SIXTH FRAMEWORK PROGRAMME

PRIORITY 2

Information Society Technologies

Project n°033841

EMIL

EMergence In the Loop: simulating the two way dynamics of norm innovation

Deliverable 5.1

EMIL-T

Due date of the Deliverable: 31 Oct 09 + 45 dd.

Actual Submission date: 15 Jan 10

Start date of the Project: 01 Sept 2006 Duration: 38 months

Organization responsible for this Deliverable: MMU

Dissemination Level: PU

The EMIL Consortium and Report Authors

1.1	***ISTC-CNR, Laboratory on Agent-Based Social Simulation, Italy***	Rosaria Conte Giulia Andrighetto Marco Campennì
	Universität Koblenz-Landau, Germany	Klaus G. Troitzsch Ulf Lotzmann Iris Lorscheid Michael Möhring Jens Villard, Robin Emde, Manuel Pauli, Steffi Henn, Peyman Jazayeri, Magnus Oberhausen, Mehmet-Hadi Tohum, Jannik Weyrich
	University of Surrey, Centre for Research on Social Simulation, United Kingdom	Maria Xenitidou
	University of Bayreuth, Dept. of Philosophy, Germany	Rainer Hegselmann Martin Neumann Oliver Will
	AITIA International Informatics Inc. A, Hungary	László Gulyás Attila Szabó
	Manchester Metropolitan University, Centre for Policy Modelling, United Kingdom	Bruce Edmonds Pablo Lucas dos Anjos

Table of Contents

The EMIL Consortium and Report Authors 2

Table of Contents 3

Aims and Outline 4

Chapter 1 Introduction 5

Chapter 2 Theoretical Foundations: The Revised EMIL Ontology 9

Chapter 3 Theoretical Foundations: Two Way Dynamics of Social Processes 15

Chapter 4 Theoretical Foundations: Normative Behaviour and Economic Wealth (The Development of Economic Wealth Based on Trust in Large Groups of People) 23

Chapter 5 Empirical Context: Studying Emergent Social Order in Interaction: The Case of Wikipedia 33

Chapter 6 Relationship to Other Thought: State of the Art in Modelling Norm Emergence 49

Chapter 7 Relationship to Other Thought: The BDI Approach to Norm Immergence 57

Chapter 8 Theory Essentials: The Added Value in Normative Agents 65

Chapter 9 Making the Theory Explicit: The EMIL-A Architecture 77

Chapter 10 Supporting the Theory: The Derivation of EMIL-S from EMIL-A (From Logical Architecture to Software Architecture) 89

Chapter 11 Supporting the Theory: Formal Description of EMIL-S 93

Chapter 12 Supporting Theory Use: The MEME Simulation Methodology and Tools 109

Chapter 13 Demonstrating the Theory: An Introduction to the Scenarios 135

Chapter 14 Demonstrating the Theory 1: Traffic 139

Chapter 15 Demonstrating the Theory 2: Wikipedia 149

Chapter 16 Demonstrating the Theory 3: Micro Finance 169

Chapter 17 Demonstrating the Theory 4: Multi-scenario World – the Sequel 191

Chapter 18 Demonstrating the Theory: Conclusion of Scenario Results 201

Chapter 19 Summary of Major Advances in EMIL-A and EMIL-S 203

Chapter 20 Looking Forward: On Norm Internalization 213

Chapter 21 Looking Forward: A Mini-road-map of Questions about Normative Behaviour 221

Chapter 22 Epilogue 227

References 231

Aims and Outline

This deliverable was described as follows in the Annex "Description of Work".

> *EMIL-T:*
>
> *WP5, is aimed at checking and formulating the theoretical advances obtained, comparing them with the criteria put forward in WP1, with the simulation results (WP4) and the application examples examined (WP2).*
>
> *After having constructed a theory of norm innovation at the social and cognitive levels, described an empirical example of norm innovation (WP1), the development of norms in the open-source movement (WP2) and applied a computational version of the theory to the empirical example by building and executing a simulation (WP3-4), EMIL-T shall evaluate the success of the theory in understanding the development of norm innovation in the open source movement by comparing the results of the simulation with the empirical data documented in the open-source scenario (D5.1, project month 36). The comparison will lead to a revision and improvement of the theory.*
>
> *The output of WP5 will be a documented restatement of theory, in the form of scientific papers and a monograph consisting of contributions from members of the project (M5.1, project month 33; D5.2, project month 36). The results will be presented in the main conferences and workshops of the field, in tight collaboration with the diffusion group (M5.2, project month 36).*
>
> *EMIL-T MAIN DELIVERABLES DESCRIPTION:*
>
> *D5.1: EMIL-T: a final theoretical model including revised sociocognitive model of norm innovation (project month 36).*
>
> *D5.2: Monograph: the reformulated theory will be documented as part of a monograph describing the results of the project (i.e. the theoretical approach, the theory in detail, the empirical example, the fit of the theory to the example, and the application of the theory to other empirical phenomena) (project month 36).*

This document is a summary of D5.1, the final model resulting from the EMIL project – an ambitious sociocognitive model of norm innovation. This is arises from a dynamic and complex view of norms, it is based on and has shed new light on the ontology developed in EMIL-M, was specified in EMIL-A, has been implemented in the architecture EMIL-S; has been used to develop a suite of simulations of normative phenomena and will be available as a theory, architecture and tool for future work by academics. This is the first theory of norms that comprehensively relates ontology, theory, implementation, architecture – integrating both cognitive and social aspects of norms as they co-evolve as a result of interacting, emergent and immergent processes.

Chapter 1 Introduction

Rosaria Conte and Bruce Edmonds

1.1 Questions

What are norms? What are the differences and commonalities among social, moral, and legal norms? How do norms emerge and change? Why and how do people abide with or violate them? Should we differentiate norms from the most frequent or normal conduct, on one hand, and from coerced behaviour on the other? And if we should, which mechanisms or factors should we call into play? These are the questions the EMIL project was focussed upon. These are all questions addressed within the EMIL project.

Social scientists often view norms as regular behaviours, possibly enforced by social expectations and sanctions, seeing no reason for a specific, norm-related form of cognition. On the other hand, philosophers of law and logicians conceptualise norms as expressions of the authority's will. The former see norms as regular behaviours, the latter as issued obligations.

The present manuscript presents a summary of the main scientific contribution of the EMIL project: a dynamic, computational and cognitive theory of norms, labelled "EMIL-T". EMIL-T is mainly aimed at accounting for the emergence and innovation of social norms, but it also provides tentative answers to some questions enlisted above, especially (a) what might be common to different kinds of norms; (b) how to differentiate norm-based behaviour (any type of norms) from coerced behaviour, on one hand, and regular behaviour, on the other; (c) what are the principal mechanisms allowing intelligent autonomous agents to comply with norms. EMIL-T is characterized by a two-way approach to social dynamics, in which bottom-up (in particular, emergent) and top-down (in particular, immergent) processes are seen as strictly intertwined and accounted for.

EMIL-T adopted a simulation-based methodology as both a theory-building and a theory-testing approach. A number of simulations studies reported on in the manuscript were initially carried out for exploratory purposes, but many of them were later replicated for validation. In accordance with a cross-methodological approach, an empirical study was conducted - and is reported upon - within a natural domain (actual Wikipedia), which was later reproduced in a simulation study of collaborative writing.

Hence, the manuscript includes:

i. a review of different approaches to the simulation of norm emergence (cfr. Chapter 6)
ii. an inventory of the main concepts upon which the project has been built (cfr. Chapter 2)
iii. reviewed versions of the previous deliverables, EMIL-M, EMIL-A, and EMIL-S, i.e.
    1. a report of the model of the 2-way dynamics of social processes, applicable to norm innovation (cfr. Chapter 3)
    2. a description of the normative agent architecture (cfr. Chapter 9)
    3. a description of the simulation platform for carrying out artificial experiments on the dynamics of social norms (cfr. Chapter 11)
    4. a description of the MEME platform allowing to conduct extensive analyses of the simulations (cfr. Chapter 12)
    5. an introduction to the scenarios chosen for simulation studies (cfr. Chapter 13)
    6. a set of reports on the results of empirical and simulation studies carried out in different scenarios, in particular,
        1. Hume Model, showing conditions under which positive social action can emerge without full-fledged norms coming into existence
        2. Micro-finance, illustrating the conditions under which a collective positive behaviour, namely compensating for members' failures, emerges in groups of loaners (cfr. Chapter 16).
        3. Multi-Scenario World, investigating the conditions under which normative agents are

needed for full-fledged norms to emerge (cfr. Chapter 17).

4. Wikipedia, simulating the emergence of full-fledged norms of neutral style in natural and simulated collaborative filtering (cfr. Chapter 15).
5. Traffic, showing the emergence of complementary norms from car-drivers and pedestrians interacting in a simulated environment (cfr. Chapter 14).

7. A set of replications of the simulation studies on different platforms and in different languages (MatLab, NetLogo, EMIL-S).

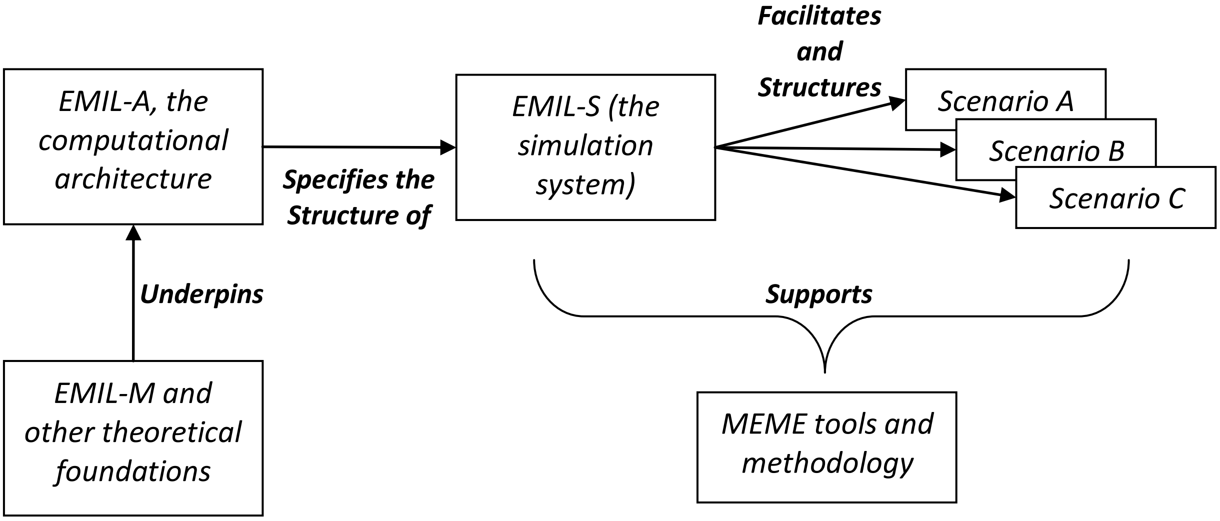

EMIL-A, the computational architecture
Specifies the Structure of
EMIL-S (the simulation system)
Facilitates and Structures
Scenario A
Scenario B
Scenario C
Underpins
Supports
EMIL-M and other theoretical foundations
MEME tools and methodology

Figure 1. An Illustration showing the relationship of the parts of EMIL-T framework

1.2 Assumptions and Claims

The main thesis underlying this manuscript is that observable conformity is only the tip of the normative iceberg. The crucial dynamics takes place beneath the line of observation, in the minds of the agents. We deem the process by which norms get converted into mental representations (normative beliefs and goals) and operations (decision, reasoning, etc.) as immergent, in analogy with the complementary process of norm emergence, by which norms spreading through the behaviours of agents become observable.

The process of immergence allows us to answer some of the initial questions. Norm immergence is a process common to both social and legal norms, which means that norm-based behaviour needs to be kept distinct from imitation on the one hand, and acquiescence under menace on the other. The key difference is that agents abiding with norms, or violating them, act on a set of mental representations.

EMIL-T proposes an integrated view of norms, in which the complex, bidirectional dynamics between immergence and emergence is tested on a simulation platform against a number of social scenarios.

To be noted, not all of the scenarios we have simulated actually necessitate norm immergence, in the sense defined above. Indeed, not all virtuous, or prosocial behaviour, is based upon full-fledged norms. Other mechanisms may lead agents to perform in a positive or virtuous way. For example, larger and wealthier markets are shown (cf. Chapter 4) to emerge from among non-familiar agents endowed with heterogeneous competencies. Analogously, collective dependence among group members favours collective responsibility (cf. Chapter 3). Hence, a question arises: which observed social phenomena require norm immergence?

In other simulated environments, we found at least a partial answer to the last question. It is possible to show (cf. Chapter 8) that under given conditions, for example in a multi-scenario world, behaviours will not converge unless agents are endowed with the capacity to form autonomous normative beliefs while observing others' behaviours. Imitation per se is not sufficient. More precisely, our simulations show that under given social conditions, namely in multiscenario worlds, norms operate in society while operating in the mind, neither after nor before. In those circumstances, the two components of norm dynamics –

emergence and immergence – need to be strictly intertwined, such that one cannot occur without the other. In multiscenario worlds, agents do something more than simply imitate one another; they put pressure on one another to act accordingly. As Mary Douglas (1996) efficaciously put it, they “squeeze one another into the same practices”. This is possible because social agents are as autonomous as socially responsive. They are autonomous in that they act on their own beliefs and goals. However, they are also responsive to their environment, and to the inputs they receive from it, especially to social inputs.

What type of norms can we account for by this means? Any, indeed. We chose different scenarios for our simulations - Wikipedia (Chapter 15), street Traffic (Chapter 14), Micro Finance (Chapter 16). Thanks to the simulation platform – EMIL-S – built on top of our normative agent architecture – EMIL-A, results obtained in each of these studies closely resemble phenomena observed in natural societies. In the simulated environments, regularities emerge as solutions to different social problems, such as useful and reliable collaborative writing, spatial coordination among pedestrians and car drivers, maintenance of financial circuits under critical conditions. In each of the simulated scenario, solutions critically depend on agents’ learning and their capacity to form own beliefs, whether right or wrong, shared or not shared, and accomplish “norm-invocation”, urging others to accomplish with the norms, on the grounds of these beliefs, thereby spreading around normative commands.

The conceptual and theoretical foundations that our work is based upon and the methodological computational instruments that made it possible are presented in the specific contributions collected within this manuscript. The main contributions will be preceded by a discussion of the state of the art in the treatment of norms.

As we shall see the conceptual framework has been transformed into a theoretical architecture and this into a simulation framework. The simulation framework will be described in detail and used for simulating the emergence of norms in the different scenarios.

Chapter 2 Theoretical Foundations: The Revised EMIL Ontology

Giulia Andrighetto and Rosaria Conte

2.1 Introduction

Since norm innovation results from a complex collection of nested theoretical definitions, it is necessary to provide a shared ontology, or in other words, to forge a working vocabulary of interrelated notions. By ontology, we mean a conventional and operational tool, a set of theoretical notions that are defined one in relation to the others. Its goal is to make conceptual links among normative notions explicit.

The definitions included in this ontology have been informed by the projects outcomes. For example, the notions of emergence and immergence have been modified and refined on the basis of the computational models realized, which have allowed us to describe in more detail all the steps of these social dynamics.

To have a first impression of the ontology, we will now have a brief overview of the main concepts.

Below, the reader is offered some guidelines for understanding the rationale of this ontology.

2.2 Rationale

The purpose of the present ontology obviously derives from the objective of the EMIL project, which is aimed at delivering a simulation-based theory of norm-innovation/emergence. By norm-innovation/ emergence we mean a complex loop, in which the emergent effect determines new properties on the producing level, by means of which the effect is reproduced. A recursive interaction between both levels is established in a complex feedback loop. This includes two sub- processes:

- **Emergence,** i.e. the process by means of which macro effects are generated by (inter)acting micro(social) entities, and implemented upon (for a detailed discussion see Conte et al. 2007).
- **Immergence**, i.e. the gradual and complex process by which the macro-social effect, in our case a specific norm, impacts the minds of the agents, generating a number of intermediate loops. Before any global effect emerges, specific local events affect the generating systems, their beliefs and goals, in such a way that agents force one another into converging on one global macroscopic effect (Castelfranchi, 1998; Conte et al., 2007; Andrighetto et al., 2007a, b).

The emergence of social norms is a major circuit made of local loops, in which:

- partial or initial observable macroscopic effects of local behaviours occur
- retroact on (a subset of) the observers' minds, modifying them (producing new internal states, emotions, normative beliefs, normative goals, etc.)
- agents communicate internal states to one another, thus activating a process of normative influencing (see Conte and Dignum, 2001)
- these normative beliefs spread through agents' minds
- behaviours progressively conform to spreading states
- initial macroscopic effects get reinforced/weakened depending on the type of mental states spreading.

Hence, we need norm-related notions that are

- **Dynamic** to be compatible with a simulation-based investigation. Attention will be paid to modifications rather than typologies of norms and their functions.
- **Innovation-oriented**, this is a special case of dynamics. By innovation, we mean a process designed or wanted by institutional or social agencies (if only an opinion movement). A merely conventionalist view of norms, a spontaneous and emergent dynamics, are insufficient to account for this process: rather than waiting for new regularities to emerge, agencies aim to impose new obligations or rights, new permissions or forbiddances. In a word, new norms.
- **Hybrid**, incorporated both in social and mental objects. In this perspective, the following views are

deemed to be inadequate:

1. **Epiphenomenal**, according to which norms are but observable social patterns, interpreted "as if" they resulted from any normative force or process. On the contrary, we are interested into social patterns *actually* resulting from the action of norms in society,
2. **Behaviorist**, characterizing norms as observable regularities resulting from agents' squeezing each other into common practices. Conversely, we start from the assumption that it is important to look at what happens in the mind of agents in order to understand how norms operate,
3. **Conventionalist**, in which norms are seen as conventions. Although necessary, this is still an insufficient view of norms, especially when we want to deal with innovation.

2.2.1 Command

A command is a coercive request of action, based upon the commander's (pretended) power over the recipient.

2.2.2 Norm

We consider a norm[1] – be it social, legal or moral – as *"a prescribed guide for conduct which is generally complied with by the members of society"* **(**Ullmann-Margalit, 1977). A norm spreads trough a population thanks to the diffusion of a particular shared belief, i.e. the normative-belief. A normative belief, in turn, is a belief that a given behavior, in a given context, for a given set of agents, is either forbidden, obligatory, permitted, etc (Wright, 1963; Kelsen, 1979; Conte and Castelfranchi, 1999, 2006). Stated differently, a normative belief is a belief that there is a command based on a deontic[2]. Of course, a normative belief does not imply that a given norm has in fact been deliberately issued by some institutional authority. Social norms are often set up by virtue of unwanted effects. However, once emerged, a given social norm is believed to be based upon some normative authority, if only an anonymous and impersonal one ("You are wanted, expected (not) to do this…": "It is generally expected that…"; "This is how things are done…", etc.).

It has to be pointed out that a norm is a prescription that is *requested* to be adopted because it is a norm and is *fully applied* only when it is complied with for its own sake (although this "felicity condition" rarely applies *de facto*). Even normative commands are often adopted under the effect of reinforcement. Nonetheless, this type of adoption is not satisfactory, so to speak, from the norm's point of view, if any such a perspective can ever be hypothesized. The *happiness condition* is that the norm is accepted, to say it in Hart's terms (1968), or internalized, to state it in Durkheim's terms (1897/1951), because it is recognized as a norm. In other words, in order for the norm to be satisfied, it is not sufficient that the prescribed action is performed, but it is necessary to comply with the norm because of the normative goal, that is, the goal deriving from the recognition and subsequent adoption of the norm.

Thus, for a norm-based behavior to take place, a normative belief has to be generated into the minds of the norm addressees, and the corresponding normative goal has to be formed and pursued. In this sense, norm emergence and stabilization implies its immergence (Castelfranchi, 1998a; Andrighetto et al., 2007a, b; Conte et al., 2007) into the agents' minds.

[1] A lively debate on the concept of norm has been developed in several branches of Philosophy, Logic, Cognitive Science, Theory of Agents, Social Theory and Game Theory. Here are few references. Social and Legal Philosophy, Raz, 1975; Kelsen, 1979; Logic of Action and Deontic Logic, Horty, 2001; Jones and Sergot, 1993; Wright, 1963; Cognitive Science and Theory of Agents, Conte and Castelfranchi, 1995, 2006; Castelfranchi and Conte, 1999; Social Theory and Game Theory, Bicchieri, 1990, 2006; Coleman, 1990; Young, 1998, 2006; Ullman-Margalit, 1977; Social Simulation, Axelrod, 1984; Macy and Skvoretz, 1998; Macy and Sato, 2002; Sen and Airiau, 2007.

[2] Although necessary for the spreading of the prescribed behaviour, the normative command is insufficient: additional factors consist of the mandatory force (obligatoriness and enforcement) of the command; the persuasiveness and credibility of the source; compatibility with existing norms (norm conflicts often lead to violating one or the other); etc.

Below follow some components of the mental processing of norms:

- ***Normative belief***[3], the belief that a given behaviour in a given context for a given set of agents is forbidden, obligatory, permitted, etc. More precisely, the belief should be that "there is a norm prohibiting, prescribing, permitting that..." (Wright, 1963; Kelsen, 1979; Conte and Castelfranchi, 1999, 2006). Indeed, norms are aimed at and issued for generating the corresponding beliefs. In other words, norms must be acknowledged as such in order to properly work.
- ***Normative belief of pertinence*** Believing that a norm exists and concerns us requires at least a second group of beliefs: the beliefs of pertinence. The norm says what ought to be done by whom: (i) the obligation/permission/prohibition and (ii) the set of agents on which the imperative is impinging. For example, if I am addressed by a given norm (say, "be member of a professional order"), and the norm has to take effect on me, I must recognize this. The prescription is about a set or class of agents, and since I am an instance of that class, the norm applies to me.
- ***Normative goal***[4], an internal goal relativized to a normative belief. From a cognitive point of view, goals are internal representations triggering-and-guiding action at once: they represent the state of the world that agents want to reach by means of action and that they monitor while executing the action (see Conte, 2009). A goal is relativized when it is held because and to the extent that a given world-state or event is hold to be true or is expected (Cohen and Levesque, 1990)[5].
- ***Norm adoption***, the formation of a normative goal from a normative belief, thanks to some intervening rules (see Chapter EMIL-A: The Architecture, for a detailed description). For example, a normative goal of a given agent *x* about action *a* is a goal that agent *x* happens to have as long as she has a normative belief about *a*. More specifically, *x* has a normative goal only if she believes to be subject to a norm.
- ***Normative equity principle***, agents want their normative costs to be no higher than those of other agents subject to the same norm.
- ***Normative reasoning,*** mental operation upon the internal representation of a given norm, which may lead to that norm being adopted, thereby forming a normative goal.

2.2.3 Convention

A convention (cf. Gilbert, 1981, 1989; Lewis, 1969; Sugden, 1986/2004; Young, 1993, 2006) is a behavioral regularity, i.e. a practice or procedure widely observed by members of a given social network, based on

the agent's goal of conforming to that behavior in order to act like the others,
the mutual expectation that the others will conform to that behavior as well.

More specifically conventions are a class of problems (arbitrarily selected from a potential of alternative candidates) classified as (pure) coordination games (viz. convention of keeping to the right (or left) when driving, pointed out by David Lewis, 1969), based on interdependency and mutual expectations (see Andrighetto et al., Forthcoming b).

The confine between conventions and norms is not clear-cut. Conventions may acquire a mandatory force over time (Andrighetto et al., Forthcoming b), sometimes conventions get to be prescribed, and this is one factor leading to norm emergence. A good example is etiquette, which is halfway between a social norm (with obligations and possibly sanctions) and a convention. Greeting is a polite behavior, and how to greet

[3] In EMIL-A, normative beliefs, together with normative goals, are organized and arranged in the normative board according to their respective salience (see chapter The EMIL-A Architecture for a detailed description). By *salience* we refer to the norm's degree of activation, which is a function of the number of times a given norm enters the agent's decision-making.

[4] A normative goal differs, on one hand, from a simple constraint, which reduces the set of actions available to the system, and, on the other, from ordinary goals. With regard to behavioural constraints, a normative goal is less compelling: an agent endowed with normative goals is allowed to compare them with other goals of her and, to some extent, to choose which one will be executed. Only if an agent is endowed with normative goals she can be said to comply with, or violate, a norm. With regard to ordinary goals, a normative goal is obviously more compelling: when an agent decides to give it up, she knows she is both thwarting one of her goals and violating a norm.

[5] An example is the following: tomorrow, I want to go gather mushrooms (relativized goal) because and to the extent that I believe tomorrow it will rain (expected event). The precise instant I cease to believe that tomorrow it will rain, I will drop any hope to find mushrooms.

someone - whether by shaking hands or waving hallo - is ruled by conventions; on the other hand, when you receive greetings, it is mandatory to reply, probably due to the social norm of reciprocity.

2.2.4 Power Over

Given the capability of an agent to bring about a set of world states and given the goals agents have in these objects we can define power over of a group of agents *I* towards a group of agents *J* as the possibility for I to realize/thwart a set of world states *G* wanted by *J*, such that *J* is not able without *I* to achieve *G*. This definition regards objective power over. We could analogously say that if *J* has power over *I* w.r.t. *G*, *I* is objectively dependent on *J* for *G*. There exist epistemic variants of dependence. In fact we claim that *J* is dependent on *I* for *G* if *J* believes *I* has power over him w.r.t. G.

2.2.5 Norm Frame

The frame is a set of features that characterize the norm and that define its crucial aspects, which are further decomposable. This means that when we specify a norm we need to say something about each of these aspects. A research question is thus whether these features are sufficient and necessary (thus, unique) conditions to construct a norm. We outline five aspects of the norm frame (a-e):

a) Deontic

A Deontic is basically a way of partitioning situations between good/acceptable ones and bad/unacceptable ones[6].

What is important for general recognition issues is that the authority from which the obligation emanates need to be recognized and accepted by the agents in order for the obligation to be dealt with and fulfilled or violated. As for validity based deontics, we can further distinguish them into:

- ***Obligations***: it is obligatory to do so,
- ***Prohibition***: it is prohibited to do so,
- ***Permissions***: It is permitted to do so.

These constructions are clearly interdefinable. If the definitions, and even the intertranslations, are clear (Prohibited, Permitted, Obligatory) the mechanisms need to be investigated further. For mechanisms we mean: what happens when something is obligatory? And "when is something obligatory?"

b) Source

A source is the locus from which the norm emanates. We distinguish the source into:

- ***Personal***: "the locus from which the norm emanates" means "the nonempty set of agents that performed that action enabling the norm (and after which the norm existed)";
- ***Impersonal***: "the locus from which the norm emanates" means "the community that enabled the norm". It is clear that considering the "community" equivalent to the "set of all agents" would make the two notions collapse. One of the aims of understanding impersonal source ought to be the understanding of this difference.

c) Normative Role

With normative role (Conte and Castelfranchi, 1995) we mean the partition of the agents involved in a norm. We distinguish:

- ***Legislators***, the personal source;
- ***Addressees***, those agents that are mentioned by the norma s asked to carry out or not carry out a given action;
- ***Defenders***, that is those agents that share and enforce the norm;
- ***Observers***, those that acquire beliefs about a norm, that is whether it is enforced, violated, emanated.

[6] On the concept of deontic see: (Wright, 1963), for validity based deontics; (Meyer, 1988), for deontic logic as a variant of dynamic logic; (Alchourròn, 1993), for recognition based deontics.

d) Enforcement mechanisms

These are the operations that attempt to modify agents' actions in order to make them compliant to a norm (cf. Axelrod, 1984; Conte and Castelfranchi, 2006; Conte and Paolucci, 2001; Ullmann-Margalit, 1977). Very schematically, we distinguish:

- ***Sanctions***: enforcement mechanisms that inhibit agents' actions;
- ***Incentives***: enforcement mechanisms that favour agents' actions.

The way actions can be favored or inhibited follows precise paths in cognitive agents (Andrighetto et al., 2009). In this sense enforcement mechanisms follow a path in agent minds, exploiting intra-agent processes. Moreover social artifacts can be used to sanction or to favor agents' actions. Reputation, for instance, can work as a normative sanction. But it can also be used as a normative incentive.

e) Control

Control is the way enforcement mechanisms are applied (Conte and Dignum, 2001; Conte and Paolucci, 2004). It implies both monitoring - that checks violation - and influence - that actively pushes cognitive agents' towards compliance. Normative influence will be analyzed further on.

They can be:

- ***Centralized***: only one agent (individual or supra-individual) is entitled to sanction;
- ***Distributed***: everybody is able to defend the norm.

Therefore, it has to be said that centralized control makes use of institutional rules for regulation, while distributed control does not presuppose any delegation.

Chapter 3 Theoretical Foundations: Two Way Dynamics of Social Processes

Rosaria Conte, Giulia Andrighetto, Marco Campennì

Abstract

When produced by autonomous social agents, emergent macro-social effects undergo a complex loop between bottom-up and top-down processes. Emergence of properties at aggregate level cannot be effectively accomplished unless properties feedback on the lower level through a complementary process of immergence. Immergence is a necessary correlate of emergence in at least a subset of macro-social phenomena, such as norms, typical of those societies of agents that are endowed with cognitive ingredients.

In this chapter the notion of emergence in complex social systems is discussed as a necessary instrument for a theory of the macro-micro link. Next, we will show how a given macro-effect is implemented on the lower levels, and two specific mechanisms of implementation, 2nd order emergence and immergence, will be discussed.

3.1 Introduction

At the beginning of the last century, some social scientists and anthropologists (Alexander, 1920; Broad, 1925) referred to the emergent macro-social effects as properties that cannot be deduced from properties at the lower social level. This assertion was heavily criticised (Hempel and Oppenheim, 1948) and argued to build on a logical confusion between propositions and properties. As the epistemologists observed, only propositions, and not properties, can be deduced. Consequently, the emergentist assertion must be referred to a *given* theory at a *given* stage of its development. By this means, the assertion gets weakened and transformed into a relativistic one, which states that propositions about macro-social properties cannot be deduced from propositions about micro-social ones under current theoretical boundaries.

However, such a formulation dispenses away with the notion of emergence at once: in the new relativistic assertion, emergent properties are simply not (yet) deduced. Hence, what emerges, is what is (still) unexplained. Once explained (Epstein, 2006), a property is no more emergent. Consequently, the notion of emergence comes to lose scientific value.

In the present work, we take a different perspective on the subject matter. We start from a crucial aspect of complex social systems, i.e. the difference between *implementation* and *incorporation*: a macro-social entity is always implemented on a micro-social one since it may act and take effect only through the actions of micro-social entities, i.e. individuals. When the producing units get modified in such a way that the emergent macro-social effect is more likely to occur again, we speak about incorporation.

In the successive section, we will discuss the way back from macro to micro, i.e. *downward causation*, a process that is certainly not new to the scientific community (see Campbell, 1974). Indeed, the micro-macro dynamics may be shown to consist of several, simple and complex, feedback loops (see Andersen et al., 2000). We will illustrate how a given macro-effect can retroact on the lower level, and will discuss one specific form of downward causation, i.e. evolutionary downward causation, defined as the emergent effect retroacting on the producing entities and determining new properties of these that contribute to select and replicate the emergent effect[7].

We will concentrate on one evolutionary form of downward causation, i.e. incorporation, in which the retroacting effect shows through the shape or the mechanisms ruling the behaviour of the generating systems. Evolution often determines the morphological incorporation of fitness-enhancing effects: this is

[7] In this treatment, evolution is not meant as a metaphorical notion, imported from biology. Following Dennet's view (1995), we argue that evolution is an abstract process concerning living organisms as well as cultural artefacts and social systems (see Conte et al., 2009).

apparently the case with the upright stance in the human species, or the size of the pelvis in the female of the same species. But sometimes emergent effects get incorporated into the behavioural mechanisms. Among primates, for example, grouping is incorporated into, shows through, specific behavioural dispositions of the corresponding species, such as grooming, or more complex social cognitive mechanisms, such as gossip (Dunbar, 1997). Gossipers *incorporate* a certain evaluation about the target, even if they do not actually share it, and contribute to its transmission. The emergent effect, the target's reputation, is incorporated, shows through, the process of transmission. Hence, whereas an emergent effect is always implemented upon (inter-)acting micro-social entities, it is not necessarily retroacting on, nor a fortiori incorporated into, them. In turn, incorporation, as we shall endeavour to show, includes different processes. Within incorporation we will discuss at some length one further distinction, that between 2nd order emergence and immergence.

2nd order emergence (Dennet, 1995; Gilbert, 2002) is a case of knowledge acquisition, in which the macrosocial effect gives rise to a new belief of the producing units. Immergence instead is a gradual process in which the emergent effects determine new mental phenomena in the agents involved, not necessarily aware of the effect produced. Several examples, including the case of *Potlatch* will be discussed (see section 3.2.5).

In these different types of downward causation, one can envisage a quintessential feature of social evolution and complexity. A typical example is norm-emergence. In our view, norm-emergence is characterized by the occurrence of two complementary processes, emergence and immergence: norms cannot emerge unless they simultaneously immerge into the agents' minds. In Chapter 9 (see also Andrighetto et al., 2007a), we present a normative agent architecture and illustrate its functioning when dealing with the immergence process.

3.2 The Way Back: Downward Causation

Can an emergent, macro-social property generate effects at the lower level? Yes, indeed. In Figure 2, there are two main ways in which downward causation occurs:

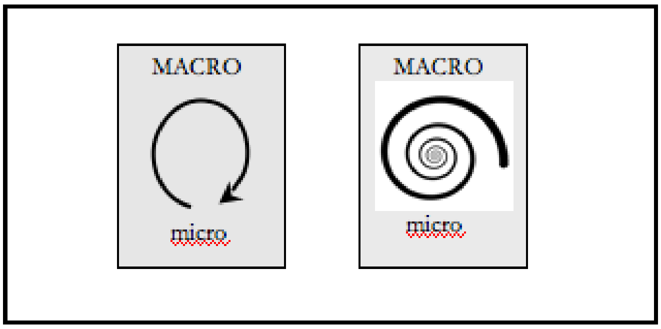


Figure 2. Figure on the left represents the simple loop of downward causation; figure on the right represents the complex loop.

Simple loop, which close the macro-micro circuit. The emergent effect retroacts on the lower level by determining a new property of the generating system. Below, we will present negotiation power as the result of a simple loop.

Complex, or evolutionary, loop, in which the emergent effect determines new properties on the micro-level by means of which the effect is reproduced. A recursive interaction between both levels is established by a complex feedback loop. This may occur in two ways, which are not incompatible:

2nd order emergence: i.e. the process by means of which, once produced, an emergent effect is recognised by the producing systems and by this means it is likelier to be reproduced (Dennett, 1995; Gilbert, 2002). In the Schelling model, Gilbert modelled the segregation-reinforcing effect of agents perceiving the clustering.

Emergent effects may retroact on their generating systems either closing the circuit, or opening up a new loop.

Immergence, i.e. the gradual and complex process by which the macro-social effect get reinforced by a new way of operating of the agents producing it. This result is usually generated through a number of intermediate loops. Before any global effect emerges, specific local events affect the generating systems, their beliefs, goals, and operating rules, in such a way that agents are likelier to reproduce the macroscopic effect (Castelfranchi, 1998; Conte et al., 2007; Andrighetto et al., 2007a). Segregation of course may favour the formation of new local rules. If inequality and the consequent need to be perceived as occupying the higher positions, two complementary rules might gradually appear: move to the socially higher neighbourhoods, if any, and move away from the lower, if any. It has been shown that this combination of rules has a remarkable segregating effect, which reinforces the social hierarchy (see Conte and Pedone, 2001).

3.2.1 Simple Loop

The emergent effect retroacts on the generating systems, determining new properties that might interfere negatively or positively with the micro systems' further activity. This is the case with a number of properties, such as rights, social status and social power, as well as the evaluations that agents form about one another.

Let us see one example of simple downward causation: the *power of negotiation*.

3.2.2 Dependence Networks

Dependence networks emerge from the interplay between one's utility to achieve others' goals and one's dependence from them (Castelfranchi et al., 1992). Agents involved derive a further property from such networks, i.e. negotiation power. This is the relationship between the agent's utility and her dependence (for the concept, see Conte and Castelfranchi, 1995), and emerges from the network into which the holder is plunged. If she moved to another dependence network, her negotiation power might increase or decrease, depending on the new interplay between her social utility and dependence in the new context.

In a common environment, actions done by one agent take effects on the goals of other agents. These are limitedly self-sufficient in the sense that they not always possess all the resources required to achieve their goals. Under these conditions, social *dependence networks* (Sichman et al., 1994; Sichman and Conte, 2002) emerge as interconnections among agents endowed with a finite number of goals and resources for achieving them. Suppose for example that in the set of agents <a, b, c>, *a* is endowed with goal *p* and action *a(q)*, while *b* and *c* are both endowed with goal *q* and action *a(p)*. Their interconnections result in a dependence network, where agents *b* and *c* are socially dependent on *a*, while *a* depends on either *b* or *c*. In turn, this non-uniform distribution of *exchange power* determines a new effect at the lower level: agents derive an equal power of choice, or, as we called it, negotiation power. In particular, *a* gets a higher negotiation power than either *b* or *c*: *a* will be in the position to make a choice, i.e. to choose its partner of exchange, while *b* and *c* have no choice. Presumably, due to this heterogeneous distribution of power, exchange will provide unequal outcomes (payoffs) to the participants, where agent *a* will be better off than either *b* or *c*.

This example clearly shows that an emergent effect (for example, a dependence network) may affect the lower level. This type of downward causation generates new properties (negotiation power) of the lower level systems, interfering positively or negatively with their successive achievements.

A problem about downward causation is to what extent it contributes to further dynamics in the global system. Undoubtedly, properties like *social power*, including negotiation power, and reputation have a definite but indirect impact on agents' further achievements: agents may suffer from or enjoy their effects in force of the actions that others, who interact with them, undertake based on their representations of such properties. Sometimes, these new properties not only interfere with the degree of adaptation of the individual agents, but also set off new emergent effects at the higher social level. This is what we call a complex loop.

3.2.3 Complex Loop (Incorporation)

Sometimes, this retroaction on lower levels may start up a new complex dynamics, by means of which the new properties reinforce or reproduce the emergent effect. The effect at the higher level more or less gradually gets implemented, selects a specific routine by which it is (repeatedly) executed by the individual agents. When does this happen, and how? This is what we turn to in the next two sub-sections.

2nd Order Emergence

Often, agents gradually become *aware* of the effects they contribute to generate. In this case, they form a mental representation of the emerged effect. This is what some authors call *2nd order emergence*.

Sometimes, agents' representations of emergent effects modify their actions, thereby taking further effect on the higher level. The social dynamics becomes recursive. How does this happen? As described in Dennett (1995), the process called 2nd order emergence is insufficient to account for this complex dynamics since beliefs do not automatically trigger action. Indeed, sometimes becoming aware of a given emergent effect may interfere and even counteract it. For example teachers are often warned against the Pygmalion effect (cf. Rosenthal and Jacobson, 1992) - by which their expectations about pupils act on the latter's future development and performance as self-fulfilling prophecies – in order to contrast and reduce its negative results.

Segregation

In his replication of the Schelling's model, Gilbert (2002) provides an example of 2nd order emergence that reinforces the emergent effect (in this case, the clustering). This happens because the new belief provides a guideline for action, for example "move only if there are spots where you will be happier". The new belief reinforces the macro-social effect (a stronger effect of clustering) to the extent that it allows a more efficient satisfaction of the local rule (the rule of happiness). The link between the new belief and the consequent adjustment of the rule affects the dynamics of the whole system. The macro-social effect is reinforced by the mental pattern that includes the new belief and the rule execution.

With this type of 2nd order emergence, clustering is implemented on the generating rule. By this means, Gilbert showed how and why 2nd order emergence may in turn affect the dynamics of the global system, and turn it into a complex bidirectional micro-macro loop.

3.2.4 Immergence

Here, the macro-social effect affects the generating systems through the latter's mechanisms, increasing the probability to be reproduced by them. As the swarming behaviour of lower species shows, collective effects evolve thanks to simple rules incorporating them into the local units with no need for a perception or understanding of the resulting effect. This is what we call immergence.

To be noted, immergence is not an exclusive feature of simple organisms. Even among humans, behavioural regularity may be implemented on a number of different mechanisms, which include but are not reduced to a real majority rule. Indeed, the latter is not always applied, nor is always efficient (see below Stalemate). Consider the example of the gift economy. The *Potlatch*, a rite common to some North American native tribes living on the North-Western Pacific coast of US and Canada, like *Haida, Tlingit, Tsimshian, Salish, Nuu-chah-nulth* and *Kwakiutl*, is the most important example of gift economy. During the *potlatch* the hosting tribe shows its importance and wealth through the distribution of its goods, inducing guests to reciprocate when time will come for them to hold their own potlatch.

Some tribes, for example, the Kwakiutl, used the potlatch as an arena for competition. Often, goods were destroyed right after the gift. Classic anthropologists - Franz Boas (1911) and Marcel Mauss (1922) – argued that by this means goods that would have caused a deep alteration of the system's equilibrium, were removed, thus maintaining the system's stability.

For the purpose of our argument, we can take this interpretation for granted. Obviously, there was no need for rite participants to know what was the real heart of the matter, so to speak. They might have come to believe that bringing gifts was necessary to ingratiate the divinity and obtain her favour, for example a famine's end. Whatever the most likely reason for the success of the rite and its further replication, the wrong interpretation allowed the macro-social effect (*potlatch*) to get incorporated into a new system of rules. Rather than 2nd order emergence, i.e. a shared perception of what was really going on, the collective effect (exchange of gifts) led to rules, conventions and institutions based on wrong assumptions (god's ingratiation) but working efficiently (society stability). This is what we call *immergence*. While in 2nd order emergence, the global effect is replicated because agents perceive its positive impact, in immergence it is replicated by virtue of a complex mental process, in which agents form beliefs, goals and rules that ensure the implementation of the initial effect without perceiving nor aiming at it.

Of these two different forms of incorporation, 2nd order emergence is certainly more rational, but requires causal reasoning and anticipatory skills. However, irrational thinking may obtain competitive results. A good example is Jim Doran's (1998) study of collective *misbeliefs*, showing the impact of shared false beliefs on the survival of an artificial population. This example clearly illustrates why we need to distinguish between the two: sometimes, if agents perceive the real emergent effect of their behaviour, they will cease to bring it about. In our example, we cannot say whether natives would have maintained the tradition of *potlatch*, had they perceived its real *raison d'être*.

In short, immergence is a hybrid process leading to (a) new beliefs in the agents' minds, which do not necessarily correspond to the emergent effect (in the *Potlatch* example, god wants food offer); (b) new goals (to ingratiate god); (b) new types of mental constructs (e.g., normative prescriptions); (d) new rules (e.g., norm adoption).

Example of immergence abound in social life, resulting from more or less intelligent forms of inter-agent adaptation (for a review, cf. Conte and Paolucci, 2001). Let us see some phenomena that might be interpreted as produced by more or less complex immergent processes.

Arena Effect

Often in common environments, participants are forced to exhibit increasingly augmenting values on certain behavioural dimensions in order to maintain efficiency in achieving their goals. For example, in a noisy pub, each must raise own voice on others' to be barely audible by one's neighbours. Obviously, no-one wants the noise to grow. Nor must participants be aware of the global effect obtained: they will automatically and rapidly adjust to the external standard.

A special case of the same phenomenon is the *vulnerable position*, in which agents are urged to behave as others do to avoid a risky position (for example, cars are forced to speed up on the highway).

In such examples, while the immergent rule tells you to adjust your behaviour to external standard, the emergent effect is an asymptotic increase of behavioural dimensions (e.g., noise or speed), and a loss of efficiency.

Stalemate

Consider the famous witness effect found out by Latané and Darley (1970) in social emergencies. A great deal of experimental and observational evidence shows that the probability of intervention in these situations drops dramatically when bystanders exceed number *three*. Why?

The authors put forward a rather elegant explanation, according to which a majority rule (i.e., check what the majority is doing under uncertain conditions) leads to a stalemate when a majority exists, i.e. when bystanders are at least three. Under such conditions, since each one checks what the majority is doing, nobody moves. Participants are frozen in the role of witnesses.

The witness effect provides a clear example of emergence: although no one intends to bring it about, the effect is generated by the majority rule, precisely as much as segregation was generated by the rule of happiness. A fragment of the generative process is emergent. To see this, consider that agents may be trained to avoid the witness effect, which is a highly socially undesirable phenomenon, simply becoming aware of it.

Moreover, the witness effect shows that an emergent effect may modify the mechanism that produced it at the lower level, without being recognised by the generating system. The stalemate leads to a reinforced local rule: the more likely the stalemate (macro-effect), the stronger the local rule (majority rule): agents are lesser and lesser likely to break it. The witness effect retroacts on the producing systems, at least temporarily reinforcing the rule. In this meantime, agents have no idea what is going on. All we may say is that the witness effect is implemented on a rule at the agent level, i.e. the majority rule, which gets reinforced by the effect in question while at the same time reproducing it.

Immergence of Norms

The most striking example of immergence is social norms. Social norms are social prescriptions implicitly transmitted from one agent to another, through deontics of the type "one must do ...", "people are obliged to ... ", and sometimes conveyed under evaluations in the form "it is good/bad to do ...".

Norms emerge as a mechanism of social regulation or to solve problems of coordination. agents do not need to represent the effects of norms in order to comply with them. All they must do is accept the norm. How is this possible?

In a view of norms as two-sided, external (social) and internal (mental) objects (Conte and Castelfranchi, 1995, 1999, 2006), norms come into existence only when they emerge, not only *through* the minds of the agents involved, but also *into* their minds. In other words, they work as norms only when the agents recognize them, reason and take decisions upon them as norms. The emergence of norms implies their immergence into the agents' minds. Only when the normative, i.e. prescriptive, character of a command or other action is recognized by the agent, a norm gives rise to a normative behaviour of that agent. Thus, for a norm-based behaviour to take place, a normative belief[8] has to be generated into the minds of the norm addressees, and the corresponding normative goal has to be formed and pursued. Our claim is that a norm emerges as a norm only when the associated belief immerges into the minds of the agents involved; in other words, when agents recognize it as such. Obviously the effect of the norm, which is probably the reason why it evolved or was issued, does not need to be perceived. What is more, even when perceived, it is not such a perception that leads the agent to implement the norm. The 2nd order emergence of norms is often a simple loop: the agents' recognition of the effect brought about by the norm may simply close the micro-macro circuit, without contributing to replicate the effect. To see why, one should simply wonder why we need norms at all: if the simple perception of their effect were enough for agents to reproduce it, there would be no need for norms with their enforcement mechanisms, surveillance system, police, institutions, and social order. *Norms are thus mechanisms for the immergence of social order.*

In previous works (Castelfranchi, 1998; Andrighetto et al., 2007a; Conte et al., 2007), we described the process of norm emergence as a gradual and complex dynamics by which the macro-social effect, in our case a specific norm, emerges in the society *while* immerging in the minds of the agents producing it, generating a number of intermediate loops. The generation/emergence of social norms is a major circuit made of local loops, in which:

partial or initial observable macroscopic effects of local behaviours retroact on (a subset of) the observers' minds, modifying them (producing new internal states, emotions, normative beliefs, normative goals, etc.) agents communicate internal states to one another, thus activating a process of normative influencing (see Conte and Dignum, 2001) these normative beliefs spread through agents' minds behaviours progressively conform to spreading states initial macroscopic effects get reinforced/weakened depending on the type of mental states spreading.

Thus, before any global effect emerges, specific local events affect the generating systems, their beliefs and goals, in such a way that agents influence one another into converging on one global macroscopic effect. Emergence of social norms is due to the agents' behaviours, but the agents' behaviours are due to the mental mechanisms controlling and (re)producing them (immergence). Of course, our view of norms calls for a cognitive architecture of normative agents, which is not new to the field of agents and multiagent systems (think of the BOID architecture, for example). But previous BDI (Beliefs-Desires-Intentions)

[8] Drawing upon Kelsen (1979), von Wright (1963) and a long tradition of deontic philosophy and logic-based theory of action, we define a normative belief as a belief that a given behaviour, in a given context, for a given set of agents, is either forbidden, obligatory, or permitted (Conte and Castelfranchi, 1999, 2006).

approaches to normative reasoning suffer from some drawbacks: norms are *pre-established* and *built into* the agents. Instead, we endeavour to have agents able to find out new norms and transmit them to one another. In Chapter 9 (see also Andrighetto et al., 2007a), an analysis of our normative architecture, EMIL-A, is presented.

3.3 Advantages of the Present Approach

- The present model attempts to contributing to the study of the micro-macro link, and more specifically to a generative view of this process. The generative paradigm will play a decisive role in future developments of the social science, as indicated by some evidence:
- recent official formulation of the generative paradigm for the social science (cf. Epstein, 2006)
- fast development of generative methodologies for the study of social phenomena (agent based social simulation)
- continuous growth of simulation toolkits and platforms (from swarm libraries to the *logo languages)
- accessibility of such languages and toolkits to non-expert programmers, etc.

However, generative social science is still formulated in a somewhat unsatisfactory way, namely as a bottom-up process (again see Epstein, 2006, and more generally see the vast majority of simulation and computational models of social and economic processes).

Hence, the notion of emergence itself is usually intended as a one-way process. Indeed, this notion is being substituted by that of generation.

The present analysis can contribute to

- a theory of emergence as distinct from generation
- a view of the micro-macro link as a recursive
- loop, in which emergent effects at the macro-level retroact on the lower levels, modifying them,
- thereby providing a more dynamic, generative view of the micro- level entities.

3.4 Concluding Remarks

Society is generated by its members, and is implemented on them: it works thanks to and through their actions and their minds. But this does not implies that society's members aim at, nor are aware of, the way society works.

Moreover, they may be aware of emergent effects, but still this representation is not what makes society works. Sometimes awareness is a requirement for the implementation for society on its members sometimes, this is not the case. An interesting empirical question is when it is, and when instead new properties do not imply a representation of the effects they contribute to produce.

Chapter 4 Theoretical Foundations: Normative Behaviour and Economic Wealth (The Development of Economic Wealth Based on Trust in Large Groups of People)

Rainer Hegselmann and Oliver Will

Abstract

David Hume delivered an informal theory of how humans managed their way from a rather poor life in small groups to comparatively high wealth based on division of labour in large groups of people without personal ties. The dynamics are driven by two antagonistic forces: on the one hand specialisation entails incentives for division of labour but on the other hand the interaction structure of exchange regimes is that of a social dilemma. In this chapter an agent-based model is introduced that formalises important elements of Hume's theory and thereby integrates the emergence of trust and that of division of labour. The main concepts that capture Hume's ideas are described and some results are shown that illustrate the importance of trustful behaviour and the trouble to establish it.

4.1 Introduction and Motivation

In this chapter, we present a model on a fundamental process in human societal life: the development from a rather poor life in small groups to a remarkable economic wealth based on division of labour in large groups of people. David Hume was the first who delivered a rich informal theory on this issue.[9] His theory contains many elements that remained scientifically appealing until today.[10] For that reason, we use it as a theoretical background of our work. Our model can thus be understood as a formalization of Hume's ideas and that is why we betimes refer to it as HUME1.0 in the remainder.[11]

Though in principle everybody in a society could benefit from division of labour, there is a big hurdle to take: Division of labour requires exchange and exchange tends to be risky. Exchange is risky since one or both involved parties may deviate from agreements, default on payment or delivery, or defect in some other way. To overcome these problems, humans made two important inventions: virtue and government. In our context, the essential virtue is Hume's notion of *justice*. It means keeping promises and fulfilling contracts. Government serves as an enforcement agency where the force of virtue turns out to be too weak. At the current stage of our formalization of Hume's theory we concentrate on the role of justice and leave the origin of government for future research.

Though Hume wrote in a jargon of virtues, his ideas can easily be expressed in terms of norms, e.g.: If you agreed on delivering a product to a certain price and received the respective payment then you should deliver the promised product! In HUME1.0 we concentrate on norms like the one above. They determine whether or not the preparatory effort made by one of the involved parties should be rewarded by the other. Hume's theory deals with norms at two levels. At the behavioural level, he delivers a theory on how norm compliance could emerge. At the mental level his work is concerned with how humans develop a special internal relation to norms, i.e. how norms develop to work as motivating factors.

HUME1.0 focusses on the behavioural level. We investigate whether, under what assumptions and to what extend certain behaviours prove successful. Success is measured in material payoffs in the more or less long run when agents face situations that can be described as a *trust game*. The evolved behaviour is a "social norm" in the following sense:

[9] See Of Morals – that is part III of his A Treatise of Human Nature (1739f., Hume, 2007) – and Enquiry concerning the Principles of Morals (1751f, Hume, 1998)

[10] In the past two decades a lot of work was done on this theory. For detailed analysis, reconstruction, and elaboration of Hume's theory see e.g. (Kliemt, 1985; Kliemt, 1986; Sugden, 2004; Binmore, 1994; Lahno, 1995; Hardin, 2007).

[11] The relation between our model and Hume's moral and political philosophy is described in much more detail in (Hegselmann and Will, in press).

i. It guides behaviour in a social dilemma situation such that overall welfare is improved.
ii. It is followed only conditionally, i.e. only if enough others are following the same norm.

The mental level of norms is left out in this version of our model. In particular, there are no components of mental processing of norms (e.g. *normative beliefs*, *normative goals*, *rules of norm adoption*). Anyhow, HUME1.0 is compatible with the approach to norms presented in later chapters of this report. It could be extended by cognitive modules presented there but, for reasons of complexity, it is not yet. At the moment, the model serves as a base case that lacks the mental level. The results we have so far indicate that the cognitive level might be a missing element in explaining the evolution of division of labour in large groups.

In what follows, we present our model without going into technical detail. Subsequently, we show what levels of wealth would emerge given that all agents followed a certain norm (in our weak sense). Afterwards, we investigate the levels of wealth evolving if certain assumptions on the agents' abilities to detect each others trustworthiness are given.

4.2 Interaction Structure and Specialisation

The dynamics suggested by Hume's theory is affected by two antagonistic forces: on the one hand, agents have different levels of competence in solving certain problems and those competencies change via practice, innovation etc. Thus there is an incentive for specialisation and division of labour. On the other hand, the interaction structure of exchange regimes is that of a social dilemma. This will become more clear in the remainder of this section.

The interaction structure that HUME1.0 focusses on is the *trust game* (*TG*), a simple 2-person game that plays a central role in Hume's theory. In a TG, two players could gain from mutual co-operation but one of them, player1, has to choose to contribute in advance (*trust*) while the other, player2, chooses *afterwards* whether he contributes as well (*reward*) or not (*exploit*). Since for player2, exploiting is better than rewarding, and being exploited is worse than distrusting for player1, it can be seen by backward induction that for *rational* players the *non-iterated*, *one-shot TG* has only one solution: anticipating that player2 will go for exploitation, player1 decides not to trust. Result is an inefficient outcome. The interaction structure captured by the trust game is the very core of HUME1.0 but it is incorporated into an *enriched setting*.

Key ingredients of this setting are:

1. In each iteration half of the agents have one of K special problems. Those with a problem are referred to as P-agents the others are called S-agents.
2. Agents have a competence vector with K entries that represent their competence in solving the respective problem. By working on a certain problem, agents become better in solving that problem. However, at the same time their other competencies deteriorate. Formally, this is realised by adding a certain Δ to the component in question and afterwards re-normalising the whole competence vector in such a way that $\sum_{k=1}^{K} c_{i,k} = 1$ holds again.
3. The more competent an agent is in solving a problem k, the less are his costs of producing a solution and the higher is the quality and thus the value added.
4. Agents can solve their problems on their own or look for more competent agents that can solve it in a cheaper and better way.
5. Pairs of agents result from a matching process in which the competence and trustworthiness of agents plays an important role (see section 4.4).
6. If a match is established, the P-agent has to do some prepayment. Only afterwards, the S-agent that was "hired" to solve the problem, starts working on the solution—or not. Prepayment of the P-agent and the resulting temptation for the S-agent to keep the prepayment without delivering the solution, makes the setting strategically analogous to the trust game described above.[12] Figure 3 illustrates the interaction structure. The payoff structure is the same as in the simple trust game:

[12] See (Hegselmann and Will, in press) for a motivation of the chosen and a discussion of other plausible exchange regimes for HUME1.0.

a P-agent likes the outcome from him trusting and the S-agent rewarding best, him distrusting second best and is worst off if he trusts and the S-agent exploits. For the S-agent, the highest outcome results from exploiting a trusting P-agent, followed by rewarding a trusting P-agent and earning nothing in case of a distrusting P -agent. While this structure never changes in the model the total payoffs depend on the S-agent's competence in solving problem k, parameters concerning costs and value function and the share of the value added[13] that is earmarked for the S-agent, β, which is an exogenous parameter.

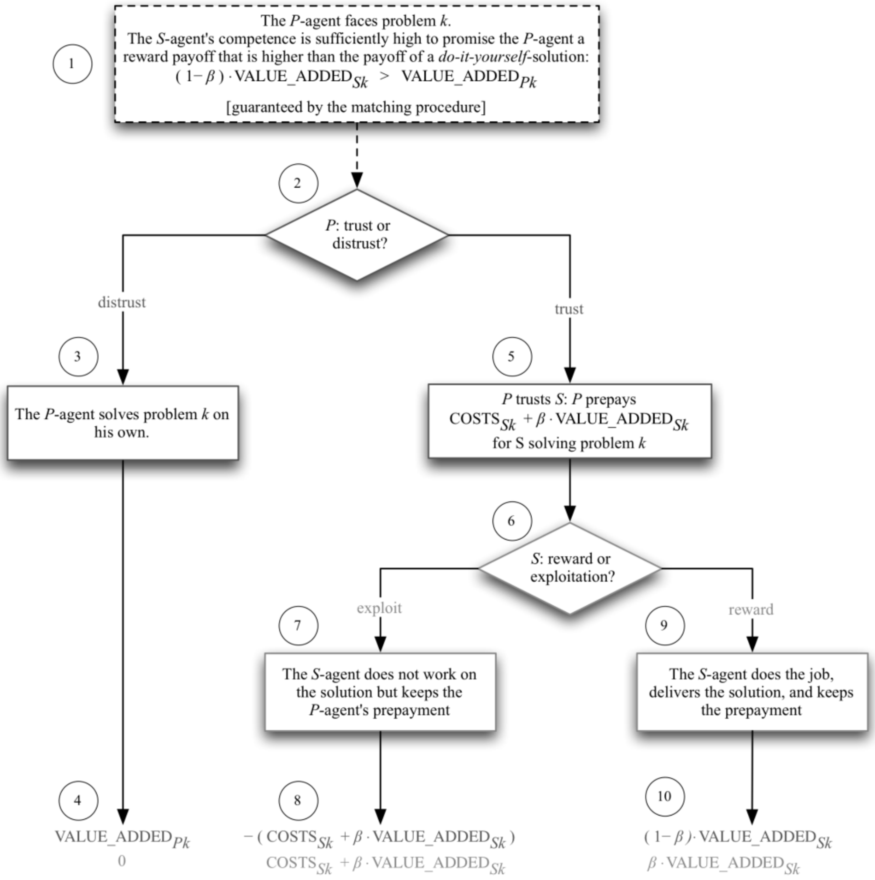

1
The *P*-agent faces problem *k*.
The *S*-agent's competence is sufficiently high to promise the *P*-agent a reward payoff that is higher than the payoff of a *do-it-yourself*-solution:
$(1-\beta)\cdot \text{VALUE\_ADDED}_{Sk} > \text{VALUE\_ADDED}_{Pk}$
[guaranteed by the matching procedure]
2
P: trust or distrust?
distrust
trust
3
The *P*-agent solves problem *k* on his own.
5
P trusts *S*: *P* prepays
$\text{COSTS}_{Sk} + \beta\cdot \text{VALUE\_ADDED}_{Sk}$
for S solving problem *k*
6
S: reward or exploitation?
exploit
reward
7
The *S*-agent does not work on the solution but keeps the *P*-agent's prepayment
9
The *S*-agent does the job, delivers the solution, and keeps the prepayment
4
VALUE_ADDED_{Pk}
0
8
$-(\text{COSTS}_{Sk} + \beta\cdot \text{VALUE\_ADDED}_{Sk})$
$\text{COSTS}_{Sk} + \beta\cdot \text{VALUE\_ADDED}_{Sk}$
10
$(1-\beta)\cdot \text{VALUE\_ADDED}_{Sk}$
$\beta\cdot \text{VALUE\_ADDED}_{Sk}$

Figure 3. The trust game in HUME1.0. The matching algorithm ensures that the *S*-agent is sufficiently competent. At the first node the *P*-agent decides whether he trusts the *S*-agent or not. If the *P*-agents trusts, the S-agents decides whether or not he delivers a solution or not. Upper payoffs are those of the *P*-agent, lower payoffs refer to the *S*-agent.

4.3 The Spatial Scenario

HUME1.0 analyses the trust game in different spatial scenarios (Hegselmann and Will, in press) but here we focus on only one: the **p**artition and **m**arket based scenario (PM-scenario). In the PM-scenario agents are randomly distributed among an exogenously given number of partitions. In each iteration, they decide on whether they enter a central market or stay in their local partition. If an agent enters the market, he searches for a partner among all those agents that chose to go for the market as well. Agents that stay in their partition search for a partner among all those agents in the same partition that also chose not to enter the market. Potentially agents that enter the market can choose among a larger set of partners. Thus they have a better chance of finding a good partner.

[13] That "value added" is the difference between the value and production costs of a solution.

The decisive structural details are:[14]

With an agent-relative, dynamic probability, $p_i^{P\rightarrow market}$, *P*-agents look for a partner to solve their problem at the market. They search for partners that are trustworthy and as competent as possible with regard to their. Correspondingly, with probability $p_i^{P\rightarrow local}(t) = 1 - p_i^{P\rightarrow market}(t)$ *P*-agent *i* looks locally, i.e. within his partition, for a trustworthy and competent *S*-agent. Analogously, *S*-agents have an propensity, $p_i^{S\rightarrow market}$, to go for the market.

S-agents reward with a certain agent-relative and dynamic probability: With probability $p_i^{reward\_market}(t)$ agent *i* rewards in *market*-interaction. In *local* interaction their probability to reward is $p_i^{reward\_local}(t)$.

4.4 Matching Agents

The matching procedure of HUME1.0 is not an explicit model of how agents search for partners. It is a random-based mechanism that is designed to make pairs that agents—with all the limitations concerning cognitive abilities, information, and time restrictions—would or could plausibly have brought about by their activities. The effects that the matching procedure should produce, can be translated into the following requirements (R1 to R5):

in all pairs, *P*- and *S*-agent are in each other's pool of possible partners,

all *P*-agents that have a partner think that this partner is trustworthy,

all pairs are economically reasonable in a sense that is described below,

two agents' probability to be matched is positively correlated with the payoff they expect from an interaction, and

the matching does neither constitute privileges for *P*-agents nor for agents of *S*-type.

The matching procedure is a two-stage mechanism that takes as input the set of all possible pairs of a *P*- and a *S*-agent. In the first stage, the subset of plausible pairs is identified. A pair is plausible if it meets the demands of requirements 1 to 3. In stage two, the actual pairs of the current time step are drawn at random from the set of plausible pairs. To fulfil requirement 4, the probability of a pair to be drawn depends on the payoff expected by the involved *P*-and *S*-agent.

4.4.1 Identifying the Set of Plausible Pairs (R.1 to R.3)

Let $\mathcal{P}$ be the set of all *P*-agents and $\mathcal{S}$ that of all *S*-agents. Then $\mathcal{P} \times \mathcal{S}$ is the set of all possible pairs of a P -and a S-agent. In the PM-scenario there are spatial reasons why some of these pairs are implausible (R1). Only those pairs in which *P*-and *S*-agent are located in the same partition and at the same time search for a partner within the partition and those in which both agents search for a partner on the global market are plausible and the subset of those pairs is denoted by $(\mathcal{P} \times \mathcal{S})_{spatial} \subseteq \mathcal{P} \times \mathcal{S}$.

The second step of our matching algorithm reduces the set of plausible pairs to those spatially plausible pairs in which the *P*-agent classified the *S*-agent as a trustworthy partner (R2). In this version of the model *P*-agents simply make guesses on their potential partners *actual behaviour* in the current time step, i.e. on whether or not a potential partner would reward.[15] These guesses are wrong with probability p_{wrong}^{local} or p_{wrong}^{market} depending on whether the respective agent searches for a partner on the market or in the local partition. Since we assume classification mechanism to be more effective with people that have lower social distance must hold that $p_{wrong}^{local} \leq p_{wrong}^{market}$. $(\mathcal{P} \times \mathcal{S})_{trusted} \subseteq (\mathcal{P} \times \mathcal{S})_{spatial}$ denotes the subset of pairs in which the *P*-agent trusts the *S*-agent.

[14] The PM-scenario has structural similarities with the Macy-Sato-model (Macy and Sato, 2002; Will and Hegselmann, 2008; Will, 2009b) but in HUME1.0 the probability for rewarding on the market can be different from the probability for rewarding locally. An endogenous mechanism of mobility between partitions is planned to be implemented in future versions of the model.

[15] A alternative classification mechanism that is based on the agents' probability to reward rather than their actual behaviour is described in (Will, 2009a).

Requirement 3 concerns economic reasonability. In HUME1.0 *P*-agents can solve their problems on their own. Thus only pairs in which the *S*-agent's competence is high enough to ensure that the *P*-agent's payoff in case of reward exceeds his payoff in case of solving his problem on his own are plausible in terms of economic reasonability. The matching procedure finds the subset of pairs, $(\mathcal{P} \times \mathcal{S})_{competent} \subseteq (\mathcal{P} \times \mathcal{S})_{trusted}$, that contains all such pairs.

We end up with a subset of pairs that are plausible with regard to spatial reasons, trustworthiness and competencies. Thus we have a set of pairs that fulfil requirements 1 to 3 and which is therefore a set of plausible pairs, $(\mathcal{P} \times \mathcal{S})_{plausible}$, as it was defined in the beginning of this section. Now, that this set is determined we can step forward to the second stage of the matching mechanism.

4.4.2 Finding the Pairs of the Current Time Step (R4 and R5)

In the second stage of the matching process, we take the set of plausible pairs and randomly determine the actual pairs of the current time step. This process cannot be random in the sense that all pairs have the same chance of being drawn since requirement 4 demands this process to be assortative to a certain degree: the probability of a pair to be drawn for the current time step must be positively correlated with the payoff expected from the respective *P*- and *S*-agent. To fulfil requirement 5 the process must be assortative without implying type privileges, i.e. it should neither favour *P*- over *S*-agents nor the other way round.

The determination of the pairs of the current time step starts with listing all *P*- and *S*-agents that occur in plausible pairs. Two lists result: a list of all *P*- and a list of all *S*-agents in plausible pairs. To avoid any risk of effects caused by the sequence in which the agents appear on the lists, random permutations of these lists are generated. Afterwards the drawing of pairs begins. This drawing procedure consists of a number of steps in which elements are deleted from the two lists. It is repeated until either the list of *P*- or the list of *S*-agents is empty.

1. A random number between zero and one is drawn to **determine whether a *P*- or a *S*-agent's perspective is taken**. To keep the process neutral we take on a *P*-agents perspective if the random number is smaller than the ration of P -and S-agents, i.e. if $r < \frac{\#list\text{-}of\text{-}p\text{-}agents}{\#list\text{-}of\text{-}s\text{-}agents}$ where *r* is a uniformly distributed random number between 0 and 1.
2. We **get a chosen agent** by taking -depending on the perspective decided in step 1 -the first entry on the list of *P*- or *S*-agents. This *chosen agent* is the agent for whom a partner is drawn in this run.
3. **The chosen agent's expected payoffs with all his partners in plausible pairs are calculated.** For a *P*-agent this means that we calculate the reward payoffs since the set of plausible pairs does not contain a pair in which the *P* -agent expects the *S*-agent to exploit. In case of the chosen agent being a *S*-agent, the expected payoffs depend on whether or not the chosen agent intends to reward or to exploit.
4. **A vector of the chosen agent's expected payoffs is formed**.
5. **The elements of the vector of expected payoffs are normalised to unity**, i.e. each value of expected payoff is divided by the sum of the expected payoffs from all the chosen agent's partners in plausible pairs.
6. **The chosen agent's *actual* partner is determined using a uniformly distributed random number between zero and one.** This is done by consecutively summing up the values in the vector of expected payoffs until this sum exceeds the drawn random number. That chosen agent's possible partner to whom the expected payoff relates that was the last one added to the sum is the chosen agent's actual partner in the current time step.
7. **The chosen agent and his actual partner are removed from the lists of *P*- and *S*-agents.** This ensures that a paired agent cannot appear as a chosen agent afterwards.
8. **All plausible pairs that contain either the chosen agent, his partner or both are removed from the set of plausible pairs** to ensure that agents that are already paired do not appear as a chosen agent's possible partner afterwards.

At the end of step 8 we go back to step 1 and draw the next pair of agents for the current time step. This process is repeated until one of the lists is empty in which case no plausible pairs are left and our matching for the current time step is complete.

4.5 Learning

Agents develop their competencies by working on problems. Besides this technical learning agents learn on how to decide in certain situations. First, they have to choose on whether they search for a partner on the market or in partition. Since they could be either in a *P*- or a *S*-agent's position they have two propensities to go for the market: $p_i^{P \rightarrow market}$ and $s_i^{P \rightarrow market}$. Second, S-agents decide on whether or not they reward their partner and they differentiate between market interaction and interaction in the partitions. Thus for this decision two further propensities are needed: $p_i^{reward\_market}$ and $p_i^{reward\_local}$. These four propensities constitute the agents' *decision vector*.

The transformation of an agent's decision vector is always *success driven*, i.e. it is assumed that moral attitudes (the propensities to reward) cannot evolve if morality is a *loser strategy*. Thus in HUME1.0 trust and trustworthiness erode if agents that use it have lower payoffs than those who do not. The same is true for their propensities to go for the market. This idea is implemented using a kind of role model learning. For each agent *i* in the population it works as follows:

> The pool of agents from which agent *i* selects his role model is determined by randomly drawing a exogenously given number of agents that inhabit the same partition as agent *i*.
>
> Given that there is an agent in agent *i*'s learning pool whose sum of payoffs exceeds that of agent *i*, that agent with the greatest sum of payoffs in the pool is agent *i*'s role model in the current time step.[16] If agent *i*'s sum of payoffs exceeds that of all agents in his learning pool, agent *i* does not have a role model and therefore does not change his decision vector.
>
> Each value in agent *i*'s decision vector is replaced by the corresponding value in the decision vector of his role model with a probability given by an exogenous parameter.[17]

Besides this learning mechanism agents' propensities change due to some random mutation. In every time step each component of each agent's decision vector changes with a certain probability that is given by an exogenous parameter. A further parameter determines to what amount the component changes.

4.6 The Importance of Moral Behaviour

Agents in HUME1.0 repeatedly have to decide on whether or not they reward a trusting partner. They face this decision either on the market or in local interaction. Thus one can think of four plausible rules of behaviour that agents might follow:

> Be trustworthy regardless of whether you are in your neighbourhood or on the market!
>
> Be trustworthy in neighbourhood interactions but not on the market!
>
> Be trustworthy on the market but not in your neighbourhood!
>
> Be untrustworthy in all interactions!

To analyze the importance of what behaviour the agents in our model adopt, we conducted experiments in which we simply assumed that all agents follow one of the four rules above. That is, we conducted 20 runs of simulations in which all agents were always trustworthy and trusted all potential partners. Afterwards we simulated societies in which agents were trustworthy and trusted only in local interaction but not on the market. These experiments were followed by others in which only market interactions were trustful. Finally we assumed all agents to be untrustworthy and distrusting all the time. In this way, we measured the average level of wealth that evolves *if* one of the behavioural rules above were generally accepted.

[16] Actually, the sum of payoffs is discounted by a rate that is given as an exogenous parameter.

[17] Note that this does not mean that either all or none of the components in agent *i*'s decision vector are changed but rather that all, none, or some could be changed.

Figure 4 displays the wealth that a population produces in time step 10.000 on average of 20 repetitions given that all agents follow the respective rule. The average wealth in a population in which trust is generally established is our benchmark. All values in the figure are normalized to ratios of this benchmark.

We can see from the figure that a population in which agents do not trust each other reaches only approximately 26% of the wealth in a population of agents that have trustful interaction. This shows us that our model captures the importance of trust for the development of economic wealth.

Furthermore, the plot shows us that the wealth gained in a population in which everyone has trustful local interactions is only 71% of the wealth in a population where all agents interact trustfully on the market. This means that the model implements opportunity costs for restricting the set of potential partners.

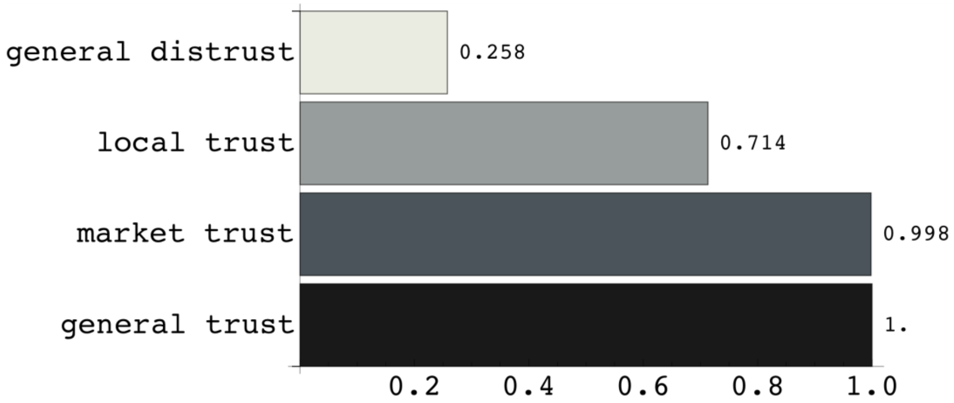


Figure 4. Average levels of wealth given that all agents generally follow one of four rules of behaviour. Bars show the wealth that is reached in percentage of the wealth that develops if every agent is trustworthy and trusts in all interactions. Mean values of 20 repetitions in a population of 500 agents in 10 neighbourhoods facing 5 types of problems are shown.

4.7 What Level of Wealth Evolves if Moral Behaviour is not Determined?

In the previous section we have shown that the wealth produced in a society depends crucially on the agents' ability to have trustful interactions. Now we want to find out what wealth evolves if we assume that our agents' choices on their behaviour are success-driven. We therefore remove the assumption that our agents' behaviour in the trust game is determined but assume that it changes according to the learning dynamics described in section 4.5.

The mechanism to promote trustful interactions implemented in our model is the classification mechanism described in section 4.4.1. To gain insights in the importance of the precision of classification, we conducted experiments with several combinations of probabilities to get a partner's intention wrong in local and market interactions. For each combination, we ran 20 simulations and measured the mean values of wealth at time step 10.000. Figure 5 shows the results. The color of the patches indicates how close the evolved wealth is to the wealth that evolves in our benchmark populations in which every interaction is trustful (see previous section).

We can state that with a classification mechanism that works perfect in local and market interaction (patch 0 0), the average level of evolving wealth is similar to that in the benchmark population (98%). Given perfect classification, agents that are not trustworthy are always recognized, never trusted and thus do not gain any profits. Therefore, they switch to trustworthiness and we observe very high levels of trust that are only disturbed by some random mutation. Furthermore, agents head for the market since there the average payoffs are higher due to a larger number of possible partners. Thus perfect classification leads to agents that are involved in trustful market interactions and reach very high levels of wealth.

Unfortunately, as soon as agents make mistakes in guessing their partners' trustworthiness, the evolved wealth decreases dramatically. Those experiments in which classification was very unreliable in market and local interactions (upper-right corner) lead to levels of wealth around 30%. This is only slightly better than what we saw in the population in which no interaction is ever trustful.

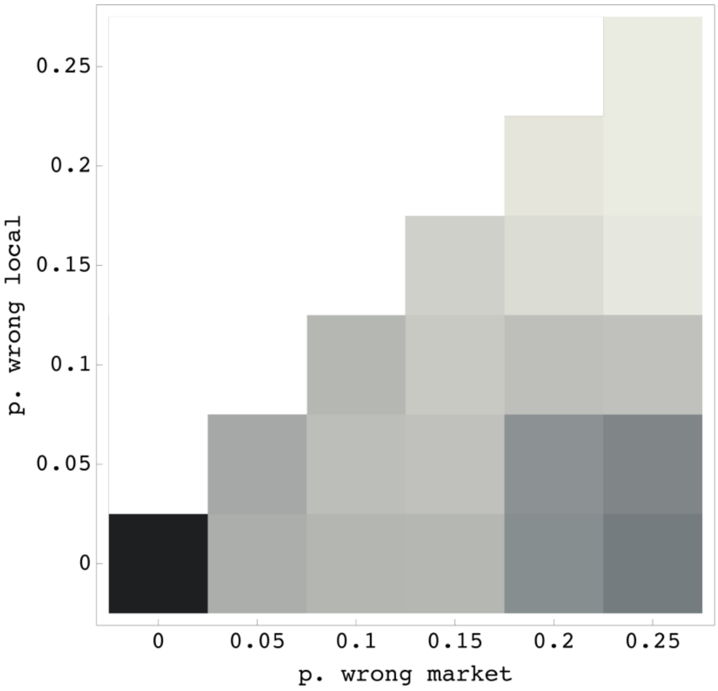


Figure 5. Average levels of wealth given several combinations of classification reliability in local and market interactions. Mean values of 20 repetitions in a population of 500 agents in 10 neighbourhoods facing 5 types of problems are shown.

In the lower-right corner of the plot we see the darkest grey tones indicating comparatively high levels of wealth. Values range from 43 to 62% percent of the wealth seen in our benchmark populations. This is a much higher wealth than in populations in which no one ever trust. Thus the good news is that even with only moderate precision of classification we reach levels of trust that lead to a substantial increase in wealth.

On the other hand, these levels of wealth are still below the wealth of populations in which agents have trustful local interactions. Thus given more realistic precision of classification there are bad news: We do not reach levels of wealth that could be obtained in large groups of agents that have trustful interactions. The plots shown here stem from simulations with 500 agents distributed among 10 neighbourhoods and 5 types of problems. However, we conducted simulations with different numbers of neighbourhoods and problems and this result turned out to be pretty stable. Our model, therefore, suggests that mechanisms of classification are likely to be insufficient to promote division of labour in large groups.

Besides these findings, it is interesting to see that higher precision of classification does not necessarily lead to higher levels of wealth. Given that local classification works quite reliable, we see that wealth increases as the precision of classification on the market decreases. This finding illustrates that agents in the model do not only face a social dilemma but that the model does furthermore imply a *coordination problem*. An agent's probability to find a suitable partner increases with the number of possible partners. Thus if an agents chooses to interact in the neighborhood, his opportunity costs are lowest his neighbors stay there as well. The same is true for interactions on the market. Worse classification on the market helps to solve this coordination problem. It leads to an decreasing attractiveness of market interactions and thus "forces" agents to stay in their neighborhoods. In this way it causes a decrease in opportunity costs in local interactions.

4.8 Conclusion

In the previous sections we presented a model an the development of economic wealth based on division of labour in large groups. The model formalizes some of David Hume's central ideas on this process and we presented the most important components.

In section 4.6 we learned that agents could gain a substantial increase in wealth if the follow a norm of trustworthiness. Thus the model demonstrates the importance of moral behaviour for the development of economics wealth based on division of labour in large groups. It was furthermore shown that the model captures the idea, that there are opportunity costs occurring if agents decide to interact only with partners they know.

Afterwards we demonstrated that agents can learned to be trustworthy even if their learning is success-driven. The degree trust and the resulting levels of wealth depends crucially on the reliability of the classification mechanism. Given that it is perfect, we see levels of wealth that match that of the benchmark population in which every interaction is trustworthy.

If a more realistic precision of classification is assumed then things are less optimistic. On the one hand, levels of trust can still be much higher than in populations in which no one trusts. This is the good news. The bad news is that these comparatively high levels of wealth are still below the wealth gained by populations in which agents have trustful local interactions. This finding indicates that trust in large groups can hardly be obtained by means of classification only. Perfect classification is an implausible assumptions and already small probabilities of mistakes lead to substantial decreases in wealth. Thus our work supports Hume's theory that division of labour in large groups requires the enforcing power of a government.

The development of government will be a major topic of our future work on the model. Another interesting issue that our results leads to is the implementation of the mental level of norms. It might be that the emergence of trust in large groups depends crucially on how agents mentally process norms of trust, i.e. the immergence of norms. Thus implementing the mental level of norms, using the techniques presented in later chapters, is a promising perspective for our research.[18]

[18] Though we are aware of the problem that cognitive architectures would only help in explaining the development of trust if their evolution could be explained as well.

Chapter 5 Empirical Context: Studying Emergent Social Order in Interaction: The Case of Wikipedia

Maria Xenitidou, Jens Villard and Klaus G. Troitzsch

Abstract

This chapter is based on empirical work conducted on Wikipedia as an emergent self-regulated and self-organised community. Two studies were conducted focusing on the discussion pages of Wikipedia articles. One study is about featured and controversial articles which cover special and restricted topics while the other focuses on discussions about the person and role of Sarah Palin. These studies cover a wide range of collaborative writing and discussing about collaborative writing and aim to support the design of a simulation architecture (EMIL-S) suitable for the wider study of normative mechanisms.

5.1 Introduction

In trying to understand how social order emerges, how people subsequently comply with sets of socially shared norms and how self-organisation and self-regulation is achieved, the Wikipedia was considered a prime example as an emergent self-regulated community. In particular, probing into the Wikipedia was expected to highlight the mechanisms through which the above take place. Therefore, the Wikipedia was selected in order to better understand:

How people influence one another and converge on common expected patterns of behaviour. Specifically,

> The role and contribution of norms[19] and rules[20] to self-organisation processes in communities which are voluntarily formed.
>
> The range and type of rules and norms used to self-regulate open global volunteer communities where there is little to no hierarchy and limited capacity for formal sanction;
>
> How these norms and rules are invoked, maintained and modified through communicative and administrative acts and the effectiveness of such acts;
>
> The relationship between goal, (social) context, environment and social structures and the exercise of individual agency in self-regulation in volunteer (online) communities.

Focusing on the Wikipedia has meant that the type of community was online, computer mediated, thus the environment itself unavoidably became enmeshed in the object of study as well as that of normative behaviour. This chapter starts with a description of the Wikipedia, a discussion of self-regulation in the Wikipedia and of the context/environment in which it occurs. It then shifts to how the particular case was framed, the methodology and results as well as a discussion on the particulars of (emergent, online) normative behaviour and the ways in which these contribute to the study of the emergence and immergence of norms.

5.2 The Case: Wikipedia[21]

The predecessor of Wikipedia was Nupedia, a Web encyclopaedia founded by Jimmy Wales with Larry Sanger on the principles of the open source movement. Nupedia is said to have failed due to relying on experts to contribute content. The WikiWiki software platform was introduced as an experiment in 2001.

[19] Norm refers to “a prescribed guide for conduct which is generally complied with by the members of society” (Ullmann-Margalit, 1977; see Chapter 2)

[20] Rules are treated here as the micro tools through which norms are sustained and executed. While the term “rule” is used throughout this chapter, the definition of rules corresponds to that of “meta-norms”: “general rules telling agents how to reason, decide upon and apply specific norms” (Chapter 2) which in the case of Wikipedia refer to a rule present in a Wikipedia guideline, etiquette guide or style guide (see Appendices).

[21] Sections 5.2 and 5.3 draw on Goldspink (2008b).

The openness that this enabled attracted increasing numbers of contributors and quickly developed a life of its own as it is observed today.

Therefore, Wikipedia is based on Web 2.0 technology. According to Wikipedia[22],

> *Web 2.0 refers to a perceived second generation of web development and design, that facilitates communication, secure information sharing, interoperability, and collaboration on the World Wide Web. Web 2.0 concepts have led to the development and evolution of web-based communities, hosted services, and applications such as social-networking sites, video-sharing sites, Wikis, blogs, and folksonomies.*

The main idea of Web 2.0 technology is that of user generated content[23]. The interesting questions that may arise in this context are: How is it possible to coordinate the activity of many contributors without the hierarchical and credentialist controls typically employed for media production? How is this activity internally regulated as apart of an Open Source environment aimed to respond to the encyclopaedic genre on-line? In other words, how do contributors (voluntarily) not only conform but also produce?

Wikipedia is a good case for exploring answers to such questions. Wikipedia is a successful example of an open collaborative process of great magnitude, requiring the precision and accuracy typical of the encyclopaedic genre, that actually works, producing credible encyclopaedic articles (Giles, 2005). We now turn to consider the above questions.

5.3 Social Self-Regulation in Wikipedia

The use and enforcement of principles and rules has been an ongoing issue within the Wikipedia community with a division emerging between the founders and within the wider community about whether rules were necessary and how they should be policed. The power to police rules or impose sanctions has always been limited by the openness of the Wiki technology platform. Initially Sanger and Wales, were the only administrators with the power to exclude participants from the site. In 2004 this authority was passed to an Arbitration Committee which could delegate administrator status more widely. The Arbitration Committee is a mechanism of last resort in the dispute resolution process, only dealing with the most serious disputes. Recommendations for appointment to this committee are made by open elections with appointment the prerogative of Wales.

In the early stages Sanger argues the need was for participants more than rules and so the only rule was “there is no rule”. The reason for this, he explains, was that they needed to gain experience of how Wikis worked before over prescribing the mechanisms. However, “*As the project grew and the requirements of its success became increasingly obvious, I became ambivalent about this particular ‘rule’ and then rejected it altogether”* (Sanger, 2007). However, in the minds of some members of the community, it had become “the essence” of Wikipedia.

In the beginning, complete openness was seen as valuable to encourage all comers and to avoid them feeling intimidated. Radical collaboration – allowing everybody to edit everyone’s (unsigned) articles – also avoided ownership and attendant defensiveness. Importantly, it also removed bottle necks associated with “expert” editing. That said, the handpicking of a few core people is regarded by Sanger as having had an important and positive impact on the early development of Wikipedia. Sanger argues for example “*I think it was essential that we began the project with a core group of intelligent good writers who understood what an encyclopaedia should look like, and who were basically decent human beings”* (2005). In addition to “seeding” the culture with a positive disposition, this statement highlights the ultimate purpose of Wikipedia – establishing a style consistent with the encyclopaedic genre as a model to shape the subsequent contributions of others.

[22] Special page at the time of the study.

[23] Web 2.0 is radically different from Web 1.0 which consisted of platforms to upload accessible information and Web 3.0 or semantic web which is seen as shifting the primacy back to the platform so that it “reads”, “understands” and “links” the content uploaded.

Sanger argues that in the early stages "force of personality" and "shaming" were the only means used to control contributors and that no formal exclusion occurred for six months, despite there being difficult characters from the beginning. The aim was to live with this "good natured anarchy" until the community itself could identify and posit a suitable rule-set. Within Wikipedia rules evolved and as new ones were needed they were added to the "What Wikipedia is not" page. Wales then added the "Neutral Point of View" (NPOV) page which emphasised the need for contributions to be free of bias. The combination of clear purpose and the principle of neutrality provided a reference point against which all contributions could be judged[24]. However, the success of Wikipedia was not attributed to these rules[25].

Bryant et al. (2005) suggest that newcomers are introduced to these rules by serving a kind of apprenticeship, in order to learn and conform with the rules and norms. In particular, they argue that this is evident in new editors of Wikipedia initially undertaking minor editing tasks before moving to more significant contributions, and possibly, eventually, taking administrative roles. However, the authors tend to project a rather idealistic view of involvement overlooking a key attribute of the Wiki environment – newcomers have the same rights as long standing participants and experts and this mechanism for socialising newcomers can be and frequently is bypassed owing to the primacy of the task at hand.

Apart from "learning" to self-regulate as suggested above, self-regulation may be attributed to personal goals, interests and gains such as the quest for attaining reputation on the part of contributors. Certainly, in some Open Source environments (such as Open Source Software) it is possible to gain reputation which may be usable in the wider world. However, in the Wikipedia environment there is no list of contributors to which an editor can point as evidence of their contribution (although they can self-identify their contributions on their user page). In addition, anonymity impedes gaining reputation (although identification of contribution and behaviour can be traced through the token names or pseudonyms contributors use, especially if one focuses on the Discussion pages). Contributions are, therefore not immediately attributable to "natural" or "legal" persons. Other reasons of involvement and commitment to the task may be identification with product, community, values or personal satisfaction (see discussion of results below).

5.4 Methodology

In Wikipedia there are two classes of activity: editing and discussion about editing. This study was not concerned with the editing activity but with the discussions which help to coordinate it as it was hypothesised that the emergence or not of social order is to be verbally traced in discursive interaction based on the assumption that norms are stored in a sentence-like format (see Chapter 15).

Insight into this was gained by examining the Discussion pages which accompany many of the articles rather than the articles themselves. The activity on the Discussion pages comprises of "utterances[26]" between contributors about editing activity and the quality of product. On the face of it then, these pages were expected to provide a fertile source of data to support analysis of how (social) order is achieved in Wikipedia. Within these pages we expected to see attempts by editors to influence[27] the behaviour of one another through the only means available to them – communicative acts (cf. Habermas, 1976). We anticipated that these may exhibit some regularity which would allow us to examine both the range and type of events that led to the explicit invocation of rules and norms and which revealed emergent influence patterns which were themselves due to normative beliefs. We wanted also to examine what conventions prevailed and how these compared and interacted with the goal of the community and its policies. A convention is defined here is a behavioural regularity, i.e. a practice or procedure widely observed by members of a given social network, based on (i) the agent's goal of conforming to that behaviour in order to act like the others, (ii) the mutual expectation that the others will conform to that behavior as well (cf.

[24] This could be used to explain the conclusions of this study (see Section 5.6).

[25] But to the fact that contributors complied.

[26] We employ the term "utterance" to refer to what is being said, voiced or uttered. In other words, an utterance constitutes a unit of speech (Austin, 1955). For coding purposes see Appendices.

[27] In social psychological research (see inter alia Turner, 1991) social norms are tied to (social) influence - the way people affect the thoughts, feelings and behaviours of others.

Gilbert, 1981, 1989; Lewis, 1969; Sugden, 1986/2004, 1998; Young, 1993, 2006). Policies include explicit codes of conduct as well as guidelines (etiquettes) and principles.

Two studies were conducted to identify the mechanisms which underpin the emergence of social order and the attainment of self-organisation and self-regulation in Wikipedia, as a volunteer (online) community. The aim was to specify the mechanisms involved in order to support the design of a simulation architecture (EMIL-S) suitable for the wider study of normative mechanisms. The first study preceded and informed the focus of the second. The second study aimed to validate the first and to provide further insights into the emergence of normative behaviour.

Data collection focused on the Wikipedia Discussion pages and data analysis was premised upon the principles of Speech Act Theory. Speech Act Theory was employed as a result of treating mental states as intentional, having a causal relationship with speech acts in reinforcing each other and being recognised by the listener through communicative acts (Searle, 2002). The emphasis on “acts” within speech act theory is based on the premise that language is performative and, therefore, language utterances are actions rather than statements (Austin, 1955). Since early Speech Act theory Austin (1955) had argued that all utterances are locutionary, which refers to the act *of* saying something (form of an utterance); all locutions have illocutionary force, which refers to the act *in* saying something (intent of an utterance); and are perlocutionary in producing consequential effects upon the feelings, thoughts or actions of the recipients. The study, therefore, focused on both *literal – what is being said* - and *pragmatic – what is the intent of what is being said* - (Stiles, 1992) meaning.

In addition, the importance of the “uptake” (see Potter, 2002) of the utterance has been also noted in speech act theory (see Searle, 1969, 2002). For Habermas (1976), a successful speech act is one in which the listener both comprehends and accepts the validity claims made by the sender and thus enters into the intended relationship. The tests of validity include comprehensibility, truth, sincerity and rightness. Thus for Habermas, a speech act only serves to support the maintenance of effective communicative exchange to the extent that it is held as valid by a listener. The intrinsic openness of Wikipedia and the absence of formal means of compulsion or sanction mean that the majority of exchanges will be communicative acts – i.e. bounded and influenced by norms rather than through the exercise of formal authority, power or coercion.

The data considered for the first study was randomly selected from the Wikipedia Discussion pages associated with either Controversial or Featured articles. A controversial article is one which is constantly being re-edited or is the focus of *edit warring* due to its content or the issue it is dealing with. A featured article *is the polished result of the collaborative efforts that drive Wikipedia*. Featured articles are considered to be the best articles in Wikipedia, as determined by Wikipedia’s editors[28].

It should be noted that at the time of the study (May/June 2007) there were 583 articles identified by the Wikipedia community as controversial and approximately 1900 as featured. This may be considered as a preliminary indication that the norm was towards producing featured articles and its adoption rate was higher than not.

Analysis was conducted on the discussion of a sample of Controversial (N=19) and Featured (N=11) articles. Controversial articles were chosen as they were more likely to involve the need to resolve conflict and hence place greater demand on effective normative regulation; Featured articles by contrast may be so rated due to the attainment of a higher level of consensus among participants. The broad categories[29] that these 30 articles sampled may be classified into are:

- Science
- Medicine
- Politics

[28] At present, there are 2,662 featured articles, of a total of 3,074,404 articles on the English Wikipedia which constituted the case of the study.

[29] It is acknowledged that this categorisation may frame understanding from the types of articles to the variables and their values.

