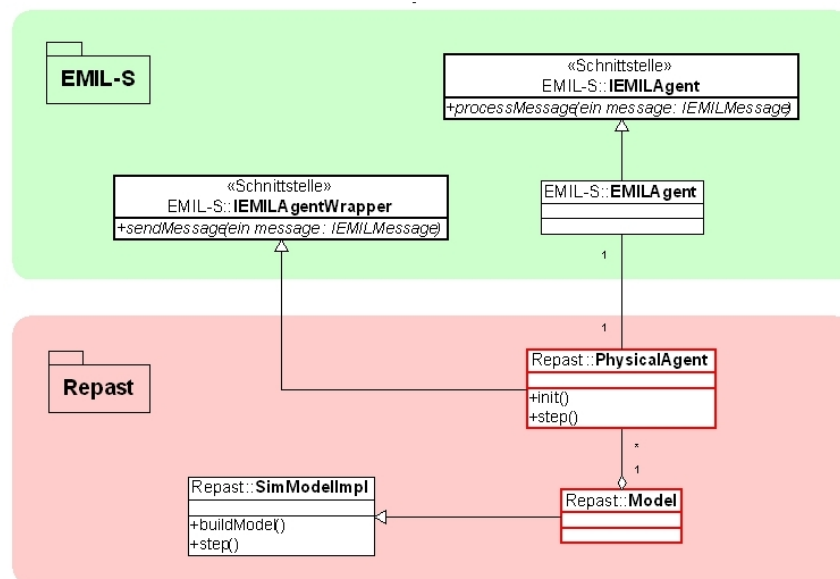


Abbildung 30: Die Schnittstelle EMIL-S/RePast, Quelle: Ulf Lotzmann/Michael Möhring, Konferenzvortrag Universität Koblenz, 29.01.2009

## 3.2.2 Programmierung der Modellumgebung

### 3.2.2.1 Die Grundzüge

Nachdem die Agenten wie in Kapitel 3.2.1 gezeigt erzeugt wurden, braucht das Programm, das die Modellumgebung herstellt, noch alle die Methoden, welche EMIL-S mit den richtigen Eingaben versorgen und die von EMIL-S kommenden Ausgaben verstehen, um daraus neue Ereignisse, entsprechend dem Micro Finance Modell, abzuleiten.

In RePast wird die Programmiersprache JAVA verwendet, welche zu den objektorientierten Sprachen zählt. Deshalb bietet es sich an für jeden Agententypen eine eigene Klasse zu erstellen, die beliebig oft instanziiert werden kann. "A natural way of programming agents is to use an 'object-oriented' programming language. In this context, 'objects' are program structures that hold both data and procedures for operating on those data. [...] objects are created from templates called 'classes' [...] . [...] Once a set of classes has been defined, individual agents are generated by creating instances from them ('instantiation')." <sup>34</sup> Für dieses Modell bedeutet dies eine Klasse für Mikrofinanz-Gruppenmitglieder und eine Klasse für Banken. Da in dieser Arbeit nur jeweils eine Gruppe betrachtet wird, muss von der Klasse Bank jeweils nur eine Instanz erschaffen werden.

<sup>34</sup> Quelle: *Simulation for the Social Scientist*, S.183, [SSS.05]

Zusätzlich benötigt das Modell eine Klasse zur Steuerung und Berechnung, also für jene Aufgaben die ein Gruppenagent übernommen hätte. Diese Klasse wird selbstverständlich auch nur einmal benötigt und enthält die von RePast vorgegebenen Steuerungsmethoden.

In jedem RePast Zeitschritt, in diesem Modell ein Quartal wird die Methode von Abbildung 31 in der Steuerungsklasse einmal aufgerufen, aus der sich alle weiteren Tätigkeiten ergeben.

```
public void execute() {
    setupCycle();
    displaySurface.updateDisplay();
    setupQuarter();
    quarter++;
    endCycle();
}
```

Abbildung 31: Klasse *MicroFinanceGeneral*:  
Quartal-einleitende Methode

### 3.2.2.2 Eine beispielhafte Kommunikation mit EMIL-S

Ein Vorkommnis kann sein, auf eine ausstehende Quartalszahlung hinzuweisen. Die Methode der Steuerungsklasse aus Abbildung 32 leitet den Vorgang ein. Bei jeder Instanz des Typen Group Member wird die Methode aufgerufen, die EMIL-S informiert. Dabei wird bei jedem neuen Ereignis von der Steuerungsklasse ein einzigartiger Zeitstempel ("eventStamp") erzeugt und weitergegeben, an dem das Ereignis seinen Folgen eindeutig zuzuordnen ist:

```
public void e2RateRequired() {
    eventStamp++;
    for (MicroFinanceGroupMember mem : memberList) {
        mem.e2RateRequired(eventStamp);
    }
}
```

Abbildung 32: Klasse *MicroFinanceGeneral*: Information der  
Agenteninstanzen über ein neues Ereignis

Das Gruppenmitglied verfügt über die Schnittstelle die im Fall Ereignis E2 wie in Abbildung 33 aussieht. Der erhaltene Zeitstempel wird an EMIL-S weiter übergeben. Weil es sich um eine Nachricht vom Typ ENVContent, also ein "Environmental" Ereignis handelt, wird später auch eine Ausgabe von EMIL-S an die Umgebung erfolgen.

```
public void e2RateRequired(int eventStamp) {
    emilLayer.processMessage(new EMILMessage(ID, ID, Modal.ASSERTION,
        new ENVContent(eventStamp, "E2", ID, 0)));
}
```

Abbildung 33: Klasse *MicroFinanceGroupMember*: Der RePast/JAVA Agent  
informiert seine EMIL-S Ebene über ein Ereignis

In der Ausgabe, die durch den Aufruf der Methode `sendMessage` erfolgt, wird die Nachricht (IEMILMessage), welche die Entscheidung des EMIL-S Agenten darstellt, von EMIL-S als Argument übergeben. Durch if-Sätze wird nun für jede in der IRB definierte, mögliche Aktion überprüft, um welche EMIL-S Antwort es sich handelt und die entsprechende Umgebungsveränderung eingeleitet. Im Codefragment Abbildung 34, welches Teil der Agentenklassen ist, wird die Behandlung der Nachrichten A5:Paying und A6:Not Paying dargestellt. Im zweiten Fall wird die Nachricht des EMIL-S Agenten von der Modellumgebung in einer Nachrichtenliste (JAVA ArrayList) abgespeichert, da es zu Reaktionen auf die Entscheidung kommen könnte und dann in eben dieser Liste nachgesehen werden kann zu welcher Aktion die erfolgte Reaktion gehört.

```

public boolean sendMessage(IEMILMessage arg0) {

    if (arg0.getContent().getContent().equals("A5")) { // PayRate
        System.out.println("Group Member: " + ID
            + " decides to pay his Quarter-Rate.");
    }

    if (arg0.getContent().getContent().equals("A6")) { // NotPayRate
        System.out.println("GroupMember " + ID
            + " decides to not pay his rate.");
        OriginalMessage a6 = new OriginalMessage((int) arg0.getContent()
            .getTimestamp(), arg0, this, "A6");
        this.originalMessageList.add(a6);
        model.e3GroupFine(a6);
        model.e4CoverPayment(a6);
    }
}

```

Abbildung 34: Klasse `MicroFinanceGroupMember`: Der `RePast/JAVA` Agent erhält die Reaktion der EMIL-S Ebene zu einem Ereignis

