SIMULATING NORMATIVE BEHAVIOUR AND NORM FORMATION PROCESSES

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

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

ABSTRACT

This paper describes the design of an agent model for simulating normative behaviour and norm formation processes. The model is based on a scientific theory of norm innovation, provided by the FP6 project EMIL (EMergence In the Loop). The main focus of the paper is the conversion of the theoretical framework towards a software implementation that can be applied for multiple simulation scenarios. A simple traffic scenario – where the traffic participants have to find rules to avoid collision incidents – serves as sample application.

INTRODUCTION

This paper focuses on the design and implementation of an agent-based simulation approach which describes the process of norm innovation, illustrated by a simple traffic scenario. It is part 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). The model design is based on a scientific theory of norm innovation, provided by EMIL (Andrighetto et al. 2007a). Using this theoretical framework, an agent-based norm innovation model is designed and implemented, which can be embedded in multiple simulation scenarios. A simple traffic scenario where the traffic participants have to find rules to avoid collision incidents – serves as a sample application. Dynamic model analysis as well as testing this norm innovation approach is supported by implementing the model using the agent-based traffic simulation tool TRASS (Lotzmann 2008).

THEORETICAL FRAMEWORK

The EMIL project especially focusses 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 multiagent 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 multiagent 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 in EMIL can be described mainly by two processes:

- **emergence**: inter-agent process by means of which a norm not deliberately issued spreads through a society;
- **immergence**: intra-agent process by means of which a normative belief is formed into the agents' minds (Andrighetto et al. 2008).

From a more practical point of view, agent architectures, which are principally able to represent mental states and cognitive processes of agents, usually follow the classical BDI (Belief-Desire-Intention) approach (Bratman 1987). Even if an approach like the BOID (Belief-Obligation-Intension-Desire) architecture (Broersen et al. 2001) enhances this concept towards a more norm-like modelling by introducing an additional feature "obligation", the complexity of modelling and simulating norm innovation processes proposed in EMIL do not recommend the usage of such general and predefined architectures. This is because of the fact that all agent knowledge (beliefs, desires, intentions and especially obligations) are seen as predefined and "hard-wired" agent properties. With these structures, it is impossible to answer questions regarding the origin of new norms or the communication of norms between agents (Campenni et al. 2008).

ARCHITECTURE OF A NORMATIVE AGENT

Concept

The prototype model of a norm innovation process presented here introduces normative agents, which are able to learn rules by observation and experience,

Proceedings 23rd European Conference on Modelling and Simulation ©ECMS Javier Otamendi, Andrzej Bargiela, José Luis Montes, Luis Miguel Doncel Pedrera (Editors)
ISBN: 978-0-9553018-8-9 / ISBN: 978-0-9553018-9-6 (CD)

supporting the identification of three main categories of norm innovation (Andrighetto et al., 2007b):

- **Norm extension/adaption:** Extending norms to new entities or social categories;
- **Norm instantiation:** Perceiving/Establishing a new norm;
- **Norm integration:** Determining norms by integrating conflicting norms.

For this purpose the essential communication between agents is based on a **message concept** (Figure 1) in which norm-related agent inputs (events and actions) are triggered by different message types (assertion, behaviour, valuation, sanction, ...).

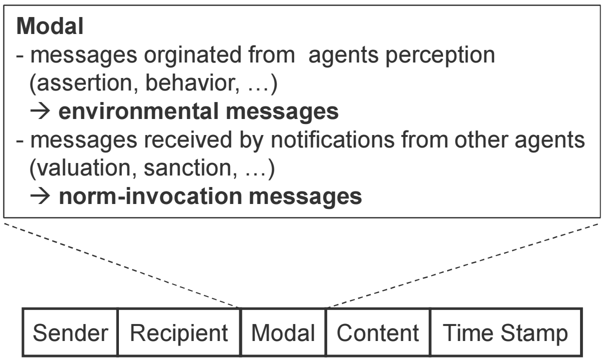

Figure 1: Message structure

Messages and the corresponding events (included in the message component "Content") are divided into two groups:

- **environmental messages/events**
(i.e. originating from the agent perception);
- **norm invocation messages/events**
(i.e. notification or valuation received from other agents).