    - Human Rights
- Philosophy
- History
    - Biographies
- Sports
- Media
- Arts
    - Music

The most recent three pages of discussion were selected for analysis from each Discussion page associated with the article included in the sample. Both qualitative and quantitative analysis was conducted in a fine grained analysis of behaviours and speech acts. Qualitative analysis was based on the Verbal Response Mode (VRM) taxonomy (Stiles, 1992; see Table 1 below) and involved the use of the Open Source qualitative analysis software WeftQDA and MaxQDA. VRM is very attractive where there is a need to capture many of the subtleties of natural language use that derive from and rely on the intrinsic flexibility and ambiguity of natural language yet map them to a more formal or axiomatic system needed for computer simulation.

Mode	Descriptors
Disclosure	Informative, unassuming, directive
Edification	Informative, unassuming, acquiescent
Advisement	Informative, presumptuous, directive
Confirmation	Informative, presumptuous, acquiescent
Question	Attentive, unassuming, directive
Acknowledgement	Attentive, unassuming, acquiescent
Interpretation	Attentive, presumptuous, directive
Reflection	Attentive, presumptuous, acquiescent

Table 1. Descriptors associated with Verbal Response Modes (Source: Stiles, 1992, p. 63)

A range of additional codes were applied including the style and topic or subject of communication, explicit invocation of norms or rules and the associated deontic[30] and trigger, whether a listener accepted (or validated) an utterance in terms of its illocutionary force or intent; and the ID and registration status of the person making the utterance. Three thousand six hundred and fifty-four (3654) utterances were coded. Quantitative analysis involved re-processing the coded utterances such that each utterance constituted a case and each applied code a variable associated with that case. This data set was then analysed using SPSS and MLwin.

The data considered for the second study were purposefully selected from the Wikipedia Discussion pages. Analysis was conducted on the discussion of one Controversial article: Sarah Palin. The article was on Sarah Palin, an American politician who served as Governor of Alaska (2006-8) and stepped down in 2008 in order to run as the Republican candidate for Vice President of the United States. The article was placed on article probation for containing controversial material on the biography of a living person which raised an edit-war as well as heated exchanges between contributors in the discussion page. Two topics (“Creationist?” and “Rape Kit Material”) were selected from the same Archive (44) of discussions on the grounds that the same

[30] A deontic is basically a way of partitioning situations between good/acceptable ones and bad/unacceptable ones (see Chapter 2).

persons contributed to these discussions. The topics “Creationist?” and “Rape Kit material” belong to the topics with the most user actions with 74 and 189 entries respectively. In both topics 39% of the users are the same. This offers the possibility to analyse the behavior of a user irrespective of the topic. Furthermore, the behavior of the users in these discussions seems to be much more controversial than in the first study.

On the discussion page of the topic “Creationist?” there are 11 users dealing in 74 entries with whether a possible relation between Sarah Palin and the Creationists should be included in the article or not. This discussion is driven by speculations arising from a discussion panel which took place on national television.

On the discussion page of the topic “Rape Kit material” the main discussion point is the exclusion of texts and their content. The topic attends to the cost and charge of Rape Kits (rape kits of evidence collected in hospitals for law enforcement use) in conjunction with Sarah Palin. Beyond that, there are “heated” discussions about some actions of several users who in spite of having a “democratic” agreement delete some text fragments.

Qualitative analysis was based on coding for six variables, primarily inspired by the first study. These were: the item number, the user’s name or ID, the communication style, the target or trigger (of the reaction), norm invocation or rule citation and the item number of the responded entry. Similarly to the first study, communication style was coded for being negative, positive, or neutral in response to the previous entry; norm invocation was coded if the contributor referred to a social standard and no rule was coded; rule was coded if the contributor cited a Wikipedia rule; the target of the reaction was coded in terms of the content or the form of the entry preceding it, i.e. whether the reaction was to the topic or subject or to the behaviour or style of communication exhibited; the reaction/feedback was coded for being uttered in a negative, positive, or neutral communication style. This data set was then analysed using SPSS.

5.5 Results

Quantitative analysis in the first study indicated that the “style of communication” in the sampled pages was primarily “neutral”, while there was a statistically significant correlation between the article group (Controversial *vs* Featured) and the broad style of communication – negative style figured higher in the discussion pages of controversial articles while positive style figured higher in the discussion pages of featured articles (see Goldspink, 2008b).

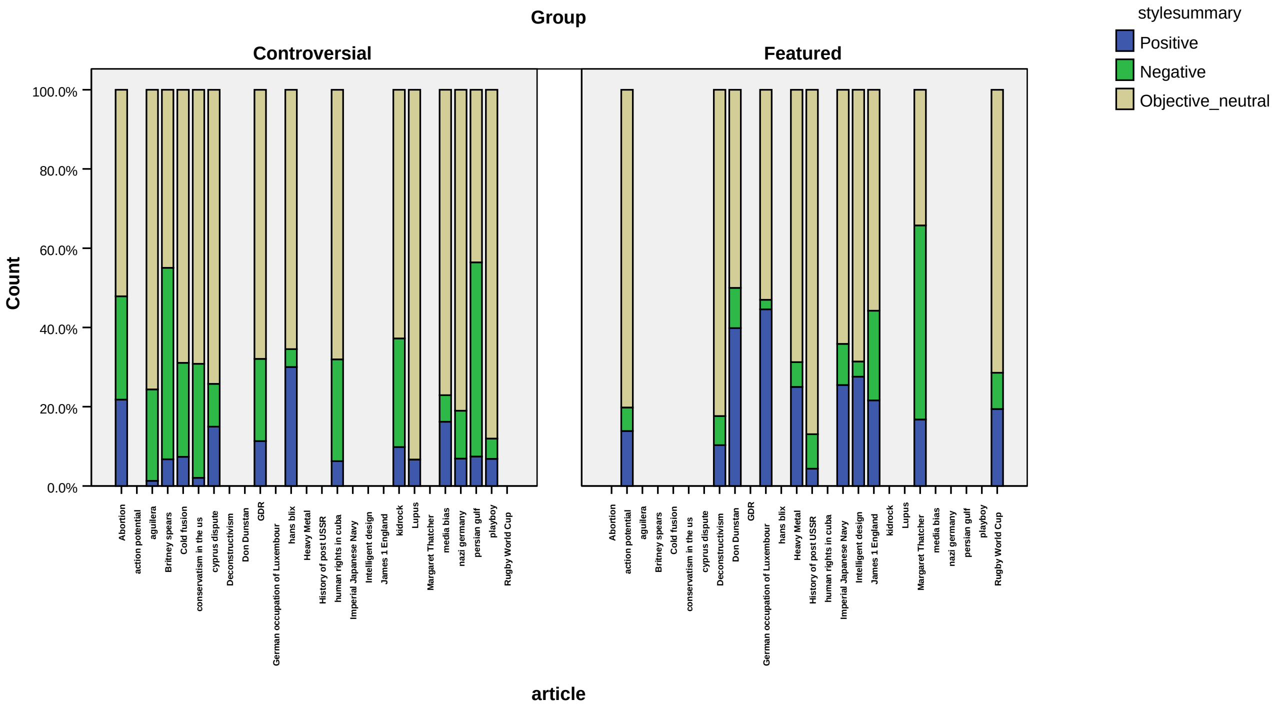


Figure 6. Percentage of style by article by group

In terms of validation, 50% of all utterances were accepted without question[31]. A further 18% were explicitly accepted by at least one editor; 11% were explicitly rejected and a substantial 22% were ignored (see Figure 7 below). This indicates that for the most part (3/4) communication was effective as interacting contributors entered the intended relationship in their discussions.

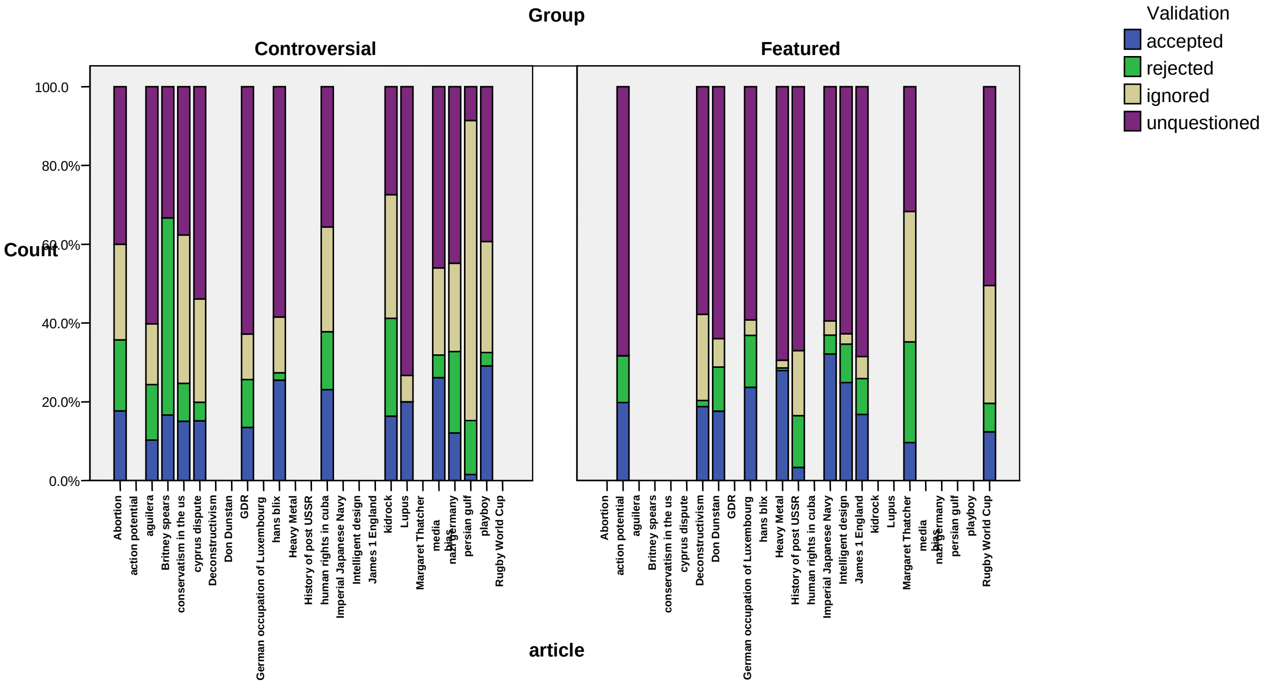


Figure 7. Percentage of validation by article by group

Overall, only 5.2% of explicit norm or rule invocation was identified in the utterances of the discussion pages sampled. Rules were most commonly invoked in response to neutral style of communication (63.9%) followed by 27% in response to a negative style and only 9% in response to a positive style. By comparison, norms were most commonly invoked in response to negative style utterances (53.2%) followed by neutral (44.2%) and then positive (2.6%).

Rule invocation was most likely to be triggered by *article form* (44.9%), an *edit action* (22%), an *article fact* or a *person's behaviour* (both 16%). A norm was most likely to be triggered by a *person's behaviour* (35.6%), an *edit action* (23.3%), *article form* (21.9%), or *article fact* (19.2%). Almost three quarters (73.6%) of rule invocations and over two thirds (61.3%) of norm invocations had the implicit deontic of "it is obligatory".

However, multilevel analysis suggested that the style of communication was associated to individuals rather than the topic of discussion or the subject of the article. In view of that, and since there was a correlation between controversial articles and negative style and between negative style and norm invocation, the second study then concentrated on discussions of a controversial article on the basis of them being held between the same individuals.

The communication style in the two topics analysed in the second study ("Creationist?" and "Rape Kit Material") was for the most part a negative one (Figure 8): 50.7% of all reactions were dismissing, contemptuous or irreverent; 44.6 % in "Creationist?" and 56.7% in "Rape Kit Material". This outcome was supported by the analysis of negative reactions/feedback to negative communication style, which was observed in 25.3% of the negative reactions. In other words, 25.3% of negative style utterances received reaction/feedback in a negative style as well. By contrast, there was only 12.6% reaction/feedback in a positive or neutral communication style to a negative coded one. Finally, 9.1% off all entries were ignored.

[31] Positive utterances were more likely to be accepted without question (61%) compared to negative (21.7%) and neutral (54.4%).

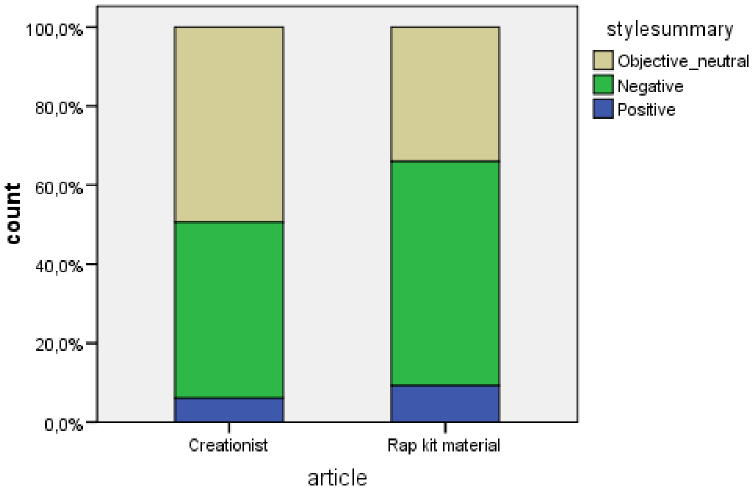


Figure 8. Percentage of style by article

In the two topics analysed as part of the Sarah Palin case, a norm or a rule was invoked in 22.2% of all of the reactions/feedback. In 61.4% of these cases a norm was invoked. A rule was cited in 38.6% of norm or rule invocations. A rule or a norm invocation was mostly triggered by the topic of discussion (total 82.4% of which 52.9% norm invocation and 29.4% rule invocation) and was typically invoked by a negative communication style (total 66.7% of which 41.7% norm invocation and 25 % rule invocation). This diverges from the results of the first study where rule invocation was mainly by article form and edit action while norm invocation was triggered by a person's behaviour, an edit action, article form and only lastly by article fact or topic. Nevertheless, discussion on the "Rape Kit Material" is heavily based on edit actions.

In addition, the purposive sample of the second study enabled further probing. Namely, it was observed that as a response to a norm invocation an anti-norm might be invoked. An anti-norm is defined here as a norm expressed against another norm (antithetical) or which potentially acts in another direction to a preceding norm (function). This was observed in 40.7% of the cases in which a norm was invoked. An example of anti norm invocation is presented below:

Norm:

> *"Be that as it may, if a politician says in different ways that they support a certain policy then it becomes increasingly hard for us to say that they don't support it."*

Anti-Norm:

> *"But again, it doesn't go anywhere in Wikipedia without a good source, and even with a source it would have to be appropriately contextualized."*

An anti-norm is typically triggered by a negative communication style (81.8%) and to a lesser extent by neutral communication style (18.2%). Ninety-one percent (91%) of anti-norm invocations are about the topic of discussion (see Figure 9).

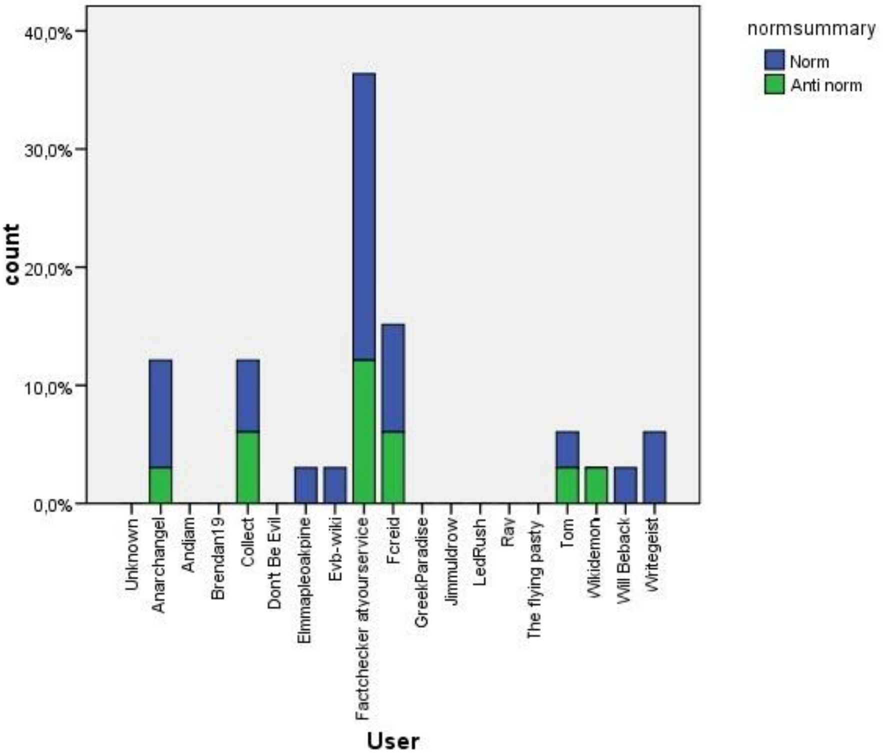


Figure 9. Percentage of norm and anti-norm invocations

Moreover, the second study found that 18.2% of all reactions/feedback were composed by "a discussion leader"; thereof 72.2% are negative ones and 75% refer to a topic. The discussion leader responds to 66.6% of all users. Fifty-eight percent (58.3%) of the leader's conversations took place with merely 11% (2 of 18) of all users. Thirty-seven percent (37.5%) of all norm invocations and 36.6% of all anti-norm invocations are produced by the leader (Figure 9 - User: Factchecker atyourservice). This enhanced the finding of the first study which indicated that individual behaviour was a stronger determinant of the style of communication than the topic of discussion or the subject of the article; the second study found that not only individual behaviour matters but also that the behaviour of a few, "core" individuals significantly contribute to the regulation of communication in controversial situations (minority influence).

Qualitative insights provided by the first study indicated that while there was no statistically significant difference in the degree to which either norms or rules were invoked between the Featured and Controversial articles, there was a qualitative difference in the role norm and rule invocation appeared to play. In Controversial discussions, (social) norms and rules were most likely to be invoked against the behaviour of an editor who was of a different view[32] while in Featured sites, norms and rules were somewhat more often used by the editor as a reflection on their own contribution – i.e. involved a level of self-check.

In terms of the illocutionary force or intent of the utterances, Edification in service of Edification (EE) was the most frequent form of utterance in the Wikipedia sample of the first study – 37% of all utterances were of this mode. The Edification mode is defined as deriving from the speaker's frame of reference, making no presumption about the listener and using a neutral (objective) frame of reference shared by both speaker and listener. This mode is informative, unassuming and acquiescent. It reflects attempts to convince by neutral argument. An example would be "That edit was made last week". The second most common mode was that of Disclosure in service of Disclosure (DD). Disclosure is defined as being from the speaker's experience, making no presumption, but being framed using the speaker's frame of reference. This is summarised as informative, unassuming but directive. Unlike EE mode, DD mode represents an attempt by

[32] This was shown in the second study.

the speaker to impose or have the listener accept the speaker's frame. Twelve percent of all utterances adopted this form. An example would be "I don't know much about this topic". The third most common mode was Disclosure in service of Edification (DE). The DE mode represents an utterance which is from the speaker's frame of reference but as if it is neutral or from a shared frame. Eight percent of all utterances used this mode. This is a somewhat neutral mode where the speaker offers clearly labelled personal knowledge as information. An example would be "I believe it occurred in 1987". The fourth most common mode was Advisement in service of Advisement (AA). AA mode represents speech from the speaker's experience, which makes presumptions about the listener and adopts the speaker's frame of reference. It can be summarised as informative, presumptuous and directive. An example would be "You should change this immediately". Approximately 7% of utterances were in this mode. A further 12% of utterances had the directive pragmatic intent of advisement masked by a less presumptuous form – Edification or Disclosure ("It should be changed immediately" or "I think it should be changed immediately"). Therefore, the intent of utterances in the discussion pages of the sampled articles indicates a tendency towards convergence to a view using mainly neutral argument.

5.6 Discussion, Conclusions and Further Research

Overall, results indicated that there was little *explicit* invocation of rules or norms in the discussion pages sampled (5.2% in the first study, 22.2% in the second study), while regulation was generally achieved. There was a noteworthy difference of norm and rule invocation between the two studies as the second sought to probe into norm invocation by purposefully selecting a controversial article for analysis. In comparison, the two empirical studies on Wikipedia as a case study of emergent social order – self-regulation and self-organisation – showed that the communication style in controversial situations is mainly negative. Negative communication style provokes reaction/feedback in negative communication style and it is the style in which explicit norm invocation is mainly observed. Norm invocation is usually followed by anti-norm invocation, which is the invocation of another norm in response to the preceding one. This negotiation is significantly mediated by individual behaviour and dominated by the persistence of a core, few.

Generally, apart from the cases of explicit norm and rule invocation, it seemed that behaviour accorded to sets of conventions or shared understandings which (minimally) accommodated what needed to be done to satisfy the task at hand in a context of divergent personal goals. In addition, in the "snapshots" sampled in the first study, there was a lack of evidence of "active" negotiation of expectations and standards and of "active" convergence of behaviour towards a norm. As a result, it seemed that within the discussion pages sampled there appeared to be little obvious or explicit norm innovation, evolution, adaptation or extension. After further probing as mentioned above, the second study showed that in controversial situations, explicit norm and rule invocation was higher and correlated with communication style but also that anti-norms – offering a counter norm – and the behaviour of a core few "regulate" order in Wikipedia.

Regardless of explicit norm and rule invocation, there appears to be an emergent order. This may be explained in terms of neutrality and pro-sociality as norms, with the addition of other pre-existing norms in editors' normative board – as specified by EMIL-A the normative of N-board is an archive of long term memory where active norms are stored, arranged according to their salience in terms of the frequency of being adopted (see Chapter 9) – and the norms which govern the task at hand[33]. The overall neutrality in the mode of communication and behavioural acts may be first and foremost attributed to the fact that we took a "snapshot" of Wikipedia discussions at a time when the etiquette and the NVOP had already become internalised and diffused in "routine" operations of collaborative writing. Thus, neutrality may be a consequence of the evolution of Wikipedia into a self-regulated community. Second, due to the pre-existing, normative style of encyclopaedic writing which is brought from "outside" Wikipedia, from external understandings of the task and which is ultimately "neutral". Third, due to conformity being a norm itself in collaborative tasks as in social settings/society at large and which is an implicit understanding of how things

[33] It should be noted here that while pro-sociality and pre-existing norms are inductively used to explain regularities in Wikipedian's behaviour, the task at hand is a fact and neutrality – neutral style and behaviour – is an empirical finding. With the exception of explicit normative behaviour in controversial situations, all other (normative) behaviour seems established and is implicit.

are “done” in these settings. Fourth, anonymity and the open nature of the environment may contribute to feelings of suspicion and uncertainty and the lack of trust. Thus, conformity and compliance may function to create stability and predictability in a potentially volatile environment (see Sherif, 1936). Fifth, the mean, average opinion may be fostered as an approximation to information from a Neutral Point of View. Therefore, neutrality seems established as a norm.

Pro-sociality is assumed in the “type” of individuals attracted by voluntary collaborative tasks, the similarities in profile they may share and the respective dispositions they may have towards different articles. Pro-social behaviour, especially in the case of long term editors, may be a consequence of the volunteer nature of Wikipedia and the level of commitment required. In addition, such a pro-social behaviour, which is not personalised, may be due to pre-existing social norms, external to the Wikipedia, brought into and instantiated in the Wikipedia application (see below). Pro-sociality therefore, may be a pre-existing norm these individuals bear, which in addition to their own pre-existing identities brought to the task, account for the way Wikipedia works. Finally, collaborative writing based on the encyclopaedic genre for open access and public use may be determined by both producing the optimal result but also one that is satisficing to contributors. This implies that the overall goal is the primary orientating point. However, the lack of evidence on negotiation to achieve consensus is indicative of a brief encounter between different and established milieux which struggle to find common understanding on the task at hand rather than of a community committed to a general common goal or meta-goal (Becker and Mark, 1997). This might suggest that the primary influence of the utterance strategies employed by individuals derive from their extra-Wikipedia identity, from their wider life as their primary environment – not the immediate environment of the Wikipedia. If this were the case then we would expect to see speech acts which are a minimal accommodation: are minimally concerned with establishing understanding and aimed at a pragmatic accommodation or satisficing of presenting demands from different editors. Certainly this is one way of interpreting the patterns observed in the data. Similarly we would expect to find either that local norms and rules had little effect and that social behaviour was primarily influenced by the socialised norms consistent with the editors’ background – that is to say – brought in from outside the Wikipedia, or that the two are enmeshed in a – by now – “established” way of doing things in Wikipedia. In this context, a little explicit norm/rule invocation may suffice in the context of “implicit”, shared understandings and conventions regulated by a “core” of contributors and the “nature” of the task at hand, to encourage compliance and/or drive non-compliers away.

Finally, the empirical studies on Wikipedia aimed to highlight the mechanisms involved in emergent social order, self-regulation and self-organisation, and ultimately support the design of a simulation architecture (EMIL-S) suitable for the wider study of normative mechanisms. In reflection of the results produced and their theoretical embedding the following two sets of hypotheses were formulated:

“It’s all in the editing”

Chance association hypothesis: Chance variations in the pro-sociality of individuals drawn to edit a common article at a particular point in time, could explain differences in the effectiveness of the coordination process and hence the quality of outcome.

Primed attraction: In Wikipedia though, editors are not randomly assigned but rather self-select articles which are of interest. It seems reasonable that some attributes of the article such as pre-existing style or neutrality and balance may influence who is attracted.

Wisdom of crowds: Given sufficient diversity and independence of opinion, decentralization of knowledge and a means to aggregate or bring these diverse opinions together, crowds can out-perform experts in many areas of prediction and problem solving.

“Norms influence discussions”

Diffused effect: This hypothesis assumes that norms do work to improve coordination by a directive mechanism but allows for several alternative pathways. Namely, rule and norm communication may happen less by explicit invocation and more by more subtle means.

Group salience: A subject will use cues in various artefacts, including the article itself and Discussions and Talk pages, to read the group environment and to identify other subjects with particular groups. Where the subject does not identify with a group, norms invoked by members of that group will have little or no affect on his/her behaviour and the individual will act in a manner consistent with his or her own individual attitudes or group appropriate norms.

The above conclusions and hypotheses were built into the simulation of the Wikipedia scenario which already implements the whole history of Wikipedia, from the first article ever written (Chapter 15). In particular, the empirical work on Wikipedia contributed to enhancing the social aspects and processes of normative behaviour in EMIL-A, in addition to the cognitive infrastructure of agents and the mental processes of normative behaviour (Chapter 9). Thus, the Wikipedia scenario was extended (i) to take into account varying (3) levels of pro-sociality and (ii) respective disposition towards different articles, (iii) to associate article quality to edits and links between articles (where the mean opinion could approximate the neutral position), and (iv) to include group formation or allegiance on the basis of pre-existing dispositions (i.e. the normative board or N-Board) and (v) group salience in terms of obedience (norm adoption).

5.7 Appendices

5.7.1 Appendix One: Wikipedia Case Study One Code Frame and Definitions

Codes are applied at three different levels:

Utterance: defined as “a simple sentence, an independent clause, a non-restrictive dependent clause, an element of a compound predicate, or a term of acknowledgement, evaluation or address”.

Contribution: defined as a single contribution by an individual editor.

Thread: A series of exchanges which represent responses to prior contributions with a related subject.

Communicative Style

Communicative style codes are applied at the level of utterance.

The communicative style codes are intended to capture

1. The general valence of an utterance
2. The attitude or style expressed by the utterance

If an utterance is **negative**, it is coded to negative and also to the style of negative.

- Abusive: To assail with contemptuous, coarse, or insulting words;
- Aggressive: characterized by enmity or ill will, threatening
- Contemptuous: exhibiting lack of respect; rude and discourteous
- Defensive: attempting to justify or defend
- Dismissive: showing indifference or disregard; not having or showing interest

If an utterance is **neutral/objective** then it is coded to this node only.

If an utterance has a **positive** intent then it is coded to positive and also to the style of positive.

- Affirming: To support or uphold the validity of
- Apologetic: Self-deprecating; humble
- Encouraging: furnishing support and encouragement
- Placative: To allay the anger of, especially by making concessions; appease

Editor_Status

Editor status code is applied at the level of a contribution.

Unregistered: is applied if the editor has no username at the end of the contribution but that contribution terminates with only an IP address.

Registered is applied if the editor has a username. In this case the contribution should also be coded to the username under Editor_ID. This may require creation of a new node of that name.

Normative behaviour

The normative behaviour codes are applied at the level of utterance.

If, in an utterance an editor:

- Specifically invokes a rule present in a Wikipedia guideline, etiquette guide or style guide, and the Editor links the invocation to that rule (e.g. "at Wikipedia we always provide a source to support any fact") then code to the associated **rule_descriptor** (create new node where necessary).
- Specifically invokes a norm which is a) not the subject of an existing rule or b) in the invocation, the editor does not link to a specific Wikipedia practice or rule (even if one exists) but rather refers to a wider standard (e.g. "it is not good practice to play the man rather than the ball"), then code to **norm_descriptor**. This may require creation of a new node with an associated descriptive name for that norm.

Code to the associated **deontic** operator

- it is impermissible that
- it is non-obligatory that
- it is obligatory that
- it is optional that
- it is permissible that

Code to the appropriate **trigger**.

- Administrative_action
- Article_Fact
- Article_form_presentation
- Edit_action
- Person_behaviour

Subject of communication

Subject of communication is coded to the level of utterance.

This is the subject of the utterance. It may need to be inferred from the context.

- About_Administrative_action
- About_Edit_Action
- About_Editor_person
- About_person_behaviour
- Article_Fact (this includes reference to sources).
- Article_form_presentation (includes presentation, structure, location, format, style etc)

Thread

The thread code is applied at the thread level. Contributions linked to a single distinctive thread are coded as a block.

Validation

Validation is coded at the level of utterance. To apply the code it is necessary to read ahead and to determine if the utterance was:

- **Accepted**: by being met with an acknowledgement, explicit acceptance or implicit acceptance (e.g. by taking the action asked for)
- **Rejected**: by being explicitly questioned or rejected as invalid, inappropriate or where the motive of the speaker for making the utterance is questioned.

- **Ignored**: No response discernable either explicitly (i.e. there was no explicit questioning or rejection of the utterance) or implicitly (i.e. no evidence of a requested action having been undertaken).
- **Unquestioned** where the conversation proceeds without specific reference to the utterance but it is apparent from the subsequent conversation or action that the utterance was accepted.

VRM

VRM codes are applied at the level of utterance.

An utterance is coded to the appropriate **mode** descriptor twice: once for the **literal** meaning and once for the **pragmatic** intent.

Descriptors associated with Verbal Response Modes

Mode	Source of Experience	Presumption	Frame of reference	Descriptors
Disclosure	Speaker	Speaker	Speaker	Informative, unassuming, directive
Edification	Speaker	Speaker	Other	Informative, unassuming, acquiescent
Advisement	Speaker	Other	Speaker	Informative, presumptuous, directive
Confirmation	Speaker	Other	Other	Informative, presumptuous, acquiescent
Question	Other	Speaker	Speaker	Attentive, unassuming, directive
Acknowledgement	Other	Speaker	Other	Attentive, unassuming, acquiescent
Interpretation	Other	Other	Speaker	Attentive, presumptuous, directive
Reflection	Other	Other	Other	Attentive, presumptuous, acquiescent

VRM Principles

VRM principles are used to identify the VM modes. There are three principles, source of experience, presumption and frame of reference.

Source of Experience: Who's experience is the topic of utterance?

Speaker: Informative: utterance based on speakers personal experience, reveals feelings or things he/she knows.

Other: Attentive, questioning or describing and/or, reflecting others experience, inviting contribution or keeping space open for other to comment.

Presumption: Does the speaker need to presume knowledge of other?

Speaker: unassuming, signals of receipt of message but not agreement/disagreement or comparison. Concerns objective information and hence no need for presumption.

Other: presumption made about others experience or intention – how he/she is, was, will be or should be. Also utterances that seek to guide the others behaviour (i.e. to impose a view or compel an action). Includes only utterances where the presumption is necessary to the meaning. In order to agree or disagree or compare presumption is necessary unless it is about objective information held by the speaker.

Frame of Reference: Would listener have to read the speakers mind?

Speaker: reference is made to own constellation of meaning – directive. Listener would need to be able to read your mind to verify. Seek to impose speakers view: Includes personal perceptions, intentions, thoughts, feelings and value judgements.

Other: Acquiescent: reference is made to a common or shared set of meanings. Objective – external to the speaker, placeholders are objective/neutral.

Ask who gets to say what is true – code here if objective or other.

5.7.2 Appendix Two: Wikipedia Case Study Two Code Frame and Definitions

A user is a person who edits/distributes text or who discuss about an article at Wikipedia.

An entry is a collection of sentences written by one user.

A reaction is a response entry to another entry.

User

Known User: is a user who signifies her or his entry with a short signature. She/he can be identified by her/his username.

Unknown User: is applied if the editor has no username at the end of the entry but the entry is identified by intend or by a timestamp.

Communicative Style

If a reaction is **negative**, it is coded to the communication style of negative.

- Aggressive: characterized by enmity or ill will, threatening
- Contemptuous: exhibiting lack of respect; rude and discourteous
- Dismissive: showing indifference or disregard; not having or showing interest

If an utterance is **neutral/objective** then it is coded to this node only.

If an utterance has a **positive** intent then it is coded to positive and also to the style of positive.

- Affirming: To support or uphold the validity of
- Apologetic: Self-deprecating; humble
- Encouraging: furnishing support and encouragement

Target of a reaction

If the reaction aims to the form of the discussion, **style** is coded.

If the reaction aims to the topic of the discussion, **topic** is coded.

Norm

Rule is coded, if a user cites a Wikipedia netiquette or behavioural code.

Norm is coded, if a general social standard is consulted.

Chapter 6 Relationship to Other Thought: State of the Art in Modelling Norm Emergence

Martin Neumann

Abstract

This chapter analyses the current state of the art in normative agent-based social simulation models. Broadly speaking, two approaches can be distinguished: on the one hand models that are inspired by the conceptual terminology of game theory and on the other hand models that are based on architectures of cognitive agents with some roots in Artificial Intelligence. Opportunities and drawbacks of these approaches are identified. The former class of models focus on norm dynamics by strategic adaptation of agents to changing environmental conditions. The latter class provides insights into the functional effects as well as – to some degree – the cognitive processes of normative reasoning. The main deficit of both approaches is a lack of a dynamics of – cognitively rich – mental objects. While game theoretic models are dynamic, norms are typically regarded as merely the aggregated product of individual interactions. They thus lack the concept of mental objects. Cognitive models on the other hand, include mental objects, however, either as static or only with a limited concept of normative obligations. Thus the two way dynamics of emergence and immergence of norms is only barely captured.

6.1 Norms between Conventions and Legal Norms

The development of social simulation models has increasingly paid attention to the modelling of norms. A central problem of multi-agent simulation is the co-ordination of groups of agents. In principle two options exist, to co-ordinate the agents' action: top down regulation by a central processor or bottom-up regulation. In bottom-up approaches freedom of individual action is restricted by some kind of social constraints that become operative in the individual decision making process without a central controlling instance. This is related to the concept of norms. Thus norms refer to a central problem in the design of agent-based simulation models. This is of interest for engineering purposes, such as the regulation of electronic institutions (López y López and Márquez, 2004; Vazquez-Salceda et al., 2005). However, it is also a central problem for the sociological question of understanding the wheels of social order (Conte, 2001), and for the philosophical problem of the foundation of morality (Axelrod, 1986; Skyrms, 1996, 2004). This chapter analyses the current state of the art in this highly dynamic research field: how are norms implemented, what are findings and open questions and what are current trends and directions?

The focus here is on theoretical models with the purpose to understand the operations and effects of norms in human societies. Obviously, such models are crucially dependent on the concept of norms in the theoretical literature. However, sociological concepts of norms remain in itself ambiguous. Broadly speaking, it is possible to distinguish two approaches: individualistic theories and normativistic macro-theories. Individualistic theories describe norms as the *aggregated product* of individual interactions (e.g. Young, 2003; Opp and Hechter, 2001). They concentrate on the explanation of the spreading of norms. Such attempts are often framed in terms of rational choice theory. These approaches rely on Olson's (1965) theory of collective action and in particular on evolutionary game theory (e.g. Ziegler, 2000; Opp, 2001). Sometimes this is even regarded as a self-evident feature of norms rather than a certain point of view. For instance, Young writes that "norms can be viewed as equilibria of appropriately defined games" (Young, 2003, p. 389f.). From this perspective differences between norms and convention are of marginal interest (Burke and Young, forthcoming). On the other hand, normativist macro-theories follow a role theoretic approach in the tradition of Durkheim and Parsons. Role theory regards social roles as defined by a bundle of social norms. This account describe norms as *structural constraints* of individual behaviour. From this perspective the notion of obligations is a central element of the concept of norms. This includes a cognitive element: norms are not merely a behaviour regularity (i.e. a convention), but actors have to know what kind of behaviour is prescribed by a certain norm. The paradigm of an obligation-based concept of norms are legal prescriptions (Comp. also the introduction to this report). Different concepts of norms are placed in the scale between conventions and the legal code. These can be summarised in the two different conceptions of norms as aggregated product of individual behaviour or as a structural constraints on individual behaviour.

These different approaches are reflected in attempts to simulate norms with the means of agent based simulation models. Even though existing models are clustered around various problems and intuitions, commonly, they can be traced back to (or are at least influenced by) two traditions in particular: game theory on the one hand and an architecture of cognitive agents with some roots in Artificial Intelligence on the other hand. Of course, this is a tendency and does not constitute a clear-cut disjunction. To some degree, the distinction between game theory and (distributed) Artificial Intelligence is a distinction in the mode of speech. As recent developments in normative simulation models show, some problems of game theoretic models could also be formulated in a DAI language and vice versa. The categorisation of models as following the DAI tradition shall only indicate that the agents employed by these models are in some way cognitively richer than those in the so-called game theoretic models. Nevertheless, this distinction gives a rough sketch of the line of thought followed by the models, the kind of problems, and the concepts for their investigation. A fundamental characteristic of the game theoretic approach is the fact that the agents' behaviour can only be *interpreted* as normative by an outside observer. While the method of resolution in game theoretic simulation models is not that of analytical game theory (Binmore, 1998), these models are intimately tied to the research program of game theory and – more generally – rational choice theory. In fact, this is a highly active research field. For instance, the indirect evolutionary approach (Güth and Kliemt, 1998) investigates the evolution of trust (a norm of reciprocity) through the means of (analytical) game theory. Esser (2000) described the absolute authority of norms in the framework of rational choice theory with a theory of frame selection. Hechter and Opp (2001) explain the emergence of property rights (a social institution) in North American natives with the means of individual utility maximisation under varying circumstances. Game theoretic simulation models are integrated in a coherent research programme. They are part of the individualistic programme that regards norms as an aggregated product of individual interactions. This implies that the investigations in such simulation models concentrate on the problem norm spreading, i.e. they examine the "process of aggregation".

In the case of cognitive agents the scientific integration is not that clear. At first sight, it seem obvious that the concept of norms as obligations calls for cognitively rich agents. Thus cognitive models seem to be linked to normativist theories. In fact, as it is the case in these theories, norms are typically implemented as structural constraints in cognitive models. This approach is coherent with the role theoretic account. Moreover, it will be shown below that also the research questions of models of cognitive agents overlap to some degree with the role theoretic approach to society. So far, however, the design of existing models is more heterogeneous as it is the case in game theoretic models. Also the impact of the simulation models on this research field is far less than the corresponding one of game theoretic simulation models on rational choice theory. To a certain degree this might be due to the fact that, typically, research methods in this tradition diverge from formal models whereas rational choice theory is familiar with formal modelling.

In the following, models of these two traditions will be analysed. First game theoretical models are scrutinised and then cognitive agents are examined. Finally, results and open questions of both lines of thought are summarised.

6.2 The Game Theoretical Way of Simulating Norms

In this section the game theoretic way of modelling norms will be examined in more detail. Game theory is a mathematical theory of the rational choice of courses of action in situations of strategic interaction, i.e. in situations in which the individual success depends on the choices of others (Gintis, 2000). Insofar as it is assumed that the choice of actions is determined by the goal of individual utility maximisation, game theory follows the rational choice account. It searches for rational behaviour in situations of strategic interaction. The structure of the game is known to all actors. A number of games have been described, analysing the problems of co-ordination and co-operation. While Co-ordination games describe situations in which adjustment to others is in the interest of all agents (this are win-win situations e.g. like driving on a certain side of the street), the structure of co-operation games is framed as a dilemma: the individual best choice would lead to sub-optimal results in the case of strategic interaction. The prisoner's dilemma is the most famous example. Already in 1954 the philosopher Richard Braithwaite suggested to use the theory of games as a tool for moral philosophy (Braithwaite, 1954). This opened up a research programme of an axiomatic moral philosophy. Starting with the assumption of rational, self-interested individuals, games can be used as a means to analyse why and how normative behaviour regulations can increase individual expected utility. A well-known example is Edna Ullman-Margalit's (1977) analysis of two artillerymen who can choose to either flee from an enemy of to fire back. If both stay, they can be wounded but will hold their strategic point. If both flee, they will be overthrown. If one flees and the other stays, the brave artillerist will die but the other has time enough to escape. Obviously, this is a structure of a prisoner's dilemma. Thus individual rationality is not Pareto efficient. At this point normative binding forces (i.e. that the artillerymen fulfil their solder's duties instead of pursuing their short time self interest) can increase individual expected utility. The function of norms is thus to prevent sub-optimal results.

On shot games can be analysed with the means of analytical mathematics. The goal of analytical mathematics is to identify equilibria in which no player can change the strategy without losses in the payoff. For instance, a Nash equilibrium is the set of strategies which represents mutual best responses to the other strategies. The method of simulation is used for the exploration of evolutionary game theory, i.e. of repeated games. Evolutionary game theory relaxes the assumption of rationality. Individual rationality is replaced by differential replication, dependent on the relative success of different strategies. In the long run, this produces at least locally optimal results. While a number of analytical results exist also for evolutionary games, often these games exceed the limits of analytical mathematics. At this point, simulation can then be used to explore the behaviour space.

Since game theory describes situations of strategic interaction it is an obvious temptation to introduce norms as a means of behaviour regulation in this framework. It has to be noted, however, that the calculation of *individual* decisions is always based on the principle to maximise the expected payoff. Norms are not an element of the individual decision. Norms are introduced in game theory as an equilibrium, or at least semi-stable state in which agents co-operate. This reflects the conception of norms as the aggregated product of individual decisions. Since agents are faced with a strategic (typically binary) decision situation in the game theoretic framework, the character of norms is prescribed by the game theoretical description of the situation. Typically agents can decide to co-operate or defect in the strategic interaction. This is associated with the intuition that co-operation is morally "good", while defection is morally "bad". No clear cut distinction between simulation and analytical methods of resolution exist. A number of results of simulation models have later been proven with analytical mathematics (comp. e.g. Galan and Izquierdo, 2005) and a number of purely analytical evolutionary models exist (e.g. Young, 2003; Ziegler, 2000; Güth and Kliemt, 1998). However, because simulation models provide an overview of the behaviour space even when no analytical results are known, it is a particular strength of simulation methods to investigate the emergence of normative behaviour in evolutionary models. Therefore simulation allows to investigate more complex systems.

6.3 A Classical Model

The classical paper in the game theoretic tradition is Robert Axelrod's "an evolutionary approach to norms" of 1986. It has been analysed and replicated several times (e.g. Deguchi, 2001; Galan and Izquierdo, 2005), and remains the point of reference for this line of tradition. Since Axelrod's model has been analysed with analytical methods of resolution (Galan and Izquierdo, 2005), it is a good example of how simulation methods and analytical mathematics are interwoven in evolutionary game theory. In order to give an impression of this research field it will be briefly described in the following.

In this classical paper simulation models of a norms game and a meta-norms game are described. It does not rely on the assumption of individual rationality, but on the assumption that effective strategies are more likely to be retained than ineffective ones. This is interpreted as a form of social learning.

The norms game works in the following manner: Individual players have the options to defect (e.g. by cheating in an exam) or not defect. This is accompanied by a certain chance of being observed by other players. The defector receives a certain payoff, while all other players are slightly hurt. Yet, if player j observes the defection of player i, player j can decide to punish (or not to punish) player i. In the case of punishment, player i gets a negative payoff. However, player j has to pay an enforcement cost. The choice of the strategies is dependent on two variables: the boldness B(i), which determines the probability of defection, and the vengefulness V(i), which determines the probability of punishment. Initially the strategies of the players are set at random. However, the reproduction rate of strategies is dependent on their relative success. Strategies which gain a higher payoff are more likely to be reproduced. The simulation yield ambiguous results: different simulation runs resulted in either a high degree of vengefulness and a low degree of boldness or vice versa. Also results of a moderate level of both variables could have been observed. It has been proven analytically (Galan and Izquierdo, 2005) that these different outcomes are due to one common mechanism: at first, the boldness level starts to decrease because of the costs of being punished. Thus, the rate of defection decreases. However, this leads to a decrease in the level of vengefulness, because punishment is also costly. This in turn makes it attractive to defect again. Following the game theoretic conception of norms, the final stable state is a state without any norms at all.

For this reason Axelrod introduced a meta-norms game, in which not only defectors might get punished but also those agents that do not punish defectors. The result of the simulation is unambiguous: All runs resulted in the final stable state of a high degree of vengefulness and a degree of boldness near to zero. Axelrod concludes that a norm against defection has been established. Note, that this reflects the intuition of norms as the aggregated product of individual interaction. The mechanism behind this result is that the players have a strong incentive to be vengeful, simply to avoid punishment.

6.4 Further Development

Axelrod's model was pioneering insofar as it introduced the study of norms in the framework of social simulation and inspired further research. Nevertheless, the past decades have witnessed a number of attempts to widen the perspective by incorporating other games and further research questions. Research questions are related to the structure of the game: different questions call for different games. Already in the 1980s James Coleman (1987) investigated the effect of interaction structures on the evolution of co-operation in a prisoner's dilemma situation. Coleman examined how interactions are shaped by the fact whether agents know each other and varied the group size. The finding was that only small groups can prevent the exploitation of strangers.

In particular since the millennium change the number of models in this framework is growing rapidly. For instance, *trust games* are investigated by Macy and Sato (2002) and Bicchieri et al. (2003). Obviously, the research question is the emergence of trust. Bicchieri et al. (2003) investigate how trust emerges even if social exchange involves a time lag between promise and delivery. Several strategies of conditional co-operation survive in an evolutionary setting. This is interpreted as the evolution of a trust norm among strangers. Macy and Sato (2002) introduce a similar framework for an application in economic theory. They introduce a trust game to investigate the formation of exchange in an anonymous market. Parochial and open societies are compared as stylised representation of the Japanese and US society. It is concluded that increasing mobility supports the evolution of trust among strangers unless mobility is not too high. *Ultimatum games* are examined in models of Vieth (2003) and Savarimuthu et al. (2007a). These models study the emergence of fairness norms. The question is the evolution of fair division of a commodity. In Savarimuthu's model the interaction of two different societies, with different norms of sharing, is investigated. Two different learning mechanisms are introduced: a role model agent and a normative advisor. In the long run the norms of how to share the commodity converge in the process of interaction between the two societies. This framework has been extended with the introduction of dynamically changing networks (Savarimuthu et al., 2007b). Moreover, Sen and Airiau (2007) analyse the effectiveness of different learning strategies within the context of different games.

These examples demonstrate the spectrum of varieties in which norms have been analysed with the means of a game theoretical terminology. To consult different games enables the examination of various aspects of what is commonly associated with norms and morality. For instance, in ultimatum games players divide a sum of money that is given to them. The first player makes a proposal how to divide the sum between the two players, and the second player can either accept or reject this proposal. However, if the second player rejects, neither player receives anything. This is associated with the notion of fairness. While rationality suggests that the first player exploits the second, a fair share would be to divide the sum equally. In trust games one player (the trust maker) has to decide whether to delegate a certain task to another agent (the trustee). First, the trust maker has to decide whether to trust or not to trust the trustee. Then the trustee has to decide whether fulfil the task or to exploit the trust maker. Again rationality would suggest that the trustee would exploit the trust maker. Because the structure of the game is known to the players, the trust maker would not trust the trustee. However, trust is a central feature in social exchange. Trust and fairness are forms of normative behaviour regulation.

6.5 Analysis

All models investigate the process of norm spreading. Thus all models share a dynamical perspective on norms. Indeed, the emergence of a certain behaviour regularity is the characteristic of the game theoretic definition of norms. A norm is an equilibrium with a behaviour that can be associated with an intuition of morally good behaviour. This is the main contribution of this approach: a clear understanding of the emergence of behaviour regularities. The process of norm spreading is implemented as a dynamical updating of the propensity to co-operate or defect. The mechanism of norm spreading is determined by the premise that agents maximise their expected payoff. Therefore agents react to losses in the payoff. This is the mechanism by which agents adopt norms. Predominantly, losses in the payoff are due to sanctions, as in the models of Axelrod, Coleman or Bicchieri et al. However, non co-operative behaviour can also lead to situations in which the agents have less profitable options. For instance, in Macy and Sato's model non co-operative agents are more likely to be distrusted. In Ultimatum games, as in the models of Vieth and Savarimuthu et al., unfair proposals to share a commodity get rejected, which leads to a situation in which no agent gets anything.

It has to be noted, however, that the transmission of norms in the agent population is implemented in many models by a replicator dynamics. Typically this is *interpreted* as social learning by imitation. The effect might be true. Measured by the relative overall success of their type of behaviour, more successful types of behaviour might become more frequent. However, in a context where no real natural selection is at work, it does not indicate a mechanism how individual agents learn. Exceptions are the models of Sen and Airiau and in particular the models of Savarimuthu et al. While Sen and Airiau investigate learning algorithms known in computer science, Savarimuthu et al. introduce the notion of a role model. This is a substantial extension of the assumption of rationality. While evolutionary games still produce some kind of optima (i.e. stick to the notion of utility maximisation in an evolutionary perspective), the concept of a role model refers to the idea that behaviour is guided by a social prescription. This enables a perspective on the notion of norms as structural constraints from a game theoretic point of view.

Savarimuthu's model is exceptional because, commonly, the central weakness of this approach is the lack of any representation of the obligatory force of norms. This is closely related to the game theoretic problem description. Faced with a situation of strategic interaction, agents choose the alternative that maximises their expected utility. However, behaviour change goes not along with goal change. The question where the ends of action come from is out of the scope of this approach. Agents do no more than react to different conditions of the social environmental. The agents' behaviour is guided strategic adaptation. An active element of a normative orientation in the choice relating to the ends of action cannot be found. This is due to the fact that norms are not represented as cognitive objects and therefore agents cannot reason about norms. Agents do not act because they want to obey (or deviate from) a norm. They do not even "know" norms. Even though the modelling of behaviour transformation is the strength of this kind of models, the ends of the action remain unchanged: the goal is to maximise utility. It is only the diffusion of a behaviour regularity that is then regarded as a norm. The agents' behaviour can only be *interpreted* as normative from the perspective of an external observer. In the same way, it cannot be analysed in this context how complex strategies such as described by Axelrod's meta-norms game come up in the first place. They simply have to be introduced by the modeller. This objection leads to the question of intrinsic motivation of the agents. For instance, what are the proximate causes in the evolution of punishment? Game theory can prove the fitness of a strategy, however, it cannot account for the innovation of strategies. This is due to the lack of cognitive complexity.

6.6 The Cognitive Way of Modelling Norms

To model an explicit recognition of norms calls for cognitively more complex agents. These can be found in the tradition of a software architecture of cognitive agents with a background in Artificial Intelligence. Unlike game theoretic simulation models, which are an extension of analytical game theory, these models are not positioned in a comparable theoretical framework. They are a genuine product of research on artificial societies. The classical model in the tradition of models employing cognitive rich agents is a model described in Conte and Castelfranchi's (1995) paper on "Understanding the functions of norms in social groups through simulation". It has been replicated and extended several times (Castelfranchi et al., 1998; Saam and Harrer, 1999; Hales, 2002). Still it remains the point of reference for authors in this tradition. While it has to be noted that in this very first model no actual emergence of norms take place (norms merely remain implemented rules for action), the model framed the style of cognitive modelling. In order to give an impression of this research field it will be briefly described in the following.

6.7 A Classical Paper

The paper addresses norms that include explicit prescriptions, directives, or commands and investigates the *functions* performed by norms on the level of the *whole society*. In particular, normative control and reduction of aggression is investigated in the paper. Hence norms are regarded as structural constraints. It is investigated how a normative structure determines individual actions. This reflects Parsons' (1937) problem description of the structure of social action as well as his answer: namely, that this is determined by social norms. Moreover, to investigate the functions of norms on the level of the whole society is a functional analysis of social systems, which is associated with the role theoretic account to social theory (by investigating the functions of social roles for the society as a whole). The simulation takes place in a grid world, containing agents and food resources. Agents are equipped with an initial strength value and every action is reducing the agents' strength. Therefore agents are in need of consuming food. The agents can move in the world to search for food. If an agent occupies a cell which contains a food resource, it eats the food. In particular, the agents are *aggressive:* if a neighbouring agent is eating food, they are able to attack them. The result of the attack is dependent on the relative strength of the agents. An attack reduces the strength of both agents. However, the winner gains the food. Three kinds of experiments are undertaken:

> 1) *blind aggression:* Agents always attack eaters when no other food is available. In particular, they do not take the agent's strength into consideration. They will attack, even if they are weaker and bound to lose the battle.
>
> 2) *Strategic aggression:* this is a first step in aggression control. Strategic agents only attack those agents whose strength is not higher than their own.
>
> 3) *Normative agents:* in this setting, norm-based action control is introduced. Normative agents obey a "finder-keeper" norm: the agent that initially detects a resource is regarded as its possessor, even when agents move away from their possession. Normative agents do not attack agents eating their own food.

These experimental settings are investigated with regard to the questions of the frequency of aggression, the aggregated welfare and the equality of these different societies. The result is that the normative society exhibits the best performance: it shows the lowest number of attacks, is the richest society, and has the most equal distribution of welfare.

6.8 Further Development

This framework has inspired several extensions. Castelfranchi, Conte, and Paolucci (1998) extended the first model by introducing interaction between the different kind of societies, i.e. between the aggressive, strategic and normative agents. This leads to a breakdown of the beneficent effects of norms. However, it can be preserved with the introduction of normative reputation and communication among agents. Saam and Harrer (1999) investigate the influence of social inequality and power relations on the effectiveness of a "finder-keeper" norm. Hales (2002) introduces stereotyping in the extended Conte/Castelfranchi model (with includes reputation). Reputation is projected not on individual agents but on whole groups. This works effectively only when stereotyping is based on correct information. Even slight noise causes a breakdown of norms. Staller and Petta (2001) examine Conte and Castelfranchi's findings from a perspective that regards emotions as important for the sustenance of social norms. They could replicate the original findings within this framework. This holds even for the case that agents are able to deliberately decide whether to obey or violate the "finder-keeper" norm dependent on their hunger. Possession norms are also studied in a model of Flentge et al. (2001). However, the authors study the emergence of such a norm by processes of memetic contagion. The norm is beneficent for the society, but has short-term disadvantages for individual agents. For this reason, the norm can only be retained in the presence of a sanctioning norm. Other models concentrate not on specific norms but on the more abstract mechanisms related to the operations of norms, such as the process on the individual learning of norms and their spreading in a population of agents (e.g. Epstein, 2000, 2006; Burke et al., 2006; Verhagen, 2001). Epstein (2000, 2006) and Burke et al. (2006) generated patterns of local conformity and global diversity with agents of only moderate cognitive complexity. In this respect, they recover and refine the findings of game theoretic models: spatial patterns of local conformity and global diversity are typically not a feature of game theoretic models (although it is possible to include spatial patterns). Additionally, Epstein concluded that norms release agents from individual thinking. Norms allow for "thoughtless conformity". Verhagen (2001) examines the tension between predictability of social systems while preserving autonomy on the agent level through the introduction of norms. In the model, the degree of norm spreading and internalisation is studied with a sophisticated interplay between a self-model and a group-model. While the self-model represents the autonomy, the group model represents the assumptions of the agent about norms hold in the group.

6.9 Analysis

Typically, a much stronger notion of norms is deployed in these model than in game theoretic models. The actions performed by the agents cannot be reduced to a binary decision to co-operate or defect. Norms are more than a behaviour regularity. They are an explicitly prescribed action routine. This agenda has been set with the very first models. It matches with the perspective to regard norms as a structural constraint of individual actions. This is consistent with the role theoretic conception of norms in sociological theory. This holds also for the investigation of concrete norms such as possession norms and research question such as social inequality or power relations. These questions address issues which concern a society as a whole. What are consequences of structural constrains for the society? It is striking that already the Conte/Castelfranchi model investigated the functional effects of norms on a population level, such as the aggregated welfare or the equality of the distribution of welfare. This conception of norms implies that more complex behaviour rules have to be applied than to update the propensity to co-operate or defect. Often these are based in conditional logic. However, it has to emphasised that in the very first model of Conte/Castelfranchi the agents are bound to execute the prescribed rules. Agents do not deliberate and therefore agents have no individual freedom in this model, but remain merely normative automata. Norms remain static. This limitation has been overcome by the subsequent developments. As a number of models show, norm spreading can now be modelled within this account. However, key differences to game theoretic models still remain: while in game theoretic models sanctioning has a prominent role in behaviour transformation, in models of cognitive agents sanctions are only employed by Flentge et al.. By comparison, these models refer more explicitly to communication than game theoretic models. The decision and learning processes utilised are more realistic mechanisms than the replicator dynamics of game theoretic models.

However, even though these agents are cognitively richer than game theoretic agents, the recursive feedback loop between inter agent processes and intra agent processes still remains underspecified. This can be illustrated by the question of how a dynamics of cognitive objects is represented. This includes their spreading in an agent population as well as a dynamics of these objects themselves. Most of the models concentrate on norm spreading and employ a number of different mechanisms. The models of Epstein and Burke et al. are restricted to observation. Memetic contagion is a further candidate. However, it might be doubted whether the mechanisms applied are a theoretically valid representation of real processes. The models of Staller and Petta and Verhagen provide examples of more sophisticated accounts. Staller and Petta's model is exceptional with regard to the inclusion of emotions. Insofar as the decision whether to comply to a norm is regulated by emotions, it employs a highly sophisticated decision process. This concerns the micro level of intra agent processes. However, the process of how norms find their way into the individual agents' mind (i.e. the feedback loop from inter agent processes to intra agent processes) remains a black box also in this model. Agents deliberate whether to comply to a norm. However, the norms itself are known and have to be given (i.e. programmed-in). In Verhagen's model a quite sophisticated account of norm internalisation is undertaken, including a self-model, a group model and a degree of autonomy. Thereby it constructs a feedback loop between individual and collective dynamics. By the combination of a self- and a group-model a representation of the (presumed) beliefs held in the society is integrated in the belief system of individual agents. The interaction of self- and group-models of interacting agents enables the possibility that norms itself might change.

Thus a number of different accounts can be discerned. However, even though the static concept of norms has been overcome, a comprehension of norm innovation is only in the beginning.

6.10 Results and Open Questions

An overview of existing models of normative agents reveals that these models originate primarily in two different traditions: game theory and cognitive agents models with some roots in Artificial Intelligence. These different approaches reflect different intuitions about norms: while game theoretic models regard norms as the aggregated product of individual interactions, cognitive agent models treat norms as structural constraints of individual actions. The former concept of norms has some affinity to conventions. The cut-off point of the latter are legal norms. Game theoretic models investigate norm dynamics. They provide a sound mechanism that explains the dynamics, namely losses in the expected payoff, mainly through punishment. The latter approach is more heterogeneous. Recent models include the ability that agents are able to deliberate about norms. Moreover, a particular feature of cognitive agents is to demonstrate the effects of norms on a population level.

However, recent developments show a convergence of both traditions. Norm dynamics is no more out of the focus of cognitive agents and also game theoretic models (developed by Savarimuthu et al.) exist that investigate the effects of norms on a population level. Nevertheless, there is still some need for further research with regard to a comprehension of the feedback loop between inter agent processes and intra agent processes. Steps in this direction can be found in a number of models, such as the model of Savarimuthu et al., the model of Staller and Petta or Verhagen's model. These models have introduced sophisticated mechanisms to balance between individual autonomy and the obligatory force of norms. However, a dynamics of cognitive objects is only barely captured. How can the formation of norms be comprehended and how are they transmitted into the individual mind? This becomes particular apparent when considering norm innovation: This would require both cognitive processes in the invention and recognition of norms as well as the inter agent process of the spreading of the new norm. So far, dynamics is restricted to the spreading or decline of given norms. The invention of new norms is out of the scope of normative simulation models. Since in game theoretic models the suggested norms are already given by the description of the particular game, norm innovation is impossible in principle. Agents have only the choice between the alternatives defined by the situation. Even ultimatum games remain restricted to the problem situation described by the game, even though an infinity of numerical proposals is possible. However, it is also a challenge for the design of cognitive agents to model creative processes. While cognitive agents can deliberate about norms, the deliberation is restricted to means-ends calculations.

Chapter 7 Relationship to Other Thought: The BDI Approach to Norm Immergence

Martin Neumann, Marco Campenni, and Giulia Andrighetto

Abstract

In order to model and operationalize the process of norm immergence, agents have to be endowed with internal mechanisms and mental representations allowing norms to affect the behavior of autonomous intelligent agents. Such representations are commonly realized by architectures inspired by the modular design of Artificial Intelligence approaches.

So far, no unequivocal concept for the design of normative agents exist. In fact, the development of normative architectures is a burgeoning research field. However, architectures of normative agents are predominantly informed in some way by BDI (Belief-Desire-Intention) architectures, which can be regarded as the point of departure for further developments. The BDI framework is intended to model human decision-making. A particular striking example of this approach is a straightforward extension of the BDI architecture, denoted as BOID (Belief-Obligations-Intentions-Desires) agent architecture (see Broersen et al., 2001), which includes obligations among its mental components.

This chapter is aimed to provide a comparative analysis of selected cases of normative agent architectures in order to identify common trends and needs for further research.

7.1 Introduction

In this chapter it will be examined, how immergent processes can be discerned in current attempts to develop architectures of normative agents. In fact, the development of normative architectures is an expanding research field. While agent based modelling has provided substantial contributions to the question of how the process of norm spreading can be accounted for (compare Chapter 6 on the “state of the art” in this report), the reverse process of how norms are recognised by individual agents is far less understood so far. This concerns the question of norm immergence. To examine this question this chapter will have a look on agent architectures. In fact, the number of conceptually oriented articles on the architecture of normative agents exceeds the number of existing models. Typically, norms in concrete models are less sophisticated than concepts proposed in formal architectures (Conte and Dignum, 2001). However, focusing on implementation before having achieved a proper understanding of formal architectures risks losing key norm-related intuitions (Dignum et al., 2002). The development of architectures is thus a kind of requirement analysis: here it is examined what are the essential components that have to be considered in the process of implementation to represent the problem in question. So the problems and suggested solutions evaluated in this chapter concern the future of agent based modelling of norms as well as possible solutions that are presented in this report.

However, first and foremost it has to emphasised that the current state of the art does not allow to identify an unequivocal direction of what the future will bring about. So far a number of diverging approaches exist. To a large degree this is due to the fact that norms can contribute to a large number of highly divergent research fields. Norms have turned out to be a useful tool *and* a crucial theoretical concept in agent-based modelling: the motivations range from technical problems in the application of multi-agent systems (Garcia-Carmino et al., 2006; Vasquez-Salceda et al., 2006) to theoretical interests in the foundations of Distributed Artificial Intelligence (Dignum et al., 2002; Broersen et al., 2005), the foundations of morality (Axelrod, 1986; Skyrms, 1996) to the “wheels of social order” (Conte and Dellarocas, 2001).

This chapter will re-examine the current approaches to normative architectures. However, this examination is guided less by technical and engineering problems, such as, for instance electronic auctions (Garcia-Camino et al., 2006) or web searching robots (Dignum et al., 2002), than by the theoretical question of how a process of norm immergence can be accounted for (Conte et al., 2007). Therefore in the following, the theoretical perspective will be enfolded first. A common frame to both theoretical and engineering problems is the notion of complex systems. It will be specified how the concept of norms is placed within this framework. Subsequently, those components in the current state of the art that are of particular relevance for a comprehension of these questions will be subject to a further examination. Finally, opportunities and drawbacks of current attempts are identified.

7.2 The Theoretical Perspective on Norm Immergence

Complex systems are characterised by a high number of interacting components that are nevertheless not completely determined by their relational description. The research on complex systems has shown that such systems are characterised by a number of features that cannot be found in classical systems. This includes non-linear causation, emergence, downward causation, to mention just a few phenomena that are characteristics of complex systems. To characterise the question investigated in this chapter, however, it is useful to introduce a further crucial aspect of complex social systems, namely the difference between implementation and incorporation. This is linked to the notion of emergence and downward causation: while emergence characterises processes of how new, macro level phenomena arise from interacting micro level entities, the notion of downward causation was introduced to characterise the reverse process of how this emergent macro level enfolds causal power on the generating micro level (Conte et al., 2007). This concept implies but is not equivalent to implementation since a macro-social entity is always implemented on a micro-social one. Any macro social entity may act and take effect only through the actions of micro-social entities. In social systems these are individuals (Conte et al., 2007).

An example of such a recursive process can derived from network analysis. Dependence networks (Sichman, 1994; Sichman and Conte, 2002), for instance, constrain the scope of actions that can be taken by an individual agent. Dependence networks emerge due to the fact that in a common environment, actions done by one agent influence the goals of other agents (Conte et al., 2007). If, for example, in the set of agents <a, b, c>, a is endowed with goal p and action a(q), while b and c are both endowed with goal q and action a(p), their interconnections result in a dependence network, where agents b and c are socially dependent on a. A, however, depends on either b *or* c. In turn, this non-uniform distribution of exchange power determines a new effect at the lower level: a gets a higher negotiation power than either b or c: a will be in the position to make a choice, i.e. to choose its partner of exchange, while b and c have no choice (Conte, 1998). This effect is a property of the whole network, not of one single agent. It is a macro property of the network that determines the actions of individual agents. However, the structure of the effect need not be recognised by the agents to enfold its power. This is an example of an *immergent* process.

Sometimes, however, a macro-social entity may be *incorporated* into a lower level one. This means that the macro social entity not only determines individual action but is explicitly represented within the individual agent on the micro level (Conte et al., 2007). This becomes particular apparent in the case of social norms. While norms are generated from the micro to the macro, the macro level property of social norms becomes incorporated in the mind of social actors. For this reason it is useful to differentiate between the two concepts of immergence and incorporation. While immergent effects are those, where macro-social properties cause new micro properties that reproduce or support the effect, the notion of incorporation includes the concept that the emergent effect gets represented in the producing system. They become a mental object. This representation in turn contributes to the replication of this effect. This is a characteristic feature of social norms. This feature implies the fact that the actor can deliberate about norms. Actors can pose questions such as “what is the scope of a particular norm” and reason whether he or she should comply or deviate from the norm. In particular, the concept of incorporation enfolds a framework that enables to characterise norm innovation, a feature that is obviously apparent in human society. To mention just one example, the spread of motorised traffic has given rise to the norm to drive on one particular side of the street. This implies that people do not only unconsciously drive on a particular side of the road[34]. In fact, humans learn such norms (at latest) in the driving school and their knowledge of

[34] Such co-ordination processes are already implemented in a number of simulation models (comp. e.g. Sen and Airiau, 2007). These models are characterised by the fact that a certain co-ordination of the agents action emerge in course of the simulation run because this is in the benefit of all agents. If learning agents compute their utilities, gradually their behaviour appears to become co-ordinated since this maximise expected utilities. However, the agents do not need to ‘know’ the norm. The norm does not

these mental objects is subject to a driving test. The norm thus becomes incorporated in the actors mind. This leads to the conclusion that norm-innovation is characterised by the occurrence of the two complementary processes, emergence and immergence: norms cannot emerge unless they simultaneously become incorporated in the agents' minds (Conte et al., 2007). Thereby norms are a link between the individual and society. However, the relation can be rather complicated (Campenni et al., 2009).

7.3 Norms in Current Architectures

To represent such processes in the architecture of software agents, however, is a demanding task. The development of normative architectures is a burgeoning research field (see also Chapter 10). This suggests to investigate current architectures to check to what extent they can contribute to a comprehension of this problem. While no exhaustive overview of existing cases can be provided, the examination will draw upon a sample of 13 cases. Subsequently, it will be examined how these approaches contribute to a comprehension of how social norms become incorporated in the agent's mind. Such a focus implies that not *all* aspects that are relevant for the development of normative architectures can be investigated (Neumann, 2008). This chapter concentrates on those aspects that are of particular relevance for the question of how the formation of a new mental object can be implemented in an agent model. Here two aspects are of particular relevance:

i. First, it is apparent that the formation of mental objects is a dynamic *process*.
ii. Second, this process is a social process. The question of how macro level properties become mental objects on the individual agents' level implies some kind of relation between macro and micro level. In the case of *social* simulation the macro level is a social level.

However, before it is investigated how these aspects are represented, first, some very basic questions will be examined: where and how norms are implemented in the architectures.

7.4 Normative Modules

Architectures of normative agents are predominantly informed in some way by BDI (Belief-Desire-Intention) architectures (Rao and Georgeff, 1991). The BDI framework is intended to model human decision-making. One of its key insights is directed at amplifying logical models with cognitive components. While BDI architectures still apply logical operations, the notions of beliefs, desires and intentions refer to intuitive cognitive meaning in everyday language. The idea to identify and provide distinct modules for different components of the agent architecture is the starting point for normative agent models (Broersen et al., 2005). To implement social behaviour in agents, next to beliefs, desires and intentions a further component is to be added: obligations. In this component, social norms are implemented to include social rationality in the agent's design. To implement social norms as a separate component enables more flexibility to the agent's actions. While the agent may have its individual desires, social norms can be implemented as an obligations component that can be described as "desires of the society" (Dignum et al., 2002). By explicitly separating individual and social desires, it is possible that the agent can decide based on the components' priorities Conflicts may arise among different components. For instance, I may want to smoke a cigarette after dinner but am obliged to refrain from smoking in restaurants. If everything was stored in a single component, conflicts could not be modelled since logically everything could be deduced from a contradiction. However, if the desire to smoke and the obligation (i.e. "society's desire") not to smoke is stored in different components, the agent can decide which desires (i.e. social or individual) to fulfil. Thus, the agent is able to violate obligations, thereby showing autonomy. This intuition can be further elaborated by adding further components and complex rules about how these components are related to one another. However, the bulk of the BDI approach is to modularise the agents' mental states into interacting components.

become incorporated in the agent's mind. Such models contribute to a comprehension of an important aspect of self organising systems. However, they do not capture an important aspect of human societies, namely the recognition of emergent macro-properties.

7.5 Concepts of Norms

In some way, norms have to be specified computationally. The existing approaches can be regarded as a hierarchy of increasingly sophisticated accounts. The simplest and most straightforward way is to regard norms as mere constraints on the behaviour of individual agents. For example, the norm to drive on the right-hand side of the road restricts individual freedom. In this case, norms need not necessarily be recognised as such. They can be implemented off-line or can emerge in interaction processes. It follows that what is normative is derived from what is normal, a derivation that Hume called "naturalistic fallacy". More sophisticated accounts treat norms as mental objects (Castelfranchi et al., 2000; Conte and Castelfranchi, 1995). This allows for deliberation about norms and, in particular, for the conscious violation of norms. Norms intervene in the process of goal generation, which might – or might not – lead to the revision of existing personal goals and the formation of normative goals. However, two further approaches need to be distinguished in this case: a number of accounts (such as the BOID architecture) rely on the notion of obligations. Obligations are explicit prescriptions that are always conditional to specific circumstances. One example of an obligation is don't smoke in restaurants. Agents may face several obligations that may contradict with one another. For this reason, some authors differentiate between norms and obligations. Norms are regarded as more stable and abstract concepts than mere obligations (Conte and Dignum, 2001; Dignum et al., 2002). One example of such an abstract norm is "being altruistic": further inference processes are needed for the formation of concrete goals from this abstract norm. Such abstract norms are timelessly given and not context-specific. All alternatives can be found in the selected cases. However, the concept of obligations is predominant. Figure 10 summarises the relative frequencies of these alternatives in the sample.

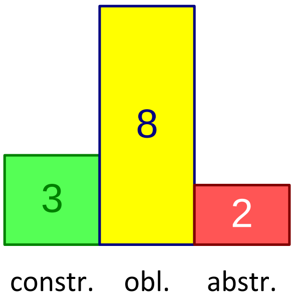


Figure 10. Norm concepts

7.6 Norm Dynamics

Obviously, norms may change in the course of time. The example of driving on a particular side of the street shows that in human societies norms are not static. It is thus a crucial question whether this aspect is reflected in normative architectures. However, in most architectures norms remain static. In these architectures norms have to be programmed-in by the designer of the model, so they cannot change in the course of a simulation. Only two of the 13 selected cases consider norm dynamics. In the case of norm dynamics, however, two different cases have to be distinguished: the innovation of new norms and a changing scope of norm validity. The most prominent example of the latter is the case of norm spreading. Indeed, the spreading of norms is an essential component in game theoretic models of norms (compare Chapter 6 on the "state of the art" in this report). Norm spreading is also considered in the two architectures that include norm dynamics. Norm innovation, however, cannot be found in the selected cases. The relative frequencies are displayed in Figure 11.

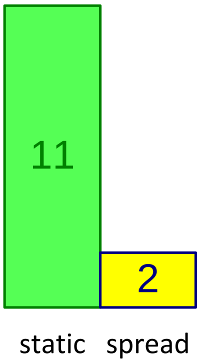


Figure 11. Norm dynamics

7.7 Single or Multiple Agents

In the literature on existing architectures, norms are described as the “social burden” (Garcia-Camino et al., 2006) or “desires of the society” (Dignum et al., 2002). They become relevant when agents interact with other agents. However, norms are also a feature of individual agents, regulating individual behaviour. It is thus an essential question how society is represented in the agent architecture: should the cognitive architecture concentrate on a single agent or should the agents’ interaction structure taken into account? Typically, to concentrate on a single agent allows for a more analytical architecture of the intra-agent processes. A precondition for investigating norm dynamics is to take agents’ interaction into account. The number of accounts that decide for either of these alternatives is balanced. 3 architectures include a population of agents and 5 cases concentrate on a single agent. A compromise between these two alternatives is to represent society in an indirect manner. Society can also be modelled implicitly as a mental representation of a single agent. This alternative can be found in 4 cases. In the case of a mental representation of society within the architecture of a single agent, the cognitive architecture of the agent includes a component to store information about the society. The information has to be implemented externally because no other agent is actually present. No interaction is described. Hence the society is only implicitly represented in the agent’s mind. Figure 12 summarises the relative frequencies of these alternatives as they are found in the sample.

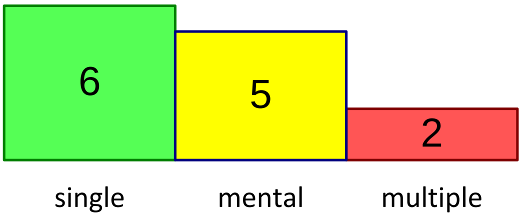


Figure 12. Single or multiple agents

Moreover, the representation of social complexity is highly different in different architectures. This is independent of the question whether society is explicitly represented as a population of agents or is only represented in the agent’s mind. One central distinction is between two central roles related to norms: addressees of norms who are obliged to fulfil the norms and beneficiaries of norms. This distinction can be found in architectures that explicitly introduce populations of agents as well as in architectures that solely introduce society as a mental representation of an individual agent. However, roles can be further specified: In the so-called “KGP” architecture (Sadri et al., 2006), roles are assigned to agents, which are initiated and terminated by an event calculus; e.g. *assign(a, t_(chelsea, 9,7))* means that agent *a plays the role of a traffic warden in Chelsea between the times 9 am and 5 pm.* This implies that agent a is addressee of different obligations between 9 am and 5 pm (i.e. its working duties) than at other times.

7.8 Norm Conflicts

Finally, it will be considered how conflicts arise and are handled. Already the intuition to implement norms in a separate component has been based by an attempt to differentiate individual desires and “desires of the society". This allows already the emergence of conflicts between different desires. This is not possible in agents in which social norms are not implemented in a separate component. For instance, Shoham and Tennenholtz (1992) example of robots with implemented rules how to pass, are unable to violate these rules, i.e. conflicts cannot arise. Hence, the regulation of intra-agent conflicts (between different goals) is an essential feature of this intuition of norms. Yet the concept of conflicts is not as unequivocal as it might appear at first sight. The question is where to identify the source of conflicts. Conflicts may arise between agents, between agents' goals and social norms, and there might even be contradicting social norms. However, it may also be the case that conflicts are not considered. This is the simplest case, when no differentiation between components is introduced. For instance, the rule "drive on the right hand side" might simply be implemented in the agent as one rule next to others. Contradictions would cause errors. Norm invocation can be regarded as a conflict between agents. The desire to smoke in a non-smoking restaurant exemplifies a conflict between the desires and obligation components of an agent. Finally, norms may contrast one another. This is particularly the case when different normative authorities are taken into account. Figure 13 shows which architectures take which intuitions into account.

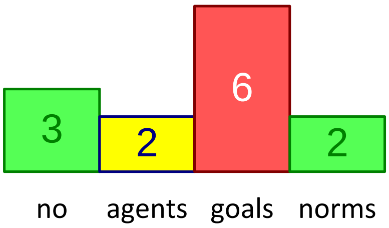


Figure 13. Conflicts

The consideration of conflicts calls for some kind of conflict resolution, even though in some cases this is not considered explicitly. Other architectures develop highly sophisticated techniques. A straightforward idea is the maximisation of expected utility (López y López and Márquez, 2004). Another approach is to introduce a priority ordering among the different components regardless of the utility that can be expected from actions following each of them respectively. For instance, an agent may always follow obligations even if they contradict its individual desires. Such a priority ordering allows to differentiate typologies of agents, for instance selfish agents, always following own desires, or socially responsible agents that always give obligations priority over desires (Broersen et al., 2001). To deal with conflicts among different social obligations other concepts would be necessary, including ordering various levels of abstraction As it is known from default reasoning, the lower level of abstraction specifies exceptions to default (Vazquez-Salceda et al., 2005). However, the development of techniques to resolve such problems is only in the beginning.

7.9 Conclusion

What are common trends and needs for further research? How can processes of incorporation of social norms be represented?

It is striking that a great variability of different approaches exist. However, a common feature of cognitive architectures is the introduction of modular components. While the idea of norms as an abstract concept remains under-represented, the major advantage of cognitive architectures is to introduce norms as explicit obligations. This enables behavioural flexibility and allows to handle conflicts between social norms and individual goals. In this respect, norms are treated as mental objects enabling agents to recognise and deliberate about norms. This goes beyond pure co-ordination games. This has been realised due to the introduction of several modules for different cognitive components. Thus, current architectures take a step towards a comprehension of norm incorporation. However, it is a major shortcoming of current approaches, that predominantly architectures of normative agents are static. The process of how norm *become* incorporated in the mind of individual agents remains out of the scope of such approaches. Presumably, the fact that only 2 of 13 cases include norm dynamics is one of the most striking results of this survey. Moreover, norm innovation cannot be found in any of these architectures. However, architectures exist that include a social level. This is realised by an explicit introduction of populations of agents as well as indirectly by representing society as a mental component of the agent architecture. However, the latter approaches lacks a representation of social dynamics. Future trends may thus include both dynamics and an explicit introduction of populations of agents to comprehend the dynamics of incorporation and innovation.

7.10 Appendix

The case study is based on the following cases:

1) "Norm Governed Multiagent Systems" (Boella and van der Torre, 2003). This paper differentiates between three types of agents: agents who are the subject of norms, so-called defender agents, who are responsible for norm control, and a normative authority that has legislative power and that monitors defender agents.
2) "An architecture of a normative System" (Boella and van der Torre, 2006). The authors rely on John Searle's notion of institutional facts (so-called "counts-as" conditionals) to represent social reality in the agent architecture. A norms base and a "counts-as" component transforms brute facts into obligations and permissions.
3) "The BOID Architecture" (Broersen et al., 2001). The Belief-Obligation-Intentions-Desire (BOID) Architecture is the classical approach to represent norms in agent architectures. Obligations are added to the BDI Architecture to represent social norms while preserving the agent's autonomy. Principles of the resolution of conflicts between the different components are investigated in the paper.
4) "Norms in Artificial Decision Making" (Boman, 1999). This paper proposes the use of super-soft decision theory to characterise real-time decision-making in the presence of risk and uncertainty. Moreover, agents can communicate with a normative decision module to act in accordance with social demands. Norms act as global constraints on individual behaviour.
5) "Deliberative normative agents" (Castelfranchi et al., 2000). This paper explores the principles of deliberative normative reasoning. Agents are able to receive information about norms and society. The data is processed in a multi-level cognitive architecture. On this basis, norms can be adopted and used as meta-goals in the agent decision process.
6) "From conventions to prescriptions" (Conte and Castelfranchi, 1999). A conventionalist (in rational philosophy) and a prescriptive (in philosophy of law) perspective on norms is distinguished in this paper. A logical framework is introduced to preserve a weak intuition of the prescriptive perspective which is capable of integrating the conventionalist intuition. The notion of a normative authority is therefore abandoned.
7) "From Social Monitoring to Normative Influence" (Conte and Dignum, 2001). This paper argues that imitation is not sufficient to establish a cognitive representation of norms in an agent. Agents infer abstract standards from observed behaviour. This allows for normative reasoning and normative influence in accepting (or defeating) and defending norms.
8) "From desires, obligations and norms to goals" (Dignum et al., 2002). This paper investigates the relations and possible conflicts between different components in an agent's decision process. The decision-making process of so-called "B-doing agents" is designed as a two-stage process, including norms as desires of society. The authors differentiate between abstract norms and concrete obligations.
9) "Norm-oriented programming of electronic institutions" (Gracia-Camino et al., 2006). In this paper, norms are introduced as constraints to regulate the rules of interaction between agents in situations such as a Dutch auction protocol. These are regulated by an electronic institution (virtual auctioneer) with an explicit normative layer.
10) "An architecture for autonomous normative agents" (López y López and Márquez, 2004). This paper explores the process of adopting or rejecting a normative goal in the BDI framework. Agents must recognise themselves as addressees of norms and must evaluate whether a normative goal has a higher or lower priority than those hindered by punishment for violating the norm.
11) "Normative KGP Agents" (Sadri et al., 2006). The authors of this paper extend their concept of knowledge, goals and plan (KGP) agents by including norms based on the roles played by the

agents. For this reason, the knowledge base KB of agents is upgraded by KB_{soc}, which caters for normative reasoning, and KB_{rev}, which resolves conflicts between personal and social goals.

12) "On the synthesis of useful social laws for artificial agent societies" (Shoham and Tennenholtz, 1992). The authors propose building social laws into the action representation to guarantee the successful coexistence of multiple programs (i.e. agents) and programmers. Norms are constraints on individual freedom. The authors investigate the problem of automatically deriving social laws that enable the execution of each agent's action plans in the agent system.
13) "Norms in Multi-Agent Systems: From theory to practice" (Vazquez-Salceda et al., 2005). This paper provides a framework for the normative regulation of electronic institutions. Norms are instantiated and controlled by a central institution, which must consist of a means to detect norm violation and a means to sanction norm violators and repair the system.

Chapter 8 Theory Essentials: The Added Value in Normative Agents

Marco Campennì, Giulia Andrighetto, Rosaria Conte

Abstract

Traditionally, the scientific domain of normative agent systems presents two main directions of research: the first, related to Normative Multiagent Systems (see Boella et al., 2006); the second focused on much simpler agents and the emergence of regularities from agent societies (Bicchieri, 2006; Epstein, 2006; Sen and Airiau, 2007).

The present chapter aims to test the effectiveness of a third possible approach: the norm recognition and the role of normative beliefs in norm emergence and innovation using agent based simulation. We aim to find out the sufficient (if not necessary) conditions for existing norms to change, and to show whether agents provided with a module for telling what a norm is, can generate new (social) norms by forming normative beliefs, even irrespective of the most frequent actions. We also aim to find out which simple cultural or material artifacts facilitate the process of norm innovation. To see this, we modeled a simple case in which subpopulations are isolated in different scenarios for a fixed period of time.

8.1 Introduction

Traditionally, the scientific domain of normative agent systems presents two main directions of research. The first, related to Normative Multiagent Systems, focuses on *intelligent* agent architecture, and in particular on normative agents and their capacity to *decide* on the grounds of norms and the associated incentive or sanction (see Boella et al., 2006). This scientific area absorbed the results obtained in the formalization of normative concepts, from deontic-logic (von Wright, 1963; Alchourròn and Bulygin, 1971), to the theory of normative position (Lindhal, 1977), to the dynamics of normative systems (Alchourròn et al., 1985). Those studies have provided Normative Multiagent Systems with a formal analysis of norms, thus giving crucial insights to *represent* and *reason* upon norms.

The second is focused on much simpler agents and on the *emergence of regularities* from agent societies. Very often, the social scientific study of norms goes back to the philosophical tradition that defines norms as regularities emerging from reciprocal expectations (Lewis, 1969; Bicchieri, 2006; Epstein, 2006). Indeed, interesting sociological works (Oliver, 1993) point to norms as public goods, the provision of which is promoted by 2nd-order cooperation (Heckathorn, 1988; Horne, 2007). This view inspired most recent work of evolutionary game-theorists (Gintis et al., 2003), who explored the effect of *punishers* or *strong reciprocators* on the group's fitness, but did not account for the individual decision to follow a norm.

While the latter approach has been mainly interested in how social norms emerge, spread and change over time, the Normative Multiagent System approach has focused on the question why agents comply with norms and how is it possible that norms operate upon autonomous intelligent agents. No apparent contamination and integration between these different directions of investigation has been achieved so far. In particular, it is unclear how something more than regularities can emerge in a population of intelligent autonomous agents and whether agents' *mental capacities* play any relevant role in the emergence of norm.

The aim of this chapter is help clarify what aspects of cognition are essential for norm emergence and norm innovation. We will concentrate on *one* of these aspects, i.e. *norm recognition*. We will simulate agents endowed with the capacity to tell what a norm is, while observing their social environment.

One might question why start with norm recognition. After all, isn't it more important to understand *why* agents observe norms? Probably, it is. However, whereas this question has been answered to some extent (Conte and Castelfranchi, 1995, 1999) the question how agents tell norms has received poor attention so far.

In this chapter, we will address the antecedent phenomenon, *norm recognition*, postponing the consequent, norm compliance, to future studies. In particular, we will endeavour to show the impact of norm recognition on the emergence of a norm. More precisely, we will observe agents endowed with the capacity to recognize a norm (or a behavior based on a norm), to generate new normative beliefs and to transmit them to other agents by communicative acts or direct behaviors.

We intend to show whether a society of such normative agents allows social norms to emerge. The notion of norms that we refer to (Conte and Castelfranchi, 2006) is rather general. Unlike a *moral* notion, which is based on the sense of right or wrong, norms are here meant in the broadest sense, as behaviors spreading to the extent that and because (a) they are prescribed by one agent to another, (b) and the corresponding normative beliefs spread among these agents (for a more detailed characterization of social norms see section Norm Innovation).

Again, one might ask why not to address our moral sense, our sense of the right or wrong. The reason is at least twofold. First, our norms are more general than moral virtues. They include also social and legal norms. Secondly, and moreover, agents can deal with norms even when they have no moral sense: they can even obey norms they believe to be unjust. But in any case, they must know what a norm is.

8.2 Objectives

The present work consists in the implementation of an exploratory model and aims to test the effectiveness of norm recognition and the role of normative beliefs in norm emergence and innovation by means of agent based simulation.

Some preliminary simulations, discussed in (Andrighetto et al., 2008b), compared the behavior of a population of normative agents provided with a norm recognition module and a population of social conformers whose behavior is determined only by a rule of imitation. The results of these simulations show that under specific conditions, i.e. moving from one social setting to another, imitators are not able to converge on one behavior, even if this is common to different settings, whereas normative agents are. Norm recognition has appeared to represent a crucial requirement of norm emergence.

In this chapter we want to find out the sufficient (even if not necessary) conditions for existing norms to change, using the same architecture developed in (Andrighetto et al., 2008b). In particular, we want to show if a simple cultural or material constraint can facilitate norm innovation. We wonder if under such a condition, agents provided with a module for telling what a norm is can generate new (social) norms by forming new normative beliefs, irrespective of the most frequent actions. To see this, we imagined a simple case in which subpopulations are temporarily isolated in different scenarios for a fixed period of time, so each subpopulation cannot switch from one scenario to another. The metaphor here is any physical catastrophe or political upheaval that divides one population into four separate communities. The recent European history has shown several examples of this phenomenon.

8.2.1 (Social) Norms

Norms are social artifacts, emerging, evolving, and decaying. If it is relatively clear how legal norms are put into existence, it is much less obvious how the same process may concern social norms. How do new social norms come into existence?

As said in section Introduction, we consider a social norm as a behaviour that spreads trough a population thanks to the spreading of the corresponding normative prescription, and consequently of a particular shared belief, i.e. the normative-belief (von Wright, 1963; Ullman-Margalit, 1977; Kelsen, 1979; Conte and Castelfranchi, 1999, 2006). Thus, for a norm-based behavior to take place, a normative belief has to be generated into the minds of the norm addressees, and the corresponding normative goal[35] has to be formed and pursued. Drawing upon Kelsen (Kelsen, 1979), von Wright (von Wright, 1963) and a long tradition of deontic philosophy and logic-based theory of action, we define a normative belief as a belief

[35] Throughout the paper, we will speak of goals from the point of view of computer science and autonomous agent theory. In particular, a goal is a wanted world-state that triggers and guides action (Conte, 2009).

that a given behaviour, in a given scenario, for a given set of agents, is either forbidden, obligatory, or permitted (Conte and Castelfranchi, 1999, 2006).

Our claim is that a norm emerges as a norm only when it is incorporated into the minds of the agents involved (Conte and Castelfranchi, 1995, 2006); in other words, when agents recognize it as such. In this sense, norm emergence and stabilization implies its immergence (Castelfranchi, 1998) into the agents' minds. This meaning that norm emergence is a two way dynamics, consisting of two processes:

- emergence: process by which a norm not deliberately issued spreads through a society;
- immergence: the gradual and complex process by which the macro- social effect, in our case a specific norm, affects the generating systems, their beliefs and goals, in such a way that agents force one another into converging on one global macroscopic effect (Castelfranchi, 1998; Conte et al., 2007).

Emergence of social norms is due to the agents' behaviors, but the agents' behaviors are due the mental mechanisms controlling and (re)producing them (immergence).

Some simulation studies about the emergence of social norms have been carried out, for example Epstein and colleagues' study of the emergence of social norms (Epstein, 2006), and Sen and Airiau's study of the emergence of a precedence rule in the traffic (Sen and Airiau, 2007). In these studies, social norms are essentially seen as conventions, that is, behavioural conformities that do not imply explicit agreements among agents, and do emerge from their individual interests. Within this perspective, the function of norms is found in allowing participants in coordination games to choose one among equivalent alternative equilibriums. Agents repeatedly interact with other agents in social scenarios. Such interactions can be formulated as stage games with multiple equilibriums (Myerson, 1991), which make coordination uncertain. Norms gradually emerge from interactional practice, essentially through mechanisms of imitation and social learning, establishing who should do what. So far, simulation-based studies have been applied to investigate which norm is chosen from a set of alternative equilibriums. In this framework agents are not provided with normative minds, but with strategic reasoning. No attention is paid to norm immergence, and therefore to the role of mental mechanisms in norm-emergence.

A rather different sort of question concerns the emergence and innovation of social norms when no alternative equilibria are available for selection. This is a subject still not widely investigated and references are scanty if any[36].

We propose that a possible answer to the puzzling questions posed above, i.e. how do new social norms come into existence?, might be found out examining the interplay of communicated and observed behaviors, and the way they are represented into the minds of the norms' addressees. If any new behavior α is interpreted as obeying a norm (for a detailed description of the norm recognition process see section 5.1), a new *normative belief* is generated into the agent's mind and a process of normative *influence* will be activated (Conte and Dignum, 2001). Such a behavior will be more likely to be replicated than would be the case if no normative belief had been formed (Andrighetto et al., 2007a). As shown elsewhere (Conte and Castelfranchi, 1999; Andrighetto et al., 2007a), when a normative believer replicates α, she will influence others to do the same not only by ostensibly exhibiting the behavior in question, but also by explicitly conveying a norm. People impose new norms on one another by means of *deontics* and explicit *normative valuations* (for a description see 8.2) and propose new norms (implicitly) by means of (normative) behaviors. Of course, having formed a normative belief is necessary but not sufficient for normative influence: we will not answer the question *why agents do* so (a problem that we solve for the moment in probabilistic terms), but we address the question how they can influence others to obey the norms (Conte and Dignum, 2001; Castelfranchi, 1999). They can do so if they have formed the corresponding normative belief, if they know how one ought to behave.

[36] For example, Posner and Rasmusen (1999) cope with the creation and destruction of norms, but with special reference to sanctions. The issue is certainly not uninteresting, but for the aim of this paper we prefer to focus just on detecting which are the cues that lead an agent to interpret a social behavior as normative, putting for the moment aside questions about sanctions and enforcement mechanisms.

Hence we propose that normative recognition represents a crucial requirement of norm emergence and innovation, as processes resulting from both agents' interpretations of one another's behaviors, and their transmitting such interpretations to one another.