### 3.2.2.3 Ereigniseintritt basierend auf gespeicherten Nachrichten

Abgelegt werden die erhaltenen Nachrichten in einem selbstdefinierten Format, der Klasse `OriginalMessage`, die neben der ursprünglichen `IEMILMessage`, den Ereigniszeitstempel, die Art der Aktion und den Agententyp, enthält. Die Klasse besitzt zwei Konstruktoren um im Parameter jeweils einen Agenten vom entsprechenden Typ übergeben zu können. Abbildung 35 zeigt Variablen und Konstruktoren dieser Klasse.

```

public class OriginalMessage {
    private int originalEventStamp;
    private IEMILMessage originalMessage;
    private String actionType;
    MicroFinanceGroupMember groupMember = null;
    MicroFinancialInstitution bank = null;

    public OriginalMessage(int originalEventStamp,
        IEMILMessage originalMessage, MicroFinanceGroupMember groupMember,
        String actionType) {
        this.originalEventStamp = originalEventStamp;
        this.originalMessage = originalMessage;
        this.actionType = actionType;
        this.groupMember = groupMember;
    }

    public OriginalMessage(int originalEventStamp,
        IEMILMessage originalMessage, MicroFinancialInstitution bank,
        String actionType) {
        this.originalEventStamp = originalEventStamp;
        this.originalMessage = originalMessage;
        this.actionType = actionType;
        this.bank = bank;
    }
}

```

Abbildung 35: Klasse *OriginalMessage*: stellt ein Format bereit in dem Nachrichten gespeichert werden

Wie sich im Codefragment der Steuerungsklasse, Abbildung 36, zeigt werden Umgebungsentscheidungen auf Basis der so gespeicherten Nachrichten getroffen. Der innere if-Satz zeigt die Abfrage der Nachrichtenliste nach dem Vorhandensein eines bestimmten Nachrichtentyps. Ist dieser vorhanden, wird auf ein Fehlverhalten geschlossen, welches nun geahndet wird. In diesem Fall wird der Agent aufgefordert eine Strafe zu bezahlen.

```

public void e6GroupFineIsRequired() {
    ArrayList<MicroFinanceGroupMember> temporaryMemberList =
        new ArrayList<MicroFinanceGroupMember>();
    temporaryMemberList.addAll(memberList);
    for (MicroFinanceGroupMember mem : temporaryMemberList) {
        ArrayList<OriginalMessage> temporaryMessageList = new ArrayList<OriginalMessage>();
        temporaryMessageList.addAll(mem.getOriginalMessageList());
        for (OriginalMessage originalMessage : temporaryMessageList) {
            if (!expelledMembers.contains(mem)) {
                System.out
                    .println("\nGroup Member "
                        + mem.getID()
                        + " shall pay a required Group-Fine."
                        + " The incident of not-paying occurred at event-stamp: "
                        + originalMessage.getOriginalEventStamp());
                if (originalMessage.getActionType().equals("A6")) {
                    eventStamp++;
                    mem.e6GroupFineRequired(eventStamp, originalMessage);
                }
            }
        }
    }
}

```

Abbildung 36: Klasse *MicroFinanceGeneral*: Ereignisse werden abhängig von Nachrichten der Vergangenheit ausgelöst

### 3.2.2.4 Die Berechnung von Gruppenentscheidungen

Dabei sind die meisten Ereignisse des Micro Finance Modells durch eine vorangegangene Gruppenentscheidung bedingt. So kommt die Aufforderung zur Zahlung einer Strafe nur zustande nachdem die Gruppe ein Quartal zuvor entschieden hat, daß der Nichtzahler eine solche Zusatzzahlung zu leisten hat. In Abbildung 37 wird gezeigt wie eine Gruppenentscheidung ermittelt wird. Zuerst wird allen Mitgliedern mitgeteilt welcher Agent nicht bezahlt hat; dies geschieht in der ersten for-Schleife. Die darauf erfolgenden Einzelentscheidungen werden nun ausgezählt. Dazu werden zwei Hilfsvariablen zur Zählung erschaffen und im Anschluß, durch die zweite for-Schleife, bei jedem Agenten in der Nachrichtenliste gesucht wie er sich entschieden hat. Die ausgezählte Entscheidung wird dann aus der Nachrichtenliste wieder gelöscht. Generell werden in der Simulation Nachrichten sofort gelöscht, nachdem sie nicht mehr benötigt werden; somit entspricht die Nachrichtenliste keiner Agentengeschichte sondern ist tatsächlich nur ein Zwischenspeicher.

```

public void e3GroupFine(OriginalMessage originalMessage) {
    eventStamp++;
    MicroFinanceGroupMember memberNotPayingRate = originalMessage
        .getGroupMember();
    for (MicroFinanceGroupMember mem : memberList) {
        if (mem != memberNotPayingRate) {
            mem.e3GroupFine(eventStamp, originalMessage,
                memberNotPayingRate);
        }
    }
    int numberOfJudgingMembers = 0;
    int numberOfGroupMembersForFining = 0;

    for (MicroFinanceGroupMember mem : memberList) {
        if (mem != memberNotPayingRate) {
            numberOfJudgingMembers++;
            OriginalMessage messageToRemove = null;
            for (OriginalMessage message : mem.getOriginalMessageList()) {
                if (eventStamp == message.getOriginalEventStamp()) {
                    messageToRemove = message;
                    if (message.getActionType().equals("A7")) {
                        numberOfGroupMembersForFining++;
                    }
                }
            }
            mem.getOriginalMessageList().remove(messageToRemove);
        }
    }
}

```

Abbildung 37: Klasse *MicroFinanceGeneral*: Die Gruppenmitglieder werden für eine Gruppenentscheidung befragt

Der zweite Teil der Methode *e3GroupFine*, in Abbildung 38, errechnet zuerst den Quotienten aus

Agenten die eine Strafe befürworten und der Gesamtzahl der richtenden Agenten. Der Quotient ist immer eine Zahl zwischen 1 und 0 und stellt die Stärke der Gruppenentscheidung dar. Der Quotient wird dann durch Abschnittsweise Definition in eine Bestrafungsstärke umgerechnet. Die Sanktion wird dann mit der ermittelten Bestrafungsstärke an den Nichtzahler EMIL-S Agenten weitergegeben. Zusätzlich findet, falls mehr als 50% der Gruppe für eine Bestrafung stimmen, eine Änderung in der Modellumgebung statt. Die verursachende Aktion des Nichtzahlens, also die Nachricht A6 des Nichtzahlers, welche zum Aufruf des Ereignisses E3 geführt hat, wird explizit in der Nachrichtenliste des Nichtzahlers, über das Quartal hinaus, belassen. Dadurch kann die Modellumgebung im nächsten Quartal feststellen, ob eine Strafzahlung fällig ist (vgl. Abbildung 36). Diese Nachricht wird solange in der Liste belassen, bis der betreffende Agent die zugehörige Strafzahlung geleistet hat.

```

double groupDecisionFining = (((double) numberOfGroupMembersForFining) /
    ((double) numberOfJudgingMembers));
double strengthOfPunishment = 0;

if (groupDecisionFining == 0) {
    strengthOfPunishment = +0.2;
}
if (0 < groupDecisionFining & groupDecisionFining <= 0.25) {
    strengthOfPunishment = -0.1;
}
if (0.25 < groupDecisionFining & groupDecisionFining <= 0.5) {
    strengthOfPunishment = -0.4;
}
if (0.5 < groupDecisionFining & groupDecisionFining <= 0.75) {
    strengthOfPunishment = -0.7;
}
if (0.75 < groupDecisionFining) {
    strengthOfPunishment = -1;
}
eventStamp++;
System.out.println("The Group Decision for fining is: "
    + groupDecisionFining);
memberNotPayingRate.sanction(eventStamp, originalMessage,
    strengthOfPunishment);
if (groupDecisionFining <= 0.5) {
    memberNotPayingRate.getOriginalMessageList()
        .remove(originalMessage);
}
}

