ENHANCING AGENTS WITH NORMATIVE CAPABILITIES

Ulf Lotzmann
Institute of Information Systems Research
University of Koblenz
Universitätsstraße 1, Koblenz 56070, Germany
E-mail: ulf@uni-koblenz.de

KEYWORDS
Norm innovation, Social simulation, Normative agents, Emergence, Immergence.

ABSTRACT

This paper describes the derivation of a software architecture (and its implementation called EMIL-S) from a logical normative agent architecture (called EMIL-A). After a short introduction into the theoretical background of agent-based normative social simulation, the paper focuses on intra-agent structures and processes. The pivotal element in this regard is a rule-based agent design with a corresponding "generalised intra-agent process" that involves decision making and learning capabilities. The resulting simulation dynamics are illustrated afterwards by means of an application sample where agents contribute to a Wikipedia community by writing, editing and discussing articles. Findings and material presented in the paper are part of the results achieved in the FP6 project EMIL (EMergence In the Loop: Simulating the two-way dynamics of norm innovation).

INTRODUCTION

This paper gives an introduction into several major outcomes of the FP6 project EMIL (EMergence In the Loop: Simulating the two-way dynamics of norm innovation) as an example of simulating emergent properties in complex social systems, funded by the EU initiative "Simulating Emergent properties in Complex Systems" (no. 033841). After an overview on the logical normative architecture EMIL-A (which is based on a scientific theory of norm innovation (Andrighetto et al. 2007)), with EMIL-S a software component dedicated to introduce normative capabilities in existing or newly designed multi agent systems is presented.

The EMIL project especially focuses on understanding and analysing norm innovation processes in social systems, which can be seen here as a special case of complex systems, composed of many different interacting intelligent autonomous agents. In general, including norms in multi-agent models seems to be a promising concept to understand human (and artificial) agent cooperation and co-ordination. Therefore, the design of agents with normative behaviour (i.e. normative agents) is of increasing interest in the multi-agent systems research (Boella et al. 2007).

Because of the fact that norms can be seen as a societal regulation of individual behaviour without sheer pressure, special attention in modelling and analysing norm innovation processes should be given not only to the inter-agent behaviour but also to the internal (mental) states and processes of the modelled agents (intra-agent) (Neumann 2008). Following this, the dynamics of norm innovation can be described mainly by two processes:

- immergence: intra-agent process by means of which a normative belief is formed into the agents' minds (Andrighetto et al. 2008). If this happens often enough, the resulting behaviour becomes a "sociological phenomenon" (Durkheim 1895/1982):
- emergence: inter-agent process by means of which a norm not deliberately issued spreads through a society.

These two processes are the basis for the logical architecture. Together with the process of its transformation into a software architecture it will be sketched in the following section. In the subsequent section software architecture is presented in more detail, followed by an application example.

FROM LOGICAL TO SOFTWARE ARCHITECTURE

To derive a software architecture from the logical and cognitive architecture EMIL-A introduced in (EMIL 2009), several steps are necessary. First of all, it is inevitable to formalize and implement several sequential core procedures:

- "norm recognition", i. e. the discrimination between norms and other social phenomena,
- "norm adoption", i. e. the generation of normative goals,
- "decision making", i. e. checking "against potential obstacles to the goal's pursuit" (EMIL 2009), and
- "normative action planning".

Related to the procedures, several data structures for the individual agents have to be defined for representation of:

- "normative beliefs",
- "normative goals" and
- "normative intentions".

Additionally, a "normative board" as a central inventory of norms is necessary as a shared data storage.

Proceedings 24th European Conference on Modelling and Simulation ©ECMS Andrzej Bargiela, Sayed Azam Ali David Crowley, Eugène J.H. Kerckhoffs (Editors)
ISBN: 978-0-9564944-0-5 / ISBN: 978-0-9564944-1-2 (CD)

Furthermore 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 (EMIL 2009) 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.

The process going on in what one could call a primordial artificial society can be illustrated by an everyday example: *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 (but it has a similar consequence): 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 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". 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", (EMIL 2009)) 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 (EMIL 2009)) – 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 a normative belief is generated 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 to the normative goal derived from the adopted norm, it is the turn of the 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.

The EMIL-S architecture uses very similar concepts to the ones specified in EMIL-A. The rule concept of EMIL-S is somewhat more complex as rules are also responsible for action planning, not only normative action planning. Scenarios run under EMIL-S must also reflect non-normative behaviour of agents.