8.3 The Norm Recognition Module

Our normative architecture (EMIL-A) (see Andrighetto et al., 2007a, and Chapter 9 for a detailed description) 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. Unlike BOID in which obligations are already implemented into the agents' minds, EMIL-A is provided with a component by means of which agents infer that a certain norm is in force even when it is not already stored in their normative memory. In this situation the norm has not already been incorporated into schemata, scripts, or other pragmatic structures (Bicchieri, 2006); hence, agents are not facilitated by any of these. Actually, the norm needs to be found out, and only thereafter, stored.

To implement such a capacity is conditioned to modeling agents' ability to recognize an observed or communicated social input as normative, and consequently to form a new normative belief. In this chapter, we will only describe the first component of EMIL-A, i.e. the norm recognition module (see Figure 15). This is most frequently involved in answering the open question we have raised, i.e. how a new norm is found out. Topic that we consider particularly crucial in norm emergence, innovation and stabilization.

Our Norm Recognizer (see Figure 15) consists of a long term memory (on the left), the normative board, and in a working memory (on the right), presented as a three layers architecture. The normative board contains normative beliefs, ordered by salience. With salience we refer to the degree of activation of a norm: in any particular situation, one norm may be more frequently followed than others, its salience being higher. The difference in salience between normative beliefs and normative goals has the effect that some of these normative mental objects will be more active than others and they will interfere more frequently and with more strength with the general cognitive processes of the agent[37].

The working memory is a three layer architecture, where *social inputs* are elaborated. Agents observe or communicate social inputs. Each input[38] (see Figure 14) is presented as an ordered vector, consisting of four elements:

- the source (X), i.e. the agent from which we observe or receive the input;
- the action transmitted (α), i.e. the potential norm;
- the type of input (T): it can consist either in a *behaviour* (B), i.e. an action or reaction of an agent with regard to another agent or to the environment, or in a *communicated* message, transmitted through the following holders:
  - assertions (A), i.e. generic sentences pointing to or describing states of the world;
  - requests (R), i.e. requests of action made by another agent;
  - deontics (D), partitioning situations between good/acceptable and bad/unacceptable. Deontics are holders for the three modal verbs analyzed by von Wright (von Wright, 1963): "may", indicating a permission, "must", indicating an obligation, and "must not", indicating a prohibition.
  - normative valuations (V), i.e. assertions about what it is right or wrong, correct or incorrect, appropriate or inappropriate (i.e. *it is correct to respect the queue*).
- the observer (Y), i.e. the observer/addressee of the input.

[37] At the moment, the normative beliefs' salience can only increase, depending on how many instances of the same normative belief are stored in the Normative Board. This feature has the negative effect that some norms become highly salient, exerting an eccessive interference with the decisional process of the agent. We are now improving the model, adding the possibility that, if the normative belief is inactive for a certain amount of time, its salience will decrease.

[38] It has to be said that the input we have modelled is far from accounting for the extraordinary complexity of norms.

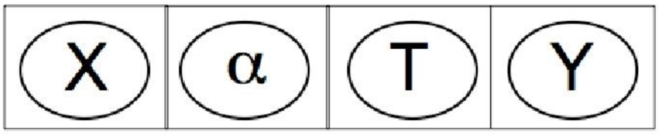


Figure 14. The social input

Once received the input from another agent, the agent will compute, thanks to its norm recognition module, the information in order to generate/update her normative beliefs. Here follows a brief description of how this normative module works. Every time a message containing a deontic (D), for example, "You must answer when asked", or a normative valuation (V), for example "It is impolite to not answer when asked", is received, it will directly access at the second layer of the architecture, giving rise to a candidate normative belief "One must answer when asked", which will be temporally stored at the third layer. This will sharpen agents' attention: further messages with the same content, especially when observed as open behaviors, or transmitted by assertions (A), for example "When asked, Paul answers", or requests (R), for example "Could you answer when asked?", will be processed and stored at the first level of the architecture. Beyond a certain normative threshold (which represents the frequency of corresponding normative behaviors observed, e.g. n% of the population), the candidate normative belief will be transformed in a new (real) normative belief, that will be stored in the normative board. The normative threshold can be reached in several ways: one way consists in observing a given number of agents performing the same action (α) prescribed by the candidate normative belief, e.g. agents answering when asked. If the agent receives no other occurrences of the input action (α), after a fixed time *t*, the candidate normative belief will leave the working memory (see *Exit*).

Aiming to decide which action to produce, the agent will search through the normative board: if more than one item is found out, the most salient norm will be chosen.

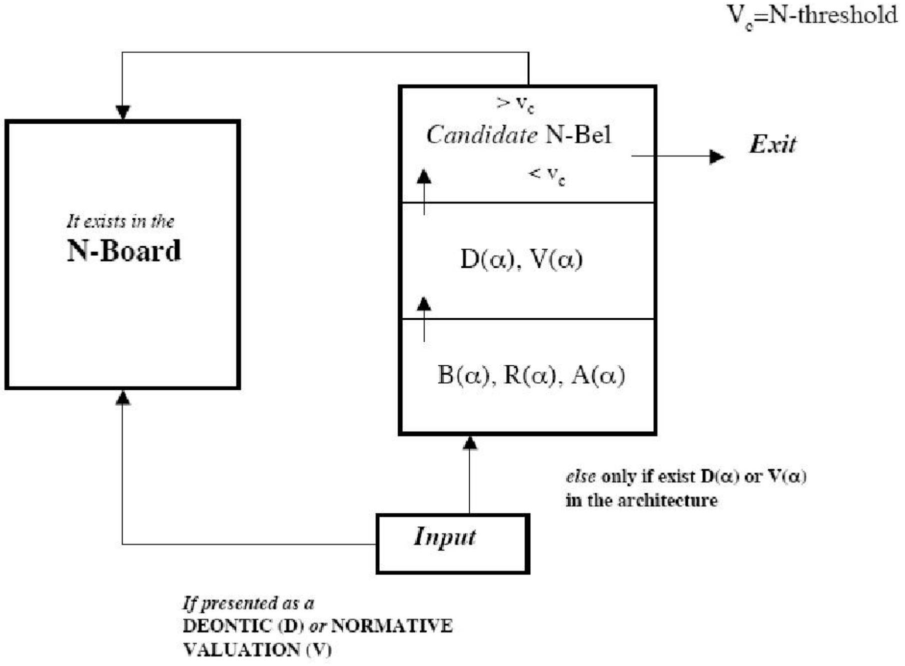


Figure 15. The norm recognition module (in action): it includes a long term memory (on the left), i.e. the normative board, and a working memory (on the right). The working memory is a three layer architecture, where the received input is elaborated. Vertical arrows in the block on the right side indicate the process regulating the generation of a new normative belief

8.4 The Computational Model

In our simulation model, the environment consists of four scenarios, in which the agents can produce three different kinds of actions. We define two scenario-specific actions for every scenario, and one action common to all scenarios. Therefore, we have nine actions. Suppose that the first scenario is a postal office, the second an information desk, the third our private apartment, and so on. In the first scenario the action *stand in the queue* is a scenario-specific action, whereas in the second a specific action could be *occupy a correct place in front of the desk*. A common action for all of the scenarios could be, *answer when asked*. Each of our agents (Norm Recognizers) is provided with a personal agenda (i.e. a sequence of scenarios randomly chosen), an individual and constant time of permanence in each scenario (when the time of permanence is expired, the agent moves to the next scenario) and a window of observation (i.e. a capacity for observing and interacting with a fixed number of agents) of the actions produced by other agents. Agents are also provided with the three-layer architecture described above (see Figure 15), necessary to analyze the received information, and a normative board in which the normative beliefs, once arisen, are stored. The agents can move across scenarios: once expired the time of permanence in one scenario, each agent moves to the subsequent scenario following her agenda. Such irregular flow (each agent has a different time of permanence and a different agenda) generates a complex behavior of the system, tick-after-tick producing a fuzzy definition of the scenarios, and tick-for-tick a fuzzy behavioral dynamics.

We have modeled two different kinds of environmental conditions. In the first set of simulations, agents can move through scenarios (following their personal agenda and in accordance with the personal time of permanence). In the second set of simulations, from a fixed time t, agents are obliged to remain in the scenario they have reached, till the end of the simulation: in this case agents can explore the scenarios exchanging messages with one another and observing others' behaviors. When they reach the last scenario at time *t*, they can interact with same-scenario agents till the end of the simulation. We hope this second setting allows us to show that the mere statistical frequency is sufficient (but not necessary) to the agents' convergence on the common action.

At each tick, the Norm Recognizers (NRs), paired randomly, interact exchanging messages (for a detailed description of the simulation scheduling, see the pseudo code description in appendix). These inputs are represented on an ordered vector (social input, see Figure 14), consisting of four elements: the source (x); the modal through which the message is presented (M); the addressee (y); the action transmitted (a).

Codifying the input in such a way allows us to (a) access the information even later, if necessary; (b) recognize the source, a piece of information that might be useful to store inputs from recognized authorities; (c) account for a variety of information, thanks to the modals' syntax; (d) compute the received information in order to generate a new normative belief. NRs produce different behaviors: if the normative board of an agent is empty (i.e. it contains no norms), the agent produces an action randomly chosen from the set of possible actions (for the scenario in question); in this case, also the modal by means of which the action is presented is chosen randomly. Vice versa, if the normative board contains some norms, the agent chooses the action corresponding to the most salient among these norms. In this case the action produced is presented with one of these modals: deontic (D), normative valuation (Vn) or behavior (B). This corresponds to the intuition that if an agent has a normative belief, there is a high propensity (in this chapter, this has been fixed to 90% of cases) for her to transmit it to other agents under strong modals (D or Vn) or open behavior (B). We run several simulations for different values of the threshold, testing the behaviors of the agents in the two different experimental conditions.

8.4.1 Results and Discussion

We briefly summarize the simulation scheme. The process begins by producing actions (and modals) at random. The process is synchronic. The process is more and more complex runtime: agent i provides inputs to the agent who precedes her (*k=1*), issuing one action and one modal. Action choice is conditioned by the state of her normative board. When all of the agents have executed one simulation update, the whole process restarts at the next step.

Simulations' Results

Figure 16 (a) and Figure 16 (b) show the trend of simulation in terms of number of agents in each scenario runtime in both cases (the first with the external barrier, the second without it).

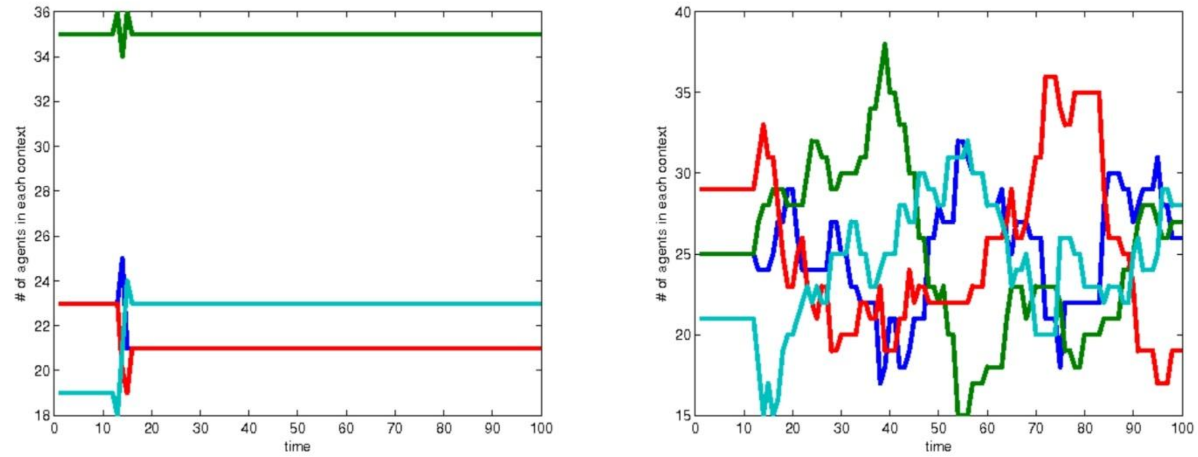


Figure 16. (a - on the left - and b - on the right). Number of agents in each scenario runtime – with (left) and without (right) external barrier

First of all we present the results obtained when imposing the external barrier. Then, we present the results obtained when no barrier was imposed; finally we compare the former with the latter results.

Figure 17 (a) shows the overall number of different new normative beliefs generated at the end of the simulation: as we can see, in the barrier condition (Figure 17 (a) on the left), agents form more than one normative belief, whereas in the no barrier condition (see Figure 17 (b), on the right) they form one normative belief only.

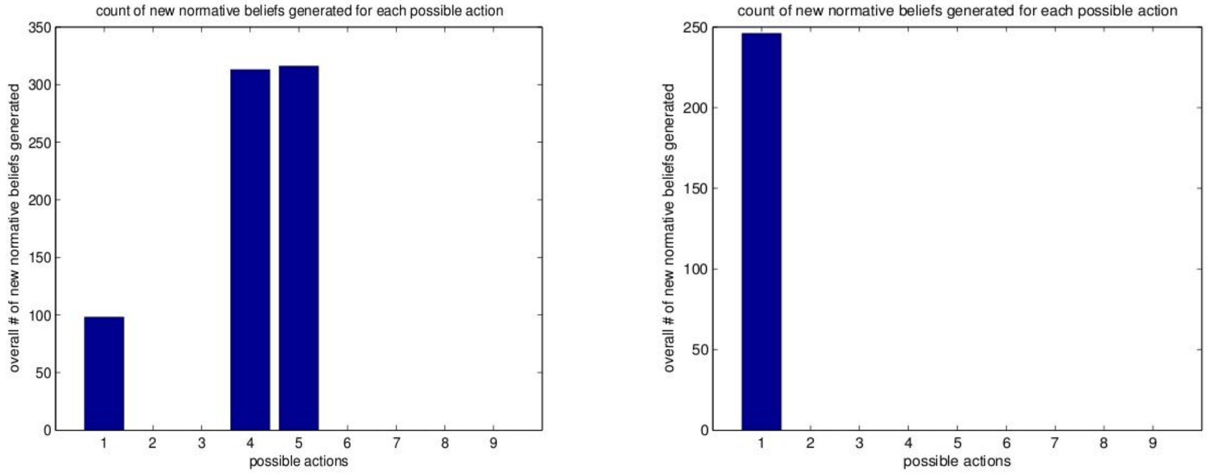


Figure 17. (a - on the left - and b - on the right). Overall number of new normative beliefs generated for each type of possible action - with (left) and without (right) external barrier

Figure 18 (a), on the left, shows the trend of new normative beliefs generation runtime for a certain value of the norm threshold (threshold = 99), which is a good implementation of our theory: each line represents the generation of new normative beliefs corresponding to an action (i.e. each line corresponds to the sum of different normative beliefs present in all of the agents). To be noted, a normative belief is not necessarily universally shared in the population. However, norms are behaviors that spread thanks to the spreading of the corresponding normative belief. Therefore, they imply shared normative beliefs.

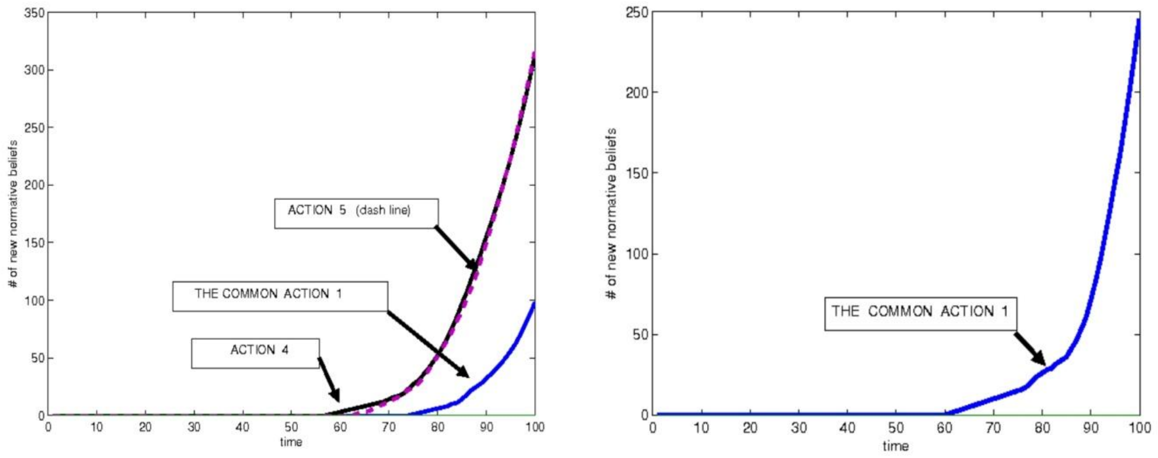


Figure 18. (a - on the left - and b - on the right). New normative beliefs generated runtime - with (left) and without (right) external barrier

Figure 19 (b) and Figure 19 (a) are very similar (even if in the no-barrier variant - Figure 19 (b), we find less regularity in the end of the dash line which represent the number of performed actions for the common action). In these plots, we cannot appreciate significant differences pointing to the normative beliefs acting on the effective behaviors: we cannot distinguish the clear effect corresponding to the agents' convergence on a specific norm (namely, we do not see that the dash line is significantly increasing).

Figure 20 (b) and Figure 20 (a) show that also the convergence rate in the case with and without barriers is quite similar.

This is due to the length of these simulations, which is not sufficient to include the latency time of norms. In the previous study (see Andrighetto et al., 2008a), indeed, we showed that for a normative belief to affect behavior, a certain number of ticks has to elapse, which we might call *norm latency*.

Indeed, if we run longer simulations, we can appreciate the consequences of the results of our investigation: in Figure 21 (a) and Figure 21 (b) we can observe two different (but related) effects: (a) more or less at the same time both in the barrier and no barrier condition, a convergence on the common action (dash line) is forming, much more significant in second case than in the first one; (b) however, in the barrier condition, other lines of convergence are also emerging (increasing). If we observe Figure 22 (a) and Figure 22 (b) we can appreciate that in the first case (the case with barriers) we find a very low convergence rate; but, in the second case (the case without barriers) we find a high convergence rate.

This corresponds to what is shown in Figure 18 (a) and Figure 18 (b) on one hand, and Figure 17 (a) and Figure 17 (b) on the other: with external barrier, we can see that the higher overall number of new normative beliefs generated does not correspond to the common action (action 1) and the trend of new normative beliefs generated runtime shows the same results.

With no external barrier, instead, only normative beliefs concerning the common action (action 1) are generated.

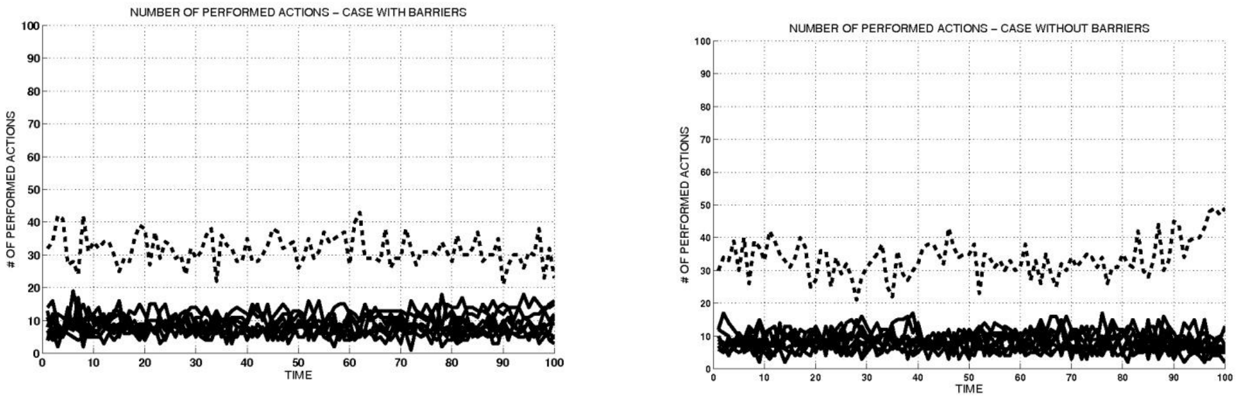


Figure 19. (a - on the left - and b - on the right). Actions performed by NRs. On axis X, the number of simulation ticks (100) is indicated and on axis Y the number of performed actions for each different type of action. The dash line corresponds to the action common to all scenarios

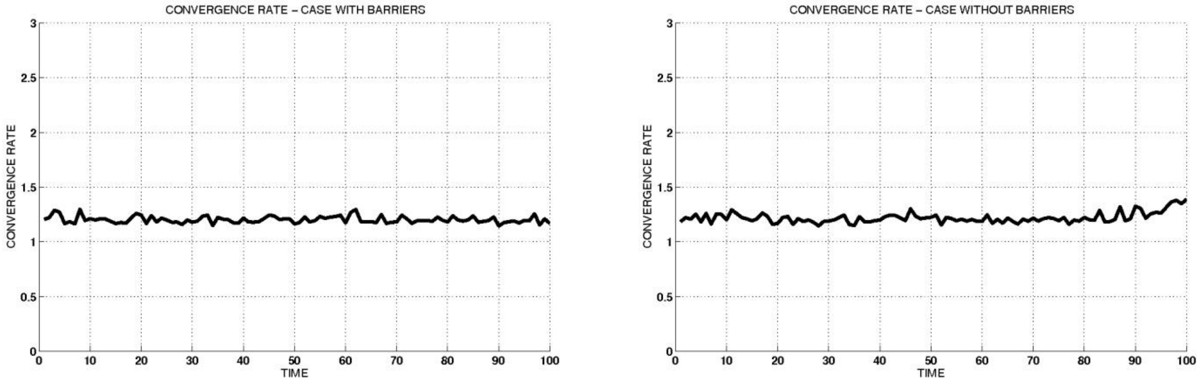


Figure 20. (a - on the left – and b - on the right). On axis X, the flow of time is shown; on axis Y the value of convergence rate

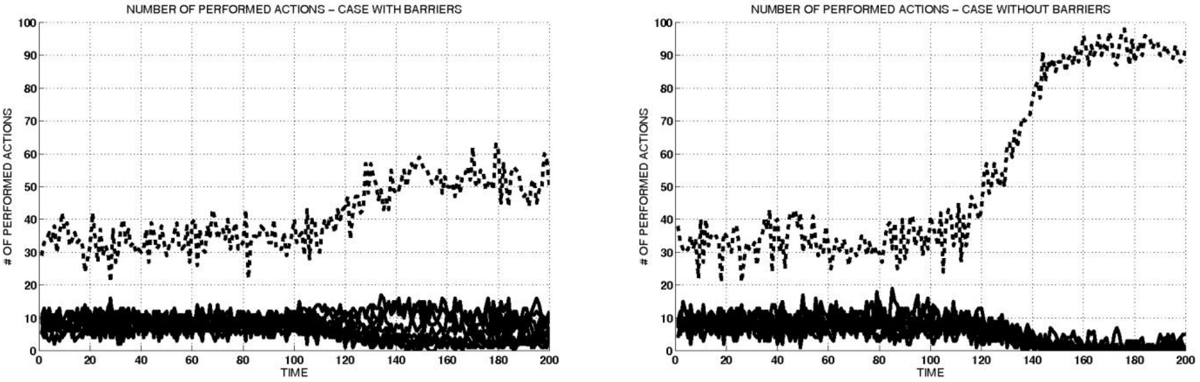


Figure 21. (a - on the left – and b - on the right). Actions performed by NRs. On axis X, the number of simulation ticks (200) is indicated and on axis Y the number of performed actions for each different type of action. The dash line corresponds to the action common to all scenarios

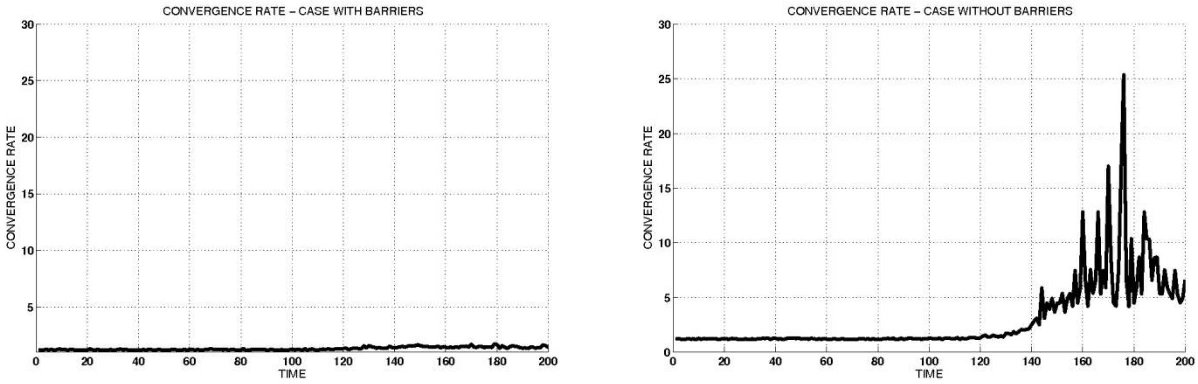


Figure 22. (a - on the left – and b - on the right). On axis X, the flow of time is shown; on axis Y the value of convergence rate

8.4.2 Concluding Remarks

We have shown how our model allows new norms, which do not corresponding to the common action to emerge. Some rival norms now compete in the same social settings. Obviously, they will continue to compete, unless some further external event or change in the population (e.g. the barrier removal) will cause agents to start migrating again. It would be interesting to observe how long the rival norms will survive after barrier removal, whether and when one will out-compete the others, and if so, which one. It should be said that, as we observe a latency time for a normative belief to give rise to a new normative behaviour, we also expect some time to elapse before a given behaviour disappears while and because the corresponding belief, decreasingly fed by observation and communication, starts to extinguish as well. We might call such a temporal discrepancy inertia of the norm. Both latency and inertia are determined by the twofold nature of the norm, mental and behavioural, which reinforce each other, thus preserving agents' autonomy: external barriers do modify agents' behaviours, but only through their minds.

More than emergence, our simulation shows a norm innovation process; in fact, Figure 19 (a) shows that, starting around tick=60, two normative beliefs appear in the normative boards and the overall number of these two new normative beliefs generated is three times higher than the overall number of normative beliefs concerning the common action 1. We might say that, if stuck to their current location by external barriers, norm recognizers resist the effect of majority and do not converge on one equilibrium only. Rather, they will form as many normative beliefs as there were competing beliefs on the verge of overcoming the normative threshold before the agents had been stuck to their locations.

No such effect is expected among agents whose behavior depends only from the observation of others. In sum, is statistical frequency sufficient for a norm to emerge? Beside action 1, common to the four scenarios, other norms seem to emerge in our simulation. Hume seemed to doubt it (Hume, 2007). Normative agents can recognize a norm; infer the existence of a norm by its occurrences in open behavior under certain conditions (see the critical role of previous deontics); and finally spread a normative belief to other agents. Future studies are meant to investigate on the effect of barrier removal and the inertia of normative beliefs.

8.5 Appendix

Pseudo code of the Simulation scheduling.

```
T = number of ticks
N = number of agents
for t from 1 to T
  for a from 1 to N
        do consult(agenda(a),time_of_permanence(a))
        do move(a,right_scenario)
  end
  do randomly_pair(N)
  for p from 1 to N/2
        do interact_exchanging_messages(a1,a2)
        do create_normative_belief(a1)
        do create_normative_belief(a2)
  end
  for a from 1 to N
        do perform_action(a)
  end
end
```

Chapter 9 Making the Theory Explicit: The EMIL-A Architecture

Giulia Andrighetto, Marco Campennì, Rosaria Conte

Abstract
So far, normative architectures did not render justice to the formation of normative beliefs, and more specifically to the norm recognition procedure. On the contrary, we claim this to be a fundamental aspect of norm emergence. Our normative architecture (EMIL-A) is meant to implement normative behaviour based on agents' capacity to find out, reason, decide and act upon norms. EMIL-A consists of mechanisms and mental representations allowing agents i) to form normative beliefs and goals, and decide whether to realize them or not, and ii) to be more or less reactive to external inputs by means of shortcuts. The EMIL-A architecture is accessed through the norm recognition module: before an input is recognized as normative, the norm cannot immerge in the minds of agents and, as a consequence, cannot affect their behaviours, nor *a fortiori* emerge in society.

9.1 The Normative Architecture: EMIL-A

In the EMIL project (but see the background theory in Conte and Castelfranchi, 1995, 1999, 2006), we claim that a norm emerges *as a norm* only when it is incorporated *into the minds* of the agents involved. In this sense, norm emergence implies norm *immergence* (Castelfranchi, 1998; Conte et al., 2007; Andrighetto et al., 2007a). Only when its normative, i.e. prescriptive, character is recognized by the agent, a norm gives rise to a norm-based behaviour. In opposition with a norm-corresponding behaviour, which not necessarily implies norm immergence (I may stop just while the traffic light was turning red only because I saw a friend passing by), by a norm-based behaviour we intend the output of a decision to comply with the norm. The normative decision in turn involves one or more normative representations.

One still insufficiently explored (see Broersen et al., 2001) aspect of norms is their mental representations and the mechanisms allowing them to rule the behaviours of autonomous intelligent agents (see the chs. Ontology and The State of Art on Norm Immergence). Norms not only regulate behaviour but also act on different aspects of the mind.

This chapter is aimed to provide an analysis of the so called *intra-agent* processes, i.e. the mental dynamics that norm-related representations undergo; attention will be drawn onto a normative architecture, EMIL-A, and on both its internal components for representing norms - such as beliefs, goals and obligations – and rules for their interaction. Finally, some words will be spent on normative routines and other shortcuts, speeding up and simplifying the execution of normative actions.

Figure 23 illustrates the components of EMIL-A, consisting of:

- three types of representations:
  - Normative Beliefs (N-Beliefs) (see the Glossary at the end of this chapter)
  - Normative Goals (N-Goals) (see the Glossary at the end of this chapter)
  - Normative Intentions (N-Intentions) (see the Glossary at the end of this chapter)
- four procedures:
  - Norm Recognition
  - Norm Adoption
  - Decision Making
  - Normative Action Planning[39].
- one inventory containing the Normative Board (N-Board) and the Repertoire of Normative Plans, together with knowledge about the world and general normative knowledge.

The outputs of EMIL are two different kinds of normative actions, compliance/violation and norm-defence.

[39] Regarding this module, we will not provide a description in this chapter since we will refer to the description included in Chapter 11 .

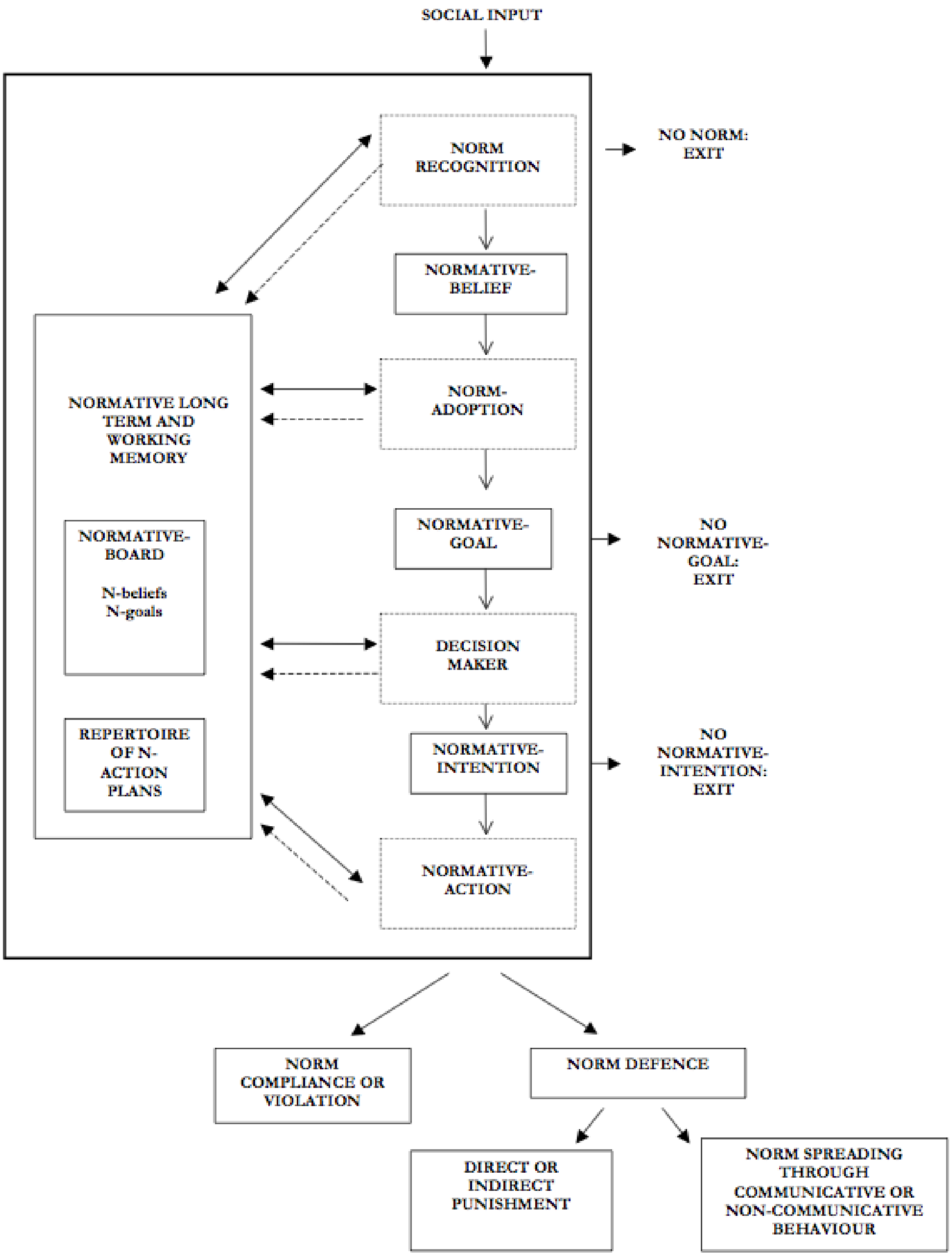

SOCIAL INPUT
NORM RECOGNITION
NO NORM: EXIT
NORMATIVE-BELIEF
NORM-ADOPTION
NORMATIVE LONG TERM AND WORKING MEMORY
NORMATIVE-BOARD
N-beliefs
N-goals
REPERTOIRE OF N-ACTION PLANS
NORMATIVE-GOAL
NO NORMATIVE-GOAL: EXIT
DECISION MAKER
NORMATIVE-INTENTION
NO NORMATIVE-INTENTION: EXIT
NORMATIVE-ACTION
NORM COMPLIANCE OR VIOLATION
NORM DEFENCE
DIRECT OR INDIRECT PUNISHMENT
NORM SPREADING THROUGH COMMUNICATIVE OR NON-COMMUNICATIVE BEHAVIOUR

Figure 23. The main components of EMIL-A are four different procedures, indicated by the dotted boxes, three mental representations and a long term memory, indicated by the thick boxes. Dotted arrows indicate activation-search; bi-directional thick arrows stand for storing; one-directional thick arrow stands for information flow.

9.1.1 Normative Mental Representations

In this section, we shall endeavour to clarify some components of the mental processing of norms.

Normative Belief

First of all, the process of norm immergence requires the formation of a normative belief, i.e., the belief that a given behaviour in a given context for a given set of agents is forbidden, obligatory, or permitted.

Assuming for simplicity that normative beliefs are explicit and expressed in a propositional form – like sentence in a language – a normative belief reads as follows: "there is a norm n_i prohibiting, prescribing, permitting that e_i/p_i" where e_i is an event and p_i a state of the world (Wright, 1963; Kelsen, 1979; see Conte and Castelfranchi, 1999, 2006). Indeed, norms are first aimed at and issued for generating the corresponding beliefs, thereby eliciting the prescribed behaviours. In other words, in order to work properly, norms must be acknowledged as such. Of course, a normative belief does not imply that a given norm has in fact been deliberately issued by some sovereign. Social norms are often set up by virtue of unwanted effects. However, once emerged, a given social norm gives rise to a general belief that some normative authority has put it into existence, if only an anonymous and impersonal one ("You are wanted, expected (not) to do this…": "It is generally expected that…"; "This is how things are done…", etc.).

In EMIL-A, normative beliefs, together with normative goals, are organized and arranged in a portion of the long-term memory, which we will call *normative board* (see section 9.3 in this chapter), according to their respective salience. By *salience* we refer to the norm's degree of activation, which is a function of the number of times a given norm enters the agent's decision-making (including the number of norm-invocation it receives see Chapter 8, section 8.3 and the simulation studies on Wikipedia and the Traffic).

Normative Goal Adoption and Normative Goal

However, a belief is not yet a decided action. Normative beliefs are necessary but insufficient conditions for norms to be complied with. What leads agents endowed with one or more normative beliefs to execute them, especially since, by definition, norms prescribe costly behaviours?

In the BDI (Beliefs-Desires-Intentions) approach to the computational modelling of human mind, introduced by the pivotal work of Rao and Georgeff (1992) and now extensively used in the agent systems environment (see State of the Art on Norm Immergence, intentions and actions originate only from desires. On the contrary, a great deal of our actions are not elicited by our desires but by external pressures and requests. Duties and norms are one of the external sources of our goals. How is this possible? How can norms generate goals?

From a cognitive point of view, goals are internal representations triggering-and-guiding action at once: they represent the state of the world that agents want to reach by means of action and that they monitor while executing the action (see Conte, 2009). If I want to deliver this manuscript in time, I will set to do the drafting, while continuously checking how far I am from deadline, how much is left to do, and whether the speed at which I am proceeding is a good compromise between accuracy and punctuality.

Under the effect of social inputs, goals can be generated anew via cognitive factors, as *relativized* to other mental states (e.g., social beliefs). A goal is relativized when it is held because and to the extent that a given world-state or event is held to be true or is expected (Cohen and Levesque, 1990)[40]. When goals are positive/pro-social, the process of generation is called *goal-adoption* (see Conte and Castelfranchi, 1995). By this, it is meant that an agent (adopter) comes to have another agent's (adoptee) goal as her own. When does an autonomous agent adopt the goal of another agent? When she believes to have good reason to do so, i.e., when she believes that the adoptee's achievement of his goal will increase the chances that she (the adopter) will in turn achieve one of her previous goals. The expected benefit does not always depend on reciprocity, but also on natural means-end relationships: I may lend my car to my roommate tonight, if I want to invite my fiancé for dinner.

There seems to be a correspondence between the process of *social goal adoption,* leading to adopt a request, and the process of *norm adoption*, leading from a normative belief to a normative goal: a normative goal of a given agent *x* about action *Ga* is a goal that *x* happens to have as long as she has a

[40] An example is the following: tomorrow, I want to go gather mushrooms (relativized goal) because and to the extent that I believe tomorrow it will rain (expected event). The precise instant I cease to believe that tomorrow it will rain, I will drop any hope to find mushrooms.

normative belief about *Ba*. More specifically, *x* has a normative goal only if she believes to be subject to a norm[41]. However, the reverse is not necessarily true: from *xBa* does not necessarily follow *xGa*.

Norms are aimed at obtaining adoption. This is why they are often transmitted as a communicated *will* (see Kelsen[42], 1979). In particular, they express the will that someone does a given action or brings about/maintains a certain world-state.

So far, so good. But why norm subjects decide to obey norms? We identified three main reasons:

- *instrumental* reasoning: prizes or sanctions enforcing the norm, including others' approval and reputation. However, sanctions are not defining elements of the norm, they simply enforce it. Agents reason about norms, and often adopt them on the basis of the norms' *cogency* - i.e. the criterion for choosing whether to execute a (normative) goal or not: the lower the costs of execution compared to the costs of violation, the more it is cogent. For example, car drivers can evaluate whether or not to stop at the red light when no vehicles or policemen are passing by and the chances of collision are low. In our view this is not how norms are *ideally* aimed to work (see Conte et al., 2009; Andrighetto et al., 2009). They are rather sub-ideal cases of normative influence based on norm-enforcement. A norm - be it social, legal or moral – is aimed at being adopted for compliance to the normative will, because it is a norm and "norms must be obeyed" (see von Wright, 1963). Of course this motive can be absent or weak in the minds of real people depending on the socialization and education process and the credit of current institutions. As people do not always follow the norms, norm-enforcing mechanisms are often necessary corollaries of normative imperatives[43].
- *Cooperation*: subjects adopt one particular norm because they share it.
- *Terminal* adoption: subjects want any norm to be respected.

Mental Path of the Norm

To have an idea of how EMIL-A works, a sketch of an ideal and complete mental path of a norm will be provided (see Figure 24).

Suppose mother and child walk along a park. Child is eating a snack when she notices someone littering a snack package on the ground. Imitating this behaviour, she drops her package too. Perceiving this rude action, mother scolds child, saying "It is forbidden to litter the rubbish!", "you have to throw the rubbish in the dustbin". As a consequence, it is possible that child forms an embryo of normative belief, at first confused with mother's will. Over time, more people throwing the rubbish in the dustbin child will see, the stronger her normative belief - that it is forbidden to litter the rubbish - will become. The child will have *recognized* a norm: this will become a *belief* in her mind, stating that litter the rubbish is prohibited.

Does it mean child will also adopt the norm? Not necessarily, unless she forms the corresponding goal. As said in section 9.9.1, norms work through *norm-adoption*: *x* generates a normative goal, only as relativized to a normative belief. If child had observed people around her throwing the rubbish into the appropriate container and had started to do the same, having no idea that such a behaviour is prescribed by a norm,

[41] A normative goal differs, on the one hand, from a simple constraint, which reduces the set of actions available to the system, and, on the other, from ordinary goals. With regard to behavioural constraints, a normative goal is less compelling: an agent endowed with normative goals is allowed to compare them with other goals of her and, to some extent, to choose which one will be executed. Only if an agent is endowed with normative goals she can be said to comply with, or violate, a norm. With regard to ordinary goals, a normative goal is obviously more compelling: when an agent decides to give it up, she knows she is both thwarting one of her goals and violating a norm.

[42] In Kelsen's view, the will is not to be meant in the psychological sense of an individual act of volition. The normative will outlives the act of volition. Here, we refer to the same view. Moreover, we believe norms to extress a sort of impersonal will, which is transmitted from one agent to another. In this sense, normative will not only outlives but also dispenses away with the act of volition.

[43] However, we do not intend to escape a fundamental problem, posed by the mechanisms of acceptance. In a cognitive approach, while giving a motivated and reason-based foundation to autonomous norm-obedience, one cannot ignore that even intentional actions become automatic, when habitual. The related behaviour is no longer really decided nor deliberated, but it is just executed as a response to the recognition of a given stimulus in a given context; and the corresponding action is performed under reduced controls and a higher attentive threshold.

she would have adopted no norm yet. Nor, which is the same, would she have formed any normative goal. Analogously, at first, child might pick up the package and throw it in the dustbin out of simple goal-adoption, to adopt, say, mother's will. Still, no full-fledged norm-adoption is at stake, in such a case, but only a weak forerunner: in a child's mind, as moral psychologists warn us, the adult's and the normative will are initially confused. Norm-adoption may gradually develop from goal-adoption according to a process that psychologists have described at length. The computational, or even the pre-computational, model of such a process, however, goes beyond the scope of the present project.

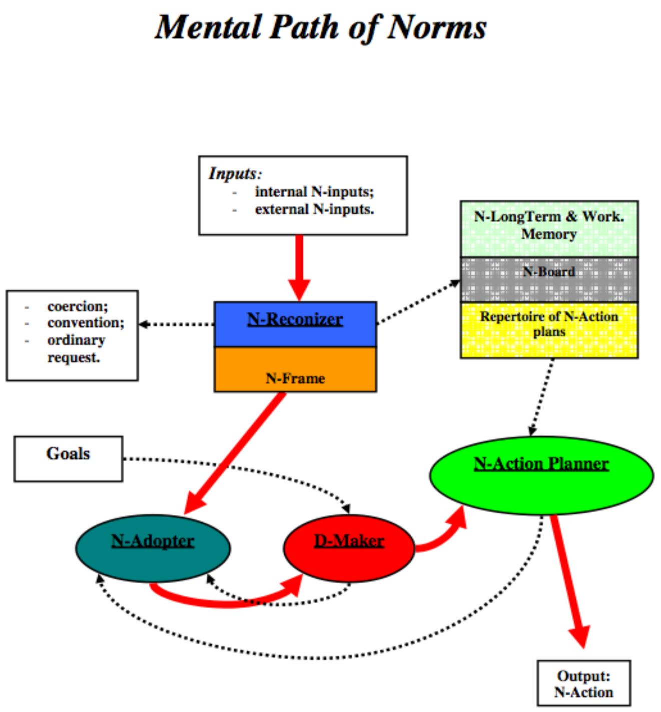


Figure 24. Thick red arrows represent the standard information flow. Dotted black arrows represent alternative directions of the information flow.

An agent endowed with normative goals is allowed to compare them with any other goal (*norm-decision maker*) of hers and choose whether to drop the former in favour of the latter or transform them into normative intentions, i.e. in executable goals. A normative goal can be transformed into a normative intention (and then into an action, i.e. a performed goal) depending on its cogency, comparing the costs of execution (giving up the potentially concurrent goals) with the costs of violation. The higher the latter, the more cogent the normative goal and the higher the probability that the decider will put the normative goal to execution. Obviously, positive and negative sanctions play a role in this decision, but other factors may interfere positively or negatively. Suppose child is walking along the park with a group of school friends. Having learned that littering the rubbish is inappropriate and rude, she behaves in accordance with her beliefs even when mother is absent. Now, she perceives no more the appropriate behaviour as a highly demanding task, and is usually willing to comply with it. But today, she observes her idols, her schoolmates, perform the opposite behaviour, and hears them laughing at a somebody picking up the packages they had littered, and put them into the closer dustbin. This time, child will find it hard to make her mind. What to do, obtain her clique's approval or follow the norm? What is harder to stand, their laughter or one's own disapproval?

Finally, once a normative intention is formed, norm subjects will execute it, performing either normative actions, observe the norm (compliance) or defending it. Norm defence in turn comes under two possible forms: urge others to obey the norm or apply direct or indirect punishment to transgressors, once the norm has been violated (defence).

So far, we illustrated the mental path of norms as a unidirectional process, proceeding from beliefs to actions. However, in real matters the reverse process, from actions to beliefs, is not infrequent. Sometimes, we imitate others for unclear reasons. For example, in non-familiar settings, we apply the policy of following others' behaviours for reasons that might become clear only later. Getting back to our example, if Mother were not there to tell Child what she *ought* to do, Child could take the use of the dustbin as a measure of prudence. Others use it, god knows why; it might be convenient, obligatory, prudent or simply customary. We'll see. For the time being, better do what others do. Over time and gradually, C will learn to ground her behaviour on a normative base. She will learn to justify it *normatively*. Until then, she performs a *norm-corresponding behaviour*, in accordance with a social norm, without knowing it. EMIL-A allows for such a later discovery of norms: in all of our simulations, a subset of agents form normative beliefs concerning actions they already (learned to) perform.

Let us know describe step by step each component of EMIL-A.

9.2 Norm Recognizer

The norm recognition module is the main entrance, so to speak, to the EMIL-A architecture[44]. Agents need to be able to discriminate between norms and other social phenomena, such as coercion, ordinary requests, conventions, etc. Our claim is that other normative architectures did not render justice to the recognition procedure (see Andrighetto et al., Forthcoming a; Campennì et al., 2009).

Simplifying, a given norm is recognized if current input

i. matches with a norm already stored in our (normative) memory;
ii. leads to a new norm being inferred or induced by the agent on the grounds of given indicators.

In the first case, the agent is facilitated by schemata, scripts, or other pragmatic structures (Wason and Johnson-Laird, 1972; Schank and Abelson, 1977; Fiske and Taylor, 1991; Barsalou, 1999; see Markus and Zajonc, 1985 for an overview) the norm is embedded in (see Bicchieri, 2006, for a description). Once these are activated for any reason, the corresponding normative beliefs, expectations and behavioural rules are prompted. If Child has the normative belief that to litter the rubbish is forbidden, she will be facilitated in choosing whether to follow the norm or not.

The second option is followed when such scripts, and consequently the corresponding pattern-matching operations, are not possible. The agent has no corresponding norm. This is why the norm recognition module is needed. Indeed, the norm-recognizer that we are going to describe attempts to answer the question how agents tell new norms, not yet stored in their memory (see also Sripada and Stich, 2006). Telling norms implies agents' ability to take an observed or communicated social input as normative, and consequently to form a new normative belief[45]. The first time Child receives an input concerning the use of the dustbin, she can store this information in her mind as a *candidate* norm, a first weak version of a normative belief (in this case, the role of Mother contributes to confer authority to this normative belief).

EMIL-A's module for norm recognition consists of a normative frame by which the received inputs are elaborated and interpreted, and a long term memory - called normative board - where normative beliefs and normative goals once formed are stored and ordered by salience (see section 9.3).

9.3 The Normative Board

When EMIL-A deals with an external input, such as a NO SMOKING sign, the norm recognition module will explore the N-Board. Suppose a corresponding normative belief is found (DO NOT SMOKE WHEN PROHIBITED), a normative belief is fired that will follow the path described previously. If Child meets someone littering the package of a snack on the ground, and the normative belief DO NOT LITTER THE

[44] See (Andrighetto et al., Forthcoming a; Campennì et al., 2009) and chapter "The added value in Normative agents" for an implementation of the Norm Recognizer.

[45] Concerning the cues that the norm recognition mechanism is able to identify and interpret as normative inputs it is an important topic for discussion (see Sripada and Stich, 2006 for a discussion; Blair, 1995; Blair et al., 1997; Edwards, 1987). See (Andrighetto et al., Forthcoming a; Campennì et al., 2009) and chapter "The added value in Normative agents" for a description and formalization of the normative input.

RUBBISH already exists in her normative board, she can choose to defend this (social norm), for example by reproaching the inappropriate behaviour and who performed it.

The normative board is an archive in the long-term memory where active norms are stored, arranged according to the *salience* gained[46] (by salience we refer to the degree of activation of a norm: in any particular situation, one norm may be more frequently followed than others, its salience being higher, cf. section 9.1.1). Difference in salience has the effect that a subset of norm-related representations interferes more frequently and strongly with the general cognitive processes of the agent. To decide which action to execute, the agent will search through the normative board: if more than one item is found out, the most salient norm will be chosen. Suppose Child's normative board also contains the norm DO NOT SPEAK TO STRANGERS; if the salience of the latter is higher than the previous one's, Child will be less likely to reproach the passenger who littered the package on the ground.

If an agent has a normative belief in her board, but has never adopted it – i.e., never formed the corresponding normative goal – the salience of this norm will decrease, and sooner or later the normative belief will decay (see section 2.1). On the contrary, a norm that is frequently processed by the decision-maker will increase in salience. If Child observes a lot of people performing the action suggested by Mother (THROW THE RUBBISH IN THE DUSTBIN), the salience of the normative belief corresponding to this action will increase. As we will see in the final chapter, salience may increase to the point that the norm becomes internalized, i.e. converted into an ordinary goal, or even in an automated conditioned action, a routine. In such a case, the norm will exit the normative board, as it has been incorporated to agent's ordinary actions.

9.4 Norm Adoption

Imputed by normative requests, the agent will generate a normative goal thanks to the norm adoption procedure (see the *normative goal adoption* procedure described in section 9.1). This does not imply, by the way, that the request will certainly be complied with. Our claim is that whenever an input gives rise to a normative belief, a process of norm-adoption will be activated. Of course, this does not mean that the norm will certainly be adopted. In the present model, norms are adopted *unless* agents have good reasons *not* to do so. In short, we believe agents have a sort of weak disposition, a positive default, to take normative requests into account and adopt them forming a corresponding normative goal, with a value corresponding to the norm's salience.

Norms have a *motivational effect*. This claim is supported by evolutionary psychologists (see for example, Cosmides and Tooby, 1992), who refer to this motivation as *intrinsic* and granted by an innate normative module[47]. The motivational nature of the norm can be better understood if we explode norm-based complex representations in their components - normative beliefs, goals and intentions - and pay attention to the mental path they follow and the procedures and rules that make possible their elaboration. Unlike ordinary adoption, in which agents must have positive reasons for adopting others' requests, norms are grounded on the obligation itself: if one recognises an input as normative, one believes there is a good reason, however feeble, for accepting it. The value of a normative goal may be recomputed later on, while taking decision about whether to comply with the norm in question or not (see above the discussion of Child's decision-making when she is in a *bad* company).

9.5 Decision Maker

Every time a goal is activated, the decision-making is called into question (except when shortcuts are taken, see section 9.9.3).

[46] Concerning the representational format through which norms are stored, we assume that it is a sentence-like format, with a formalism of a deontic logic. However, we believe that it is very much open question whether this is the way in which norms are typically stored. The recent literature on the psychology of categorization suggests a number of plausible alternatives, such as the exemplar theory (Smith and Medin, 1981; Murphy, 2002) or the prototypes, stereotypes, theories and narratives (see Murphy, 2002 for a comprehensive review).

[47] To this view, we would like to object that the existence of a norm module is either too strong or insufficient: it is too strong because it leaves no room to autonomy and norm violation. It is insufficient because little is said about how it effectively works: what are norms? How are they learned? What is their internal processing, the path they follow in the mind?

The decision-making module checks against potential obstacles to the goal's pursuit. There are two such obstacles, material impossibility and goal conflict. In the former case, the current goal is interrupted (possibly, the planning module is queried for alternative or preliminary actions). In the latter priorities are computed. If no obstacle is found, the goal is put to execution, either by applying existing plans or routines, or by planning anew. Child may not find the dustbin (and in this case she could choose to put the package of the snack in her pocket) or, as discussed before, she could be ashamed to be the only one who throws the package of the snack in the dustbin.

There are factors reinforcing or weakening goals. In particular, the value of an active goal may be temporarily strengthened by sanctions of variable entity and probability, or weakened by potentially incompatible goals. Suppose that at night, while approaching a crossroad, I see the traffic light turning red. It is late in the night and neither cars nor pedestrians are visible. It is also most unlikely that any policeman is observing me. In this situation, the value of the normative goal decreases with cogency - i.e. the criterion for choosing whether to execute a (normative) goal or not. By definition, cogency is higher when the costs of violation exceed those of execution. In our example, we can have two different agents:

i. a recently qualified driver, for whom the norm *stop-if-traffic_light-is-red* is highly cogent as she feels uncertain in driving;
ii. an expert driver, who feels confident at driving and finds the norm-compliance less cogent.

Consider also that if agents perceive some norm violation around, the value of the corresponding normative goal will decrease and they will proportionately be discouraged from observing it.

9.6 Norm Defence

Agents compliant with a norm are likely to defend it. In many circumstances, a compliant agent will exercise a special form of social control, getting others to comply with the norm, reproaching transgressors and reminding would-be violators that they are doing something wrong. Social control probably stems from a special case of the equity principle, according to which people do not want others in their own conditions to sustain lower costs, benefits being equal. For one's costs not to exceed others', the compliant agent wants other subjects either to observe the norms, or to be punished. Punishment can be either direct – sanctioning - or indirect - spreading a bad reputation of the observed violators.

Norm defence is extremely important in spreading norms over a population of autonomous agents (see Conte and Dignum, 2001). The larger the number of agents conforming to one given norm, the more they will be likely to urge others to conform with it. On the other hand, observed norm violation discourages compliance. If agents are driven to defend, directly or indirectly, the norm they have honoured, they are likely to ignore those that others have violated. If one's previous compliance leads to norm defence, others' violation turns deciding agents into partners in crime. It will be very difficult to defend the norm DO NOT LITTER THE RUBBISH in a population where the majority make no use of dustbins.

9.7 State of the Art of Existent Normative Architectures[48]

Usually, in the formal social scientific field, that is in utility and (evolutionary) game theory (Bicchieri, 2006; Epstein, 2006; Sen and Airiau, 2007; Ullman-Margalit, 1977; Young, 1998), the spread of new social norms and other cooperative behaviours is not explained in terms of internal representations. The object of inquiry is usually the conditions for agents to converge on given behaviours, which proved efficient in solving problems of coordination (Lewis, 1969) or cooperation (Axelrod, 1984), independent of the agents' normative beliefs and goals (Binmore, 1994). In this field, no theory of norms based on specific mental representations has yet been provided.

Game theorists essentially aimed to investigate the dynamics involved in the problem of norm convergence. They considered norms as conditioned preferences, i.e. options for actions preferred as long as those actions are believed to be preferred by others as well (Bicchieri, 2006). Here, the main role is played bysanctions: what distinguishes a norm from other cultural products like values or habits is the fact that norm-adoption is enforced by many phenomena, including sanctions (Feld, 2006; Axelrod, 1986). The

[48] For a more detailed comparison with other normative architectures, see Chapter 7 .

utility function, which an agent seeks to maximize, usually includes the cost of sanctions as a crucial component.

In the field of Multi-Agent Systems (Dignum, 1999; Jones and Sergot, 1996; Van der Torre and Tan, 1999), instead, norms are explicitly represented. However, they are implemented as built-in mental objects, answering the question how autonomous intelligent agents represent and reason upon norms. Even when norm emergence is addressed (Savarimuthu et al., 2007), the starting point is some preexisting norms, and emergence lies in integrating them. When agents (with different norms) coming from different societies interact with each other, their individual societal norms might change, merging in a way that might prove beneficial to the societies involved (and the norm convergence results in an improvement of the average performance of the societies under study against some relevant measures of well-being, stability, etc.).

Of late, decision making in normative systems and the relation between desires and obligations has been studied within the Belief-Desire-Intention (BDI) framework, developing an interesting variant of it, i.e. the so-called Belief-Obligations-Intentions-Desires or BOID architecture (Broersen et al., 2001). This is a feedback loops mechanism, which considers all effects of actions before committing to them, and resolves conflicts between the outputs of its four components. Examples for such an approach are given in Broersen et al. (2005), Boella (2001), López y López et al. (2002). Obligations are introduced to constrain individual intentions and desires on the one hand, while preserving individual autonomy, on the other. Agents are able to violate normative obligations, based on their capacity to reason upon norms.

In none of these approaches including the last one, however, it is possible for an agent to tell when a given input is a (new) norm. On the contrary, obligations are hard-wired into the agents' minds when the system is off-line. Unlike the game-theoretic model, multi-agent systems exhibit all of the advantages deriving from an explicit representation of norms. Nevertheless, we claim that the existing BDI approach suffers from some limitations, which have not only a theoretical, but also a practical and computational relevance.

First, multi-agent systems overshadow one of the advantages of autonomous agents, i.e. their capacity to filter external requests. Such a filtering capacity affects not only the decisions about norms already acquired, but also the acquisition of new norms. Indeed, agents take decisions even when they decide to form normative beliefs, and then new (normative) goals, and not only when they decide whether to execute the norm or not (Conte et al., 1998).

As to the practical relevance, if agents are enabled to acquire new norms, there is no need for exceedingly expanding their knowledge-base, since they may be optimized on-line (Shoham and Tennenholtz, 1992)[49]. Despite the undeniable significance of the results achieved, these studies leave some fundamental questions still unanswered, such as how and where norms originate, how agents acquire norms, and more specifically, how agents tell that something is a norm. Our feeling is that the question how norms are created and innovated has not received so far the attention it deserves. We claim that this circumstance may be ascribed the way the normative agent has been modelled up to now.

9.8 Value Added of EMIL-A

So far, the study of norm emergence has been identified with the study of behavioural regularities (see section 9.7). However, not all the regularities are mandatory, and not all the norms are observed. Hence, the logical and pragmatic priority is how agents find out what the *normative* regularities are. Only afterwards, does it make sense to model the reasons why they conform to them. The value added of EMIL-A is to account for this specific aspect of norm-based regulation, how agents find out the norms they decide whether or not to conform to.

[49] In a successive work (Shoham and Tennenholtz, 1994), indeed, these authors have introduced the notion of colearning, which refers to a process in which several agents simultaneously try to adapt to one another's behavior so as to produce desirable global system properties. Of particular interest are two specific co-learning settings, which relate to the emergence of conventions and the evolution of cooperation in societies, respectively. Despite the indubitable significance of this work, the treatment of norms as emerging conventions resulting from co-learning processes, can only deal with how preexisting actions are gradually generalized or dropped.

Norm recognition is an important requirement of norm-emergence. In previous works (see also Castelfranchi, 1998; Conte et al., 2007), emergence has been defined as a gradual and complex dynamics by which the macro-social effect, in our case a specific norm, is brought about in society *while* immerging in the minds of the agents, generating it through a number of intermediate loops.

Unlike moral dispositions, norm-recognition is poorly sensible to subjective variability, and rather robust. It allows us to (a) account for the universal appearance of norms in human and primate societies; (b) render justice to the intuition that humans violate norms, but have little problems in finding them out; (c) account for the evolutionary psychological evidence (see Cosmides and Tooby, 1992, 2008) that agents easily apply counterfactual reasoning to social rules, but find it difficult to do so with logical ones; finally, (d) explain why, as pointed out by developmental psychological data, norm acquisition follows a stable ontogenetic pattern starting quite early in childhood (Bandura, 1991; Nucci, 2001; Cummins, 1996; Piaget, 1965; Kohlberg, 1981; Kohlberg and Turiel, 1971; Henrich et al., 2001; Shweder et al., 1987).

In short, the intuition behind our normative architecture is twofold: on the one hand, the emergence of norms is based upon a universal capacity to tell norms; on the other, this capacity is supported by a norm frame, an internal "model of a norm", which agents use as a processing instrument in norm recognition (see Chapter 8).

The emphasis laid on the innate and universal features of EMIL-A should not be mistaken, leading to think that no space is left to subjective variability. If norm recognition is a must, equally accomplished by a vast majority of agents, moral attitudes - i.e. the results of normative and moral experience accumulated during lifetime that affect different normative procedures - are not. They are definitely subjective.

Furthermore, the reinforcement effects that occur on different EMIL-A procedures vary among agents. Personal experience, for example, impacts on norm salience. Analogously, the normative frame, being in constant interaction with the social environment and the other procedures, is liable to their influence. In these terms, a normative architecture is allowed to elegantly ignore the culture/nurture controversy.

9.9 EMIL-A in Use

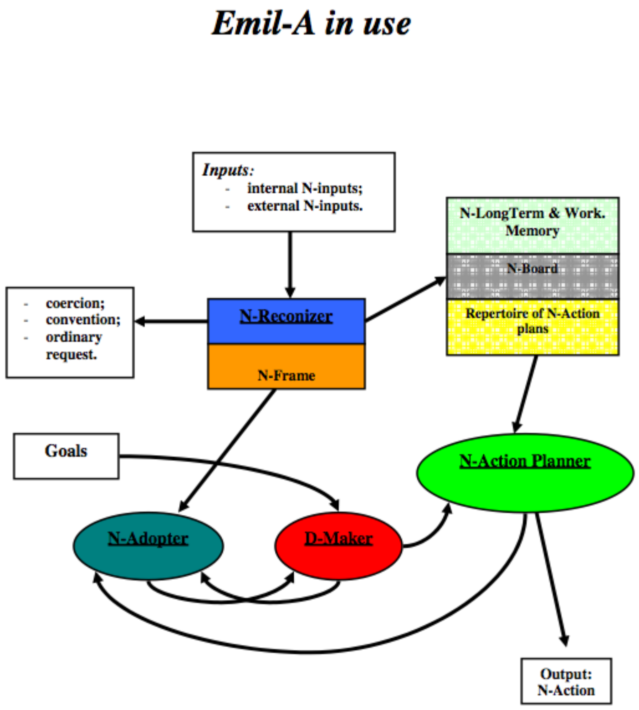


Figure 25. Tick black arrows represent all of the possible ways for flowing information

A crucial aspect of the agent architecture is the occurrence of interruptions, modifications and deviations from the processes described so far (see Figure 25).

We will examine three specific phenomena:

- decay, i.e. the process by means of which a given norm decays leaving no trace in the mind,
- internalization, i.e. the process by means of which a norm is transformed into an internal motivation. For a detailed description of this mechanism, see chapter *"On norm internalization"*
- shortcuts, occurring when one or more decisions in the ordinary processing are bypassed; in particular, we will examine at some length reactive normative behaviours i.e. routines fired under given input conditions.

9.9.1 Decay

Are norms permanent objects in agents' minds? Certainly not. As any other representation, norms may be acquired, modified and lost. Norms may disappear under the effect of cognitive and non-cognitive mechanisms.

Among the cognitive factors, we enumerate electro-chemical, physiological and traumatic phenomena: people affected by post-traumatic, post-surgical or chronic neurological disorders may exhibit behavioural anomalies and socio-pathologies consequent to a loss of social norms and conventions (Damasio, 1994; Anderson et al, 1999).

As to the multiple cognitive mechanisms responsible for a radical loss of norms, we would like to call the reader's attention at least to the following:

- norm-revision: reasons leading to a given input being recognised as a norm are found no more adequate. This may occur with
    - Perceived error in previous recognition: observer is led by the state of the world currently perceived (current behaviour of the source) to reconsider and revise previous interpretations.
    - Perceived change in the current state of the world: for example, observer perceives a modification in the source's behaviour.
    - Simple forgetting, generally associated to a gradually reduced salience of the norm, no longer fit to a changing environment.
- norm-revocation: the reasons that led adopter to accept a norm are found no more adequate. Again, this includes
    - Revised norms: decider finds the norm to be no more compatible with decision-based modifications of its normative board.
    - Modified personal goals: reasons for adopting the norm are insufficient if the decider's goals have changed in the meantime. This may also be due to a non-cognitive cause, such as the normal course of life.

To implement norm-invocation requires more or less complex mechanisms. A simple solution is to include a forget rate, among the agent parameters: if the salience rate of a norm remains very low for a certain period of time (to be specified), then the norm is bound to decay from the agents' normative board. However, more interesting solutions for norm revision and revocation exist. Indeed, norm-revision is already included in the norm-recognition process, as this is a non-linear process by means of which a given input (either a prescription or a regularly observed behaviour) is analysed.

9.9.2 Internalization

This is the process by means of which a norm is transformed into an internal motivation. For a detailed description of this mechanism see Chapter 20.

9.9.3 Shortcuts. Normative Routines

Normative routines are (semi) automatic executions of norms, fired whenever a given (subset of) input(s) is recognised. Automatisms are frequent in intelligent systems, including humans. What makes them possible is far from clear, whereas it is relatively easier to answer the question as to when it is likely to occur: the better and more frequent a given behaviour, the more likely it is performed as an automatic, non-controlled (i.e. non-decided upon and inadvertent) routine (Bargh, 1992). In the case of norms, the more frequent a certain normative action, the more likely it will become a routine. For example, when stopping at a red streetlight, in many cases we do not really take any decision. After a while, such behaviour becomes an automatism, a reflex, something which is accomplished inadvertently: the recognition of the stimulus is enough for the action to be fired. However, during the learning process (necessary to establish even a mere conditioning, a simple reflex behaviour) subjects need at first to explicitly formulate a goal - what they want to do - that coincides with the norm. For example, while learning to drive and learning to recognize and appropriately react to signals (e.g., "give precedence"), subjects will formulate the goal to slow down and let others move on. Only with later practice, a direct shortcut develops between stimulus and behavioural response. Thus, in some circumstances the normative action is performed thoughtlessly, on condition that some control mechanisms stay active. If for example normative conflicting inputs are detected, control mechanisms lock the shortcut, reactivating the standard path. This means that we can bring this action back under conscious control and real deliberation to evaluate the circumstance and decide to obey after having considering the possibility and convenience of transgression. Norm obedience cannot be reduced to an instinct or any uncontrolled automatism, although sometimes it acquires the features of it (see Epstein, 2000; Epstein, 2006 for an alternative view). However, for a treatment of normative routines, see also Chapter 20

9.10 Conclusions and Future Works

In this chapter we have proposed a normative architecture, EMIL-A and we have illustrated how it allows for existing norms to be recognized and complied with (see Figure 25 for alternative directions of the information flow). We are aware that a number of advances would greatly enhance the scientific interest and plausibility of this architecture. In particular,

- a deeper integration between EMIL-A components;
- a model of the interplay between the normative and the non-normative representations;
- a model and implementation of normative emotions, for example shame and guilt
- a model of the moral aspects and in particular of
    - emotions (such as remorse, the sense of moral duty, etc.)
    - innate moral dispositions

and how they both impact on different aspects of a normative architecture.

9.11 Glossary

Adoption rule (AR), a corollary of the GGR; if an agent believes that adopting a goal of a given agent is a means for his obtaining one of his own goals, he will adopt that goal.

Autonomous agent, endowed with the capacity to generate and pursue its own goals.

Beliefs, states of the world as it is (to a variable degree of certainty). These may be:

- non social;
- social, about others, including their mental states.

Goals, wanted states of the world that might or not be verified.

Goal-adoption rule (GAR), according to which an agent adopts another agent's goal if she believes that the latter achieving this goal will lead her to achieve one of hers.

Goal generation rule (**GGR**), an agent will have as a goal any new world state if she believes this to lead to a previous goal of hers being achieved.

Intentions, executable goals.

Salience, a norm's degree of activation, i.e. the number of times that norm is decided upon

Chapter 10 Supporting the Theory: The Derivation of EMIL-S from EMIL-A (From Logical Architecture to Software Architecture)

Ulf Lotzmann, Michael Möhring, Klaus G. Troitzsch

Abstract

This section describes the process of converting the logical architecture of EMIL-A (Chapter 9) — the model of the process going on in a human actor's mind when he or she receives norm invocations or observe other actors' behaviour – into a software architecture which can execute simulation models of social processes of norm emergence in different kinds of scenarios. The structure of this section is as follows: First a short overview of the logical architecture of EMIL-A is given, followed by a list of correspondences between the main terms of both models. In the end the remaining differences between the EMIL-A design and the EMIL-S implementation are discussed, and it is argued that these differences are mainly technical and that EMIL-S necessarily implemented more details than EMIL-A foresaw.

10.1 Overview of the Logical Architecture of EMIL-A

Chapter 9 introduced the logical and cognitive architecture of a normative agent. To derive a software architecture from this logical architecture, several steps are necessary. It is not sufficient just to implement the four procedures of EMIL-A:

- norm recognition
- norm adoption
- decision making
- normative action planning

and define data structures for the three types of representations:

- normative beliefs
- normative goals
- normative intentions

in the individual agents and the normative board as a central inventory of norms. Instead we have to start with the idea that for norms to emerge and to undergo innovation it will be necessary that agent societies must not consist of agents that are entirely lenient with respect to the behaviour of their fellow agents..Thus agents will have to be endowed with a set of goals which they do not necessarily share with all of their fellow agents.

Goals (see Chapter 9 and Conte, 2009) "are internal representations triggering and guiding action at once: they represent the state of the world that agents want to reach by means of action and that they monitor while executing the action." Thus the process of norm emergence or innovation in an artificial society of agents will have to start with actions arising from individual agents' goals.

To illustrate the process going on in what one could call a primordial artificial society we will use an everyday example which is very similar to but not identical with the one used in Chapter 9: A does not want to be exposed to the smoke of cigarettes (her goal is a state of her environment which does not compel her to inhale smoke and which makes her cough).At this moment this is not yet a normative goal[50], but it has a similar consequence as the one described in EMIL-A: to achieve the goal of living in a smoke-free world when the current environment contains a smoker, say B, a decision has to be taken which leads to one of several possible intentions which in turn lead to respective actions. One of the possible decisions A might take will be to demand from B, the smoker, to stop smoking at once and to abstain from smoking in A's presence in all future. When B receives this message as a social input he will have to evaluate this message in the norm recognition procedure. If this event (A asks B not to smoke in her presence) is the first of this

[50] In terms of EMIL-A, this is the "alternative information flow" from "Goals" to "D-Maker" in Figure 25 of Chapter 9 .

kind, B will not recognise a norm but store this message and the situation in which he received it as an event in his event board (an ingredient not explicitly mentioned in EMIL-A). When an event like this is more often observed by B (but also by observers C, D, …) this kind of messages might be interpreted (or recognised in terms of EMIL-A, "inferred or induced by the agent on the grounds of given indicators" Chapter 9) as a norm invocation, and a normative belief — "the belief that a given behaviour in a given context for a given set of agents is forbidden, obligatory, permitted, etc." (see Chapter 9) — is stored in all the recipients of the repeated message.

As soon as a social input (such as a message from another agent in a certain situation) is recognised as a norm invocation EMIL-A generates a normative belief which may (or may not) be adopted, i.e. transferred to the individual normative long term and working memory which consists mainly of the individual normative board for storing normative beliefs and normative goals). If it turns out that the current state of the world does not conform with the normative goal derived from the adopted norm, it is the turn of EMIL-A's decision maker to select from a repertoire of action plans — which in the case of our artificial primordial society must be predefined. The decision maker generates a normative intention which in turn ends up in an action. EMIL-A foresees that these actions can be

- either of the norm compliance or violation type: actions which influence some physical environment
- or of the norm defence type: actions which lead to norm invocations, direct or indirect punishment or just norm spreading through communicative or non-communicative behaviour.

And as a matter of course, an initial repertoire of action plans must be available in each of the agents of the artificial agent society.

10.2 Correspondence between EMIL-S and EMIL-A

Without going into the details of the EMIL-S implementation, we can say that the EMIL-S architecture is much the same as the EMIL-A architecture. The concept of messages (which is not an integral part of EMIL-A as EMIL-A is devoted to intra-agent processes) as described in Chapter 8 is implemented exactly as in the logical architecture (see Chapter 11). The norm recogniser module as described in detail in Chapter 8 is implemented in EMIL-S also using two distinct agent memory with a functionality similar to the EMIL-A layers (see Chapter 11, event board and valuation history as part of the normative frame). But other details of EMIL-S had to be implemented in a less straightforward correspondence to EMIL-A.

In order to implement the two different capabilities of EMIL-A's norm recogniser for the two cases where current input

- "matches with a norm already stored in … memory;
- leads to a new norm being inferred or induced by the agent on the grounds of given indicators"

it turned out reasonable to implement a complicated process involving of the event board and the (individual) normative frame (see Chapter 11). These interdependent features fulfil the functions of EMIL-A's norm recogniser.

The norm adopter in EMIL-A is implemented in EMIL-S in terms of a reinforcement learning procedure changing probabilities in event-action trees which in turn implement EMIL-A's repertoire of action plans (but these event-action trees also play an additional role, see below).

The role of EMIL-S's event-action trees is somewhat more complex than EMIL-A's repertoire of N-action plans as the event-action trees are responsible for action planning, not only normative action planning. Scenarios run under EMIL-S must also reflect non-normative behaviour of agents which are not covered by the EMIL-A concept. Thus it seemed reasonable to realise all decision making and action planning in the same engine.

10.3 Differences between the Logical and the Implemented Model

The main difference between EMIL-A and EMIL-S lies in the fact that the former only addresses the mind of human actor and does not inform about the interface between these agents and their environment. For a simulation to be run it is necessary to endow software agents with at least some of the capabilities that human actors have by nature: perceiving at least part of the state of the simulated world and acting, i.e. changing the state of the simulated world. Although EMIL-S – as Chapter 11 will show in much more detail – restricts itself to model the mind of human actors whereas modelling the body of human actors is the task of the simulation system below EMIL-S (TRASS; MASS, Repast etc.), agent design in EMIL-S has to include modelling that goes beyond modelling the normative processes described so far in this chapter and in much more detail in 14.3.1 and 15.3.1

Pure EMIL-A agents, coming together in a simulation environment, would either be identical (thus no new norms could emerge as all of them share the same normative board or frame and norm violations would be extremely unlikely) or they would not be able to understand each other because they would not share knowledge about each other and their environment. Real human actors, even if they came from entirely different cultural backgrounds, share at least part of their knowledge about their environment and even of each other with their fellow actors. Thus EMIL-A agents have to be endowed with the knowledge they need to behave and act reasonably in their environment. The place where EMIL-S accommodates this knowledge is mainly the collection of event-action trees which are also used for the normative processes going on in the software agents but which contain also the initial behavioural rules necessary for “living” in their environment, and this is why EMIL-S makes a difference between “environmental events” and “normative events”.. In all other respects, EMIL-S is more or less a one-to-one implementation of EMIL-A.

10.4 Additional Assumptions about Cognitive Processes Used in EMIL-S

As mentioned in previous sections, cognitive processes in agents are not restricted to normative ones. Beside social inputs they have to process environmental inputs as well. In some of the scenarios described in Chapter 13 these environmental inputs play a minor role, in others they play a major role, and in some of these cases it is only environmental inputs that first have to be interpreted as social inputs. This is because norm invocations addressed to a group member, for instance in the micro finance scenario, first have to be agreed upon by the other members of the group. When social inputs do not come from an individual but from a group this can be understood, modelled and implemented as a message sent by somebody like the speaker of the group, but it can also be understood, modelled and implemented as a the minutes of a meeting of the other members of the group in which a decision referring to the behaviour of the absent fallible member was passed. In both cases the norm invocation will be ascribed to one or more other agents by the recipient, but only by way of interpreting a text or a symbol (such as a traffic sign or some other passive object – trespass board, no-smoking sign etc.).

All these social and environmental inputs have to go through the same cognitive process, including those which do not qualify as normative (and which, according to EMIL-A, immediately leave the normative process through the “no norm exit”, see Chapter 9, Figure 23, p. 78). The following example may show that that the cognitive process dealing with non-normative environmental input is more or less the same as the cognitive process dealing with a norm invocation. In a traffic scenario, a traffic sign announcing a dead-end street can have an effect on the recipient agent’s behaviour (namely not to enter this street just because it is useless in the current situation) although it is neither forbidden nor commanded nor allowed nor recommended. Processing the dead-end street sign would be more or less the same as processing norm invocations — except that it would not make any changes in the agent’s rule base: If using the dead-end street conforms to the non-normative goal of exploring what is behind the dead end of the street, the decision would be to use this street; if the goal is to reach some other place, the decision would be not to use the dead-end street but to search for other routes leading to the goal of the walk.

Thus EMIL-S will have to go beyond EMIL-A in so far as it will use the same mechanisms both for normative and non-normative decision making wherever the two are similar enough.

EMIL-A says next to nothing about the process of norm learning. Normative actions can change probabilities in the norm recogniser (Campennì, 2007) but there is no precise description of how this works. EMIL-S introduces additional assumptions about this process (see Chapter 11). If an EMIL-S agent receives a norm invocation which matches one of the actions it has taken before, it has to analyse its valuation history and to check and adapt the probabilities of taking an action from the group to which the action belongs which was an object of the norm invocation. How this works might be different between scenarios.

Chapter 11 Supporting the Theory: Formal Description of EMIL-S

Ulf Lotzmann, Michael Möhring

Abstract

This section describes the design of an agent model for simulating normative behaviour and norm formation processes. The model is based on a theory of norm innovation as laid down in Chapter 3. The main focus of this section is the conversion of the theoretical framework towards a software design. In particular, this includes the formal description of the static structures of the simulation environment and EMIL-S agents as well as the dynamic processes of agents, in their different roles as actors or observers.

The formal description of static structures covers

- the message concept,
- agent-specific event boards and normative frames, and
- a normative board, describing general norms valid in a concrete scenario.

These structures hold the data for which a number of intra-agent processes are specified (formally by UML-based activity diagrams):

- normative behaviour: the reaction on events within the environment (actor role) or the valuation of actions from other agents (observer role);
- the recognition of dependencies between (perceived) event-action-event-sequences and behavioural regularities, and
- the recognition of norms within these regularities.

11.1 Theoretical Background

Beyond the requirement derived from EMIL-A and discussed in the previous section, other – in the sense of computer science more technical – requirements have to be regarded. Firstly, the intra-agent process design must allow the handling of complex and adaptive rules. Secondly, the software design should be modularized in a way that general parts of the process are separated from scenarios dependent parts. These two kinds of requirements are essential for all architectural aspects of EMIL-S agents.

The EMIL-S architecture is influenced by widely accepted achievements from software engineering and other disciplines. The following list only gives a short (and incomplete) overview:

- architectures of AI software systems (e.g. rule engines and knowledge bases) and normative and cognitive agents (e.g. BDI or BOID, cf. Boella et al., 2007; Bratman, 1987; Broersen et al., 2001; Neumann, 2008);
- adaptive rules and learning algorithms (cf. Lorscheid and Troitzsch, 2009);
- simulation models (e.g. modelling human needs for market simulations and Wikipedia model, cf. Norris and Jager, 2004; Troitzsch, 2008);
- simulation tools (e.g. Repast, cf. North et al., 2006, 2007).

The final architecture is a composition of aspects from these topics, enriched by novel aspects which are worked out in the subsequent chapters.

11.2 Architecture of a Normative Agent

It is common sense that each classical intra-agent process consists of three basic steps (as with every other data handling, covering input, processing and output):

- At some point of time an agent as an autonomous entity must check the state of the environment in which it is situated. This is usually done within a perception process. This process changes the agent-internal model of the environment.
- Due to the changed environmental state and due to the agent’s internal state, a decision about

measures to achieve some individual goals of the agent must be drawn. This is done within a decision process, in many cases based on some sort of rule engines.

- The decision leads to actions directed to the environment with the result of environmental changes.

These steps are shown in Figure 26, together with two additional steps which are essential for adaptive behaviour:

- The action that was performed can change the environment with certain intensity in either a positive or a negative way. Thus, the impact of the action must be evaluated in order to show modified (and preferably better in respect of goal achievement) behaviour at the next appearance of a similar environmental state. This process step is called valuation.
- The result of the evaluation from the previous step must be translated into an appropriate rule change, i.e. the actual rules must be **adapted** in some way.

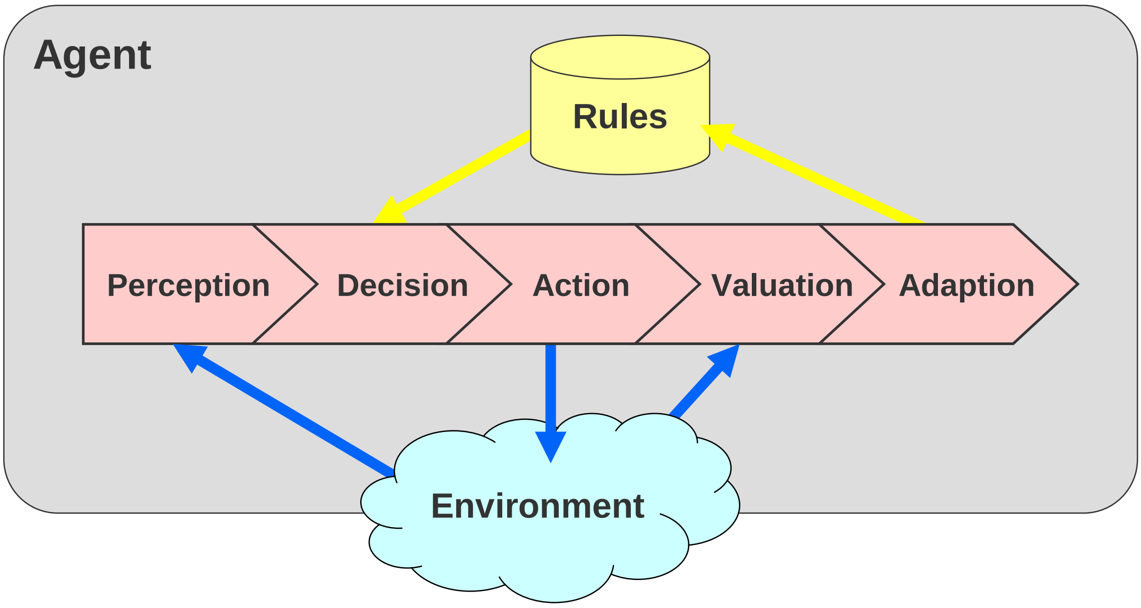


Figure 26. Generalized intra-agent process

The separation into these five process steps gives way to another separation according to the foci of the steps. While the three steps "perception", "action" and "valuation" are connected to environmental affairs, the two other steps "decision" and "adaption" are mainly dedicated to rule base access. This allows the definition of two categories of steps, expressed by two layers:

- A "physical layer", encapsulating all environmental properties, can be seen as a model counterpart to a (human) body. The representation of the body differs from simulation model to simulation model. For example, the physical layer of a traffic participant includes attributes like shape, dimension, orientation, velocity and, furthermore, must feature a special perception sensor able to perceive other agents representing traffic participants as well as topographic elements from the static environment (e.g. a road network). For other models no such complex physical representation is necessary. E.g. for the simulation of agents contributing to a Wikipedia the environment consists of a shared workspace, the physical abilities can be reduced to writing, searching, reading of articles and commenting on them – no aspects of a "real" physical body are relevant.
- On the other hand, the rule decision and adaption process can be abstracted from the environment and coalesced in a "strategic layer". This induces that within the strategic layer a common rule definition language must be established which is used for any kind of simulation scenario.

The advantage of an approach like this is evident: a simulation tool can be realized that completely covers the strategic layer and can be attached (via some sort of interface) as an add-on to other existing simulation tools or programs. Furthermore, due to the rule engine functionality of the strategic layer, the specification of the strategic model aspects can be done at a higher level of abstraction – no "programming" in computer science style is necessary in this respect.

To allow a high level of abstraction, a way must be found to raise the interactions between agents also on an abstract level. On the physical layer a lot of interaction may happen which is not relevant for strategic decisions. On the other hand, all relevant happenings that may occur within the environment also must find their abstract representations. For this purpose a concept based on **events** and **actions** is introduced. While events describe all incidents that require attention on the strategic layer, actions express all possible outcomes from the strategic layer. Two different types of events and actions[51] must be introduced:

- so-called environmental events and actions, directed from or to the physical layer, respectively, and
- so-called valuation events and actions. Origin of a valuation is the "measurement" of the "success" a performed action has had achieved within the environment. For normative simulations according to EMIL-A another source of valuations comes into play: an observing agent valuates (by the act of sending a valuation) another acting agent (which receives this valuation as an event) for an environmental act. This type of valuation is called norm invocation.

The involvement of agents capable to observe and valuate other agents is not only one of the key properties of the EMIL-A framework but also a crucial design element of the agent architecture and, moreover, one of the major innovations of simulator design. For this purpose an agent must be able to cover both the (classical) "**actor**" as well as the (novel) "**observer**" roles. This new observer role has some concrete implications, both for intra-agent and for inter-agent processes:

- for inter-agent matters a communication infrastructure must allow to observe ("listen" to) perceptions and actions of observed agents;
- the agent must be equipped with capabilities to generate a model of an observed agent;
- a suitable rule set of the observed agent must be available also for the observer.

While all environmental (and partly valuation) interactions between agents are by definition based on the physical layer, the norm-invocation interactions are situated only within the strategic layer. This kind of interaction, together with a shared "statute book" holding all regular norms that have been either predefined in the scenario or have emerged during the simulation[52], are the key properties of the EMIL-A framework. The two-layer architecture allows to fully integrating these elements within the strategic layer, hence all aspects of the normative process can be made independent from the concrete scenario realization.

In the following the intra-agent process for the actor role is demonstrated for the first simulation cycle. For this example it is of course necessary to include the inter-agent perspective (transmission of events and actions between the corresponding entities) which is not described so far. The message concept used in EMIL-S for this purpose was already proposed by EMIL-A, and is described in detail in the following chapter.

Figure 27 shows the steps of the decision process that is initiated by an event (E2), which had occurred at the environment and was perceived and by the physical layer. At step 2, only the initial rule base is inspected because the normative frame (as preferred source of rules) is empty at simulation start (time t_0).

[51] Perhaps there is no sharp distinction. But the utterance "I know that smoking is allowed here, and generally I have no objection against people smoking in my presence, but today I have a bad cold and would like to ask you to abstain from smoking!" is certainly not a norm invocation, but — in terms of EMIL-S — an environmental action.

[52] This does not imply that all agents necessarily have the same norms. What is allowed in a statute book might still be forbidden from the point of view of some agents.

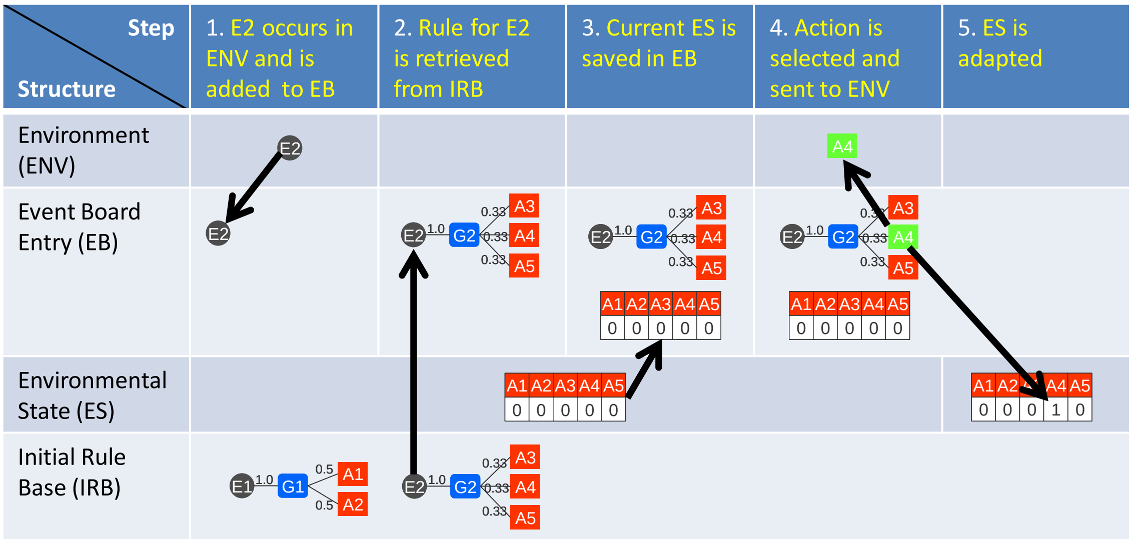


Figure 27. Intra-agent decision process for time t_0

During the following adaption process, triggered by a norm invocation, the first normative frame entry is created (Figure 28).

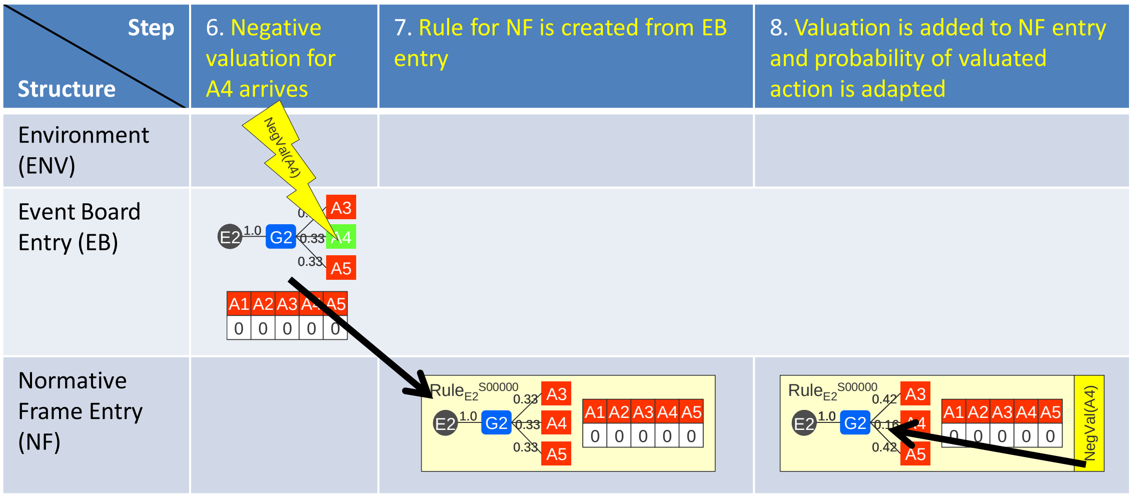

Step
Structure
6. Negative valuation for A4 arrives
7. Rule for NF is created from EB entry
8. Valuation is added to NF entry and probability of valuated action is adapted
Environment (ENV)
Event Board Entry (EB)
Normative Frame Entry (NF)
NegVal(A4)
Rule$_{E2}^{S00000}$
E2
G2
A1
A2
A3
A4
A5
1.0
0.33
0.42
0.16

Figure 28. Intra-agent adaption process for time t_0

11.3 Design of a Normative Agent

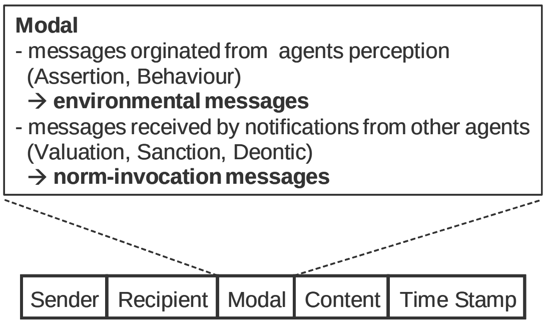
11.3.1 Concepts

Based on the theoretical background of agent-based approaches on one hand and principles of norm innovation dynamics on the other: How can an agent be equipped with capabilities that allows norm-oriented behaviour and establishing, perceiving and, extending norms? In this chapter a possible answer to this question is presented, realizing the theoretical concepts proposed in EMIL-A and putting the general architecture of the EMIL-S simulator, which is introduced in the previous chapter, in concrete terms.

As usual in agent-based simulations, the communication between agents is based on a concept of messages, which trigger the processing of events agents perceive within the environment in which they are situated and which they influence by corresponding actions. For example, in a traffic simulation scenario a car driver perceives a pedestrian on one side of the road and reacts by slowing down the car. This event motivates the pedestrian to begin with crossing the road, which, in turn, makes the car driver stop the car. These events, originating from an agent's perception, are called **environmental events** in EMIL-S.