```

Abbildung 38: Klasse *MicroFinanceGeneral*: Die Gruppenentscheidung wird gewichtet und ausgeführt

### 3.2.2.5 Lernen durch Auslösung von Sanktionen

Ist eine Sanktion wie im gerade erwähnten Fall notwendig, wird sie beim jeweiligen Agenten aufgerufen, unter Angabe der Entscheidung die zur Bestrafung geführt hat, sowie der



Bestrafungsstärke. Die Abbildung 39 zeigt den Aufruf als Teil der Agentenklassen.

```

public void sanction(int eventStamp, OriginalMessage message, double sanction){

    IContent valuatedContent = null;
    if (message != null) {
        valuatedContent = message.getOriginalMessage().getContent();
    }
    emillayer.processMessage(
        new EMILMessage(
            ID,
            ID,
            Modal.SANCTION,
            new NIContent(
                eventStamp,
                "Explicit NI",
                ID,
                sanction,
                valuatedContent
            )
        )
    );
}
    
```

Abbildung 39: alle Agentenklassen: explizite Sanktion wird der EMIL-S Ebene gemeldet

Die explizite Sanktion löst nun im jeweiligen Agenten ein Lernen aus. Die Wahrscheinlichkeiten für bestimmte Aktionen verändern sich als die Folge. Die Zuordnung der Bestrafung zum Vergehen, wie es für den Großteil der in der Simulation erfolgenden Sanktion implementiert ist, hilft dem Agenten einerseits beim Lernen, andererseits ist der Spielraum für EMIL-S eigenständig diese Zuordnung vorzunehmen dadurch eingeschränkt. Es wäre allerdings mit wenig Aufwand möglich die explizite Zuordnung außer Kraft zu setzen und somit beide Varianten zu testen.

### 3.3 Simulationsläufe und Verhalten des Szenarios

#### 3.3.1 Vorbemerkungen zur Analyse

Zur Auswertung des Lernverhaltens in der Simulation wird das Eintreten bestimmter Ereignisse im Zeitverlauf beobachtet. Dazu wird während der Simulation ein Graph gezeichnet, für den vier beispielhafte Ereignisse ausgewählt wurden. Jedes Simulationsquartal wird, falls von einem bestimmten Ereignistyp Ereignisse stattfanden, die neuen Ereignisse zu der Gesamtzahl der in der Simulation stattgefundenen Ereignisse diesen Typs addiert. Es entstehen Kurven die bei einer Steigung größer null durch ihre Steilheit die Häufigkeit des jeweiligen Ereignisses zum jeweiligen Zeitpunkt anzeigen. Dabei wird der im Graphen quartalsweise addierte Wert für die Ereignishäufigkeit, durch Umrechnung, meist zwischen null und eins liegen. Dieses Umrechnungsverfahren, das folgend beschrieben wird, dient zur Verbesserung der Übersichtlichkeit

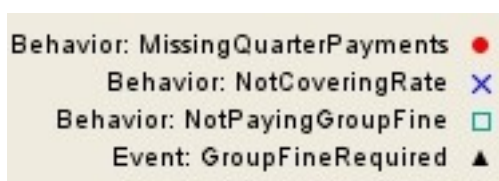
der Werte ohne dabei wichtige Informationen zu verlieren.

Die vier beobachteten Ereignisse sind:

- 1. Entscheidung Ratenzahlung nicht zu leisten (Behavior: MissingQuarterPayments): Diese Entscheidungen, summiert in einem Quartal, geteilt durch die Anzahl der Gruppenmitglieder, ergibt in jedem Quartal einen Zahl zwischen 0 und 1, und ist Ausdruck für die aktuelle Zahlungsmoral in der Gruppe. Bei einem Wert von 1 hieße dies alle Mitglieder hätten sich entschieden nicht zu zahlen. Als Formel: *verpasste Zahlungen / Gruppenmitglieder*
- 2. Entscheidung fehlende Ratenzahlung nicht auszugleichen (Behavior: NotCoveringRate): Die Anzahl der Entscheidungen wird geteilt durch die Anzahl der Gruppenmitglieder abzgl. des Nichtzahlers, sowie durch die Anzahl der verpassten Ratenzahlungen (für jede verpasste Zahlung müssen die anderen Gruppenmitglieder eine Entscheidung fällen) im Quartal. Als Formel: *Entscheidungen des Nichtausgleichens / (Gruppenmitglieder-1) / verpasste Zahlungen*
- 3. Aufforderung zur Bezahlung der "GroupFine" (Event: GroupFineRequired): Die Anzahl der Entscheidungen wird geteilt durch die Anzahl der Gruppenmitglieder.
- 4. Entscheidung keine "GroupFine" zu bezahlen (Behavior: NotPayingGroupFine): Die Anzahl der Entscheidungen wird geteilt durch die Anzahl der Gruppenmitglieder

Das Teilen durch die Anzahl der Gruppenmitglieder wird auch deshalb nötig, da während der Laufzeit Agenten aus der Gruppe ausgestoßen werden können und somit folgende Ereignishäufigkeiten nicht vergleichbar mit den vorhergegangenen wären.

Die Legende in Abbildung 40 zeigt die, für alle Graphen gültige, Zuordnung zu den verwendeten Farben.



Behavior: MissingQuarterPayments	●
Behavior: NotCoveringRate	×
Behavior: NotPayingGroupFine	□
Event: GroupFineRequired	▲

Abbildung 40: Legende für Simulationsgraphen

Die ersten beiden Ereignisse (1. und 2.) sowie die zwei weiteren (3. und 4.) gehören jeweils zusammen. Derart daß das Ereignis "NotCoveringRate" ein vorhergehendes Ereignis "MissingQuarterPayments" benötigt und ein Ereignis "NotPayingGroupFine" ein vorhergehendes "GroupFineRequired". Die Differenz jeweils beider Ereignishäufigkeiten zeigt die Reaktion der Gruppe.

Da das Lernen in der derzeitigen EMIL-S Version, wie bereits erwähnt, noch nicht selbsttätig funktioniert, musste es in diesem Stadium durch explizite Sanktionen ausgelöst werden. Zur Verdeutlichung wie die vier genannten Ereignisse also in ihrer Eintrittshäufigkeit durch Lernen verändert werden, wird hier ein Teil der expliziten Lernzusammenhänge erläutert:

- Entschließt sich ein Agent seine Ratenzahlung auszusetzen, entscheiden die anderen Agenten ob eine Strafe ausgesprochen wird; diese bezieht der Nichtzahler explizit auf die Aktion des Nichtzahlens und passt sein Verhalten an. Dies geschieht durch EMIL-S als Änderung der Wahrscheinlichkeiten im Regelwerk des XML Regelsatzes.
- Entschließen sich weniger als die Hälfte der urteilenden Gruppenagenten die Rate des Nichtzahlers zu ersetzen und somit der Bank den vollen Betrag zu zahlen, kann die Bank eine Strafe aussprechen die alle Agenten die für das Nicht-Ersetzen votiert haben explizit auf ihre Entscheidung beziehen, und daraus lernen werden.
- Entschließt sich ein Agenten die Strafzahlung die ihm von der Gruppe auferlegt wurde nicht zu bezahlen, kann dies eine scharfe Zurechtweisung durch die Gruppe nach sich ziehen, die der Nichtzahler als explizite Sanktion auf die Entscheidung des Nichtzahlens der Gruppenstrafe beziehen wird.