The processing of environmental messages (see section: Agent Behaviour) implements basically the **(norm-oriented) behaviour** of agents by **selecting and executing actions** coming from

- existing regular norms or
- agent-related preliminary norms or
- the initial rule base (see section: Basic Structures).

The norm-invocation messages are in particular responsible for the **norm formation process**, which includes

- **recognition of existing norms or agent-related preliminary norms** by evaluating the valuation history of events;
- **creation and adaption of preliminary norms** by detecting regularities in event sequences and increasing/decreasing action probabilities;
- **creation of regular norms** by evaluating agent-related preliminary norms**.**

Basic Structures

Agents.
The above mentioned processes require at least two kinds of agent-internal memories: Firstly, an **event board** memorizing a history of incoming environmental events including the conducted actions. And, secondly, a **normative frame** holding preliminary norms, derived from the event board as a result of the simulation. Finally, each agent needs an initial set of rules (**initial rule base**), which contains rules describing basic behavioural elements, and thus constituting the seeds for more complex rules emerging from the simulation process.

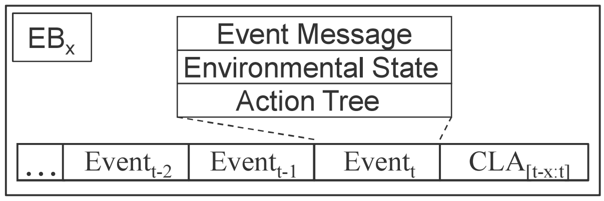

Figure 2: Event board

The event board is a chronologically sorted sequence whose elements contain the following data (Figure 2):

- the environmental message;
- the current (environmental) agent state (e.g. velocity and perception mode for a car driver in the traffic scenario);
- the associated action tree (see Figure 4) with the individual selection probability function.

At any point of time, the actual event board is characterized additionally by

- a classifier CLA (i.e. a value reflecting the similarity of different sequences independent from the chronological order or length), which will be calculated from a subsequence of the event board (starting from the recent event entry) and allows comparisons of subsequences 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 3):

- 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.

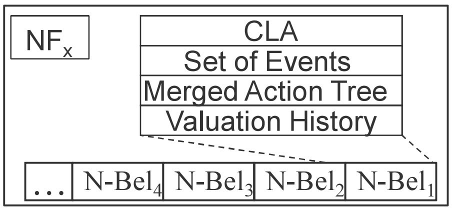

Figure 3: Normative frame

Finally, 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. Each action group represents a number of mutually exclusive actions. The edges of the tree are attached to selection probabilities for the respective action groups or actions (Figure 4).

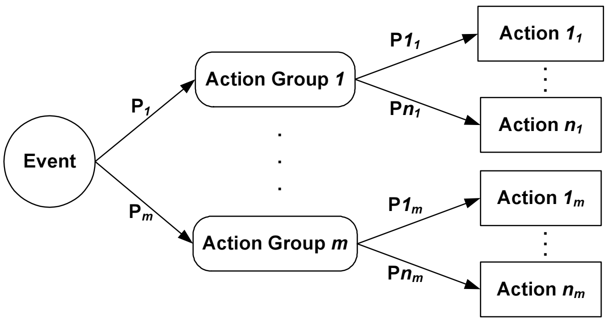

Figure 4: Event Action Tree
(For each action group $x \in [1..m]$ is $P_x \in [0..1]$; for actions $y \in [1..n]$ of action group x is $\sum y_x = 1$)

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 5) consists of the same elements as the normative frames, except of the validation history, which is used to decide, if a preliminary norm will become a norm or not.