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. Therefore, EMIL-S provides means to include an interface between agents and their environment. Although EMIL-S restricts itself to model the mind of human actors whereas modelling the body of human actors is the task of a simulation system below (Repast (North et al. 2006), TRASS (Lotzmann, 2009) etc.), agent design in EMIL-S has to include modelling that goes beyond modelling the normative processes.

ARCHITECTURE OF EMIL-S

Any multi-agent simulation system will have to be able to simulate the processes which go on within agents (recognition, memory, decision making), among agents (communication) and between agents and their environment (action). Mental processes within agents are thus separated from actions that agents take with respect to other agents (including communication) and their environment (here we have the usual restricted meaning of environment which does not include other agents proper; in this meaning the environment provides means for communication and other resources). Thus one of the central requirements for this kind of simulation is that agents do not communicate by mind-reading but by messages which have to be interpreted by the recipients of messages before they can have any effect on the recipient agent's behaviour.

The strict separation between mental processes and actions, or to put it in another way, between the "mind" of an agent and its "body", allows also to define a simulation tool which can be used for a very wide variety of simulations of human behaviour – as it incorporates some relevant features of mental processes – without trying to be appropriate for simulating a wide variety of settings in which humans would act. This means that the simulation tool created for designing mental processes – mainly decision making processes –

can be rather general and can be reused for a wide variety of situations, no matter what the concrete actions and their consequences for other agents and the environment are. Decision making does not only concern observable actions, but also internal actions, such as one to form or not to form a given mental state.
What agents must be able to do depends on the scenario one wants to simulate but how agents decide which actions they take can be modelled independently of the concrete scenarios.
Consequently, the general structure of the simulation system mainly consists of the module EMIL-S, in which agents with norm formation capabilities and normative behaviour can be defined ("mind"), and the Simulation Tool module, which contains the physical world of a concrete scenario ("body"). On basis of a common interface specification between these modules, different simulation scenarios – realized by using different simulation tools – can be enriched with normative agents.
The EMIL-S module is the core of the simulation system which represents the "minds" of normative agents and realises and implements the logical architecture of EMIL-A. It also provides a user interface (Agent Designer, shown in Figure 1), which allow modellers to design the mental processes of agents which they believe to be relevant in his or her scenario. More precisely, each agent must be equipped with a set of initial rules, which allows him to act in the simulation environment. Rules in EMIL-S are represented as so-called event-action trees, which is a kind of decision tree that represents the dependencies between events and actions. For each event an arbitrary number of action groups are defined (in the figure G1, GNI1-A and GNI1-O). Each action group represents a number of mutually exclusive actions (A10 to A12 or ANI10 and ANI11, respectively). The edges of the tree are attached to selection probabilities for the respective action groups or actions.

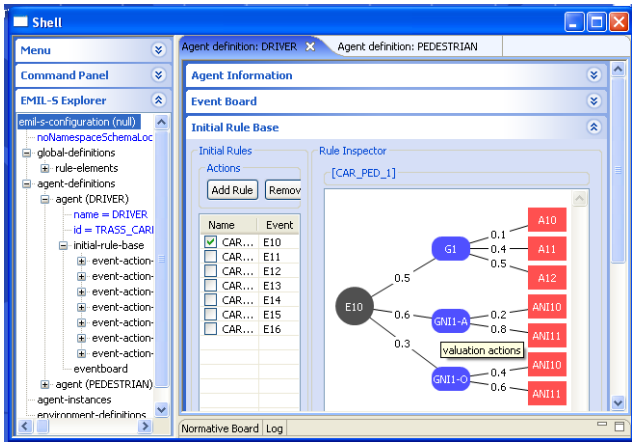

Figure 1: EMIL-S Agent Designer, displaying an event-action tree from the initial rule base

Different scenarios can recommend different simulation tools for the physical layer. In general different simulation tools play the role of the "bodies" of the agents and allow them to act on each other and their environment (which, of course, is also represented in the simulation tool). In addition to the definition of normative agents equipped with an initial rule base in the EMIL-S module, the completion of an executable simulation scenario requires the development of an interface (by using available templates) between EMIL-S and the selected simulation tool. Basically, this interface realises a link between the normative agent parts and their "physical" counterparts in the simulation tool (Figure 2).