The modelling of agent behaviour based on societal regulations (or norms) and the generation of this kind of regulations (e.g. by learning processes) obviously requires some fundamental extensions of the approach described so far:

First of all, the introduction of an additional category of events is necessary, which allows the **assessment of (environmental) events** and corresponding actions, performed in a concrete application scenario. For example, it should be possible to evaluate the behaviour of another agent (e.g. admonish a car driver for jumping a red light) by positive/negative valuations or sanctions, or even advising an agent what to do under a particular condition by "deontic" assertions (e.g. "Do not cross the street until the traffic light switches to green!"). These events are called **norm-invocation events**. An appropriate message structure chosen for EMIL-S can be seen in Figure 29.



Modal
- messages orginated from agents perception (Assertion, Behaviour)
  → **environmental messages**
- messages received by notifications from other agents (Valuation, Sanction, Deontic)
  → **norm-invocation messages**

Sender	Recipient	Modal	Content	Time Stamp

Figure 29. Message structure

Secondly, these so-called **norm invocation events** should not only be activated by agents directly involved in a concrete situation, but also by agents who observe this situation (e.g. watching an accident). Therefore, EMIL-S distinguishes between two agent roles: **actor** and **observer**.

With reference to the two layer concept presented in the previous chapter, a number of different occasions for message exchanges between various entities – beyond the distinction into environmental and norm-invocation messages – are defined.

A further distinction into two types of environmental messages is made as follows:

- a perception is sent from the physical layer to the EMIL-S layer via an environmental message with modal A:

$$PHYS(Agent_x) \xrightarrow{ENV(A)} EMIL(Agent_x)$$

- the command to perform an action is sent from EMIL-S to the physical layer via an environmental message with modal B:

$$EMIL(Agent_x) \xrightarrow{ENV(B)} PHYS(Agent_x)$$

Furthermore, there are three possible ways to exchange norm-invocation messages (valid for any modal):

- an implicit norm invocation is sent independently from the EMIL-S layer of an observer agent x to the EMIL-S layer of an observed agent y:

$$EMIL(Agent_x) \xrightarrow{NI(D,V,S)} EMIL(Agent_y)$$

- an explicit norm invocation is defined within an special action and sent from the EMIL-S layer of agent x to the EMIL-S layer of another agent y after the action was triggered by an environmental message:

$$PHYS(Agent_x) \xrightarrow{ENV(A)} EMIL(Agent_x) \xrightarrow{NI(D,V,S)} EMIL(Agent_y)$$

- an explicit norm invocation is directly generated within the physical layer of agent x and sent to the EMIL-S layer of another agent y:

$$PHYS(Agent_x) \xrightarrow{NI(D,V,S)} EMIL(Agent_y)$$

Using the described environmental and norm-invocation events by actors and observers, the **learning capabilities** in EMIL-S (to form a normative belief into the agents' minds) can be described as follows:

- **Reinforcement**: learning from an agent's own experience (e.g. a pedestrian has a near-collision with a car because of not using the striped area for crossing a street)
- **Imitation**: learning by observing other agents' experience (e.g. observing a near-collision between a pedestrian and a car because this pedestrian did not use the striped area for crossing a street)
- **Normative learning**: listening to other agents' reports of their experiences (e.g. "You should use the striped area for crossing a street!")

11.3.2 Basic Structures

Agents

The above mentioned intra-agent processes require at least two kinds of agent-internal memories:

- **Event boards** memorizing the history of incoming environmental events including the conducted actions are required. For this purpose each agent has an event board for its own perceptions as well as additional event boards for each observed agent.
- **Normative frame** holding preliminary norms, derived from the experiences logged in the event board during the simulation.

Finally, each agent needs an initial set of rules (**Initial Rule Base, IRB**), which contains rules describing basic behavioural elements, and thus constituting the seeds for more complex rules emerging from the simulation process.

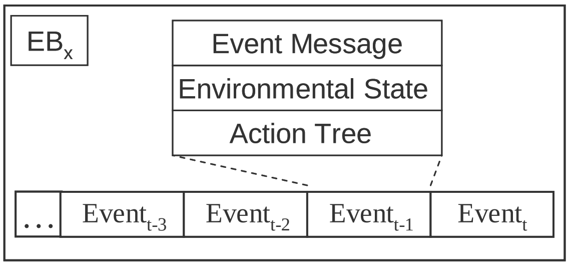


Figure 30. Event board

The event board is a chronologically sorted sequence whose elements contain the following data (Figure 30):

- the environmental message;
- the current (environmental) agent state (e.g. velocity and perception range for a car driver in the traffic scenario);
- the associated action tree (see Figure 32) with the individual selection probability function.

For each subsequence of the event board a so-called Classifier (CLA) can be generated. It allows comparisons between event board subsequences and normative frame entries in the norm formation process later on.

Each event board sequence describes a consecutive fragment of agent behaviour, thus introducing a higher level of complexity. It must be assumed that only within this complexity level, regularities and in particular norms are residing.

An entry of the normative frame, which can be given in advance by the modeller, or which arises from the detection of regularities in event board sequences, contains the following elements (Figure 31):

- the associated classifier of the event board sequence;
- the events from the corresponding event board sequence;
- a generated rule (a merged action tree);
- a valuation history, holding a statistical report of the valuations received on the respective rule.

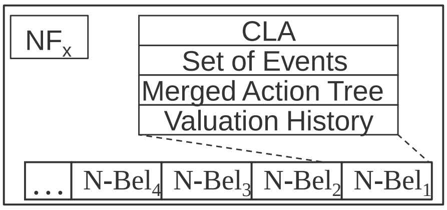


Figure 31. Normative frame

Finally, each agent must be equipped with a set of initial rules (IRB), which allows it to act in the simulation environment. Rules in EMIL-S are represented as so-called **event-action trees**, a kind of decision trees that represent the dependencies between events and actions. For each event an arbitrary number of action groups are defined. Each action group represents a number of mutually exclusive actions. The edges of the tree are attributed with selection probabilities for the respective action groups or actions (Figure 32).

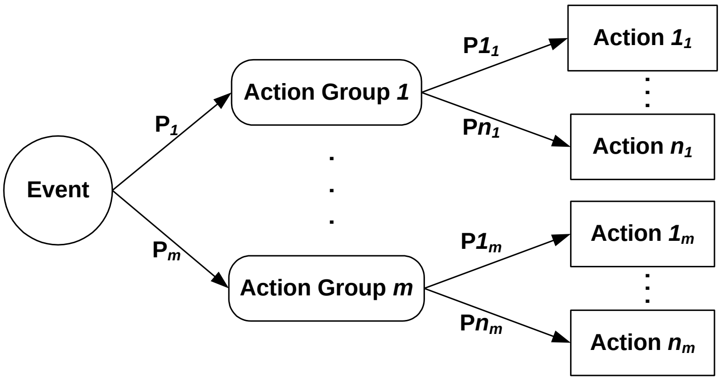


Figure 32. Event Action Tree

For each action group $x \in [1..m]$, $P_x \in [0..1]$; for actions $y \in [1..n]$ of action group x is $\sum Py_x=1$

System Environment

On the system level an additional data structure is necessary, which contains regular norms, valid for the complete model. Again, this can be given in advance by the modeller or derived by the evaluation of preliminary norms from the agent's normative frames. Consequently, an entry of the normative frame (Figure 33) consists of the same elements as the normative frames, except the validation history, which is used to decide if a preliminary norm will become a norm or not.

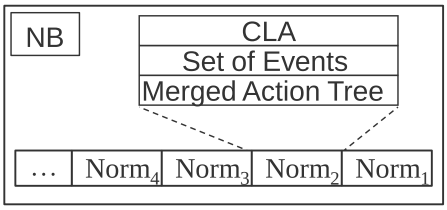


Figure 33. Normative board

11.3.3 Agent Behaviour

Basically, the agent behaviour is triggered by discrete events packed into incoming messages. The (UML-based) activity diagram in Figure 34 shows the main loop of the processing of incoming messages including the pre-processing steps which are necessary for handling environmental (ENV) and norm-invocation (NI) events later on.

Thus, the first activity **"determine role"** of this process is dedicated to the distinction of the role of the message receiver **x** (actor or observer), stored temporarily in the state object R(x).[53]

Secondly, based on the role information (x) and on the content (i) of the message field "Recipient", the relevant event board of the message receiver has to be selected (**"select actual event board"**), either the "actor" event board, or one of the event boards of observed actors, and stored in EB(x, i).

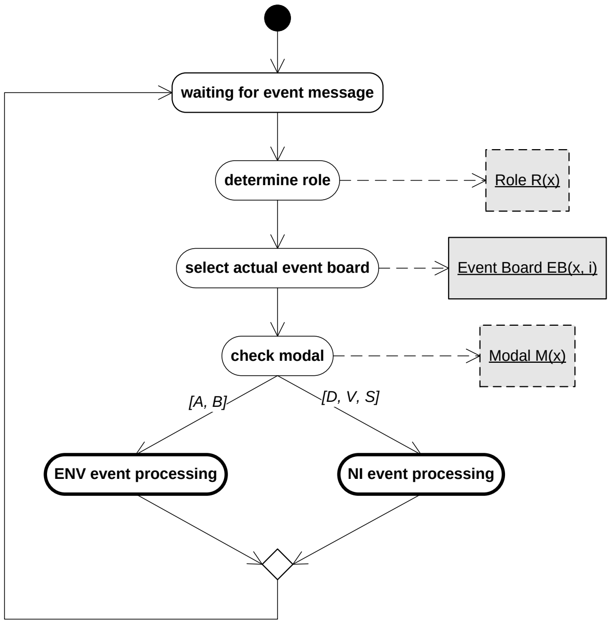


Figure 34. EMIL-S event processing

Finally, the identification of the incoming message type is done by evaluating the modal of the message ("**check modal**").

Modal values of A (=Assertion) and B (=Behaviour) identify "environmental" (ENV) events, whereas modal values of D(=Deontic), V(=Valuation), and S(=Sanction) characterize "norm invocation" (NI) events. Again this information is stored temporarily in M(x).

Environmental Event Processing

Figure 35 shows the processing of incoming environmental events for agent x.

An incoming environmental message triggers this process. The message originates either from the perception of an event within the agent's environment (a so-called assertion), or is the capture of an assertion or behaviour message from an observed agent.

The first activity **"modify event board"** stores information about the incoming event into agent x's event board EB(x, i) for agent i. This means that EB(x, x) is the "own" actor role event board, while all other event boards are for observed agents. If the current modal is "Assertion" (M(x)=A) a new event board entry is generated, whereas the action field of an already existing event board entry is completed if the modal is "Behaviour" (M(x)=B).

[53] See the footnote 51 for the distinction between norm invocation and environmental action.

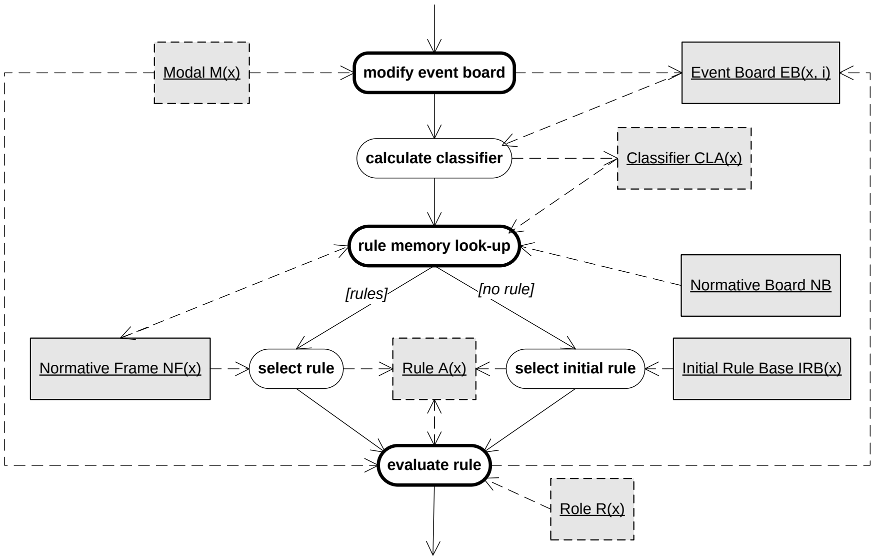


Figure 35. Environmental (ENV) event processing

The recently created or modified entry is the base for the **"calculate classifier"** activity This classifier represents the event board subsequence of a certain length (determined by a model parameter), with this entry as most recent element. The actual classifier is then saved in the temporary state object CLA(x).

CLA(x) is the key for the following **"rule memory look-up"** activity. This complex activity searches for an already existing rule for the sequence of currently recorded events in both of the long-term memories NB and NF(x). Within this activity two steps are processed:

1. The common normative board NB is examined for norms valid according to the classifier. If norms are found, they are copied into agent x's normative frame NF(x) after a decision process which reflects the agent's disposition on abiding by norms.
2. The normative frame NF(x) is searched for matching entries which can be preliminary norms or norms copied from the normative board before.

If one or more entries are found, one of them is finally selected within the activity **"select rule"** and stored into the state object A(x). This procedure is important to recognise typical and already known (complex) situations within the environment at an early stage to allow an adequate and timely reaction (e.g. to avoid undesired incidents).

In case that no normative frame entry is found, the event-action-tree that belongs to the incoming event is fetched from the initial rule base (IRB) within the activity **"select initial rule"**.

In both cases, a rule (represented by an action tree) is available as input for the activity **"evaluate rule"**. The effect of this complex activity is furthermore determined by modal M(x) and role R(x). The following modes of operation are specified:

- R(x)=ACTOR: The rule is evaluated for selecting and executing appropriate actions (by sending a message with modal B), and therefore determining the agent's behaviour. Afterwards the modified rule (in terms of reinforcing the just selected actions) is stored in the event board entry that was created at the beginning of the process.
- R(x)=OBSERVER and M(x)=A: A deontic (i.e. a norm-invocation message with modal D) is sent to the observed agent, expressing which actions would be executed by the observer agent according to its own rule.
- R(x)=OBSERVER and M(x)=B: A valuation or sanction (i.e. a norm-invocation message with modal V or S) is sent to the observed agent, blaming or praising the action which the observed agent has executed. Type and strength of the norm invocation is determined by comparing the observed actions with the rule of the observing agent.

Based on the sketched process the agent's norm oriented behaviour is implemented.

Norm-invocation Event Processing

Incoming norm invocation events are handled by the process specified in Figure 36.

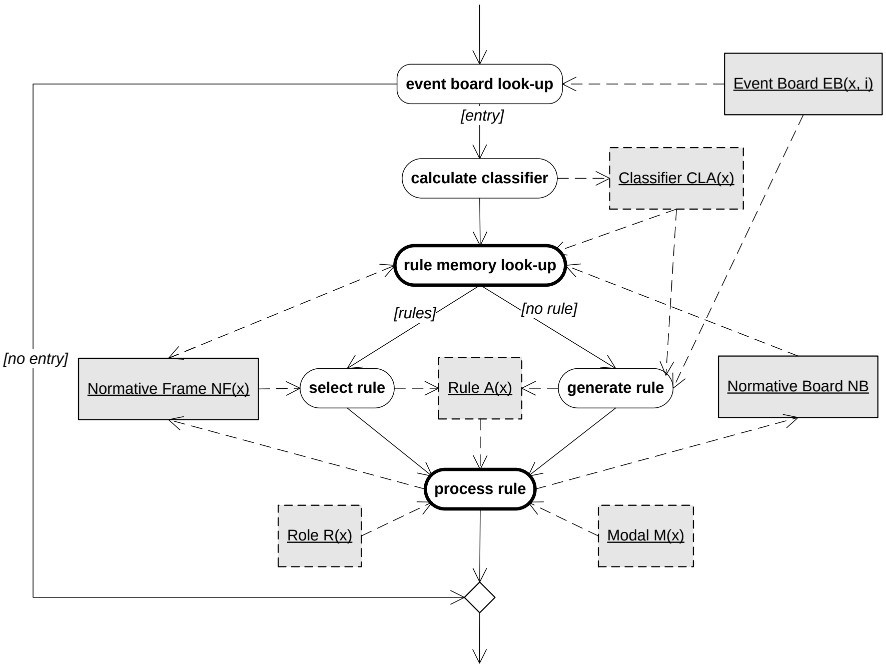


Figure 36. Norm-invocation (NI) event processing

This process is triggered by the reception of messages that contain either valuations or sanctions with respect to already executed actions, or deontics disclosing information about what to do in environmental situations already experienced. In both cases, the norm invocation refers to data stored within the event board. Thus, the first activity implements an **"event board look-up"** which usually returns the valuated event entry. If no entry is found, then the valuation is invalid (e.g. the valuated action is outdated and already removed from the event board), and the processing is aborted. On the other hand, if an entry is found it becomes the most recent event of an event board subsequence for which a new classifier CLA(x) is generated within the activity **"calculate classifier"**. This activity as well as the following activity **"rule memory look-up"** is identical with the equally labelled activities of the environmental event process introduced in the previous section.

Again, the result of the look-up process is either

- a set of rules, from which the entry with the highest similarity related to the classifier is selected and stored in A(x) by "select rule", or
- the information that there is no similar rule found. In this case, a new rule is generated by merging the rules stored in the event board subsequence for which the classifier has been calculated (**"generate rule")**.

This (either new or already existing) rule then undergoes a complex sub-process within the **"process rule"** activity, parameterised by state objects R(x) and M(x). Basically, this activity covers the following steps:

- According to R(x) and M(x), the probabilities within the rule A(x) are modified in a way that the execution of positive valuated (or sanctioned) actions will be more likely in the future, and vice-versa.
- The current norm invocation is added to the valuation history.
- The valuation history is inspected in order to decide whether the preliminary norm can be transformed into a regular norm. This decision algorithm considers (a) from how many different agents valuations are stored in the valuation history, and (b) the change rates of the probabilities

attached to the action tree during the recent time period. If the agent decides to transform the preliminary norm into a regular norm, the content of the normative frame entry is proposed to the normative board NB.

This brief overview should nonetheless give an impression of the norm formation process implemented in EMIL-S.

11.4 Process Refinements

11.4.1 Modify Event Board

The first activity of the environmental event process of agent x is the modify event board sub-process (Figure 37). Input data for this activity is the incoming (already pre-processed) message. Firstly the value of the modal state object M(x) is checked. If the agent is in actor role, the only allowed value is "A", this means that a perceived event from the environment has to be treated. In the case that the agent is in actor role, the modal value can be either

- "A" when agent x observes, that agent i has made an environmental perception, or
- "B" when agent x observes an action performed by agent i.

If the modal state is "A" an event board entry is created and afterwards stored into the previously selected event board of agent x.

Otherwise, if the modal state is "B", agent x searches the event entry within the event board (of the observed agent) for which the currently received action fits. The action is then written to the entry.

Output data for this activity is a reference to the just created or modified event board entry.

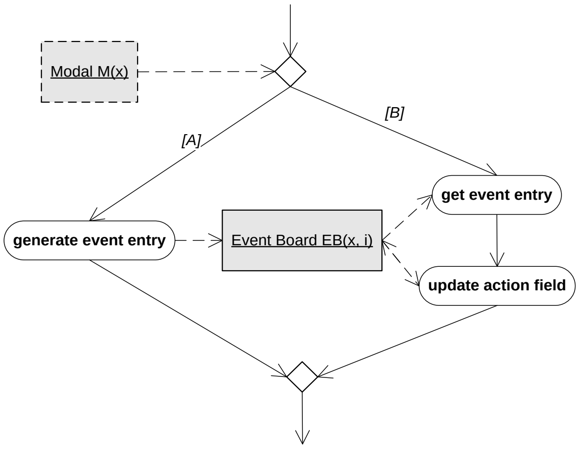


Figure 37. Modify Event Board

11.4.2 Rule Memory Lookup

The rule memory lookup sub-process (Figure 38) is a central activity of both the environmental and the norm invocation event processing. The mode of operation is very similar for both process types and mainly manages the access to the individual normative memory (normative frame) of agent x as well as to the shared normative memory (normative board). The input data is the previously calculated classifier CLA(x), which serves as a "key" for both data structures.

The process firstly examines the normative frame. When a matching entry is found, it is copied into the normative frame of agent x, possibly overwriting existing entries with similar CLAs.

In any case the normative frame of agent x is inspected afterwards, using the same CLA. This procedure enables the agent to decide, whether to use the possibly valid norm, or to rely on his own beliefs. The sub-process then returns the matching rule (or more than one rule, if the result of the rule fitness function is ambiguous).

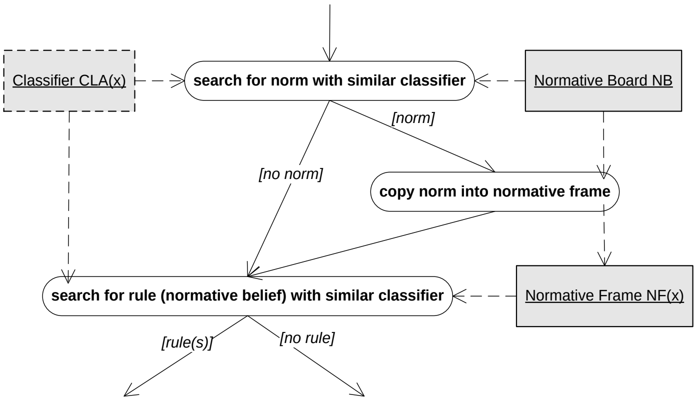

Classifier CLA(x)
search for norm with similar classifier
Normative Board NB
[norm]
[no norm]
copy norm into normative frame
search for rule (normative belief) with similar classifier
Normative Frame NF(x)
[rule(s)]
[no rule]

Figure 38. Rule Memory Lookup

The most challenging feature of this sub-process is the algorithm for calculating the fitness of a norm or normative belief, respectively, to the classifier. This is a parameterized function that has to be adapted to

- the scope of the classifier,
- the structure of the initial rules, and
- the concrete simulation scenario.

The consequence of improperly selected parameter settings may lead either to "inflation" of normative beliefs (if granularity of the CLA is too fine) which results to divergent behaviour, or to an insufficient number of different rules, leading to a too fast convergence of rules, respectively.

11.4.3 Evaluate Rule

The "evaluate rule" sub-process (Figure 39) is part of the environmental event process of agent x. Although the activity is dedicated to handling a rule selected in the preceding process steps (and which is the input data of the activity), the impact of this activity depends on the state object role R(x) and is fundamentally different for the actor and observer role.

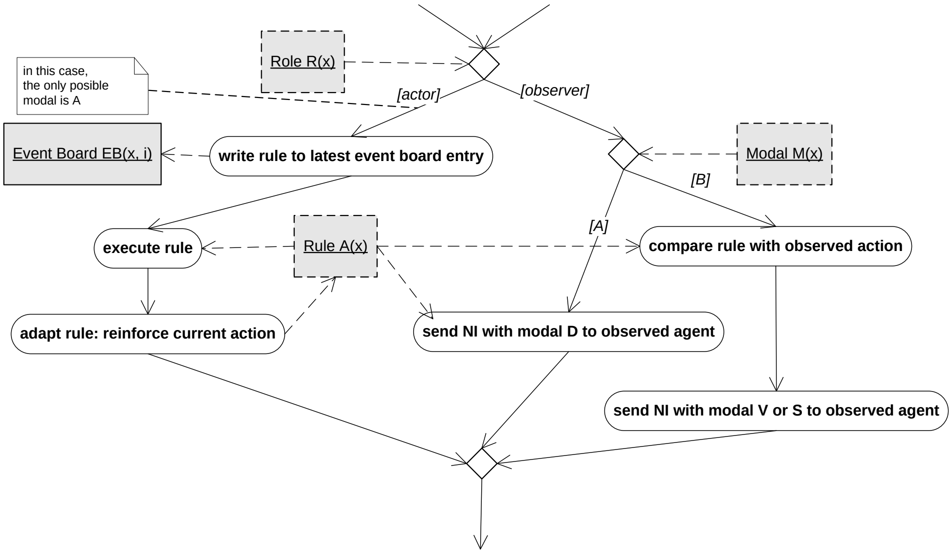

in this case, the only posible modal is A
Role R(x)
[actor]
[observer]
Event Board EB(x, i)
write rule to latest event board entry
Modal M(x)
[B]
[A]
execute rule
Rule A(x)
compare rule with observed action
adapt rule: reinforce current action
send NI with modal D to observed agent
send NI with modal V or S to observed agent

Figure 39. Evaluate Rule

For the actor role case, only events with the modal "A" have to be regarded. In the first activity the rule A(x) chosen before is written into the event board entry created previously, thus completing this data structure. Afterward this rule is executed by triggering one or more actions in a stochastic selection process. The following "adapt rule" activity might change the probability setting of Rule A(x). Thus, this activity covers the reinforcement learning strategy of the agent and is parameterized by external attributes.

In the other case that agent x is in observer role an additional status object comes into play, the modal M(x) of the currently arrived message. Here both modals “A” and “B” for environmental matters are allowed:

- If a behaviour (i.e. an action performed by the observed agent – with modal “B”) was observed, the observing agent x examines the own just selected rule in order to figure out the (own) probability for the observed action. According to the probability value an implicit norm invocation with an adequate characteristic is sent afterwards: in the case of equal probabilities the norm invocation will be a positive valuation (or sanction), in the case of significant differences rather a negative valuation or sanction. If required for the simulation scenario (and expressed by respective simulation parameters), the action probability can also be modified at this point in order to imitate the observed behaviour. Thus, this activity additionally covers the “imitation” agent learning mechanisms.
- Alternatively, the incidence of an environmental event was observed, i.e. the observer listens to a message with modal A, directed to the observed agent x. In this case one or more actions are chosen from the selected rule in a similar way as it is done in the actor role path of this activity, but instead of executing the actions, norm invocation messages with the modal “D” are created and sent to the observed agent (thus expressing the actions the observer would perform in the situation the observed agent is situated).

11.4.4 Process Rule

The “process rule” activity (Figure 40) of the norm-invocation event processing is in some respect a counterpart to the “evaluate rule” activity in the environmental process. Input data for this activity is again a rule selected within the preceding process, but the focus here is the adaption of the individual agent behaviour for future occurrences of environmental events. Thus, this activity covers the normative agent learning strategy.

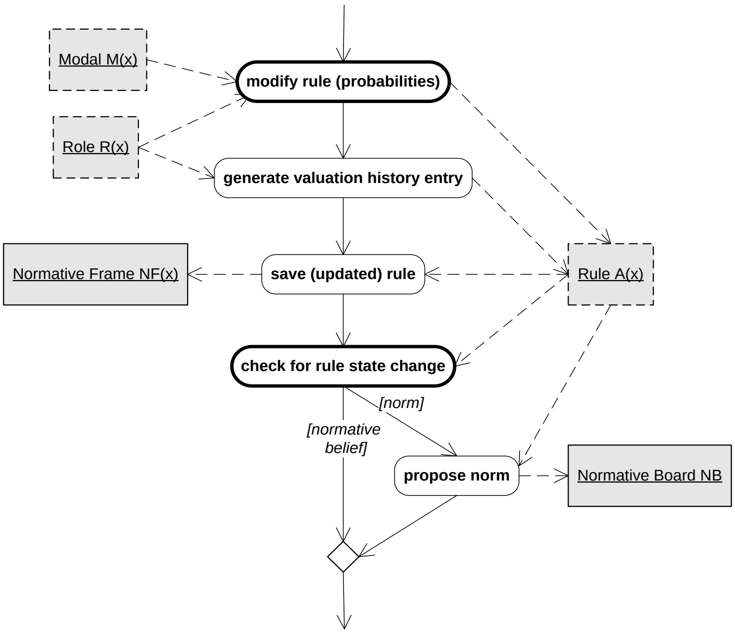


Figure 40. Process Rule

Since the agent behaviour is pre-defined by stochastic rules, the behaviour adaption is achieved first and foremost by adjusting the probability values for certain actions. This is done within the (complex and parameterized) sub-activity “modify rule (probabilities)” which is explained in detail below. Additionally, the message is appended to the valuation history of the addressed normative frame entry within the activity “generate valuation history entry”. The updated entry is then resaved in the normative frame.

As proposed in EMIL-A, the agent behaviour is additionally influenced by the process of norm immergence, implemented in the “rule memory look-up” activity described above. The norms that can immerge into the agents must, on the other hand, come into existence within a correspondent process of norm emergence, which is another subject of the "process rule" activity.

According to EMIL-A the emergence of a norm can be realized by an individual assessment process. In the actual design the basic elements of norms are the normative frame entries. Each time a valuation (in form of a norm invocation message) arrives, an activity "check for rule state change" is performed. Purpose of this complex sub-activity – which is more specifically detailed below – is the decision whether an individual rule/normative belief should be proposed as a norm candidate to the normative board and, thus, to the agent community.

Modify Rule Probabilities

The way the rule properties are modified is essential for the normative learning characteristics. Subject matter of this activity is the calculation of the change rate for the probability of an action, considering type and strength of a norm invocation as parameters. It seems conjecturable that there won't be an algorithm that perfectly satisfies the requirements of all conceivable simulation scenarios that could in principle be implemented with EMIL-S. In fact it will be necessary especially for complex simulation models to find the best fitting algorithm. For this reason the software design provides an encapsulation of this particular algorithm, which allows for a quick and easy replacement and even makes of a "stock" of different predefined learning algorithm modules possible.

The EMIL-S implementations of the scenarios described in this report rely on a basic algorithm that does not distinguish between valuation and sanction modals, but that is parameterized by the strength and "direction" (positive v/s negative effect) of norm invocations (for these simple models it should be admissible to define a valuation as a sanction with a small strength value). Figure 41 shows the pseudocode specification of this algorithm.

Of course this algorithm must furthermore ensure the consistency of the related event action tree (by proper adjustment of the probabilities for all other actions included in the tree).

```
factor = 0.5 // learning rate parameter
changeRate = abs(strengthOfSanction) * factor
if strengthOfSanction >= 0
    then
        incr = (1 – oldProbability) * changeRate
        newProbability = oldProbability + incr
    else
        decr = oldProbability * changeRate
        newProbability = oldProbability - decr

endif
```

Figure 41. Pseudo code of "modify rule probabilities" algorithm

Check for Rule State Change

The algorithm behind this activity classifies a rule (the actual normative frame entry) as a normative belief or as a (regular) norm candidate. Input of the algorithm is the valuation history attached to each normative frame entry.

The valuation history is a collection of all norm invocations received for this rule, stored in a chronological order. For each norm invocation the respective sender is also saved as it has to be regarded by the algorithm.

Similar to the modify rule probabilities activity, the algorithm for this activity has a crucial impact for the normative process, and it is also very likely that further research in combination with more complex simulation scenarios will yield more sophisticated techniques in this matter. The pseudocode of the basic algorithm used with the scenarios at hand is shown in Figure 42.

```
valEntry = get latest valuation entry
while valEntry is not empty
    loop
        add valEntry.sender → senderlist
        add valEntry.purpose_of_valuation → purposelist
        if (senderlist contains x percent of agent population)
            then
                if (purposelist is composed of y percent
                        of similar valuations)
                    then
                        return rule_state = “norm”
                    else
                        return rule_state = “normative belief”
                    endif
            endif
        valEntry = get next older valuation entry
    end loop
return rule_state = “normative belief”
```

Figure 42. Pseudo code of “check for rule state change” algorithm

11.5 Technical Aspects

The norm formation process described in the previous chapter is basis for the agent implementation in the EMIL-S simulator. EMIL-S is not realized as a stand-alone software, but rather as an extension of existing simulation scenarios, e.g. based on REPAST (North et al., 2006) or TRASS (Lotzmann and Möhring, 2008; Lotzmann, 2008). Thus, the EMIL-S software must provide several interfaces:

- an interfaces to the physical layer of concrete simulation scenarios, consisting of (JAVA) interface definitions and static controller classes;
- an agent design interface to support the modelling of agent definitions at the strategic layer by a graphical user interface, and which allows additionally the tracing of simulation runs.

The decision to distinguish between “physical” and “strategic” layers of simulation models and to implement only the latter one into EMIL-S, yields further implications of a more technical nature as already described in the “architecture of a normative agent” chapter.

- EMIL-S is designed as a reactive system for discrete events and is otherwise completely independent from the physical agent representation within the underlying simulation tool – it makes no difference whether a discrete event simulation model is used as well, or some other approach for time representation is present (e.g. round based models). Any other agent property (e.g. the triggering of proactive actions) remains part of the physical agent layer.
- EMIL-S does not include any simulation control or scheduling mechanisms – these will always be matter of the underlying simulation tools. In contrast, EMIL-S brings a communication infrastructure (only) for normative messages that can be used optionally to keep the physical layer completely free from all normative concerns.
- The modeller must be aware that simulation models involving EMIL-S always consist of two parts (or configurations, respectively) which are deeply related to each other.

- Due to the fact that simulation data arises from two different layers a higher effort is needed to collect and process the simulation data.

How these properties influence the model design in terms of concrete implementations will be explained in the scenario section of this report.

Chapter 12 Supporting Theory Use: The MEME Simulation Methodology and Tools

László Gulyás and Attila Szabó

Abstract

Models of complex social systems typically depend on a number of assumptions, quantified in the form of specific values to certain model parameters. Ideally, any such model should be tested with any meaningful combination of these parameters, in order to determine the validity of the model. Such experiments also allow for the evaluation of possible alternatives for the crucial model components.

In this chapter, we start with an overview the problem of efficiently executing and analyzing large computational simulations. Then we discuss potential solutions, like the options of distributed execution and non-naïve approaches for sampling the parameter space. This is followed by a detailed discussion of the Model Exploration Module (MEME), a tool developed for addressing the above issues.

12.1 Introduction

Models of complex social systems typically depend on a number of assumptions, quantified in the form of specific values to certain model parameters. Ideally, any such model should be tested with any meaningful combination of these parameters, in order to determine the validity of the model. Such experiments also allow for the evaluation of possible alternatives for the crucial model components.

In addition to the dimensionality and the size of the parameter space, the *sensitivity analysis* of complex system models has to face the additional challenge of establishing the results' statistical validity, independent of the probabilistic model elements. Because computer programs, and thus computational models, are inherently deterministic, random factors are modeled by using so-called pseudo random number generators (RNGs). RNGs generate a deterministic sequence of numbers, with the desired statistical properties, depending on an initial value termed *seed*. Different seeds result in different random number sequences. Thus, the task of establishing the results' statistical validity involves running the simulation with various RNG seeds and analyzing the collected results.

This chapter starts with an overview the problem of efficiently executing and analyzing large computational simulations. Then we discuss potential solutions, like the options of distributed execution and non-naïve approaches for sampling the parameter space. This is followed by a detailed discussion of the tools developed for addressing these issues, and for adopting the previously discussed solutions, focusing on general purpose tools and methods.

12.2 Problem Analysis

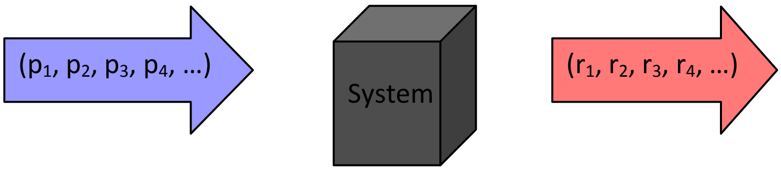

$(p_1, p_2, p_3, p_4, \ldots)$
System
$(r_1, r_2, r_3, r_4, \ldots)$

Figure 43. Simulation as data processing: input parameters are transformed into simulation results

Simulations are computer programs and computational experiments are carried out to investigate their behavior. Generally speaking, computer programs transform their input data into output data (user-interaction can be and often is viewed as real-time input). In this sense, computer programs and thus computer simulations are multi-variate, multi-valued functions (Figure 43). However, the "transformation rules" are typically very complex even for very simple programs. Therefore, it is generally infeasible to derive closed form solutions for the above "function". This is why simulation execution is used to map the "behavioral function" of the simulation.

This is akin to non-computational experiments where experimenters try to determine how the system's *response* (output) depends on controllable *factors* (parameters), while controlling for natural variability (e.g., stochasticity in case of computer simulations) by doing *replicates*.

The problem is, however, the *parameter space* (i.e., the set of relevant inputs) of any minimally interesting simulation tends to be large. (Having 5 relevant parameters each having 10 discrete values of interest immediately yields a set of 5^{10} ~10 million potential input combinations.) Considering that the run-time of simulations is typically in the order of minutes or hours, it becomes clear that running simulations at this scale and complexity is very computationally intensive task.

There are two basic approaches to address this problem. One increases the computational capacity available, while the other attempts to lower the size of the sample taken from the parameter space without losing too much information. (Given the size of the potentially relevant parameter space, in practice all simulations are only studied by *sampling* the parameter space.) In other words, the first approach tries to explore as large an area of the parameter space as possible, but attempts to make this possible by using more than a single computer. On the other hand, the second approach derives carefully designed plans to pick the parameter sample, in order to be able to learn as much about the model's behavior, as possible.

In the following, we discuss these two approaches in more detail, also looking at software tools supporting these approaches.

12.2.1 Feasibility of Distribution

Even in case of carefully designed experiment plans, the task or executing the simulation with the selected parameter combinations (i.e., initialized with the selected points in the parameter space) may easily exceed the abilities of today's PCs or workstations. Therefore, it is often useful if the execution of computational experiments is distributed among several computers on a network. There are two basic approaches to do this.

In the first, each simulation instance is run on a single computer, but the many instances required for parameter space exploration are distributed over the network. In the second, even components of the same simulation run (e.g., the agents) may be divided up among the participating computers. In the former case, neither the size nor the computational requirements of any of the simulation runs to be executed can exceed the capabilities of the participating (single) computers. However, this approach is relatively easy to implement, and can speed-up the execution of the set of experiments (typically involving thousands or tens of thousands single simulation runs).

On the other hand, the latter case allows for the execution of simulation runs that exceed the capacity of any participating computer. However, this comes at a price of considerable implementation efforts on the side of the engineers of the execution platform, and typically also on the side of simulation developers. It also raises the issue of close synchronization among the participating computers, since discrete time models typically assume a single "central clock", which is often inconvenient in a networked environment with computers with varying computational abilities. (It can often be the case that the entire "computational pool" operates at the rate of the weakest participating unit.)

Moreover, in case of models with strong interaction/communication among the agents, the second type of distribution is highly impractical due to the increased communications costs. Also, according to common wisdom, it is impossible to develop "general" parallel or distributed code – thus, the modeler's collaboration is needed to create a distributed version of their simulations, which renders general purpose tools of this kind unlikely and impractical. Ongoing efforts, like in the QosCosGrid project (The QosCosGrid Project), are to be noted that are aimed at the development of template implementations for a *certain communication patterns* that are common in agent-based computer simulations (Gulyás et al., 2008). This approach has the potential to offer semi-automatic parallelization for modelers, who will only need to extend the appropriate parallel simulation template with the lines relevant to their particular model. However, the mentioned templates are not available yet and the offering is admittedly incomplete at the time of writing.

On the other hand, the first type of distributed execution is within reach. Distributed parameter sweeps are possible because individual simulation runs in parameter sweep experiments are *completely independent* (often also termed “embarrassingly parallel” or “perfectly parallel”). That is, the output of a particular run does not depend on the result of any other run. Since the runs to be executed (i.e., the parameter combinations to be explored) can be fixed in advance, this means that there is no need to wait for the completion of any run in order to be able to start another one.

Note, however, that in its entirety, the above statement only holds for the naïve approach of setting the parameter combinations. With the introduction of the more “intelligent” parameter exploration methods discussed below, this independence condition might be violated. However, a distributed approach can still be useful there. On one hand, in case of several experiment designs the independence condition is preserved. On the other, even in the case of methods that introduce dependence between runs, it is typically possible to arrange runs in “batches” containing independent runs that can be executed in parallel. This way, dependence is confined to among batches and the entire experiment can be executed as a sequence of batches that each contains numerous distributed simulation runs.

Distributed parameter sweeps are relatively easy to implement, but they are better handled at the simulation platform level than at the level of individual simulations. First, this is the natural level of generality. Second, this task might be too technical for modelers. Therefore, we have adapted this approach and developed a general, easy-to-use tool for computational experiments that is capable of distributed simulation runs. This will be detailed in a separate section below.

12.2.2 Specially Assembled Parameter Configurations

The second approach to handle the large computational demand of simulation experiments is the “smart” assembly of the set of parameter combinations, so that it reduces the size of the set without heavily compromising the results and their validity.

There are two simple, traditional approaches to “batch experiments” as large scale explorations of the behavior of computational simulations are often called. One is the so-called *One-Factor-At-A-Time (OFAT)* method that changes the value of a single parameter, while keeping the others at their default value. This approach is very effective in reducing the number of simulation runs and works well if there is little interaction among the parameters and the parameters have a reasonably meaningful “null value”. (For example, this is the case if the model is to investigate the effect of changes from the present situation of the system.) However, in case of agent-based simulation models, default (null) values are often very hard to find or argue for, and interactions are more often than not play a very important role. (In these cases, the effect of changing several parameters together affects the output more heavily than changing the parameters individually.) The other traditional approach addresses this problem by the application of “brute force” – combining and testing all values of all parameters that would be tested in an OFAT experiment, thus creating a *full factorial* design. However, while this solution handles the problem of interactions perfectly, it does so at the maximum price, i.e., measuring the response at all possible points of interest. In practice, one must be smarter than that! Luckily, it is also possible to do so.

In many non-computational disciplines experiments are costly, hard to carry out and/or imply ethical issues (like experimenting with living beings, etc.). In these fields limiting the number of *experimental runs* is imperative. It is for this reason that *experiment design* has a long tradition and a large literature dating back as far as the 1920s (Box et al., 2005). Unfortunately, these methods are often overlooked in the practice of agent-based computational simulations, or more generally, in complex systems studies. Recently, this methodology has found its way to the general computational simulation community, but as of yet, it is not general practice there either (Koehler and Owen, 1996; Santner et al., 2003).

Below, we provide a short overview of the *Design of Experiments (DoE)* methodology and concepts and introduce some of the simplest designs, following in the footsteps of (NIST/SEMATECH, 2008).

12.2.3 Design of Experiments

In the context of DoE, the definition of an experiment is the study of a given system by “(…) deliberately changing one or more process variables (or factors) in order to observe the effect the changes have on one or more response variables. The (statistical) design of experiments is an efficient procedure for planning experiments so that the data obtained can be analyzed to yield valid and objective conclusions" (NIST/SEMATECH, 2008).

A *design* is a detailed plan in advance of doing the experiment. The goal is to maximize the amount of "information" that can be obtained for a given amount of actual experiments carried out. There are several motivations to design an experiment. According to (NIST/SEMATECH, 2008), one can be interested in "choosing between alternatives; selecting the key factors affecting a response; regression modeling; response surface modeling including hitting a target, maximizing (or minimizing) a response, reducing variation, making a process robust and seeking multiply goals." Among these, selecting the key factors and response surface modeling are the most relevant in the context of agent-based computational simulation. Regression modeling and reducing variation might also be interesting, albeit most agent-based models are non-linear and high variation regimes might themselves be an important result of such models.

For the most relevant motivations, the DoE literature offers the following broad categories of designs, with several objectives for each (NIST/SEMATECH, 2008).

Screening Designs

Screening designs, as the name suggests, are used for initial screening of the system's behavior. They help identifying which factors/effects are important. If there are only 2-4 potential factors of interest, one may be able to afford a *full factorial* design (essentially, the traditional "brute force" approach of batch computational experiments). However, when having more than 3 potential factors of interest, one may want to begin with as small a design as possible. Screening designs are also useful when having qualitative factors or when suspecting non-monotonic effect on one on the quantitative factors.

Response Surface Modeling

Response Surface modeling is fundamentally a way to create a (simplified) closed-form model of the response surface. This can be used to achieve various further objectives, like hitting a target value, maximizing or minimizing a response, reducing variation, etc.

Regression Modeling

Regression Modeling, as its name suggests, is used to estimate a precise model of the response and quantifying the dependence of response variable(s) on process inputs. This is typically done the assumption of linear response.

Naturally, the real art of DoE lies in the particular design addressing one of the above objectives. In the following, we will introduce a few of them. Some of these designs are already available for computational experiments in the Parameter Sweep tool developed in the EMIL project (see their discussion below in a separate section), while the rest and others are being currently developed.

Special Designs and Techniques in DoE

This section introduces a few simple experiment designs and techniques in order to provide a glimpse on the kind of techniques offered by the DoE literature. Here, again, we are relying on the very useful summary of (NIST/SEMATECH, 2008).

Full Factorial Designs in Two Levels

Above we have discussed the traditional approaches of computational experiments that often apply "brute force" to the problem of studying the behavior of the system. In fact, this technique creates a *full factorial* design, in which all possible combinations of all values of all factors are included in the design. A simpler and more common experimental design is the *two level* full factorial design, in which all input factors are allowed two values (are set at two levels) each. For convenience, these levels are called "high" and "low" or "+1" and "-1", respectively.

Naturally, if there are k factors, each at 2 levels, a full factorial design has 2^k runs. When the number of factors is large, a full factorial design requires a large number of runs, even with two levels, and thus it is not very efficient. In these cases, a fractional factorial design or a Plackett-Burman (Plackett and Burman, 1946) design is a better choice.

Fractional Factorial Designs

Fractional designs reduce the size of the design by omitting some of the combinations of a factorial element. Or more precisely, a fractional factorial design is "*A factorial experiment in which only an adequately chosen fraction of the treatment combinations required for the complete factorial experiment is selected to be run*" (ASDQ, 1983). In general, about ½, ¼ of the combinations defined by the full factorial is picked. Of course, the art is in selecting the "adequate" fraction, but luckily, there is a large literature containing good advices. Various strategies exist, which we cannot discuss here. However, it is important to emphasize that properly chosen fractional factorial designs for 2-level experiments have the desirable properties of being both balanced and orthogonal. This means that all combinations have the same number of observations, and that the effects of any factor balance out (sum to zero) across the effects of the other factors (which ensures that the design does not "waste" experiments in regard to the information gained, even if one factor has no effect at all on the response).

In case of two-level fractional designs, the set of combinations are often extended by a "center point" (between the low and high values for each factor studied) in order to estimate the curvature of the response function.

Box-Wilson Central Composite Designs

The Box-Wilson Central Composite Design, also called "central composite design", consists of two parts. It contains a factorial or fractional factorial design with center points. In addition, it contains a group of "star points" that allow the better estimation of curvature. Central points are specified depending on the variety of the central composite design. In the *circumscribed* design variety, if the distance from the center of the design space to a factorial point is normalized to ±1 unit for each factor, then the same distances for the star points are $\pm\alpha$, where $|\alpha\ | > 1$. The precise value depends on certain desired properties of the design and also on the number of factors. The star points represent new extreme values (low and high) for each factor in the design. On the other hand, if the limits specified for the factors are true limits, the *inscribed* variety of the design creates a factorial or fractional factorial design within those limits (i.e., it is a scaled down circumscribed design with each factor level divided by α). Finally, in the *face centered* variety the star points are at the center of each face of the factorial space, so $\alpha=\pm 1$. Importantly, central composite designs always contain twice as many star points as there are factors in the design (NIST/SEMATECH, 2008).

Box-Behnken Designs

The designs discussed so far were fundamentally based on the assumption of linear response, with center points testing for curvature and thus checking the validity of this assumption. Since computer simulations often produce non-linear responses, it is very useful to have designs that can handle such situations. One of the simplest of such designs is the Box-Behnken design, which is a quadratic design that does not contain an embedded factorial or fractional factorial design. Here the combinations are at the midpoints of edges of the parameter (process) space and at the center. Geometrically, this design resembles a sphere within the parameter space such that the surface of the sphere protrudes through each face with the surface of the sphere tangential to the midpoint of each edge of the space. Given their spherical nature, these designs are rotatable (or near rotatable) and require 3 levels of each factor. The designs have limited capability for orthogonal blocking, in comparison to the central composite designs (NIST/SEMATECH, 2008).

Latin Hypercube Designs

Latin Hypercube Designs (LHDs) have the specialty that they consist of *n* runs in a setting where each factor has *n* distinct levels (not necessarily the same *values* for each factor). More formally, a *k*-dimensional Latin Hypercube Design is a set of *n* points $x_i = (x_{i1}, \ldots, x_{ik}) \in \{0, \ldots, n\text{-}1\}^k$ such that for each dimension *j* all x_{ij} are distinct. The best Latin hypercube designs are often based on orthogonal arrays. Usually the factor levels are equally spaced, which is often achieved by ensuring the "maximin" property, i.e., that maximizes the separation distance (the minimum distance among points of the design) among all LHDs of a given size *n*, according to some distance measure. Maximin LHDs designs are hard to find, however, therefore for larger designs and higher dimensionalities, often approximations are used.

Latin Hypercube Designs are especially useful for computer experiments, because they ensure that few design points are redundant when there is effect sparsity. It was also observed that "the designs proposed for computer experiments have almost exclusively been Latin Hypercube Designs" (Butler, 2001).

Randomized Block Designs

In certain cases, the experimenter has initial knowledge or assumption suggesting the primary the factor of interest, yet, she needs to prove it excluding the possible effects of other, "nuisance" factors. Such situations are more common in non-computational experiments, but they also exist in the context of computer simulations. For example, in real world experiments nuisance factors might be the specific operator who carried out the experiment or applied the prescribed treatment, the time of the day the experiment was run, the room temperature, etc. In short, nuisance factors are those that may affect the measured result, but are not of primary interest.

If it is possible to control nuisance factors, the technique known as *blocking* can be used to reduce or eliminate their contribution to experimental error. The basic concept is to create homogeneous blocks in which the nuisance factors are held constant and the factor of interest is allowed to vary. Within blocks, it is possible to assess the effect of different levels of the factor of interest without having to worry about variations due to changes of the block factors, which are accounted for in the analysis.

When nuisance factors abound, Randomized Block Designs may come to help. In this scenario, blocking is used to handle a few of the most important nuisance variables, while *randomization* is used to reduce the contaminating effects of the remaining ones. This technique can also be viewed as a set of completely randomized experiments, each of them run within its own block (i.e., in one of the blocks in the entire experiment).

12.2.4 Iterative Approach to Designing Experiments

The designs introduced so far were all "static" in that they designed the entire experiment in advance, fixing all the design points and excluding feedback from the already observed responses to the selection of the design points tested in the future. Indeed, this is the classic DoE approach, but even there the importance of an iterative approach is emphasized, calling it a mistake to believe that "one big experiment will give the answer" (NIST/SEMATECH, 2008). The concept of the iterative approach is that *each stage provides insight for next*.

The heavy computational demand of the planned simulation experiments in the EMIL project creates a special emphasis on the design and sequencing of the individual runs, even when they are executed distributed on multiple computers. The concept is that the sequencing of the runs should allow for branching depending on earlier results and also for revisiting previously explored areas of the parameter space with greater "resolution".

Therefore, a special emphasis was put on studying and developing methods and heuristics with the above dynamic and iterative nature. Preliminary results of these efforts are discussed at the end of the next section, introducing the software tools for effective simulation execution developed in the EMIL project.

12.3 Tools for Efficient and Effective Simulation Execution

12.3.1 Preparing Simulation Runs

In the EMIL project we have developed a general purpose, easy-to-use simulation execution platform (Parameter Sweeper) for agent-based simulations, integrated in a dedicated software module for computational experiments, the Model Exploration Module (MEME) that is designed to maintain, process and analyze results of computational experiments. Both of these tools are integrated to AITIA International's Multi-Agent Simulation Suite (MASS) (Iványi, Bocsi et al., 2007; Iványi, Gulyás et al., 2007).

The Parameter Sweeper works with any Repast J 3.1 simulations (North et al., 2006) (even with those written without knowledge about the tool) and offers user friendly options for the design and execution of simulation experiments, as well as for the collection of simulation results. The tool has limited support for the new version of Repast, Repast Simphony as well (North et al., 2007) (about 80% of the functionality is available), but it is also integrated with EMIL-S. The Parameter Sweeper's "back-end" has a plug-in architecture that offers real multi-platform support for modelers. (In addition to EMIL-S and Repast Simphony, the architecture enables support for simulations written in pure Java and in NetLogo. In the longer term, support for other platforms, like MASON, will also be possible, see Luke et al., 2003; Wilensky, 1999).

The basic functionality of the MEME Parameter Sweeper is the following:

- The loading in of the model.
- Automatic identification of the model parameters.

In most modeling environments, (global) variables can be flagged as parameters, i.e., values whose initial value can and should be changed before the simulation starts.

Optional Extension of the Parameter Set.

Sometimes the modeler is tempted to explore the model's response to the changing of certain values that were not flagged as parameters in the original model. Unfortunately, it is often inconvenient to do this, because it involves referring back to the original modeling environment (and, e.g., recompiling the source code). Therefore, the MEME Parameter Sweeper offers the functionality to extend the model's parameter set (i.e., flagging new variables as parameters) "on the fly", using a user friendly graphical wizard and without the burden of referring back to the original code and recompiling it.

Assembling the Parameter Combinations to be Explored.

The MEME Parameter Sweeper offers an easy-to-use, user friendly graphical wizard for the specification of the set of parameter combinations (i.e., the "parameter space") to be explored. (This base level feature generates *full factorials*, or traditional "brute force" experiments. Advanced features of the Parameter Sweeper supporting DoE designs will be discussed later.)

Definition of the Data to be Collected.

Simulation experiments are useless without the values of certain model variables (responses) are measured and collected. In most existing simulation platforms this measurement and collection must be implemented by the modeler within the code of the model. This is an unfortunate practice for several reasons. First, the modeler uses its time and efforts on coding routine tasks. Second, the model should be changed regularly, following the data requirements of the different experiments. Or conversely, the model's code becomes packed with uninteresting, technical code. Third, data collection can be tricky in case of distributed simulation experiments or in case of adaptive, dynamic designs. Therefore, in contrast to most existing simulation packages, the MEME Parameter Sweeper offers an easy-to-use, very flexible, user friendly graphical wizard for the specification of data collection requirements. Any global variable of the model can be selected, specifying the frequency or condition when the data is to be recorded.

Optional Extension of Variables (Definition of Statistics).

The Data Collection wizard mentioned earlier is a useful tool for picking global variables whose responses need to be measured. However, often the required measurement process is more complicated that storing the value of a simple variable. Rather, it often involves storing the result of a model method (activity), or the even more problematic case when a certain statistics of a variable or some collection of variables needs to be recorded. To address the first need, the MEME Parameter Sweeper can take the output of any model method returning a numerical value. For the latter need, a new wizard is offered that provides a menu-based wizard for the point-and-click assembly of descriptive statistics (based on the colt package, see Colt, 2004), as well as, tools for construction and transformation of data collections from variables and return values available in the model. The data construction operations and the statistics defined using the wizard will result in Java code that is "attached" to the original model on-the-fly, using advanced Java techniques.

Expert users can go even one step further and providing their own statistical routines (scripts) by a few lines of Java code, but without the burden of specifying the entire context or a fully fledged Java method. These lines are also be attached to the model on-the-fly.

The statistics and scripts defined in this wizard also appear among the selectable items in the Data Collection wizard of the previous bullet point, thus making them as flexibly configurable for recording as any original, "first class" variable.

Execution of the Computational Experiment

After the specification of the parameter space to study and the measurements to be performed and the data to be recorded, the MEME Parameter Sweeper wizard executes the specified simulation experiment. Depending on the particular settings (editable via the Settings panel of the tool), the same experiment can be executed i) on a local computer, ii) on a local cluster of computers, or iii) on a remote grid system. (These options are further detailed below.) After the experiment execution starts, the MEME Parameter Sweeper offers a Monitor application to follow the progress of the experiment. In case of distributed experiments (i.e., options ii) and iii) above) this Monitor application can be closed without stopping the experiment and the submitting computer can be shut down or disconnected from the executing network. The system can be configured to send an e-mail notification on the completion of the experiment, when the Monitor application can be started up again to collect the results.

Collection of the Experimental Results

The collection of the experimental results is handled automatically by the MEME Parameter Sweeper, even in case of distributed experiments. For the downloading of remotely collected results (i.e., in case of distributed experiments) the Monitor application needs to be started to initiate the process. However, the downloaded results can be automatically imported to the data organization and analysis part of MEME.

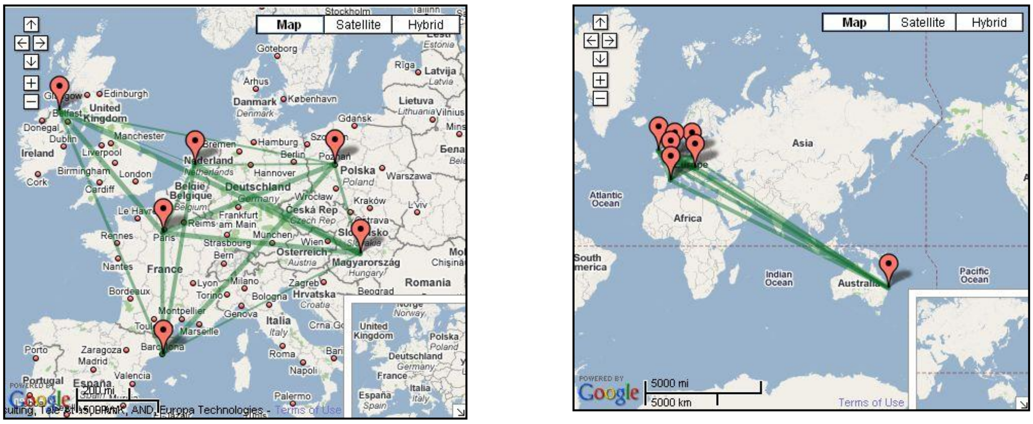
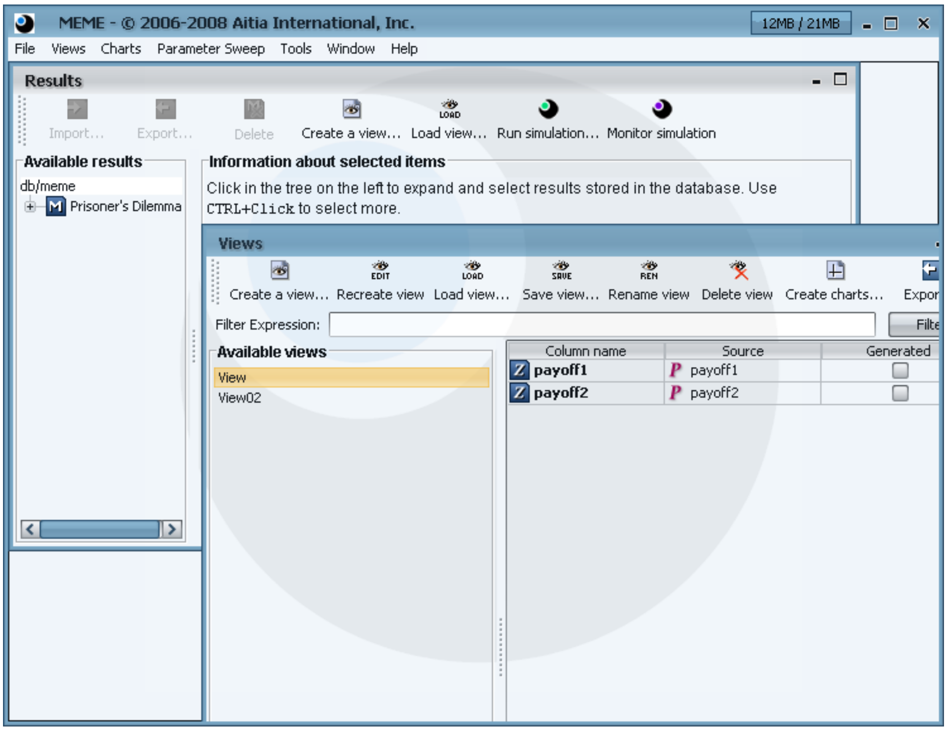


Figure 44. The configuration of the QosCosGrid experimental testbed during the summer of 2008

The analysis of the collected experimental results can be carried out by either conventional statistical packages, or can be based on the results maintenance functions of MEME.



Click in the tree on the left to expand and select results stored in the database. Use CTRL+Click to select more.

Column name	Source	Generated
payoff1	payoff1	☐
payoff2	payoff2	☐

Figure 45. The general window of MEME

The MEME Parameter Sweeper has two layers of additional functionality as well. This allows for the distributed execution of computational experiments by distributing individual runs across several computers. The functionality exists at two levels: the experiment can be distributed on a local cluster (e.g., connected by a high-speed LAN), or it can be executed on a grid system (i.e., on a “cluster of clusters”, where geographically dispersed local clusters are connected by the internet). The latter functionality requires that a version of the QosCosGrid middleware is installed on all computer clusters. Figure 44 shows the configuration of the experimental QosCosGrid server during the summer of 2008.

The other layer of additional functionality addresses one of the key tasks of simulation execution at the large scale. Its functionality concerns the “intelligent”, automatic definition of the set of explored parameter combinations for efficient and effective, methodologically sound exploration of the model’s behavior in the parameter space. This layer is based on the concept of *experiment design* and draws heavily from the Design of Experiments (DoE) literature, both discussed earlier (Box et al., 2005). Technically, the layer is implemented using a plug-in architecture, which allows for the flexible and dynamic addition of new methods and heuristics, and thus for incremental development. Currently, a few basic DoE designs are available, while more advanced tools are being currently developed. Among these are several that augment classic DoE designs with tools from Artificial Intelligence and Data Mining.

In the following we introduce the functions of the Model Execution Module (MEME) and those of the Parameter Sweeper integrated in it.

12.3.2 Storing and Organizing Simulation Results

MEME (shown on Figure 45) stores simulation results in a database, that includes all fix (constant) and changing parameters, and various additional information about the model (i.e. name, version, description, etc.). The software has a built-in Java-based database engine that can manage databases up to 8 GB in size, but the system is built in a way that it is independent of the particular SQL engine used, it supports professional database engines through the JDBC protocol. The program organizes raw data in a 3-level hierarchy (model, version and batch). The database-structure is created to be able to handle repeated exploratory runs, iterative, gradual import of results (i.e. new parameters being introduced and old ones deleted between versions of the same model).

The basic functions of MEME and its data flow are summarized on Figure 46. MEME can obtain simulation results in two general forms. Results from batches of simulations designed and ran through MEME are automatically acquired. The other option is running simulations separately and then importing Repast result files (see Figure 47) or generalized CSV files into the database. The program offers a very flexible wizard for reading in data from text files. This wizard combines the simplicity of CSV (comma separated values) format readers with the power of regular expressions and printf-like formatting. To enable importing data from a high number of external simulations, the program supports the simultaneous import of multiple files, as well as allows the saving of import settings for later use.

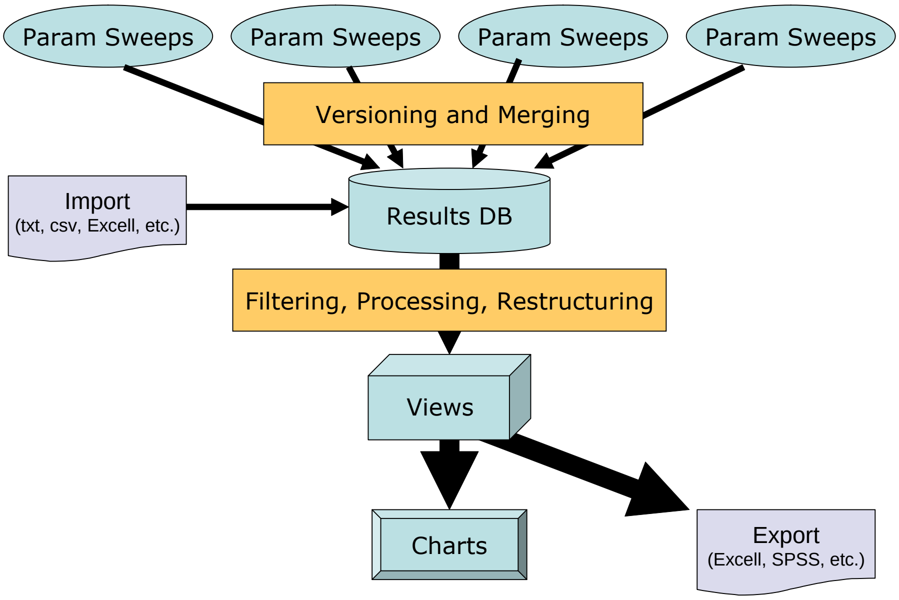
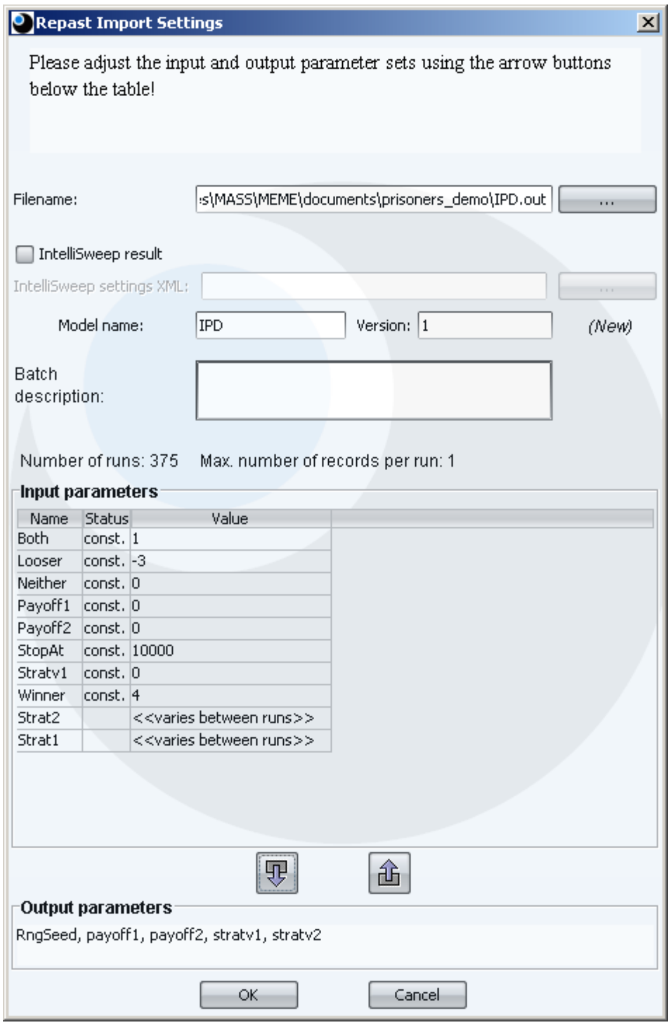


Figure 46. The basic activity diagram of MEME

The results obtained and stored in the database can later be transformed into computed tables that can be visualized and/or provide input for more sophisticated analysis. MEME enables the user to filter out rows or columns, calculate new, derived columns, split columns into multiple new columns (a generalized form of cross tabulation), aggregate values (i.e., calculate the average, variance, etc. of a selection), reorganize, etc. and execute various custom computations on the data without any or with minimal coding. Once the data is organized into the desired form, it can be exported in CVS format for analysis in advanced statistical software of the user's choice, or visualizations can be created with MEME's built-in Charting Wizard.

These steps are discussed in more detail below.



Please adjust the input and output parameter sets using the arrow buttons below the table!

Filename: :s\MASS\MEME\documents\prisoners_demo\IPD.out

☐ IntelliSweep result

IntelliSweep settings XML:

Model name: IPD Version: 1 (New)

Batch description:

Number of runs: 375 Max. number of records per run: 1

Input parameters

Name	Status	Value
Both	const.	1
Looser	const.	-3
Neither	const.	0
Payoff1	const.	0
Payoff2	const.	0
StopAt	const.	10000
Stratv1	const.	0
Winner	const.	4
Strat2		<<varies between runs>>
Strat1		<<varies between runs>>

Output parameters

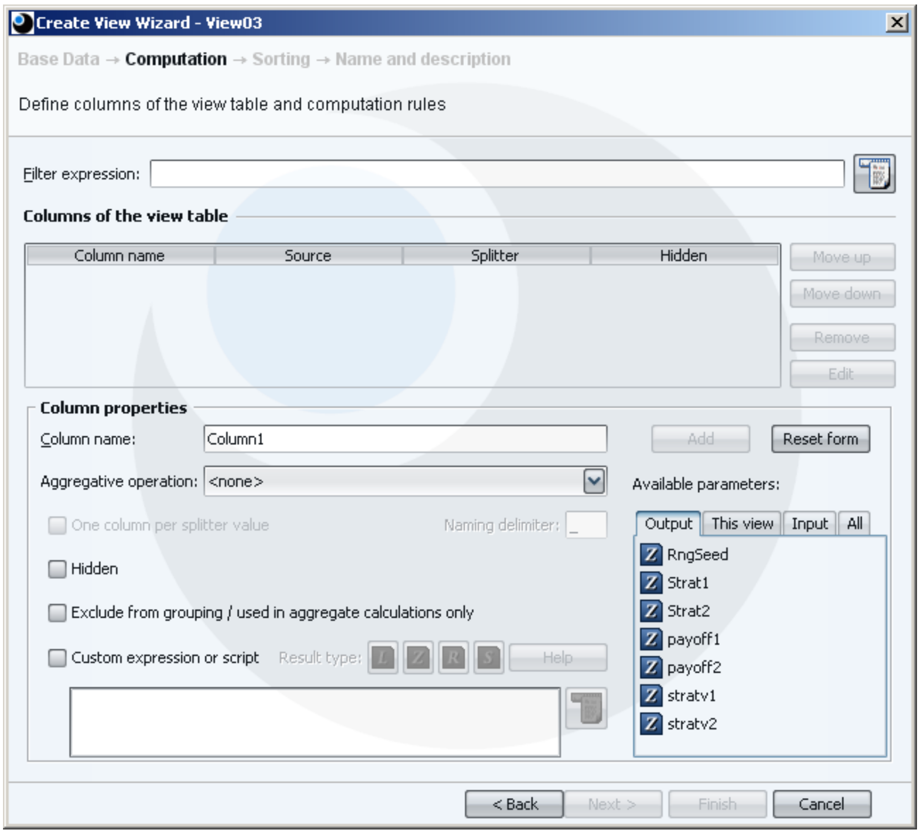
RngSeed, payoff1, payoff2, stratv1, stratv2

Figure 47. The MEME import wizard for Repast simulation outputs

Storing Simulation Results

Simulation results are stored in a three-level hierarchy. Results belonging to different versions of the same model can be grouped, since models are often developed in an iterative fashion. Similarly, results generated by separate experiments (batches) of the same version can also be stored together. Result tables belonging to the same version of a model always have the same set of parameters. Adding results with different parameter sets to the database is possible albeit not recommended. In this case the new parameters – that are missing from the already stored results – will be inserted with “null” values into all stored result tables of that version and model; and if some of the existing parameters are missing from the new result, “null” values will be added there, too. In result tables MEME supports numeric, textual (string) and logical (true, false) values.

The results stored together can be processed independent of one another, or together, merging separate batches or various versions. This is done by selecting the appropriate level of the hierarchy or the appropriate subset within a hierarchy as the object of the view creation process discussed below.


Create View Wizard - View03

Base Data → Computation → Sorting → Name and description

Define columns of the view table and computation rules

Column name: Column1

Aggregative operation: <none>

Figure 48. The view creation wizard

Processing Simulation Results

Processing simulation results in MEME is centered on the concept of *view tables*. Views are derived result tables that are assembled from and contain manipulated raw simulation data. View creation is assisted by a user friendly graphical wizard shown on Figure 48. This is a rather complicated tool, but its general logic is to

i. specify the columns from among the ones available in the result tables selected,

ii. provide a filter expression for the selection of rows to be kept,

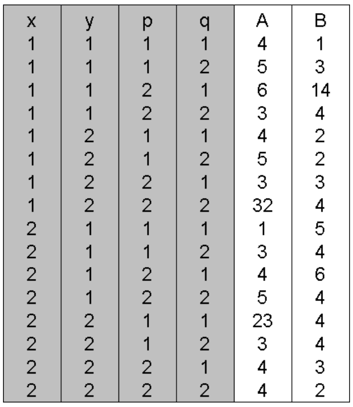
iii. (optionally) create new columns whose values are computed from the columns available in the model, using basic formulas or advanced scripts (e.g., statistics),

iv. (optionally) specify aggregative operations for values of columns that could have more than single values for unique combinations of the rest of the columns,

v. (optionally) split columns into multiple other columns based on unique values of another column (or based on the unique combination of values in other columns). E.g., create columns X1, X2, …, X*n* from the column X, based on individual values in column Y (assumed to be 1, 2, …, *n* in the example), and

vi. (optionally) sort rows of the generated columns as necessary.

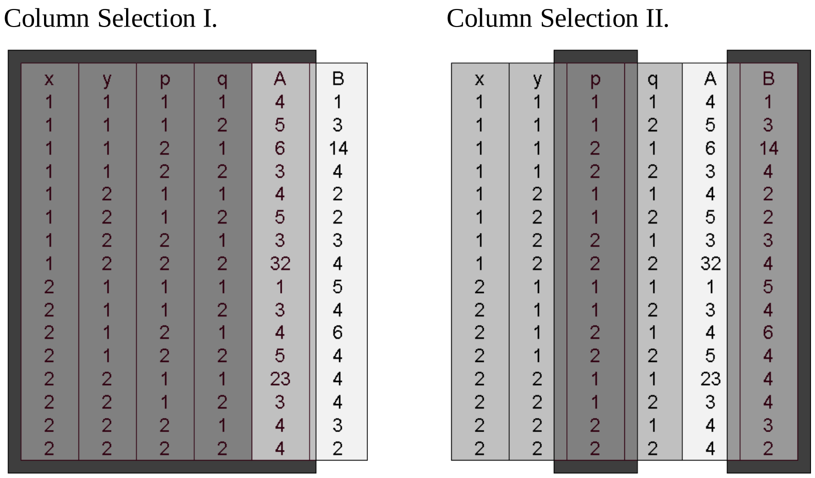
The settings creating a particular view can be saved and re-used, even on a multiple selection of results, allowing for a single click batch processing of large data sets and for the generation of several hundreds of derived data sets.



x	y	p	q	A	B
1	1	1	1	4	1
1	1	1	2	5	3
1	1	2	1	6	14
1	1	2	2	3	4
1	2	1	1	4	2
1	2	1	2	5	2
1	2	2	1	3	3
1	2	2	2	32	4
2	1	1	1	1	5
2	1	1	2	3	4
2	1	2	1	4	6
2	1	2	2	5	4
2	2	1	1	23	4
2	2	1	2	3	4
2	2	2	1	4	3
2	2	2	2	4	2

Figure 49. The result table of the original data set

The above functions of view creations are illustrated by the following example. Let's assume that our original data set is stored in the result table shown on Figure 49. It contains two parameters *x* and *y*, two pseudo random number generator seeds *p* and *q* and two output (response) variables *A* and *B*.



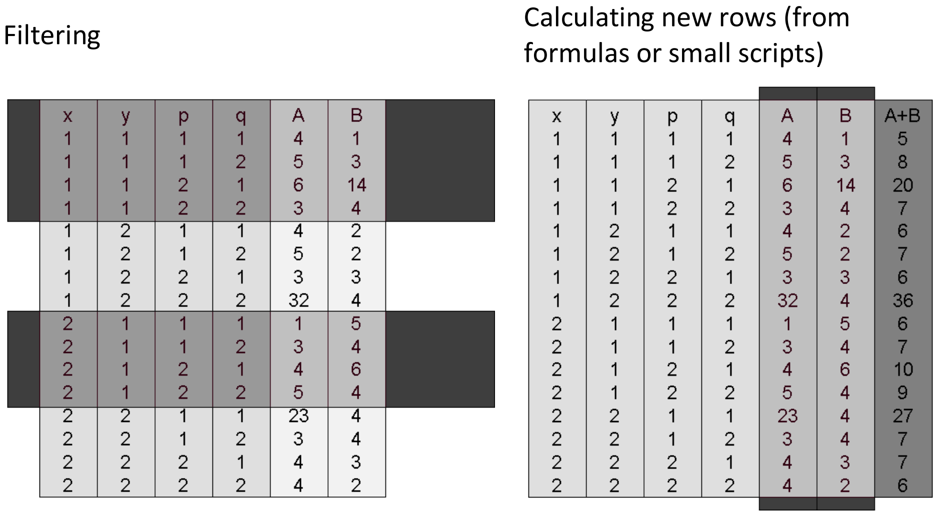
Column Selection I.

x	y	p	q	A	B
1	1	1	1	4	1
1	1	1	2	5	3
1	1	2	1	6	14
1	1	2	2	3	4
1	2	1	1	4	2
1	2	1	2	5	2
1	2	2	1	3	3
1	2	2	2	32	4
2	1	1	1	1	5
2	1	1	2	3	4
2	1	2	1	4	6
2	1	2	2	5	4
2	2	1	1	23	4
2	2	1	2	3	4
2	2	2	1	4	3
2	2	2	2	4	2

Column Selection II.

x	y	p	q	A	B
1	1	1	1	4	1
1	1	1	2	5	3
1	1	2	1	6	14
1	1	2	2	3	4
1	2	1	1	4	2
1	2	1	2	5	2
1	2	2	1	3	3
1	2	2	2	32	4
2	1	1	1	1	5
2	1	1	2	3	4
2	1	2	1	4	6
2	1	2	2	5	4
2	2	1	1	23	4
2	2	1	2	3	4
2	2	2	1	4	3
2	2	2	2	4	2

Figure 50. Column selection – overshadows represent selected columns



Filtering

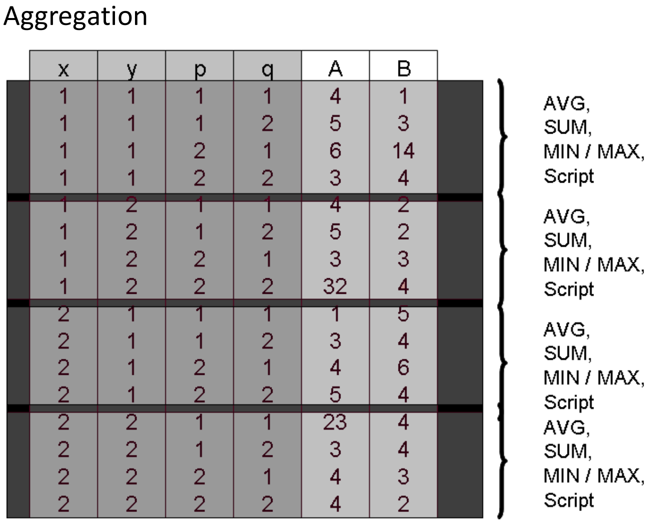
x	y	p	q	A	B
1	1	1	1	4	1
1	1	1	2	5	3
1	1	2	1	6	14
1	1	2	2	3	4
1	2	1	1	4	2
1	2	1	2	5	2
1	2	2	1	3	3
1	2	2	2	32	4
2	1	1	1	1	5
2	1	1	2	3	4
2	1	2	1	4	6
2	1	2	2	5	4
2	2	1	1	23	4
2	2	1	2	3	4
2	2	2	1	4	3
2	2	2	2	4	2

Calculating new rows (from formulas or small scripts)

x	y	p	q	A	B	A+B
1	1	1	1	4	1	5
1	1	1	2	5	3	8
1	1	2	1	6	14	20
1	1	2	2	3	4	7
1	2	1	1	4	2	6
1	2	1	2	5	2	7
1	2	2	1	3	3	6
1	2	2	2	32	4	36
2	1	1	1	1	5	6
2	1	1	2	3	4	7
2	1	2	1	4	6	10
2	1	2	2	5	4	9
2	2	1	1	23	4	27
2	2	1	2	3	4	7
2	2	2	1	4	3	7
2	2	2	2	4	2	6

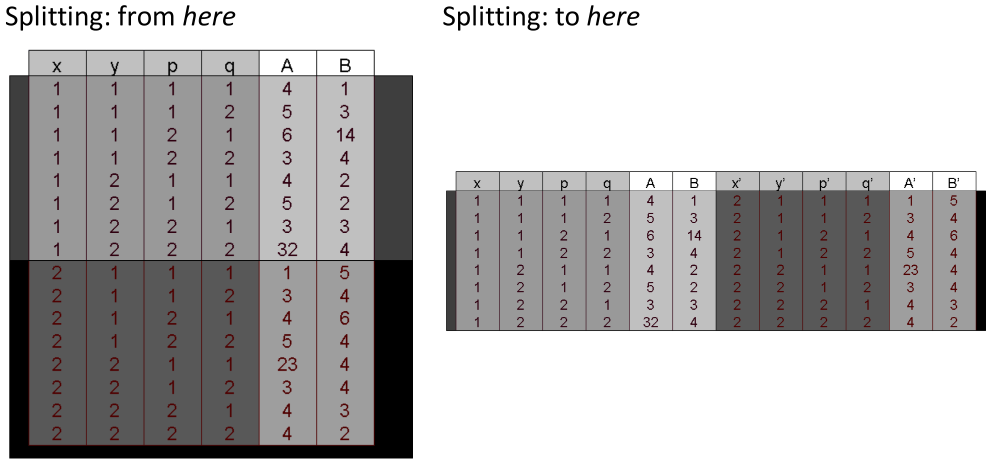
Figure 51. Row filtering and the generation of derived columns. Overshadows denote selected, while the dark background represents derived values

The column selection of item i) above is illustrated on Figure 50. Overshadows denote selected columns. On the other hand the filtering of rows (item ii) above) and the generation of new, derived columns (item iii)) is similarly illustrated on Figure 51. (The dark grey background denotes derived values.)



x	y	p	q	A	B
1	1	1	1	4	1
1	1	1	2	5	3
1	1	2	1	6	14
1	1	2	2	3	4
1	2	1	1	4	2
1	2	1	2	5	2
1	2	2	1	3	3
1	2	2	2	32	4
2	1	1	1	1	5
2	1	1	2	3	4
2	1	2	1	4	6
2	1	2	2	5	4
2	2	1	1	23	4
2	2	1	2	3	4
2	2	2	1	4	3
2	2	2	2	4	2

Figure 52. Aggregative operations. The overshadows denote sets of values to be aggregated



x	y	p	q	A	B
1	1	1	1	4	1
1	1	1	2	5	3
1	1	2	1	6	14
1	1	2	2	3	4
1	2	1	1	4	2
1	2	1	2	5	2
1	2	2	1	3	3
1	2	2	2	32	4
2	1	1	1	1	5
2	1	1	2	3	4
2	1	2	1	4	6
2	1	2	2	5	4
2	2	1	1	23	4
2	2	1	2	3	4
2	2	2	1	4	3
2	2	2	2	4	2

x	y	p	q	A	B	x’	y’	p’	q’	A’	B’
1	1	1	1	4	1	2	1	1	1	1	5
1	1	1	2	5	3	2	1	1	2	3	4
1	1	2	1	6	14	2	1	2	1	4	6
1	1	2	2	3	4	2	1	2	2	5	4
1	2	1	1	4	2	2	2	1	1	23	4
1	2	1	2	5	2	2	2	1	2	3	4
1	2	2	1	3	3	2	2	2	1	4	3
1	2	2	2	32	4	2	2	2	2	4	2

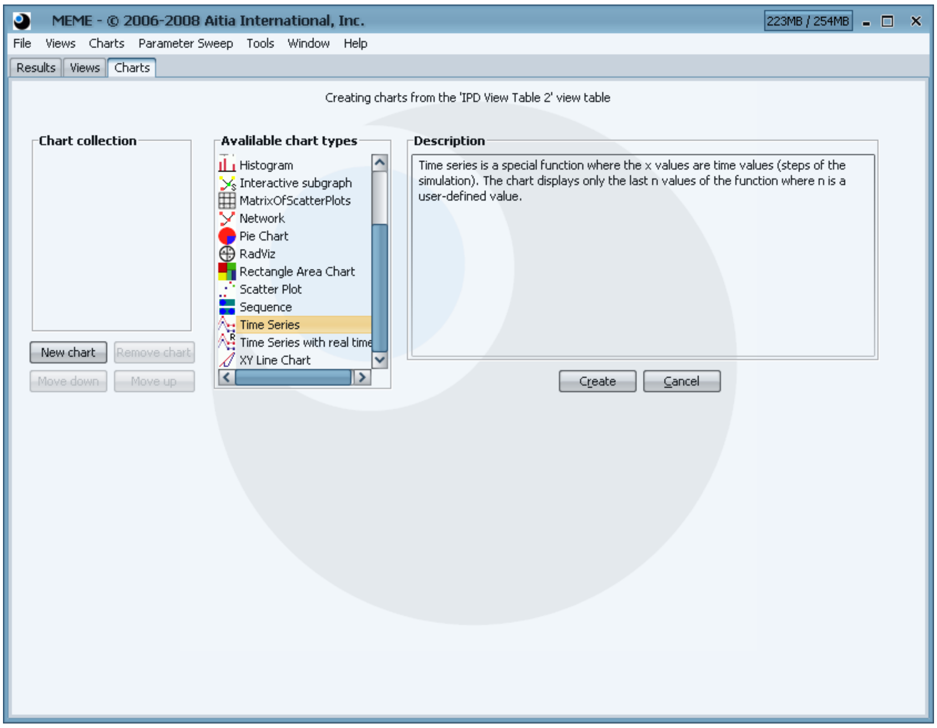
Figure 53. The splitting operation. The overshadows denote values belonging together

Figure 52 explains the aggregative operations of item iv), while Figure 53 illustrates the splitting functionality of the wizard (discussed under item v)). In the former case, the overshadows denote sets of values to be aggregated, while in the latter, the various colors of overshadows point out the values belonging together.

View tables are the essential output of the MEME’s processing of simulation data. They can be exported to a text file in a generalized CSV (comma separated values) format for further processing in a dedicated statistics package. (MEME does not intend to compete with these. Rather, it offers an easily accessible set of functions for results processing tasks common in computational simulations.) Alternatively, the processed results can be previewed by MEME’s own built-in charts.

Charting

Previewing simulation results in MEME is supported AITIA International’s Charting Package (CP) another component of the Multi-Agent Simulation Suite (MASS) that is integrated into MEME. Via this component, MEME offers a menu of common chart types, including time series, line, bar and pie charts, scatter and box plots (box and whisker charts), matrices of scatter plots, histograms, radial visualizations for data of high dimensionality (RadViz charts), grid plots, rectangle area charts, sequence diagrams, network (graph) displays, and some others (see Figure 54).


Creating charts from the 'IPD View Table 2' view table

Time series is a special function where the x values are time values (steps of the simulation). The chart displays only the last n values of the function where n is a user-defined value.

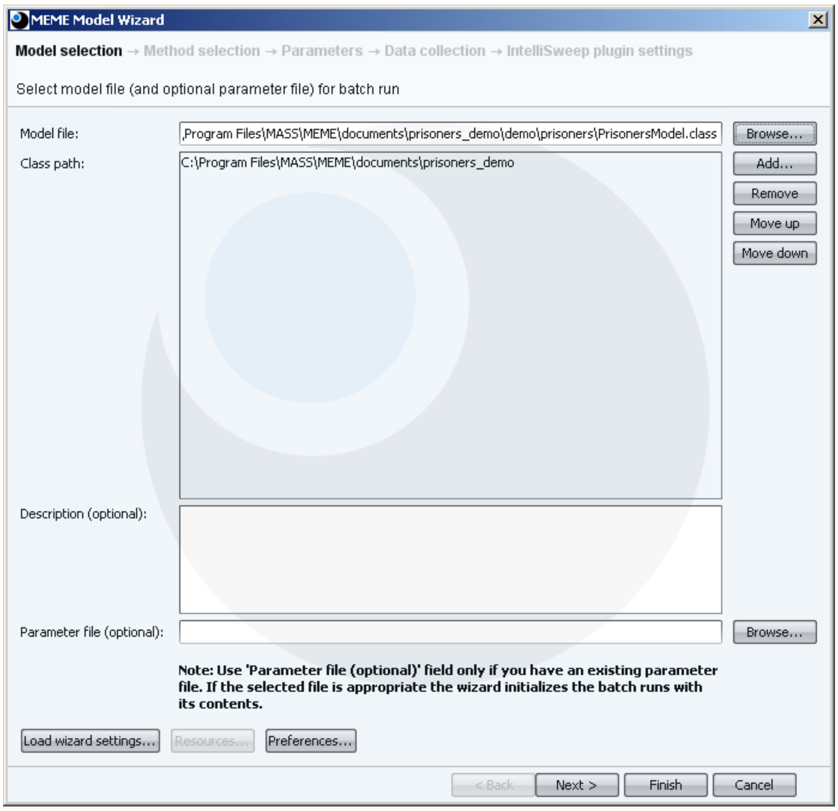
Figure 54. MEME's chart settings panel

The settings panel of every chart type contains drop-down lists allowing the user to select from the columns of the view table. The selected column will provide the data for the corresponding data variable of the chart. The displayed charts can also be exported in various file formats that can be directly inserted into documents.

12.3.3 The MEME Parameter Sweeper

The base functionality of MEME, concerning the storing, organizing and processing simulation results was discussed in the previous section. We now turn our attention towards the MEME Parameter Sweeper, the tool to assist modelers in the orchestration of large scale simulation experiments. The functionality of this component has been outlined earlier. Here we concentrate on the usage details of the tool, albeit we refrain from the level of details of software manuals. (The complete User Manual of MEME is available at the MASS website, http://mass.aitia.ai/, or more specifically, at the section dedicated to meme, http://meme.aitia.ai/.)

Selecting the Model



Select model file (and optional parameter file) for batch run

Model file: ,Program Files\MASS\MEME\documents\prisoners_demo\demo\prisoners\PrisonersModel.class

Class path: C:\Program Files\MASS\MEME\documents\prisoners_demo

Parameter file (optional):

Note: Use 'Parameter file (optional)' field only if you have an existing parameter file. If the selected file is appropriate the wizard initializes the batch runs with its contents.

Figure 55. The first step of the Parameter Sweeper wizard

The Parameter Sweeper tool is presented to users in the form of a graphical wizard. The first step of this wizard (shown on Figure 55) assists the user with the specification of the model to be explored. In principle, the *.class* file containing the model should be selected, together with the paths to all related classes. The directory where the model class can be found (in regard the package structure) is automatically added to the *Class path*. The user can extend this list manually selecting one or more directories or JAR files in a file dialogue. Alternatively, the user may take a more careless approach, letting the wizard detecting any missing components. This is because at the end of this step (after the user pressed the Next button) the wizard attempts to load the model. If the *Class path* is defined properly the wizard will move on to the next phase. Otherwise, if the wizard is unable to find a class, it will specifically prompt for its location in a file dialogue.

The *Description* and the *Parameter file* settings of the first wizard step are optional. The latter is for legacy purposes, allowing users with manually created Repast parameter files to be able to continue using their old settings. If an existing and appropriate parameter file is selected the wizard initializes the experiment with its content (but it is still modifiable). If no parameter file is defined, the wizard will generate one in later steps.

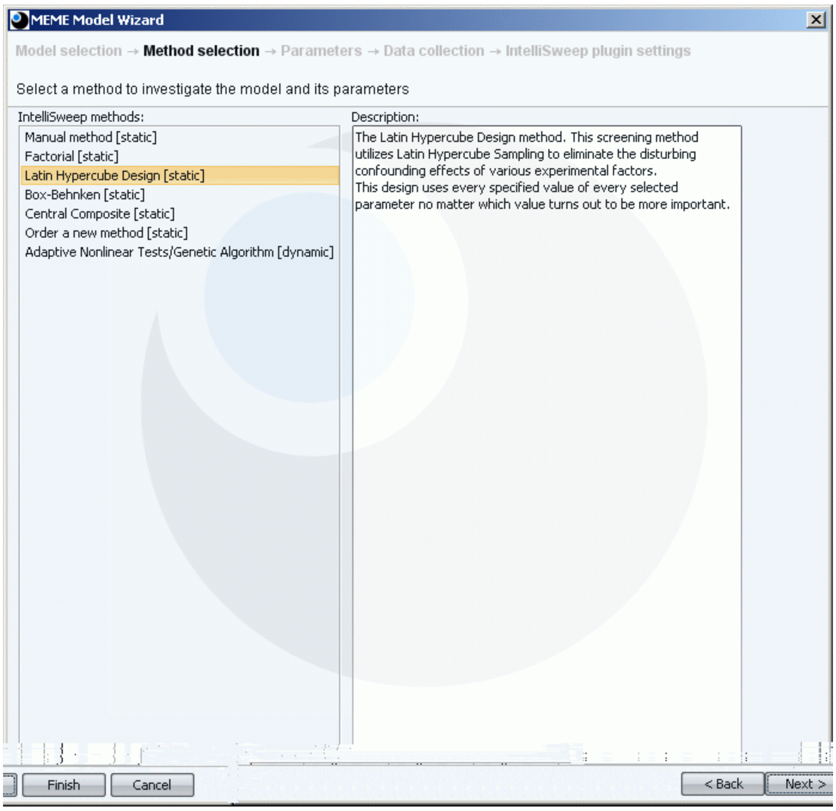
Certain models operate on external files (e.g., text files, XML files, database files etc.) called resources. Running these models on a remote grid or cluster of computers requires the wizard to know the name and location of the resource files. Resources can be defined in a *Resources* dialogue.

When running a model, the wizard automatically saves all model settings (e.g., the parameter file, recorders, scripts, etc. defined at later steps). Previously stored settings can be also loaded at the first step for a repeated experiment or for modification at later steps of the wizard.

The general preferences of the Parameter Sweeper can be also accessed from this page. Among these settings the most important is whether the experiment is to be executed on a single computer (default), on a set of networked computers (locally or on a grid).

Method Selection

After the specification of the model, in the second step of the wizard, the user needs to select an experiment method that fits her purposes best (see Figure 56). The default is the *Manual method*, which lets the user build a parameter tree manually (resulting in a traditional “brute force” or in an OFAT experiment). The other possible choices are Design of Experiments (DoE) methods discussed later. In the method selection dialogue, each method has a short description, including a few lines on its usability and strengths.