### 3.3.2 Simulationsläufe

Nach gestarteter Simulation zeichnet sich ein Graph wie in Abbildung 41. Auf der horizontalen Achse findet sich die Zeit in Form von Quartalen, auf der vertikalen Achse die zuvor erklärten Werte für das Verhalten der Gruppenagenten:

Das Gruppenverhalten für die ersten fünf Quartale erscheint nahezu konstant. Tatsächlich findet allerdings ab dem ersten Quartal ein Lernen statt, daß die bestraften Entscheidungen immer seltener auftreten läßt. Ab dem fünften bis siebten Quartal kann man deutlich erkennen, daß die Gruppenmitglieder ihr Verhalten ändern:

- Das Nichtzahlen der Quartalsrate (roter Graph) nimmt ab.
- Das Nichtausgleichen einer verpassten Zahlung (dunkelblauer Graph) nimmt ab, allerdings sehr langsam und im Graph fast nicht sichtbar. Dies liegt daran, das die Bestrafung für eine falsche Entscheidung nur dann erfolgt, wenn mindestens die Hälfte der Gruppe gegen einen Ausgleich stimmt. Nur dann wird die Bank informiert, nur dann wird die Bank die Gruppe bestrafen, nur dann werden diejenigen die für Nichtausgleichen gestimmt haben lernen. Da sich allerdings nach ein paar anfänglichen Bestrafungen die Gruppenentscheidung bei einem Niveau einpendelt, daß

gerade oberhalb der Mehrheit liegt, findet trotz einer immer noch sehr hohen Gruppenstimmung des Nichtausgleichens nur noch selten weiteres Lernen statt. Sozusagen müssen die Einzelagenten bei Gruppenentscheidungen weniger Verantwortung übernehmen, da sich ihre Einzelentscheidung oft neutralisiert.

- Die Aufforderungen zur Bezahlung einer Gruppenstrafe (schwarzer Graph) nehmen ab, was nicht von dem roten Graph abhängig ist, da eine nicht gezahlte Gruppenstrafe ins nächste Quartal übernommen wird und somit die Steigung des schwarzen Graphen unverändert bliebe, sogar zunehmen würde wenn sich weitere Strafen kumulieren. Bis zum zehnten Quartal sammeln die Gruppenagenten, wie im Graph ersichtlich, tatsächlich pro Quartal weitere Strafzahlungen an ohne sie abzuzahlen. Ein Indikator dafür ist die steilere Steigung des schwarzen Graphen gegenüber dem roten Graphen.
- Das Nichtbezahlen einer Gruppenstrafe (hellblauer Graph) determiniert, zusammen mit den neu hinzukommenden Strafzahlungen, die Aufforderungen zur Bezahlung für das folgende Quartal. Deutlich erkennbar setzt ab dem fünften Quartal ein starkes Lernen bei dieser Entscheidung ein, was dazu führt, daß ab dem zwanzigsten Quartal nahezu keine Gruppenstrafen mehr unbezahlt bleiben, obwohl, wie aus dem schwarzen Graph ersichtlich, auch danach noch viele Zahlungsaufforderungen folgen. Dies deutet auf ein sehr effektives Lernen hin.

Betrachtet man zum weiteren Verständnis die Quartale 16-18, so sieht man einen stark steigenden roten Graph (MissingQuarterPayments), dagegen einen flach verlaufenden dunkelblauen Graph (NotCoveringRate). Das heißt die Gruppenmitglieder haben beschlossen für jede fehlende Rate die Zahlung an die Bank zu übernehmen. Das ist eine Änderung zum anfänglichen Verhalten; in der XML IRB dieses Simulationslaufes war die anfängliche Wahrscheinlichkeit für NotCoveringRate gleich 0,9.

In den gleichen Quartalen 16-18 sieht man ein starkes Ansteigen des schwarzen Graphen (GroupFineRequired), jedoch ein nahezu flachen hellblauen Graphen (NotPayingGroupFine). Dies bedeutet, daß bei hohen Bußgeldforderungen durch die Gruppe, sich nur sehr wenige der Strafzahlung entziehen, was wiederum einen deutlichen Unterschied zum anfänglichen Verhalten darstellt, welches in den ersten 2 Quartalen gut erkenntlich ist, durch die nahezu übereinander verlaufenden schwarzen und hellblauen Graphen.

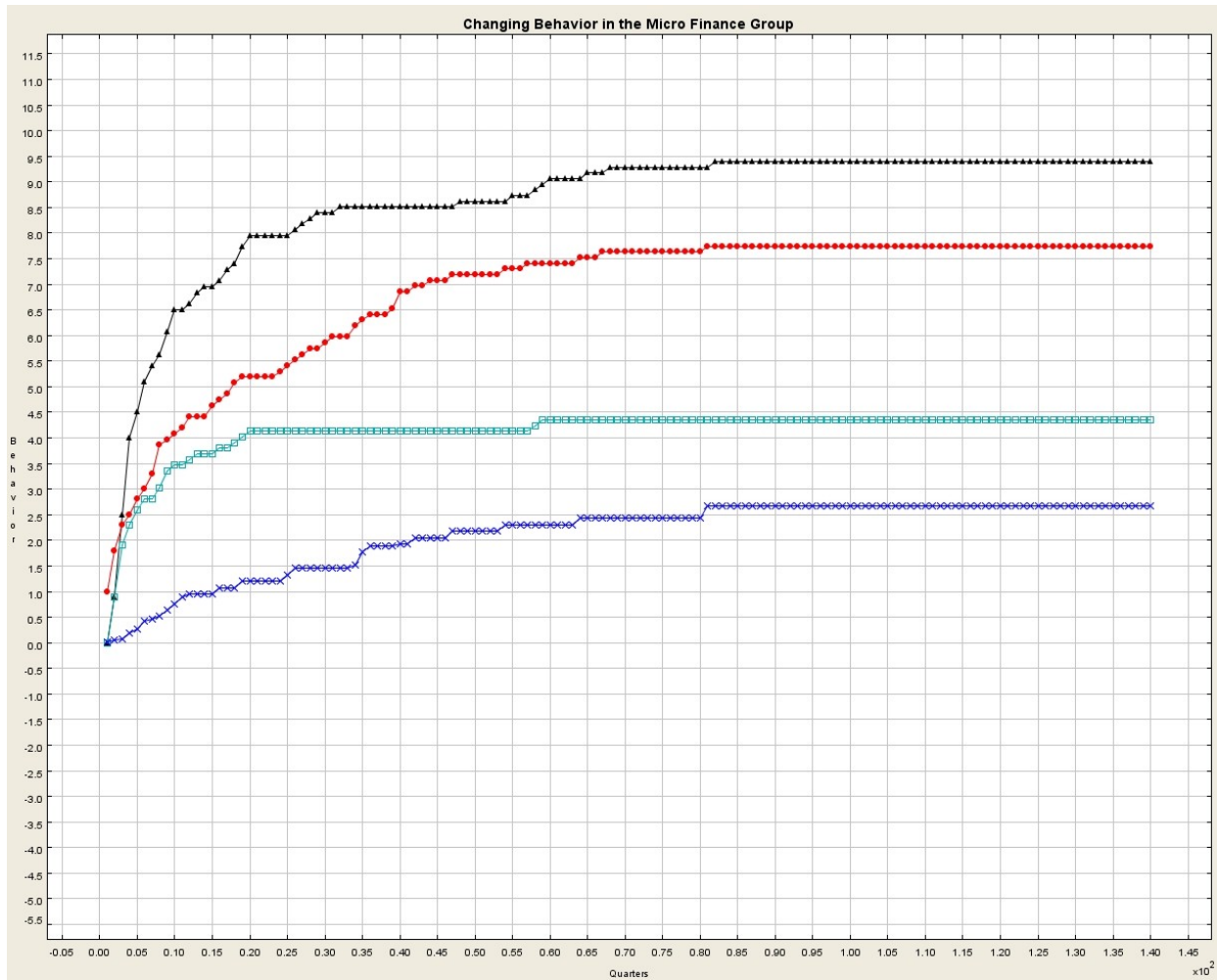


Abbildung 41: Simulationslauf 1

Nach ungefähr 85 Quartalen erreicht die Simulation einen stabilen Zustand.