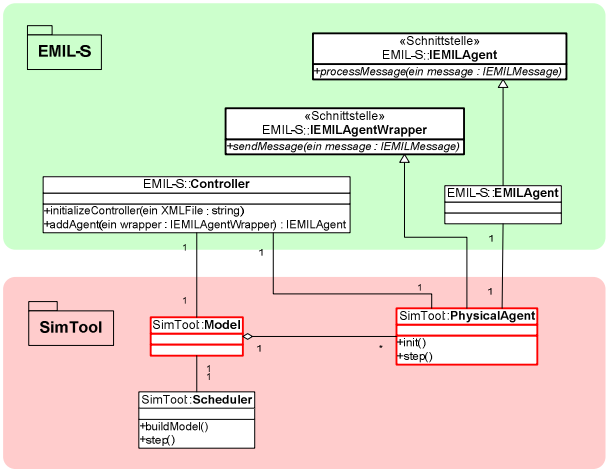

Figure 2: Components of the simulation system: example of an interface between EMIL-S and Repast

EMIL-S AGENT DESIGN

Beyond the requirements derived from the logical architecture, other – in the sense of computer science more technical – aspects 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.
Thus, EMIL-S also draws from achievements of several disciplines, in particular from:

- 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).

It seems to be generally accepted 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 3, 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.

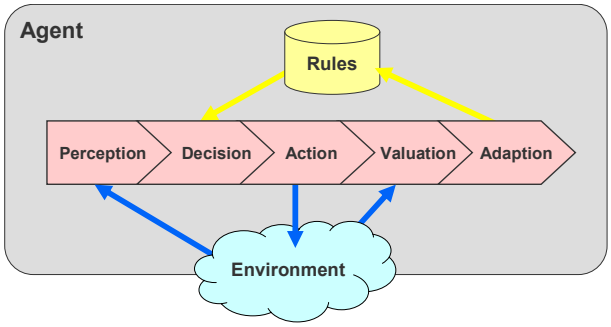

Figure 3: Generalized intra-agent process

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 reasonable 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 must be distinguished:

- 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 ensure the compatibility to 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 cycle of a simulation run.

Figure 4 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).

During the following adaption process, triggered by a norm invocation, the first normative frame entry is created (Figure 5).

USE CASE: WIKIPEDIA

In this subsection a simulation model which makes use of EMIL-S in combination with the Repast simulation toolkit will be presented. In this use case, 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 (Troitzsch 2008). 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 are endowed with the capability to produce text in this language and to evaluate something like the "style of writing", thus being able to take offence at certain features of texts and blaming the authors of such text. From this kind of