MEME Model Wizard
Model selection → Method selection → Parameters → Data collection → IntelliSweep plugin settings
IntelliSweep methods:
Manual method [static]
Factorial [static]
Latin Hypercube Design [static]
Box-Behnken [static]
Central Composite [static]
Order a new method [static]
Adaptive Nonlinear Tests/Genetic Algorithm [dynamic]
Description:
Finish
Cancel
< Back
Next >

Select a method to investigate the model and its parameters

The Latin Hypercube Design method. This screening method utilizes Latin Hypercube Sampling to eliminate the disturbing confounding effects of various experimental factors.
This design uses every specified value of every selected parameter no matter which value turns out to be more important.

Figure 56. The Method Selection page

Manual Definition

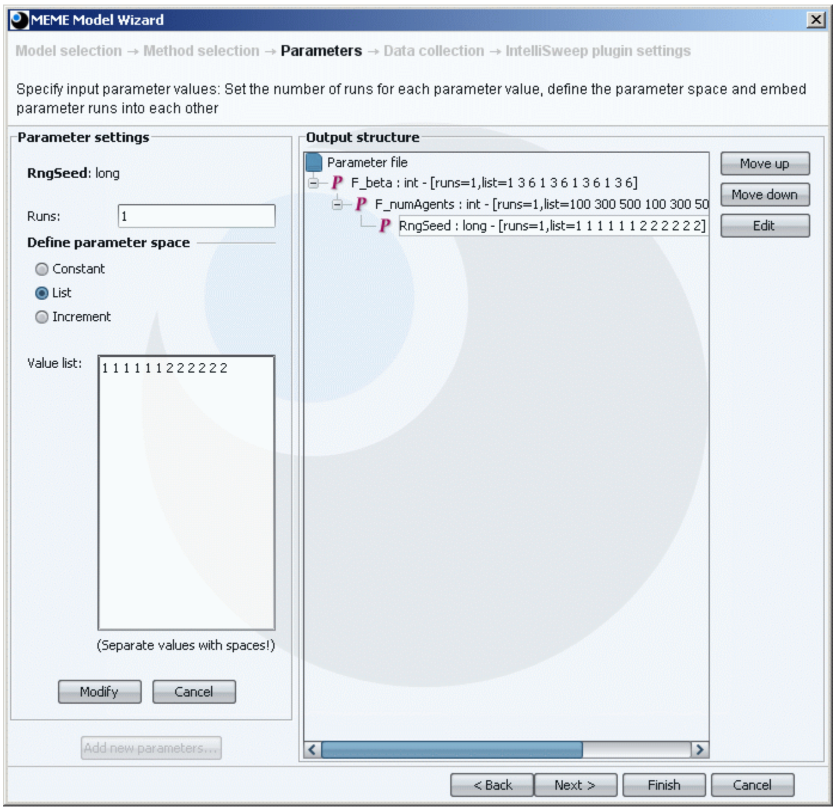
In this section we will assume that the user selected the Manual method. The parameter space the model is to be run on can be defined at the next step of the Parameter Sweeper wizard. The parameter combinations are represented by a tree, as shown on Figure 57. If a parameter is nested in another, *all* combinations of the two parameters’ values will be created. The values of any selected parameter can be specified in three ways (following general conventions):

by specifying a constant value,

by specifying a list of values, or

by specifying an iteration by providing a start value, an end value and an increment.

From the parameter combinations assembled a parameter file is created by the wizard.



Specify input parameter values: Set the number of runs for each parameter value, define the parameter space and embed parameter runs into each other

RngSeed: long

Runs: 1

Define parameter space

○ Constant
● List
○ Increment

Value list: 1 1 1 1 1 1 2 2 2 2 2 2

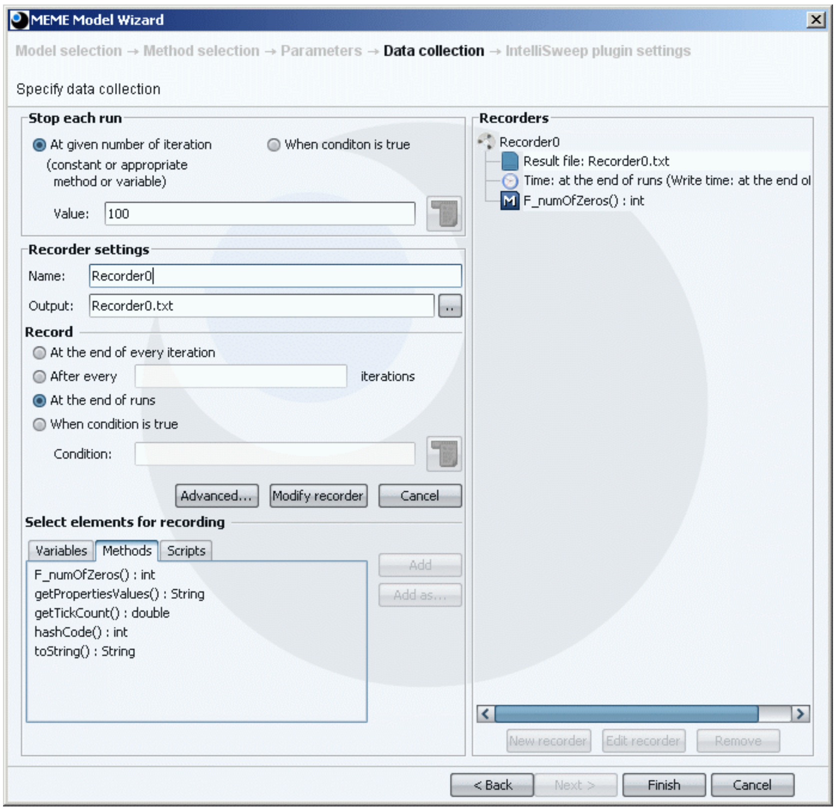
```
Parameter file
⊟─ P F_beta : int - [runs=1,list=1 3 6 1 3 6 1 3 6 1 3 6]
   ⊟─ P F_numAgents : int - [runs=1,list=100 300 500 100 300 50
      └─ P RngSeed : long - [runs=1,list=1 1 1 1 1 1 2 2 2 2 2 2]
```

Figure 57. The Manual Definition page

The Runs panel describes how many times a parameter is going to take each value. This option is offered for reasons of compatibility with old RepastJ 3 parameter files. As discussed in the overview, it is also possible to extend the group of available parameters by flagging a variable as a parameter. This feature is also available in this step.

Specifying Data Collection

In the next step of the Parameter Sweeper wizard is common for all exploration methods. Here the user specifies what data is to be collected during the experiment and sets the stopping condition for simulation runs on the page shown at Figure 58.



MEME Model Wizard

Model selection → Method selection → Parameters → **Data collection** → IntelliSweep plugin settings

Specify data collection

Stop each run

● At given number of iteration (constant or appropriate method or variable) ○ When conditon is true

Value: 100

Recorder settings

Name: Recorder0

Output: Recorder0.txt

Record

○ At the end of every iteration
○ After every ____ iterations
● At the end of runs
○ When condition is true

Condition:

Select elements for recording

F_numOfZeros() : int
getPropertiesValues() : String
getTickCount() : double
hashCode() : int
toString() : String

Recorders

- Recorder0
  - Result file: Recorder0.txt
  - Time: at the end of runs (Write time: at the end o
  - F_numOfZeros() : int

Figure 58. The Data Collection page

The stopping condition can be a number (i.e., the specific time step at which runs will be stopped), it can be a variable with the appropriate type (i.e., an integer), a method with the appropriate return type (integer), or a logical expression built from the model's variables and methods (i.e., in the time step when this condition becomes true for the first time the simulation will be stopped).

The data to be collected during the simulations is organized into *recorders* (following the Repast 3 tradition).

Recorders have the following properties;

name;

name of the output file (one per recorder);

time/frequency of recording (and time of writing data to file in brackets, see details below);

what to record (names of variables and/or methods returning numeric, textual or logical values).

The recorder collects the values of the specified variables and methods during the simulation. There are four time/frequency options for recording data:

Record at the end of each iteration (i.e., at every time step), or

record after every *x* iterations (e.g., at every 10th time step), or

record at the end of each run (i.e., one recording for every defined parameter combination), or

record when the given condition is true.

The condition in the last option must be a logical expression built from the model's variables and methods. The collected values are then eventually saved into the specified output file. In order to speed up simulations, recorders store collected information in memory first and save them to files only at a given frequency.

As discussed earlier, in addition to variables and methods already existing in the model, derivative values (e.g., statistics) can also be specified as the source of information to be collected. This is done using a simple point-and-click interface and/or simple scripting. (The latter option is for expert users, ready to write Java code). In addition, so-called operators can also be managed with the point-and-click interface. These are for assembling and transforming collections of data, to be used as input in one of the statistics or scripts.

Finishing the Data Collection page of the Parameter Sweeper wizard starts the computational experiment.

Simulation Execution

When the experiment is run the local computer, the *Local Monitor* is started, see Figure 59. The monitor displays the following information about the running experiment:

Full name of the model (with the date and time of the generation/start);

Name of parameter file;

Name of the output file(s);

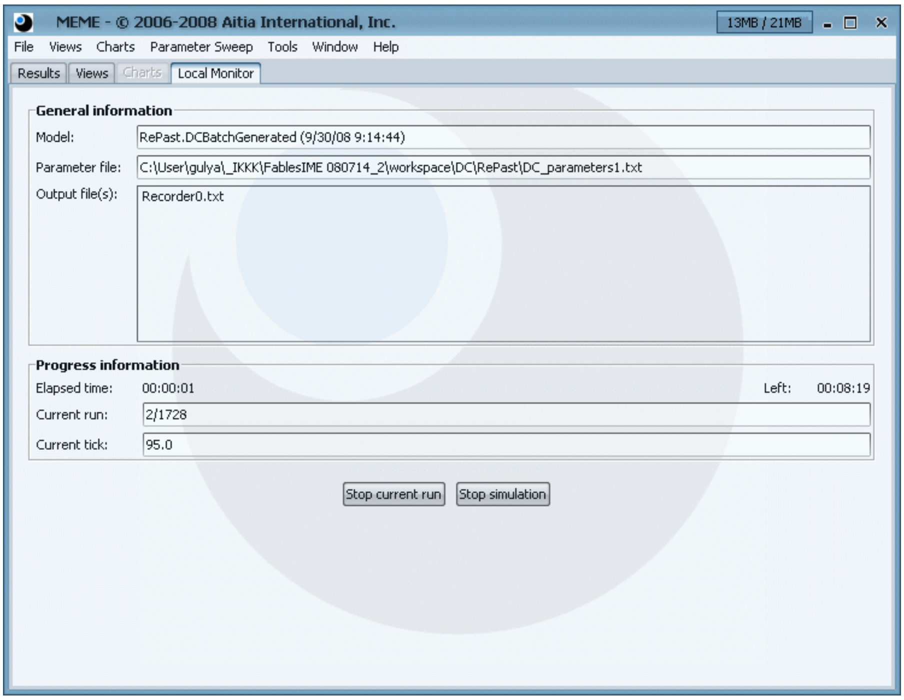
Elapsed time;

An estimate of the remaining time;

The number of currently executing run and the total number of runs, and

The current time step (also called *tick*) in the actual run.

In addition, the monitor offers the stopping of the current run or the entire experiment.



General information

Model: RePast.DCBatchGenerated (9/30/08 9:14:44)

Parameter file: C:\User\gulya_IKKK\FablesIME 080714_2\workspace\DC\RePast\DC_parameters1.txt

Output file(s): Recorder0.txt

Progress information

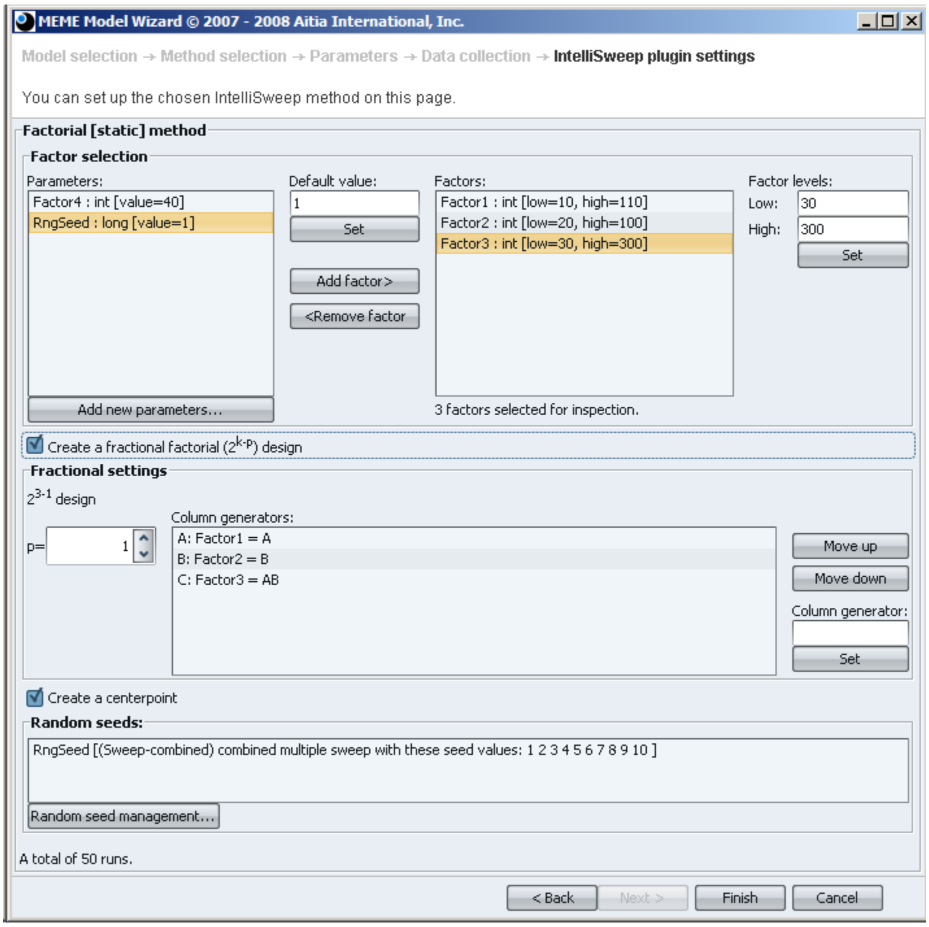
Elapsed time: 00:00:01 Left: 00:08:19

Current run: 2/1728

Current tick: 95.0

Figure 59. The Local Monitor

On the other hand, if the experiment is executed on a cluster of networked computers, a different *Monitor* tool is started. This Monitor enables users to follow the progress of simulations running on remote clusters or grids of computers. It is also capable of downloading and importing the output of finished simulations into the results database of MEME (or of simply downloading it to the local file system). The Monitor collects and displays information about the experiment currently being executed by the simulation server, similarly to the information displayed by the Local Monitor. The full name and description of the experiment is shown, together with the time of upload, the name of the parameter file and those of the output files. Naturally, the progress of the experiment is also displayed. The estimated remaining time, the number of the current run and the total number of runs are shown. The *Last sender host* field contains the name of the computer that performed the last received result.



MEME Model Wizard © 2007 - 2008 Aitia International, Inc.

Model selection → Method selection → Parameters → Data collection → **IntelliSweep plugin settings**

You can set up the chosen IntelliSweep method on this page.

Factorial [static] method

Factor selection

Parameters:
Factor4 : int [value=40]
RngSeed : long [value=1]

Default value: 1

Factors:
Factor1 : int [low=10, high=110]
Factor2 : int [low=20, high=100]
Factor3 : int [low=30, high=300]

Factor levels:
Low: 30
High: 300

3 factors selected for inspection.

☑ Create a fractional factorial (2^{k-p}) design

Fractional settings

2^{3-1} design

p= 1

Column generators:
A: Factor1 = A
B: Factor2 = B
C: Factor3 = AB

☑ Create a centerpoint

Random seeds:

RngSeed [(Sweep-combined) combined multiple sweep with these seed values: 1 2 3 4 5 6 7 8 9 10]

A total of 50 runs.

Figure 60. The Factorial Design page of the Parameter Sweeper wizard

The second tab of the monitor displays information about and gives access to *Finished simulations*. Results of completed experiments can be downloaded from here or imported directly to the database of MEME.

The third tab is for listing *Waiting* simulations. When the user starts a simulation with the Parameter Sweep wizard, the wizard sends a simulation request message to the server application. The server application stores the simulation requests in a queue and runs the simulations one after the other. The earliest request will be served first. (The current version of the Parameter Sweeper is limited to this simplistic method of scheduling.)

12.3.4 Advanced Experiment Methods

In addition to the traditional approaches to simulation experiment design (i.e., one-factor-at-a-time or "brute force" full factorials), the MEME Parameter Sweeper is also built to support advanced experiment designs ("IntelliSweep" methods) as well. These can be selected at the second, Method Selection page of the Parameter Sweeper wizard.

Currently three advanced designs are supported from among the ones discussed earlier at the general introduction to DoE (see below). The rest is under ongoing development and is expected to be completed by the end of 2008. We will shortly refer to other planned designs and methods at the end of this section.

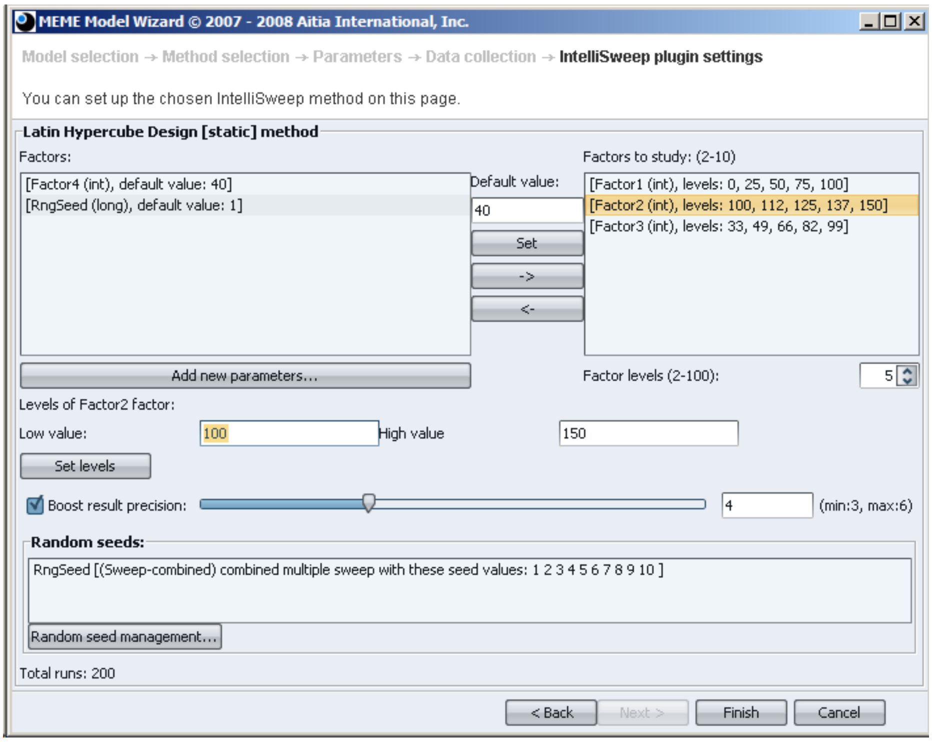
Fractional Factorial Designs

If the user selected this design then at the Method Selection page, the dialog box shown on Figure 60 is presented to her in lieu of the Parameters page discussed earlier.

On this page the user selects the explored factors from model parameters and sets their default, low and high values. New parameters can be created from the model class, similarly to the functionality of the

Manual method page. The factorial design can be converted to a fractional design, by checking the appropriate check box. With this option, the design will only use the ½, ¼, etc. of the original 2^k runs. The plugin automatically determines the appropriate fraction.

If the user decides to add a center point to the design, the plugin calculates its location automatically. Replications for the same parameter combinations can also be specified at the random seed management dialog which is discussed later.



MEME Model Wizard © 2007 - 2008 Aitia International, Inc.

Model selection → Method selection → Parameters → Data collection → **IntelliSweep plugin settings**

You can set up the chosen IntelliSweep method on this page.

Latin Hypercube Design [static] method

Factors:

[Factor4 (int), default value: 40]
[RngSeed (long), default value: 1]

Default value: 40

Factors to study: (2-10)

[Factor1 (int), levels: 0, 25, 50, 75, 100]
[Factor2 (int), levels: 100, 112, 125, 137, 150]
[Factor3 (int), levels: 33, 49, 66, 82, 99]

Factor levels (2-100): 5

Levels of Factor2 factor:

Low value: 100 High value 150

☑ Boost result precision: 4 (min:3, max:6)

Random seeds:

RngSeed [(Sweep-combined) combined multiple sweep with these seed values: 1 2 3 4 5 6 7 8 9 10]

Total runs: 200

Figure 61. The Latin Hypercube Design page of the Parameter Sweeper wizard

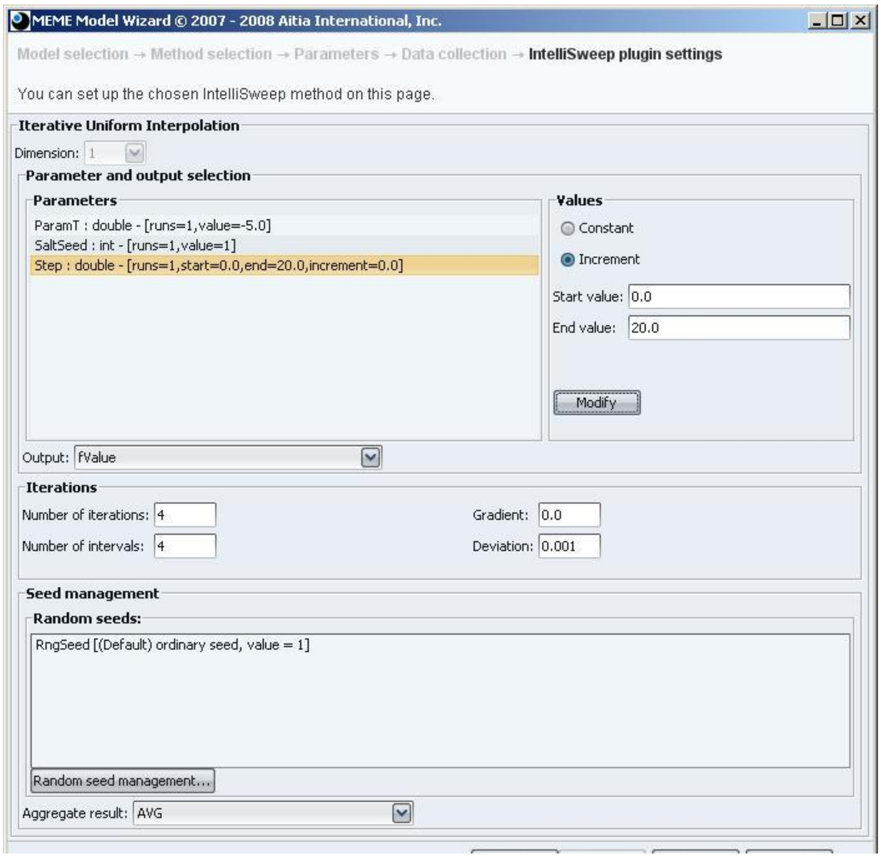
Latin Hypercube Designs

If the user selected this design then at the Method Selection page, the dialog box shown on Figure 61 is presented to her instead of the Parameters page discussed earlier.

The Latin Hypercube method produces designs where all chosen parameters have the same number of levels, and every parameter level is used only once in the design. So the number of runs is defined by the number of parameter levels in the basic case. The coordinates of the points in the design were obtained from published space-filling designs (Tilburg University). These are approximated maximin designs that attempt to maximize the minimum distance between the design points.

Similarly to that seen on the Fractional Factorial Design page, the user selects the explored factors from the parameters and sets their high and low values. She also sets the number of levels to examine (which is the same for all factors in this design). The method will create equidistant design points in the hypercube specified by the low and high values of the factors. The *Boost result precision* slider is an extra feature, which tries to make the estimations of the design more accurate. It creates new permutations of the design’s dimension-parameter associations to create additional design points. If k factors are selected to study, then there are k! possible permutations. Notice that creating all of them will yield a full factorial design and thus a rather inefficient experiment.

This plugin also supports the replication of simulations with the same parameter combinations via the pseudo random number generator seed management function, discussed later.



MEME Model Wizard © 2007 - 2008 Aitia International, Inc.

Model selection → Method selection → Parameters → Data collection → **IntelliSweep plugin settings**

You can set up the chosen IntelliSweep method on this page.

Iterative Uniform Interpolation

Dimension: 1

Parameter and output selection

Parameters

ParamT : double - [runs=1,value=-5.0]
SaltSeed : int - [runs=1,value=1]
Step : double - [runs=1,start=0.0,end=20.0,increment=0.0]

Values

○ Constant
● Increment

Start value: 0.0
End value: 20.0

Output: fValue

Iterations

Number of iterations: 4
Number of intervals: 4
Gradient: 0.0
Deviation: 0.001

Seed management

Random seeds:

RngSeed [(Default) ordinary seed, value = 1]

Aggregate result: AVG

Figure 62. Iterative Uniform Interpolation plugin settings page

Iterative Uniform Interpolation

In addition to static experiment designs, the MEME Parameter Sweeper is also designed to support dynamic, iterative methods as well. Among these currently only the *Iterative Uniform Interpolation (IUI)* method is available (Figure 62).

The IUI plugin implements a simple response analyzer method, designed to map the response of the simulation in a given parameter range. (The current version of the plugin only supports one-dimensional mappings.) The idea is to automatically detect “interesting” regions of the parameter space, based on initial assumptions about the response function (e.g., linearity, specified slope, etc.).

To achieve this, the IUI method starts with an equidistant sampling of the relevant parameter interval and interpolates the selected output values (model variable, return value of a function, etc., as with other methods) of the model as a function of a chosen parameter. The interpolations are then compared to earlier assumptions. If between two sampled points the interpolated function appears to be significantly different from earlier expectations, then the IUI methods applies itself recursively to the interval in question.

For setting up an IUI experiment, the user needs to specify the starting interval of the parameter to be explored, the maximum number of analyzed subintervals (per iteration) and a limit on the depth of recursions. The initial assumptions about the response functions also need to be specified, by giving an expected gradient and an allowable deviation from it.

The Iterative Uniform Interpolation can reduce the number of sampled points, while approximating the response function well – compared to the brute-force parameter sweeps. (See Figure 63 for an example. Red dots denote the measurement points. Notice their rarity at the upper, linear interval of the function.)

The IUI is also able to handle replications (to be set up via the random seed management panel discussed next). Replicated measurements belonging to the same parameter (factor) value are aggregated (e.g., using averaging, minimum or maximum operations) the response's interpolation are based on these aggregated values.

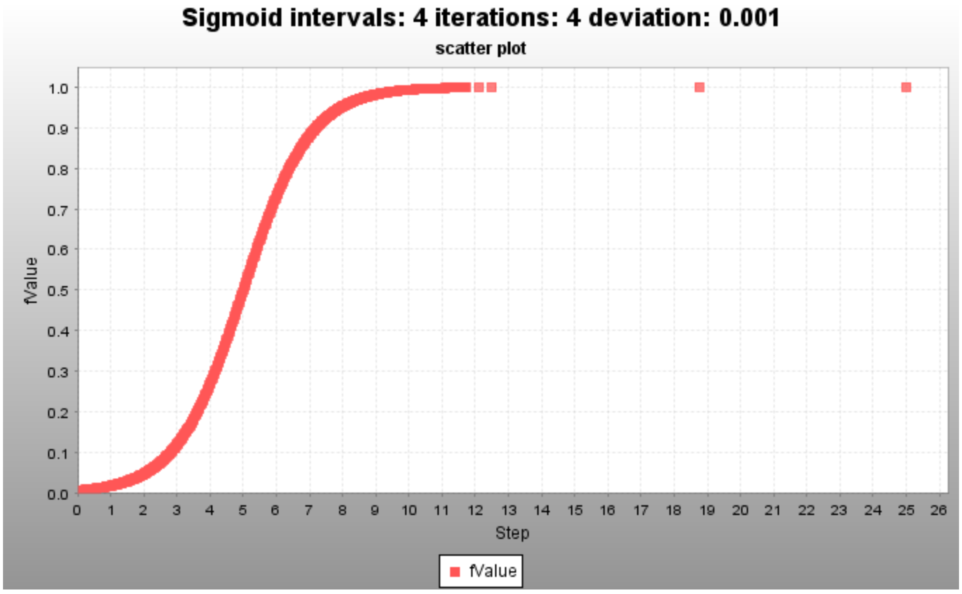
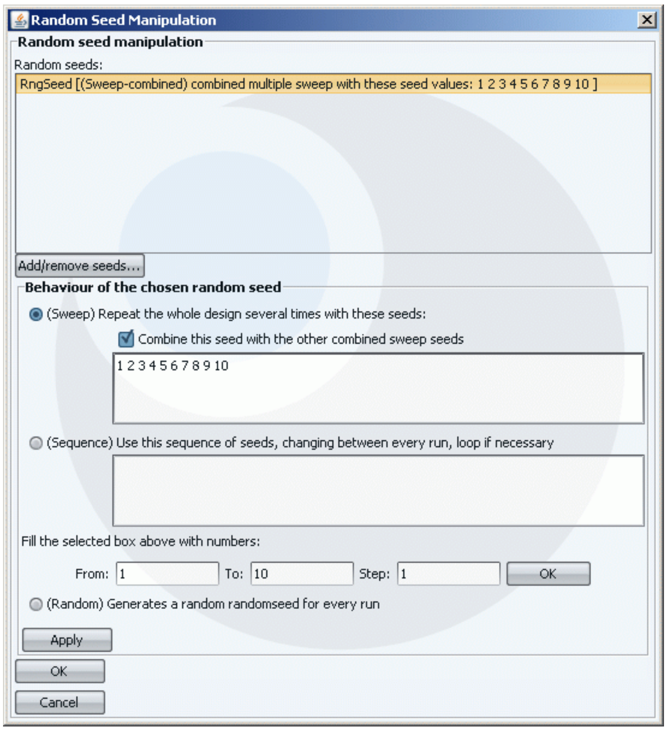


Figure 63. Example output of the Iterative Uniform Interpolation method. Red dots denote the measurement points.



Random Seed Manipulation

Random seed manipulation

Random seeds:

RngSeed [(Sweep-combined) combined multiple sweep with these seed values: 1 2 3 4 5 6 7 8 9 10]

Add/remove seeds...

Behaviour of the chosen random seed

● (Sweep) Repeat the whole design several times with these seeds:

☑ Combine this seed with the other combined sweep seeds

1 2 3 4 5 6 7 8 9 10

○ (Sequence) Use this sequence of seeds, changing between every run, loop if necessary

Fill the selected box above with numbers:

From: 1 To: 10 Step: 1 OK

○ (Random) Generates a random randomseed for every run

Apply

OK

Cancel

Figure 64. The Random Seed Management panel

Pseudo Random Number Generator Seed Management

In computational simulations stochasticity is implemented using pseudo random number generators (Zeigler, 1976). This means that replications must be handled by the selection of the values for the seeds of these generators. The Parameter Sweeper's DoE plugin architecture is equipped with a wizard (shown on Figure 64) that is available to all DoE plugins and which provides means for the flexible handling and setting these seeds.

The random seed management panel provides access to all model parameters. The user can flag parameters of discrete number types as random seeds. The behavior of the flagged seeds must also be selected from among several following options, including randomized values, all combinations (in case of multiple generator seeds), simultaneous stepping of values (also for multiple seeds), etc.

Design Specific Estimations and Charts

In the previous sections we summarized the DoE plugins currently available for the MEME. However, the Parameter Sweeper tool's DoE architecture offers another very important and convenient feature for all DoE plugins.

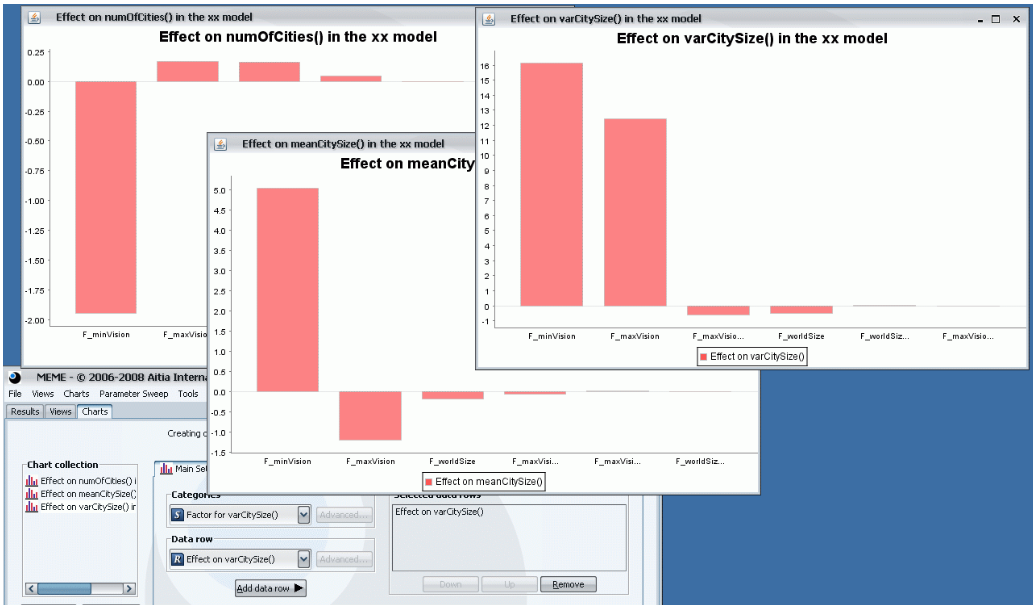


Figure 65. Effect charts automatically generated for (fractional) factorial designs

Since experiment designs are created for particular objectives and formulate specific assumptions about the systems, they typically imply a special way of processing, charting and assessing their results as well. Therefore, the tools DoE architecture supports the automatic post-processing of simulation results collected during IntelliSweep experiments (i.e., using one of the DoE plugins). All DoE methods currently available in MEME and discussed so far take advantage of this option. This means that when their results are imported to MEME, special scripts designed for the outputs of such types of experiments are executed. These automatically create informative views of the data set, together with the most common charts used in connection with the particular design (Figure 65). Naturally, the raw data set also remains to be available in MEME, allowing for the further, more ambitious processing of the collected results.

Further Methods Planned

In addition to the methods and design discussed so far, in the longer run we are also planning to implement parameter sweeper plugins that apply heuristic methods borrowed from Artificial Intelligence techniques.

In particular, we plan to develop a Genetic Algorithm-based method for the optimization of response values (Holland, 1975). More importantly, we plan to extend this plugin and convert it to a plugin for *Active Non-linear Tests (ANTS)* (Lambrecht et al., 1998). The essence of the latter technique is the "automatic falsification" of theorems derived from simulation results. The concept is to formulate the candidate theorems in terms of the response variables and then define a *metric* that yields a numeric value for the distance between the response of the actual run and the formulated theorem. (Naturally, this metric is to be defined by the user, i.e., the modeler, probably in the form a few lines worth of scripting.) Once this metric is defined, a Genetic Algorithm can be used that attempts to *maximize the above distance*, thus trying to find the *maximum error* between the theorem formalized in advance and the simulations output, in terms of the metric introduced.

12.4 Conclusions

This document overviewed the options for the methodologically correct execution of computational simulations at the large scale. It discussed options for the distributed exploration of the parameter space and overviewed advanced statistical designs for the non-naïve sampling of the response surface. The application of these methods is within reach for the general social simulation community, using the newly developed user friendly software tools that were also briefly introduced and discussed in the chapter. These tools offer a wide array of complex and advanced features to support the effective and methodologically sound simulation execution.

Chapter 13 Demonstrating the Theory: An Introduction to the Scenarios

Klaus G. Troitzsch

Abstract

This section will give an overview of the scenarios – applications of the EMIL-A theory – so far implemented in EMIL-S and argue why these scenarios have been selected to test the EMIL-A theory for applicability. It will also analyse those features which are similar and which are different among the scenarios and also give an account of empirical validations where these have been possible so far, even if not for all scenarios empirical validations are possible and/or fruitful such that in some cases only validations against stylised facts are presented.

13.1 Why These Scenarios were Selected

This and the following four chapters contain the documentation of four scenarios using EMIL-S for modelling and simulation. Some of them use TRASS, others Repast as the simulation engine representing the physical environment of the agents, some of them use MEME for simulation execution and output collection and analysis.

The selection of these scenarios goes back to several criteria. The traffic scenario, which is the first to be discussed, was selected for the reason that it just discusses some everyday behaviour where it is not necessary to give it any sophisticated theoretical background. On the contrary, it is quite appropriate to explain the central features of EMIL-S and its necessary interfaces to a simulation engine representing the physical environment of the agents of the scenario and to make all these features more graphic. Thus we found it would be a good scenario for introducing these features to the readers without burdening them with a theoretical and/or empirical background which would have to be explained in much detail.

The second scenario is one of the core scenarios of the EMIL project, as it is devoted to represent "the rise of collaborative community norms in the Open Source community where new conventions have established and new norms are being invoked" [Proposal, p. 6]. Although the Wikipedia example is not an example from the Open Source software development community, it serves as a better replacement, as the data are much more easily accessible, as not only the Wikipedia is public, but also most of the discussions about articles, as the empirical analysis in Chapter 5 has shown. It was, of course, necessary to simplify the simulated community of collaborative authors or developers, as the EMIL-S agents (and agents in all other multi-agent simulation applications) do not have natural language at their disposal with the help of which they could write articles about the world (and moreover, software agents do not even have access to the world outside their computer, and the environment within the their computer is usually too scanty to write a Wikipedia about it). And the same would have applied to a simulation model of collaborative software production — simulated agents would have never had a chance to produce anything with some similarity to some software product, whereas the texts produced in the Wikipedia simulation at least produce text corpora which can be evaluated with respect to style and consistence.

The scenario of micro-finance groups also has a sound empirical background which goes back to the empirical work of Lucas dos Anjos et al. (2008a). It is the first scenario which contains two levels of actors, namely the individual group members and the bank as individual agents and the group as a corporate agent which — as will be discussed in the respective chapter — is not a normative agent (and thus not modelled with the agent designer of EMIL-S, but it is just a simple Repast agent which does not much more than counting the votes of group members and forwarding bank messages to them.

The fourth scenario ("multiple contexts") brings two types of agents together and into interaction, namely one group of agents which have the relatively rich cognitive structure as discussed in Chapters 8 and 10 and another groups of simpler agents which are only social conformers without any internal compass —perhaps these are similar to Riesman's "other-directed" people as compared to the "inner-directed" people

(Riesman, 1950). This scenario comes in several different versions, one just comparing one run with only social conformers and one with norm abiders, while another version mixes these two populations. Another interesting feature is that in this scenario agents meet and interact in different context, perhaps (but not necessarily) learning in one context what might prove useful in another context.

13.2 Similarities and Differences

All four scenarios are similar in so far as they contain a medium number of agents capable of communicating, of deliberating and of making and executing decisions. And, as a matter of course, all agents — with the sole exception of the social conformers of the fourth scenario and of the collective actor of the micro-finance scenario — have the same software architecture of EMIL-S as derived from the logical and cognitive architecture of EMIL-A discussed in Chapters 9 and 10. This shows, by the way, that both EMIL-A and EMIL-S seem to be applicable to a wide variety of scenarios (although there is no proof that the EMIL concept is applicable to all kinds of social systems).

It goes without saying that all scenarios lead to some emergent phenomena. Perhaps this is most visible in the simple traffic scenario where the "striped area" in a street acquires a meaning for both car drivers and pedestrian, as after some time they agree that this is the optimal place for pedestrians to cross a street (for no other reason than that it is distinguishable from the rest of the street). In the other three scenarios, the emergent phenomenon is the convergence of individual behavioural regularities or individual normative beliefs to a norm explicitly adopted by the whole community (but still with the possibility that this norm — or these norms — are sometimes violated). Those versions of some of the scenarios are of special interest where the individual normative beliefs which the agents have at the beginning of the simulation run are differing — here it is not foreseeable which of the individual normative beliefs prevails in the end.

Not in all scenarios do agents have a choice among normative beliefs — a norm commanding car drivers to hurt and kill pedestrians would lead to the extinction of the latter, and a norm commanding or even only allowing loaners and loaner groups not to pay their loans back to the bank or a norm commanding banks to waive the loans would lead to the breakdown of the micro-finance system. On the other hand, it does not really matter whether collaborative writers agree in the end on writing long or short sentences in their articles (or on using long or short words, or on obeying or violating word building rules such as the vowel harmony of certain languages — in the latter case they would change their common language, but this would not impair their communication).

But even in the former case (where the whole system would not be operable at all when the "wrong" norm is applied) it remains interesting that the normative beliefs of some individuals prevail in the end without a long evolutionary process lasting many generations of car drivers and pedestrians or of banks and loaners.

Another difference — between the micro-finance and the other three scenarios — is the occurrence of a collective agent in — and only in — the micro-finance scenario. This collective agent posed some difficulties to its modelling (as will be discussed in the respective chapter), as the collective agent is not endowed with any deliberative or even cognitive capacity (a group can only speak through its speaker, and the metaphorical "group learning" consists of individual learning and of, say, structural changes within the group that have to be designed by the individual group members, as structural changes within the group which are not designed explicit do not deserve the description of "learning"). This is why the collective agent was designed as nothing more than an agency which forwards and counts messages of individual agents and sends the results back to them without any other kind of processing, let alone interpretation (one could have thought of a secretary agent of the group whose it would have been to take the minutes of group meetings — but either this secretary agents reports exactly what was decided in a group meeting, but then it performs exactly as the non-cognitive agent (much like a tape recorder), or this secretary unfaithfully counterfeits its reports — but then it is an agent with a particular normative belief, and this would be an extension of the model.

The last difference worth mentioning is the one between the multiple contexts and the other three scenarios. Not only do we have the comparison between and/or the concourse of normative agents and non-normative agents (social conformers), but these agents meet and interact in different context where they have to take care of the needs of other agents and can learn how to behave properly in one context and can apply what they learnt in another context.

And finally, two of the scenarios need a topographically described environment for their interactions, while in the other two scenarios the topography within which the agents interact does not really matter (collaborative writing happens on the same blackboard or internet for all agents, and whether the loaners meet personally or just exchange their opinions in written form or by telephone would not make any difference in the real world.

13.3 Empirical Validations if These are Possible and / or Fruitful

As in all simulation models, empirical validation of results is highly desirable. In the four scenarios the best that can be achieved is a comparison between the model output and stylised facts (Kaldor, 1961/1968).

In the case of the traffic scenario, readers will be able to compare the simulation results (which are also available as videos) with their own ideas how children could learn how to cross a street properly. Another similar simulation (not reported in this report), which models the transition to a strict rule to use the right-hand (in some countries: left-hand side) of the street for car traffic, the situation is the same. The target systems for both primordial situations do not exist any longer as the relevant norms (traffic laws) were already passed decades ago, even youngest children learnt them, and nobody would dare to start the necessary experiments (having children move between two meadows and cross the street at the danger of being run down by careless or unskilled car drivers, or having car drivers decide on their own which side of the street to use) as these experiments are much too dangerous for all participants.

In the Wikipedia scenario comparisons are possible, albeit restricted due to the fact that the contents of real world Wikipedia articles and, consequently, of the discussions about these articles is much richer that in a simulated scenario — as will be discussed in the introduction to the scenario description and in the final chapter of this section.

The case with the micro-finance model is more promising as the field study (Lucas dos Anjos et al., 2008a) was the starting point of the development of the simulation model (or: models, as there was a Prolog predecessor of the EMIL-S version). Thus it will be possible to discuss the comparison between target system and simulation model(s) in the respective chapter.

Finally the case of the multiple-context scenario is similar to the case of the traffic scenario — wherever people are involved in waiting and service processes, they are already influenced by their cultural initiation, such that from the very beginning a large majority of the people present in such a simulation behaves the same because all of them have already internalised the commonly accepted norms (and the others are norm violators per-se who would always defect). Thus readers will have to content themselves with a comparison between the model results and stylised facts which again here are rather fictitious (but, as (Solow, 1970) remarked about stylised facts, “there is no doubt that they are stylized, though it is possible to question whether they are facts.”).

Chapter 14 Demonstrating the Theory 1: Traffic

Ulf Lotzmann

Abstract

The traffic scenario dealt with in this section is one of the stylised fact scenarios, as the implemented simulation models reflect what the ordinary car driver or pedestrian will have in mind when describing and analysing the (normative) behaviour of traffic participants. This section shows several simple scenarios in which software agent learn from experience and observation which kinds of behaviour in street traffic are dangerous and / or unwelcome (as dangerous behaviour will not always need to be explicitly forbidden, and as sometimes behaviour is explicitly forbidden – or at least blamable – even if no danger arises from this behaviour).

Main purpose of the chapter is, however, to demonstrate the usage of EMIL-S as well as the interplay of EMIL-S with external simulation tools. It is preceded by an introduction to TRASS, the Traffic Simulation System used for the physical simulation layer.

14.1 Introduction: Target Description

The field of microscopic traffic simulation is dominated by agent-based approaches which are focusing on precise mathematical modelling of physical parameters. In many cases, the interaction between traffic participants is restricted to measuring and keeping distance with the aim to reproduce realistic traffic flows while avoiding collisions (e.g. in classical car-following models). Psychological aspects of traffic participants as well as social relations between them are much less treated in current approaches, although these are crucial factors for the dynamics of any real traffic system in which usually various kinds of traffic participants (drivers, cyclists, pedestrians …) appear. In particular, social capabilities are the key for integrating differing perspectives (of distinct kinds of traffic participants) on a joint event.

Normative behaviour and learning introduce social capabilities for agents in traffic simulations. The following model is a step toward this more comprehensive view on traffic systems.

The inclusion of this model in the report has an additional purpose: Due to the fact that it can be taken for granted that basically all potential readers are familiar with traffic matters in everyday life, it seems beneficial to call on such kind of scenario for the technical introduction to handling and integrating EMIL-S.

14.2 Scenario History: Earlier Models of Norm Emergence in Traffic Simulation

The initial idea to involve traffic scenarios was born in 2006 after first experiences with the newly created traffic simulation tool TRASS showed that with such models also social simulations can be visualized and, thus, be made more descriptive. The first prototype was a replication of the famous Weidlich-Haag opinion formation model (Weidlich and Haag, 1983), transferred into a fictitious scenario in which car drivers repeatedly change from left-hand to right-hand driving and vice versa in order to avoid collisions on a shared two-lane road.

Results from the prototype development (as well as progress with the realizing of mixed traffic participant scenarios) not only led to further development of the TRASS system towards a practicable traffic simulation system, but also encouraged the developers to take it into consideration for normative simulation scenarios. Some results are summarized in this chapter.

The focus of such an application is not primarily the exploration or analysis of serious social phenomena, but to show the feasibility of a normative approach realized in EMIL-S. For this reason there is no validated theoretical model behind the traffic scenario, but the model assumptions are derived from everyday experience.

14.3 Scenario Implementation

The traffic scenario (Lotzmann et al., 2008) consists of a simple topography (see Figure 72), which is composed of a straight one-way road and two meadows to the left and right of the road. A small segment of the road has a special mark (much like a crosswalk). Situated within both meadows, a number (which is constant during a simulation run) of pedestrian agents move around. From time to time each pedestrian gets an impulse to reach within a given period of time a target point on the opposite meadow. For this activity, the agent can choose between the direct way to the target or a detour via the crosswalk. The road is populated by car agents who are aimed at reaching the end of the road at a given point of time.

For both types of agents, the deviation from the permitted duration leads to a valuation of the recent agent activity: a penalty when more time was required and accordingly a gratification when the target was reached earlier.

Due to the interaction between agents, occasional collisions are likely to happen. Such an event, when occurring between a car and pedestrian, is classified as undesirable. Observations of a collision provoke other agents to issue sanctions against the blamable agents. The extent of the sanction is determined by various factors reflecting the environmental situation (e.g. the road section where the collision occurred) and the normative beliefs of the valuating agent (e.g. a collision on a crosswalk might result in a harder sanction than on the rest of the road). Sanctions lead to a temporary stop of motion for the involved agents. Hence, to avoid sanctions is a competing goal to the aforementioned aims (reaching the target point or end of the road, respectively, in due time).

To map this informal concept into the EMIL-S scheme, a classification of the expected information transfer to message types is necessary in a first step. Every agent is able to perceive its environment and to conduct actions in order to adjust the (own) parameters of motion or perception, respectively. These (agent internal) processes can be mapped to message modals “assertion” and “behaviour”. In addition, agents can send and accordingly receive messages to/from other agents. The contents of the messages are different kinds of notifications (like positive or negative valuations and sanctions, e.g. “admonish” or “honk the horn”…). While these message exchanges are either intra-agent matters or speech acts between exactly two agents, another agent property is important for a norm formation process. This is the ability to listen to the communication of other agents in order to gain information about the experience of those agents, and to learn from this information.

With regard to this message classification, rules for the specification of agent behaviour have to be defined in a second step. The structure of a rule follows an event-action scheme where events trigger actions from an action set with certain probabilities. In this model, all events are coupled with received messages and most actions are expressed by message sending activities. Furthermore, additional actions for learning have to be defined.

All rules are constructed with the help of event-action trees. Figure 66 shows a sample event-action tree for a car driver agent.

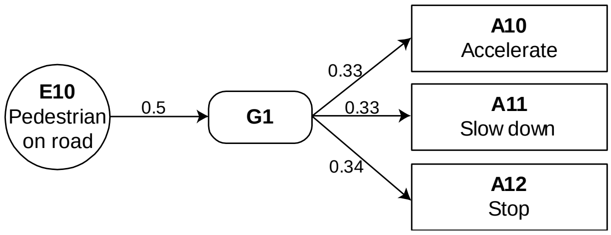


Figure 66. Event-action tree with environmental actions

While the structure of the scenario-specific (initial) event-action trees is static, the selection probabilities may change during the simulation in a learning process. Furthermore, more complex rules emerge by relating several event-action trees to each other. With these behavioural rules, norms emerge as soon as the majority of the population starts to use the same rules with similar probability distributions.

According to the EMIL-S architecture, the traffic model consists of a physical layer implementation based on the TRASS framework, and the related EMIL-S configuration. Both parts are described in the following subchapters.

14.3.1 Physical Layer

The TRASS Software

The TRAffic Simulation System (TRASS) is a framework designed mainly for spatial agent-based simulations. It consists of several software components, which can be used and extended in various ways.

The simulation core incorporates scheduler and communication infrastructure, topography and physical agent model, and features the following key properties:

- continuous space (with a full set of geometric algorithms and tools to handle corresponding topographies);
- discrete time (controlled by a scheduler component);
- multi-threaded architecture allowing parallel simulation execution on multi-processor hardware;
- message-based inter-agent communication with single and group recipients, enabling the agents to exchange symbolic messages;
- mediator-based communication infrastructure makes an extension for distributed execution possible.

The TRASS core does not bring any visualization capabilities or graphical user interface. It defines a programming interface that allows for the integration of external components. The interface consists of sections for:

- definition of agent types with specific behaviour by writing Java code;
- XML based configuration of simulations;
- access to the internal data structure of the simulation kernel for collecting attribute values while executing a simulation;
- administration of simulation runs.

In order to offer a complete and concerted interface for user interaction, an Integrated Development Environment has been developed. This software incorporates features like topography designer, agent modeller as well as a runtime environment for simulation execution and visualization. Figure 67 shows the component structure of the TRASS system.

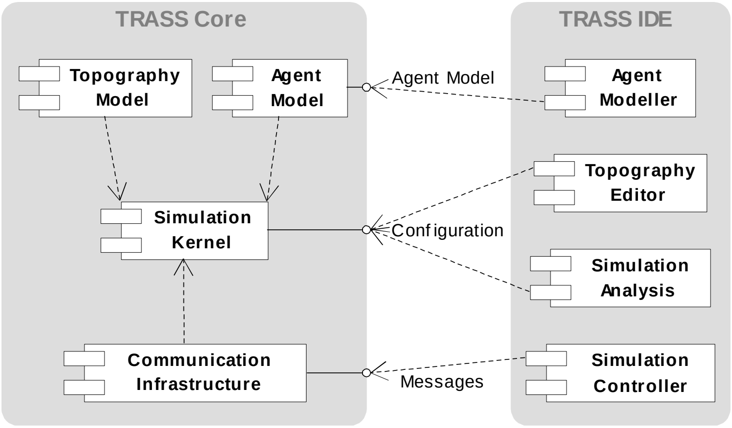


Figure 67. Component diagram of TRASS

The TRASS Agent Model

An agent model describing a traffic participant has to consider further aspects besides the deliberation capabilities. Especially the complex interaction between a traffic participant and the environment is of importance. In other words, the AI potential needs a fundament of physical and technical capabilities.

The different aspects of the agent model for traffic participants are structured in three distinct layers: the physical layer, the robotic layer and the AI layer (Figure 68), together constituting an “embodied” agent.

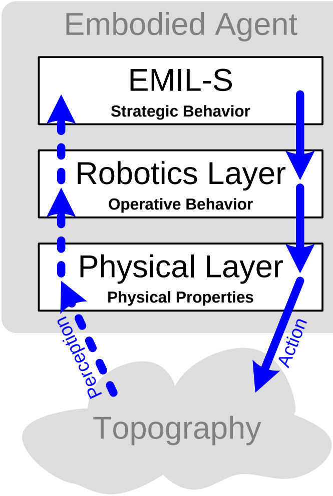


Figure 68. Layers of the TRASS agent model

The **physical layer** describes the abstraction of the static properties and the parameters of motion. Thus it defines the physical appearance of the agent within the artificial environment.

This layer constitutes the interface to the spatial environment defined by the topography model. It is obvious that the physical layer and the topography model must be coordinated. For this reason both model descriptions are part of the TRASS core framework.

The topography model is a two-dimensional continuous space that can be split into districts and that is structured by a mesh of polygon-shaped regions. Each region describes one distinct area of the topography, such as a lane of a road section, a crossing or building. The corresponding physical layer of the agent model is composed of five elements, which are in detail:

- The shape of the agent is constituted by an arbitrary number of circles (with different radii) approximating the outline of the entity to be represented. The usage of circles is due to performance issues when executing necessary calculations during simulation runs (e.g. distance between agents).
- The sensor unit is constituted by an arbitrary number of (various) circle sectors, defining the zones where an agent is able to perceive the environment.
- The communication area is similar to the sensor unit in terms of the geometrical shape involved. It is used to define zones in which an agent can appear for other (observing) agents in different manners. For example, a traffic light agent sends its information (red, yellow or green light) in one distinct direction, affecting only car agents arriving from one distinct way. For any other agent the traffic light is just an obstacle.
- The reference point defines the current position in the two-dimensional world.
- Attributes of motion are velocity and direction of motion.

The **robotics layer** fills the gap between the “low-level” attributes of the physical layer and the abstract strategic decisions treated by the AI layer. Thus, the design of the robotic layer is of utmost importance for the entire system.

The name “robotic layer” is referring to the methods that are used here to implement the transformation process: As in many autonomous robot systems, a form of enhanced finite state machine (Hopcroft and Ullman, 1979) controls the parameter changes depending on the active state and triggered by incidents perceived in the environment (resulting in reactive behaviour) or generated by the AI layer (resulting in proactive behaviour).

In the robotic layer a related concept of different “levels of mind” is used where each level corresponds with a stage in a nested hierarchy of automata. The following levels are considered here:

- basic actions which humans execute “automatically” without thinking (e.g. turning the steering wheel);
- elementary activities (composed of basic actions) which are conducted intuitively by humans (e.g. hold the centre of a lane);
- complex activities as sequences of elementary activities where a human’s attention is required (e.g. lane change operation).

In order to show “intelligent” – and thus realistic – behaviour, the **AI layer** of an agent model for traffic participants requires a control process that is comprised of the following activities:

- Perceive the environment containing geographical elements and neighbouring agents.
- Compose and memorize an agent-individual view of the simulation world based on the perceived environment.
- Communicate with a subset of the agent population.
- Reasoning about different alternatives to act in order to achieve predetermined goals on base of the current environmental state and the agent memory.
- Execute “physical” action based on the capabilities offered by the robotic layer (e.g. driving a curve or changing a lane).
- Estimate the success of action, leading to revaluation of agent memory.

While most of the issues mentioned in the list above are of a more or less technical nature in terms of processing and memorizing perception data and adjusting of parameters, the subject of “reasoning about alternatives” in combination with “estimating the success of action” constitute the actual AI capabilities.

In the traffic scenario, the AI layer is taken over by EMIL-S. For this purpose the robotic layer must be extended in a way that allows the creation of relevant events from aggregated perceptions, and the triggering of (physical) actions according to the “instructions” received from EMIL-S.

The specification of the interface between EMIL-S and a general simulation tool (in this context TRASS) can be seen in Figure 69. To realize an implementation towards the interface the following activities have to be performed:

- Initializing EMIL-S during the model initialization by invoking the static method `initializeController(XMLFile)` of the EMIL-S Controller singleton class with an appropriate parameter file. This can be done within the Model class of the simulation tool (that is connected to the scheduler component).
- The agent classes at the tool layer must implement the `IEMILAgentWrapper` interface and override the method `sendMessage(action)` that is used by EMIL-S to send response messages to the physical agent.
- When instantiating an agent object on the tool layer the corresponding EMIL-S layer object must be also created by invoking the method `addAgent(agentObject)` of the EMIL-S Controller.
- During simulation runs the physical agent sends event messages to the EMIL-S object by invoking the `IEMILAgent` method `processMessage(event)`.

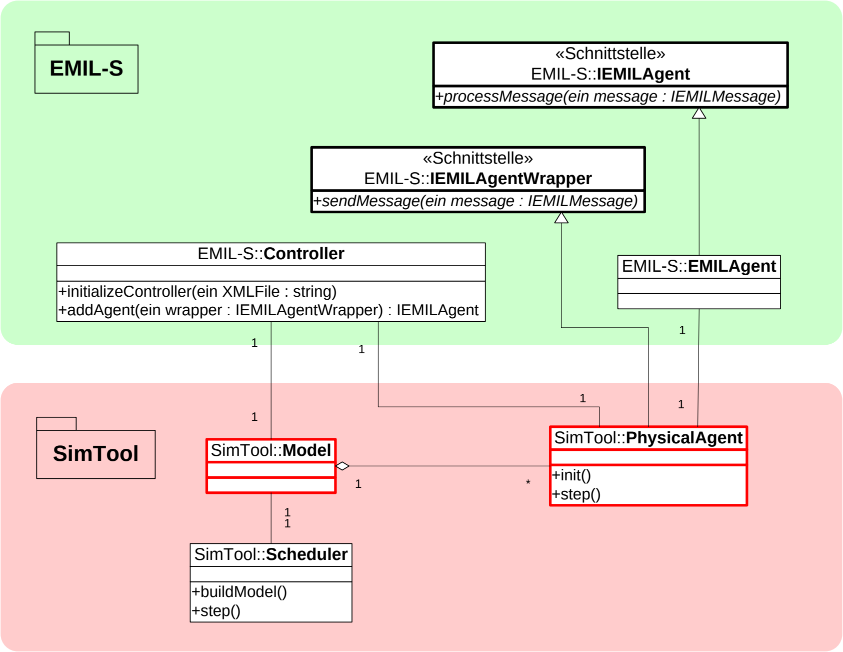

EMIL-S
«Schnittstelle»
EMIL-S::**IEMILAgent**
+processMessage(ein message : IEMILMessage)
«Schnittstelle»
EMIL-S::**IEMILAgentWrapper**
+sendMessage(ein message : IEMILMessage)
EMIL-S::**Controller**
+initializeController(ein XMLFile : string)
+addAgent(ein wrapper : IEMILAgentWrapper) : IEMILAgent
EMIL-S::**EMILAgent**
1
1
1
1
1
1
SimTool
SimTool::**Model**
SimTool::**PhysicalAgent**
+init()
+step()
1
*
1
1
SimTool::**Scheduler**
+buildModel()
+step()

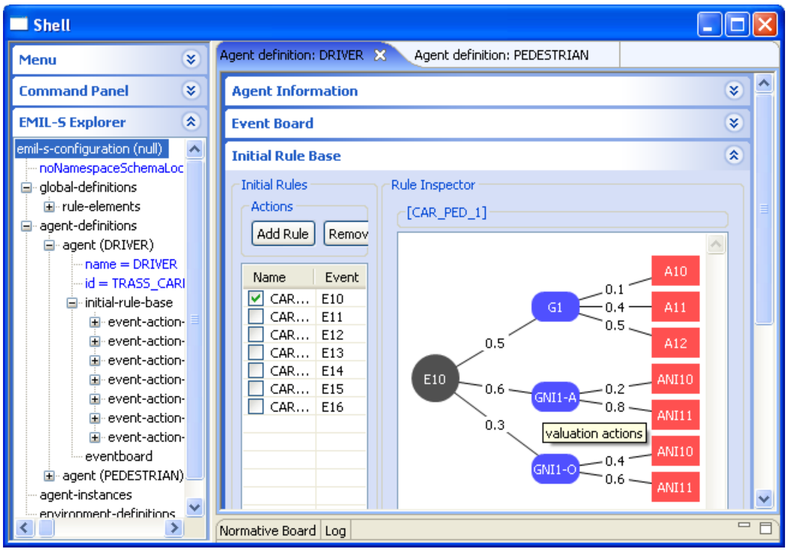
Figure 69. Interface between EMIL-S and simulation tool

14.3.2 EMIL-S Layer

For specifying the simulation model at the EMIL-S layer, three major steps must be taken:

1. Specifying the environment: defining the events, actions and corresponding dependencies that are valid for the designated environment;
2. Specifying the possible explicit norm invocations: desirable and unwanted environmental events for the specified environment must be identified, which is required to define the norm invocation events, actions and corresponding dependencies;
3. Formalizing the scenario description by generating event-action-trees (with appropriate probability assignment).

All these steps are supported by a GUI tool called “EMIL-S Agent Designer” (Figure 70), which facilitates the input of event-action trees, and additionally allows the observation and analysis of output data during simulation runs.



Name	Event
☑ CAR…	E10
□ CAR…	E11
□ CAR…	E12
□ CAR…	E13
□ CAR…	E14
□ CAR…	E15
□ CAR…	E16

Figure 70. User interface: Agent designer, displaying an event-action tree from the initial rule base

Several versions of the traffic scenario have been created during the development of EMIL-S. Some selected model details of the current version are presented below.

Table 2 shows all events that are sent to EMIL-S and, hence, regarded for the normative process. The "interactive road section" mentioned at E15 and E16 is the striped area of the road in this model, but can also be replaced by a dynamic marker, which for instance signalizes a road section where a notably high number of collisions occurred in the past. The events E30 and E31 are only used in explicit norm invocation messages expressing a positive or negative sanction, respectively.

Event ID	Event description
E10	pedestrian on road is observed by car driver
E11	pedestrian on lawn is observed by car driver
E12	collision with pedestrian
E15	"interactive" road section was entered by car
E16	"interactive" road section was left by car
E20	new target has appeared
E21	car on road detected by pedestrian
E22	target was reached by pedestrian
E23	collision with car
E24	opposite side of road was reached by pedestrian
E25	pedestrian is about to enter the road
E30	reward - faster than average
E31	sanction - slower than average

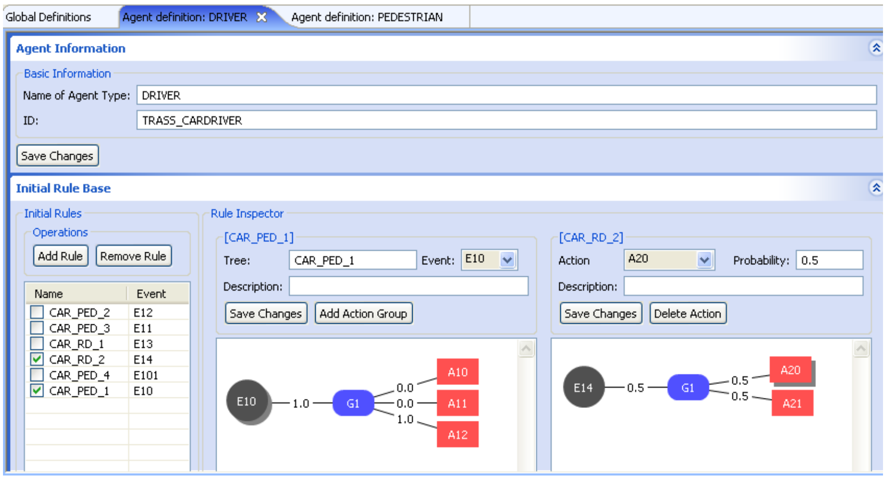
Table 2. Events of the traffic scenario

Table 3 lists the normative actions EMIL-S has to choose between. The motion actions (A10 to A12) and perception actions (A20 and A21) are valid for both the pedestrians and car drivers. A30 is interpreted by a pedestrian reaching for the target as a command to make a detour and use a presumably saver zone for crossing the road; A31 means to use the direct way. The two actions ANI10 and ANI11 are norm invocation actions, i.e. impact of the action is the sending of a sanction (with the strength determined by the expression) to the opponent in a collision.

Action ID	Action description
A10	accelerate
A11	slow down
A12	stop
A20	enable wide perception
A21	disable wide perception
A30	enable detour
A31	disable detour
ANI10	admonish – expression=ANI10(SANCTION, -0.1)
ANI11	blame – expression=ANI11(SANCTION, -0.6)

Table 3. Actions of the traffic scenario

Finally, Figure 71 shows an agent designer screenshot disclosing two (of 12 in total) examples of initial rules (here for the car driver agent type).



Name of Agent Type: DRIVER

ID: TRASS_CARDRIVER

Name	Event
☐ CAR_PED_2	E12
☐ CAR_PED_3	E11
☐ CAR_RD_1	E13
☑ CAR_RD_2	E14
☐ CAR_PED_4	E101
☑ CAR_PED_1	E10

[CAR_PED_1]

Tree: CAR_PED_1 Event: E10

Description:

[CAR_RD_2]

Action A20 Probability: 0.5

Description:

Figure 71. Agent designer: two rules of the traffic scenario

14.4 Simulation Runs and Results

Primarily the TRASS IDE is used as execution environment for the traffic model; however, also a MEME plug-in for TRASS-based models was developed. While in the first case results are presented by animation sequences or snapshots of the simulation progress (see Figure 72), the other method is applied mainly when the collecting of raw numerical data is required (which is the case for the “Multi-scenario world – the sequel”, see the respective chapter).

The different versions of the traffic scenario have been executed numerous times in the past with different intentions. The largest fraction of runs was necessary to verify the operational reliability of all involved interfaces. But also a considerable number of runs was (and will be) dedicated to create demonstration material.

When a simulation run starts, pedestrians and cars run into each other. These incidents are interpreted as **norm invocations** (in this case only sanctions with different strength):

- An incident perceived as a near-collision leads to an admonition (action ANI10), which can be translated to “You have to stop when I am stepping on the street!” (pedestrian) or “You must not step on the street when I am around with my car!” (car driver).
- A collision leads to a blame (action ANI11) and can be translated to “It was bad that you have dented my car/torn my clothes!”.

As a typical situation after a sufficient number of interactions (which is usually the case after a few hours of artificial simulation time, equivalent to several minutes of real time, depending on hardware) the pedestrians have learnt that they have to use the striped area for street crossing (in spite of a potentially longer route), car drivers have learnt that they are expected (obliged) to slow down or stop in front of the striped area (which has emerged into an institution after the first successful norm learning happened there) when there are pedestrians visible in the neighbourhood.

Figure 72 shows a screenshot of a simulation run after a learning period. There are 20 pedestrian agents observable, heading towards the target. In the shown situation the majority of the pedestrians use the crosswalk to cross the road, while a minority chose the direct way. As soon as all pedestrians have “touched” the target, a new target appears on the opposite side of the road, thus another turn for interactions is opened.

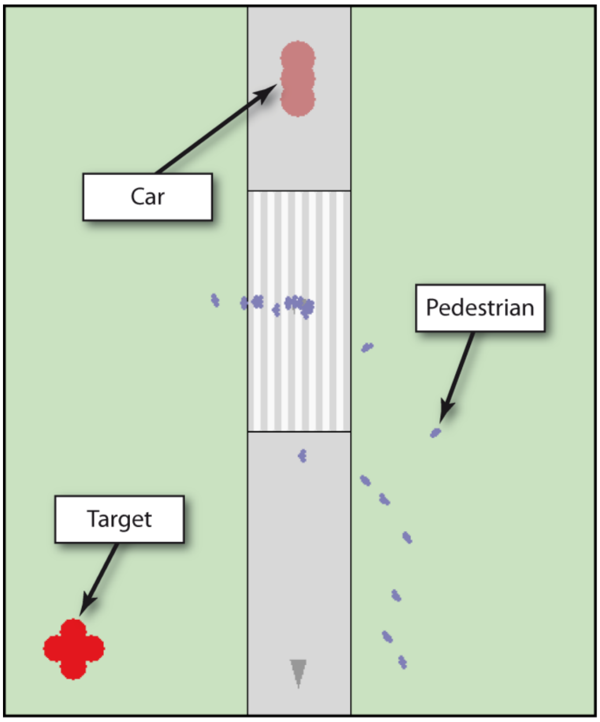


Figure 72. Simulation run screenshot

Chapter 15 Demonstrating the Theory 2: Wikipedia

Ulf Lotzmann, Robin Emde and Klaus G. Troitzsch

Abstract

In this chapter, some but not all of the findings of the empirical analysis of the behaviour of contributors to and discussants of Wikipedia articles are used to build a simulation model of collaborative writing. As software agents are still not able to use natural language to produce texts, an artificial language whose symbols do not refer to anything in the real world was invented, and the software agents were endowed with the capability to produce text in this language and to evaluate something like the “style of writing” in this language, thus being able to take offence at certain features of texts and blaming the authors of such text. From this kind of communication, norms emerge in the artificial society of software agents.

15.1 Introduction: Target Description

From the findings of the two empirical studies of collaborative writing and discussions about its results we define an abstract model of agent behaviour in situations like those described in Chapter 5 and convert this model into executable simulations. At the beginning of a simulation agents have a repertoire of messages (or an algorithm that enables them to formulate new messages) that they want to send to all other agents, i.e. for inclusion in the Wikipedia (where it is copied into a member of the class article), and all agents are allowed to do so, i.e. they will have an attribute that informs whether an agent is an authorised author or not (thus, at the beginning all of them will be entitled to contribute). After some time the Wikipedia, i.e. the environment, contains a list of articles which are not yet connected. These articles contain, of course, the name of the author or sender, have the modal A, as they are just assertions; their recipients are all other agents, and the contents is some string of arbitrary length, the first few elements of which have the special meaning of a keyword.

The contents string could be of any kind, for instance a simple bit string whose first eight bits serve as the entry title (such that exactly 256 entries would be possible). But it might be more helpful to have something like a text separated into words by blanks or some other punctuation. In this case, words could consist of alternating vowels and consonants forming a very primitive language which conveys no meaning, but agents could (perhaps more easily than in a simple bit string) mull over similarities and differences among Wikipedia articles. To limit the number of possible tokens (words), the character repertoire could be restricted to only three vowels and five consonants (“aloha wiki sisal”).[54] Another option could be the task of sorting letters alphabetically and separating different fonts from each other, but this implies an exogenous “fitness measure” for the character strings instead of an emergent practice of article writing and citing.

Besides writing and reading articles and comments and edits, agents will also have to generate a list (each of them separately) of other agents whom they have identified and evaluated for their authority (the “board of authorities”, in Campennì, 2007), as their norm recogniser — an engine that matches new perceptions with the contents of agent’s memory — does not only use the contents of a message but also its sender to evaluate the importance and salience of an utterance (thus an utterance could be more important when it comes from an agent with high “importance” — see below —, or be more salient when it has been received several times).

Agents scan the articles for similarities and comment on them. These commenting actions can be of the following types:

- If agent x finds a match between a keyword of one of its own articles ax and an article by published by someone else (y), then it includes a link to x’s own article ax in y’s article by (which makes it

[54] As a prototype we have a small NetLogo program whose agents write articles in this primitive language, sort them by keywords, and also sort all the occurring words both alphabetically and according to word frequency.

necessary to add an instance variable keeping a link list to the article class (an element of a link list also contains the sender and, perhaps, a time stamp). Adding a link would qualify as modal B (behaviour).

- Articles that have no similarity at all to any other article (i.e. articles with no or few links to other articles) could be less welcome than those that contain several keywords of other articles — nobody would be interested in an article in a Napoleonic Wars Wiki that does not contain any of the words Borodino, Beresina, Waterloo, Napoleon, Blücher and Austerlitz. Thus articles with no links to other articles could be removed and their authors publicly or secretly blamed.
- If an agent finds a similarity between two articles (which it finds only with a certain probability while scanning the Wikipedia for articles containing words that are similar to the keywords this agent has used in its own articles), then it sends a message to the author of the two similar articles to make them aware that their articles are similar. This would again be just an assertion A.

The article published second might be a result of plagiarism[55]. In this case the modal might be Vm (a moral valuation).

The message could also contain the request to remove the plagiarism, then the modal would additionally be a deontic D.

If the same agent has been suspect of plagiarism several times, then the message might also have the modal S (sanction), and the fallible agent might be removed from the list of those who are authorised to contribute, at least for a while.

- If an agent x finds an article by that is similar but shorter than the article ax that it is about to publish, it might merge the old article by with its planned article ax (see section 15.2).
- If an agent z finds two articles ax and by belonging to the same keyword (or to similar keywords) where the similarity between the contents of the articles is low, it will communicate this finding to both agents (an assertion A) and ask (a deontic D) both agents x and y to discuss whether they could merge or purge their articles to avoid contradictory or otherwise misleading content (although in this very artificial language it will be difficult to define what “contradictory” or “misleading” means; see below for an idea how the concept of contradiction could be implemented in this language).

Other types of comments are conceivable. These few examples should suffice to discuss whether this could be a promising concept for the simulation of Wikipedia communication and co-operation. Measures for similarity can easily be found for bit strings (even in case they are of different length, in this case the comparison function must return two values, the point in the longer bit string where the substring starts that is most similar to the shorter string, and the degree of the similarity[56]).

The language described above might not be sufficient for expressing comments; depending on how detailed the modal of a message is described in the M part of the message, it might be sufficient to just mention the identifier of the article in the contents part of the message, letting the recipient know that the reproach refers to behaviour with respect to this article. Further extensions of the toy version of the Wikipedia simulation will make clear what else is needed.

[55] The plagiarism idea was originally developed for the simple bit string, but in a more complex language (such as the one presented in section 3) plagiarism could be modelled as well (note that plagiarism in the real world does not originate from random effects). Perhaps one could design the model in a way that writing a new article is considerably more costly than copying from an old article.

[56] The minimum Hamming distance between the shorter bit string of length l_1 and all the substrings of length l_1 of the longer substring is calculated and returned as the degree of similarity, while the number of the bit in the longer string where the best matching substring starts is returned as the starting point.
Another mode of checking for similarity and novelty at the same time is to compress each text separately and jointly and to compare the lengths of the compressed versions — if the length is not increased in the compression result of the joint version, then nothing new is added. This kind of algorithm is appropriate for the case of the simple language. The prototype uses a slightly simpler version: it just counts the different words that occur in the two texts separately and the different words that occur in the concatenated texts, and if the sum of the two former counts equals the latter count than the two texts have no word in common.

Another question is what might emerge from communication like this. Obviously, assertions have no direct consequences for the agents' behaviour. Deontics and validations will be processed by the norm recogniser and the norm adoption engine and be converted into a goal which is then processed by the decision maker. It is questionable whether the normative action planner is necessary at all in the Wikipedia scenario, as the action to be performed will consist of just "pressing a submit button" for the next contribution to either Wikipedia or the discussion forum. In other scenarios, the normative action planner might be necessary.[57]

15.2 Scenario History: A NetLogo Prototype and Its Results

In the NetLogo Wikipedia simulation[58], agents can perform different activities (see Figure 73):

- write an article (A1) and either submit it (A2) or add it to an existing article referring to the same keyword (A3),
- plagiarise, i.e. copying an existing article and publishing it for a new keyword (A4),
- search the current state of the encyclopaedia for double entries, for words that do not obey the vowel harmony or for plagiarisms (A5), and reproach the respective author or authors (A6),
- count articles that contain a word about which they wrote an article (A7),
- do nothing.

Which of these activities they select depends on the profits they individually generated when performing these activities in the past (a simple form of reinforcement learning). These profits can be selected with the help of NetLogo sliders.

An article is a string consisting of the characters "aei bklsw." (including the blank and the full stop separating words and sentences, respectively) which is introduced by a word, followed by a colon, as the keyword, and ending in a reference to both author and time, enclosed in square brackets. In longer simulations, it might be necessary to reduce the number of possible words, either by restricting the word length (increasing the probability that the next character is a blank — in the current version the maximum word length is five letters) or by forbidding certain co-occurrences of letters (for example by vowel harmony as in Hungarian and several other languages, such that "a" and "e" would not co-occur in a word). This, however, is currently done in an emergent manner, where commenting authors take offence at words violating the vowel harmony. Word length and/or vowel harmony can be conceived of as correlates of style (which is often an object of discussion in the real-world Wikipedia, see Goldspink, 2007). Other accidental features of the words of this language can also be interpreted for the process of commenting simulated articles, e.g. the words "wasal" and "lawas" could be defined as opposites to each other.

Writing an article starts with constructing a keyword out of the letters "aei bklsw" (including blank but not the full stop) where the probability of selecting a particular letter as the first letter in the word is equal (with the exception of the blank) whereas the probability of selecting the next letter depends on the previous letter, according to a stochastic matrix which is currently constant (but could as well change over time, according to the practice developing in the community). The blank character is selected with a certain probability, ending the construction of the word. The first word of an article is marked by a following colon as a keyword (and for some trivial technical reason it is preceded by the character ">"). The following words are constructed the same, and after each word a full stop is inserted with a certain (low) probability, such that the chain of words is separated into something like sentences. At the end of a sentence the article ends with another (low) probability, and the author adds its name (NetLogo's `who` number) and the time when it is published (NetLogo's `ticks`). When an agent has finished a word it might decide to discard it as it violates the vowel harmony (in the beginning this is not forbidden although other agents might take offence at such words).

If an agent has decided to submit its article it first has to find out whether an article referring to the same keyword already exists, if so it has to decide whether it wants to insert its text into the existing article or

[57] Think of a smoker–non-smoker scenario where the decision is "smoke" which can result in a number of different action plans (leave the restaurant and smoke outside, smoke within the restaurant with or without the consent of the other guests, etc.).

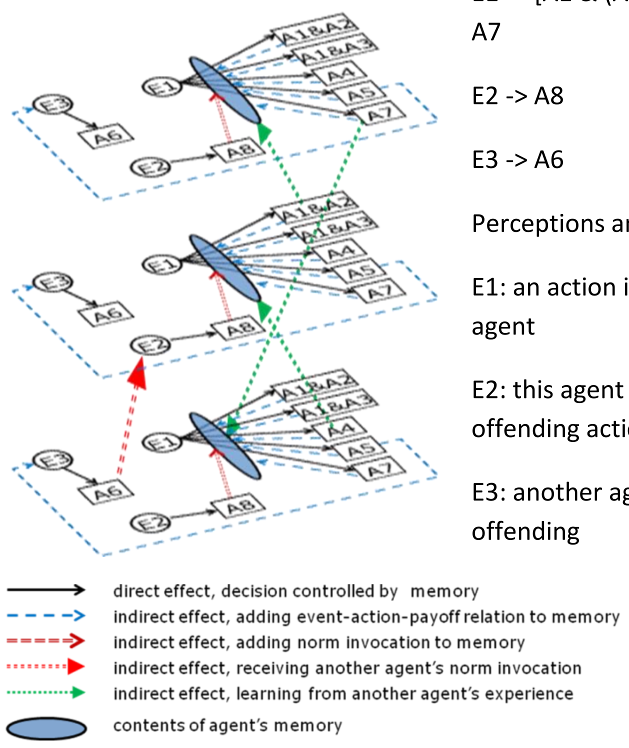
[58] The NetLogo program can be downloaded from http://userpages.uni-koblenz.de/~kgt/Pub/Wikipedia.nlogo.

whether the article is just going to be published (in the beginning this is not forbidden although other agents might take offence at double entries).

If an agent has decided to add to an existing article it first has to find out whether such an article exists, and if not publishes its text as an ordinary article.

Commenting on articles is currently implemented as a scan of all articles, searching for entries which refer to the same keyword and for entries referring to a keyword that violates vowel harmony. The scanning agent generates a list of all those keywords and identifies all the authors that wrote a second or third article referring to the same keywords as blamable (a mild deontic of the proscription type) and deletes all younger articles (a sanction in terms of Andrighetto et al., 2007a, as the "authority" or "importance" of an author is measured in terms of the number of entries published by this author, i.e. the number of articles published by this author is decremented — currently this has no consequences in the implemented model, but it could have, e.g. in a way that a sanction or validation coming from an important author is more severe, as explained above). Authors having published articles referring to keywords violating vowel harmony are also blamed, and such a keyword is replaced with a similar word obeying the vowel harmony (by exchanging an "a" with an "e" or the other way round; this replacement also affects the article list, as the entries referring to the two words have to be merged).

A third activity of agents is the search for similarities between two randomly selected articles. If an agent detects a co-occurrence of more than a certain percentage of words between two articles, it identifies the younger of the two articles as a potential plagiarism and blames its author who loses the profit generated before from intentional plagiarism. If the similarity is just by chance, the same hard punishment occurs.


A1&A2
A1&A3
A4
A5
A7
E1
E2
E3
A6
A8
direct effect, decision controlled by memory
indirect effect, adding event-action-payoff relation to memory
indirect effect, adding norm invocation to memory
indirect effect, receiving another agent's norm invocation
indirect effect, learning from another agent's experience
contents of agent's memory

E1 -> [A1 & (A2|A3)] | A4 | A5 | A7

E2 -> A8

E3 -> A6

Perceptions and events

E1: an action is profitable for this agent

E2: this agent was blamed for an offending action

E3: another agent's action was offending

Figure 73. Relations between events, perceptions, actions and the effects of actions

More formally, perceptions, events, actions and the relations among them can be described as in Figure 73.

When an agent finds itself in a situation where it could contribute to the encyclopaedia it consults its memory to find how profitable the different actions available in this situation might be. The memory does not only contain information about earlier payoffs, but also information about norm invocations from the side of other agents who might have taken offence at earlier actions of this agent (as these — E3→A6 — will have resulted in a reproach which in turn was received — E2→A8 — and stored in the agent's memory as an indirect consequence of one of its earlier actions). Beside learning from own experience about payoffs and from reproaches, agents can in principle also learn from observing other agents' actions and the resulting payoffs.

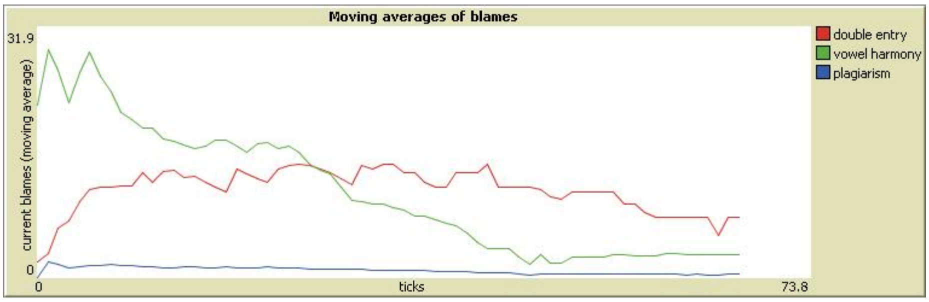


Figure 74. Moving averages of issued blames for the three offences (over 25 time steps, after approximately 70 time steps)

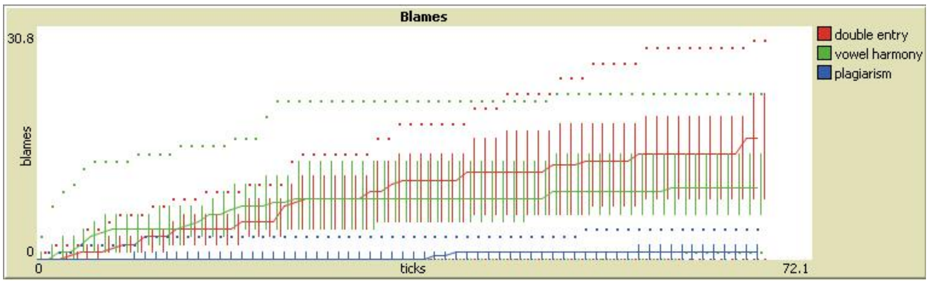


Figure 75. Distribution of individually received blames. Vertical bars are quartile differences, the upper dots are maxima, the dots below are minima, and the curves in the middle are the medians of the distribution of received blames

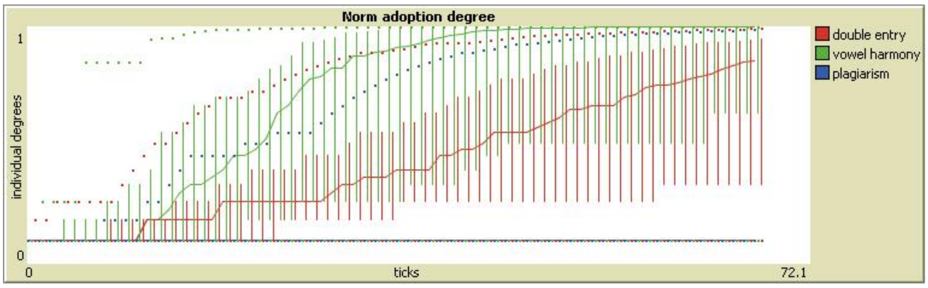


Figure 76. Distribution of the individual norm adoption degrees. Vertical bars, dots and curves have the same meaning as in Figure 75

Figure 74, Figure 75 and Figure 76 show three preliminary results of a long simulation run (70 ticks with 50 agents), in which the agents perform their activities depending on previous profits. The probabilities for each activity are proportional to the sum of all previously received profits (including the negative profit for detected plagiarisms). When one agent takes offence at the outcome of another agent it issues a blame. The individually received blames are shown in Figure 75, where one can see that in the beginning most blames were received for the violation of vowel harmony — bad style. Figure 74 shows how many blames were issued on an average during the past 25 ticks. Here one can see that vowel harmony violations occurred and were detected rather often in the early phase of the simulation, but soon decreased in number, whereas double entries occurred only after some time and plagiarisms even later; again it should be stressed that most suspected plagiarisms were unintentional as only three actually occurred on purpose — it must be mentioned that agents are initialised with a certain chance of abstaining from plagiarism once they had decided to commit plagiarism (to be controlled by a slider).

Figure 76 shows the process of norm emergence. The first blame an agent receives with respect to one of the possible offences (duplicated entries, vowel harmony violations, plagiarisms) puts it to its individual normative frame and makes it deliberate before it gets into danger to commit the same offence again: the probability of offending again is 90 per cent in the first deliberation, in other words: the norm adoption degree is 10 per cent after the first reproach, and later on it increases to $1 - (1 - \nu_0) \exp(-\varphi n)$ where ν_0 is the initial norm adoption degree and φ is a flexibility parameter (both are 0.1 in this simulation run) and n is the number of deliberations performed with respect to this activity.

As soon as at least one half of all agents have received at least one blame and have performed their first deliberations with respect to this kind of offence, the offence is copied to the public normative board (in the current simulation, this happened at times 13, 19 and 34 for vowel harmony, double entry and plagiarism, respectively).

It may remain an open question whether this norm emerged from the individual behaviour rules: these say that agents may blame other agents for a certain behaviour and that agents may consider such a blame and change their behaviour. Perhaps this is still a regularity, but the appearance of a norm on the normative board (although initiated by NetLogo's observer, which — in a way — counts the ballots in a referendum) could be considered as the emergence of an explicit norm. Moreover, one could very easily add the possibility that the first agent observing that blames became suddenly rarer could decree the norm (of course, this necessitates a formalisation of "suddenly" and "rarer").

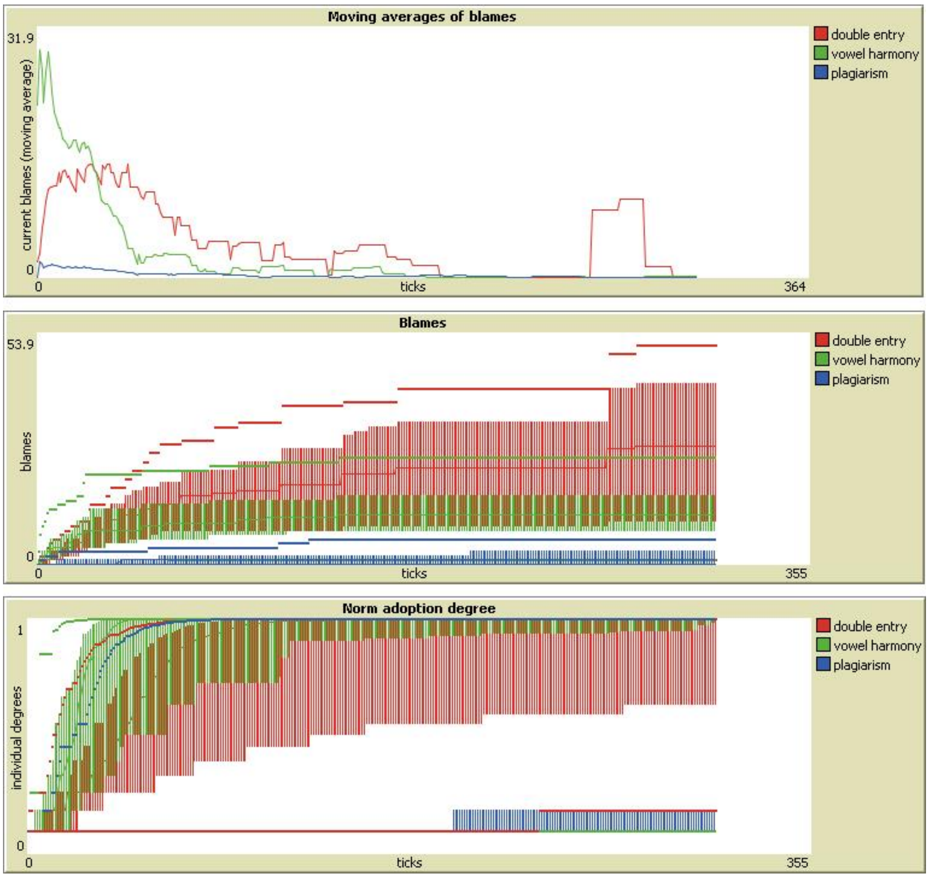


Figure 77. The same simulation run as in Figure 74, Figure 75 and Figure 76, but after approximately 320 simulation ticks

Implementing a representation base and norm recogniser is not trivial in NetLogo, but the current implementation endows every agent with a normative frame, a directory with currently only up to three entries, each consisting of the name of the respective rule, the number of individually received blames for violating this rule and the number of instances when it abode by the respective rule. Thus the norm recogniser is a simple comparison between the received message and the names of the rules already stored in the individual "normative frame" (as it is called in Andrighetto et al., 2007a). Thus if an agent receives a blame containing the hint at double entry or at vowel harmony violation or at plagiarism, it increases the respective number of received blames. After the first blame it takes into account both the possibility of abiding by the pre-norm or violating it, and whenever the decision is in favour of the norm, the respective norm adoption degree (or: the salience of the normative belief) is increased.

The implementation of this prototype clearly suggests that for every type of message ("key") there must be a receptor in the linguistic repertoire of the authors ("lock") that responds to this message; other messages cannot be understood. Thus much of the agents' complexity lies in their linguistic repertoire. The normative frame of the individual agents is currently capable of receiving messages of any content (as the list of entries can easily be added to), but currently no agent is in a position to blame other agents except for the three implemented cases, and agents cannot even react behaviourally upon the blame for plagiarism, as they have no method for withdrawing a blamable article; not plagiarising is just a consequence of the low probability of executing the respective decision and the low profit generated from plagiarising (as the profit for an undetected plagiarism has to be reimbursed after detection).

Other types of acting, commenting and discussing can be — and must be! — implemented, too, e.g. replacing old articles with one's own which in turn can be blamed by the author of the replaced article. These extensions can be programmed by adding to the lists of activities and of offences and by adding procedures describing the related behaviour of the agents. Due to the structure of NetLogo and its language, a more extended version of the current simulation would not be easy to understand.

15.3 Scenario Implementation

The EMIL-S implementation rests upon a Repast model which represents the physical level of the scenario (the collection of articles and the capabilities of agents to read and write). The logical (EMIL-S) level contains an initial configuration (initial rule base for the agents). Several parameters of the model are either input manually, or MEME is used for controlling a series of simulation runs. Figure 78 shows how these components collaborate. The figure shows three levels of detail: The top level shows in a use case diagram what the system is expected to do from the point of view of the intended audience. The second level shows the components of the different parts of the simulation toolbox in which the model has to be implemented. This is shown in the class diagram in the bottom part of the figure.

The three classes — `WikiGUI`; `WikiModel` and `WikiAgent` — are the implementation of the physical level of the simulation. This implementation is executed in the simulation toolbox and interacts with the two other components: MEME will control the simulation in the end and have it run several runs with different parameter combinations, whereas the interaction with EMIL-S consists of messages exchanged between these two components.