Zur genaueren Analyse des Lernprozesses stehen die Konsolenausgabe von RePast sowie die Log-Datei von EMIL-S zur Verfügung. Erstere enthält die einprogrammierten Ausgabezeilen und zusätzlich Nachrichten von EMIL-S über erfolgende Sanktionen. Die Log-Datei hält detailliert fest was der Agent zu welchem Zeitpunkt gelernt hat, also wie sich sein Regelsatz daraufhin veränderte. Die Abbildungen 42, 43, 44 und 45 geben einen Einblick. Abbildung 42 ist ein Fragment der Konsolenausgabe und zeigt wie die Gruppenmitglieder auf eine versäumte Ratenzahlung reagieren. Die Mehrheit entschließt sich den Sünder zu bestrafen. Die letzte Zeile ist eine Meldung von EMIL-S, daß eine Bestrafung erfolgt.

```

This is Cycle: 1, Quarter: 1
GroupMember 1 decides to not pay his rate.
GroupMember 2 decides to not fine.
GroupMember 3 decides to not fine.
GroupMember 4 decides to fine
GroupMember 5 decides to not fine.
GroupMember 6 decides to fine
GroupMember 7 decides to not fine.
GroupMember 8 decides to fine
GroupMember 9 decides to fine
GroupMember 10 decides to fine
The Group Decision for fining is: 0.5555555555555556
Sanction for Group Member1 AT TIME: 3

```

Abbildung 42: RePast Konsolenausgabe: Bestrafung eines Nichtzahlers (Quartalsrate)

Durch einen Blick in die Log-Datei kann das zugehörige Fragment durch Vergleich des Zeitstempels gefunden werden. Dies ist in Abbildung 43 gezeigt. Dieses Fragment enthält folgende Informationen:

- Das Lernereignis ist eine explizite Sanktionen
- Die Bestrafungsstärke für den Nichtzahler wurde auf -0.75 festgelegt
- Die zu verändernde Regel wurde als die Regel die auf das Ereignis E2 folgt festgelegt oder erkannt (abhängig davon ob EMIL-S eine konkrete Ursache bereits mitgeliefert wurde)
- Die Ursache wurde zum Zeitpunkt T=1.0 ausgeführt
- Die Aktion A6 wird also als ursächlich festgelegt oder erkannt
- Es wird die alte Eintrittswahrscheinlichkeit für A6 gezeigt und eine neue, abhängig von der Bestrafungsstärke, festgelegt.

```

----- NORM INVOCATION FOR AGENT 1 AT TIME 3.0-----
EVENT: C=Explicit NI M=SANCTION P=-0.75 ENTRY=E2 T=1.0 RULE=null
LEARN ACTION=A6 OLD_PROB=1.0 NEW_PROB=0.625

```

Abbildung 43: EmilS Log-Datei: Bestrafung eines Nichtzahlers (Quartalsrate)

Man erkennt also das der Agent gelernt hat, daß das Ereignis welches auf seine Aktion A6 folgt für ihn unangenehm ist und deshalb die Aktion A6 von ihm weniger oft ausgeführt werden wird.

Abbildung 44 und Abbildung 45 zeigen ähnlich das Lernen für die Entscheidung eine verpasste Ratenzahlung nicht auszugleichen. Der Unterschied ist hier, daß die Sanktion welche die Bank ausspricht gleichzeitig an alle Gruppenmitglieder gesendet wird, die ein Fehlverhalten gezeigt haben.

```

GroupMember 1 decides to not cover
GroupMember 2 decides to not cover
GroupMember 4 decides to cover
GroupMember 5 decides to cover
GroupMember 6 decides to not cover
GroupMember 7 decides to cover
GroupMember 8 decides to not cover
GroupMember 9 decides to not cover
GroupMember 10 decides to cover
The Group Decision for covering is: 0.4444444444444444
e5PaymentIncomplete
The Bank decides to punish the group.
Sanction for Group Member1 AT TIME: 16
Sanction for Group Member2 AT TIME: 16
Sanction for Group Member6 AT TIME: 16
Sanction for Group Member8 AT TIME: 16
Sanction for Group Member9 AT TIME: 16

```

Abbildung 44: RePast Konsolenausgabe: Bestrafung der Gruppe bei unvollständiger Zahlung an die Bank

In Abbildung 45 erkennt man, daß nun eine Veränderung an der Regel die auf E4 folgt vorgenommen wird.

```

----- NORM INVOCATION FOR AGENT 1 AT TIME 16.0-----
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=14.0 RULE=emil.agent.rule.EventActionTree@119e583
LEARN ACTION=A10 OLD_PROB=0.45 NEW_PROB=0.225

----- NORM INVOCATION FOR AGENT 2 AT TIME 16.0-----
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=14.0 RULE=emil.agent.rule.EventActionTree@1250ff2
LEARN ACTION=A10 OLD_PROB=0.45 NEW_PROB=0.225

----- NORM INVOCATION FOR AGENT 6 AT TIME 16.0-----
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=14.0 RULE=emil.agent.rule.EventActionTree@1541147
LEARN ACTION=A10 OLD_PROB=0.225 NEW_PROB=0.1125

----- NORM INVOCATION FOR AGENT 8 AT TIME 16.0-----
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=14.0 RULE=emil.agent.rule.EventActionTree@107108e
LEARN ACTION=A10 OLD_PROB=0.45 NEW_PROB=0.225

----- NORM INVOCATION FOR AGENT 9 AT TIME 16.0-----
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=14.0 RULE=emil.agent.rule.EventActionTree@cfe049
LEARN ACTION=A10 OLD_PROB=0.45 NEW_PROB=0.225

```

Abbildung 45: EmilS Log-Datei: Bestrafung der Gruppe bei unvollständiger Zahlung an die Bank (Fragment)

Bei der Überprüfung des Szenarios mit Durchläufen, können verschieden initiale Einstellungen verändert werden, um den Effekt auf den Simulationsablauf zu testen: So kann zum Beispiel die XML IRB mit verschiedenen Ausgangswahrscheinlichkeiten belegt werden oder die Stärke für explizite Sanktionen verändert werden. Wie sich eine Veränderung der Bestrafungsstärke, die auf das Nichtzahlen der Quartalsrate folgt, auswirken kann, zeigt der nächste Simulationslauf. Im ersten Lauf waren die genannten Stärken wie in Abbildung 46 eingestellt, für die folgenden Läufe wird die Einstellung von Abbildung 47 verwendet.

```
if (groupDecisionFining == 0) {
    strengthOfPunishment = 0;
}
if (0 < groupDecisionFining & groupDecisionFining <= 0.25) {
    strengthOfPunishment = -0.25;
}
if (0.25 < groupDecisionFining & groupDecisionFining <= 0.5) {
    strengthOfPunishment = -0.5;
}
if (0.5 < groupDecisionFining & groupDecisionFining <= 0.75) {
    strengthOfPunishment = -0.75;
}
if (0.75 < groupDecisionFining) {
    strengthOfPunishment = -1.0;
}
```

Abbildung 46: Bestrafungsstärken bei Nichtzahlen der Quartalsrate; Simulationslauf 1

Die neuen Bestrafungsstärken sind etwas höher gewählt als die alten, so daß ein schnelleres Lernen zu erwarten ist.

```
if (groupDecisionFining == 0) {
    strengthOfPunishment = 0;
}
if (0 < groupDecisionFining & groupDecisionFining <= 0.25) {
    strengthOfPunishment = -0.4;
}
if (0.25 < groupDecisionFining & groupDecisionFining <= 0.5) {
    strengthOfPunishment = -0.6;
}
if (0.5 < groupDecisionFining & groupDecisionFining <= 0.75) {
    strengthOfPunishment = -0.8;
}
if (0.75 < groupDecisionFining) {
    strengthOfPunishment = -1.0;
}
```

Abbildung 47: Bestrafungsstärken bei Nichtzahlen der Quartalsrate; Simulationsläufe 2+3

Der Simulationslauf von Abbildung 48 zeigt auch tatsächlich, daß diese kleine Änderung, während alle anderen Variablen gleichbleiben, dazu führt, daß ein stabiler Zustand bereits nach 45 Quartalen erreicht wird, also 40 Quartale früher als im ersten Simulationslauf.