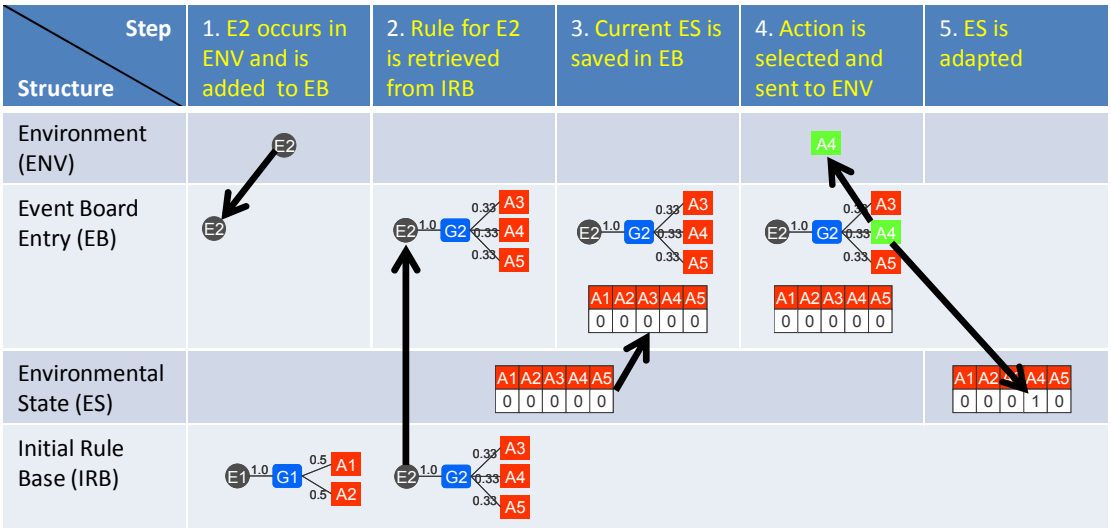

Figure 4: Intra-agent decision process for time t_0

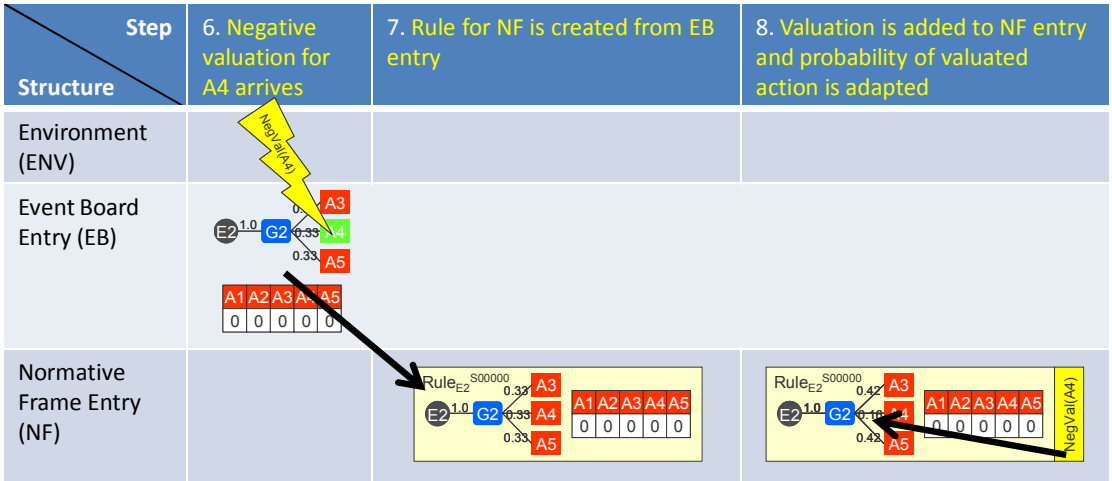

Figure 5: Intra-agent adaption process for time t_0

communication, norms emerge in the artificial society of software agents.

At the beginning of a simulation run agents have a repertoire of messages (or, more precisely, an algorithm that enables them to formulate new messages) that they want to add to the Wikipedia, and at the beginning all agents are authorised to do so. After some time the Wikipedia contains a list of articles which are not yet connected. These articles contain, among others, the name of the author and the content in terms of a string of arbitrary length, the first few elements of which have the special meaning of a keyword. Words consist of alternating vowels and consonants, forming a very primitive language which conveys no meaning, but agents can mull over similarities and differences among Wikipedia articles.

Besides writing, reading and editing articles, agents scan the articles already present for several criteria and comment on them (in the sense of EMIL-S valuations). These commenting actions are of the following types:

- vowel harmony: obeying or violating a special word building rule is assessed, see below;
- duplicate keywords: if more than one article with the same keyword exist then a mild blame (a deontic of the proscription type) is sent to the author(s) of the younger article(s);
- plagiarism: suspected authors are blamed by sanctions (up to the loss of authorisation to write/edit articles).

There are three groups of agents interacting in the scenario:

- "Normal" agents who obey a special vowel harmony (Troitzsch 2008; 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").
- "Rebel" agents who 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").
- "Anarchist" agents who 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.

Figures 6 to 8 show several output graphs of one of the simulation runs. The initial group affiliation of the agents is set randomly. This assignment has a significant influence for the simulation results, as other runs of exactly the same model reveal (EMIL 2009).

The graph in Figure 6 shows the total number of articles, the number of newly created articles and the deleted articles. 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 graph 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.

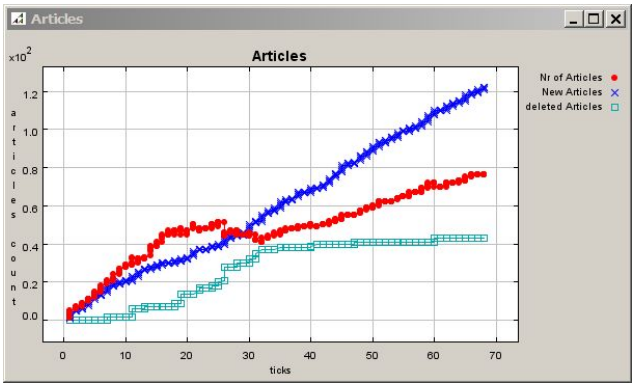

Figure 6: Articles

The graph in Figure 7 shows 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 diagrammed in this graph. The simulation started with 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. 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. In the 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).