The implementation of the physical level can be described as follows. The three classes are separate components of this physical level. `WikiGUI` bundles all elements for user interaction, including windows for textual and graphical output. These technical aspects will be touched only superficially.

`WikiModel` which controls the simulation within the physical level and generated the physical parts (the "bodies" of the agents) contains all methods which are necessary on the physical level: methods to generate agents, to set initial parameters and to control the progress of the simulation run. `WikiAgent`, finally, contains all methods which generate agent behaviour. A simulation run proceeds as follows.

In `WikiModel` agents are created and initialised. The simulation user interface is initialised and a time schedule for simulation steps is generated. This schedule is defined as follows:

Within each tick (comparable to a day in the real world) each agents is asked to perform its activities.

Within `WikiAgent` all methods are implemented which are necessary to execute all possible activities. As soon as an agent is called by `WikiModel` to perform its activities, this agent follows a well-define schedule. The behaviour and actions of an agents is controlled partly by cognitive processes, and partly by physical sequences which are predefined and do not need any consideration. The cognitive processes are exclusively modelled in the EMIL-S component.

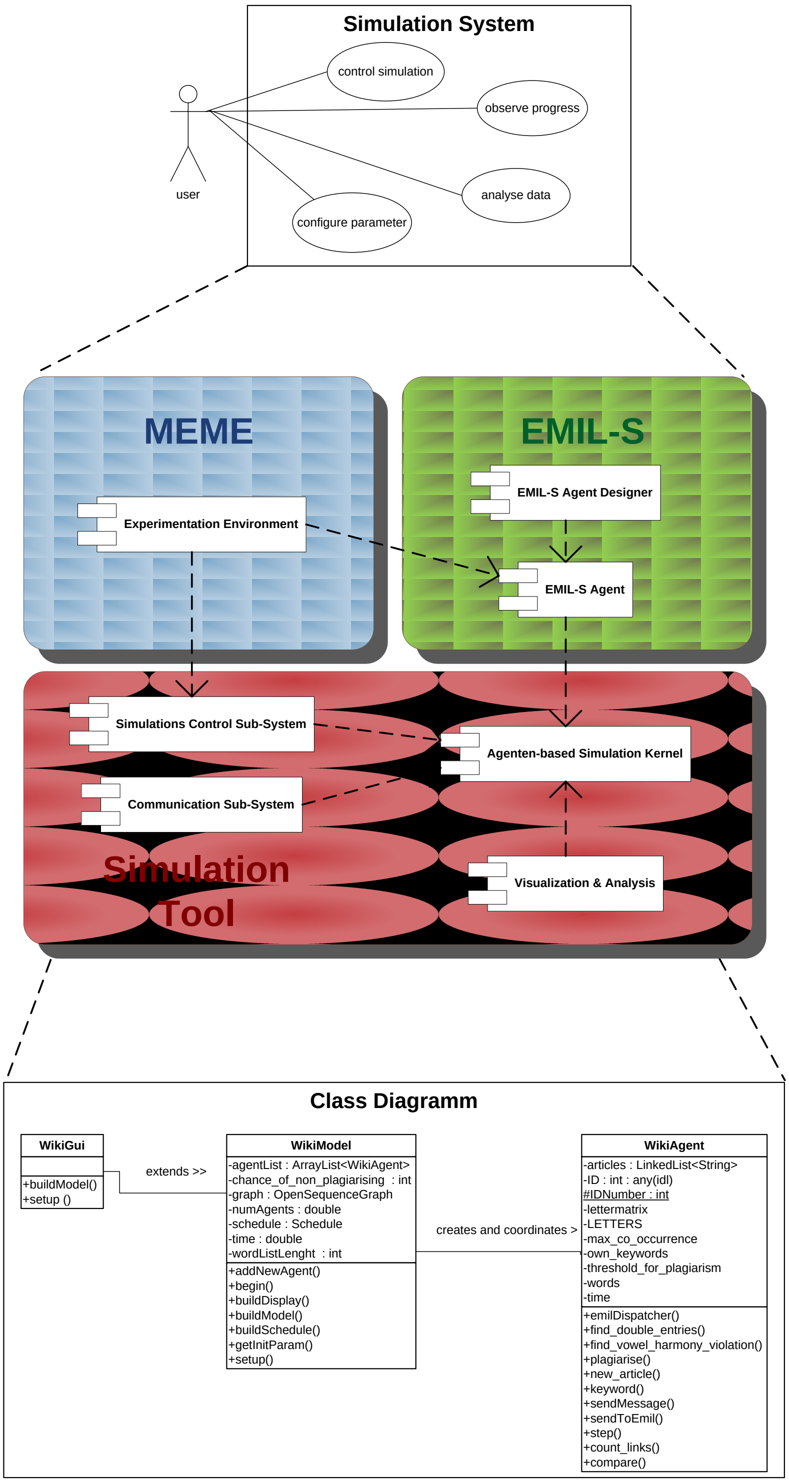


Figure 78. Overview of the components of the EMIL-S version of the Wikipedia scenario

Figure 79 describes the collaboration between the physical and the EMIL-S agent in an activity diagram. Each agent which is called by `WikiModel` will have several actions that can be taken in a certain situation. To decide which action is taken is up to the EMIL-S agent (the "mind" of the agent). The physical agent will thus have to check whether it received any messages from its EMIL-S counterpart. If this is not the case, it sends a request to EMIL-S which asks the EMIL-S agent to make the necessary decision. As soon as this decision was received the physical agent identifies the action to be executed and performs it. These actions can lead to the necessity to make further decisions which again make an interaction with EMIL-S necessary.

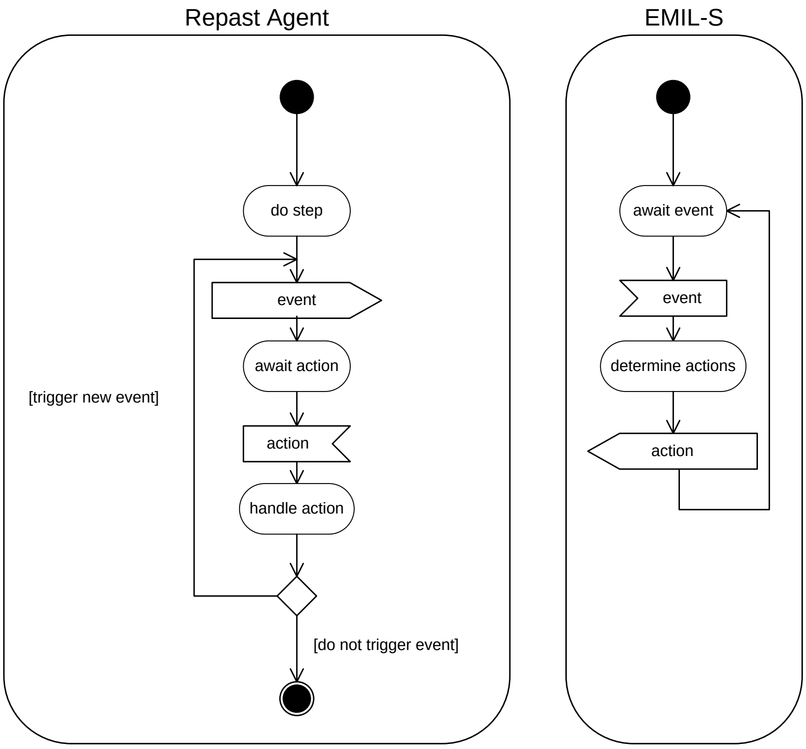


Figure 79. Collaboration between the physical agent and the EMIL-S agent

An example of the communication between the physical and the EMIL-S agent is shown in the sequence diagram of Figure 80.

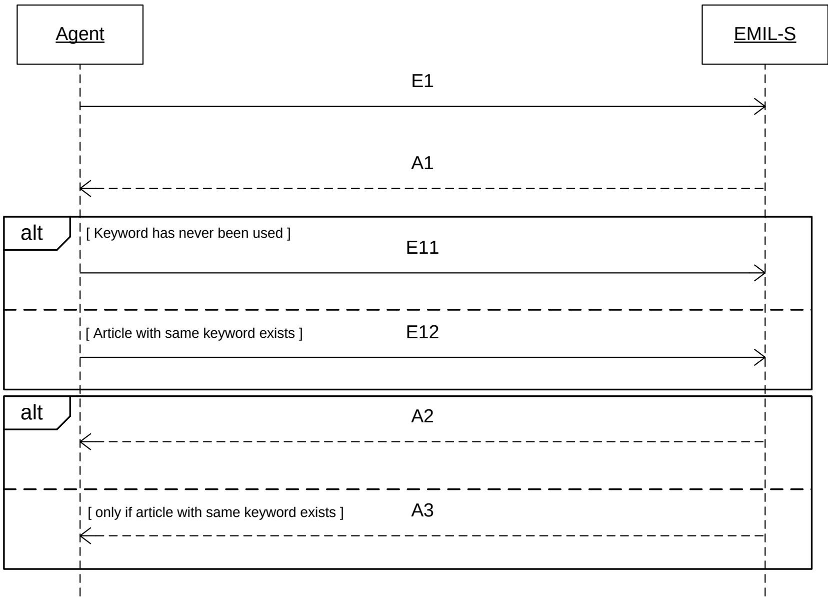

Agent
EMIL-S
E1
A1
alt
[Keyword has never been used]
E11
[Article with same keyword exists]
E12
alt
A2
[only if article with same keyword exists]
A3

Figure 80. Communication between physical and EMIL-S agent, sequence diagram

The physical agent described in this diagram initially sends a message to its EMIL-S counterpart. This message contains the event E1. The EMIL-S agent can now choose between different actions associated with this event (these actions are described in the agent's event-action tree which will be discussed later). The EMIL-S agent may have chosen action A1 which means that the physical agent has to write a new Wikipedia article. This action is executed by the physical agent, but it consists of several activities: First the entry keyword has to be generated and checked whether it already exists in the Wikipedia. If this is the case, the event E12 has to be sent to the EMIL-S such that this one can decide whether another article is published under the same keyword or whether the existing article referring to this keyword has only to be extended. In the opposite case the event E11 is sent and the only possible decision is to write and publish an article with a certain probability. All possible events and the decisions to be taken as consequences of these events are kept in the initial rule base written in XML or put together with the help of the EMIL-S agent designer.

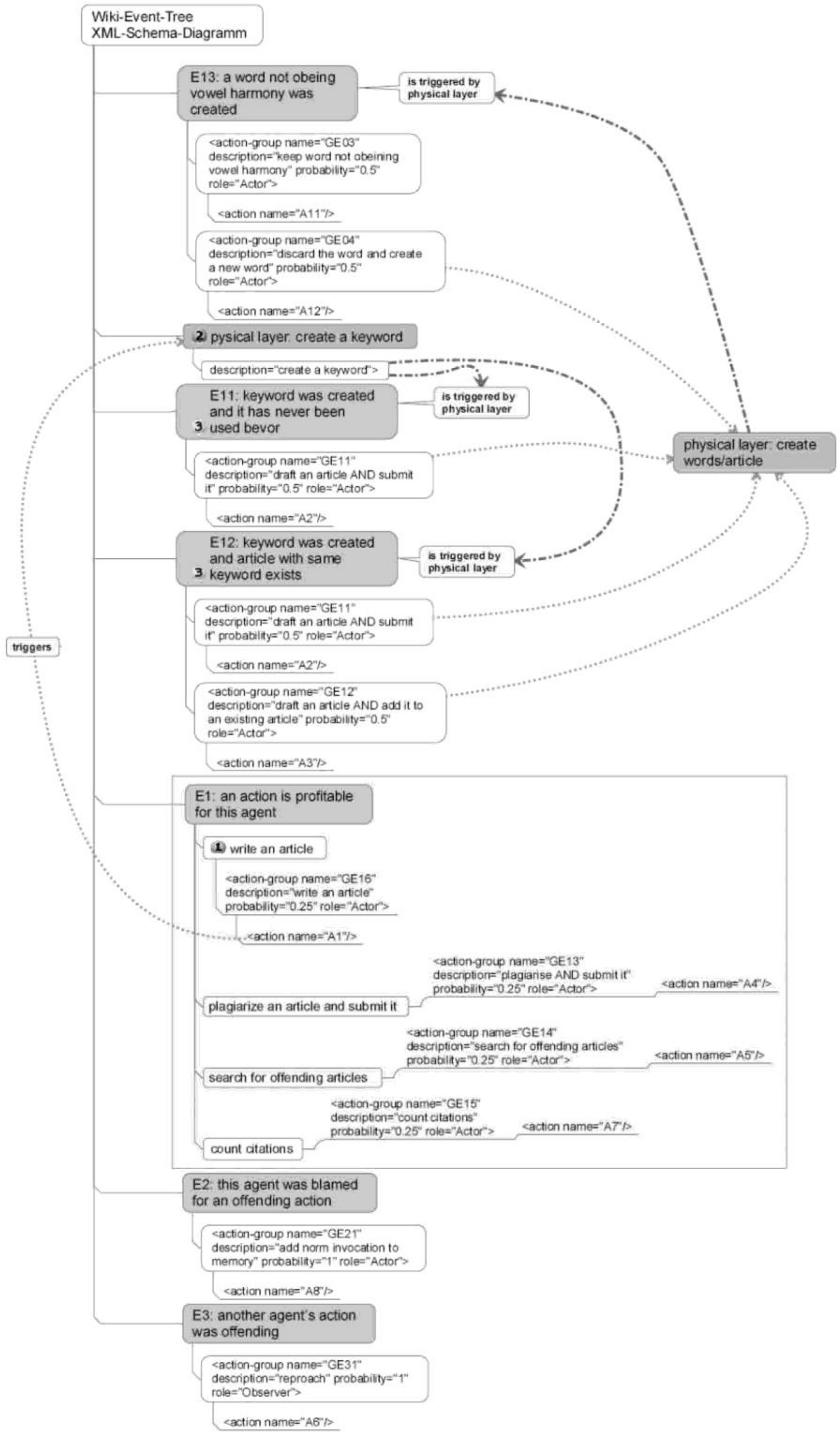

Wiki-Event-Tree
XML-Schema-Diagramm
E13: a word not obeing vowel harmony was created
is triggered by physical layer
<action-group name="GE03" description="keep word not obeining vowel harmony" probability="0.5" role="Actor">
<action name="A11"/>
<action-group name="GE04" description="discard the word and create a new word" probability="0.5" role="Actor">
<action name="A12"/>
2 pysical layer: create a keyword
description="create a keyword">
E11: keyword was created and it has never been used bevor
3
is triggered by physical layer
physical layer: create words/article
<action-group name="GE11" description="draft an article AND submit it" probability="0.5" role="Actor">
<action name="A2"/>
E12: keyword was created and article with same keyword exists
3
is triggered by physical layer
<action-group name="GE11" description="draft an article AND submit it" probability="0.5" role="Actor">
<action name="A2"/>
triggers
<action-group name="GE12" description="draft an article AND add it to an existing article" probability="0.5" role="Actor">
<action name="A3"/>
E1: an action is profitable for this agent
1 write an article
<action-group name="GE16" description="write an article" probability="0.25" role="Actor">
<action name="A1"/>
<action-group name="GE13" description="plagiarise AND submit it" probability="0.25" role="Actor">
<action name="A4"/>
plagiarize an article and submit it
<action-group name="GE14" description="search for offending articles" probability="0.25" role="Actor">
<action name="A5"/>
search for offending articles
<action-group name="GE15" description="count citations" probability="0.25" role="Actor">
<action name="A7"/>
count citations
E2: this agent was blamed for an offending action
<action-group name="GE21" description="add norm invocation to memory" probability="1" role="Actor">
<action name="A8"/>
E3: another agent's action was offending
<action-group name="GE31" description="reproach" probability="1" role="Observer">
<action name="A6"/>

Figure 81. Initial Rule Base of the Wikipedia scenario

Figure 81 is a graphical representation of this rule base. Events are marked with the letter E and a number, and associated actions and action groups are annotated below the respective events in a tree structure. Actions trigger new events on the physical layer. Physical events and actions are marked in green whereas decisions made and events happening in EMIL-S are marked blue. With the design in

Figure 81 both the EMIL-S configuration and the implementation of the methods necessary on the physical layer are easily done now.

All decisions with respect to agent behaviour and actions are made in the EMIL-S layer, but some decisions are instead made on the physical layer, namely the decision about the length and contents of the articles written. As the language used by the agents for writing articles does not have any semantics or relation to the world outside the scenario, length and contents need not be decided by the agents, but in extended scenarios at least the length and the style might have an effect on the reputation of agents — which necessitates some changes in the basic configuration of

Figure 81. But to demonstrate the XML notation of the configuration file, a simpler scenario is more appropriate.

```
<?xmlversion="1.0"encoding="utf8"?>
<emil-s-configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="file:///Path/emil-s-configuration.xsd">
 <global-definitions>
  <rule-elements>
    <event name="E1" description="an action is profitable for this agent"/>
    <event name="E2" description="this agent was blamed for an offending action "/>
    <event name="E3" description="another agents action was offending"/>
    <event name="E11" description="keyword was created and it has never been used bevor"/>
    <event name="E12" description="keyword was created and article with same keyword exists"/>
    <event name="E13" description="a word not obeing vowel harmony was created"/>
    <action name="A1" description="create a keyword">
      <type>External</type>
      <expression>1</expression>
    </action>
    <action name="A2" description="write an article and submit it">
      <type>External</type>
      <expression>2</expression>
    </action>
    <action name="A3" description="write an article and add it to existing keyword">
      <type>External</type>
      <expression>3</expression>
    </action>
    <action name="A4" description="plagiarise an article and submit it">
      <type>External</type>
      <expression>4</expression>
    </action>
    <action name="A51" description="search the current state of the encyclopaedia for double entries">
      <type>Internal</type>
      <expression>51</expression>
    </action>
    <action name="A52" description="search the current state of the encyclopaedia for words that do not
         obey the vowelharmony (they are bad style)">
      <type>Internal</type>
      <expression>52</expression>
    </action>
    <action name="A53" description="search the current state of the encyclopaedia for plagiarisms">
      <type>Internal</type>
      <expression>53</expression>
    </action>
    <action name="A6" description="reproach the respective author or authors">
      <type>External</type>
      <expression>6</expression>
    </action>
    <action name="A7" description="count articles that contain a word about which they wrote an article">
      <type>Internal</type>
      <expression>7</expression>
    </action>
    <action name="A8" description="add norm invocation to memory">
      <type>Internal</type>
      <expression>8</expression>
    </action>
    <action name="A11" description="keep word">
   <type>External</type>
   <expression>11</expression>
  </action>
  <action name="A12" description="discard word and create a new word">
   <type>External</type>
   <expression>12</expression>
  </action>

 </rule-elements>
</global-definitions>
<agent-definitions>
 <agent name="author"id="Jimmy_Wales">
  <initial-rule-base>

   <event-action-tree name="judge-word" description="base event-action-tree">
    <event name="E13"/>
     <action-group name="GE03" description="judge-word" probability="1" role="Actor">
      <action name="A11" probability="0.5"/>
      <action name="A12" probability="0.5"/>
      </action-group>
   </event-action-tree>

   <event-action-tree name="write article process" description="desicion weather or not to add to existing article">
    <event name="E12"/>
    <action-group name="GE12" description="desicion weather or not to add to existing article"
            probability="1" role="Actor">
     <action name="A2" probability="0.5"/>
     <action name="A3" probability="0.5"/>
    </action-group>
   </event-action-tree>

  <event-action-tree name="write article process" description="add article to Wiki">
   <event name="E11"/>
   <action-group name="GE11" description="draft an article AND submit it" probability="1"
         role="Actor">
    <action name="A2" probability="1"/>
   </action-group>
  </event-action-tree>

  <event-action-tree name="do something" description="following event-action-tree">
   <event name="E1"/>
   <action-group name="GE16" description="write an article" probability="0.8" role="Actor">
    <action name="A1" probability="1"/>
   </action-group>
   <action-group name="GE13" description="plagiarise AND submit it" probability="0.5" role="Actor">
    <action name="A4" probability="1"/>
   </action-group>
   <action-group name="GE14" description="search for offending articles" probability="0.3"
      role="Actor">
    <action name="A51" probability="0.2"/>
    <action name="A52" probability="0.1"/>
    <action name="A53" probability="0.7"/>
   </action-group>
   <action-group name="GE15" description="count citations" probability="0.3" role="Actor">
    <action name="A7" probability="1"/>
   </action-group>
  </event-action-tree>

  <event-action-tree name="learning process" description="this agent was blamed for an offending action
            and learns from that">
   <event name="E2"/>
   <action-group name="GE21" description="add norm invocation to memory" probability="1"
         role="Actor">
    <action name="A8" probability="1"/>
   </action-group>
  </event-action-tree>

  <event-action-tree name="reaction" description="another agents action was offending, this is the
         reaction">
   <event name="E3"/>
   <action-group name="GE31" description="reproach" probability="1" role="Observer">
    <action name="A6" probability="1"/>
   </action-group>
  </event-action-tree>
  </initial-rule-base>
 </agent>
</agent-definitions>
```

```
 <environment-definitions>
       <normative-board>
         <rule>
           <event-action-tree>
              <event/>
               <action-group>
                          <action>
                          </action>
              </action-group>
           </event-action-tree>
         </rule>
       </normative-board>
 </environment-definitions>
</emil-s-configuration>

<!-- ab hier Bedingungen für referentielle Integritaet / begin of integrity constraints -->
  <!-- fuer die events /for the events -->
  <xsd:key name="my-event-Id">
    <xsd:selector xpath="./global-definitions/rule-elements/event"/>
    <xsd:field xpath="@name"/>
  </xsd:key>
  <xsd:keyref name="my-event-Id-ref" refer="my-event-Id">
  <xsd:selector xpath="./agent-definitions/agent/initial-rule-base/event-action-tree/event"/>
  <xsd:field xpath="@name"/>
</xsd:keyref>
<!-- und fuer die actions /for the actions -->
  <xsd:key name="my-action-Id">
    <xsd:selector xpath="./global-definitions/rule-elements/action"/>
    <xsd:field xpath="@name"/>
  </xsd:key>
  <xsd:keyref name="my-action-Id-ref" refer="my-action-Id">
    <xsd:selector xpath="./agent-definitions/agent/initial-rule-base/event-action-tree/action-group/action"/>
    <xsd:field xpath="@name"/>
  </xsd:keyref>
<!--ende der Integritaetsbedingungen / end of integrity constraints -->
```

15.3.1 Repast for the Physical Layer

Repast (an acronym for “Recursive Porous Agent Simulation Toolkit”) is a tool for agent-based simulation, originally developed by Sallach, Collier, Howe, North and others at the University of Chicago (Repast Development Team, 2009). Repast allows to implement models with the help of JAVA. It offers a large number of methods facilitating implementation, for instance with methods for simulation control. A basic class for agents is available which can easily extended by methods which are special for the model in question. Repast also abounds in graphical elements for the environment of the agents and for plotting simulation results, the graphical user interface can be used for textual input and output of parameters and results.

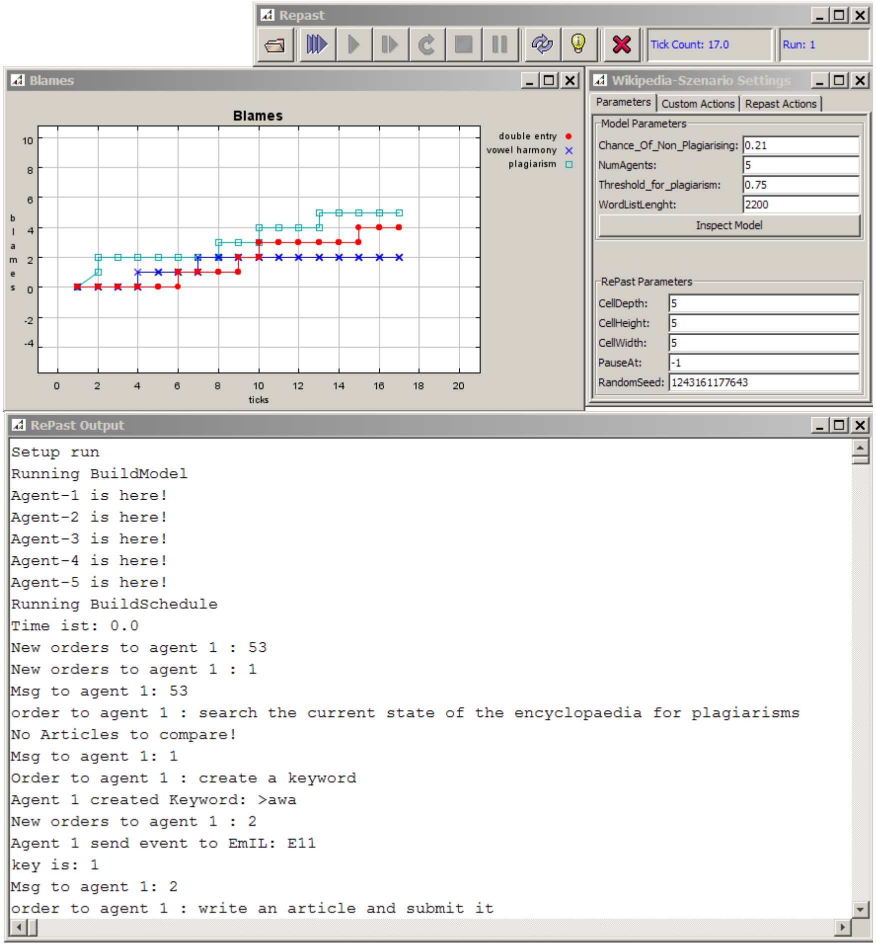
For the purpose of controlling (starting, interrupting, resuming and stopping) the Wikipedia simulation and to initialise it with some parameters a graphical user interface (see Figure 82) is necessary and easily available in Repast. The definition of the agents in the physical layer and their communication with their EMIL-S counterparts has to be programmed in JAVA. The communication between the physical layer and the user (or alternatively with the MEME simulation control engine) is also programmed in JAVA.

As the components to be implemented will be executed within the Repast simulation tool, they have to conform with the Repast conventions. Generally speaking the minimum Repast model consists of two classes: the model class and the agent class. While the model class implements simulation parameters and schedule and controls the simulation, it is the agent class (or classes) that bundles all the activities that define the behaviour of the respective agents.

The model class `WikiModel` — as a specialisation of Repast’s `SimModelImpl` — has to contain several operations the most important of which are the following:

- `setup()` — this method is called when the setup button of the graphical user interface is pressed; it is responsible for all initialisations of the model. In our case it also initialises `agentlist` and `schedule`. `agentlist` is an `Arraylist` in which references to the agents are stored in order to call them during the simulation. This methods also loads the EMIL-S configuration XML file.

- `begin()` just calls the next two methods.
- `buildModel()` — this method builds the model (as its name says), for instance it creates the agents and initialises them. It also puts the references to the agent into the `agentlist Arraylist` and calls the `report` method of each individual agent (just to check whether all agents have been correctly created). In extended versions of this model, individual initialisations of agents can also be implemented here.
- `buildSchedule()` — this method allows to implement a time schedule. All agents are asked to execute the `step()` method., and the plot or plots are updated. All this is done within the `WikiStep` class which is instantiated at the beginning of each Repast time step by the statement `schedule.scheduleActionBeginning(0,new WikiStep()` — this leads to a new round for all agents until there is no new event `ActionBeginning`.. Other important events are `AtEnd` and `AtPause` — the former is triggered by pressing the stop button of the user interface, the latter leads to one line of output.



Model Parameters

Chance_Of_Non_Plagiarising: 0.21
NumAgents: 5
Threshold_for_plagiarism: 0.75
WordListLenght: 2200

RePast Parameters

CellDepth: 5
CellHeight: 5
CellWidth: 5
PauseAt: -1
RandomSeed: 1243161177643

```
Setup run
Running BuildModel
Agent-1 is here!
Agent-2 is here!
Agent-3 is here!
Agent-4 is here!
Agent-5 is here!
Running BuildSchedule
Time ist: 0.0
New orders to agent 1 : 53
New orders to agent 1 : 1
Msg to agent 1: 53
order to agent 1 : search the current state of the encyclopaedia for plagiarisms
No Articles to compare!
Msg to agent 1: 1
Order to agent 1 : create a keyword
Agent 1 created Keyword: >awa
New orders to agent 1 : 2
Agent 1 send event to EmIL: E11
key is: 1
Msg to agent 1: 2
order to agent 1 : write an article and submit it
```

Figure 82. The Repast interface to the Wikipedia simulation model

The `WikiModel` class contains some more methods which are necessary to input parameters from the graphical user interface.

The agent class contains all elements of the implementation that describe the physical behaviour of the agents. Its most important method is `step()` which was already mentioned as the method called by the scheduler at any time step. Other important methods are

- `emilDispatcher()` — a method that interprets the messages received from the physical agent's EMIL-S counterpart. The message is decomposed into its constituent parts and calls additional methods depending on the contents of the message. As these messages contain the actions (as defined in the configuration), `emilDispatcher()` now calls the respective methods which executes the requested action. If for instance, the message has the contents “1” (for action A1) the method keyword() is called and executed which generates an entry keyword for future use; depending on whether this keyword already exists, the event E11 or E12 is sent to EMIL-S (as described in less detail above) which makes EMIL-S respond with another action. In a way this method acts as a switchboard which converts the decisions of EMIL-S agents into actions in the physical layer and reports the outcomes of these actions back to the respective EMIL-S agent or agents.

- `AgentMsgComProcessing()` is quite similar to `emilDispatcher()` but here messages from other agents are processed.. In the Wikipedia scenario these are sanctions sent from one agent to another agent. The EMIL-S agent’s decision to blame a fellow agent results in a physical action which is communicated to the physical version of the blamable agent which forwards this reproach to its EMIL-S counterpart. This message contains a time stamp and a reference to the action which the blamable agent had performed earlier on such that the blamable agent can learn from the reproach.

The described methods call auxiliary methods, such as

- `sendMessage()` — is used for sending messages from the EMIL-S layer to the physical layer.
- `sendToEmil()` — is used the other way round.
- `sendToAgent()` — is used by a physical agent to send a message to another physical agent.
- `keyword()` generates an entry keyword.
- `new_article(String my_keyword)` produces a new article (a string consisting of substrings — words —, separated from each other by blanks or stops) and puts the keyword into a list which allows the individual agent to find out how many of its articles were linked or quoted.
- `find_double_entries()` is a method which allows agents to search the Wikipedia for articles referring to the same keyword — which is one situation which agents can take offence at..
- `find_vowel-harmony_violation()` is a method which allows agents to detect articles which are written in a style that violates the vowel harmony, to correct this article and to reproach its author.
- `count_links()` allows agents to search the Wikipedia for articles which contain words about which they had written an article before, such that they can find out how often their articles had been cited.
- `plagiarise()` allows an agent to copy an existing article and to publish it as its own.
- `compare()` allows an agent to compare an arbitrarily selected Wikipedia article with all other articles. If the similarity between two articles is higher then `threshold_for_plagiarism` then the younger article is supposed to be a plagiarism, it is deleted and the author is reproached.

Several other methods have an even more auxiliary function.

The third class worth mentioning is WikiGUI. It complements WikiModel with elements of a graphical user interface. It was only implemented to separate these elements from the internal simulation control. It contains methods for creating windows for plotting. Other graphical elements could be added for outputting the values of other interesting variables.

15.3.2 Communication with EMIL-S

The communication between the physical layer and the logical layer (between “bodies” and “minds” of agents) is implemented in a special EMIL-S interface to Repast called `emil.agent.IEMILAgent-Wrapper` defined within `WikiAgent`. With the help of this interface an EMIL-S agent is created as a counterpart of the physical (Repast) agent, and a reference to the EMIL-S agent is stored in the Repast agent. This is done with the statement `emil.Controller.getInstance().addAgent( agentType, agentID, refAgentWrapper)` where `agentType` had been defined in the configuration XML file and `agentId` is the unique identification of the Repast agent. `refAgentWrapp-er` is a reference to the active agent object. The actual statement in this program is hence `emIlRef =`

`Controller.getInstance().addAgent("author", ID, this);`. Henceforth the Repast agent can send messages to its EMIL-S counterpart and to receive message from there. This is done with two additional methods. for sending to EMIL-S, `processMessage` is used, with a message of type `IEMILMessage` as its argument. The other way round, EMIL-S sends messages to its Repast counterpart by `sendMessage` which also expects a message of type `IEMILMessage` as its argument. Repast stores incoming messages in a `Queue` from which the addressee agents reads it when it is its turn. All messages are of type `IEMILMessage` which is defined as `EMILMessage(Object sender, Object recipients, Modal modal, IContent<?> content)`.[59]

15.4 Simulation Runs and Results

The Wikipedia/Collaborative Writing Scenario has been extended in order to model the hypotheses about the behaviour of Wikipedia collaborators as developed in Chapter 5.

In the new scenario there are now three groups:

1. The "normal" agents who obey the vowel harmony (e.g. the word `"aseka"` does not conform to the vowel harmony so applying the phonetic process the word would become either `"asaka"` or `"eseke"`).
2. The "rebel" agents, they interpret the vowel harmony in the inverse sense, i.e. they prefer words which contain both front and back vowels. (e.g. the word `"asaka"` would be changed to `"aseka"`). They are the exact counterpart to the normal agents.
3. The "anarchy" agents, they have their own word formation rules (e.g. they change every occurrence of the letters: `"be"` either to `"ab"` or to `"eb"`, i.e. in addition to a possible vowel change they practice metathesis, such as from Middle English *hros* to Modern English *horse*)

The agents were enabled to change their group membership. Every time an agent searches the database for words not obeying its philosophy it changes all occurrences of these words to the "correct" word and blames the author of these words for obeying the latter's rules and/or for making the wrong decision (to keep a wrong word either than dropping it and creating a new one which might be correct).

If an agent creates a keyword which has been used as a keyword in an existing article, it can choose to either add its article referring to the same keyword to the Wikipedia (so there would be two articles with the same keyword which is bad style!) or it could add its article to the content of the existing one. This case was extended as compared to the prototype version.

The user now can choose between two modes:

1. The agent now checks if the existing article is sufficiently similar to the article the other agent wrote (for now this is done by comparing the number of words in both articles) if the similarity is below a configurable threshold, the contents of the new article are added to the old article.
2. A second kind of an agent's decision depends on the "quality" of an article. This "quality" is the number of links, a link being defined here an occurrence of the keyword of the given article in another article. This quality value is attributed to the article. The author of the new content can now check if the quality of the article is good enough and then decide to add his content.

Norm invocations from inside a group have a stronger effect than norm invocation from outside. The idea to inverse this effect to the contrary was not implemented for the following reason: If a "normal" agent blames a "rebel" agent for being in the wrong group and the effect is inverted, the "rebel" agent would be confirmed in its group choice such that no agent would ever change from one group to another. The idea of a contrary effect might be reasonable in other environments. Before the implementation of this feature (norm invocations from inside a group have a stronger effect than norm invocation from outside) it

[59] More technical details can be found at: http://userpages.uni-koblenz.de/~emil/code_examples/CollaborativeWriting/ and http://mass.aitia.ai/applications/emil.

happened that only a single agent convinces a whole group of agents to convert to the same philosophy as this single agent.

A few more kinds of graphical output were added so the change of values of more variables can be traced more easily. All of the following figures (Figure 83 through Figure 86) refer to two simulation runs, the left-hand graph always stems from simulation run 1, the right-hand graph stems from run 2.Both simulation runs were outcomes of exactly the same model with the exception of the seed of the random number generator for both initialisation and run; thus the differences between the two runs are only due to the stochastic effect, and they show that the final outcome of a run depends sensitively on the initial conditions — in this case the initial group affiliations of the agents.

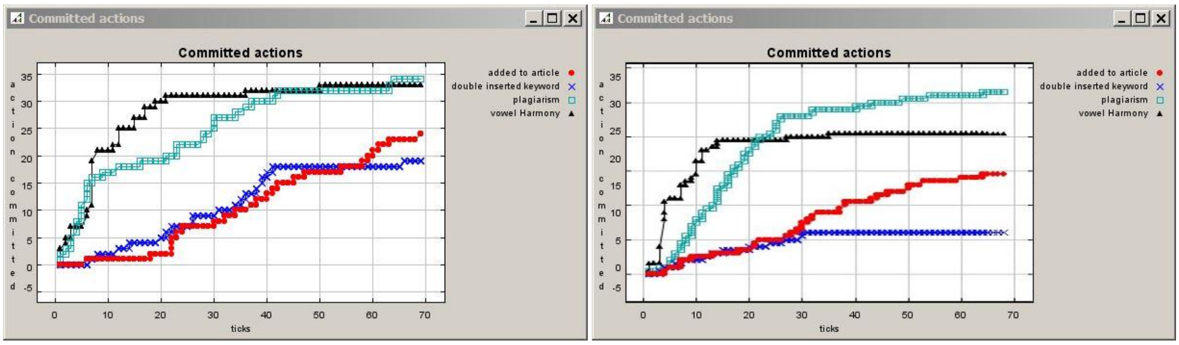


Figure 83. Committed actions

- a graph showing the committed actions (Figure 83) these are the actions which really happened during the simulation run (there is a difference between the actions send from EmIL-S and the actions committed because for some functions the repast layer has a threshold or decision after EmIL-S). These two graphs show how often four different actions were taken by the agents during the two simulation runs.

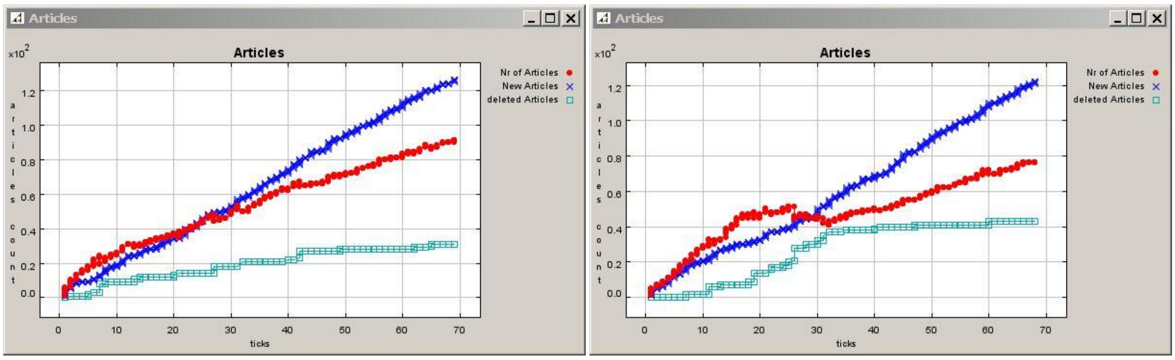


Figure 84. Articles

- a graph showing the total number of articles, the number of newly created articles and the deleted articles (Figure 84). Articles are deleted if they are the result of plagiarism or have been added to the Wikipedia as double articles (two articles with same keyword). The total number of articles is increasing slowly; the deletion of articles happens only when “bad” (plagiarism, double) articles are found. In the simulation run shown in the lower half of Figure 84 one can see a link between time steps 25 and 30, as at this time a lot of double entries and plagiarisms were found and deleted.

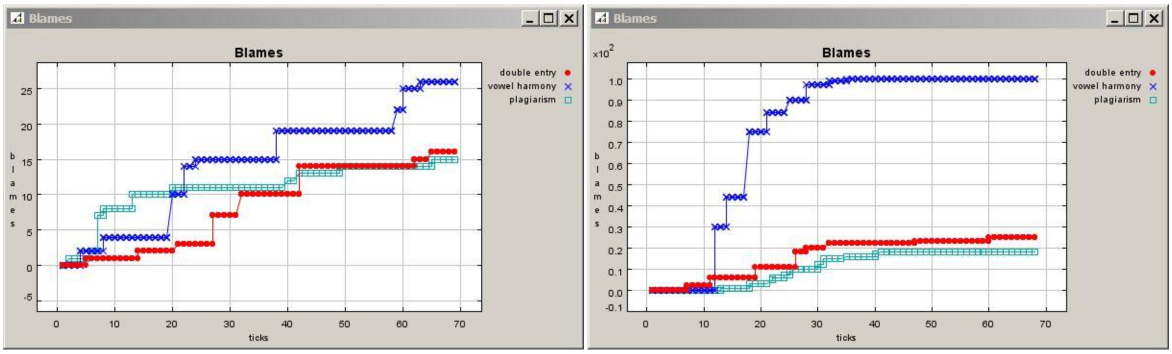


Figure 85. Blames

- a graph showing all blames. Every time an agent blames someone for an action it sends a norm-invocation-message to EmIL-S, and these messages are counted and shown in this graph (Figure 85). One can see a big difference between the two graphs of the two simulation runs. This is due to the fact that on the right-hand side, the simulation started two strong groups (the "normal" ones and the "rebels") which blamed each other. After a short while, one of the groups established itself as the dominating one. In the simulation run represented on the left-hand side, the "anarchists" have gained the victory due to the fact that they are not entirely opposed to the others' philosophies and thus they do not send so many norm invocations. This also leads to a dominating group which is not as stable as the one shown on the right-hand side. Every time an agent moves to one of the other groups ("rebels" or "normal" agents), they send a lot of norm-invocation messages for vowel harmony violation. This is shown in the blue curve. In the right graph this curve rises only in the first half of the simulated time (i.e. there are no more vowel harmony blames during the second half of the simulation run) whereas in the left graph it rises during the entire simulation run (i.e. vowel harmony blames occur through the end of the simulation run).
- a graph showing the groups. In this kind of graph (Figure 86), the current number of agents belonging to a group is counted and visualized. In this graph one can see that in the beginning of the simulation run the agents change quite often between the three groups. At the beginning of each time tick, every agent decides to which group it wants to belong. Initially, this decision is defined by the initial configuration, but this configuration changes due to norm-invocations. After a short period of time the group membership stabilizes. In the second simulation run (right-hand graph) this happens between time 11 and 13; when a particularly high number of norm-invocations for vowel-harmony violation were issued which led to very fast norm learning and consequently to stable group affiliation. As mentioned above, the group membership depends only on the word formation roles the agents comply with.

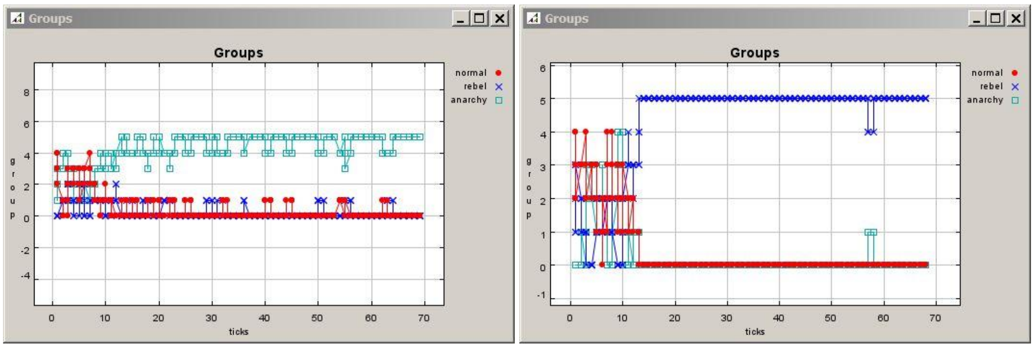


Figure 86. Group sizes

15.5 Conclusion: Comparison between the Prototype and the EMIL-S/Repast Version

The main difference between the two implementations is the strict separation between normative and environmental processes in the latter. Due to the structure of NetLogo programs, it was not possible to make any separation between these two aspects in the prototype as NetLogo programs are always one string of programming language statements (which by the way makes them difficult to maintain and to partly reuse). Moreover, due to the unclear structure of a NetLogo program of more than 800 lines of code, only the EMIL-S/Repast version could be extended to the version presented above which also profits from the findings of the empirical analyses reported in Chapter 5.

Among these differences, the most prominent one is the idea of having several groups whose members influence each other as a member of one group would take offence at others' behaviour in a different way. Another is that additional behavioural features could be introduced beyond the vowel harmony, the double entries and the plagiarism, namely the different decisions to add to earlier articles depending on their "quality" and their similarity.

This made it easier to implement the hypotheses underlying the empirical Wikipedia studies of Chapter 5 .

- The *chance association hypothesis* demanded agents which are initialised inhomogeneously (e.g. with different degrees of being in favour of and against vowel harmony, respectively) — this is why three groups of agents were created: "normal" agents which follow the vowel harmony, rebel agents which follow the inverted vowel harmony and a third group, the "anarchy" agents which have no interest in vowel harmony at all and additionally practise metathesis, creating new variants of words. Working with a real-valued degree of being in favour of or against vowel harmony turned out impossible because this is a binary decision, and setting variable propensities to follow this or another rule has no obvious empirical correlates. Each agent chooses the group at the beginning of each time-tick. If an agent finds a word not obeying its philosophy it blames the author of this word for the "wrong" word and for its group choice.
- *Primed attraction:* Agents adding to existing articles do this only when they find that what they produced is sufficiently similar to the article to which they want to add their material. A measure of similarity had already been defined in the prototype (percentage of words co-occurring), another — "quality" as the number of links to other articles it contains — was added in the new Wikipedia scenario (see next paragraph).
- *Wisdom of crowds:* The quality of an article could be set proportional to the number of occurrences of its keyword in other articles or to the number of extensions by other agents. Agents could then decide to add material to high-ranked (or low-ranked!) existing articles. This is why a field in the articles to specify the "quality" of an article was added. When the article is created, the author checks how many other articles have links to this article and puts this number as a "quality" degree in the article. If someone else now creates a new article with the same keyword this agent checks if its new article is similar enough to the other article. If this is the case it checks if there is more than one and adds its new article to the one with the highest quality.
- *Diffused effect:* Looking at the linguistic competence of EMIL-S agents there was never much hope that material could be produced and linked to this hypothesis. Moreover, other than real-world agents, simulated agents belong exclusively to their Wikipedia community, not to other communities whose influence could interfere with the norms emerging within the Wikipedia community — a problem which cannot be solved in this scenario.
- *Group salience:* Technically speaking this problem was solved together with the one raised by the first — *chance association* — hypothesis (see above), but not with normally or uniformly distributed degrees of being in favour of or against a certain writing style, but with a clear binary decision distribution, but the effect of norm invocation within and across groups had to be modelled differently: Norm invocations from inside the group have a strong effect, norm invocations from outside have smaller effect (an effect to the contrary, envisaged initially, turned out to lead to premature and unlikely stability).

Most of these extensions were only with respect to initialisation and statistical output. Only the extensions mentioned with regard to the *chance association* and the *group salience* hypotheses had to be implemented in a slightly different manner, as agents have to make different decisions according to the group from which an agent's norm invocation comes. Thus the original version of the EMIL-S design turned out to be extremely stable.

Chapter 16 Demonstrating the Theory 3: Micro Finance

Pablo Lucas dos Anjos, Ulf Lotzmann and Manuel Pauli

Abstract

This research is aimed at better understanding how conventional social behaviour amongst microfinance clients influences the success or failure of their groups, in terms of managing debts and defaulters. Efficacy in timely dealing with both aforementioned challenges is crucial, as traditional assets required by banks to backup credit applications are unavailable. Instead, most microfinance methodologies adopt the evaluation of social collateral amongst applicants by using a criterion of group membership and identity, bearing as reference a poverty line. In this case, a sociological and financial analysis has been completed through surveying six hundred microfinance clients, their two thousand four hundred four active loans, thirty-five credit officers plus the microfinance board of directors. The fieldwork in question took place in the southern state of Chiapas, Mexico, from September 2007 and February 2008 and the detailed data analysis was carried out during that period up to July 2009. All findings obtained through this process were discussed with local policy makers and this proved key in providing relevant information for improving the regulatory framework in which the microfinance groups operate. Moreover research findings also guided the development of three computer simulations: a purely reactive model, another declarative one using positive/negative endorsements and an adaptation of the latter using EMIL-S. The first experiment highlights limitations such as sensitivity to initial conditions and path-dependency issues of embedding numerical properties to represent qualitative data. The declarative model does not include any of these in the representation of individual decisions and memory. Instead a data structure chronologically stores events to be analysed according to the patterns unveiled through fieldwork on how clients deal with defaulters. The EMIL-S modelling approach differs from the two other models as, albeit based on the same results obtained in fieldwork, credit groups can be renewed at runtime. Albeit modellers intended to experiment potential uses of simulation beyond consideration of hypothetical scenarios, it has become clear by discussing with the microfinance directors and policy-makers that the current approach has limitations for that matter. Particularly in assessing risks of using inputs provided by simulation results for changes in actual policies. Nevertheless, fieldwork findings have been used during this project to adapt existing policies by taking into consideration the conventional social behaviour influencing groups in southern Mexico.

16.1 Introduction: Target Description[60]

An extensive number of academic and third-sector, i.e. the service industry, publications have been dedicated to discussing good-practices in microfinance. Such papers predominantly focus on economical or managerial aspects that are traditionally relevant either to financiers or policy-makers, usually by involving policy issues with immediate applicable appeal, having political or economical value. Whilst self-sustainability remains a top priority for microfinance institutions (MFI) seeking more leeway in managing their own policies, without overpowering external dependence or pressures, there has also been an increasing interest amongst practitioners in grasping a solid understanding of how social processes can influence success or failure of groups bounded by MFI norms. This is evident when loans are linked to social collateral, e.g. those based on solidarity lending schemes, as then collective responsibility over credit is paramount due to the non-existence of traditional assets to backup repayments. However collecting and analysing behavioural data of microcredit clients is often prohibitively expensive and time-consuming for MFIs themselves, especially when this task is being undertaken for the first time. Despite the likelihood of improving understanding on how clients self-organise regarding defaulting members and debt, there is no *a priori* guarantee that such findings would be relevant for policy-making purposes. This contrasts with financial and socio-economical analyses, as then quantitative evidence is arguably less likely to be strongly

[60] Empirical section of this chapter is based on analysis from working paper (Lucas dos Anjos, 2009c) and reports (Lucas dos Anjos et al., 2008b; Lucas dos Anjos, 2009a, b).

contested than qualitative data interpretations due to existence of retroactively comparable time-series sourced from trustworthy governments or stakeholders. In this sense, real case studies are necessary to introduce the worthiness of adopting guidelines assembled from research methodologies to meaningfully engage practitioners for producing, updating and interpreting behavioural data. Only then established MFI routines tied to –usually tight– budgetary constraints may change with partial guidance of relevant new knowledge developed through research.

16.1.1 Characterising the Fieldwork and the Microfinance Institution

The MFI has been operating in Chiapas for ten years and is a non-profit institution, which has achieved financial self-sustainability after two years of having branched themselves out of the Grameen Foundation using Consultive Group to Assist the Poor guidelines. According to our publishing agreement, their precise identity and location is omitted. Knowing that social behaviour is affected by financial constraints, the fieldwork was set out to analyse financial and behavioural data as essential elements to interpret the key factors influencing the internal management of micro-credit groups. The micro-loans comprise about twenty thousand clients, spread over several geographically distributed groups, each with three to seven women only. Additional services include life insurance in cooperation with Zurich Financial Services, micro-savings, educational and nutritional programs prioritising clients in rural communities (Micro-finance Inc., 2009). Credit officers are trained to facilitate, using Spanish or one of the six Mayan-descendent languages, using finance to support MFI social missions by reinvesting earnings obtained with quota repayments – periodically collected from all financed groups.

The fieldwork was designed to better understand how conventional social behaviour amongst members of micro-credit groups could influence their success or failure over time. As traditional collateral is not used to assess credit applications, it is crucial for the MFI to employ a socio-economic criterion to have an initial evaluation of the social collateral amongst first-time applicants. Credit collateral amongst participants in this case has been assessed according to their socio-economic circumstances, affiliation, economical activities and a poverty line. Some criterion is then adapted once group and individual financial performances become known to credit officers. In this sense fieldwork results served as a complementary source of information regarding how clients' and advisors' act upon top-down rules enforced by board members. The sociological and financial analysis comprised of surveying six hundred Mayan-descendent microfinance clients and thirty-five credit officers with four questionnaires, classifying their respective two hundred sixty one groups plus two thousand four hundred four micro-loans in the Chiapas state, southern Mexico. Merged data from five financial databases distributed throughout MFI operation centres has also been analysed, tracking quota repayments, interest rates, gains and losses for every financed individual. In addition to that, there were important insights on the economic and advice social networks of clients. The main finding included the fact that most rural groups are formed by relatives and social pressure by means of household visits are more effective than group-level fines carried out amongst urban group members, which consists mostly of unrelated neighbours. Surveys also elucidated the client's understanding of ethical and moral principles that actively contribute to enforce the success of collective responsibility, i.e. to fully repay their credit, amongst group members, plus some of the crucial motivations for repayment and defaulting, along with their practical understanding of social collateral via trust relationships developed under the solidarity lending methodology. The complete description of all social and financial findings is available in three fieldwork reports (Lucas dos Anjos et al., 2008b; Lucas dos Anjos, 2009a, b). To help interpreting the plausibility of positive/negative endorsements amongst individuals, it would have been useful to know how much each person contributed when eventual losses occurred in groups, but unfortunately this data has not been recorded by the MFI. Thus, the only way to take this into consideration would be to monitor these events during credit cycles of groups that started soon after the fieldwork begun. It has not been possible to collect data on this matter as funding for this project only covered travelling and pro-rata costs involved in administering the surveys. However MFI directors noticed the benefits of having up-to-date behavioural data, so their new information system now is accounting for how individuals cover losses within groups. Prior to this project the MFI had no record about clients' and advisors' behaviours. This is an important milestone, as it will allow analysis of current, and historical, data about groups' organisation regarding credit rules. Borrowers provide the most important information as solely through analysing their data one can reach conclusions about the debt managing mechanisms imposed to defaulters at the group level during credit cycles. The interplay between institutional norms and bottom-up conventions within groups have been exemplified with evidence of how this two-way process adapts over time, in this case study, as being related to group structures and locations. This has been useful for the MFI to adapt financing policies and to guide development of simulation models to credibly test hypotheses on micro-credit groups with different social and financial circumstances. The methodology, structure and simulation results are fully discussed in (Lucas dos Anjos, to be submitted).

16.1.2 Relating Behaviour amongst Microfinance Clients to Social Norms

Social norms were researched as conventional behaviour, which is a pervasive feature in micro-finance due to social collateral for guiding aspects of group cooperation. For instance, through this research it has been unveiled that institutional norms (effectively obligations on how to repay credit) imposed by the MFI interplay closely with emergent (informal) cooperation and penalisation mechanisms managed entirely by group members. That is, apart from credit rules, there are group-level criteria for assessing and managing defaulters. The fieldwork proved particularly effective in better understanding such behaviours as MFI board members interpreted these findings as valuable assets, mainly as it became apparent how their top-down policies help facilitate or disrupt inner group organisation. All results from the fieldwork described hereby have been summarised in three reports (Lucas dos Anjos et al., 2008b; Lucas dos Anjos, 2009a, b), a how-to guide for using the declarative simulation model (Microfinance simulation model: HowTo guide, 2009) and presentations given in Spanish to stakeholders in Mexico at the National Autonomous University. All these seven documents have also been anonymised, due to our publishing agreement, and translated into English for discussing with research collaborators at the Centre For Policy Modelling in Manchester, ETH Chair of Sociology[61] and Modelling (SOMS) in Zurich and the World Institute for Economic Development Research in Helsinki. Despite being central to microfinance, literature reviews suggests that most institutions do not have thorough understanding of how conventional social behaviour amongst group members can affect their collective responsibility to manage regular instalments. This happens partially as collecting and maintaining this type of data is time-consuming and drains resources from other MFI areas that often deserve prompter attention. Analyses of microfinance clientele are valuable to financiers, as these typically provide relevant new insights with pragmatic usefulness to them. Nevertheless if an MFI has never systematically collected behavioural data, administering questionnaires to many hundreds – or thousands – quickly enough to allow timely analysis and presentation of findings can be troublesome. This is particularly evident when institutional resources have been stretched close to full capacity with other needs, as then a financial portfolio risks extra expenditures if resources have not been well planned. As previously mentioned, albeit potentially useful, still there is no guarantee of research producing knowledge to stakeholders that is directly relevant for policy-making purposes. One can also notice in the microfinance literature that there are few detailed discussions on the internal management mechanisms amongst microfinance clients, where the so-called Solidarity Group[62] concept is central. Considering that little is known about how penalization and group liability are managed by clients themselves in these normative and cooperative processes, understanding conventional social behaviour in detail require taking cultural contexts into account. As exemplified in the administered fieldwork, sanctions (in form of social pressure or group-level fines) and cooperation (covering losses) amongst micro-credit clients are in this case directly related to group composition and understanding of what is considered acceptable behaviour (Lucas dos Anjos, 2009a). Knowing that credit conditions (18%), covering losses and mutual supervision (each with 16%), abiding to group rules and mutual monitoring (15%), supporting who is sick (50%) are the most influential factors for repayments (Lucas dos Anjos et al., 2008b), facilitated the specification of behavioural patterns adopted by most successful groups. According to surveyed clients, the order and structure of social conventions adopted by them to manage defaulting members starts with evaluating two events: missed meeting or missed payment. The former does not imply defaulting, but if so

[61] The author thanks to ETH SOMS for the funding granted for carrying out the fieldwork in Mexico.

[62] Solidarity group decides covering losses from other members collectively, with or without fines.

participants may cover the outstanding amount to avoid an MFI fine. Group members assess events and these may lead to social pressure, by visiting the household of who has not paid on time in rural areas. Such occasion involves only the group leader or all participants. Group-level fines are most frequently charged amongst urban clients depending on the evaluation of what has triggered a default. In case negative cycles repeatedly affects the finances of a group, both rural and urban members may be considered for expelling (Lucas dos Anjos, 2009a).

16.2 Scenario History: The Reactive and Declarative Models

Fieldwork findings guided the development of three computer simulations: a purely reactive model, another declarative one using positive/negative endorsements and an adaptation of the latter using EMIL-S. The first experiment highlights limitations such as sensitivity to initial conditions and path-dependency issues of embedding numerical properties to represent individual qualitative data. The declarative model does not include any of these in representing individual decisions and memory. Instead a data structure that chronologically stores events are analysed according to the patterns unveiled through fieldwork on how clients deal with defaulters. The EMIL-S modelling approach, discussed further in this chapter, differs from the other models as, albeit based on the same results obtained in fieldwork, credit can be renewed at runtime. Albeit modellers intended to experiment potential uses of simulation beyond consideration of hypothetical scenarios, it has become clear by discussing with the microfinance directors and policy-makers that this approach has limitations for that matter. Particularly in assessing risks of using inputs provided by simulation results for changes in actual policies. Nevertheless, fieldwork findings have been used during this project to adapt MFI policies by taking into consideration the conventional social behaviour influencing groups in Chiapas.

16.2.1 Configuring the Declarative Model

As described in (Microfinance simulation model: HowTo guide, 2009), on June 7th 2009 the simulation model and a how-to manual were presented to the MFI directors for testing purposes. Once loaded the simulation interface contains eight graphs and twelve parameters, described below in Table 4 to Table 6.

Property	Description	Range
Rural	True for a rural group, otherwise urban.	Boolean
MFI-Group	How many participants in a simulated group?	3 to 7
Bad-Investors	How many people can be affected by bad investments?	0 to 7
Unprofitable	How many people can be affected by being non-profitable?	0 to 7
Disease-Incidence	Percentage of people and payments subject to disease.	0% to 100%

Table 4: **Group circumstances**

The total parameters' range includes an MFI-Group between three to seven (according to group size distribution findings), twelve or twenty-four payments (the two options provided by the MFI), rural or urban, number of unprofitable clients (zero to seven), bad investors (zero to seven) and disease incidence (zero to hundred percent in decimal intervals). Respectively that means there are 5, 2, 2, 8, 8 and 11 possible configurations per parameter. All possible combinations amount to 14.080, without considering financial parameters. The latter, listed Table 5, includes: interest rates (from 0.5 to 3.5 in half-percent intervals), min-max debts (from 1000 to 40.000 in centesimal intervals) with equal or unequal distributions. Sweeping through all parameters is impractical and unnecessary, as many combinations are unrealistic. E.g., the more unprofitable or bad investors added to a simulation, the bigger the microcredit group must become to allow coping opportunities, which reduces the number of testable parameters. Additionally some configurations are mutually exclusive. I.e., it is not possible to have a micro-finance group in the declarative simulation model whereby all members are problematic. Groups alike in reality are most likely to be refused credit or fail.

Group size and repayments change the scale of simulation results as the total number of events depends on how many people constitute collective credit, potential problems that can affect them, meetings to attend and quotas awaiting repayment deadlines. Parameters such as MFI-Group, Bad-Investors, Unprofitable and Disease-Incidence cited in Table 4, along with Repayments in Table 5, influences how many events can be logged per simulation run.

Property	Description
InterestRate	Interest rate for the total individual debt.
EqualCredit	Will all participants deal with the same amount of credit or not? (Boolean)
MaxAgentDebt	Maximum individual debt, in case credit is not uniformly distributed.
MinAgentDebt	Minimum individual debt, in case credit is not uniformly distributed.
Repayments	How many meetings, and therefore outstanding quotas, each person has?

Table 5: Financial parameters

If EqualCredit is off, financial properties can affect participants in different ways. I.e., if an individual with higher debt eventually misses a payment, covering that without penalisations is more difficult. This is a conjecture drawn from the evidence suggesting greater behaviour variability between those with unequal credit distribution. Assessing the feasibility of such logged actions are left to the end user. Choosing an uneven credit distribution means debts will be allocated randomly between the range of MinAgentDebt and MaxAgentDebt. The MFI itself has recently started with an uneven credit modality and do not yet have enough experience regarding how clients behave in this new scheme. Each simulation generates a detailed log of what all clients decided about a problematic event during their credit cycle. Thus depending on the parameters like debt range, one can interpret individual logged actions as being plausible or not. This evaluation cannot be automated in the model without biasing the simulated behaviour, as at some point the modeller must define a hypothetical threshold and there is no known reliable evidence on this regard. If EqualCredit is set true, clients will get the same amount of debt having about 33% of MaxAgentDebt, according to such significant frequency found by analysing the MFI financial data (Lucas dos Anjos, 2009a). The last configurable properties are listed in Table 6. These parameterize functional properties of running the model. These include setting the simulation initial state (Init), whether the user wants to visualise updating graphs (Plots) after Iterations are ordered for execution (Go).

Property	Description
Plots	Update all interface graphs at runtime
Init	Initialise the model with all selected configuration parameters
Iterations	How many times the model will be executed with the same configuration?
Go	Execute simulations, plotting data graphically or simply writing it to a file

Table 6: Configurations regarding the simulation process itself

If the model is configured without any exogenous or endogenous problems affecting participants, listed in Table 4 and Table 5, the only possible result from simulations is a continuum of micro-credit groups classified as having equal number (zero) for desirable and undesirable events. Such individual registering is only triggered if a group member missing a meeting or payment. This happens as it is assumed that, without any negative interference, groups are most likely to succeed over time. Of course this is not always the case in reality, but without adding any known negative influence to the model –gathered during this case study's fieldwork, nothing relevant from clients sanctions and decisions can be processed or logged during a simulation. Financial aspects such as interest rate, equal credit distribution, min and max individual debts are relevant for interpretation purposes, as the simulation model simply keeps track of social behaviour amongst group participants and logs relevant financial aspects. When social behaviour in this model has a financial dimension, such as groups or institutional fines, assessing whether involved sanctions or supportive actions are feasible outcome is an end-user task. In other words, the stakeholder, researcher or policy-maker using the model should interpret that. Adding an automatic evaluation of these in the simulation would considerably bias and therefore change simulation results. It would also complicate interpreting data generated by the model, as then the simulation would not be exclusively based on the consistent evidence acquired in fieldwork. Instead, there would be an unfounded influence introduced by the modeller which end-users would not be able to differentiate by analysing obtained results. This is important as the declarative model was built according to an evidence-driven social simulation development methodology (Lucas dos Anjos, 2009d).

16.2.2 Interpreting Declarative Results

Groups with more desirable events, more undesirable events and equal number of those divide results per simulation. It is impossible to precisely assess whether groups would have actually been successful or not, as simulation results rather imply their likelihood of failing or succeeding by how their group dynamics evolved during each run. The Microfinance simulation model: HowTo guide (2009) describes how the model can be used and interpreted, skipping all technical details to focus solely on MFI directors experience as an end user. They could easily access the model via a webpage featuring a Java applet with the complete simulation. A full simulation run with a large number of cycles can be a very lengthy process. It can take hours to compute all interactions and generate a complete log file. For instance, running 10.000 times simulations varying an urban group size from 3 to 7, number of unprofitable clients from 1 to 6 and two repayments deadlines (12 and 24) generates about 4.67 GB of data over days of supervised execution. Running fewer times is considerably faster, but in this way it is not possible to analyse results in scale. Each time a simulation is run, there is a detailed report of which configurations were used, time-series of all individual events during credit cycles and a summary of the most important data in it. Even before testing the simulation, it is possible to understand that some combined configurations play a significant role in how results are obtained. E.g., group size and repayments alter how many opportunities clients will have to log decisions about dealing with potential problems. If both group size and repayments are configured lowly, say unrealistically both set as 3, results will differ significantly from other simulation runs with greater number participants and longer deadlines simply as the number of analysed individual events will change accordingly. In this sense, it makes sense to narrow down testing to a handful of realistic parameters instead of trying all feasible combinations.

ElapsedTime and IndividualDebts graphs in the simulation model show the dispersion of PersonalDebt and QuotasWithInterestRate, to facilitate the understanding of cluster variability obtained in different simulations runs. All other graphs plots totals of individually logged events throughout simulation runs depending whether final results contain more Desirable or Undesirable endorsements. In case these are just the same, the Equal graph is plotted. These last 3 include the following data: LeaderVisit, GroupVisit, CoveredLosses, TotalSupport, Total-Fines and ExpellingVotes. It is not possible to determine whether groups with more Desirable events actually had financial and social success, and that those with more Undesirable ones have necessarily failed. This is uncertain as the criteria for this type of interpretation depends on real circumstances of each microfinance group. The simulation model presented hereby can only suggest that groups presented in the Desirable graph, for instance, have greater likelihood of achieving social and financial success. As according to their initial state and evolution till the end of a full credit cycle, clients have dealt with problems in more similarly to the characterisation of successful groups presented in (Lucas dos Anjos, 2009a; Lucas dos Anjos et al., 2008b). The LeaderVisit occurs when a client missed a meeting for the first time, but has paid her quota. According to the fieldwork findings this is the most common action in such occasions. Usually the group representative visits the affected person and regards this fact positively if the affected person is sick, otherwise negatively. GroupVisit events are registered only after the group member in question has already missed payments during the same credit cycle. TotalSupport event accounts to two different types of support available within groups. That is, those indicating CoveredLosses that are either a PARTIAL_SUPPORT event (including group fines) and SUPPORT in case there has been enough tolerance to waive collective fines. Fines usually do not appear in the graphs are these are generally rare events, only occurring occasionally due to their strong impact. The model contains in total 24 types of events, but not all them are necessary to interpret results in graphs, as many of these are auxiliary to the 6 chosen as the most representative ones for analysing the simulation-generated data.

16.2.3 Issues Surrounding Research Evidence in Policy-making

Currently the most frequent scenario is the unavailability of behavioural datasets or the lack of a systematic management of such data by MFIs. As discussed in (Lucas dos Anjos, 2009d), datasets often are: (i) non-existent, hence requiring funding to collect and process information, (ii) unavailable due to privacy agreements or, (iii) if at hand, incomplete or outdated. Even when MFI professionals are aware of influential social conventions –or norms– amongst micro-credit group members, evidence tends to consist of sparsely scattered anecdotes. Assessing these is difficult as their scale is frequently unrepresentative in comparison with the actual social phenomena and validating such information is unreliable due to the unknown circumstances in which it was obtained. This is an enduring issue reverberating both within the MFI and social simulation communities when stakeholders are not part of the research process. It is important to notice that policy-making is ultimately a political and non-linear process, where examples of unfounded evidence being taken into serious consideration by decision-makers abound. It is key to approach the challenge of engaging with practitioners to communicate and use knowledge obtained through research (Jones et al., 2009; Young and Mendizabal, 2009). Stakeholder participation is perhaps the most important aspect to meaningfully engage them in research without a patronising sense of being lectured by theory-driven academics. Petty controversies surrounding data reliability and other irrelevant theoretical considerations can play counter-productive roles in developing dialogue between practitioners, policy-makers and researchers. It is important thus to observe that large and small-scale impact in policy through research findings usually unfolds in significantly different ways. The former indirectly encompass a long process of knowledge accumulation from different sources, often not explicitly linked, which eventually lead to long-term policy influences. This also occur within the context of large, countrywide or supranational projects aimed specifically at evaluating the impact and, if needed, also changing the *status quo* of existing policies. On the other hand, small-scale impact tends to involve commissioned projects aimed at clarifying issues of local relevance, typically dealing with shorter time-spans and order of magnitude.

Perhaps one of the best modern examples of difficulties, encountered both in small- and large-scale research roles in governmental policies, is embodied by The Intergovernmental Panel On Climate Change (IPCC). Despite widespread dissemination of influential research since 1988 via assessment reports backing the directives derived from the 1992 United Nations (UN) Framework Convention on the Climate Change treaty, until the present date after the 2009 UN Climate Change Conference, the IPCC still endures controversies involving scientists regarding their evaluation processes and data analysis on how research findings inform the development and discussions of new environmental policies. Similar controversy occurs in whether financial aid from developed countries is effective for improving living conditions in developing ones. Albeit widely published and acknowledged evidence on the overwhelmingly positive impact of aid, there are long-standing theoretical discussions and academic arguments politically entrenched on this matter for decades (Tarp, 2009). These two examples, despite not dealing with behavioural evidence, convey some key communication difficulties between researchers, practitioners and executives. Progress is sluggish when politically biased interpretations of unrepresentative or incomplete data is used to fuel controversy. As one of the primary missions of micro-finance is to invest profits in solutions for social or environmental problems, good-practices development involving stakeholders, policy-makers and researchers are crucial. These include in-depth problem understanding, data analysis, action planning and policy evaluation. Both in the sense of requiring a stable political decision-making leadership with resources to apply research findings, and effective researchers network using able to clearly communicate analyses to non-academics (Jones et al., 2009; Young and Mendizabal, 2009).

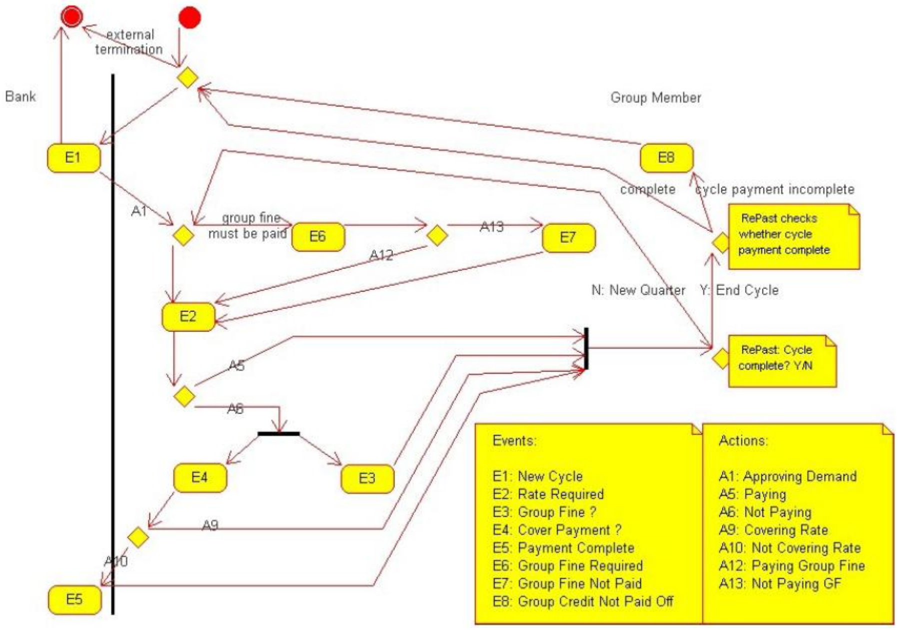
16.2.4 Final Considerations

Understanding the emergence and consequences of conventional social behaviour amongst microfinance clients proved useful both academically and for policy-making purposes. That became apparent once knowledge of how groups with different configurations tend to deal with debt and defaulters in order to avoid penalisations from the MFI. Despite the financial databases in this case study having detailed information tracking payments according to rural and urban loan interest rates, no useful insight on what is relevant for group members could be gained to understand influences on essential micro-credit events such as defaulting and covering losses. Better understanding of these issues was provided through fieldwork, which also influenced changes in funding policies plus development of simulations. Even if it seems insofar unviable using these models to assess and guide the management of an MFI, the declarative model has been able to provide behavioural data on hypothesis that before could simply not be considered in discussions held by MFI directors. Given the clear gap of studies analysing the internal structure and supporting mechanisms of micro-credit groups, this research project focused precisely on these issues, bearing in mind the determination to produce results that are useful within and outside academia. Thus the main contributions are twofold: (1) a socio-economical analysis of borrowers from a mid-sized microfinance institution, including the role of conventional social behaviour in managing micro-credit groups that influenced policy-making (Lucas dos Anjos et al., 2008b; Lucas dos Anjos, 2009a, b), and (2) the embedding of fieldwork findings in a simulation model allowing MFI directors to interpret social and financial consequences of testing different loan configurations (Lucas dos Anjos, to be submitted). Influencing a mid-size micro-finance funding policy that benefited about twenty thousand clients using behavioural and financial research complements known good-practices in micro-finance that collective credit shared by clients with similar socio-ethnical identities is key to voluntary cooperative behaviour (Anthony, 2005).

The declarative microfinance model configures simulate groups with properties such as languages, type of business and their location. Each is configured with one of the following places, extracted from the Mexican IRIS Geographical Information System: Palenque, Zinacantan, Yajalon, Tila, Teopisca, Salto de Agua, Pantelho, Larrainzar, La Libertad, Ixtapa, Escuintla, Chilon, Chenalho, Catazaja, Jitotol, Pueblo Nuevo, Chiapa de Corzo, Motozintla de Mendoza, Pueblo Nuevo Solistahuacan, San Cristobal de las Casas. All clients in the simulation model have Spanish as their main language and, in case of simulating a rural group, each will additionally have one of the following languages: Tsotsil, Tseltal, Chol, Tojolabal, Zoque or Mam. If the group is urban, possible business include: Floriculture, Beans, Coffe, Maize, Animal husbandry, Fruits and Vegetables, Groceries, Bakery, Vehicle parts, Leather Shop, Plastics, drinks, Cheese, Tortillas, Antojitos, cereal, Ice cream, Candies, Sausages, Bread, dishes, Chicken, Dairy, Tamales, Fish, Toasts, Pinata, Hammocks, Handicraft, Carpentry, Catalogue, Seamstress, Cosmetics, Shoes, Clothing and Beauty Salon. These have been classified in categories to facilitate understanding of who is working in similar markets: Personal care, Service, Art, Leisure, Food, Drink and Trade. There is literature suggesting more cooperation between clients with more similarities, yet this simulation only takes into account whether one person is tolerant towards the other and whether they share the same languages. Albeit in reality there is communication between different groups regarding their coping strategies, little evidence is available to guide the modelling of this feature. These extra individual properties allows for easy adaptation of how the model could be tested.

16.3 Scenario Implementation (The EMIL-S Agent Design for This Scenario)

The main difference between the original micro finance model and its EMIL-S implementation is that individual decisions usually are not sufficient to introduce new events. Instead several of these have to be bundled into group decisions which then can trigger a new event in EMIL-S. Thus this scenario led to additional features in EMIL-S, but does not affect EMIL-A as the latter only deals with intra-agent processes and does not take into view that groups could also be subjects of norm recognition and adoption.



Events:

E1: New Cycle
E2: Rate Required
E3: Group Fine ?
E4: Cover Payment ?
E5: Payment Complete
E6: Group Fine Required
E7: Group Fine Not Paid
E8: Group Credit Not Paid Off

Actions:

A1: Approving Demand
A5: Paying
A6: Not Paying
A9: Covering Rate
A10: Not Covering Rate
A12: Paying Group Fine
A13: Not Paying GF

Figure 87. Micro Finance UML model

A micro finance scenario consists of two players: the bank and the loaners who interact with the bank only as a group, no matter whether the interaction is about new loans or repayment. Thus a group is responsible for its loan as a whole and has to repay as a group, eventually compensating for a failure to repay on the side of a group member.. This is why a group can only interact with a bank once all members have made their decisions. These individual decisions are brought together in the EMIL-S version in the Repast part of the simulation, i.e. on the physical level. Figure 87 gives an overview of the overall process, not yet making a difference between the EMIL-S and the Repast parts of the simulation model.

The model works with two time periods of different length, one is a quarter of year, another is called a cycle, namely the period for which the loan is taken. Repayment is done at the end of each quarter; during a cycle no new loan can be asked for. Chronologically, a simulation begins with the event E1which belongs to the bank's initial rule base. The bank decides whether the group is creditworthy or not. If it is, the loan is paid out and a new cycle begins with the first quarter. Otherwise the bank stops the simulation run. In the first case, the next event for each group member agent is either E2 or E6. E6 is the event in which a group member agent is asked to pay a fine for not having repaid in a former round (which is impossible at the beginning of the simulation). E2 is the event when, at the end of the current quarter, repayment is due, and this event is connected to the actions A5 (paying) and A6 (not paying). In the case of paying the quarter is finished from the point of view of the current agent, otherwise two events are triggered for all other members of the group. E3 is the decision to fine the non-paying member, and E4 is the decision whether the group compensates for the non-paying member.

Figure 88 and Figure 89 show the event-action trees in more detail. If all actions have been taken by all group members, it is up to the Repast part of the simulation model to find out whether this quarter is the last of the cycle. If it is, the Repast part has to check whether the loan has been repaid completely. In this case a new cycle can start when the group applies for another loan; otherwise (event E5) the bank asks the group members to pay their debt.

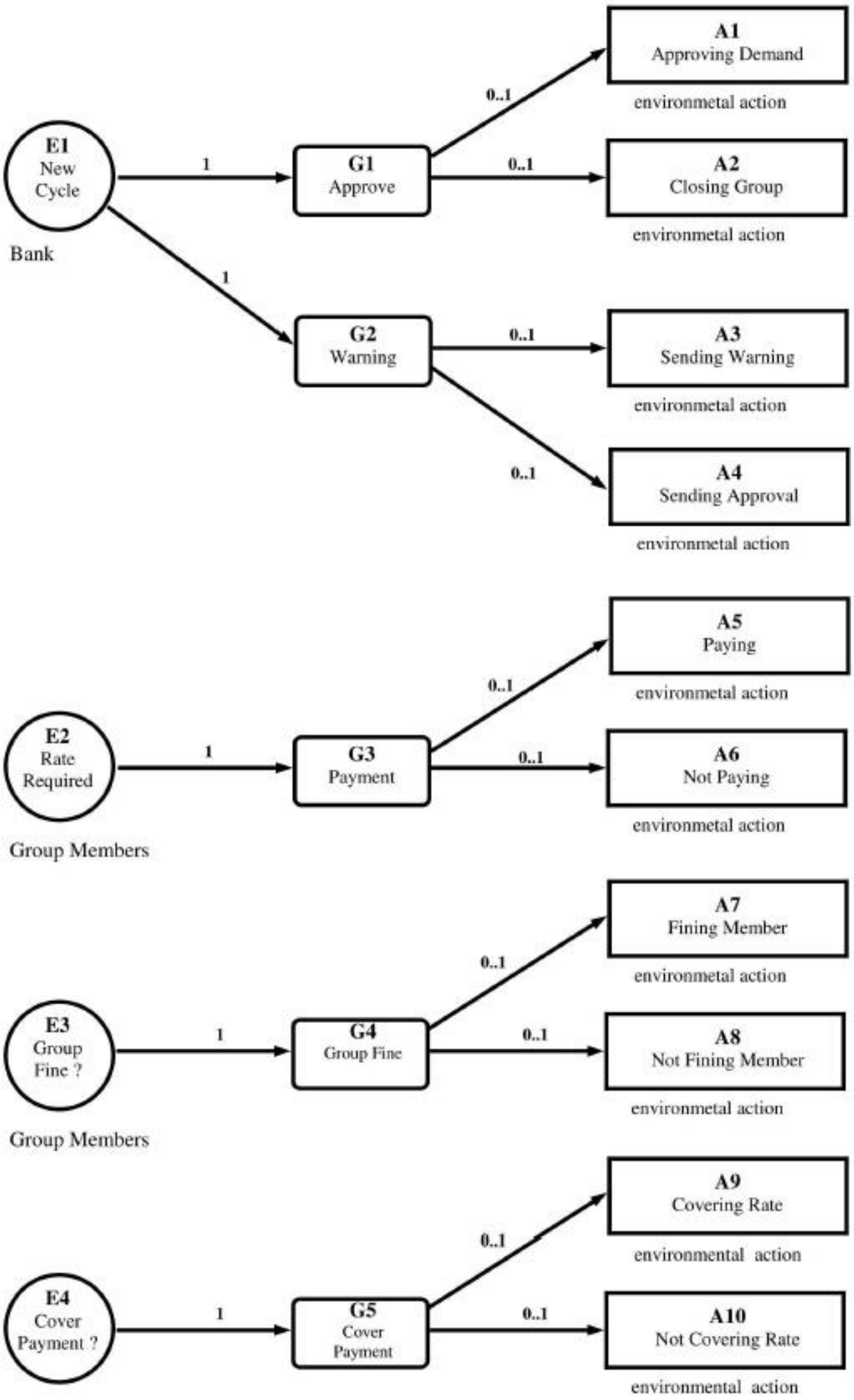


Figure 88. EMIL-S Micro Finance Event-Action Trees (1/2)

As the Repast part of the simulation model is responsible for the communication within the group, it is this part which forms norm invocations, thus if a majority of members have taken the environmental action A7 (fine a member) than the blamable agent is sent a norm invocation which triggers event E6 on which the blamable agent decides whether it pays the fine or not (environmental actions A12 or A13), and this action in turn is reported to the group via the Repast part of the model

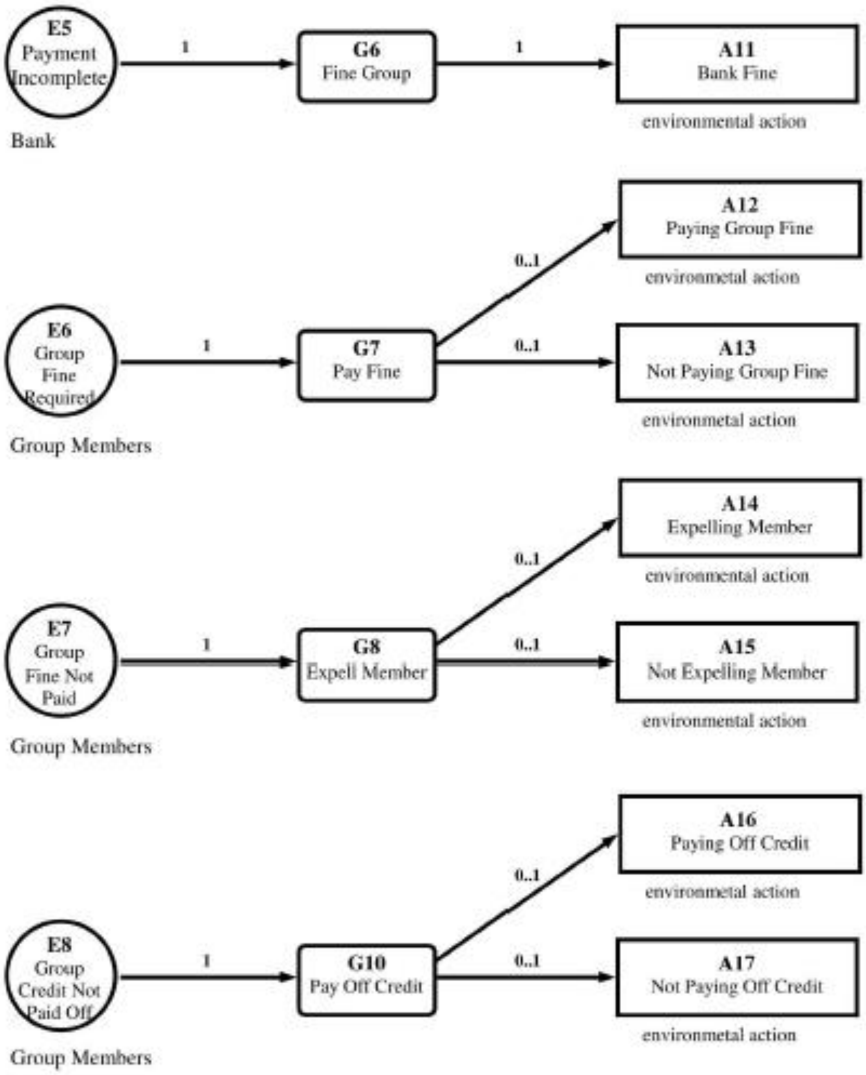
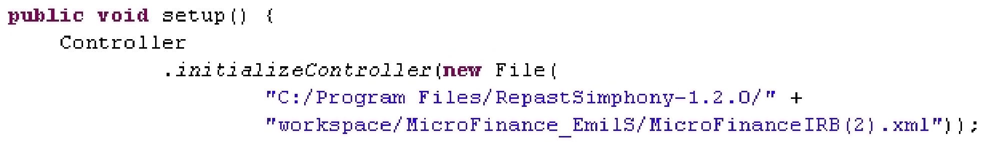


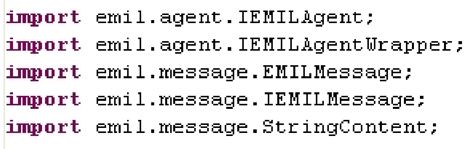
Figure 89. EMIL-S Micro Finance Event-Action Trees (2/2)

In order that Repast and EMIL-S can communicate with each other, the Repast part has to be associated with the EMIL-S part. This is done by means of calling the EMIL-S controller from Repast's `setup` method. This `setup` method is a method of the agent-type independent model class of Repast, and it is called with the path to the appropriate initial rule base of the EMIL-S part:



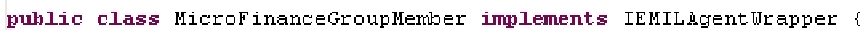
```
public void setup() {
    Controller
            .initializeController(new File(
                    "C:/Program Files/RepastSimphony-1.2.0/" +
                    "workspace/MicroFinance_EmilS/MicroFinanceIRB(2).xml"));
```

Next, every agent-type class import the following from EMIL-S:



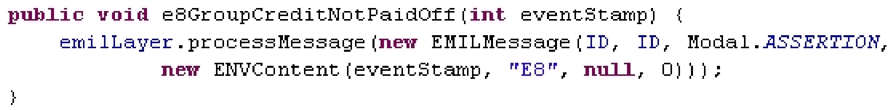
```
import emil.agent.IEMILAgent;
import emil.agent.IEMILAgentWrapper;
import emil.message.EMILMessage;
import emil.message.IEMILMessage;
import emil.message.StringContent;
```

such that each agent-type class can implement the `IEMILAgentWrapper`:



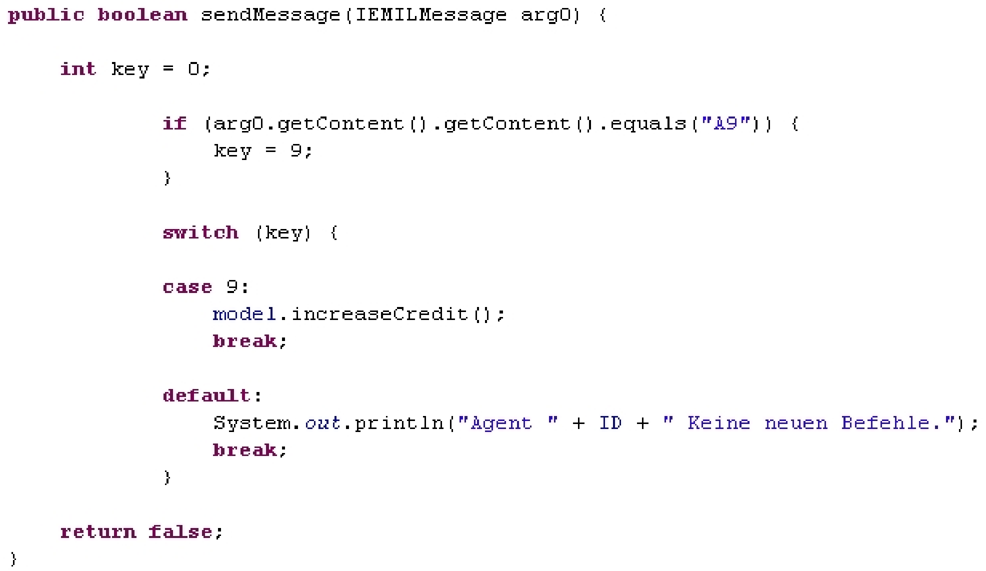
```
public class MicroFinanceGroupMember implements IEMILAgentWrapper {
```

To exchange messages between EMIL-S and Repast, the interface `EMILMessage` is used as follows. To trigger an event n EMIL-S from the side of Repast, a new `EMILMessage` is generated and sent to EMIL-S which contains the modal, the current time and the name of the event:



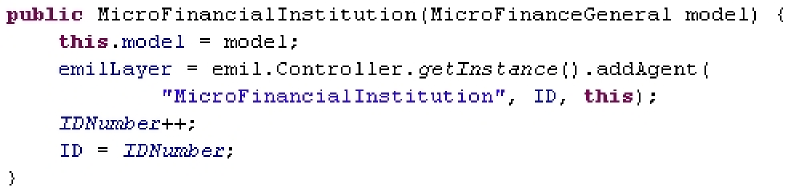
```
public void e8GroupCreditNotPaidOff(int eventStamp) {
    emilLayer.processMessage(new EMILMessage(ID, ID, Modal.ASSERTION,
            new ENVContent(eventStamp, "E8", null, 0)));
}
```

When Repast is to receive messages (environmental actions) from EMIL-S, a method `sendMessage` was implemented which is called by EMIL-S whenever it wants to send a message to Repast. EMIL-S sends a message of the `IEMILMessage` type from which Repast can read the intended action. This is done as follows:



```
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;
}
```

In the end the necessary physical agents can be created as instances of the agent-type classes and associated with the respective EMIL-S agents. To do this Repast calls an EMIL-S method with arguments pointing to agent type, ID and object reference of the Repast agent:



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

Figure 90 shows this in more detail.

After all agents are created the Repast program (which represents the environment and the “bodies” of the agents) needs all those methods which EMIL-S will provide with the appropriate information and which can understand this information in order to generate new events for EMIL-S agents. Thus the Repast program needs one class representing the “bodies” of the group members and another class representing the bank. An additional class is necessary for control and calculation, namely for those tasks which could otherwise have been allotted to something like a “group agent” (but in the real scenario, no group agent exists, and micro finance loaner groups are not so formalised that they have something like a group leader who acts on the group’s behalf).

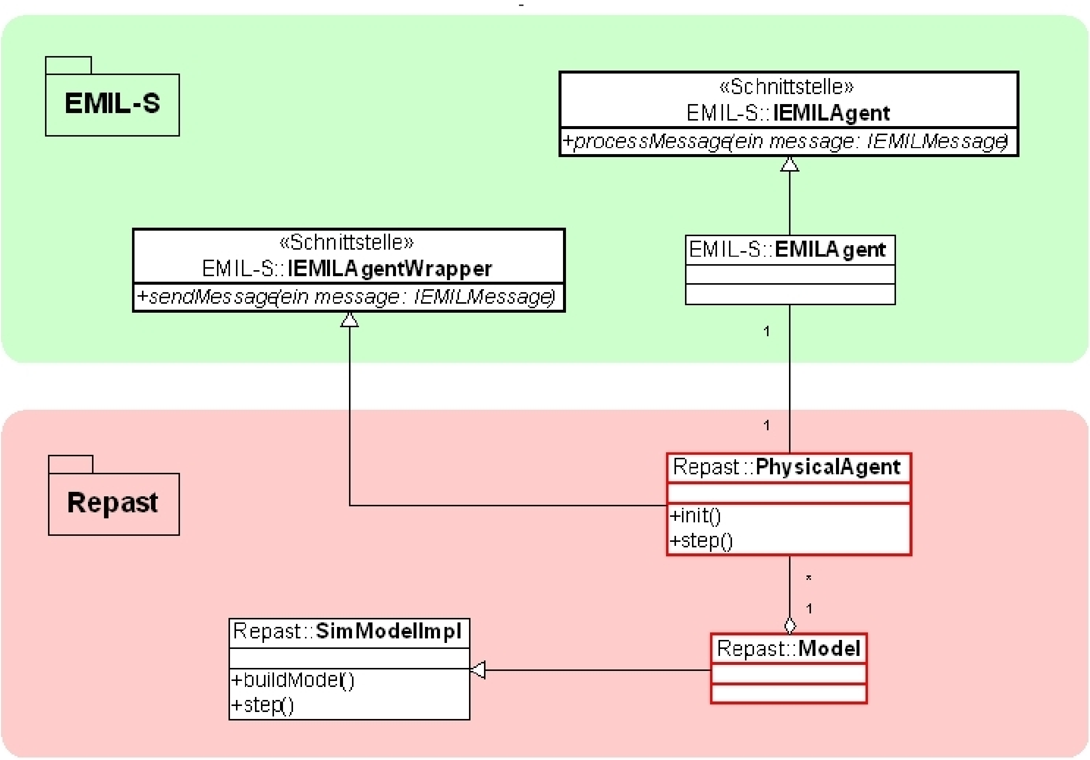


Figure 90. The EMIL-S/Repast interface

In every Repast time step (representing a quarter of a real-world scenario) the following method is called:

```
public void execute() {

    setupCycle();
    displaySurface.updateDisplay();
    setupQuarter();
    quarter++;
    endCycle();
}
```

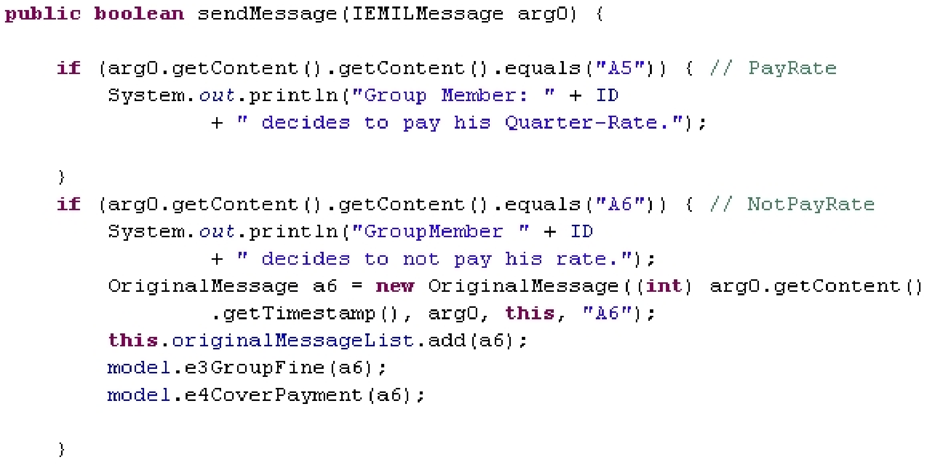
One of the possible events is the one that informs about the next repayment to be made (E2). The following call starts this process. It calls the respective method of the agents of the group-member class which informs Emil-S. Every new event gets a time stamp to make it unique:

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

The group-member agents has an interface which — in the E2 example — looks as follows. The time stamp is communicated to EMIL-S, and as the message is of the environmental type (`ENVContent`), later on EMIL-S's output will also be directed to the environment.