Versuche mit Änderungen der XML IRB haben sich für die Regeln die zur Laufzeit dem Lernen ausgesetzt sind wenig ausgewirkt, da das Lernen die Wahrscheinlichkeiten anfänglich sehr schnell verändern kann und die initiale Einstellung somit nicht stark ins Gewicht fällt.



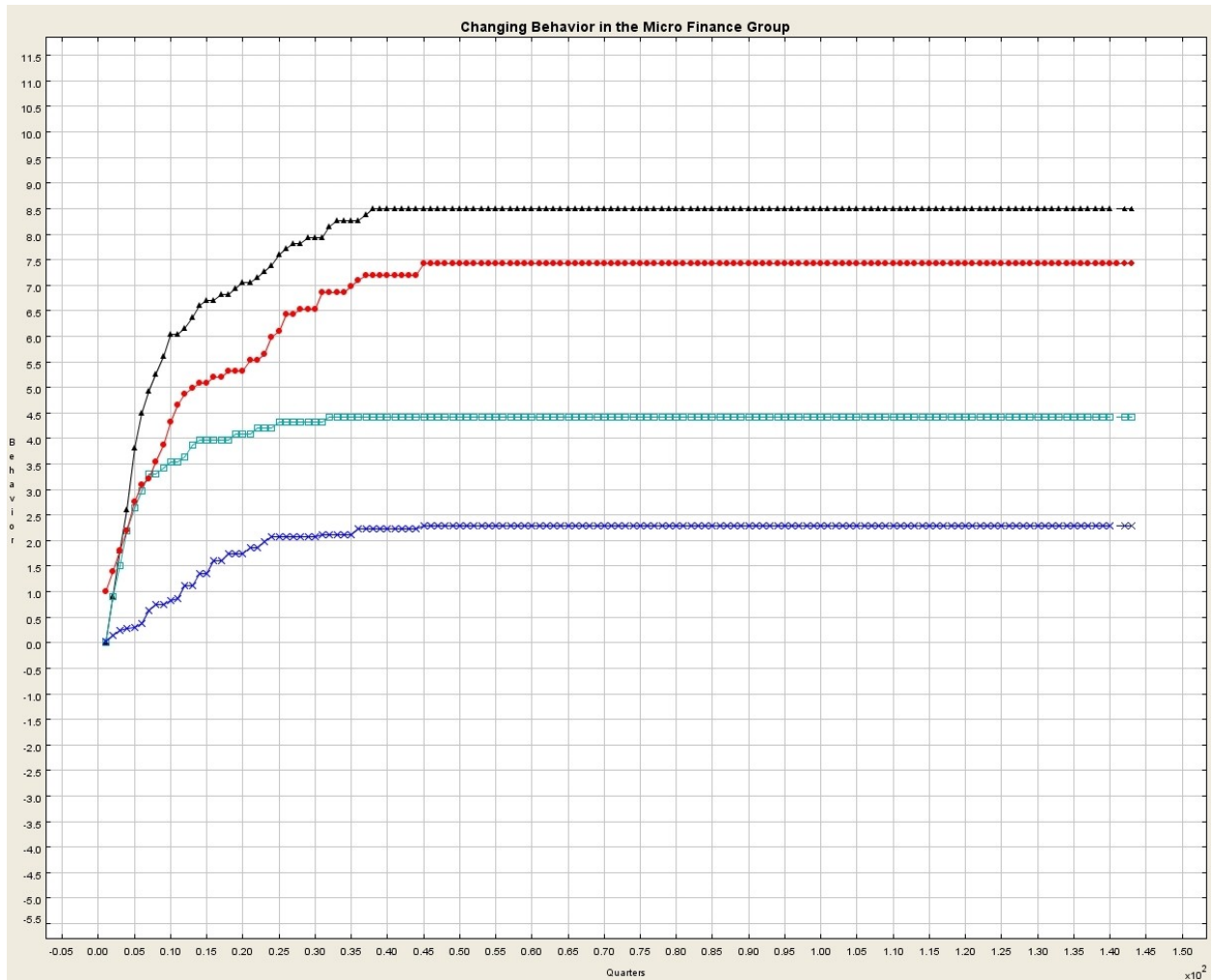


Abbildung 48: Simulationslauf 2

In einem dritten Simulationslauf wird der gelegentlich auftretende Fall untersucht, daß einem sanktionierten EMIL-S Agenten keine explizite Aktion von RePast als die Ursache gemeldet wird. Dieser Fall tritt im vorliegenden Szenario immer dann ein, wenn die Bank zu Beginn eines neuen Zyklus der Gruppe eine generelle Aufmunterung oder Warnung schickt. Diese wird von RePast nur dann mit einer zu sanktionierenden Agentenaktion verknüpft, falls am Ende des letzten Zyklus ein Ereignis E8 stattgefunden hat, welches die Einzelagenten aufforderte darüber zu entscheiden ob man angefallene Rückstände bei der Bank ausgleichen sollte. Waren keine Rückstände vorhanden, wurde also kein Ereignis E8 ausgelöst, wird die genannte Warnung/Aufmunterung der Bank ohne explizite Verknüpfung an die EMIL-S Gruppenagenten gesandt. Nun kann EMIL-S frei entscheiden, wie es die Sanktion beziehen möchte. In der hier verwendeten EMIL-S Version wird dabei die Sanktion auf die letzte eigene Agentenaktion bezogen, so daß der Gruppenagent also meint daß seine letzte Aktion zu einer Reaktion der Bank geführt haben muss. Dies ist selbstverständlich meist eine falsche Annahme des Gruppenagenten, die in der Realität allerdings

durchaus so vorkommen könnte, falls der Einzelagent über keine weitere Erfahrung zu den Gründen einer erhaltenen Sanktion verfügt.