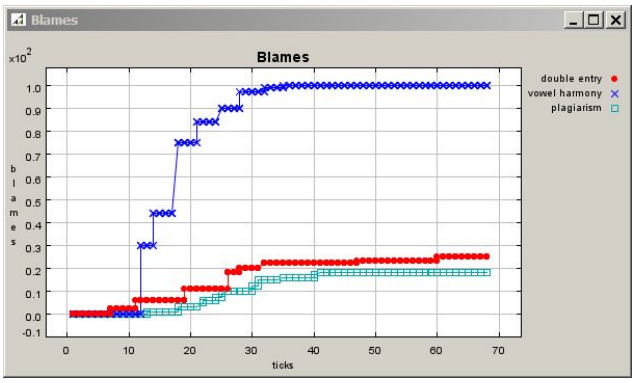

Figure 7: Blames

The graph in Figure 8 shows the group sizes. In this kind of graph, the current number of agents belonging to a group is counted and visualized. Here 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. After a short period of time the group membership stabilizes. In the actual simulation run this happens between time step 11 and 13; when a particularly high number of norm-invocations for vowel-harmony violation is 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 rules the agents comply with.

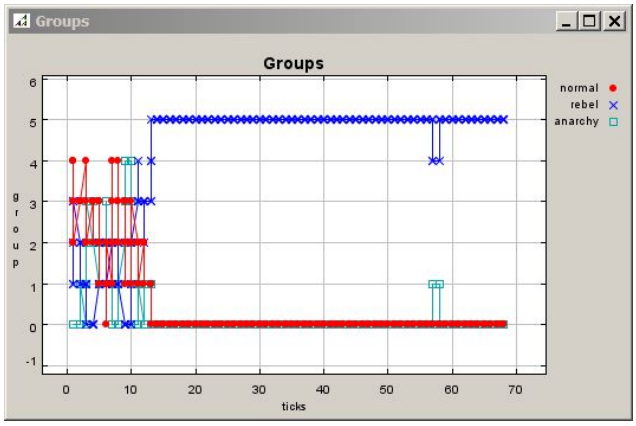

Figure 8: Group sizes

CONCLUSIONS

The paper described architectural aspects of the EMIL-S software for simulating norm formation processes, derived from the logical normative agent architecture EMIL-A. While this paper describes the involved components and processes in general terms (together with a short overview on design and simulation results of the Wikipedia use case), detailed descriptions of application examples in combination with evaluations of the proposed architectures and software would go beyond the scope of this paper. Further information and other application examples can be found in (EMIL 2009), here follows a short use case overview:

- A TRASS-based (Lotzmann 2009) traffic scenario, intended to demonstrate the central features and interfaces of EMIL-S.
- A "multiple contexts" scenario (also TRASS-based), bringing together and into interaction two types of agents, namely one group of agents with a relatively rich cognitive structure and another group of simpler agents which are only social conformers (Andrighetto 2008).
- Another Repast-based scenario of micro-finance groups which goes back to the empirical work of (Lucas dos Anjos et al. 2008).

It is likely that in the future EMIL-S will be used in other and more complex scenarios and applications. Also a further development of EMIL-S towards a powerful platform for an even broader range of rule based normative simulations is one of the most important tasks on the agenda.

REFERENCES

Andrighetto, G., Conte, R., Turrini, P., & Paolucci, M. (2007). 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., Conte, R., & Cecconi, F. (2008). Conformity in multiple contexts: imitation vs. norm recognition. In *World Congress on Social Simulation, Fairfax VA July 14-17, 2008.* Fairfax VA.

Boella, G., van der Torre, L., & Verhagen, H. (2007). Normative multi-agent systems. Dagstuhl.

Conte, R. (2009). Rational, goal-oriented agents. In R. A. Meyers, *Encyclopedia of complexity and system science* (pp. 7533-7548). Springer.

Durkheim, É. (1895/1982). *The rules of sociological method and selected texts in sociology and its methods.* London: MacMillan.

EMIL (2009). Emergence in the Loop: Simulating the Two-Way Dynamics of Norm Innovation. *EMIL-T,* Deliverable 5.1.

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.

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.

Neumann, M. (2008). A classification of normative architectures. In *Proceedings of the 2nd WCSS.* Fairfax, VA.

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.

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.

AUTHOR BIOGRAPHY

ULF LOTZMANN obtained his diploma degree in Computer Science from the University of Koblenz-Landau in 2006. Already during his studies he has participated in development of several simulation tools. Since 2005 he has specialized in agent-based systems in the context of social simulations and is developer of TRASS and EMIL-S. He is also involved in the FP7 project OCOPOMO (Open Collaboration for Policy Modelling) and several other recent projects of the research group. Currently he is doctoral student at the University of Koblenz-Landau. His e-mail address is ulf@uni-koblenz.de.