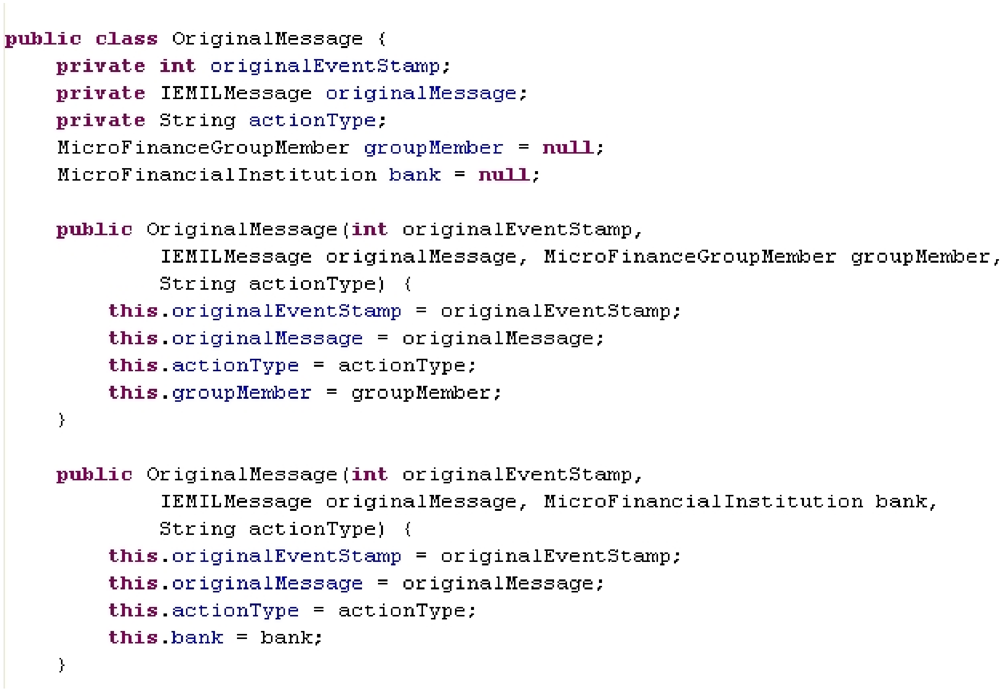
```
public void e2RateRequired(int eventStamp) {
    emilLayer.processMessage(new EMILMessage(ID, ID, Modal.ASSERTION,
            new ENVContent(eventStamp, "E2", ID, 0)));
}
```

This output effected by the `sendMessage` method is transferred to EMIL-S as an argument to `IEMILMessage`. For every possible action that can result from the respective event (according to the initial rule base) Repast checks which EMIL-S response this was, and the corresponding change of the environment is brought about. The following code fragment shows how the actions A5 (paying) and A6 (not paying) are treated. In the A6 case the message sent from the EMIL-S agent is stored in a message list (as it could lead to reactions) such that this list can be used for looking up to which original action the corresponding action belongs.



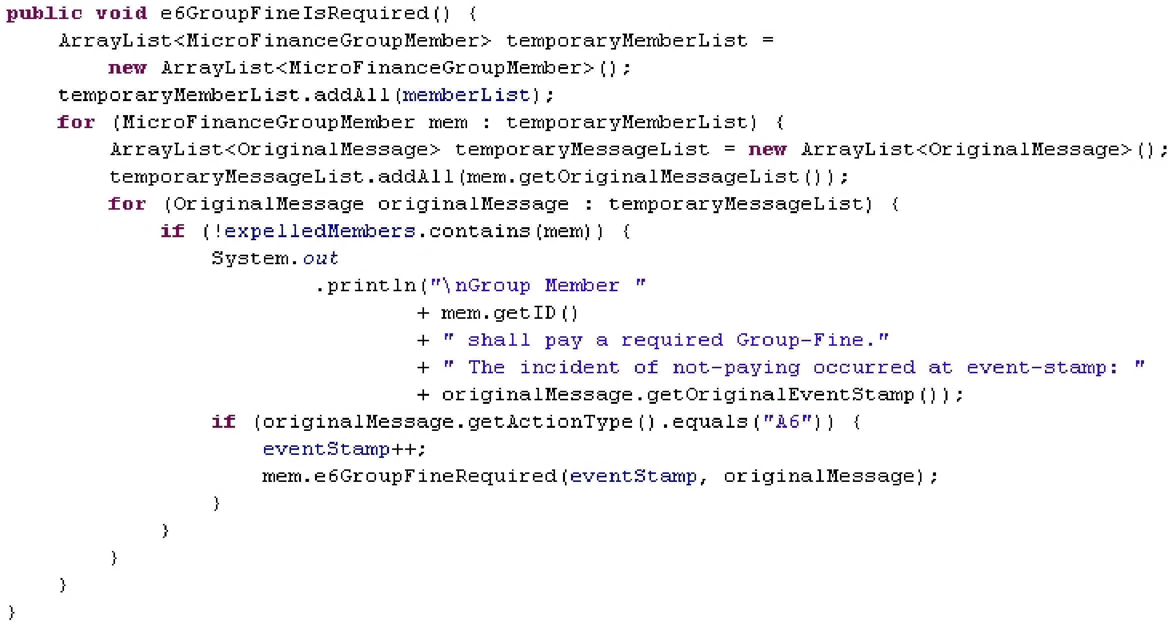
```
public boolean sendMessage(IEMILMessage arg0) {

    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);

    }
```

The format of these messages allows for storing the original EMIL-S message, the time stamp, the action type and the agent type. This message class has two constructors in order to be able to transfer the respective agent type:



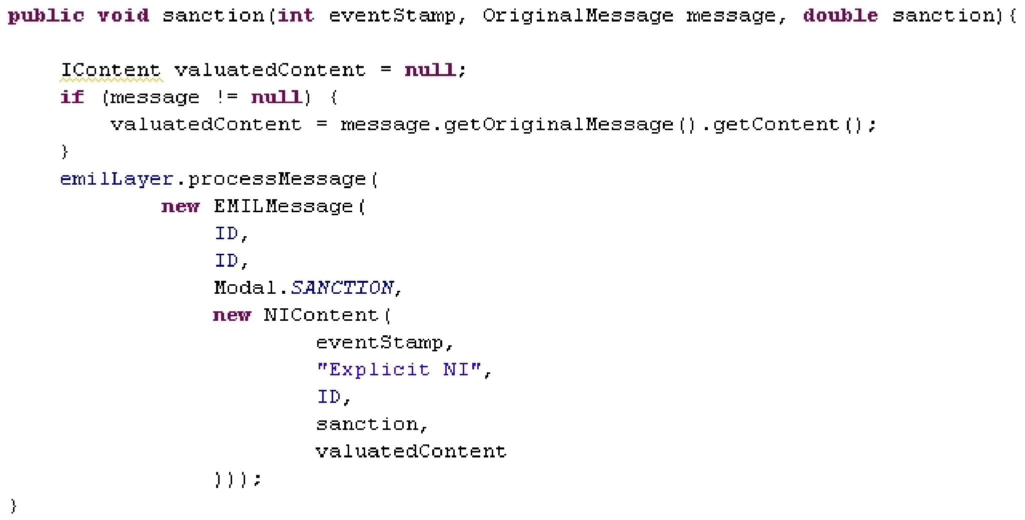
```
public class OriginalMessage {
    private int originalEventStamp;
    private IEMILMessage originalMessage;
    private String actionType;
    MicroFinanceGroupMember groupMember = null;
    MicroFinancialInstitution bank = null;

    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;
    }
```

As is shown in the following code fragment, environmental decisions are made on the base of the stored messages. The inner if statement defines the query of the message list for the existence of a certain message type. If it exists, misdemeanour can be concluded and reproached: the agent is fined.



```
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);
                }
            }
        }
    }
}
```

The misdemeanour does not only lead to a change in the model environment, but also triggers a sanction which enables learning within EMIL-S (i.e. within the “mind” of a fallible agent). If a sanction is necessary the respective method is called for the respective (blamable) agent with arguments describing the decision which led to the sanction and the salience of the sanction:



```
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
                )));
}
```

This explicit sanction now triggers learning in the receiving agent: Probabilities of certain actions will change consequently. As the sanction is associated to the misdemeanour by the implementation helps the agent to learn, but learning could also be implemented differently, with more autonomy of the agent.

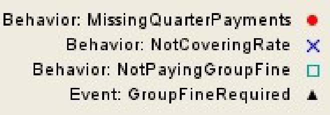
16.4 Simulation Runs and Results

To evaluate the learning behaviour during the simulation certain events are logged over time, and a graph is drawn during the simulation which shows the cumulated number of four exemplary events. In every round (trimester) new events of one of these four types are added to their current number, and these cumulated numbers are then plotted to show by their slope which events were currently particularly frequent (the graphs are normalised between 0 and 1). The four logged events are

1. the decision not to pay back a trimestral instalment (behaviour: `MissingQuarterPayments`); these decision summed over a trimester and divided by the number of group members yields a number between 0 and 1 per trimester and expresses the current ability to pay or creditworthiness of the group.

2. the decision not to cover the instalment of a fallible group member (`NotCoveringRate`); again the number of these decision is divided by the number of the group members (not counting the fallible member) and by the number of unpaid instalments per trimester.
3. the request to pay a group fine (`GroupFineRequired`); again the number of these decisions is divided by the number of group members.
4. the decision not to pay the group fine (`NotPayingGroupFine`); again the number of these decisions is divided by the number of group members.

The division by the number of group members is also necessary to cope with the fact that group members might be expelled during a simulation, absolute numbers of these decision would impair the comparability. Figure 91 shows the legend for the following graphs.



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

Figure 91. Legend for the simulation graphs

The first two events belong together, as do the other two events, as a `NotCoveringRate` must have a `MissingQuarterPayments` as its predecessor, and the same hoods for the `NotPaying-GroupMember` with respect to the `GroupFineRequired`. The difference between these decision frequencies shows the reaction of the group.

Learning in this scenario is triggered by explicit sanction. This is modelled as follows:

1. Whenever an agent decides not to pay its trimestral instalment, its fellow members will decide whether that one will be sanctioned, and thus fallible agent will associate this sanction with its unwillingness or impossibility to pay and adapt its behaviour accordingly. Thus adaptation is done in EMIL-S as programmed in the XML rule set.
2. If less than one half of the remaining group members decide to cover the instalment of the fallible group member and to pay the bank the full instalment, then the bank will punish the group, and all agents will learn from this punishment.
3. If an agent denies to pay the fine imposed on it by the group, it will receive a strong blame from its fellow agents and count this as a sanction connected to its refusal to pay the fine.

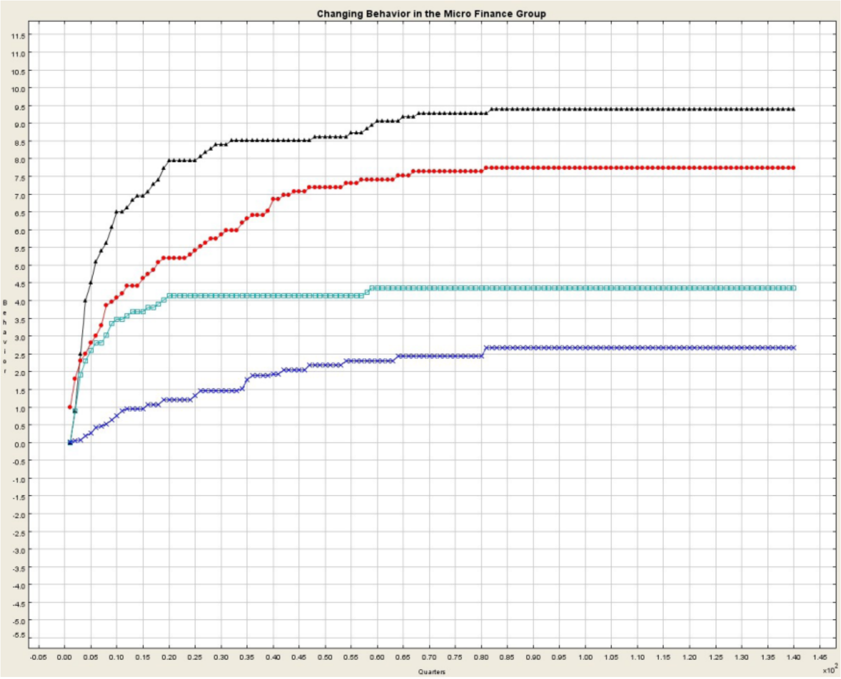


Figure 92. First simulation run

After starting the simulation, a graph as in Figure 92 is opened whose horizontal axis denotes time (in trimesters) and whose vertical axis denotes the cumulated event numbers as described above. This graph can be read as in the following example.

At t=16 and t=19, the event `NotCoveringRate` occurs twice (the dark blue curve increases), during this time span (15<t<20) the other three curves increase in every time step, which means that the events `MissingQuarterPayments` and `NotPayingGroupFines` will occur several times, and the same applies to the event `GroupFineRequired`, until during the next five following time steps all rates are covered (the dark blue curve is now horizontal).

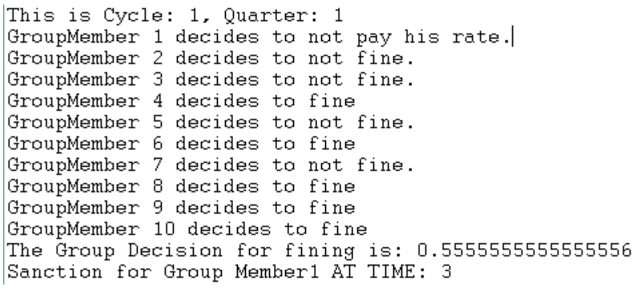
During the first five trimesters the behaviour of the group is more or less constant (the slope of the curves is more or less constant), but even this early learning occurs. Soon afterwards behavioural changes csan be observed (the slope of the curves decreases).

- The red graph shows that missing trimestral payments become rarer.
- The dark blue graph shows that missing payments are covered by the other members of the group more frequently, but this process is rather slow, as this graph is rather flat. This is because a wrong decision is only punished when there is a majority within the group against covering; only then the bank will be informed, and only then the bank can fine the group, and only then those who voted against covering have a chance to learn. And as after only few early punishments the group decision is often slightly above the threshold for covering, learning is rare although a large minority votes against covering. This means that the individual agents have to bear less responsibility for the fate of the group, as their individual decisions have no direct effect on the relations between bank and group.
- The requests to pay a group fine (black graph) become rarer (which is not directly dependent on the less missing payments, as a fine which was not paid still has to be paid in subsequent trimesters. This is why the slope of the black graph is steeper than the one of the red graph.
- The light blue graph shows a rapid decrease of tits slope. Initially group fines are paid only rarely, but after the fifth trimester the slope becomes flatter, and after t=20 only two more `NotPayingGroupFines` events occur (at t= 57 and t=58), and this is the case in spite of still many group fine requests (see the still steep black graph).

After about 82 trimesters the simulation enters into a more or less stable state.

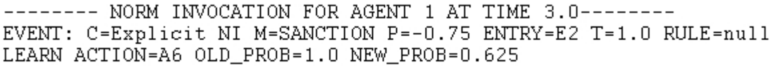
To analyse more precisely what happens during the learning process logs from both Repast and EMIL-S are available. The former contains some output about individual decisions (Figure 93) and events of learning (Figure 94). The latter contains the following information:

- learning is triggered by an explicit sanction,
- the fine was set to –0.75,
- the behavioural rule to be changed is the one connected to event E2,
- the cause of this change happened at time T=1.0,
- action A6 was recognised as the cause of the sanction,
- and the probability to take action A6 was decreased from 1.0 to 0.625.



```
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
```

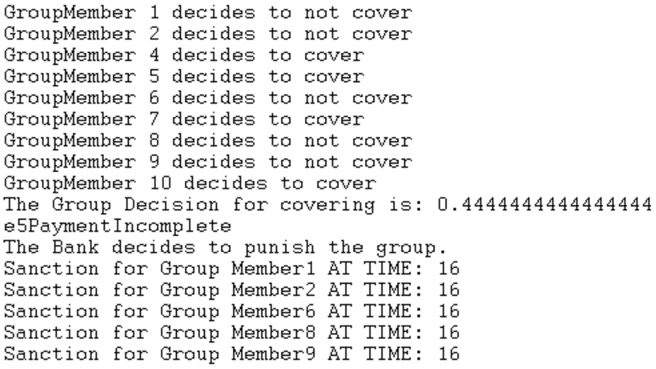
Figure 93. Extract from the Repast logfile of the first simulation run: A nonpayer is punished



```
-------- 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
```

Figure 94. Extract from the EMIL-S logfile: A non-payer is punished

The agent has thus learnt that the event which follows action A6 should be avoided and that, consequently, action A6 should be chosen with a smaller probability.

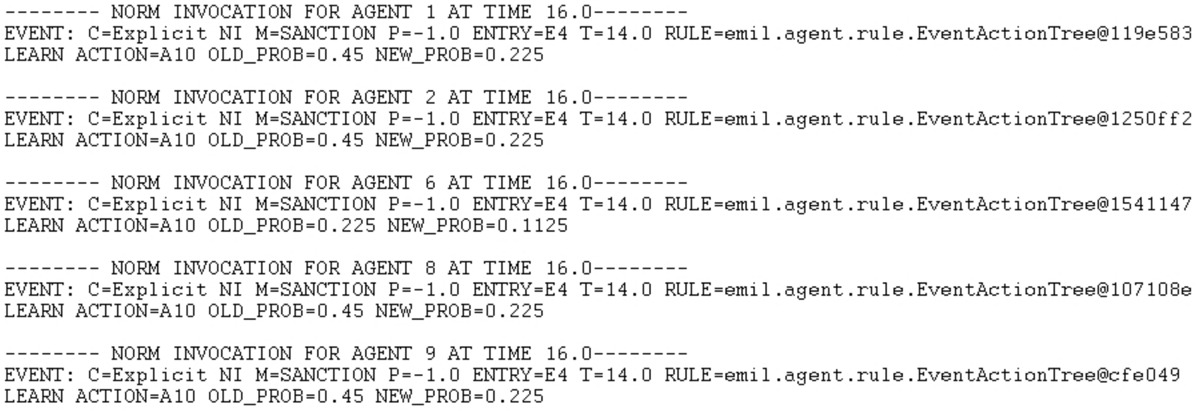


```
GroupMember 1 decides to not cover
GroupMember 2 decides to not cover
GroupMember 4 decides to cover
GroupMember 5 decides to cover
GroupMember 6 decides to not cover
GroupMember 7 decides to cover
GroupMember 8 decides to not cover
GroupMember 9 decides to not cover
GroupMember 10 decides to cover
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
```

Figure 95. Extract from the Repast logfile: The group is punished by the bank

The following figures (Figure 95 and Figure 96) show similarly that learning also affects decisions not to cover unpaid instalments of fallible group members. The difference is here that the bank issues sanctions to all group members.

Figure 96 shows that in this case the change of the event-action tree which depends on event E4 is changed.

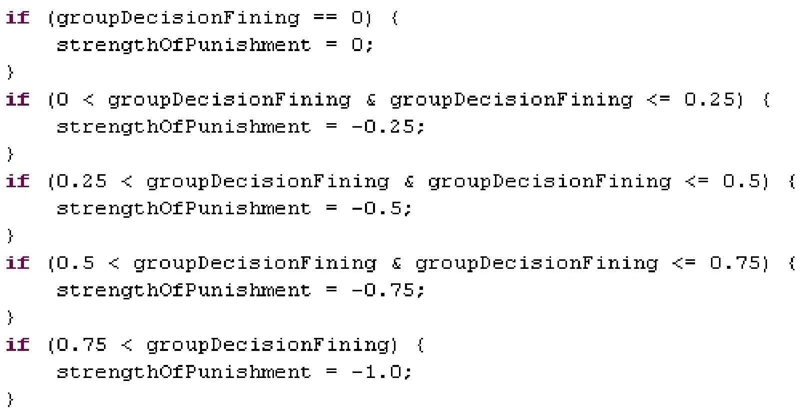


```
-------- NORM INVOCATION FOR AGENT 1 AT TIME 16.0--------
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=14.0 RULE=emil.agent.rule.EventActionTree@119e583
LEARN ACTION=A10 OLD_PROB=0.45 NEW_PROB=0.225

-------- 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
```

Figure 96. Extract from the EMIL-S logfile: The group is punished by the bank

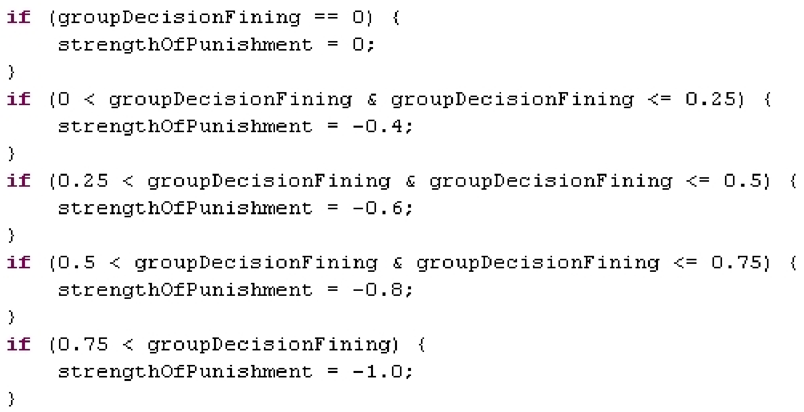
If one runs several different parameterisations of the simulation one can see the consequence of, for instance, different saliences of the sanctions on not-paying have different effects. In the first simulation run reported above, the sanction had the values reported in Figure 97, the following runs use the initialisations from Figure 98.

These new values for the sanction are expected to a faster learning, and in fact, Figure 99 shows that a stable state is already reached after 45 trimesters (instead of 82).



```
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;
}
```

Figure 97. Strength of punishment in the first simulation run



```
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;
}
```

Figure 98. Strength of punishment in the second and third simulation runs

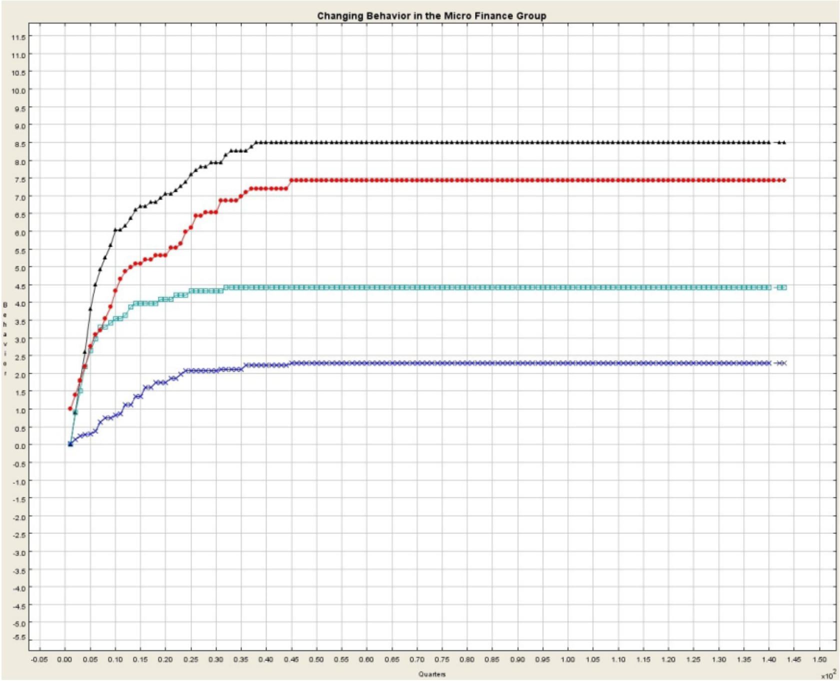


Figure 99. Second simulation run

A third simulation run analyses the rare case that no explicit action is reported to the sanctioned action. This can happen in the current scenario when the bank sends a general encouragement or warning at the outset of new cycle. In case the group owed the bank some amount at the end of the former cycle, this leads to the adjustment reported above, i.e. to learning, but if the groups owes nothing its members do not connect the warning or encouragement to their former behaviour and do not learn. The group as a whole, however, as it is currently implemented reports the warning to its members, and this leads to the misunderstanding which is visible in Figure 100 in trimester 12 (in the dark-blue `NotCoveringGroupFine` graph) and again in trimester 30 (in the red `MissingQuarterPayments` graph). The simulation also shows that the group and its agents recover from these misunderstandings, and a stable state might have been reached but much later than without these misunderstandings (which, by the way, seem utterly realistic!).

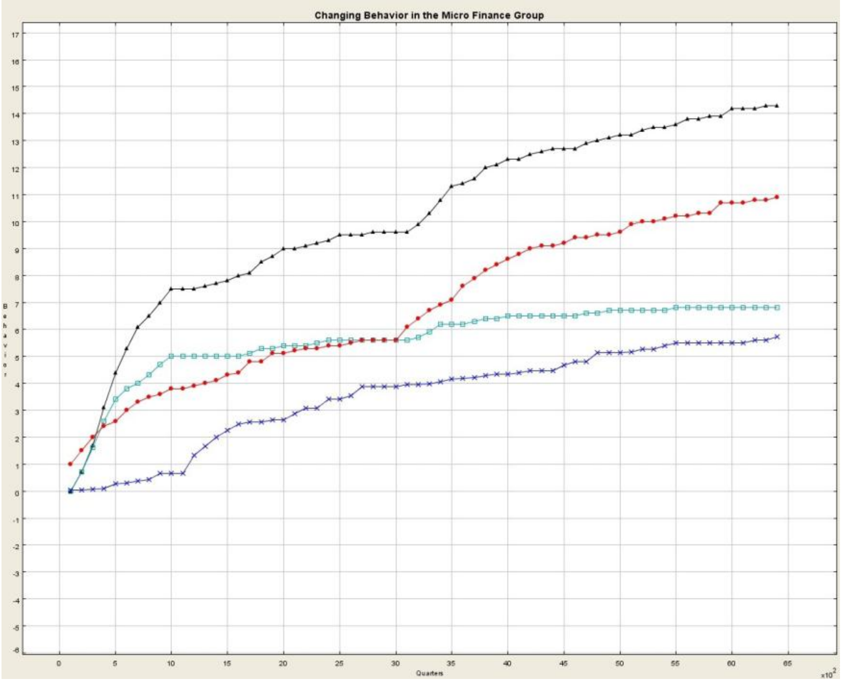
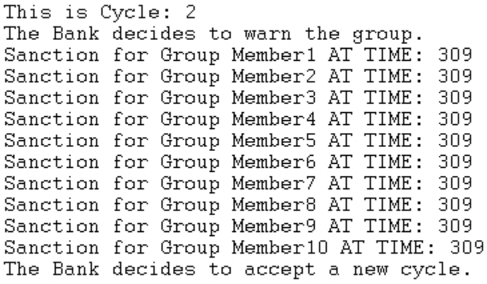


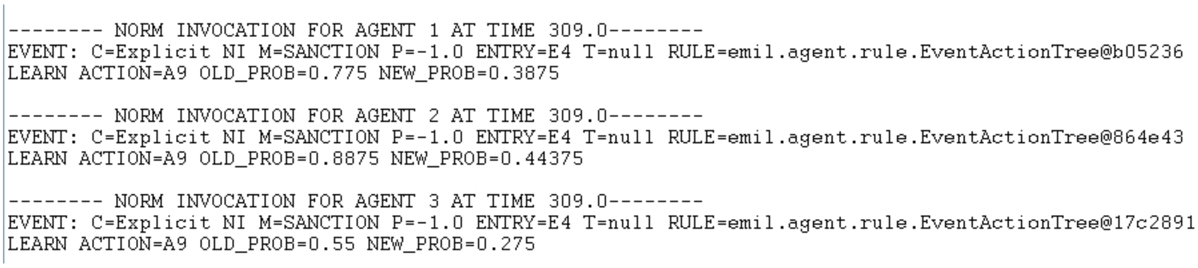
Figure 100. Third simulation run

At the beginning of trimester 10 the log shows what can be seen in Figure 101: The bank sends a warning which is not well understandable for the group members as at the end of its message it says that is willing to accept a new cycle. As Figure 102 shows, EMIL-S connects this warning to the event-action tree which belongs to E4 and reduces the probability of action A9 (covering the instalment) to one half of its former value, such that the group might vote not to pay as it learnt that the warning had no consequences for the loan.



```
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.
```

Figure 101. Extract from the Repast logfile: Sanction to all group members by the bank, first example

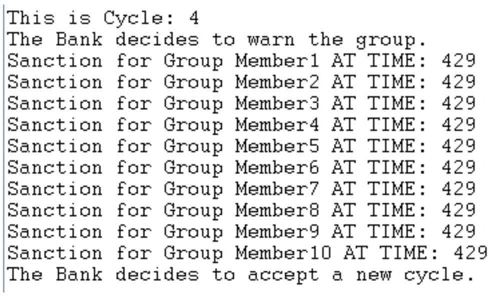


```
-------- NORM INVOCATION FOR AGENT 1 AT TIME 309.0--------
EVENT: C=Explicit NI M=SANCTION P=-1.0 ENTRY=E4 T=null RULE=emil.agent.rule.EventActionTree@b05236
LEARN ACTION=A9 OLD_PROB=0.775 NEW_PROB=0.3875

-------- 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
```

Figure 102. Extract from the EMIL-S logfile: Sanction to all group members by the bank, first example

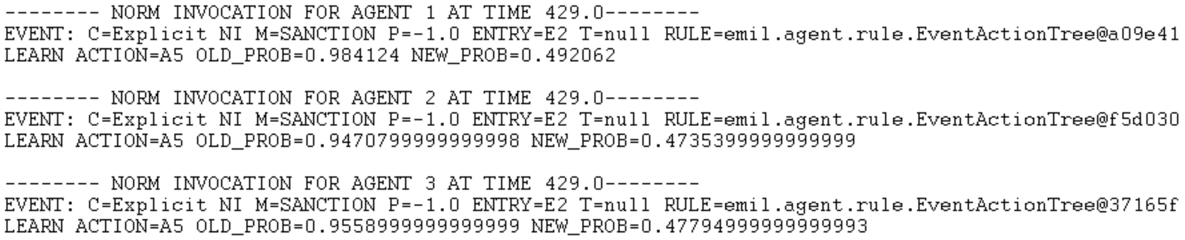
Something similar happens a little later (Figure 103), here the sanction is going to be connected to the event-action tree connected to the event E2.

This leads to a reduction of the probability of paying the trimestral instalment (Figure 104) which leads to an increase in non-payments (the red curve for `MissingQuarterPayments`) at the beginning of trimester 30 (cycle 4).



```
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.
```

Figure 103. Extract from the Repast logfile: Sanction to all group members by the bank, second example



```
-------- 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
```

Figure 104. Extract from the EMIL-S logfile: Sanction to all group members by the bank, second example

Chapter 17 Demonstrating the Theory 4: Multi-scenario World – the Sequel

Marco Campenní, Steffi Henn, Peyman Jazayeri, Ulf Lotzmann, Michael Möhring, Magnus Oberhausen, Mehmet-Hadi Tohum and Jannik Weyrich

Abstract

This chapter describes the EMIL-S replication of the multi-scenario world model already described in Chapter 8. In this model, the agents repeatedly traverse through four different stations of a virtual airport. At each station the agents have to choose between two possible behaviours. The selection process is influenced by

- own attitudes, and
- behaviour of the neighbouring agents.

Two types of agents are defined:

- social conformers, adapting almost instantly the behaviour which is performed by the majority of surrounding agents,
- norm recognizers, who change their previous behaviour only in virtue of norm invocations from other (observing) agents.

Several simulation scenarios – some of which were not considered in the original implementation – are examined and compared:

- the agent population consists either of social conformers or of norm recognizers,
- agents of both types interact (simultaneously) in the same scenario,
- both abovementioned cases are applied within an extended simulation world, where two (or more) virtual airports coexist, and agents are occasionally ported between these airports.

The implementation of the simulation environment (and the physical agent layer) makes use the TRASS framework.

17.1 Introduction: Target Description

Simplifying the highly complex process of behaviour by different actors in the social context, this scenario is a simulation model of the interaction of agents of two different kinds. On the one hand, there are social conformers who strictly react on their environment and copy the commonly observed action of others without recognizing the norm from which the decision is arising. On the other hand, norm recognizers have knowledge about the existing norm in social matters, and their actions are based on this knowledge. Joining the different agents in a simulated environment reveals the consequences of acting based on different principles. In order to achieve this, four scenarios have been developed based on TRASS in which the agents can decide between two actions regarding their principles arising from EMIL-S. The long term results can be analyzed depending on the amount of different agents deployed and the parameter settings.

17.2 Scenario History: A Short Summary of Earlier Versions and a Description of the Extended Model

The basis for this version of the multi-scenario world model is (Andrighetto et al., 2008b).

The aim of this project was to explore the effectiveness of norm recognition and the role of normative beliefs in norm emergence and stabilisation. In this case a normative belief is a belief that a certain behaviour in a given situation and environment is forbidden or permitted.

Therefore the social behaviour of two types of agents, the Norm Recognizers and the Social Conformers had to be analysed and compared.

A norm is specified as a social behaviour within a society, more precisely, the rules which exist for appropriate or inappropriate attitudes, beliefs and in particular social behaviours.

Hence a Norm Recognizer is the agent which is endowed with skills to observe its social environment and to identify what the norm is like. Thereupon they can devise new normative beliefs. Norm Recognizers are able to change their prior behaviour according to norm invocations which they derive from other observing agents. A Social Conformer only detects and adapts the social behaviour which is performed by the majority of surrounding agents.

To find out how Norm Recognizer and Social Conformers operate, the interaction between the two agents in a common hypothetical world is simulated. With this model Andrighetto et al. (2008b) try to verify whether there are significant disparities at the population level between the two kinds of agents. The other important fact to point out is to what extent the skills to identify and to devise normative beliefs are essential for norm emergence and stabilisation.

The normative architecture contains mechanisms which enable norms to influence the behaviours of different autonomous agents. But they can also affect different aspects of mind, more precisely; they affect the processes of recognition, adoption, planning and decision-making. The architecture permits that agents can observe which norm is in force although it is not stored in the normative board. The norm is stored as soon as it is detected by the agent.

Therefore the Norm Recognizer’s architecture is composed of two layers and a direct connection to the normative board. This part of the long term memory comprehends the required normative beliefs and goals, which are classified by the degree of activation in certain situations.

The first layer consists of normative beliefs and goals. To create these, agents have to assimilate received information.

Hence Andrighetto et al. (2008b) define five possible modals. The first are assertions which characterise states of the world. The next model comprises different behaviours. They show agents’ actions or reactions arising from observing the behaviour of other agents in their environment. Another possibility comprises requests which require actions produced by other agents. Deontics are separated situations in the categories good/acceptable and bad/unacceptable. The last modals are normative valuations. These can be propositions about what is the appropriate or inappropriate behaviour in a certain situation.

The second layer is the memory area, where deontics and normative valuations are accumulated.

To choose which action is to be taken, the agents have to search through their normative board, which consists of the normative beliefs and goals. It is possible that more than one norm is found applicable. Therefore the agent will select the most preeminent norm. Agents are able to develop or improve normative beliefs after obtaining input and analysing the received information. A new possible norm is stored depending on whether the received information contains deontic or normative valuations.

These ideas from the original multi-scenario model are adapted for the model described here. Especially the idea of Norm Recognizer and Social Conformers interacting in a hypothetical world and the agents architectures are important as well. The motivation was to implement these ideas in the TRASS/EMIL-S architecture and not in Netlogo as is was done before.

Otherwise several differences exist between the two models. The original simulation model consists of four scenarios which require special situations where the agents can decide between three different kinds of actions. The actions are determined as two context-specific actions and one common action for all scenarios.

To make clear how a scenario and the appropriate actions look like a description of the first scenario follows:

In this scenario the environment is a postal office which has the two context-specific actions “stand in the queue” and “occupy a correct place in front of the desk”. The common action was declared as “answer when asked". All scenarios are dependent from each other to a certain extent because of the common action.

In the model described here we only use three contexts which are set in one environment, an airport. Each context is connected to the previous context. This means, the agents will pass through a context when the previous context terminates.

In the current model, too, there are two context-specific actions, but there is no common action for all contexts because such a common action would not fit in the virtual airport model.

17.3 Scenario Implementation

As mentioned the previous chapter, the contexts are situated in the environment of an airport. The airport is modelled as a closed environment where two types of agents (Social Conformers and Norm Recognizers) can move freely. The agents have to pass various stations, which are derived from real stations having to be passed on a real airport. In a first sketch four different stations were designed as shown in Figure 105. At each station agents have to choose between two alternative actions, one being considered typical for a country or cultural area, one being considered not appropriate by the majority of the cultural area the scenario is taking place in. In a later stage of development it turned out that calling the actions just right or wrong being more appropriate since the differences among the cultures might not be as significant as assumed. The discussion of cultural aspects of the scenario can be found at the end of this chapter.

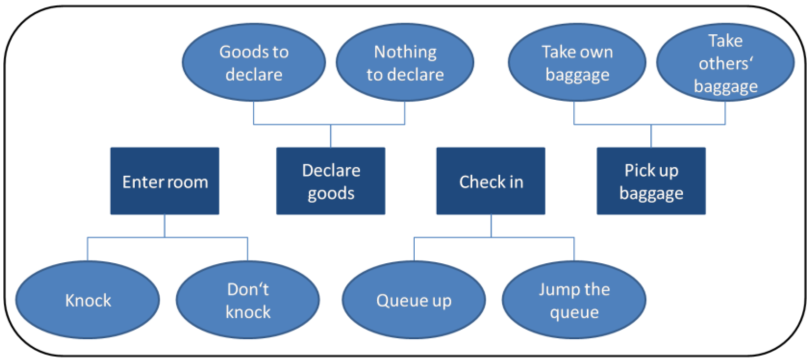


Figure 105. First sketch of the airport scenario

The first action occurs in a scene in which agents have to enter a room, for example a travel agency bureau. At this stage the agents have to decide whether they knock at the door prior to entering the room or rather enter without knocking.

At the second stage the agents have to declare their goods to be exported. Here the agents have to decide to act regarding to the norm by declaring their goods or just pass the customs station and hide their goods. A result could be a penalty (for instance a fee) if undeclared goods are revealed or a reward (for instance a feeling of satisfaction) if the agent passed without being uncovered while smuggling.

The third stage is about checking in for a flight. The decision to be taken here is to stand in the queue or ignoring the norm and going to the front regardless of the people in the line.

At stage four agents have to claim their baggage. This station confronts the agents with the decision of taking just any baggage from the conveyor systems or waiting for their own luggage.

After designing these four contexts, a decision had to be made with which software tool the scenario had to be implemented. The choice was TRASS which was also used for and explained in the traffic scenario chapter. After testing the tool the scenario had to be rearranged because of arising difficulties in modelling the environment and because of concerns about stages being too similar to each other. The baggage claim and customs stations were kept, but the check-in station was converted into a taxi stand where agents have to make the decision to queue up in order to get a taxi or just rush to any car without respecting other waiting agents. The travel agency bureau was cancelled as it would have been too similar to the taxi stand station. The procedures at the remaining three stations have been refined to make the scenario run smoothly on TRASS.

During the modelling process it was necessary to constantly deal with the trade-off between realistic model design and technical feasibility with the additional premise to reveal measurable results in the end. After some testing, a sustainable compromise between the three mentioned factors was worked out:

1. **Baggage claim**: The biggest problem with the design of the baggage claim area was to give the station a realistic look. It was decided to model it in anticipation of receiving good statistical results rather than giving it a realistic look. Agents at this station have to decide whether they claim their luggage after a flight or just abandon the luggage (e.g. to save time or avoid stress). To the agents it seems like leaving the luggage could save time and shorten their walking distance but it was decided that it is mandatory for every agent to claim its luggage. So if an agent decides to not claim the luggage at first, after some time it, however, decides to claim it. Possible explanations that this could be the case in a real world scenario are: (a) the luggage contains the wallet with passport and one cannot exit the airport without showing it, (b) the children's toys are inside the luggage and they complain to get them back, or (c) the passenger remembers that the luggage contains important business material. The norm invocation (punishment) for an agent who first decided not to claim the luggage is a longer distance it has to pass resulting in a higher probability to avoid this behaviour in the future.
2. **Customs:** At the next stage agents have to pass customs to declare their goods. In the actual case every agent has something to declare. It is not of interest here to observe an agent type having nothing to declare because these agents would not have to think how to act at this stage, thus would never do something wrong and thus would never receive a norm invocation. Due to the aim to discover the effects between agents with the same presuppositions, this case is not being observed. Agents without goods to declare are considered just being invisible to prevent the scenario from becoming too overloaded. The customs office was modelled as a stage containing two passageways, one for people with goods to declare (the "red channel"), and another one for people who do not have anything to declare (the "green channel"). Agents who declare are being sent to the next station (taxi stand) without any further procedure. Agents not declaring can – by chance – either be revealed, or they succeed with smuggling. The punishment for revealed smugglers is the longer distance for having to go back to the "declare booth" and pay the declaration fee.
3. **Taxi stand**: At station three the agents wait for cabs at a taxi stand to ride home. Agents can either wait in line until it is their turn to enter a taxi or push to the front to save time. Special about this stage is that agents doing wrong by jumping the queue do not loose time by having to go back but they get a norm invocation as punishment like on the other stages. This norm invocation can be compared to the real world scenario when people start yelling but not doing any further tasks. As the punishment might not be as strong as at the other stages the agent might be more likely to act wrong next time as well.

So where do the agents come from and how do they move from stage to stage? Agents enter the airport through a TRASS object called "source". This source represents aircrafts in the real world bringing passengers to the airport. After the agents landed at the airport they move through the environment freely until they encounter a guidepost showing how to get to the stage that has to be passed next. A guidepost in our scenario is an agent, which always stays in one place showing other agents where to go. These guideposts are equivalent to signs at airports containing directions. However since the agents normally move through the environment randomly, guideposts are used to make the agents move to a certain direction. Guideposts deliver direction information to agents when they enter a certain perception area. The information consists of coordinates to the next guidepost that has to be addressed. This way the agents move from guidepost to guidepost through the airport scenario. In the draft scenario it was also foreseen to implement a second airport in which the norms for acting on the stations would have been the exact opposite of the first airport. This consideration was based on the fact that in the real world this phenomenon can be found as well, but for the current version it was more important to focus on the effects between the two different agent types by adjusting the ratio between them and letting them pass the same airport several times to see how the norms change over time. In the scenario agents spin rounds through the airport. Each new round is being considered as a new visit at the airport some days, weeks or months later. This way the agents remember how they reacted and what they perceived last time they visited the airport and, thus, act accordingly.

As described in the previous chapter two different types of agents were used: norm recognizers and social conformers. To give the whole scenario a more realistic touch those “synthetic” agent types were set into a real world context by assigning two different cultural groups to them. These two cultures should have a language with different character sets (e.g. as for Germany and China) and the people should not be able to understand the other groups language. The airport the scenario is taking place shall only have direction signs in one language and char set (e.g. Chinese). This way it makes sense that the social conformers only follow the mass rather than making own decisions (because they cannot understand the signs). In our example the social conformers can be regarded as Germans visiting a “rural” Chinese airport which has no English (or other Latin-letter-based language) signposting.

17.3.1 Physical Layer

For the implementation on the physical layer the decision had to be made between TRASS and Repast which are both explained in the previous chapters. To develop this scenario, TRASS was chosen due to the extended features of the graphical interface provided by this tool. Furthermore, the traffic scenario had already been developed in TRASS where agents could freely move around and their decision making and learning behaviour was modelled using EMIL-S.

To fulfil the purposes of this scenario some code writing had to be done. First of all an interface was needed to store the strategies and plans which the agents should carry out in the scenario. One strategy is applicable for one agent type whereas each strategy can have several plans for different situations (example: evade another agent). Additionally this strategy class is capable to read out variables off the XML-file where the basic information about the TRASS simulation is stored.

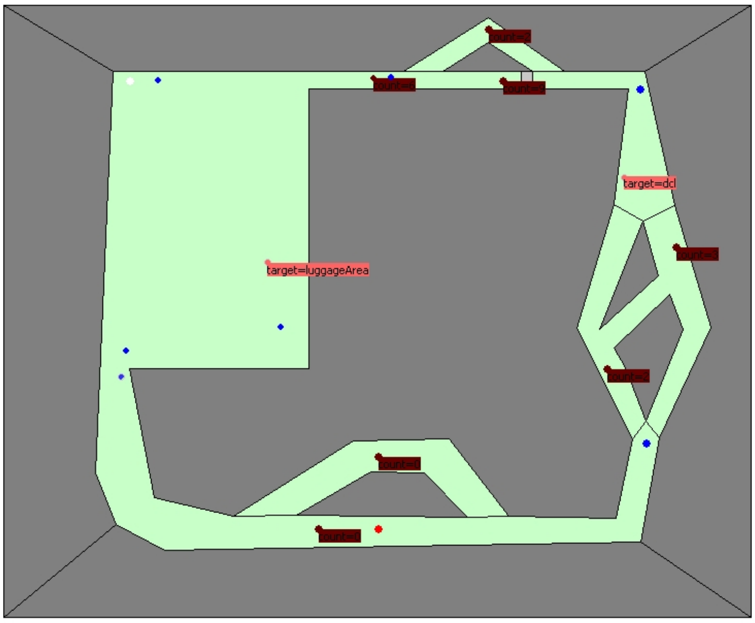


Figure 106. TRASS implementation of the scenario
White spot: agent source; Blue spots: guideposts; Brown spots: targets with counter information; Red spot: traffic light; Red spots with info box: agents heading for a target

Figure 106 shows the current state of the scenario implementation. As the scenarios have already been described in detail in the previous section, this section will only provide an insight how they are implemented.

Both agent types are generated by the source in the left upper corner. The agents are led by signposts which send them to the next scenario or provide different directions among which the agent has to choose. There are two types of signposts. They differ in the number of directions they provide. If there is only one direction provided, the TRASS agent just follows this direction, otherwise the EMIL part of the agent makes a decision which direction to take.

The EMIL-S interface is used to send and to get messages from the EMIL layer. The TRASS agent object first sends a message to EMIL-S containing the information about the current situation (time and occurred event). The `sendMessage` method is intended to receive messages sent by EMIL-S with its decision for certain situations. The simulation has been designed in a way that the name of actions in the EA tree (e.g. `takeLuggage`) fits with the name of targets in the simulation. Therefore the return values sent by EMIL-S can be taken as target values for the agents.

The first and the second contexts are working in the described way. There are two possibilities from which the agent (its EMIL-S part) has to choose one. The third context in the lower part of the simulation area is different. The agents should queue in a line and wait until the traffic light goes green. The intention of this scenario is that the agents queue in a line and wait until it is their turn. However, the agents are able to pass the queue and ignore the traffic light. This behaviour is controlled by EMIL-S which decides either to line up or to pass the queue.

All contexts provide the possibility to have agents learn the effects of certain decisions. In the first context the agent may choose to take the lower direction because it wants to pass quicker through the airport. As this is the wrong way (it has to take the long way), it has to go back all the way (first punishment) and it gets a Norm Invocation in EMIL-S (second punishment) which will affect its decision making process if it gets into the same situation once again. Therefore, a learning curve should be visible after a few simulation runs. In the second context, the customs, there is a security officer which controls some of the agents passing the "nothing to declare" area by chance and fines them if they had something to declare. This action also generates an EMIL-S Norm Invocation message. Finally, in the third scenario the agents have to wait at a traffic light which symbolizes to queue in a line at a taxi stand in front of the airport. Similar to the other scenarios, the agents have the possibility to chose the wrong action and pass the queue. Doing so, the agents get to the end much faster but they receive a norm invocation which affects the probability to choose "ignoring the queue" in a negative way.

All facts described about the simulation so far are only applicable to the Norm Recognizers, because only they are capable of thinking about the decisions to make which is done in EMIL-S. The second group of agents is not able to interact with EMIL-S, therefore they have a slightly different decision making process. The social conformers tend to copy the behaviour of Norm Recognizers as they copy the actions which were undertaken by other agents in a comparable situation in the past. In the simulation the Social Conformers have access to a board of actions for a certain decision which has been made in the recent past. They base their decision on these actions, plus a small coincidence factor. Therefore they profit from the learning effect of the Norm Recognizers. In a situation where only Social Conformers are involved, or the decisions made by NRs are too old the SCs decide by chance which option to favour.

17.3.2 EMIL-S Layer

On implementing the previous mentioned scenarios in EMIL-S the following steps had to be taken in the Agent Designer.

First events had to be defined which are needed for the Event-Action Trees (EAT's). There are three events for the airport scenario:

Event Name	Event Description
Customs	Pass customs
Queue	Queue at the taxi stand
Luggage	Get the own luggage

In the second step the actions for the EATs were defined.

Action Name	Action Description	Action type
takeLuggage	Take the Luggage	Environmental
nTakeLuggage	Do not take the Luggage	Environmental
dcl	Declare goods	Environmental
ndcl	Do not declare goods	Environmental
queue	Queue in the line	Environmental
nqueue	Pass the queue	Environmental

There are action properties which are called "Environmental" and "norm invocation": Environmental responses are sent to the agent, i.e. declare the goods, norm invocations responses remain in EMIL-S, i.e. a change in the probability to act in a certain situation.

In the customs context there are two possible actions: EMIL-S can decide to let the agent declare its goods and pay customs or not to do so. If the agent doesn't, there is a chance of getting caught by a security officer, who sends the agent back to the declaration room and fines the agent, which is done by an EMIL-S Norm Invocation.

The action properties also have got a field named expression: this field stores the content of the message sent to the TRASS agent, for example: Environmental (Expression: ndcl) — or Norm Invocation (Expression: Strong).

In the following step two agent types were created for the airport scenario: the norm recognizers (NC) and social conformers (SC).

The final step for creating the Event-Action-Trees is to add rules for the agent types (Figure 107).

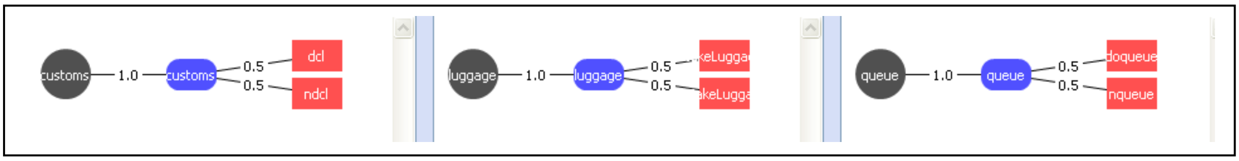


Figure 107. Event-action trees of the Norm Recognizer

As the Social Conformers don't have the ability to use EMIL-S to make decisions, they are implemented as an agent type but don't have any Event-Action Trees.

To observe learning effects in the scenario, the probabilities in the first and the last scenario were adjusted to 70/30 and 50/50 percent, respectively.

17.4 Simulation Runs and Results

After a successful simulation run data have to be gathered and evaluated systematically.

To achieve a suitable result a high number of simulation runs is needed in which parameters must be modified. The "MEME" (Model Exploration Module) tool was used to support these complex procedures.

MEME is part of the Multi-Agent Simulation Suite (MASS) which consists of three applications for different solutions in modelling. MASS and MEME were developed by AITIA International Inc. (see Chapter 12). However MEME can be detached from MASS. Therefore it is a good tool for supporting TRASS and EMIL-S simulations. It helps to plan and accomplish the simulation runs. It operates autonomously while attending to the variation of parameters or conditioning and storage of data. The stored data can be evaluated by special software.

Different approaches and assumptions have to be proven with MEME and the most important parameters have to be varied in order to validate the model.

MEME accomplishes all these procedures autonomously. After running through the model and gathering data the received information is evaluated. It is desirable to gather information about the relation between Norm Recognizers and Social Conformers. The larger the number of Norm Recognizers is in comparison to Social Conformers the faster learning effects will occur in the population, because Social Conformers only observe and adopt the behaviour of surrounding agents.

17.4.1 Simulation Runs

Prior to start the experimentation with a simulation model, it is necessary to choose the parameters that will be modified, the actions that will be counted and other settings for gaining feasible results.

There seem to be mainly two parameters interesting for modifications. First of all, the number of agents involved in the model was set to 15. The most important modification is the variation of the Social Conformer proportion. With this parameter it is possible to regulate the number of Social Conformers in relation to the number of Norm Recognizers interacting in the model. Hence three different values (0.0, 0.33, 0.66) were chosen to see how the different proportions of agents affect the social behaviours and the norm building process. The first value "0.0" means that there are only Norm Recognizers and no Social Conformers in this simulation run. The second and third values mean that the ratio of Social Conformers is about one-third and two-third, respectively.

To see how the agents behave while running through the model, the appropriate or inappropriate behaviours of agents in the different scenarios were counted. In the baggage claim area the agents which "take" or "not take luggage" were counted, also the number of agents declaring their customs and not declaring their customs. In the last scenario counters are installed to retrieve the number of agents which "queue" or "not queue" while waiting for a taxi.

17.4.2 Simulation Results

After adjusting the parameters, the simulation model was performed on the basis of 50000 steps per run. The simulation progress was recorded every 50 or 100 iterations and visualized by MEME. The results of the runs that merely contained Social Conformers were not suitable so they are left out here. Reason for this is the fact that the results of these runs depend mainly on the random decision made by the first agent.

Analyzing the results, it can be concluded that the existence of norm recognizing agents already at a minimum extend leads to a collective disposition to make norm conform decisions. A learning effect can be observed and becomes evident especially in the luggage scenario, where in the beginning the agents' decisions are quite similar (according to the initial rule configuration) and diversify only after a certain number of passes. An overall examination reflects that the trend to norm conform actions (takeLuggage, dcl, doqueue) in the long run tends to grow, while the accretion of the counter-norm actions (nTakeLuggage, ndcl, nqueue) declines.

The simulation gives clues for a strong correlation between the proportion of Norm Recognizers and the amount of agents making norm-conform decisions. Nevertheless, the model is in an early stage but has potential for future work concerning this suspicion.

An important result is the impact of a varying Social Conformers proportion. As it can be seen in Figure 108 diverse rates of Social Conformers influences the time how fast agents learn from their inappropriate behaviours. In the first case of "dcl0" no Social Conformer participated in the simulation, i.e. only agents able to learn acted in this scenario. This explains the rising curve of agents who behaved adequate. In the second case (dcl0.3) the Social Conformer proportion was 33.3%, i.e. one third of all acting agents only detected and adapted the social behaviour which is performed by the majority of surrounding agents. Therefore the number of agents who acted appropriate decreased. The last simulation run was initialized with a Social Conformer proportion of 66.6% and shows the decreasing number of adequate behaviour as well.

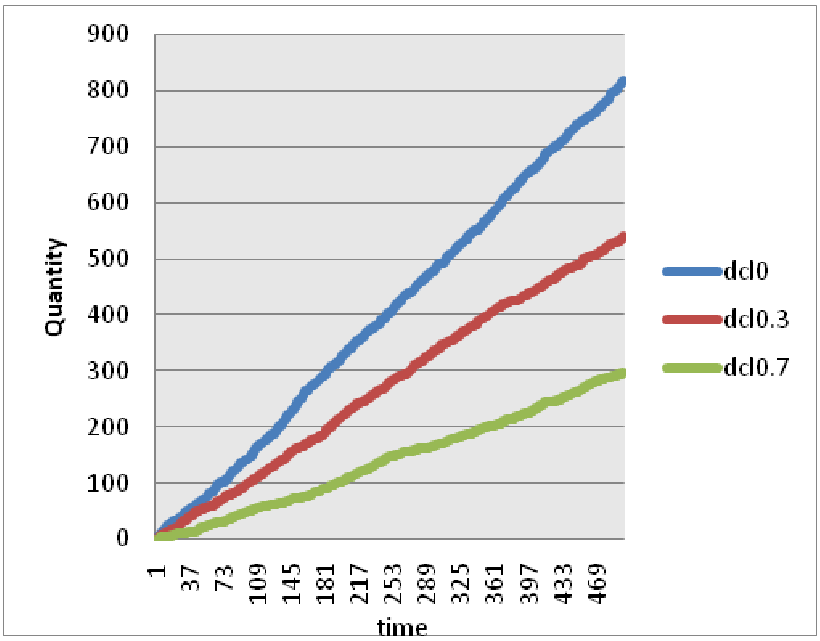


Figure 108. Impact of varying Social Conformer proportions

Concluding it can be said that the multi-scenario model runs successfully. It could be shown that in principle the combination of EMIL-S and TRASS is suitable for the replication of the original multi-scenario world model.

Chapter 18 Demonstrating the Theory: Conclusion of Scenario Results

Klaus G. Troitzsch

Abstract
This chapter summarises the experience drawn from the scenarios dealt with in this section. It will describe the extensions of both the theoretical model (the logical architecture of the normative agent) and the EMIL-S agent design that were necessary to cope with the requirements of the different scenarios.

18.1 Introduction: Differences and Commonalities of the Scenarios

While chapter one of this section mainly discussed the differences and commonalities of the target systems underlying the simulated scenarios, the purpose of this chapter is to discuss differences and commonalities between the simulated scenarios and of the problems that had to be overcome while designing the simulation models.

According to the different environments, the physical part of the scenarios had to be modelled with different tools: TRASS lends itself to modelling topographies, whereas Repast offers less straightforward options for this purpose (and moreover EMIL-S and TRASS had been developed mainly by the same group). Connecting EMIL-S to TRASS and Repast turned out slightly less easy than foreseen but due to the platform independence and modularity of the Java language, writing the necessary interfaces was not a big problem. The same, by the way, holds for the interfaces between the simulation software proper (EMIL-S plus TRASS and Repast, respectively) and the experimentation and analysis tool MEME.

According to the different agent descriptions the agent design had to be done separately for every scenario, but with the growing familiarity of the persons involved with the EMIL-S agent designer, these differences turned out negligible. As already discussed in the first chapter of this part, decisions had to be made how to model the collective agent (the loaners' group), but it turned out that a non-cognitive ("bookkeeping") agent would be the most appropriate solution.

Thus from the technical point of view of the modelling team, the commonalities among the four scenarios are much larger than the differences — another indicator for the versatility of the EMIL-A and EMIL-S concepts.

18.2 Extensions to the Original Scenarios: Were They Necessary Due to the Limitations of EMIL-S or Did They Extend the EMIL-S Concept?

During the process of designing and implementing EMIL-S there was some frequent discussion whether EMIL-S would necessitate material changes to the scenarios defined in the earlier (non-EMIL-S) simulation programmes. In all four scenarios which led to running EMIL-S simulations this was not the case, instead the EMIL-S applications can be seen as successful replications of their predecessor versions.

On the other hand, the EMIL-S version which was used to write the first version of the first scenario turned out to be insufficient for later versions of the first and later scenarios, but only to the extent that minor additions and changes had to be made (which were, although they were minor in comparison to the tool as a whole, nevertheless quite cumbersome for the developers.

18.3 Extensions to the Agent and Environment Architecture

There is one deviation from the original concept which was already mentioned in the first chapters of this report (see page 90ff.). It turned out reasonable to use the same mechanisms for normative and for non-normative deliberation and decision making processes, thus not restricting the applicability of large parts of the EMIL-A concept to normative reasoning, but using most of it for all kinds of reasoning.

18.4 Conclusion: Lessons Learnt

What can be learnt from the development of EMIL-S is that a theoretical concept which is sufficiently formalised — as turned out to be the case with EMIL-A — alleviates the development of a software tool supporting simulations following the theoretical concept and the development of simulations with the help of this tool.

In this respect EMIL-S is a nearly one-to-one implementation of EMIL-A and capable of supporting the development of simulation models of EMIL-A compliant scenarios.

It is still an open question whether the Hume model (see Chapter 4) is sufficiently EMIL-A compliant to convert it into an EMIL-S application. This effort was deferred.

Chapter 19 Summary of Major Advances in EMIL-A and EMIL-S

Iris Lorscheid, Ulf Lotzmann, Michael Möhring and Klaus G. Troitzsch

Abstract

This section summarises the main innovations of the EMIL-A theory and of the EMIL-S simulation toolbox. The main innovation of both is that mutual influence between agents is modelled in terms of symbolic communication, not in terms of a mind-reading metaphor. Agents exchange messages among themselves which only after receipt and interpretation by the receiving agent take any effect on the latter. Interpretation of an incoming message is done in the light of earlier experience stored in the agent's memory, such that different agents — and a particular agent at different times of its existence — will interpret a message with a different outcome. For the agents' communication an environment is necessary, not only to enable communication as such but also as an object of communication (different features of agent's environment are the objects of messages which can only be interpreted when the receiving agent's shares the view of the environment referenced by the sending agent). Another, more technical innovation is the feature of EMIL-S which allows users to graphically describe the behavioural rules which the agents initially have (and which change over time due to learning processes within the agents). In principle, different kinds of learning algorithms can be built into models of norm learning, norm adoption, norm emergence and norm innovation.

19.1 EMIL-A and EMIL-S as Concepts that Use the Message Interpretation Metaphor for Modelling Communication among Agents

In the EMIL project the emergence and immergence (Castelfranchi, 1998, S. 39) of norms in societies of cognitive agents is being analysed, modelled and simulated (Andrighetto et al., 2007a, 2008b; Troitzsch, 2008). It takes into account the peculiarities of human social systems with respect to the learning of norms, and for this research it makes a difference whether agent behaviour is a result of the anticipation of some profit or payoff or of the expectation of others' behaviour as in game-theoretic models of norm emergence or whether agent actions are the result of norm internalisation, i.e. the internalisation of a norm made explicit (by itself or some other agent) with respect to certain classes of event and related actions. While the game-theoretic models do not endow their agents with models of other agents, immergence related models will contain agents that are capable of forming models of other agents' deliberations. Classically a software agent would "know" that it is more or less likely for a particular other agent to behave in a certain way — which makes this behaviour more or less predictable —, and it could learn to improve the related expectations (as noted above) mainly statistically. With reasoning and communicating about explicit norms, it is additionally possible to add another quality of learning: learning not only from own experience and the observed experience of others, but additionally from the communicated experience of others — which is more than just communicating what just happens but entails the information of the "teaching" agent about its belief why it happened as it happened.

Learning mode	example of what was experience	result of learning
learning from own experience	"X happened to ego in situation S (entailing other agents) when ego performed action A to reach goal G"	statistical models of environment
learning by observing others' experiences	"Y happened to alter in situation T when alter performed action B to reach some unknown goal"	statistical models of others' expectable behaviour
learning by listening to others' reports of their experiences	"Z happened to alter in situation U when alter performed action C to reach goal H, and alter said it could not have avoided Z because it found goal H must be reached"	rule and norm based models of others' expectable behaviour

Table 7. Levels of social learning

The idea behind this distinction between statistical learning and norm learning is that it will be possible for agents of this architecture (Andrighetto et al., 2007a) to apply norms to a much wider variety of events and actions than in classical game-theoretic and related models with their "extremely simple, indeed psychologically impoverished" agents (Epstein, 2006). Agents of this architecture are able to keep a long memory of situation-action-outcome relations, can use this memory to learn which kinds of actions in which situations produced which kinds of outcomes. This makes it necessary for them to be able to abstract from the individual situation, action and outcome and to categorise at least situations and actions, perhaps even outcomes with respect to different goals that they have (most classical machine learning strategies try to optimise only one one-dimensional goal). Outcomes in the sense of the word used here are not only the payoffs of game theory, but also valuations by others, which are incommensurable with whatever goes under the name of payoff. Consider for example a pedestrian crossing a lane in spite of a red traffic light — if this person is cautious enough this action will have a positive payoff (as the waiting time is substantially decreased and some other goal lying beyond the lane can much earlier be achieved). But this action will be observed by another person who might take offence at this kind of behaviour as it provides a bad example for children present in the same situation, who might learn from this misdemeanour that crossing lanes while the traffic lights show red for pedestrians is correct adult behaviour which can be imitated without danger. From the point of view of the first pedestrian, the two outcomes of his or her action are incommensurable: On one hand there is a positive payoff in saving time, on the other hand there might be some reproach from the side of the law-abiding observer, but no damage is done to anybody as the child whom the observer had in mind was not even present in the scenery. But the first pedestrian might still keep the reproach in mind and control his or her actions with respect to the past reproach next time. In this fictitious situation, avoiding time delays and avoiding reproaches are orthogonal. Considerations like this one introduce moral social actors who deliberately control their actions both with respect to payoff expectations and with respect to norms which emerge in a society of human actors when these exchange deontics (commands, forbiddances, permissions). It has to be observed that although violations of these deontics often have negative payoffs as well (fines and other kinds of punishment), they even work when in the current situation negative payoff cannot be expected at all (pedestrian crossing, red traffic light, no children, no parents, no police far and wide). If simulations of these kinds of human behaviour existed, then the software agents representing humans in the simulated scenarios would be "normative software agents" for whom the adaption to user expectations might be much easier than for classical machine learning agents, as they could be "trained" not by pure reinforcement learning. Instead they would understand and interpret commands, forbiddances and permissions in the same way humans understand and interpret these deontics.

Taking all this into account, the EMIL project designed a new platform for describing and simulating processes of norm emergence and innovation in which agents respond to influences from their environment only indirectly. They receive messages, and these messages take only effect after a process which resembles the interpretation of messages by human beings. Unlike physical particles and most living things, humans do not directly react on stimuli, but their response is the result of at least some deliberation which takes a long memory into account (this is why human behaviour is extremely path-dependent). To put it less formal, interaction between people are by persuasion, and persuasion needs one who is persuasive and another person who is persuadable. Norm invocations thus work only in the long run.

To come back to the (too) simple concept of levels, the micro or individual level and the macro or society level, we find what is usually called "downward causation" as an effect of the macro level on the individual on the micro level, but also "upward causation" as the effect that arises from the behaviour of individuals that was changed due to the "downward causation". An example from one of our current simulation scenarios may make this clearer:

A person A is crossing the street when a bus is approaching. In order to avoid a collision, the bus driver B brakes, opens the window and admonishes the pedestrian with the words "You must not cross the street when I am approaching with my bus, as braking the bus endangers my passengers!" A accepts the reproach and will be more carefully cross streets in the future. If this happens often enough to other persons, and if often enough other persons observe the discussions between careless pedestrians and careful bus drivers, "sociological phenomena penetrate into us by force or at the very least by bearing down more or less heavily upon us" (Durkheim, 1895/1982). To use another quotation of one of the founders of sociology in the 19th century: "it is society which, fashioning us in its image, fills us with religious, political and moral beliefs that control our actions" (Durkheim, 1897/1951). And as a consequence, these norm invocations – and the resulting behaviour – occur more and more often and become a "sociological phenomenon": not only A, but others, too, abstain from crossing streets, not only in the presence of B's bus, but also in most other cases of approaching buses and other vehicles.

19.2 Agent Communication as an Object of Agent Communication

19.2.1 Learning Algorithms Usable in EMIL-S

As shown in Figure 109 we define two main learning processes, the process of reinforcement learning by updating the probability values in the event-action trees on the one hand and the process of normative learning as changing values after having received norm invocation messages on the other. The objective of this chapter is to make these processes explicit and to analyze how they influence the agents.

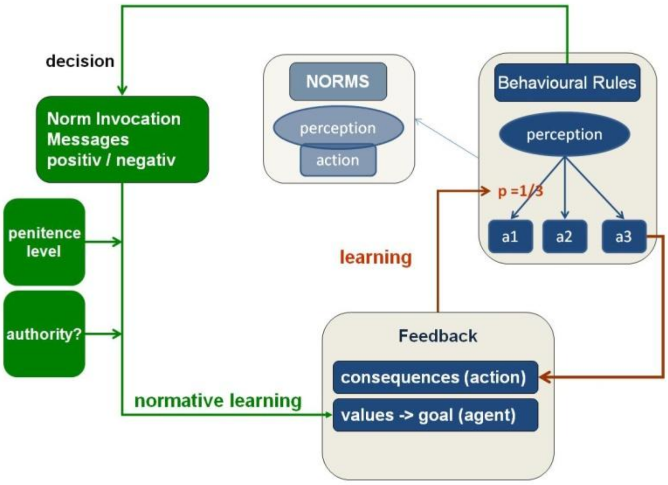


Figure 109. Processes and Dependencies

19.2.2 Reinforcement Learning

The selection probabilities on the branches in the event-action trees change over time by a reinforcement learning algorithm. In each time step all agents make a decision based on the tree that matches the current perception ("no neighbour", "neighbor" or "neighbor is coloring"). Next, the decision leads to an action. Every action has an utility given by a feedback from the environment. On this basis the agent updates the action's selection probability value. In this first approach a simplified model was chosen. The reinforcement mechanism is defined by the following formula:

$$p(a,t) = (1- \lambda) * p(a,t-1) + \lambda * u_j(a,t)$$

with *p(a,t)* as the probability value for action *a* at time *t* and λ as a discount factor that defines how strong past experience and utility should influence the learning process.

The utility $u_j(a,t)$ is the received feedback for action *a*. It is calculated for a red agent (blue agent) by setting the sum of red values (blue values) on the neighboring patches in sight of the agent in relation to the red agents' (blue agents') optimum, which means that all patches around would have its (the agent's) own red value (blue value). The view range of an agent is defined by an input parameter.

In this section we explained how an agent updates the selection probabilities in its event-action trees. One could see that the received feedback depends on the agent's internal value, which stand for the agent's goal of how red (blue) the world should be. These values are initialized randomly to represent a population of individuals with different level of goals, but they (can) change over time by influences from the society. The individuals are able to communicate with each other and therewith influence each other by norm invocation messages. These messages are the core concept of the normative learning process in this chapter and will be explained in the next sections.

19.2.3 Sending Norm Invocation Messages

After an agent observes a neighbour coloring its patch, it compares the neighbor patch color with its own value and decides whether to send a norm invocation message or not. An example: A red agent compares the red value of the neighbor field with its own value.[63] If the red value of the neighbor patch is smaller, the agent decides to send a norm invocation message to the coloring neighbor that says "do more red!".

Norm invocation messages contain valuations about actions and lead, under defined circumstances, to a change of the receivers' internal values. The basic structure of a norm invocation message is

norm_invocation(sending_agent, receiving_agent, modal, action, weight).

The message contains information about the sending and receiving agent, transmits an action and assigns one of the defined modal terms such as approval or sanction. In our example the message would be:

norm_invocation(redagent x, blueagent y, "do less", "coloring red", weight).

The weight of a norm invocation depends on the distance of the compared values. An input parameter defines the percentage of the total distance that defines the invocation weight.

Whether this message influences really the receiver and makes it color less red depends on preconditions we describe in the next section.

19.2.4 Receiving Norm Invocation Messages

After an agent sent a norm invocation message the receiver receives the message immediately. Whether it accepts the norm invocation or not depends on the agent's penitence counter and the authority status of the sender. These ideas are specified in the following sections.

Penitence Level

In this learning concept every agent has an internal penitence level, representing its ability to accept negative norm invocations from the society before changing its internal values.

The penitence level is implemented by an agent attribute *p_level* ∈ N. We distinguish different agent types concerning this characteristic. Depending on the scenario the agent population consists of one homogeneous or up to three different agent types.

[63] Equivalent to the red agents blue agents compare the blue value of the neighbor field with its own blue value.

$$p\_level = \begin{cases} p\_impervious, & \text{if } AGENT\_TYPE = \text{impervious agent} \\ p\_guilty, & \text{if } AGENT\_TYPE = \text{guilty agent} \\ p\_embarrassed, & \text{if } AGENT\_TYPE = \text{embarrassed agent} \end{cases}$$

with the experiment values *p_impervious= 15*, *p_guilty=10* and *p_embarrassed=5.* After this definition an impervious agent is able to accept up to fifteen negative norm invocation messages before it is willing to change its rules, whereas an embarrassed agent has a low threshold and wants to change after just five negative norm invocations. In our simulation we started with one penitence level for all agents. It is implemented as input parameter.

All agents take note of every received negative norm invocation message and keep these events in mind by incrementing an internal counter. In the case that the counter exceeds the penitence level, an agent's tolerance level is exceeded, and the agent is willing to accept the invocation message.

Additional an exceeded penitence level is the precondition for the genetic algorithm concept. The genetic algorithm, specified in section 3.7, creates and adds new rules, not foreseen by the modeler. Thus the exceeded penitence level is the precondition for norm innovation in this model.

In this first approach the penitence level is fixed for the whole simulation run, the implementation of changeable levels, influenced by experience and the penitence level of others (following changeable aspiration levels in satisficing learning, see Brenner, 2006), is conceivable for later implementations.

19.2.5 Learning from Authorities

The influence of a norm invocation depends on the distance between sending and receiving agent. The authority concept is based on the bounded confidence model (see Hegselmann and Krause, 2002). The smaller the difference of values between sender and receiver, the more likely is that the receiver accepts the invocation message and changes its values. Therefore the authority value of a sender is defined by the receiver by the following equations:

$$\begin{aligned} value\text{-}distance = \quad & (\ |\ redvalue\text{-}invocator - redvalue\text{-}target\ |\ ) \\ & + (\ |\ bluevalue\text{-}invocator - bluevalue\text{-}target|\ ) \end{aligned}$$

$$authority\text{-}level = \quad (\ 1 - (\ value\text{-}distance\ /\ max\text{-}distance\ )$$

To decide whether to accept a norm invocation or not, the target agent chooses a random number between 0 and 1. If the resulting value is below the authority level, the agent accepts the norm invocation.

19.2.6 Brave and Anxious Agents

If the action "look-around" is chosen, the agents look around and compare the color values of the patches within their view range with its own value. In our model the red (blue) agents go to the patch with the smallest red (blue) value. To analyze the possible influences of the learning concepts we additionally defined the concept of anxious agents, which go in the other direction than to the field with the smallest value to avoid agents with other values.

19.2.7 Genetic Algorithm (Norm Innovation)

The genetic algorithm (GA) extends the behavior rules with new rules by adding mutated copies of existing event-action trees. This process is triggered when the agent's penitence level is overshot by negative norm invocations. The GA process creates new event-action relations not foreseen by the modeler — whether they prove useful needs to be explored by the agent.

The GA proceeds in the following steps:

Choice of parent trees. The value of the input parameter *ga_rate* defines the number of event-action trees, a random choice among the trees matching the current event classification determines the set of parent trees. If the *ga_rate* value is greater than the number of matching ("firing") trees, those with the most similar classification are chosen to fill the set.

Creation of offspring trees by copying the chosen parent trees of step 1. The parent trees remain unmodified in the normative frame whereas the offspring trees go through the mutation process.

Random choice of the mutation process. We define two possible mutation processes: Crossover (mutation of event-action relations) and mutation of actions. Both have the same selection probability of 0.5.

19.2.8 Mutation Process: Crossover or Mutation of Actions

Mutated offspring trees are added to the normative frame with the same probability values and classifications as the parent trees. The new event-action trees are available as new rules in the knowledge base.

19.2.9 Crossover — Mutation of Event-Action Group Relations

This mutation process is a crossover process between two event-action trees, where randomly chosen action groups are exchanged. Therefore random pairs must be defined within the offspring set.[64] Every tree contains at least one action group with mutually exclusive actions. A randomly chosen action group per tree — including the probabilities on the branches — is exchanged between the members of a pair.

Offspring trees coming out of this process perhaps define inexpedient reactions to events but never impossible consequences, due to the fact that the GA works within one normative frame of one agent, so he doesn't gain new skills but new possible answers to events.

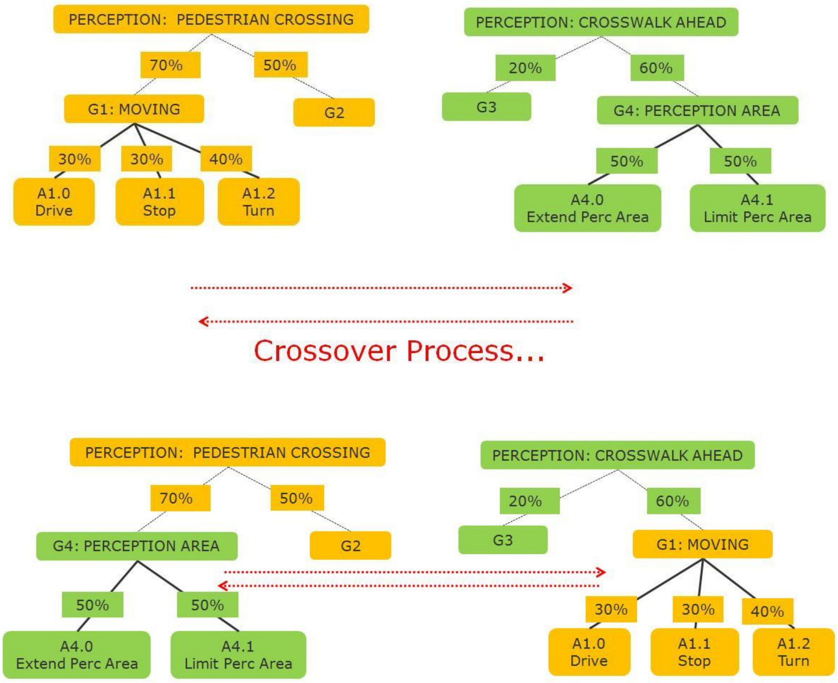


Figure 110. Genetic Algorithm, Crossover Process

Figure 110 illustrates this concept by an example based on the simulation scenario "TRASS"[65] (Lotzmann and Möhring, 2008; Lotzmann, 2008). It is an implemented scenario within the EMIL project and simulates the emergence of norms in a traffic environment. To reach their goal pedestrian agents need to cross a street where car driver agents drive their cars. The event-action trees in Figure 110 are part of the

[64] In the case of an uneven number of offspring trees, one offspring will be erased.

[65] See Chapter 14 for details.

normative frame of a car driver agent. They contain rules concerning its abilities to vary its perception area and to control the movement the car.

The picture shows two event-action trees going through the mutation process crossover. On the top (area 3.1) we see the unchanged offspring trees. The tree on the left defines the rules firing if an agent sees a pedestrian crossing; the tree on the right contains possible actions after the car driver agent sees a crosswalk ahead.

Within the crossover mutation one action group per tree will be randomly chosen with an equal selection probability of 1/*n* and *n* = number of action groups. In this example, group "G1: MOVING" in the left tree and group "G4: PERCEPTION AREA" in the right tree are chosen. Group G1 contains all actions about moving the car, action group G4 contains all actions to change the perception area.

Now the chosen action groups will be exchanged. The trees in the lower part of Figure 110 are the mutated offspring trees after the crossover process. These new trees will be added to the normative frame.

The result of the crossover process is in this example that this car driver agent can now react to a pedestrian crossing the street with changing its perception area, or, after seeing a crosswalk ahead with drive, stop or turn.

The original abilities are not deleted as the copied parent trees remain in the normative frame.

Mutation of Actions

In this mutation process one randomly chosen action per offspring tree changes. This is possible due to the fact that actions are parameterized.

Figure 111 shows this mutation process with the example of one of the event-action trees explained in section 3.7.1. In the first step of this mutation process one action out of the set of all actions in the tree is randomly chosen, with an equal selection probability of 1/*n* and *n* = number of actions.

In this example the action A1.0 (DRIVE) of action group G1 (MOVING) is chosen. The action *drive* has the attribute *speed*. If a mutation fires on the action *drive* the speed will increase or decrease randomly, again with an equal selection probability of 0.5 and a defined *change_value* (1/3 in this example).

This process implements a slight change of norms by changing the values of actions. Thus it implements a slower norm innovation process than the crossover process.

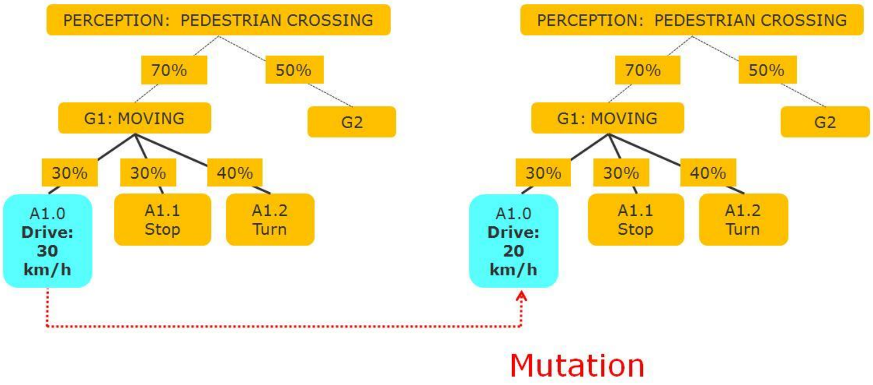


Figure 111. Genetic Algorithm, Mutation of Actions

19.2.10 Protected Rules (Higher Law)

Event-action trees that have turned out successful in terms of an agent, as they received a lot of positive norm invocation messages concerning it rules, will be protected. This approach implements the idea of higher laws like basic rights of individuals within a society.

For this purpose a counter within every tree will track the number of positive feedbacks for all its actions. When this counter exceeds the value of the specified input parameter *protect_law* the tree is safe and will never be forgotten.

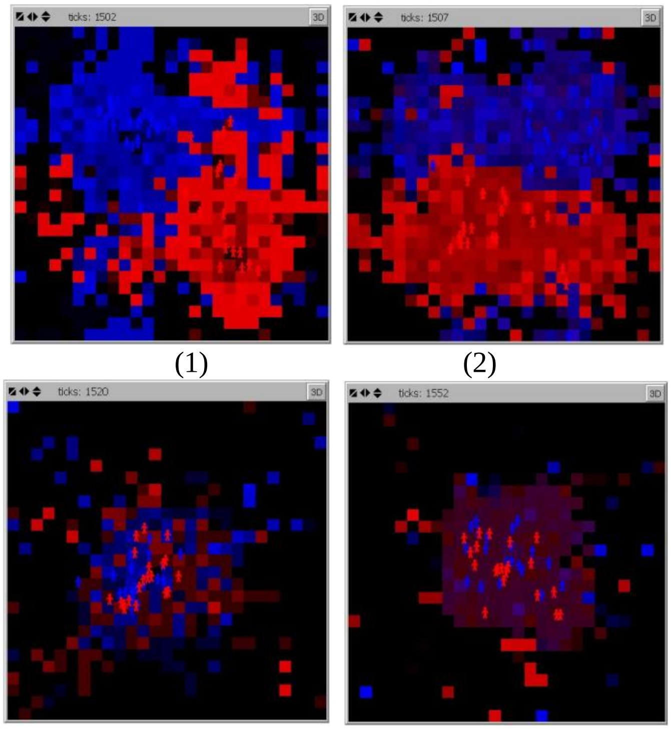
19.2.11 First Results

In our first experiments we look for possible impacts of the two main learning processes on the system behavior.

Figure 112 shows the visual results of experiments (1) – (4), which will be explained in the following.

The windows on the left hand (exp. (1) and (3)) show results of simulation runs without normative learning, but with learning of actions by changing the probabilities in the event action trees. The simulation runs of the results shown on the right hand however (exp. (2) and (4)) were conducted without rule learning but with a normative learning process.

In experiments (1) and (2) the population of agents was initialized only with anxious agents, so these agents move away from the fields with the most different (lower) values in the “look around” procedure. As one can see the learning processes don’t make a big difference in this scenario. Due to the strong specification of moving away from patches and therewith from agents with different values, a clustering of colors can be observed in both simulation runs. Red agents and blue agents built their own area and after a while they don’t deal with each other. In experiment (2) the color values slightly changed, because norm invocation messages were exchanged before the clustering was build, but the overall system behavior is in both experiments comparable.



(1) (2)

(3) (4)

Figure 112. Results of experiments (1)—(4)

The population of experiments (3) and (4) was initialized with brave agents who move in the direction of the field with the most different value. Due to this specification the agents move in the direction of other agent types so they need to deal with each other. In simulation runs without norm invocations but with rule learning (exp. (3)), the agents begin also to build clusters and learn to avoid each other, but clusters are smaller now and not as stable as in experiment (1). In experiment (4) the agents are not able to learn to avoid each other in own color clusters, they send each other norm invocations and adapt their values to a compromise color of pink as new emerged norm.

These first experiments show how crucial a learning process can be for the system behavior or not. Experiments (1) and (3) show how the specification of actions can overrule learning process effects, whereas in experiments (2) and (4) different system behavior was observable due to different learning concepts.

Chapter 20 Looking Forward: On Norm Internalization

Rosaria Conte, Giulia Andrighetto, Marco Campennì

Abstract
Internalization is at study in social-behavioural sciences and moral philosophy since long; of late, the debate was revamped within the rationality approach to the study of cooperation and compliance since internalization is a less costly and more reliable enforcement system than social control. But how does it work? So far, poor attention was paid to the mental underpinnings of internalization. This chapter advocates a rich cognitive model of different types, degrees and factors of internalization. In future work, it will be implemented on EMIL-A, and simulated on EMIL-S, in order to check the individual and social effect of internalization.

20.1 The Issue

The problem social scientists still revolve around is how autonomous systems, like living beings, perform positive behaviors toward one another and comply with existing norms, especially since self-regarding agents are much better-off than other-regarding agents at within-group competition. Since Durkheim, the key to solving the puzzle is found in the theory of *internalization of norms* (Mead, 1963; Parsons, 1967; Grusec and Kuczynski, 1997; see Gintis, 2003).

Norm internalization is one of the common themes running across all of the social-behavioral disciplines, and there are not many. Not only sociologists but also developmental, social and cognitive psychologists have perceived its crucial role in socialization. Drawing on the early work by Vygotzky (published in the US as late as 1978) and Piaget (1978), psychologists showed that a parental attitude oriented to elicit norm internalization predicts children's later wellbeing and even their inclination to other-regarding behavior (Ryan and Deci, 2000).

Nonetheless, our scientific definition and understanding of the process of norm internalization is still fragmentary and insufficient. The main purpose of this chapter is to argue for the necessity of a *rich cognitive model* of *norm internalization* in order to (a) provide a unifying view of the phenomenon, accounting for the features it shares with related phenomena (e.g., robust conformity as in automatic behavior) and the specific properties that keep it distinct from them (autonomy); (b) model the process of internalization, i.e. its *proximate causes* (as compared to the distal, evolutionary, ones; see Gintis, 2003, 2004); (c) characterize it as a *progressive* process, occurring at various levels of depth and giving rise to more or less robust compliance; and finally (d) allow for *flexible conformity*, enabling agents to retrieve full control (Bargh et al., 2001) over norms which have been converted into automatic behavioral responses (Epstein, 2006). Thanks to such a model, it will be possible to adapt existing agent architectures (such as EMIL-A, cf. Andrighetto et al., 2007a) and simulation platforms (EMIL-S, see Troitzsch, 2008, etc.) to test hypotheses concerning (a) individual and social effects of internalization, (b) factors favoring or hindering internalization, and (c) the evolution of internalization in future societies.

Throughout the text, the process of norm internalization is meant as a mental process that takes (social) norms as inputs and gives new goals of the internalizing agent (from now on, the internalizer) as outputs. Emotions, playing a significant but not necessary role in this process, will not be investigated at this stage.

20.2 Related Work

Contributions to explain internalization are sometimes based on reinforcement learning theory. Scott (1971), for example, theorized that norm internalization leads to robust compliance, provided the *external sanctioning system is never completely abandoned*. Unfortunately, this explanation is incompatible not only with the view that "…social norms can get internalized to the extent that *they do not need social enforcement*" (Basu, 1998), but also with experimental evidence. For example, subjects playing ultimatum games are found to follow fairness considerations even when unobserved (Bicchieri, 2006).

In the last few years, a strong renewal of interest around the notion of norm internalization appeared in the evolutionary game theoretic study of cooperation and pro-social behaviour. Gintis (2003) argued that increase in social complexity of early human society produced a rapidly changing environment, which in turn posed an adaptation problem to genetic mechanisms for altering goals. Internalization of norms is adaptive because it "facilitates the transformation of drives, needs, desires, and pleasures into forms that are more closely aligned with fitness maximization." But how did it evolve?

Some authors (Bicchieri, 2006; Epstein, 2006) conceive of norm internalization as a process leading to a sort of automatic, or thoughtless conformity. People, observed Epstein (2006), blindly conform to the norm: the more they have done so in the past, the more they will redo it in the future. Agents learn not only which norms to conform to, but also how much they should think about them. In the author's view, internalization is learning not to think about norms. Bicchieri (2006) has a sophisticated explanation, which leads to an equivalent conclusion. Agents learn what the norms are through shared expectations, which by definition they prefer to correspond to. Once found out what the norms are, they organize their beliefs into script-like structures, including the contexts, relationships and conditions in which they found out the norms. When later such belief structures are activated by ongoing activities or contexts, the corresponding norms will be activated as well, and conformed to *thoughtlessly*. Does this mean internalized norms are thoughtlessly complied with? What about the difference between an action done out of a "sense of duty" and a habit? Also, what about norms decided upon? Is there still space for a theoretical and operational difference between internalized and non-internalized norms? How about moral dilemmas? On the other hand, what about our capacity to take decisions also about the norms we usually apply thoughtlessly? Living in a right-hand circulation country, I don't take decision every morning whether I should go left or right. But if I happen to go to UK, I do: I retrieve control of my behaviour and think about which way I should go in order to obey the norm, and not endanger myself. How is this possible?

20.3 The Present Work

This work is aimed to propose agent based modelling, and in particular rich cognitive modelling, as a framework for casting a theory of the cognitive underpinnings of internalization, and characterize norm internalization as a progressive, multi-step process, leading from externally enforced norms to norm-corresponding goals, intentions and actions pursued for their own sake.

In order to understand this process, some preliminary notions should be clarified.

20.3.1 Goal Dynamics

People act on pre-given goals (hardwired in their minds), which are modified, expanded or reduced during the course of their lives. The process of goal modification may be caused by non-cognitive factors, such as hormonal processes, chemical substances, etc. but may also originate from learning and reasoning mechanisms (see Conte and Castelfranchi, 1995).

Under the effect of social inputs, goals can be generated anew *via* cognitive factors, as goals *relativized* to other mental states (e.g., social beliefs). A goal is relativized when it is *held because and to the extent that* a given world-state or event is held to be true or is expected to occur (Cohen and Levesque, 1990). Tomorrow, I want to go gather mushrooms (relativized goal) because and to the extent that I believe tomorrow it will rain (expected event). The precise instant I cease to believe that tomorrow it will rain, I will drop any hope to find mushrooms.

New goals may be relativized to social beliefs. These are relativized *social* goals (see again, Conte and Castelfranchi, 1995). When they are positive or pro-social, the process of generation is called goal-adoption: an agent, the adopter, generates a new goal g_i because and as long as she believes g_i to be a goal of the adoptee.

20.3.2 Present Work: Norms' Mental Dynamics

The concept of norms we refer to draws on the theoretical backgrounds (Conte and Castelfranchi, 1995, 2006) and the outcomes of the EMIL project. In particular, norms are meant as behaviours spreading through a given population thanks to the spreading of a corresponding normative belief. For example, fastening seat belts while driving a car is a norm if it spreads under the assumption that this behaviour is a norm within, say, the Road Traffic Act. In an autonomous agents' perspective, to comply with norms requires that agents form normative beliefs and achieve ordinary normative goals, *i.e., normative goals are goals relativized to normative beliefs.* As we know (see the Ontology) normative goals are generated because and maintained as long as the corresponding normative belief is held, i.e. the belief that a state of the world or an action is either obligatory or forbidden or permitted.