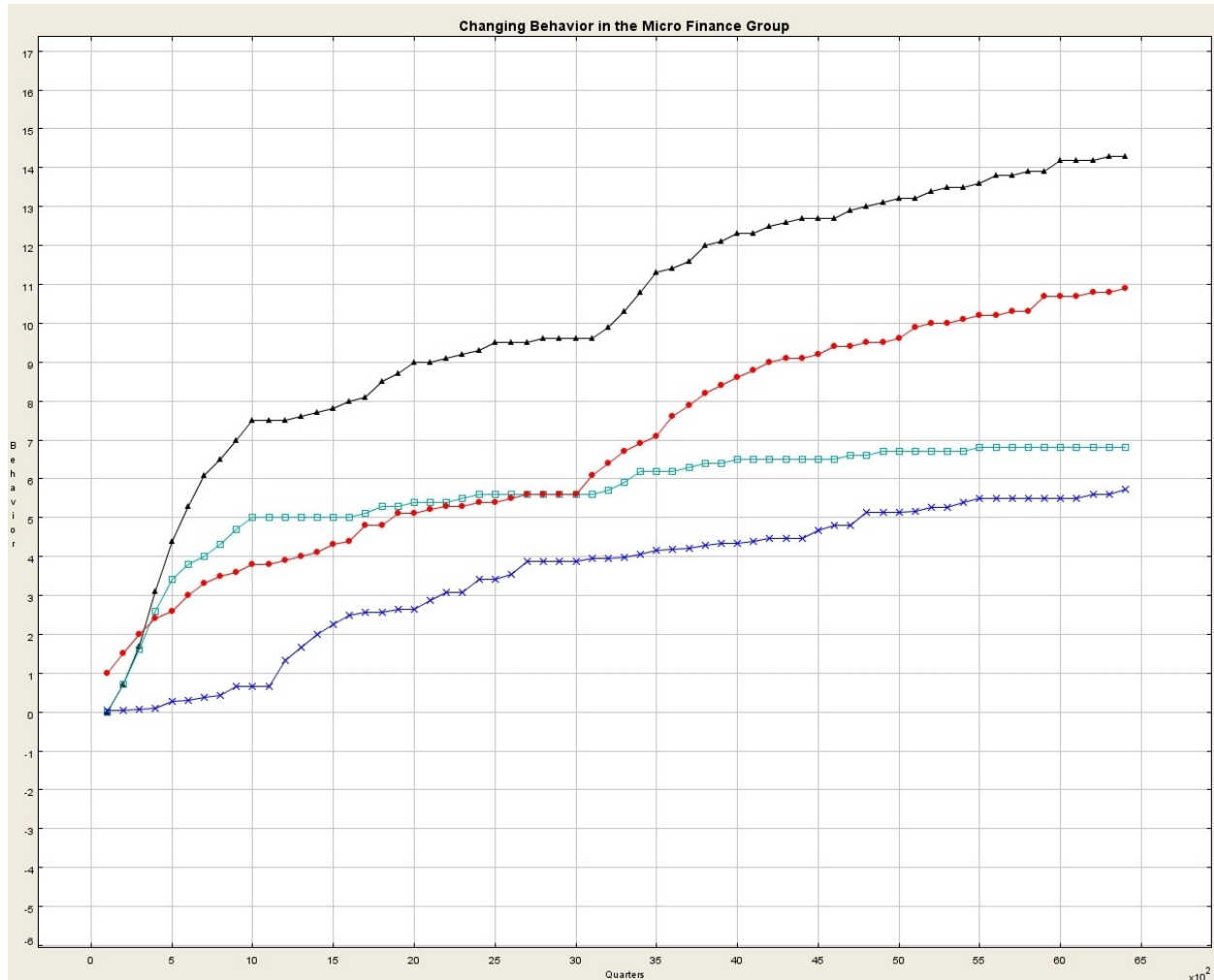


Abbildung 49: Simulationslauf 3

Der Simulationslauf von Abbildung 49 zeigt zwei solcher Missverständnisse, die sich in heftigen Verhaltensänderungen der Agenten äußern: Erstens im zwölften Quartal bei dem dunkelblauen Graphen (NotCoveringGroupFine) und zweitens im dreißigsten Quartal beim roten Graphen (MissingQuarterPayments). Erwähnenswert ist auch, daß die Simulation nach diesen Sanktionen, die das konvergieren gegen einen stabilen Zustand durcheinander bringen, sich langsam erholt und das ursprüngliche Lernverhalten wieder aufnimmt.

Abbildung 50 zeigt die Konsolenausgabe für den Beginn des elften Quartals.

```

This is Cycle: 2
The Bank decides to warn the group.
Sanction for Group Member1 AT TIME: 309
Sanction for Group Member2 AT TIME: 309
Sanction for Group Member3 AT TIME: 309
Sanction for Group Member4 AT TIME: 309
Sanction for Group Member5 AT TIME: 309
Sanction for Group Member6 AT TIME: 309
Sanction for Group Member7 AT TIME: 309
Sanction for Group Member8 AT TIME: 309
Sanction for Group Member9 AT TIME: 309
Sanction for Group Member10 AT TIME: 309
The Bank decides to accept a new cycle.

```

*Abbildung 50: RePast Konsolenausgabe: Sanktion durch die Bank für alle Gruppenmitglieder, Sanktionsziel durch EmilS gewählt (Beispiel 1)*

Die Bank sendet eine unverknüpfte Warnung, die, wie man sich in Abbildung 51 überzeugen kann, von EMIL-S selbsttätig auf den Regelbaum der auf das Ereignis E4 folgt bezogen wird. EMIL-S setzt die Wahrscheinlichkeit für das Eintreten von A9 (Covering Rate) stark herab, so daß die Gruppe als Folge eher für ein Nichtausgleichen von fehlenden Quartalsraten votieren wird.

```

----- NORM INVOCATION FOR AGENT 1 AT TIME 309.0-----
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=null RULE=emil.agent.rule.EventActionTree@b05236
LEARN ACTION=A9 OLD_PROB=0.775 NEW_PROB=0.3875

----- NORM INVOCATION FOR AGENT 2 AT TIME 309.0-----
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=null RULE=emil.agent.rule.EventActionTree@864e43
LEARN ACTION=A9 OLD_PROB=0.8875 NEW_PROB=0.44375

----- NORM INVOCATION FOR AGENT 3 AT TIME 309.0-----
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=null RULE=emil.agent.rule.EventActionTree@17c2891
LEARN ACTION=A9 OLD_PROB=0.55 NEW_PROB=0.275

```

*Abbildung 51: EmilS Log-Datei: Sanktion durch die Bank für alle Gruppenmitglieder, Sanktionsziel durch EmilS gewählt (Beispiel 1)*

Im zweiten Beispiel zu Beginn des 4. Zyklus (Abbildung 52) wird die Sanktion auf den Regelbaum bezogen, der auf das Ereignis E2 folgt.

```

This is Cycle: 4
The Bank decides to warn the group.
Sanction for Group Member1 AT TIME: 429
Sanction for Group Member2 AT TIME: 429
Sanction for Group Member3 AT TIME: 429
Sanction for Group Member4 AT TIME: 429
Sanction for Group Member5 AT TIME: 429
Sanction for Group Member6 AT TIME: 429
Sanction for Group Member7 AT TIME: 429
Sanction for Group Member8 AT TIME: 429
Sanction for Group Member9 AT TIME: 429
Sanction for Group Member10 AT TIME: 429
The Bank decides to accept a new cycle.

```

*Abbildung 52: RePast Konsolenausgabe: Sanktion durch die Bank für alle Gruppenmitglieder, Sanktionsziel durch EmilS gewählt (Beispiel 2)*

Die Wahrscheinlichkeit für das ordnungsgemäße bezahlen der Quartalsrate wird hier (Abbildung 53) stark vermindert.

```
----- NORM INVOCATION FOR AGENT 1 AT TIME 429.0-----  
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E2 T=null RULE=emil.agent.rule.EventActionTree@a09e41  
LEARN ACTION=A5 OLD_PROB=0.984124 NEW_PROB=0.492062
```

```
----- NORM INVOCATION FOR AGENT 2 AT TIME 429.0-----  
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E2 T=null RULE=emil.agent.rule.EventActionTree@f5d030  
LEARN ACTION=A5 OLD_PROB=0.9470799999999998 NEW_PROB=0.4735399999999999
```

```
----- NORM INVOCATION FOR AGENT 3 AT TIME 429.0-----  
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E2 T=null RULE=emil.agent.rule.EventActionTree@37165f  
LEARN ACTION=A5 OLD_PROB=0.9558999999999999 NEW_PROB=0.47794999999999993
```

*Abbildung 53: EmilS Log-Datei: Sanktion durch die Bank für alle Gruppenmitglieder, Sanktionsziel durch EmilS gewählt (Beispiel 2)*

Der rote Graph in Abbildung 49 zeigt auch tatsächlich ein steiles Ansteigen der Fehlraten zu Beginn des 31. Quartals (4. Zyklus).