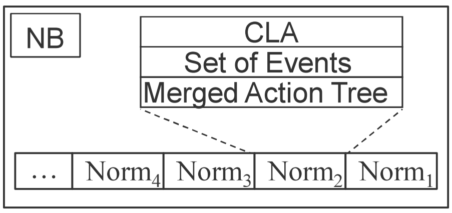

Figure 5: Normative board

Agent Behaviour

Basically, the agent behaviour is determined by a process that is triggered by discrete events in the form of incoming messages. The contents of these messages can be events derived from the agent's perception, or advice from other agents dealing with valuations regarding events the agent is responsible for. Correspondingly, the process generates different outcomes: while in the former case mainly messages describing actions that the agent has to apply on his environment are produced, in the latter case actions that change the internal agent state are performed.

Thus, the first stage of the process is dedicated to the distinction of the incoming message type (by evaluating the message field "Modal") and the selection of the appropriate process. In this context, a perceptive event is called "environmental" (ENV), while the term for a valuation message is "norm invocation" (NI).

Beyond the obvious agent role of an "actor" (i. e. the addressee of the messages is the agent itself), the agent can also behave as an "observer", i.e. the agent monitors the message input from a subset of the agent population. The different inputs are treated as follows:

- for parts of the agent memory (namely the event board) several instances are necessary – one for the actor role and one for each observed agent;
- the actions defined in the initial event-action-trees are attached with an role attribute, i.e. each role define distinct actions.

The two processes are now detailed in the following subsections with the help of UML activity diagrams.

Norm-oriented behaviour

Figure 6 shows the process for the case of an incoming environmental event.

At first, the event information is stored in a newly created event entry and inserted in the event board (EBx indicates an event board of agent "x") as latest entry. Based on this entry, the classifier (CLA) for the precedent time period is generated and stored as attribute of the event board.

This classifier is then used for lookup procedures in the normative data structures. In the first step, the (common) normative board (NB) is inspected.

If a norm with a similar classifier is found, then this entry will be copied into the normative frame (NFx) of the agent. In any case, the normative frame is searched for matching entries (which can be preliminary norms or norms copied from the normative frame) and finally one entry is selected (by an algorithm that reflects the agent's attitude against norms). 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).

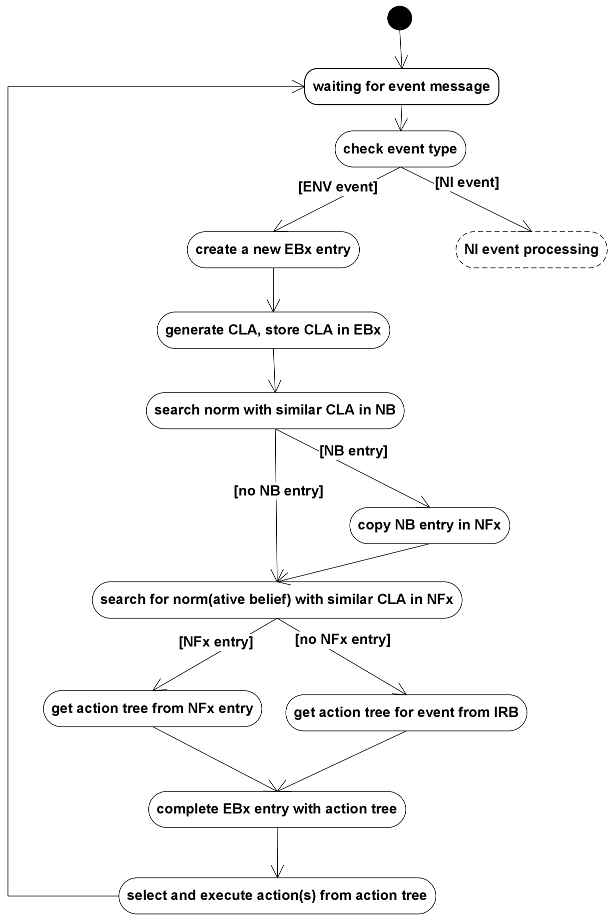

Figure 6: Environmental event processing

In both cases, an action tree can be extracted: either from the normative frame entry, or from the event-action-tree. This action tree is stored in the event board entry that was created at the beginning of the process (thus completing this entry) and afterwards evaluated for selecting and executing appropriate actions and therefore determining the agent's behaviour.