There are at least three main types of normative beliefs:

1) That either a given action or a given state of the world, for a given set of agents, is either obligatory, forbidden or permitted (main normative belief)
2) that believer is (not) a member of the set of agents subject to the norm (normative belief of pertinence)
3) that *positive sanction is consequent to norm obedience and negative santion is consequent to norm violation (norm enforcement belief)*.

Often, the last type of belief plays a crucial role in compliance: taxpayers may cease to contribute as they start to believe that tax evasion is not punished. It may also pay a role in norm recognition, since especially legal norms are expected to be supported by sanctions.

The process of goal-adoption is turned into *norm-adoption* when normative beliefs generate normative goals, usually by reference to external enforcement (sanctions, approval, etc.). If no such a goal is generated, the norm will be violated.

On the other hand, a *norm is internalized* when the norm addressee complies with it independent of external sanctions and rewards. In such a case, the normative goal is no more relativized to an expected sanction, but only to the normative belief. Hence, we claim that

> ***Internalized normative goals** are goal relativized only to the belief that there is a norm on a given action or state of the world and result from a process taking ordinary normative goals as an input.*

20.3.3 Types and Levels of Internalization

Norm-internalization occurs at different levels, concerning different aspects of the mind. To understand them, a short glossary is needed. Throughout the chapter, we will speak of goals, intentions and actions from the point of view of computer science and autonomous agent theory. In particular, a *goal* is a wanted world-state that triggers and guides action (see Conte, 2009); an *intention* is an executable wanted world-state, chosen for execution (see the BDI theory of agency); an *action* is an executed intention (see the section on action in computer science in Segerberg et al., 2009), i.e., an intention incorporated into a given behaviour.

i. *Internalized normative goal*: the normative goal is no more relativized to the external enforcement but only to the main normative belief,
ii. *Internalized goal*: this time the goal is no more normative, i.e., it is no more relativized to a normative belief. The internalizer has lost track of the normative origin of her goal. The normative belief may still persist but the agent pursues the corresponding goal irrespective of it. For example, I may adopt the norm to stop at crossroads because I do not want to get a fine. So far, I have formed a normative goal relativized to external sanctions. If I see no policemen around, I can move on ignoring the norm. Suppose I then gradually realize (maybe under the effect of a couple of non-lethal but serious car crashes) that not to stop at crossroads is dangerous. If I always stop, even when no policeman is around, I have internalized the norm. If I stop even when I know that the norm asks me only to slow down, I have transformed the norm into an internal goal. Hence, our second claim:

***Internalized goals** are goals no more relativized to normative beliefs and result from a process taking internalized normative goals as an input.*

iii. *Internalized intention:* the output is a goal no more relativized to a normative belief, chosen for execution, and activated by specified perceived events. For example, consider a request of information from a passenger. The request is a trigger that the requestee *cannot* easily ignore, without at least pretending not to have realized it. If circumstances make this incredible, the requestee has got but one option, to answer the request. This goal is *automatically* chosen for execution. It has become an internalized intention.

***Internalized intentions** are internalized goals activated by perceived events.*

iv. *Internalized action:* the output is a conditioned action fired by a trigger, a perceived event (for example, stop when traffic lights get red). The decision-making is avoided, as the trigger activates a conditioned action in the internalizer's repertoire. Interestingly, however, under the effect of other perceived events, conditioned actions may be blocked for the time interval required to process a disturbing or interfering event, and restored later (see Bargh et al., 2001) in a semi-conscious fashion. Here, not only the intention but also the behavioral response is automatic. In the traffic light example, this consists of the sequence of movements necessary to activate the car's breaks, a behavioral response so deeply internalized that one can hardly make it explicit.

***Internalized actions** are behavioural responses activated by perceived events.*

So far, we have proposed some notions corresponding to specific forms and processes of internalization in terms of goals, beliefs and their interplay. It is fairly clear that such processes require a rich cognitive platform, namely a BDI-type of architecture. EMIL-A seems a good candidate for this undertake.

Before showing how implement the processes and forms of internalization we have defined so far, let us hypothesise some factors that might lead to them, mainly based on the cognitive psychological literature.

20.3.4 Factors. Some Preliminary Hypotheses.

Factors affecting internalization should be investigated cross-methodologically, comparing simulations with experiments on real agents. What we provide below are only preliminary hypotheses.

Leading to Internalized Normative Goals

Why do agents observe a norm irrespective of external enforcement? We suggest that the principal factor of this type of internalization is *consistency* (see also McAdams, 2008). This mechanism operates at two stages: first by selecting which norm to internalize, and later by enforcing it (self-enforcement) and controlling that one's behaviour corresponds to it (self-control).

Consistency of new norms with one's beliefs, goals and previously internalized norms, plays a crucial role in the selection process. Successful educational strategies favour internalization processes (see King, 2008), often by linking new inputs with previously internalized norms. Analogous considerations apply to policymaking. Consider the antismoking legislation: the efficacy of antismoking campaigns based on frightening announcements and warning labels (e.g., sentences like "Smoking kills" on cigarette packages, see Goodall, 2005) is still controversial. One of the factors reducing efficacy is the effect known as *hyperbolic discounting* (Bickel and Johnson, 2003; see also Rachlin, 2000), a psychological mechanism that leads to invest in goal-pursuit a measure of effort that is a hyperbolically decreasing function of time-distance from goal-attainment, and leads people to procrastinate energy-consuming work until the very last moment. Due to hyperbolic discounting, people, especially young people, are unable to act under the representation of delayed consequences of current actions. Much more efficacious seems to be a previously emerged and diffuse set of social norms, the *live-healthy* precepts, highly consistent with the antismoking legislation.

Consistency is also crucial for the efficacy of self-enforcement. This time, hyperbolic discounting works in favour of norm internalization. Agents may find easier to commit to a given course of action than to actually execute it: I can show greater enthusiasm in promising myself that I will quit smoking tomorrow, than tenacity in keeping to the promise when time is come. However, my promise, which was facilitated by the mechanism of hyperbolic discounting, will activate a maintenance goal (a goal that is usually verified and is activated when it is not; see again, Cohen and Levesque, 1990), i.e. keep to the promise made to myself (or more significant variants of it, such as preserve self-esteem, be consistent, don't lose credibility before one's eyes, etc.). Thanks to hyperbolic discounting, I took a risky step: I set a new obligation on myself. This is enough, if not for incorporating it into my behaviour, at least for triggering self-enforcement, which is a process following the selection of inputs. Self-enforcement supports the process. The internalizer will take over the enforcing task, and starts to apply self-punishment when violating the new norm and self-reward when complying with it, either in terms of self-evaluation or in terms of negative emotions and feelings toward oneself.

Leading to Internalized Goals

Internalization of the source depends on a number of factors, including

i. *Self-enhancing effect* of norm compliance: the norm addressee realizes that she achieves one of her goals by observing a given norm. Suppose I succeed in refraining from smoking and that after a few days, I realize an advantage that I had not perceived before: food starts to taste again. This discovery generates a goal (quit smoking to enjoy good food), not relativized to the norm but supporting it: I have converted the norm into an ordinary goal. Whether this goal will be strong enough to out-compete addiction is another matter.
ii. Norm's *salience* (see Campennì et al., 2008), defined as the number of times any given norm is observed and defended per time unit (see also Troitzsch, 2008), is another factor of internalization. The higher the salience of the norm, the more deeply it is internalized. One good example is the observance of a special food regime. An animalist decides to go on a vegetarian diet for ethical reasons. After a while, she will strongly dislike the taste of meat, and even the faintest smell of it. This phenomenon is probably at the origin of the converts' higher efficacy in making proselytes (known as the *convert effect*, see Levine and Valle, 1975).

Leading to Internalized Intentions

Intentions are goals chosen for execution, meaning, executable goals. Internalized intentions are goals that are stored as high-priority and executable, and fired under given circumstances on the grounds of their priority. As intention priority is known to be a function of time urgency (see, for example, Zhang and Huang, 2006), it can be hypothesised that urgency plays a role in intention internalization. In particular, one can argue that the more a given norm allows to answer problems frequently encountered under conditions of *urgency*, when time for decision-making is none or scanty, the more likely that norm will be internalized as an intention, a goal chosen for execution.

Let us go back to the example of answering passengers' requests for information. The requestee can ignore the request only if she (pretends she) has not realized it. The phenomenon is rather complex. The requestee feels *entitled* to ignore the request until she perceives that communication has succeeded: if it is undeniable (as for example, eye contact inevitably reveals) that the request got through, she can no more ignore it. The social norm (give help when asked) has been converted into a complex action plan: give informational help under specified conditions (when it is patent that request got through).

Internalized Action

Norms either describe world-states to (not) be achieved without making explicit how this should be done ("*Keep your room cleaned*"), or actions to (not) be accomplished ("*Shut up!*"). In the latter case, the more salient, explicit and operational the norm, the more likely it will be internalized as a conditioned action, a routine activated under specified conditions. Thus, agents cover their mouth while yawning, smile and/or utter a greeting formula when entering a private space open to the public, etc. The norm can be not only internalized in standard routines or habits, but also incorporated into material artifacts (silverware, hankies, etc.) that activate those routines.

Doesn't this type of internalization, after all, correspond to what Epstein called *thoughtless* conformity? Of course, it does. However, this is not the only form of internalization. The crucial question, here, is to provide a common ground, an agent model that can exhibit all of them, and, what is more difficult, can shift form one to the other. We need to account for reversible routines, or, which is the same, for flexible conformity. How can the internalizer gain control again over an automated action, and refrain from applying a given routine? How can she move on when the traffic light is red but the policeman invites her to proceed? Even if another routine is activated by the new event (policeman invitation to proceed), how and why is one routine (stop) interrupted and fired the complementary one (move on)? How is the conflict solved? How can we decide upon automatic behaviors? Indeed, the confine between automated behavior and conscious will is fuzzier than is commonly believed to be the case. Cognitive psychologists (Bargh et al., 2001) show that automated behavior need not be rigidly removed from consciousness, and that goal-attainment need not be conscious nor deliberate.

In principle, a *modular* normative architecture nicely fits flexible automaticity: internalized actions do not prevent norms from being processed at higher levels. Inputs that interfere negatively with a given routine may concurrently activate a normative belief and lead to a normative goal. The decision-maker will then establish whether the new goal should be achieved or the old routine restored. It is also possible that the new goal activate another routine, conflicting with the preceding one. How to combine conscious and automatic goal attainment in intelligent agent architectures is a fascinating question, far beyond the scope of this chapter. However, it is one of the inspiring ideas at the grounds of the current, promising work on hybrid cognitive architectures (cf. Sun and Wu, 2006).

20.3.5 Internalizer: A BDI Architecture

The normative architecture EMIL-A (see Andrighetto et al., 2007a for a detailed description) consists of mechanisms and mental representations allowing norms to affect the behavior 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.

As any BDI-type architecture EMIL-A operates through modules for different sub-tasks (recognition, adoption, etc.) and acts on mental representations for goals and beliefs in a non-rigid sequence.

To show how EMIL-A works, a sketch of an "ideal" and complete mental path of a norm will be provided. After recognition (Campennì et al., 2008 for a detailed description), a norm becomes a *belief* stored in the normative board of the agent representing an obligation - a prescription or prohibition – on a given state of the world or on a given action. The normative board is a portion of the long-term memory where normative beliefs are stored, ordered by salience.

A normative belief will be inputted to the norm-adoption module (see Figure 113). Under external enforcement, a normative goal will be generated from the normative belief relative to the expected enforcement.

Under the effect of factors like high consistency, the normative belief will be sufficient to generate the normative goal (see the full black arrow in Figure 113).

Once formed, a normative goal is inputted to the decision-maker and compared with other goals possibly active in the system. The decision-maker will choose which one to execute and convert it into a normative intention (i.e. an executable goal).

Once executed, this will give rise to norm-compliance and/or norm-defense - direct or indirect punishment – and/or norm transmission through communication. Otherwise, it will eventually be abandoned, solution that brings again to norm violation.

This rich characterization of the representations and processes underlying a norm-compliant behavior should not give the idea that behavioral conformity is always based on complex reasoning and deliberation.

A crucial aspect of EMIL-A is the possibility to account for the occurrence of interruptions, modifications and deviations from the processes described so far: norm conformity and obedience can be converted into an internal goal, intention and even become a habit, a (semi) automatism, a routine behavior.

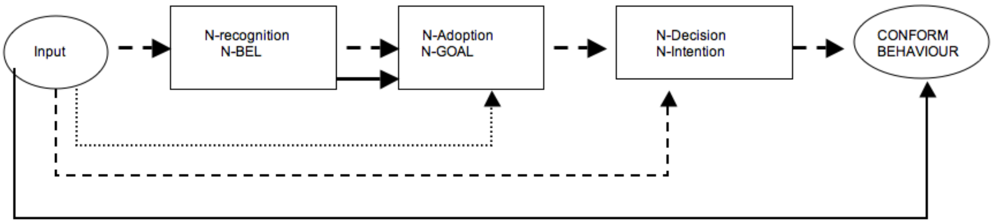


Figure 113. Illustrates how EMIL-A is able to account for different forms of norm internalization.

The thick line leading from norm-recognition to a normative goal shows an internalized normative goal. When the norm is recognized as highly consistent with one's beliefs, goal-base, and pre-existing norms, a normative goal, non-relativized to external enforcing agencies is formed.

The dotted line leading from the input to the norm-adoption mechanisms shows the generation of goals no more *relativized to normative beliefs*, i.e. goals that have lost track of their normative origin. In our architecture, if the salience rate of a normative belief increases above a certain threshold (the norm is frequently adopted and turned into a normative action), the norm leaves the normative board to become a highly-valued (internalized) goal.

The broken line linking the input to the decision-making module makes visible that, once internalized a normative goal, the normative input is directly converted into an intention activated by a perceived condition.

Finally, the grey line leading from the input to the normative behavior, describes an *automatic* behavioral answer (*thoughtless conformity*).

20.3.6 Hypotheses and Follow-up Questions

Hypotheses concerning the effect of internalization follow from the properties of internalization analyzed so far.

To some extent the advantages of internalized norms are easily identifiable: Norm compliance is expected to be more robust if norms are internalized than is the case when conducts are ruled only by external sanctions because they emancipate the norm addresses from external, sanctioning entities.

However, what should we expect from comparison between internalized and fully endogenous mental states? Internalized goals are here hypothesized to be more persistent and lead to a more vigorous *goal-attainment* (Bargh et al., 2001) than originally inner goals. The argument is based on prospect theory (Kahneman and Tversky, 1979; for a recent contribution, see Abdellaoui et al., 2007), which assumes loss aversion, i.e. people's tendency to strongly prefer avoiding losses to acquiring gains, as a prominent feature of human beings. Internalized goals are already formed in the mind: unlike fully endogenous goals, internalized ones are selected from among goals initially acquired under the effect of external influence. The more effort is invested in the attainment of these goals, the lesser they will be abandoned later, the more vigorously will they be attained.

A specific hypothesis is based on *confirmation bias* (according to which people are likely to accept inputs that confirm their beliefs, and to reject disconfirming ones; Nickerson, 1998; for a recent contribution, see Sternberg, 2007). Based on it, the internalizer is expected to show higher intolerance with regard to norm violation than both those who follow the norm under external enforcement and those who are spontaneously motivated to behave accordingly. Violation is a disturbing factor for the internalizer, which might lead her to weaken and even revise the commitment made. Hence, norm internalizers are expected to be more consistent and compliant than externally-enforced norm observers and endogenously motivated agents. A further consequence of the theory is that agents are much better at defending the internalized norms than externally-enforced observers. A consequence of the latter prediction is that norm internalization is decisive, if not indispensable, for distributed social control. Internalization is probably not only one mechanism of private compliance, but also a factor of social enforcement.

In short, internalization is a good predictor of compliance and second order cooperation (i.e. urging others to comply with the norms, see Horne, 2007). But what are, if any, the disadvantages of internalization? Again, theory leads to formulate some hypotheses. First, internalization takes long, and it is not necessarily successful: self-training may be too hard, and often requires several trials before getting through. People almost never quit bad habits and antisocial conducts on the very first try. Secondly, failures may have counter-effects: after a number of unsuccessful attempts, loss of self-esteem and feelings of helplessness may render too weak and fragile future private commitments, and jeopardize internalization. The question of course is what are the factors that may favor its success on first try. Third, internalization emphasizes selection of inputs, autonomy. Moral autonomy is often encouraged at least in Western societies, but it may have counter-effects as well. For example, it may lead to excessive variance in compliance. Fourth, how does internalization perform in societies characterized by a high degree of norms, possibly in sharp conflict with one another? One might expect that internalization is incompatible with perceived norm and value conflicts. Could it be that the future of societies is characterized by fragile and variable internalization? Another question for investigation.

20.4 Conclusions

When Vygotsky first formulated his theory of internalization, he noted that only "the barest outline of this process is known" (1978, p. 57). We do not know, yet, how people manage to internalize beliefs and precepts with a reasonably adequate success, partly because we still do not agree about what to investigate, what it is that we mean by this notion. Consequently, no useful notion and model is available for applications, despite its wide and profound implications.

Questions such as how norm internalization unfolds, which factors elicit it, which are its effects, obstacles and counter-indications, are issues of concern for all of the behavioral sciences. The internalization of social inputs is indispensable for the study and management of a broad spectrum of phenomena, from the development of a robust moral autonomy to the investigation and enforcement of distributed social control; from the solution to the puzzle of cooperation, to fostering security and fighting criminality, etc. A computational, simulation-based approach is crucial here, as it urges us to formulate the process of internalization as clearly and analytically as requested by the purpose of computational reproduction.

After a cognitive definition of the subject matter, the chapter presents and discusses the building blocks of a rich cognitive model of internalization as a multi-step process, including several types and degrees of internalization. Next, factors favoring different types of internalization are discussed. The modular character of BDI architectures, like EMIL-A is shown to fit the approach advocated in the chapter. In future studies, EMIL-A will be augmented with a model of internalization along the lines here presented, in order to run multi-agent simulations, comparing internalization with other forms of compliance against a number of measures (including convergence, robustness, and adaptation to fast-changing environment).

What is the value added of a rich cognitive model of internalization, as compared to simpler ones (e.g., reinforcement learning)? There are several competitive advantages.

First, reinforcement learning does not account for the main intuition shared by different authors, i.e. the idea that internalization makes *compliance independent of external enforcement*.

Second, a rich cognitive model, namely a BDI architecture with its specific modules, *accounts for different types and degrees of internalization*, bridging the gap between self-enforcement and automatic responses.

Third, a BDI architecture accounting for different levels of internalization *allows flexibility to be combined with automatism*, as well as thoughtless conformity with autonomy. A BDI system can host automatism, but a simpler agent architecture does not allow for flexible, innovative and autonomous action.

Chapter 21 Looking Forward: A Mini-road-map of Questions about Normative Behaviour

Bruce Edmonds, Martin Neumann et al.

The field of *all* human normative and norm-like behaviour is various and complex. Clearly there may be different process of norm development and maintenance in different social contexts and within different social contexts. Thus, inevitably, the empirical questions such as: when does such a norm develop, why does it emerge, how does it relate to the social institutions and other social mechanisms in its context, etc. have only just been started. Mapping the 4-way relationship between social context, social conditions, cognitive mechanisms and resulting processes is a huge undertaking, in which simulation can play an important part, but that will also require the integration of many different approaches that can help untangle the rich dynamic nature of norms and their dual, social & cognitive aspects. However, given the results of the EMIL project, it is now possible to untangle some of the various and intertwined cognitive and social effects over time, and thus make such an undertaking coherent.

A side-product of the EMIL project has been a set of further questions about norms and norm-like behaviour which map out some of the many issues that remain unresolved. The answer to some of these questions might well be specific to each-context or range of kinds of situation. Some of the questions might have answers with a wider scope. Untangling their answers and their scope will require a mixture of theoretical, empirical and simulation studies involving both qualitative and quantitative studies. They are listed below with a brief commentary and sub-questions below.

Their purpose is to be a “mini” road map for future research into normative behaviour. In order to focus both evidence gathering and simulation concerning the dynamics of norms, we built up a series of key questions concerning the development and nature of norms. The aim is that each of these are answerable with respect to the evidence from case-studies and simulation outcomes (at least to some degree in specific circumstances) but sufficiently important to have a wider impact as a hypothesis about Norms.

The criteria for these questions are that they should be:

- as general as possible whilst the criterion above holding
- as clear and unambiguous as possible

Answering these questions will represent substantial progress in understanding the answers and significance of the proposal questions. Each question may have a different answer in different observed cases. Each of these has a headline question; a brief explanation; related questions; and comments.

21.1 To What Extent are Norms and Conventions Linked to Group Membership?

Clearly in some cases norms and conventions seem to be very specific to a well defined group. For example, keen golfers have a mode of dress and behaviour that is specifically displayed at their golf club and when with golfing friends (at least in the UK). However other conventions seem to be have a wide distribution, for example a geographic region. Thus although norms and conventions are certainly propagated within groups, it is unclear whether the acceptance of a norm or the adoption of a convention is dependent upon a wish to be a member of a certain group. This might be explicit for the person concerned, in that they may be making an active and deliberate effort to conform (as with many immigrants at places of work) or an unconscious adoption of behavioural patterns.

21.1.1 Related Questions:

- If norms are linked to group membership, to what extent does such link depend upon the role of norms (and conventions) in maintaining/strengthening the group?
- To what extent does it depend on a sort of “memetic drift”, such that normative beliefs find lesser obstacles to their spreading within groups than across groups?

- How is norm establishment related to group formation and dynamics?
- How do group “boundaries” form and what is the effect on norm salience?
- What is the relationship (if any) between individual identity forming and norm internalization?
- What is the effect on norm salience of agents participating in multiple intersecting groups to which different norms may apply or a common norm may have different salience? also group dynamics (what about the shape of this dynamics? for example loops?)

21.1.2 Comments

Group membership (the network of peer groups and identification with – eventually even quite distant – reference groups) is essential for the formation of social Identity. Identity is the most advanced form of internalisation; social norms become part of the individual identity. In terms of agent architecture: obligations become desires – comparable to the role of salience in the normative board of the EMIL architecture. Group membership, identity and internalisation are intimately linked. Peer group norms do not only strengthen the group but also individual identity and hence individual well-being.

21.1.3 Examples

Some Wikipedians emphasised that contributing to Wiki was a duty for their sense of identity (when they find a bias in articles related to their own identity; e.g. an Algerian who found a bias in articles about Northern Africa. However, this is only personal observation and not a statistically valid argument). Both peer and reference groups refer also to interaction history. Norm establishment is thus closely related to group formation and dynamics. E.g. in the Balkan it is a norm to “hate your neighbours”: obviously this refers to interaction history – comp. Srebrenica. This is also a process of boundary formation. Such cases of ethnic (or gender, as another example) identity are instances of identity formation that cannot be chosen completely voluntarily. By considering the case of Srebrencia, it seems that interaction history might be an essential component of “memetic drift”. The norm to “kill your neighbours” has been heavily propagated by this event. However, also communication media have to be taken into account as part of interaction history. There are people who have personally experienced this event and persons who have heard about it only in the media. However, trust in the media can change personal values even without personal experience. This might be a kind of memetic drift.

21.2 To What Extent are Norms Active in Creating Conventions?

It may be that in some cases norms exist before any corresponding convention. The norm might be imported from elsewhere and the conventions might then develop along the same lines. For example, it may have happened that missionaries may have insisted that it was wrong for women to walk around bare-breasted, and later this to become a convention in that society that one indeed does not walk around like this. Of course it may not be obvious whether such a norm causes the later convention, for example it may be that the norm is rejected on its own grounds, but then mimicked due to its association with a high-status group. It is certainly the case that there have been several calls for legislation to be changed purely on the grounds that this would encourage a change in convention (e.g. laws against drugs).

21.2.1 Related Questions:

- Can one comply behaviourally without internalising a norm?
- What is the comparative role of social embedding vs. internalisation?
- Are specific phenomena meant by the notions of conventions, manners, customs, social habits etc.
- To what extent are norms important in maintaining conventions?

21.2.2 Comments

Anthropologists asked natives why they wear a certain kind of penis decoration. After long debates they were unable to answer the question and decided that it was simply a stupid question. This might be an example of a convention that is not substantiated by normative *reasoning*.

To the question whether one can comply behaviourally without internalising a norm, self-determination theory provides an answer: it posits a scale of degrees of internalisation: from purely external regulation to integration (into the self).

With regard to the question of the comparative role of social embedding vs. internalisation, I think, both is closely connected: identity theory posits that peer groups (social embedding) are crucial for social identity and identity is a strong form of internalisation.

21.3 To What Extent Do Norms Arise Out of Existing Conventions?

The opposite of the above case is when the convention arises first and then is recognised and reified as an explicit norm. Thus in the UK early in the 20th Century a law was passed that people should queue at bus stops when waiting for a bus. Before that people had, in fact, always queued (this being a queue-loving culture), but it was felt that it was necessary to enforce this. In fact nobody was ever prosecuted using this law, and it was repealed at the end of that century. It may be that when a convention has arisen and has been explicitly recognised as useful, that people might seek to reinforce this by attempting to establish explicit norms (possibly only if they think the convention might disappear).

21.3.1 Related Questions:

- If conventions help people to identify as members of a group, will the resulting barriers contribute to the emergence of norms?
- To what extent are Norms active in maintaining Conventions?
- How to account for the mandatory nature of conventions?
- In what way do social and technical artefacts influence social norm and institution forming?

21.4 To What Extent are People Aware of Norms Explicitly During Their Decision Making?

In the studies of behaviour among Wikipedia contributors explicit norms were rarely invoked, but rather the cooperation seemed to be regulated on a day-day basis as a result of an informal social convention. However this does not necessarily mean the norms are unimportant – it may be that the social convention would quickly disappear if it were not supported by a norm. So, for example, it may be a convention to be polite, but when there is a power-differential and people realise that in some circumstances there will be no come-back if they are not, then the convention of politeness may disappear. It is extremely rare for adults to remind others that they have an obligation to be polite, but the convention of politeness may be nonetheless strong.

21.4.1 Sub-questions:

- Can norms act on people without them being aware of it?
- How do norms get automated?
- To what extent and how can decision-making and reasoning upon automatic behaviours be (re)established?

21.4.2 Comments

It seems to me that the example of politeness is an example of a norm that individuals need not be aware of when they act accordingly. The rare cases of explicit norm invoking in the Wiki case seem to be quite similar: people already act with a pro-social attitude. Norm invoking is only necessary if people do not follow the norm. However, I guess, it will always be possible to deliberate about normative and provide reasons – if there is time enough. This might be different is the case of conventions.

21.4.3 Examples

Recently Wikipedia was shortly banned in Germany by a judge because a politician found information about him that he didn't like or regarded as false. Appeal to a judge is a norm invocation at a different level. In the US election campaign so-called "edit wars" took place: some changes in Wikipedia sites about politicians (see also Chapter 5) could have been traced back to ULR addresses in the offices of the politicians. It seems that the normative self-regulation of Wikipedia doesn't work in such cases. As the Wiki norm of politeness maybe traced back to pre-existent pro-social attitude, the edit wars can be traced back to pre-existent strategic interests. Hence, it seems that domain specific norms (and their violation) are embedded in a web of different and intersecting normative domains/loyalties.

21.5 To What Extent are Norms Reasoned About in the Determination whether to Fall in with a Social Convention?

21.5.1 Comments

Normative reasoning is involved in moral judgements as studied by Kohlberg (1996): In the example of burglary of a pharmacy, Kohlberg asks this question to children of 7, 14 and 20 years of age. The focus of his research is the change in the reasons the children give for their judgement of how to act in this moral dilemma: while younger children typically orient their judgements on external authorities (to act according to *conventions*: don't steal, because the police will catch you and you will be brought into prison), older children (sometimes) appeal to universal moral principles (act according to morality: the absolute value of life allows to break the norm not to steal). Hence moral *reasoning* is a way by which humans resolve *conflicts* between different obligations. However, this is only reached after a long cognitive development – social embedding is essential for the developmental process.

The (human) cognitive development that enables internalisation of social norms and to finally become a morally responsible person is correlated to the possibility to get some mental diseases. I'm not an expert on this, however, there exist mental diseases (some sorts of schizophrenia) that younger children cannot get.

21.6 To What Extent are Norms Underpinned by Specific Advantages and Disadvantages to the Individuals Concerned?

Clearly norms seem to be effective even in situations where there is no advantage to the individual concerned, for example not dropping litter even in a place one is passing through once and where no one can see you. However, it may be that norms where there is no basis (direct or indirect) w.r.t. the goals of individuals might not persist in the long run. This does not mean that norms will disappear quickly once the original "underpinning" has disappeared, but that there is some effective selection of norms by a group in the long run. This "selection" may only occur in special circumstances (e.g. a sudden change in group membership) or when there is a competing norm that also has a positive effect for the individuals concerned. Thus the norm that one should not eat everything on one's plate changed to one of eating everything up as a result of the 2nd World War, and this norm has persisted until recently when norms concerning healthy eating are being actively promoted by the government.

21.6.1 Related Questions:

- How are benefits/disbenefits assessed against a hierarchy of goals some of which are potentially in conflict?
- How does this questions link to the process of fuzzy recognition and "satisficing" of cognitive constraints - including affect?

21.6.2 Comments:

Generally, in moral *dilemma situations* as studied by evolutionary game theory, individuals have advantages in the long run by following norms.

21.7 To What Extent are Norm-like Reports a Rationalisation of Conventions that Have Arisen?

It is sometimes the case that one participant will say that a certain regularity is the result of a norm and another that this is the result of a convention. There are several possibilities here, including that a convention is reported "as" a norm. It may be that newcomers unconsciously copy others in a place of work and wear a suit, when asked why they might rationalise their behaviour saying "here one has to wear a suit" whereas in fact it was just a convention. The problem with this is that it then becomes difficult to distinguish norms and conventions when looking at evidence – instead we would have to consider the operational difference between norms and conventions reported as norms.

21.7.1 Related Questions:

- What is the ontological status of a norm?
- How could we distinguish between first-person rationalisations of their own behaviours and explanations that provide substantive information about conformative behaviour?

21.8 To What Extent are Norm-like Behaviour the Result of Immergent Processes?

An immergent process is where there is some causation from a property of the whole acting on the behaviour of its parts. This is the opposite of emergence, which is generally taken to be when the behaviour of the parts causes some (new) property of the whole. In this particular case, an immergent process may result when individuals recognise conventions and/or norms of the group, and this affects or restricts their behaviour as individuals.

21.8.1 Related Questions:

- How essential is such a process to explain norms and conventions?
- Can normative behaviour be explained my purely bottom-up coordination, that just appears to involve immergent processes?
- Are some kinds of norm-like behaviour possible without it (if so which)?
- What kind of abilities do the individual's need in order to recognise global patterns?
- What sort of communicative structures or access need to be in place for suc?

21.8.2 Comments

The (purely) immergence aspect is broadly in accordance with Talcott Parsons' view on action: according to this view, the ends of action are (to some degree) determined by norms that are external to the individual actor (Parsons, 1937/1968). Norms are a social fact, that can only marginally influenced by individuals. While it seems plausible that indeed we both consciously and unconsciously follow social norms that we simply take as given, the purely immergence view seems to be too restricted, as it does not take into account a) emergent processes, b) social change, c) deviant behaviour and d) the question how we *become* such norm obeying individuals. In short, Parsons had not an "in the loop" perspective. So far, I would argue that the Wiki simulation captures a) and b) together with immergent processes.

The question whether some kind of norm-like behaviour without immergent processes is possible seems to be related to the question of individual advantage: if there is a positive answer to the question whether norms are underpinned by individual advantage then it might result by individual strategic deliberation (without "knowing" any macro level norm). Game theory gives a positive answer to the question of individual advantage and -in fact- in evolutionary game theory the agents do not "know" any norms.

21.9 What are the Observable Macro-level Differences between Norms and Social Regularities?

21.9.1 Comments

Normative behaviour need not be a statistical regularity. In particular if moral reasoning is involved, deviant behaviour is not explained by chance variation, leading to some kind of normal distribution. There is a difference between Norms and the normal. This was the main deficit in purely behaviouristic psychological theories leading to a cognitive turn in psychology in the 1960s.

21.9.2 Examples

An example of deviant behaviour was extensively studied by Kohlberg (1996): his example was of burglary of a pharmacy: Mr Smith's wife is seriously ill, and her life can only be saved with a certain medicine. However, Mr. Smith has not enough money to buy it (As an American Kohlberg was seemingly unable to think of health insurance systems ...) and the pharmacist didn't give the medicine without payment. Should Mr Smith commit burglary, i.e. commit a crime? The decision is not based on chance and hence, the situation might not lead to a binomial distribution of people committing the crime of burglary or not.

21.10 To What Extent are Norms Active in Maintaining Social Conventions?

What are the characteristics of conventions – is it that they are: a spontaneous, non-deliberate, behavioural regularity gradually emerging among two or more agents in interaction, based on: the agent's goal of conforming to that behaviour in order to act like the others, and the mutual expectation? Another core question about conventions is: why do sometimes conventions acquire a mandatory nature, a prescriptive flavour that makes the agents feel obliged to conform to them, as if they were real norms? Let us consider the widespread praxis of tipping. The form of tipping that is the usual practice today is to give a tip after the service is provided and if customers were opportunistic, they could not tip, exploiting the fact that the service had already been rendered by the time the tip is given. Nevertheless things go differently: This practice is quite usual also in those countries where there is no social norm to establish it. Why? Why do people believe and feel they ought to do it?

21.10.1 Comments

Broadly speaking, Kohlberg's investigation of moral judgements can be interpreted as an investigation why people feel they ought to do x,y: moral judgements as judgements about what (and why!) people believe what people are obliged to do. Hence, insofar as norms provide criteria to think about how to behave in a certain situation (contrary to mere conventions) they give *reason* to behave in a certain manner (it is polite to leave a tip, it is a norm to be polite –> so I leave a tip). Here is a relation to normative reasoning.

21.11 Conclusion

Clearly the scope for further work in answering these questions is immense, requiring a considerable amount of empirical and modelling work. However, given the theoretical structures produced by the EMIL project with its socio-cognitive and dynamic approach, it is now possible to embark on these in a meaningful way. This does not lessen the work, but does mean that there can be a unifying simulation structure to disambiguate the issues, and thus lead to a more coherent and comprehensive account of social norms.

Chapter 22 Epilogue

Bruce Edmonds and Rosaria Conte

22.1 EMIL-T Value Added

EMIL-T presents a number of advantages over other approaches to norms.

It is based on a clear ontology and explicit theoretical assumptions. In many other accounts, the assumptions and ontology are not made completely clear. In the light of the development of the architecture, the simulation scenarios and evidence from the case studies, EMIL-T has been refined, and thus constrained by,

- computational plausibility,
- micro-macro linkage, and
- empirical evidence

to produce an integrated, dynamic, explicit and well-founded theory of norms. Many other accounts are essentially either static, vague or not well-founded.

EMIL-T addresses both social and cognitive aspects of norm dynamics. Indeed, it is unique in the level of integration of social and cognitive aspects of norms, as it adopts a combination of immergent and emergent processes. The apparent unity of normative behaviour is accounted for by the close loop with both immergent and emergent processes.

EMIL-T is instantiated as a computational architecture, supported by simulation tools for development and experimentation. EMIL-T is the only theory that is both computationally supported and theoretically well grounded, with a suite of open-access tools available to other researchers, and demonstrated with a suite of implemented models.

In sum, what we have presented is not just a framework and a bunch of hypotheses, but a tested theory. Its feasibility and generality have been demonstrated by a range of very different simulation models capturing phenomena as diverse as pedestrian behaviour and financial decision making.

Below, we summarize the main advances achieved within the EMIL project, wrt the state of the art, as well as the further advances obtained and the lessons learned through the project running. Final considerations about further potential impact of the project’s results will conclude the chapter.

22.2 Advances with respect to State of the Art

As recalled in the chapters discussing the state of the art, norms are hardly seen as resulting from a bidirectional process. Generally, either the process of norm emergence is modeled, or the mental process accomplished on norm-related representations (reasoning, decision-making, etc.) is described. Those who study norm emergence generally ignore how agents represent and reason upon norms. On the other hand, those who treat normative reasoning and representation ignore the process of norm acquisition.

The main advance of EMIL-T is to yield a theory of such bidirectional process, taking into account both norm emergence and norm immergence at once.

In addition, EMIL-T has some additional advantages over the previous treatment of norms, on the side of norm emergence and immergence.

As to the former, our theory allowed two major advantages.

First, it enables us to distinguish non-normative behavioural regularities (such as simple social conformity) from those that are fundamentally normative. Many previous accounts could not show that social conformity and norm-driven behaviour could be distinguished in terms of either global outcomes or effectiveness.

Secondly, it demonstrates that:

- That there is a clear micro-level distinction between leaning conformity and where there are identifiable normative commands and beliefs
- That there can be a clear and testable difference in global outcomes between social processes driven by conformity and normative mechanisms
- Thus creating clear theoretical and testable water between social conformity and norm-driven behaviour.

Third, as to norm immergence, EMIL-T includes a clear account of norm acquisition in normative agents. This clearly distinguishes the EMIL socio-cognitive architecture from those of other architectures, such as the BOID approach. Previous accounts have either left this process as unspecified or only modelled in terms of a quasi-teleological explanations. In EMIL-T, norm immergence is modelled as a two stage pattern recognition in which

- candidate beliefs about norms are formed and
- progressively checked and pruned so that final full-fledged normative beliefs are formed.

This allows for norms from the group-level to become part of individual agent's beliefs and thus affect their individual and hence collective behaviour.

The main result of the EMIL project at the theoretical level is therefore a 2-way theory of norms. In particular, this approach was applied to show how norms can change over time and what are the conditions for their innovation. The development of a normative learning algorithm specific to and integrated with EMIL-T, presented in Chapter 19, allowed this specific objective to be met in a rather successful way.

22.3 Further Advances

During the project lifetime, two additional results have been obtained, which were not primarily aimed at the beginning, namely (a) setting the ground for a unified view of norms, and (b) a preliminary model of norm internalization. Neither has been fully achieved so far. However, EMIL-T at its current state of development sheds some light on how both results could be obtained.

As to a unified theory of norms, it should be noted that the scientific literature generally represents norms as a sort of archipelago with scant interaction among islands which constitute different subsets of norms, legal, social, moral, aesthetic, etc. Each subset is accounted for in a rather specific way, while missing a general, comprehensive notion. From the moral point of view, norms are seen as group-beneficial behaviour performed at some cost for the individual executor. From the legal point of view, norms are defined as obligations issued by established authorities. From the social point of view, a norm is a behavioural regularity solving a problem of coordination or cooperation within a group. The question is, how put bits and pieces together? What do these different notions have in common, if anything? How (far) do they differ?

Our framework paves the ground for such a unifying theory to be constructed. It does so by means of the notion of prescription, or normative command. In our terms, a normative command is a special command that is intended to be adopted by its addressees because it is normative, ie., based on obligations. Sub-ideally, norms are often complied with because they are enforced by a system of sanctions. But ideally, they are meant to be observed as based on obligations, rather than social power. Hence, prescriptions provide the core of normative concepts, while interesting specific features defining different subsets should be found out.

Concerning norm internalization, we provided only some preliminary ideas (see Chapter 20). Unlike previous treatment, we intend to model internalization as a process occurring at various mental levels, and consisting of converting a normative goal relativised to a given (social) belief into a goal no more relativised. Yielding different outputs, from internalized norms, to internalized goals, intentions and actions, norm internalization involves different factors, such as consistency between new and previously internalized norms, norm salience, urgency and operationality of the norm, etc. Interestingly, the modular nature of EMIL-A lends itself to model one fundamental aspect of internalization, ie., flexibility. For example, internalized actions, ie., automated behaviours, ought to be flexible enough as to be blocked when decision-based compliance should be retrieved.

22.4 Potential Impact

Cognitive architectures such as SOAR and ACT-R have been very influential in allowing the development of streams of closely related modelling research. However these are focused almost entirely on individual mental processes and action selection

EMIL-A (with EMIL-S) could allow the development of a similar stream of research but on socio-cognitive modelling, and thus allow the explication and detangling of many classes of social phenomena. EMIL has developed the foundations, theory, architecture, tools and demonstrators... Which allow the easy modelling of many crucially important socio-cognitive phenomena.

EMIL could be likened to the first moon landing – establishing the technology and opening up a fresh area to systematic study – to be followed by many more explorations and projects in the area of socio-cognitive based revelations of normative and norm-like behaviour

EMIL has started the exploration of

- Online behaviour in Wikipedia
- Co-evolution of Pedestrian and Vehicle norms
- Self-regulation in Micro-Finance Groups
- Emergence of trust in exchange.

Many other crucial areas await, including group behaviour and norms in criminal and terrorist organisations; norms surrounding “green” behaviour by households and communities; encouraging new norms for health and safety of citizens (e.g. Keeping to the speed limit whilst driving; understanding and underpinning norms of democratic behaviours – voting, tolerance, expressing opinions, and political involvement. In general, the outputs of the EMIL project can be expected to promote a deeper understanding of informal systems of exchange that prevail, but are often invisible, in large parts of communal life.

References

Abdellaoui, M., Bleichrodt, H., & Paraschiv, C. (2007). Loss aversion under prospect theory: a parameter-free measurement. *Management Science , 53* (10), pp. 1659-1674.

Alchourròn, C. E. (1993). Philosophical foundations of deontic logic and the logic of defeasible conditionals. In R. J. Wieringa, & J.-J. C. Meyer, *Deontic logic in computer science* (pp. 43-84). Chichester: Wiley.

Alchourròn, C. E., & Bulygin, E. (1971). *Normative systems.* Wien: Springer-Verlag.

Alchourròn, C. E., Gardernfors, P., & Makinson, D. (1985). On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic , 50*, pp. 510-530.

Alexander, S. (1920). *Space, time, and deity. 2 vols.* London: Macmillan.

Andersen, P. B., Emmeche, C., Finnemann, N., & Christiansen, P. (Eds.). (2000). *Downward causation. Minds, bodies and matter.* Århus: Aarhus University Press.

Anderson, S. W., Bechara, A., Damasio, H., Tranel, D., & Damasio, A. R. (1999). Impairment of social and moral behaviour related to early damage in human prefrontal cortex. *Nature Neuroscience , 2*, pp. 1032-1037.

Andrighetto, G., Campennì, M., Conte, R., & Paolucci, M. (2007a). On the immergence of norms: a normative agent architecture, emergent agents and socialities. In *Social and Organizational Aspects of Intelligence Symposium.* Washington DC.

Andrighetto, G., Conte, R., Turrini, P., & Paolucci, M. (2007b). Emergence in the loop: simulating the two-way dynamics of norm innovation. In *Proceedings of the Dagstuhl Seminar on Normative Multi-Agent Systems.* Dagstuhl, Germany.

Andrighetto, G., Campennì, M., Cecconi, F., & Conte, R. (2008a). How agents find out norms: a simulation based model of norm innovation. *3rd International Workshop on Normative Multiagent* .

Andrighetto, G., Campennì, M., Conte, R., & Cecconi, F. (2008b). Conformity in multiple contexts: imitation vs. norm recognition. In *World Congress on Social Simulation, Fairfax VA July 14-17, 2008.* Fairfax VA.

Andrighetto, G., Giardini, F., & Conte, R. (2009). Norms through minds. In *Proceedings of WOW04.* Bloomington: Indiana University.

Andrighetto, G., Campennì, M., Cecconi, F., & Conte, R. (Forthcoming a). The complex loop of norm emergence: a simulation model. In K. Takadama, C. C. Revilla, & G. Deffuant, *The Second World Congress on Social Simulation* (LNAI ed.). Springer-Verlag.

Andrighetto, G., Tummolini, L., Castelfranchi, C., & Conte, R. (Forthcoming b). A convention or (tacit) agreement betwixt us. In J. v. Benthem, V. F. Hendricks, J. Symons, & S. A. Pedersen, *Between logic and intuition: David Lewis and the future of formal methods* (Philosophy Synthese Library book series ed.). Dordrecht: Springer.

Anthony, D. (2005). Cooperation in microcredit borrowing groups: identity, sanctions, and reciprocity in the production of collective good. *American Sociological Review , 70*, pp. 496-515.

ASDQ. (1983). *The ASDQ glossary and tables for statistical quality control.*

Austin, J. L. (1976). *How to do things with words.* London: Oxford University Press.

Axelrod, R. (1984). *The evolution of cooperation.* New York: Basic Books.

Axelrod, R. (1986). An evolutionary approach to norms. *American Political Science Review , 80*, pp. 1095-1111.

Axelrod, R. (1987). *The evolution of strategies in the iterated prisoner's dilemma.* Los Altos, CA: Kaufmann.

Axelrod, R. (1995). A model of the emergence of new political actors. In N. Gilbert, & R. Conte, *Artificial societies: the computer simulation for social life.* London: UCL Press.

Axelrod, R., & Keohane, R. O. (1985). Achieving cooperation under anarchy: strategies and institutions. *World Politics , 38*, pp. 226-254.

Bandura, A. (1991). Social cognitive theory of moral thought and action. In W. M. Kurtines, & J. L. Gewirtz, *Handbook of moral behavior and development* (Vol. 1, pp. 45-103). Hillsdale, NJ: Lawrence Erlbaum.

Bargh, J. A. (1992). The ecology of automaticity: toward establishing the conditions needed to produce automatic processing effects. *The American Journal of Psychology , 105* (2), pp. 181-199.

Bargh, J. A., Gollwitzer, P. M., Lee-Chai, A., Barndollar, K., & Troetschel, R. (2001). The automated will: unconscious activation and pursuit of behavioral goals. *Journal of Personality and Social Psychology , 81*, pp. 1004-1027.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences , 22*, pp. 577-660.

Basu, K. (1998). *Social norms and the law*. (P. Newman, Editor) Retrieved from The New Palgrave Dictionary of Economics and Law: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=42840

Becker, B., & Mark, G. (1997). *Constructing social systems through computer mediated communication.* Sankt Augustin: German National Research Centre for Information Technology.

Berger, P. L., & Luckman, T. (1972). *The social construction of reality.* London: Penguin.

Bicchieri, C. (1990). Norms of cooperation. *Ethics , 100*, pp. 838-861.

Bicchieri, C. (2006). *The grammar of society: the nature and dynamics of social norms.* New York: Cambridge University Press.

Bicchieri, C., Duffy, J., & Tolle, G. (2003). Trust among strangers. *Philosophy of Science , 71*, pp. 286-319.

Bickel, W. K., & Johnson, M. W. (2003). Delay discounting: a fundamental behavioral process of drug dependence. In G. Loewenstein, D. Read, & R. F. Baumeister, *Time and decision.* New York: Russell Sage Foundation.

Bidwell, C. E. (1966). Values, norms, and the integration of complex social systems. *The Sociological Quarterly , 7* (2), pp. 119-136.

Binmore, K. (1994). *Game theory and the social contract* (Vol. 1: Playing Fair). Cambridge: MIT Press.

Binmore, K. (1998). *Review of the book: The complexity of cooperation: agent-based models of competition and collaboration, by Axelrod, R. Princeton, Princeton University Press*. Retrieved from Journal of Artificial Societies and Social Simulation 1: http://jasss.soc.surrey.ac.uk/1/1/review1.html

Biocca, F., Harms, C., & Burgoon, J. K. (2003). Toward a more robust theory and measure of social presence: review and suggested criteria. *Presence: Teleoperators & Virtual Environments , 12* (5), pp. 456-480.

Blair, J. (1995). A cognitive developmental approach to morality. *Cognition , 57*.

Blair, R., Jones, L., Clark, F., & Smith, M. (1997). The psychopathic individual: a lack of responsiveness to distress cues? *Psychophysiology , 34*.

Boas, F. (1911). *The mind of primitive man.* Retrieved from http://www.archive.org/details/mindofprimitivem031738mbp

Boella, G. (2001). Deliberate normative agents. Basic instructions. In *Social order in multiagent systems.* Norwell, MA: Kluwer.

Boella, G., & van der Torre, L. (2003). Norm governed multiagent systems: the delegation of control to autonomous agents. In *Proceedings of the IEEE/WIC IAT Conference* (pp. 10-27). IEEE Press.

Boella, G., & van der Torre, L. (2006). An architecture of a normative system: counts-as conditionals, obligations, and permissions. In *AAMAS* (pp. 229-231). ACM Press.

Boella, G., van der Torre, L., & Verhagen, H. (2006). Introduction to the special issue on normative multiagent systems. *Journal of Computational and Mathematical Organization Theory (CMOT) , 12* (2-3).

Boella, G., van der Torre, L., & Verhagen, H. (2007). Normative multi-agent systems. Dagstuhl.

Boman, M. (1999). Norms in artificial decision making. *Artificial Intelligence and Law , 7*, pp. 17-35.

Box, G. E., Hunter, W. G., & Hunter, J. S. (2005). *Statistics for experimenters: design, innovation, and discovery* (2nd ed.). Wiley.

Braendle, A. (2006). *Many cooks don't spoil the broth.* Retrieved from http://meta.wikimedia.org/wiki/Transwiki:Wikimania05/Paper-AB1

Braithwaite, R. (1955). *Theory of games as a tool for the moral philosopher.* Cambridge: Cambridge University Press.

Bratman, M. (1987). *Intentions, plans and practical reasoning.* Stanford: CSLI Publications.

Brenner, T. (2006). Agent learning representation: advice of modelling economic learning. In L. Tesfatsion, & K. J. Judd, *Handbook of computational economics* (Vol. 2, pp. 895-947). Elsevier.

Broad, C. D. (1925). *The mind and its place in nature.* London: Routledge & Kegan Paul.

Broersen, J., Dastani, M., & van der Torre, L. (2005). Beliefs, obligations, intentions, and desires as components in an agent architecture. *International Journal of Intelligent Systems , 20*.

Broersen, J., Dastani, M., Huang, Z., & van der Torre, L. (2001). The BOID architecture: conflicts between beliefs, obligations, intensions and desires. In *Proceedings of the Fifth International Conference on Autonomous Agents* (pp. 9-16). Montreal, Quebec, Canada.

Bryant, S. L., Forte, A., & Bruckman, A. (2005). Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work.* Sanibel Island, Florida , USA.

Burke, M. A., & Young, H. P. (forthcoming). Social norms. In A. Bisin, J. Benhabib, & M. Jackson, *The handbook of social economics.* Amsterdam: North-Holland.

Burke, M. A., Fournier, G., & Prasad, K. (2006). The emergence of local norms in networks. *Complexity , 11*, pp. 65-83.

Butler, N. A. (2001). Optimal and orthogonal latin hypercube designs for computer experiments. *Biometrika , 88* (3), pp. 847-857.

Campbell, D. T. (1974). 'Downward causation' in hierarchically organized biological systems. In F. J. Ayala, & T. Dobzhansky, *Studies in the philosophy of biology* (pp. 179-186). Macmillan Press.

Campennì, M. (2007). *The norm recogniser at work.* Presentation at AAAI'2007, Washington DC.

Campennì, M., Andrighetto, G., Cecconi, F., & Conte, R. (2009). Normal = normative? The role of intelligent agents in norm innovation. In *Mind & society.*

Castelfranchi, C. (1998a). Simulating with cognitive agents: the importance of cognitive emergence. In J. S. Sichman, R. Conte, & N. Gilbert, *Multi-agent systems and agent-based simulation.* Berlin: Springer.

Castelfranchi, C. (1998b). Through the minds of the agents. *Journal of Artificial Societies and Social Simulation , http://jasss.soc.surrey.ac.uk/1/1/5.html*.

Castelfranchi, C. (1999). Prescribed mental attitudes in goal-adoption and norm adoption. *Artificial Intelligence and Law , 7* (1), pp. 37-50.

Castelfranchi, C., & Conte, R. (1999). From conventions to prescriptions. Towards a unified theory of norms. *Artificial Intelligence and Law , 7* (4), pp. 323-340.

Castelfranchi, C., Conte, R., & Paolucci, M. (1998). Normative reputation and the costs of compliance. *Journal of Artificial Societies and Social Simulation , http://www.soc.surrey.ac.uk/JASSS/1/3/3.html*.

Castelfranchi, C., Dignum, F., & Treur, J. (2000). Deliberative normative agents: principles and architecture. In N. R. Jennings, & Y. Lesperance, *LNCS* (Vol. 1757, pp. 364-378). Berlin: Springer.

Castelfranchi, C., Miceli, M., & Cesta, A. (1992). Dependence relations among autonomous agents. In E. Werner, & Y. Demazeau, *Decentralized AI 3 - Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)* (pp. 215-231). Amsterdam, The Netherlands: Elsevier Science Publishers B.V.

Cialdini, R. B., Kallgren, C. A., & Reno, R. R. (1991). A focus theory of normative conduct: a theoretical refinement and re-evaluation of the role of norms in human behaviour. In L. Berkowitz, *Advances in experimental social psychology* (pp. 201-234). San Diego: Academic Press.

Ciffolilli, A. (2003). Phantom authority, self-selective recruitment and retention of members in virtual communities: the case of Wikipedia. *First Monday , 8* (12).

Cohen, P. R., & Levesque, H. J. (1990a). Intention is choice with commitment. *Artificial Intelligence , 42* (2-3), pp. 213-261.

Cohen, P. R., & Levesque, H. J. (1990b). Persistence, intention, and commitment. In P. R. Cohen, J. Morgan, & M. A. Pollack, *Intentions in communication* (pp. 33-71). Cambridge, MA: MIT Press.

Coleman, J. S. (1987). The emergence of norms in varying social structures. *Angewandte Sozialforschung , 14*, pp. 17-30.

Coleman, J. S. (1990). *Foundations of social theory.* Cambridge MA: Harvard University Press.

Colt. (2004). *The Colt package*. Retrieved from Open Source Libraries for High Performance Scientific and Technical Computing in Java: http://acs.lbl.gov/~hoschek/colt/

Conte, R. (1998). *L'obbedienza intelligente.* Bari: Laterza.

Conte, R. (2000). Memes through (social) minds. In R. Aunger, *Darwinizing culture: the status of memetics as science.* London: Oxford University Press.

Conte, R. (2009). Rational, goal-oriented agents. In R. A. Meyers, *Encyclopedia of complexity and system science* (pp. 7533-7548). Springer.

Conte, R., & Castelfranchi, C. (1995). *Cognitive and social action.* London: UCL Press.

Conte, R., & Castelfranchi, C. (1999). From conventions to prescriptions. Towards a unified theory of norms. *Artificial Intelligence and Law , 7*, pp. 323-340.

Conte, R., & Castelfranchi, C. (2006). The mental path of norms. *Ratio Juris , 19* (4), pp. 501-517.

Conte, R., & Dellarocas, C. (2001). Social order in info societies: an old challenge for innovation. In R. Conte, & C. Dellarocas, *Social order in multiagent systems* (pp. 1-16). Norwell: Kluwer.

Conte, R., & Dignum, F. (2001). From social monitoring to normative influence. *Journal of Artificial Societies and Social Simulation , http://www.soc.surrey.ac.uk/JASSS/4/2/7.html*.

Conte, R., & Paolucci, M. (2001). Intelligent social learning. *Journal of Artificial Societies and Social Simulation , http://www.soc.surrey.ac.uk/JASSS/4/1/3.html*.

Conte, R., & Paolucci, M. (2004). Responsibility for societies of agents. *Journal of Artificial Societies and Social Simulation , http://jasss.soc.surrey.ac.uk/7/4/3.html*.

Conte, R., & Pedone, R. (2001). Dynamics of status symbols and social complexity. *Social Science Computer Review , 19* (3), pp. 249-262.

Conte, R., Andrighetto, G., & Giardini, F. (2009). The mind as fitness landscape. An agent based approach to evolving institutions. In *Proceedings of the workshop "Do Institutions Evolve?", European University*

Institute. Firenze.

Conte, R., Andrighetto, G., Campennì, M., & Paolucci, M. (2007). Emergent and immergent effects in complex social systems. In *Proceedings of AAAI Symposium, Social and Organizational Aspects of Intelligence.* Washington DC.

Cosmides, L., & Tooby, J. (1992). Cognitive adaptations for social exchange. In J. Barkow, L. Cosmides, & J. Tooby, *The adapted mind.* New York: Oxford University Press.

Cosmides, L., & Tooby, J. (2008). Can a general deontic logic capture the facts of human moral reasoning? How the mind interprets social exchange rules and detects cheaters. In W. Sinnott-Armstrong, *Moral psychology* (pp. 53-119). Cambridge, MA: MIT Press.

Cummins, D. D. (1996). Evidence for deontic reasoning in 3- and 4-year olds. *Memory and Cognition , 24* (6), pp. 823-829.

Damasio, A. R. (1994). *Descartes' error: emotion, rationality and the human brain.* New York: Putnam.

Deguchi, H. (2001). *Mutual commitment, norm formation and indirect regulation of agent society.* Working paper No. 53, Kyoto University, Graduate School of economics.

Dennett, D. (1995). *Darwin's dangerous idea: evolution and the meanings of life.* London: Allen Lane.

Dignum, F. (1999). Autonomous agents with norms. *Artificial Intelligence and Law , 7* (1), pp. 69-79.

Dignum, F., Kinny, D., & Sonenberg, L. (2002). From desires, obligations and norms to goals. *Cognitive Science Quarterly , 2*.

Donath, J. S. (1998). Identity and deception in the virtual community. In P. Kollock, & M. Smith, *Communities in cyberspace.* London: Routledge.

Doran, J. (1998). Simulating collective misbelief. *Journal of Artificial Societies and Social Simulation , http://jasss.soc.surrey.ac.uk/1/1/3.html*.

Douglas, M. (1986). *How institutions think.* Syracuse, NY: Syracuse University Press.

Dunbar, R. (1997). *Grooming, gossip and the evolution of language.* Harvard University Press.

Durkheim, É. (1895/1982). *The rules of sociological method and selected texts in sociology and its methods.* London: MacMillan.

Durkheim, É. (1897/1951). *Suicide. A study in sociology.* (J. Spaulding, & G. Simpson, Trans.) New York: The Free Press of Glencoe.

Edwards, C. P. (1987). Culture and the construction of moral values: a comparative ethnography of moral encounters in two cultural settings. In J. Kagan, & S. Lamb, *The emergence of morality in young children.* The University of Chicago Press.

Ellis, G. F. (2006). On the nature of emergent reality. In P. Clayton, & P. Davies, *The re-emergence of emergence: the emergentist hypothesis from science to religion.* Oxford: Oxford University Press.

EMIL: Emergence in the Loop: Simulating the Two-Way Dynamics of Norm Innovation. (2008). *EMIL-S: the simulation platform.* Deliverable 3.3.

EMIL: Emergence in the Loop: Simulating the Two-Way Dynmaics of Norm Innovation. (2007). *Requirements that EMIL-S must meet.* Deliverable 3.1.

Engestrom, Y., Miettinen, R., & Punamaki, R.-L. (1999). *Perspectives on activity theory.* New York: Cambridge University Press.

Epstein, J. M. (2000). *Learning to be thoughtless: social norms and individual computation.* Working Papers, No. 6, Center on Social and Economic Dynamics.

Epstein, J. M. (2006). *Generative social science: studies in agent-based computational modeling.* Princeton:

Princeton University Press.

Epstein, J. M., & Axtell, R. (1996). *Growing artificial societies: social science from the bottom up.* Cambridge MA: MIT Press.

Esser, H. (2000). Normen als Frames: das Problem der 'Unbedingtheit' des normativen Handelns. In R. Metze, K. Mühler, & K.-D. Opp, *Normen und Institutionen: Entstehung und Wirkung* (pp. 137-155). Leipzig: Leipziger Universitätsverlag.

Feld, T. (2006). *Collective social dynamics and social norms.* Munich : Personal RePEc Archive.

Fiske, S. T., & Taylor, S. E. (1991). *Social cognition* (2nd ed.). New York: McGraw Hill.

Flentge, F., Polani, D., & Uthmann, T. (2001). Modelling the emergence of possession norms using memes. *Journal of Artificial Societies and Social Simulation , http://www.soc.surrey.ac.uk/JASSS/4/4/3.html*.

Forte, A., & Bruckman, A. (2005). *Why do people write for Wikipedia? Incentives to contribute to open-content publishing.* Georgia Institute of Technology, College of Computing.

Forte, A., & Bruckman, A. (2008). Scaling consensus: increasing decentralization in Wikipedia governance. In *Proceedings of the 41st Hawaiian International Conference of Systems Sciences.* Waikoloa Village, HI.

Franklin, S. (1998). *Artificial minds.* London: MIT press.

Fuchs, C., & Hofkirchner, W. (2005). The dialectic of bottom-up and top-down emergence in social systems. *tripleC , 1* (1).

Galan, M., & Izquierdo, L. (2005). Appearances can be deceiving: lessons learned re-implementing Axelrod's 'evolutionary approach to norms'. *Journal of Artificial Societies and Social Simulation , http://jasss.soc.surrey.ac.uk/8/3/2.html*.

Garcia-Camino, A., Rodriguez-Aguilar, J., Sierra, C., & Vasconcelos, W. (2006). Norm-oriented programming of electronic institutions: a rule-based approach. In *AAMAS 2006* (pp. 33-40). ACM Press.

Gibbs, J. P. (1965). Norms: the problem of definition and classification. *American Journal of Sociology , 60* (8).

Gibbs, J. P. (1981). *Norms, deviance and social control: conceptual matters.* New York: Elsevier.

Gilbert, M. (1981). Game theory and convention. *Synthese , 46* (1).

Gilbert, M. (1989). *On social facts.* London, New York: Routledge.

Gilbert, N. (1995). Emergence in social simulation. In N. Gilbert, & R. Conte, *Artificial societies. The computer simulation of social life.* London: UCL Press.

Gilbert, N. (2002). *Varieties of emergence.* Retrieved June 27, 2008, from Agent 2002. Social Agents Exology, Exchange, and Evolution: http://www.soc.surrey.ac.uk/staff/ngilbert/ngpub/paper148_NG.pdf

Gilbert, N. (2002). Varieties of emergence. *Paper presented at the Agent 2002 Conference: Social agents: ecology, exchange, and evolution.* Chicago.

Giles, J. (2005). *Internet encyclopaedias go head to head*. Retrieved from http://www.nature.com/news/2005/051212/full/438900a.html

Gintis, H. (2000). *Game theory evolving: a problem-centered introduction to modeling strategic behavior.* Princeton: Princeton University Press.

Gintis, H. (2003). Solving the puzzle of prosociality. *Rationality and Society , 15* (2), pp. 155-187.

Gintis, H. (2004). The genetic side of gene-culture coevolution: internalization of norms and prosocial emotions. *Journal of Economic Behavior & Organization , 53* (1), pp. 57-67.

Gintis, H., Bowles, S., Boyd, R., & Fehr, E. (2003). Explaining altruistic behavior in humans. *Evolution and Human Behavior , 24* (3), pp. 153-172.

Goffman, I. (1983). The interaction order: American Sociological Association 1982 presidential address. *American Sociological Review , 48* (1), pp. 1-17.

Goldspink, C. (2007). Normative self-regulation in the emergence of global network institutions: the case of Wikipedia. In *Australia and New Zealand Systems Conference 2007: Systemic Development: Local Solutions in a Global Environment (ANZSYS-07).*

Goldspink, C. (2008a). Social self regulation in on-line communities: the case of Wikipedia. *International Journal of Agent technologies and Systems , 1* (1), pp. 19-33.

Goldspink, C. (2008b). *Explaining normative behaviour in Wikipedia.* EMIL Deliverable 2.2. Report on norm innovation main features.

Goldspink, C., & Kay, R. (2004). Bridging the micro-macro divide: a new basis for social science. *Human Relations , 57* (5).

Goldspink, C., & Kay, R. (2007). Social emergence: distinguishing reflexive and nonreflexive modes. *AAAI Fall Symposium: Emergent Agents and Socialities: Social and Organizational Aspects of Intelligence.* Washington.

Goldspink, C., & Kay, R. (2008a). Agent cognitive capabilities and orders of emergence: critical thresholds relevant to the simulation of social behaviours. *AISB Convention, Communication, Interaction and Social Intelligence.* Aberdeen.

Goldspink, C., & Kay, R. (2008b). Agent cognitive capability and orders of emergence. In G. Trajkovski, & S. Collins, *Agent-based societies: social and cultural interactions.*

Goldspink, C., & Kay, R. (2009). Autopoiesis and organizations: a biological view of organizational change and methods for its study. In R. Magalhaes, & R. Sanchez, *Autopoiesis in organizations and information systems.* Elsevier Science.

Goodall, C. E. (2005). *Modifying smoking behavior through public service announcements and cigarette package warning labels: a comparison of Canada and the United States.* Senior Honors Thesis, Ohio State University.

Grusec, J. E., & Kuczynski, L. (Eds.). (1997). *Parenting and children's internalization of values: a handbook of contemporary theory.* New York: Wiley.

Gulyás, L. (2008a). Agent-based modeling and simulation with MASS. In *Summer School on Computational Social Sciences, National Chengchi University, August 2008.* Taipei, Taiwan.

Gulyás, L. (2008b). Agent-based modeling and simulation with the Multi-Agent Simulation Suite. In *Tutorial at the 5th Conference of the European Social Simulation Association (ESSA), Brescia, Italy, September 1, 2008.* Brescia.

Gulyás, L. (2008c). Agent-based modeling and simulation with the Multi-Agent Simulation Suite. In *Tutorial at the 22nd European Simulation and Modeling Conference, Le Havre, France, October 29, 2008.* Le Havre.

Gulyás, L. (2008d). Social simulation with agent-based modeling (with MASS). In *Complex Systems and Social Simulation, Summer University, Central European University, Budapest, July 2008.* Budapest.

Gulyás, L., Bartha, S., Kozsik, T., Szalai, R., Korompai, A., & Tatai, G. (2005). The Multi-Agent Simulation Suite (MASS) and the Functional Agent-Based Language of Simulation (FABLES). *SwarmFest 2005.*

Gulyás, L., de Back, W., Szemes, G., Kurowski, K., Dubitzky, W., & Kampis, G. (2008). Templates for distributed agent-based simulations on a quasi-opportunistic grid. In *Proceedings of the 20th European Modeling and Simulation Symposium (EMSS2008).* Campora San Giovanni, Italy.

Güth, W., & Kliemt, H. (1998). The indirect evolutionary approach: bridging the gap between rationality and adaption. *Rationality and Society , 10*, pp. 377-399.

Habermas, J. (1976). Some distinctions in universal pragmatics: a working paper. *Theory and Society , 3* (2), pp. 155-167.

Hales, D. (2002). Group reputation supports beneficent norms. *Journal of Artificial Societies and Social Simulation , http://jasss.soc.surrey.ac.uk/5/4/4.html*.

Hardin, R. (1982). Exchange theory on strategic bases. *Rationality and Society , 21*, pp. 251-272.

Hardin, R. (2007). *David Hume: moral and political theorist.* Oxford : Oxford University Press.

Hart, H. L. (1968). Prolegomenon to the principles of punishment (1959). In H. L. Hart, *Punishment and responsibility* (pp. 1-27). Oxford University Press.

Heckathorn, D. D. (1990). Collective sanctions and compliance norms: a formal theory of group-mediated social control. *American Sociological Review , 55* (3), pp. 366-384.

Hegselmann, R., & Krause, U. (2002). Opinion dynamics and bounded confidence. *Journal of Artifigial Societies and Social Simulation , 5* (3).

Hegselmann, R., & Will, O. (in press). Modelling Hume's moral and political theory: the design of HUME1.0. In M. Baurmann, G. Brennan, R. Goodin, & N. Southwood, *Norms and values. The role of social norms as instruments of value realisation.* Baden-Baden: Nomos.

Hemetsberger, A., & Pieters, R. (. (2001). When consumers produce on the internet: an inquiry into motivational sources of contribution to joint-innovation. *Paper presented at the Fourth International Research Seminar on Marketing Communications and Consumer Behaviour.* La Londe.

Hempel, C., & Oppenheim, P. (1948). Studies in the logic of explanation. *Philosophy of Science , 15*, pp. 135-175.

Henrich, J., Boyd, R., Bowles, S., Camerer, C., Fehr, E., & Gintis, H. (2001). *Foundations of human sociality.* Oxford University Press.

Hogg, M. A., & Abrams, D. (1988). *Social identifications: a social psychology of intergroup relations and group processes.* London: Routledge.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

Hopcroft, J., & Ullman, J. (1979). *Introduction to automata theory, languages and computation* (1st ed.). Addison-Wesley.

Horne, C. (2007). Explaining norm enforcement. *Rationality and Society , 19* (2), pp. 139-170.

Horty, J. F. (2001). *Agency and deontic logic.* Oxford: Oxford University Press.

Hume, D. (1998). *An enquiry concerning the principles of morals.* (T. Beauchamp, Ed.) Oxford: Oxford University Press.

Hume, D. (2007). *A treatise of human nature.* (D. Norton, & M. Norton, Eds.) Oxford: Oxford University Press.

Iványi, M., Bocsi, R., Gulyás, L., Kozma, V., & Legéndi, R. (2007). The Multi-Agent Simulation Suite. In *AAAI Fall Symposium Series, Washington DC, USA, November 8-11, 2007.* Washington DC.

Iványi, M., Gulyás, L., Bocsi, R., Szemes, G., & Mészáros, R. (2007). The Model Exploration Module. In *Agent 2007: Complex Interaction and Social Emergence Conference, Evanston IL, November 15-18, 2007.* Evanston.

Jones, A. J., & Sergot, M. J. (1993). On the characterisation of law and computer systems: the normative systems perspective. In J.-J. C. Meyer, & R. J. Wieringa, *Deontic logic in computer science: normative systems specification* (pp. 275-307). Chichester: John Wiley & Sons.

Jones, A. J., & Sergot, M. J. (1996). A formal characterization of institutionalized power. *Logic Journal of the IGPL , 4* (3), pp. 429-445.

Jones, N., Datta, A., & Jones, H. (2009). *Knowledge, policy and power: six dimensions of the knowledge-development policy interface.* Overseas Development Institute.

Kahneman, D., & Tversky, A. (1979). Prospect theory: an analysis of decision under risk. *Econometrica , 47* (2), pp. 263-291.

Kaldor, N. (1961/1968). Capital accumulation and economic growth. In F. A. Lutz, & D. C. Hague, *The theory of capital* (Reprint ed., pp. 177-222). London: Macmillan.

Kallgren, C. A., Reno, R. R., & Cialdini, R. B. (2000). A focus theory of normative conduct: when norms do and do not affect behaviour. *Personality and Social Psychology Bulletin , 26* (8), pp. 1002-1012.

Kauffman, S. A. (1993). *The origins of order: self organization and selection in evolution.* Oxford University Press.

Kelsen, H. (1991). *General theory of norms.* USA: Oxford University Press.

King, K. (2008). The politics of partnerships: peril or promise. *Special Issue of N O R R A G N E W S: Network for Policy Research Review and Advice on Education and Training (NORRAG) , 41*.

Kliemt, H. (1985). *Moralische Institutionen - Empiristische Theorien ihrer Evolution.* Freiburg: Alber.

Kliemt, H. (1986). *Antagonistische Kooperation.* Freiburg: Alber.

Koehler, J., & Owen, A. (1996). Computer experiments. In S. Ghosh, & C. Rao, *Handbook of statistics* (Vol. 13, pp. 261-308). Elsevier Science.

Kohlberg, L. (1981). Justice and reversibility. In L. Kohlberg, *Essays on moral development* (Vol. 1). Harper and Row.

Kohlberg, L. (1996). *Die Psychologie der Moralentwicklung.* Frankfurt a. M.: Surkamp.

Kohlberg, L., & Turiel, E. (1971). Moral development and moral education. In G. Lesser, *Psychology and educational practice.* Scott Foresman.

Lahno, B. (1995). *Versprechen - Überlegungen zu einer künstlichen Tugend.* München: Oldenbourg.

Lambrecht, M., Ivens, P.-L., Vandaele, N., & Müller, J. (1998, June). Active nonlinear tests (Ants) of complex simulation models. *Management Science , 44* (6), pp. 820-830.

Latane, B., & Darley, J. (1970). *The unresponsive bystander: Why doesn't he help?* New York: Appleton-Century-Crofts.

Lea, M., Spears, R., & de Groot, D. (2001). Knowing me, knowing you: anonymity effects on social identity processes within groups. *Personality and Social Psychology Bulletin , 27* (5), pp. 526-537.

Legéndi, R., Bocsi, R., Iványi, M., & Gulyás, L. (2007). Modeling with FABLES. In *Agent 2007: Complex Interaction and Social Emergence Conference, Evanston IL, November 15-18, 2007.* Evanston.

Levine, J. M., & Valle, R. (1975). The convert as a credible communicator. *Social Behavior and Personality , 3* (1), pp. 81-90.

Lewis, D. K. (1969). *Convention: a philosophical study.* Cambridge, MA: Harvard University Press.

Lih, A. (2004). Wikipedia as participatory journalism: reliable sources? Metrics for evaluating collaborative media as a news resource. *5th International Symposium on Online Journalism.* Austin: University of Texas.

Lindhal, L. (1977). *Position and change: a study in law and logic.* Springer.

López y López, F., & Márquez, A. A. (2004). An architecture for autonomous normative agents. In *5th Mexican International Conference on Computer Science* (pp. 96-103). Los Alamitos, CA, USA: IEEE Computer Society.

López y López, F., Luck, M., & d'Inverno, M. (2002). Constraining autonomy through norms. In *Proceedings* *of AAMAS '02.*

Lorscheid, I., & Troitzsch, K. G. (2009). How do agents learn to behave normatively? Machine learning concepts for norm learning in the EMIL project. In *Proceedings of the 6th Annual Conference of the European Social Simulation Association.* Guildford, UK.

Lotzmann, U. (2008). TRASS - a multi-purpose agent-based simulation framework for complex traffic simulation applications. In A. Bazzan, & F. Klügl, *Multi-agent systems for traffic and transportation.* IGI Global.

Lotzmann, U., & Möhring, M. (2008). A TRASS-based agent model for traffic simulation. In *Proceedings of the 22th European Conference on Modelling and Simulation (EMCS).* Nicosia, Cyprus.

Lotzmann, U., & Möhring, M. (2009). Simulating normative behaviour and norm formation processes. In *23rd European Conference on Modelling and Simulation, Madrid, July 2009.* Madrid.

Lotzmann, U., Möhring, M., & Troitzsch, K. G. (2008). Simulating norm formation in a traffic scenario. In *Fifth Conference of the European Social Simulation Association (ESSA), Brescia September 1-5, 2008.* Brescia.

Lucas dos Anjos, P. (2009a). *Relating financial characterisation of microfinance groups and their conventional social behaviour.* Second CFPM – ETH – UNAM fieldwork report, Manchester, England.

Lucas dos Anjos, P. (2009b). *Final data compilation.* CFPM – ETH – UNAM fieldwork report, Manchester, England.

Lucas dos Anjos, P. (2009c). *Influencing microfinance policy with fieldwork findings.* Working paper, United Nations University, World Institute for Development Economics Research, Helsinki, Finland.

Lucas dos Anjos, P. (2009d). Usefulness of simulating social phenomena: evidence. *Artificial Intelligence and Society Journal, Special Issue on the social understanding of Artificial Intelligence* .

Lucas dos Anjos, P. (to be submitted). Modelling conventional microfinance social behaviour: clients and advisors. *The Journal of Artificial Societies and Social Simulation* .

Lucas dos Anjos, P., Morales, F., & Garcia, I. (2008a). Towards analysing social norms in microfinance groups. In *8th International Conference of the International Society for Third Sector Research (ISTR).* Barcelona.

Lucas dos Anjos, P., Morales, F., & Garcia, I. (2008b). *Social conventions and dynamic of solidarity groups - an experience in Chiapas.* CFPM – ETH – UNAM fieldwork report, Manchester, England.

Luke, S., Balan, G., Panait, L., Cioffi-Revilla, C., & Paus, S. (2003). MASON. A Java multi-agent simulation library. In *Proceedings of the Agent 2003 Conference.*

Maas, C. J., & Hox, J. J. (2005). Sufficient sample sizes for multilevel modeling. *Methodology , 1* (3), pp. 86-92.

Macy, M. W., & Sato, Y. (2002). Trust, cooperation, and market formation in the U.S. and Japan. *PNAS , 99*, pp. 7214-7220.

Macy, M. W., & Skvoretz, J. (1998). The evolution of trust and cooperation between strangers. *American Sociological Review , 63*, pp. 638-660.

Malsch, T., Schlieder, C., Kiefer, P., Lübcke, M., Perschke, R., Schmitt, M., et al. (2007). Communication between process and structure. *Journal of Artificial Societis and Social Simulation , 10* (1).

Markus, H., & Zajonc, R. B. (1985). The cognitive perspective in social psychology. In G. Lindzey, & E. Aronson, *Handbook of social psychology* (3rd ed., pp. 137-229). New York: Random House.

Mauss, M. (1922/1990). *The gift: forms and functions of exchange in archaic societies.* London: Routledge.

McAdams, R. H. (2008). *Norm internalization: a comment on Philip Pettit.* Draft of 3 December 2008.

Mead, M. (1963). *Cultural patterns and technical change.* New York: The New American Library.

Meyer, J.-J. (1988). A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic , 29* (1), pp. 109-136.

Micro-finance Inc. (2009). *Antecedents and client profiling.* Monthly Operation Report, Grameen Foundation.

Microfinance simulation model: HowTo guide. (2009). Retrieved June 7th, 2009, from cfpm.org

Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the structure of behavior.* New York: Holt, Rinehart & Winston.

Misztal, B. (2001). Normality and trust in Goffman's theory of interaction order. *Sociological Theory , 19* (3), pp. 312-324.

Mitchell, T. (1997). *Machine learning.* McGraw-Hill.

Murphy, G. L. (2002). *The big book of concepts.* MIT Press.

Myerson, R. B. (1991). *Game theory: analysis of conflict.* Harvard University Press.

Neumann, M. (2008). A classification of normative architectures. In *Proceedings of the 2nd WCSS.* Fairfax, VA.

Nickerson, R. S. (1998). Confirmation bias: a ubiquitous phenomenon in many guises. *Review of General Psychology , 2*, pp. 175-220.

NIST/SEMATECH. (2008). *e-Handbook of statistical methods*. Retrieved from http://www.itl.nist.gov/div898/handbook/pri.htm

Norris, G. A., & Jager, W. (2004). Household-level modeling for sustainable consumption. In *Third International Workshop on Sustainable Consumption.* Tokyo.

North, M., Collier, N., & Vos, J. (2006). Experiences creating three implementations of the Repast agent modeling toolkit. *ACM Tranbsactions on Modeling and Computer Simulation , 16* (1), pp. 1-25.

North, M., Tatara, E., Collier, N., & Ozik, J. (2007). Visual agent-based model development with Repast Simphony. In *Proceedings of the Agent 2007 Conference on Complex Interaction and Social Emergence, Argonne National Laboratory, Novemver 2007.* Argonne, IL, USA.

Nucci, L. P. (2001). *Education in the moral domain.* Cambridge University Press.

Oliver, P. E. (1993). Formal models of collective action. *Annual Review of Sociology , 19*, pp. 271-300.

Olson, M. (1965). *The logic of collective action: public goods and the theory of groups.* Cambridge, MA: Harvard University Press.

Opp, K. D. (2001). How do norms emerge? An outline of a theory. *Mind and Society , 2*, pp. 101-128.

Opp, K. D., & Hechter, M. (Eds.). (2001). *Social norms.* New York: Sage Publications.

Parsons, T. (1937/1968). *The structure of social action. A study in social theory with special reference to a group of recent European writers.* New York, London: Free Press.

Parsons, T. (1967). *Sociological theory and modern society.* New York: Free Press.

Penner, L., Dovidio, J. F., Piliavin, J. A., & Schroder, D. A. (2005). Pro-social behaviour: multilevel perspective. *Annual Review of Psychology , 56*, pp. 365-392.

Pfeil, U., Zaphiris, P., & Ang, C. S. (2006). Cultural differences in collaborative authoring of Wikipedia. *Journal of Computer Mediated Communication , 12*, pp. 88-113.

Piaget, J. (1965). *The moral judgment of the child.* New York: The Free Press.

Piaget, J. (1978). *Success and understanding.* Routledge and Kegan Paul.

Plackett, R., & Burman, J. (1946, January). The design of optimum multifactorial experiments. *Biometrika , 33* (4), pp. 305-325.

Posner, R., & Rasmusen, E. (1999). Creating and enforcing norms, with special reference to sanctions. *International Review of Law and Economics* , pp. 369-382.

Postmes, T., Spears, R., Sakhel, K., & de Groot, D. (2001). Social influence in computer-mediated communication: the effects of anonymity on group behavior. *Personality and Social Psychology Bulletin , 27* (10), pp. 1243-1254.

Potter, J. (2002). Wittgenstein and Austin. In M. Wetherell, S. Taylor, & S. J. Yates, *Discourse theory and practice.* London: Sage.

Priedhorsky, R., Chen, J., Lam, S., Panciera, K., Terveen, L., & Riedl, J. (2007). Creating, destroying, and restoring value in Wikipedia. In *Proceedings of the 2007 international ACM conference on Supporting group work* (pp. 259-268). Sanibel Island, Florida, USA.

Rachlin, H. (2000). *The science of self-control.* Cambridge, London: Harvard University Press.

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI architecture. In *Proceedings of the KR91* (pp. 473-484).

Rao, A. S., & Georgeff, M. P. (1992). Social plans: preliminary report. In E. Werner, & Y. Demazeau, *Decentralized AI 3 - Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)* (pp. 57-76). Amsterdam, The Netherlands: Elsevier Science Publishers B.V.

Raz, J. (1975). *Practical reason and norms.* Oxford: Oxford University.

Reagle, J. (2004). *A case of mutual aid: Wikipedia, politeness, and perspective taking*. Retrieved August 18, 2008, from http://reagle.org/joseph/2004/agree/wikip-agree.html

Repast Development Team. (2009). *Repast: Recursive Porous Agent Simulation Toolit*. Retrieved January 20, 2009, from http://repast.sourceforge.net/

Riesman, D. (1950). *The lonely crowd: a study of the changing American character.* New Haven: Yale University Press.

Riva, G., & Galimberti, C. (1998). Computer mediated communication: identity and social interaction in an electronic environment. *Genetic, Social and General Psychology Monographs , 124*, pp. 434-464.

Rosenthal, R., & Jacobson, L. (1992). *Pygmalion in the classroom* (Expanded ed.). New York: Irvington.

Rullani, F. (2005). *The debate and the community: the 'reflexive identity' concept and the FLOSS community case.* Working Paper No. 2005-18, Sant'Anna School of Advanced Studies, Laboratory of Economics & Management, Pisa.

Russel, R., & Norvig, P. (2003). *Artificial intelligence. A modern approach.* Prentice Hall.

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist , 55* (1), pp. 68-78.

Saam, N. J., & Harrer, A. (1999). Simulating norms, social inequality, and functional change in artificial societies. *Journal of Artificial Societies and Social Simulation , http://www.soc.surrey.ac.uk/JASSS/2/1/2.html*.

Sadri, F., Stathis, K., & Toni, F. (2006). Normative KGP agents. *Computational and Mathematical Organization Theory , 12*, pp. 101-126.

Sanger, L. (2005). *The early history of Nupedia and Wikipedia: a memoir.* Retrieved from http://features.slashdot.org/article.pl?sid=05/04/18/164213&tid=95&tid=149&tid=9

Santner, T., Williams, B., & Notz, W. (2003). *The design and analysis of computer experiments.* New York:

Springer.

Savarimuthu, B. T., Cranefield, S., Purvis, M. K., & Purvis, M. A. (2007). Norm emergence in agent societies formed by dynamically changing networks. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (pp. 464-470).

Savarimuthu, B. T., Purvis, M. A., Cranefield, S., & Purvis, M. K. (2007a). How do norms emerge in multi-agent societies? Mechanism design. *The Information Science Discussion Paper Series* .

Savarimuthu, B. T., Purvis, M. A., Cranefield, S., & Purvis, M. K. (2007b). Mechanisms for norm emergence in multiagent societies. In *Proceedings of 6th Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS).* Honolulu, Hawaii, USA.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: an inquiry into human knowledge structures.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Scott, J. F. (1971). *Internalization of norms: a sociological theory of moral commitment.* Englewoods Cliffs, NJ: Prentice-Hall.

Searle, J. R. (1969). *Speech act: an essay in the philosophy of language.* Cambridge: Cambridge University Press.

Searle, J. R. (1990). Collective intentions and actions. In P. Cohen, J. Morgan, & M. Pollack, *Intentions in communication* (pp. 401-415). Cambridge, MA: The MIT Press.

Searle, J. R. (2002). Speech acts, mind, and social reality. In G. Grewendorf, & G. Meggle, *Speech acts, mind, and social reality: discussions with John R. Searle.* Dordrecht: Kluwer Academic Publishers.

Segerberg, K., Meyer, J. J., & M., K. (2009). *The logic of action.* Retrieved from Stanford Enciclopedia of Philsophy: http://plato.stanford.edu/entries/logic-action/

Sen, S., & Airiau, S. (2007). Emergence of norms through social learning. *IJCAI-07*, (pp. 1507-1512).

Sherif, M. (1936). *The psychology of social norms.* New York: Harper & Raw Publishers.

Shoham, Y., & Tenneholtz, M. (1992). On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proceedings of the 10th AAAI Conference* (pp. 276-281).

Shoham, Y., & Tennenholtz, M. (1994). *Co-learning and the evolution of social activity.* Technical Report STAN-CS-TR-94-1511, Stanford University.

Shweder, R., Mahapatra, M., & Miller, J. (1987). Culture and moral development. In J. Kagan, & S. Lamb, *The emergence of morality in young children.* The University of Chicago Press.

Sichman, J. S., & Conte, R. (2002). Multi-agent dependence by dependence graphs. In *Proceedings of Autonomous Agent & MAS, AAMAS 2002, Part I* (pp. 483-491). ACM Press.

Sichman, J. S., Conte, R., Castelfranchi, C., & and Demazeau, Y. (1994). A social reasoning mechanism based on dependence networks. In A. G. Cohn, *Proceedings of the 11th European Conference on Artificial Iintelligence, ECAI* (pp. 188-192). Baffin Lane, England: John Wiley and Sons.

Skyrms, B. (1996). *Evolution of the social contract.* Cambridge: Cambridge University Press.

Skyrms, B. (2004). *The stage hunt and the evolution of social structure.* Cambridge: Cambridge University Press.

Smith, E., & Medin, D. (1981). *Categories and concepts.* Harvard University Press.

Smith, J. R., Terry, D. J., & Hogg, M. A. (2007). Social identity and the attitude behaviour relationship: effects of anonymity and accountability. *European Journal of Social Psychology , 37*.

Smith, M. A., & Kollock, P. (Eds.). (1999). *Communities in cyberspace.* London: Routledge.

Solow, R. M. (1970). *Growth theory: an exposition.* New York: Oxford University Press.

Sripada, C., & Stich, S. (2006). A framework for the psychology of norms. In P. Carruthers, S. Laurence, & S. Stich, *The innate mind: culture and cognition* (pp. 280-301). Oxford University Press.

Staller, A., & Petta, P. (2001). Introducing emotions into the computational study of social norms: a first evaluation. *Journal of Artificial Societies and Social Simulation , http://www.soc.surrey.ac.uk/JASSS/4/1/2.html*.

Sternberg, R. J. (2007). Critical thinking in psychology: It really is critical. In R. J. Sternberg, H. L. Roediger, & D. F. Halpern, *Critical thinking in psychology.* Cambridge University Press.

Stiles, W. B. (1992). *Describing talk: a taxonomy of verbal response modes.* London: Sage.

Sugden, R. (1986/2004). *The economics of rights, co-operation, and welfare* (2nd ed.). New York: Palgrave Macmillan.

Sun, Y., & Wu, B. (2006). Agent hybrid architecture and its decision processes. *International Conference on Machine Learning and Cybernetics, 13-16 Aug. 2006*, (pp. 641-644).

Surowiecki, J. (2004). *The wisdom of crowds: why the many are smarter than the few.* London: Abacus.

Tarp, F. (2009). Aid and Growth. *Presentation given at the World Institute for Development Economics Research (UNU-WIDER) and Helsinki Center of Economic Research (HECER) Autumn Seminar on Aid and Growth: Implications and Prospects for Developing Nations.* United Nations University.

Terry, D. J., & Hogg, M. A. (1996). Group norms and the attitude-behaviour relationship: a role for group identification. *Personality and Social Psychology Bulletin , 22*, pp. 776-793.

Terry, D. J., Hogg, M. A., & White, K. M. (1999). The theory of planned behaviour: self-identity, social identity and group norms. *British Journal of Social Psychology , 38*, pp. 225-244.

The QosCosGrid Project. (n.d.). Retrieved from Quasi-Opportunistic Supercomputing for Complex Systems Simulations on the Grid, FP6 Progranm Project IST-033883: http://www.qoscosgrid.org

Thomas, G., & James, D. (2006). Reinventing grounded theory: some questions about theory, ground and discovery. *British Educational Research Journal , 32* (6), pp. 767-795.

Thompson, E. (2001). Empathy and consciousness. *Journal of Consciousness Studies , 8* (5-7), pp. 1-32.

Tilburg University. (n.d.). *Space-filling designs*. Retrieved from http://www.spacefillingdesigns.nl

Troitzsch, K. G. (2008). Simulating collaborative writing: Software agents produce a Wikipedia. In *Fifth Conference of the European Social Simulation Association (ESSA), Brescia September 1-5, 2008.* Brescia.

Troitzsch, K. G. (2009). Multiagent systems and simulation: a survey from an application perspective. In A. M. Uhrmacher, & D. Weyns, *Agents, simulation and applications* (pp. 57-80). Milton Park: Taylor & Francis.

Turner, J. C. (1991). *Social influence.* Milton Keynes: Open University Press.

Ullman-Margalit, E. (1977). *The emergence of norms.* Oxford: Oxford University Press.

van der Torre, L., & Tan, Y. (1999). Contrary-to-duty reasoning with preference-based dyadic obligations. *Annals of Mathematics and Artificial Intelligence , 27*, pp. 49-78.

Vazquez-Salceda, J., Aldewereld, H., & Dignum, F. (2005). Norms in multiagent systems: from theory to practice. *International Journal of Computer Systems Science and Engineering , 20* (4), pp. 225-236.

Verhagen, H. (2001). Simulation of the learning of norms. *Social Science Computer Review , 19*, pp. 296-306.

Vieth, M. (2003). Die Evolution von Fairnessnormen im Ultimatumspiel: eine spieltheoretische Modellierung. *Zeitschrift für Soziologie , 32* (4), pp. 346-367.

von Wright, G. H. (1963). *Norm and action. A logical inquiry.* London: Routledge and Kegan Paul.

Vygotsky, L. S. (1978). *Mind in society: the development of higher psychological processes* (Published originally in Russian in 1930 ed.). Cambridge, MA: Harvard University Press.

Wason, P., & Johnson-Laird, P. (1972). *Psychology of reasoning: structure and content.* Cambridge, MA: Harvard University Press.

Weidlich, W., & Haag, G. (1983). Concepts and models of a quantitative sociology: the dynamics of interacting populations. In *Series in Synergetics, 14.* Springer.

Wilensky, U. (1999). *NetLogo*. (Northwestern University, Evanston IL, USA) Retrieved from Center for Connected Learning and Computer-Based Modeling: http://ccl.northwestern.edu/netlogo/

Wilkinson, D. M., & Huberman, B. A. (2007). *Assessing the value of cooperation in Wikipedia.* Palo Alto: HP Labs.

Will, O. (2009a). HUME1.0 - an agent-based model on the evolution of trust in strangers and division of labour. In *Proceedings of MABS 2009, 10th International Workshop on Multi-Agent-Based Simulation, May 11-12.* Budapest, Hungary.

Will, O. (2009b). Resolving a replication that failed: news on the Macy & Sato model. *Journal of Artificial Societies and Social Simulation , http://jasss.soc.surrey.ac.uk/12/4/11.html*.

Will, O., & Hegselmann, R. (2008). A replication that failed on the computational model in 'Michael W. Macy and Yoshimichi Sato: Trust, cooperation and market formation in the U.S. and Japan. Proceedings of the National Academy of Sciences, May 2002'. *Journal of Artificial Societies and Social Simulation , http://jasss.soc.surrey.ac.uk/11/3/3.html*.

Young, H. P. (1993). The evolution of conventions. *Econometrica , 61*, pp. 57-84.

Young, H. P. (1998). *Individual strategy and social structure: an evolutionary theory of institutions.* Princeton University Press.

Young, H. P. (2003). The power of norms. In P. Hammerstein, *Genetic and cultural evolution of cooperation* (pp. 389-399). Boston, MA: MIT Press.

Young, H. P. (2006). Social norms. In S. N. Durlauf, & L. E. Blume, *The new Palgrave dictionary of economics* (Second ed.). London: Macmillan.

Young, J., & Mendizabal, E. (2009). *Helping researchers become policy entrepreneurs.* Briefing Paper 53, Overseas Development Institute, London.

Zeigler, B. P. (1976). *Theory of modelling and simulation.* New York, London, Sydney, Toronto: John Wiley and Sons.

Zhang, H., & Huang, S. Y. (2006). Dynamic control of intention priorities of human-like agents. In G. Brewka, S. Coradeschi, A. Perini, & P. Traverso, *Proceedings of ECAI 06, The European Conference on Artificial Intelligence* (pp. 310-314). IOS Press.

Ziegler, R. (2000). Hat der Homo Oeconomicus ein Gewissen? In R. Metze, K. Mühler, & K.-D. Opp, *Normen und Institutionen: Entstehung und Wirkung* (pp. 65-92). Leipzig: Leipziger Universitätsverlag.