## 4 Zusammenfassung und Ausblick

Wie sich gezeigt hat verfügt die hier vorgestellte Simulation mit EMIL-S über ein explizites und ein nicht explizites Lernverhalten. Das nicht explizite, von EMIL-S gesteuerte Lernen ist allerdings zum Zeitpunkt der Fertigstellung dieser Arbeit noch wenig ausgeprägt und führt bislang zu missverständlichen Ergebnissen. Das explizite Lernen dagegen, welches in dieser Simulation breit verwendet wurde, nimmt EMIL-S die Freiheiten die es in einer späteren Version braucht. Allerdings war zu diesem Zeitpunkt eine sinnvolle Simulation nur durch explizite Zuweisungen möglich.

Die Erkenntnisse über die Möglichkeiten des EMIL-S Ansatzes sind daher in diesem Stadium eher gering. Vor allem zeigt das Szenario in seiner jetzigen Form folgendes:

- Die Schnittstelle von EMIL-S zu JAVA/RePast, also der Nachrichtenaustausch zwischen Simulationsumgebung und den "denkenden" Agenten funktioniert
- JAVA/RePast ist eine geeignete Plattform um die Simulationsumgebung zu modellieren
- Eigenständig von EMIL-S gesteuertes Lernen führt anscheinend zu verlängerten Simulationsläufen, bis ein stabiler Zustand erreicht wird. Dies ergibt sich aus dem "Trial by Error"- Prinzip
- Explizit in RePast festgelegte Lernzusammenhänge können nicht durch EMIL-S geändert werden, sind also von der Normevolution ausgeschlossen
- Gruppenentscheidungen in der Simulationsumgebung vermindern die Lernfähigkeit des Einzelagenten, durch die nivellierende Natur einer Mehrheitsentscheidung
- Eine starre 1:1 Umsetzung eines vorgegebenen Sachverhalts, wie der Micro Finance Erhebungen von Pablo Lucas, ist mit der derzeitigen EMIL-S Version schwer zu realisieren, da Zusammenhänge zur Zeit noch einer expliziten Zuordnung bedürfen. Das bedeutet allerdings nicht das eine Umsetzung unmöglich erscheint, vielmehr kann diese Arbeit dazu noch keine Aussage treffen. Eine Anpassung der RePast Modellierung sowie der XML IRB, an die zukünftig erweiterten Fähigkeiten von EMIL-S, wird dann allerdings zwingend notwendig werden.

Der Szenariocode ist durch eine JavaDoc Datei dokumentiert und auf der Internetadresse [http://userpages.uni-koblenz.de/~emil/code\\_examples/](http://userpages.uni-koblenz.de/~emil/code_examples/) einsehbar. Anpassung im Laufe des Fortschritts des EMIL-S Lernverhaltens sollten somit sowohl vom Autor als auch von weiteren Personen möglich sein. Das Szenario sollte deshalb eine Grundlage bieten für weitere

Untersuchungen.

Bei der weiteren Modellierung sollte also darauf geachtet werden, explizite Sanktion nur noch äußerst sparsam zu verwenden, da sie EMIL-S zu einer Nebenrolle degradieren. Das durch EMIL-S gesteuerte Lernen muss bei der Entwicklung neuer Szenarien im Vordergrund stehen. Das Verständnis dafür ist, nach Ansicht des Autors, wichtigste Voraussetzung für zukünftig erfolgreiche Modellierungen. Das allerdings starre (RePast) und flexible (EMIL-S) Lernkomponenten durchaus zusammenarbeiten können, hat das Szenario gezeigt.

Die Benutzung von weiteren Nachrichtentypen sowie Zuweisung von Agentenrollen sollte dem Modellierer interessante neue Möglichkeiten bieten.

Das Micro Finance Szenario muss unter Benutzung dieser neu hinzukommenden Möglichkeiten überarbeitet werden.

## Literaturverzeichnis

[AND.07] : Andrighetto, G., Conte, R., Turrini, P., Paolucci, M.; Emergence In the Loop: Simulating the two way dynamics of norm innovation.; Dagstuhl Seminar on Normative Multiagent Systems, Dagstuhl, Germany (2007); ; <http://drops.dagstuhl.de/opus/volltexte/2007/907/>; ;

[CAM.08] : Campenni, M., Andrighetto, G., Cecconi, F., Rosaria Conte; Normal = Normative? The Role of Intelligent Agents in Norm Innovation. ; The Fifth Conference of the European Social Simulation Association (ESSA. University of Brescia, September 1-5, 2008; ; ;

[D33.08] : EMIL Projekt; Deliverable 3.3; EMIL Projekt; ; [http://emil.istc.cnr.it/file\\_download/7/D3.3.Project033841.EMIL.pdf](http://emil.istc.cnr.it/file_download/7/D3.3.Project033841.EMIL.pdf); 31.08.2008; 06.07.2009

[EPS.09] : EMIL Projekt; Startseite; ; ; <http://emil.istc.cnr.it/>; ; 06.07.2009

[GIG.08] : Lena Giesbert; „Magic Microfinance“ – bald auch eine Erfolgsgeschichte für Afrika?; German Institute of Global and Area Studies; Focus; [http://www.giga-hamburg.de/dl/download.php?d=/content/publikationen/pdf/gf\\_afrika\\_0809.pdf](http://www.giga-hamburg.de/dl/download.php?d=/content/publikationen/pdf/gf_afrika_0809.pdf); 2008; 06.07.2009

[PLC.09] : Pablo Lucas; Conventional Social Behaviouramong Microfinance Clients; ; ; ; 01/2009;

[PLR.08] : Pablo Lucas; Dynamics of social conventions in micro finance groups: an experience in chiapas, Mexico; Manchester Metropolitan University, Centre for Policy Modelling; CPM Report No.: 08-192; <http://cfpm.org/cpmrep192.html>; ; Feb.2008

[PSO.09] : EMIL Projekt; Objectives; ; ; <http://emil.istc.cnr.it/objectives/>; NA; 06.07.2009

[REP.09] : ; RePast 3 Projektseite; ; ; [http://repast.sourceforge.net/repast\\_3/index.html](http://repast.sourceforge.net/repast_3/index.html); ; 07.2009

[RPS.03] : John Weiss, Heather Montgomery and Elvira Kurmanalieva; Micro Finance and Poverty Reduction in Asia: What is the Evidence?; ; ; <http://www.adbi.org/files/2003.rps53.microfinance.poverty.pdf>; ; 06.07.2009

[SNB.09] : Möhring, Michael; Lotzmann, Ulf ; Simulating Normative Behaviour and Norm Formation Processes; Otamendi, Javier; Bargiela, Andrzej; Montes, Jose Luis; Pedrera, Luis Miguel Doncel: 23nd European Conference on Modelling and Simulation, June 9-12, 2009. Madrid; S. 187-193.; ; 2009;

[SNF.08] : Lotzmann, Ulf; Möhring, Michael; Troitzsch, Klaus G.; Simulating Norm Formation in a Traffic Scenario.; Squazzoni, Flaminio: The Fifth Conference of the European Social Simulation Association, September 1-5, 2008. Brescia; ; ;

[SSS.05] : Nigel Gilbert, Klaus G. Troitzsch; Simulation for the Social Scientist, 2nd edition; Open University Press; ; ; 2005;