Norm formation

The process branch for an incoming norm invocation event is shown in Figure 7.

In this process, no entry for an event board is generated, but rather the actual event board (EBx) is searched for an event that is valuated by the incoming message. If no entry is found, then the valuation is invalid (for some reason, e.g. the valuated action is outdated and thus already removed from the event board). If an entry is found, it becomes the latest event of an event board sub-sequence for which a new classifier is generated.

This classifier is subsequently used for the same lookup procedures in the normative data structures as already described for the norm-oriented behaviour process: firstly, a possible adoption of a norm from the normative board (NB), and secondly, the search in the normative frame (NFx) for a matching entry. Again there are two possible results:

- No matching entry exists. In this case a new normative frame entry is generated by merging the event-action-trees from all elements of the event board sequence for which the classifier has been generated.
- There are one or more matching entries, from which the entry with the best conformity related to the classifier is chosen and the probabilities of the action tree stored in this entry are adopted in a reinforcement learning procedure.

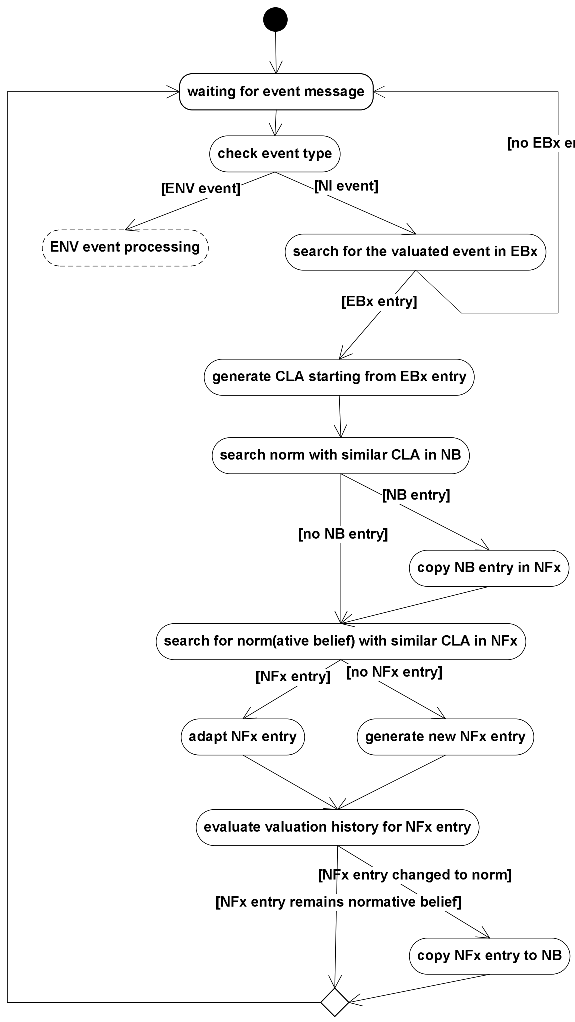

Figure 7: Norm invocation event processing

This (either new or already existing) entry then undergoes a valuation history inspection (after the current event was added to the valuation history) with the objective to decide whether the preliminary norm can be transformed into a regular norm. This decision algorithm considers

- from how many different agents valuations are stored in the valuation history, and
- 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 norm, the content of the normative frame entry will be copied to the normative board (NB). Thus a new norm has emerged.

IMPLEMENTATION REMARKS

In the final section a few remarks on design aspects as well as hints on the implementation progress of the simulation software, and a simple simulation scenario are given.

System Architecture

The norm formation process described in the previous section is the basis for the agent implementation in the EMIL-S simulator.

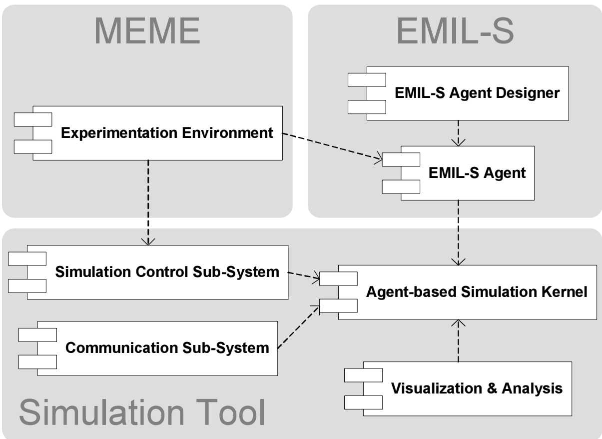

Figure 8: Component diagram of the entire simulation suite

EMIL-S is not realized as a stand-alone software, but rather as a supplemental component of a larger system (Figure 8), consisting of a general purpose simulation tools such as REPAST (North et al., 2006), MASS (Ivanyi et al., 2007b) or TRASS (Lotzmann and Möhring 2008; Lotzmann 2009) and an experimentation environment such as MEME (Ivanyi et al., 2007a), a tool for automated simulation execution with build-in parameter sweep functionality.

EMIL-S Agent Designer

From a modeller's point of view an EMIL-S model can be described completely by a set of event-action trees, which serve as an initial rule base for the initialized agents and provide them with norm formation abilities.
To support the usage of this generic "modelling language", the simulation suite provides a user interface (Figure 9), which facilitates the input of event-action trees (and allows the observation and analysis of simulation runs later on).

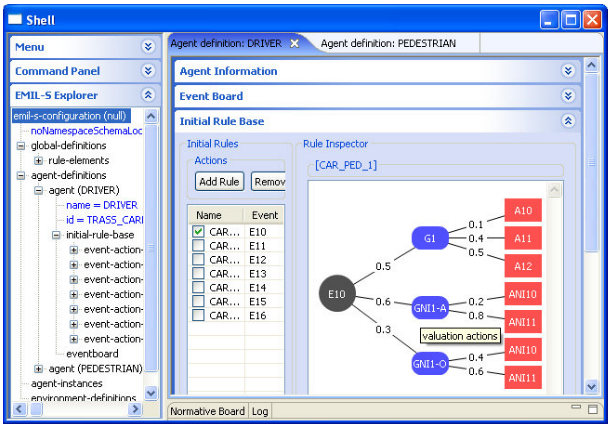

Figure 9: User interface: Model designer, displaying an event-action tree from the initial rule base

At present, the concept of describing norm formation models by event-action trees is tested by transforming different model scenario into this language concept. Examples:

- Traffic scenario (see below, completed)
- Wikipedia scenario (Troitzsch, 2008; completed)
- Microfinance scenario (Anjos, 2008; in progress)
- Scenario on conformity in multiple contexts (Andrighetto et al., 2008; in progress)

Connecting EMIL-S with Simulation Tools

The link between the EMIL-S and the simulation tool is defined by an interface which allows the integration of an EMIL-S strategy layer into agents implemented within the simulation tool. Given that an adequate agent architecture is present on the simulation tool level this can be seen as an approach for realizing embodied agents. Figure 10 shows an example of such an architecture.

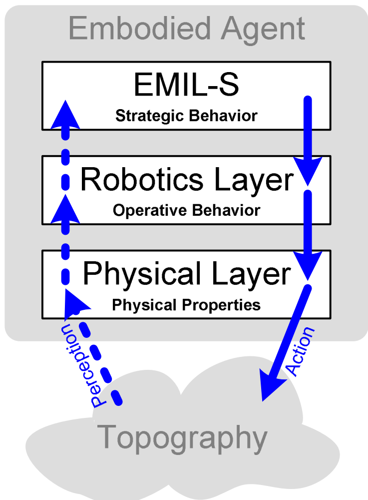

Figure 10: Integration of EMIL-S in TRASS

EMIL-S acts as the strategy layer in cooperation with the “robotics” and the “physical” layers of the TRASS framework.
An example for the combination of EMIL-S and TRASS is shown in the following section: a simulation scenario, dedicated to explore the formation of traffic norms as a result of interactions between car drivers and pedestrians.

Traffic Scenario

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 avoid collisions (as e.g. implemented 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 introduces social capabilities for agents in traffic simulations. The following model is a step toward this more comprehensive view on traffic systems.
The traffic scenario (Lotzmann et al., 2008) consists of a simple topography, 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 blameable 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 frame, 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 11 shows a sample event-action tree for a car driver agent.

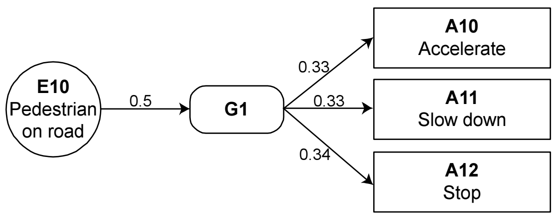

Figure 11: Event-action tree with environmental actions

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 can emerge by linking several event-action trees. With these behavioural rules, norms emerge as soon as the majority of the population starts to use the same rules with similar probability distributions.
Finally, Figure 12 shows a screenshot of a simulation run after a longer learning period. There are 20 pedestrian agents observable, heading towards the target. The majority of the pedestrians use the crosswalk to cross the road, while a minority chose the direct way. The reason for using the crosswalk in spite of a potentially longer route is due to the observation that car

drivers usually stop in front of crossroads while they are ignoring pedestrians waiting at an ordinary road segment. Even this behaviour of the car drivers is result of a learning process due to a higher number of collision incidents in crosswalk-sections.

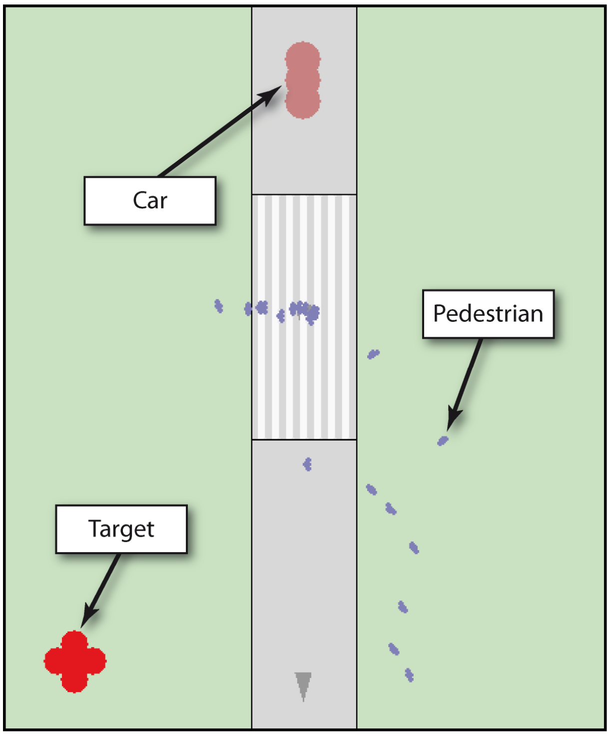

Figure 12: Simulation run screenshot

REFERENCES

Andrighetto, G.; R. Conte; P. Turrini; and M. Paolucci. 2007a. “Emergence In the Loop: Simulating the two way dynamics of norm innovation.” In *Dagstuhl Seminar on Normative Multi-agent Systems*, Dagstuhl, Germany.

Andrighetto, G.; M. Campennì; and R. Conte. 2007b. “EMIL-M: Models of Norms Emergence, Norms Immergence and the 2-way Dynamic.” Technical Report, 00507, LABSS-ISTC/CNR.

Andrighetto, G.; M. Campenni; R. Conte; and F. Cecconi. 2008. “Conformity in Multiple Contexts: Imitation vs. Norm recognition.” World Congress on Social Simulation (WCSS), Faifax, July 14-17.

Anjos, P. L. dos; F. Morales; and I. Garcia. 2008. “Towards analysing social norms in microfinance groups.” 8th International Conference of the International Society for Third Sector Research (ISTR), Barcelona.

Boella, G.; L.v.D. Torre; and H. Verhagen (Eds.). 2007. *Normative Muli-agent Systems.* Dagstuhl Seminar Proceedings 01722.

Bratman, M. 1987. *Intensions, Plans and Practical Reasoning.* Stanford: CSLI Publications.

Broersen, J.; M. Dastani; Z. Huang; and L.v.D. Torre. 2001. “The BOID-Architecture: Conflicts between Beliefs, Obligations, Intensions, and desires.” In: *Proceddings of the 5th International Conference on autonomous agents.*

Campenni, M.; G. Andrighetto; F. Cecconi; and R. Conte. 2008. “Normal = Normative? The Role of Intelligent Agents in Norm Innovation.” Fifth Conference of the European Social Simulation Association (ESSA), Brescia, Sepember 1-5.

Ivanyi, M.; L. Gulyas; R. Bocsi; and R. Meszaros. 2007a. „Model Exploration Module.” Agent 2007: Complex Interaction and Emergence Conference, November 15-17, Chicago, IL.

Ivanyi M.; L. Gulyas; R. Bocsi; V. Kozma; and R. Legendi. 2007b. “The Multi-Agent Simulation Suite.” Emergent Agents and Socialities: Social and Organizational Aspects of Intelligence, AAAI Fall Symposium Series 2007, Washington DC, USA, November 8-11.

Lotzmann U.; and M. Möhring. 2008. “A TRASS-based agent model for traffic simulation.” In *Proceedings of the 22nd European Conference on Modelling and Simulation ECMS 2008*, L. S. Louca, Y. Chrysanthou, Z. Oplatková, K. Al-Begain (Eds.). Nicosia, Cyprus (2008), 97-103.

Lotzmann, U.; M. Möhring; and K. G. Troitzsch. 2008. “Simulating Norm Formation in a Traffic Scenario.” Fifth Conference of the European Social Simulation Association (ESSA), Brescia, Sepember 1-5.

Lotzmann, U. 2008. “TRASS - A Multi-Purpose Agent-based Simulation Framework for Complex Traffic Simulation Applications.” In *Multi-Agent Systems for Traffic and Transportation*, A. L. C. Bazzan and F. Klügl (Eds.). IGI Global.

Neumann, M. 2008. “A Classification of normative architectures.” World Congress on Social Simulation (WCSS), Fairfax, July 14-17.

North, M. J.; N. T. Collier; and J. R. Vos. 2006. “Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit.” In *ACM Transactions on Modeling and Computer Simulation*, 16(1), 1-25.

Troitzsch, K. G. 2008. “Simulating Collaborative Writing: Software Agents Produce a Wikipedia.” Fifth Conference of the European Social Simulation Association (ESSA), Brescia, Sepember 1-5.

AUTHOR BIOGRAPHIES

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 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.

MICHAEL MÖHRING received his first and his PhD degree from the computer science faculty of the University of Koblenz-Landau and has worked in the research group for about 20 years. He developed a multi-level simulation tool called MIMOSE and participated in all the research projects of the group, currently specialising in both multi-agent simulation and in data mining; he has been teaching both information society and data mining for more than a decade. His e-mail address is moeh@uni-koblenz.de.