Emergence of Social Constructs and Organizational Behavior

A cognitive learning approach

Martin Helmhout, Henk W.M. Gazendam and René J. Jorna

Faculty of Management Science
Dept. of Knowledge Management
University of Groningen
PO Box 800
Landleven 5, 9700 AV Groningen
{j.m.helmhout, h.w.m.gazendam, r.j.j.m.jorna}@rug.nl

Abstract:

This paper demonstrates behavioral patterns in organizations based on cognitive modeling of organizational behavior, especially how actors learn to conform to patterns in organizations and how certain patterns of behavior eventually can be institutionalized into a social construct.

Cognitive modeling of organizational behavior argues that organizations are mental representations within an individual's mind. However, not all phenomena can be described at the individual level of description. Hence, a bridge is needed between the social/organizational (external) level and the actor (internal, cognitive) level of description. It is argued in this paper that the social construct can serve as a bridge between the organization and the actor level.

The method used to study organizational behavior is a simulation environment created in JAVA. The modeling of the actors in the environment is based on the cognitive architecture ACT-R (Anderson & Lebiere, 1998). ACT-R was originally developed for carrying out cognitive experiments with a single actor. We have developed a multi-ACT-R environment to address social/organizational behavior elsewhere (see Helmhout et al., 2004). In this paper, we experiment with two actors to observe similar pattern of behavior emerge based on their interaction. We highlight a shared pattern of behavior that can eventually be institutionalized into a rule or in sociological terms a social construct.

Introduction: organizations, actors and social constructs

In organizational studies, many researchers see the organization as an objectified entity with organizational properties and describe the entity as an intelligent object that is able to learn and interact with its environment. Others see organization as an outcome of interaction and learning between individuals themselves and explain behavior based on the collective of interaction between individuals.(see Morgan, 1986)

In this paper we adopt the constructivist point that organizations only exist as groups of actors (human beings, animals and virtual actors) that have a representation of the organization they belong to in their minds and in documents they use. There are no other carriers for organizational knowledge than actors and document. In this view, an organization can exist out of at least two actors, producing and perceiving symbols and/or signs to interact with each other and the organization is constructed reality as long as the interacting actors and their teamwork exist.

We assume that organizational are intelligent actors, because we regard human beings as being reflective, intelligent and thoughtful. According to Wooldridge and Jennings (1995), the following capabilities should be present:

1) *Reactivity*: intelligent actors are able to perceive their environment, and respond in a timely fashion to changes that occur in it.
2) *Proactiveness,* intelligent actors are able to exhibit goal-directed behavior by taking the initiative.
3) *Social ability,* intelligent actors are capable of interacting with other actors.

This description can be extended and refined with (Gazendam & Jorna, 1998):

4) *Representation and interpretation*(including learning), intelligent actors symbolize the environment internally
5) *Autonomy,* the intelligent actors can maintain themselves in their environment based on their own actions and learning.

Capabilities four and five require the actor to have an internal representation of itself and the environment. Representations are symbol structures on which operations are defined that are influenced by interactions with the environment (external) and with the actor itself (internal, cognition).

In studying the organization as a construction of multiple actors, not only the emergent properties or patterns of behavior of a single actor are of value but also the the properties of a collection of actors as a whole ("the organization"). When investigating the emergent properties of the organization, the actors can be investigated for properties that they share with others in the organization, e.g. contracts, social laws or a somehow similar cognitive map that explains their organizational behavior.

From a semiotic point of view, one could say that properties emerge because of interaction and perception which is a process of construction of signs in the actor's mind. This fits well in the view that all artefacts and sign structures have to be *constructed* within boundaries of reasonable computational costs (Simon, 1945/1976; Gigerenzer & Selten 2001). The resulting signs are organized as units of knowledge consisting of a semi-indexical representation of an affordance ad its associated habit of action. An *affordance* is a set of properties of the environment that makes possible or inhibits activity (Gibson, 1979). A *habit of action* is a commitment to act with a connected action program that governs the actual acting

Social constructs are based on the affordance theory and can be seen as representations of cooperation and coordination, based on intertwined habits and mutual commitments that are often expressed in sign structures such as agreements, contracts and plans. A s*ocial construct* (Gazendam, 2003; Liu, 2000) is a relatively persistent socially shared unit of knowledge, reinforced in its existence, by its frequent use. In organizations, social constructs take the form of , for instance shared stories, shared institutions (behavior rule systems), shared designs, shared plans, and shared artefacts. These social constructs support habits of action aimed at cooperation and coordinated behavior..

Social constructs can be differentiated in different types or levels of social constructs: the multilateral institutional or behavioral system in which actors commit to social constructs of a community, organization, country or society. These constructs are formalized and mostly written down in documents and are part of a large population. The second type is the plan or model to which individual actors or groups are committed to. They regulate appropriate behavior for smaller groups. The third type is the bilateral construct established between two actors, e.g. a marriage contract.

The social constructs can have the following properties:

1. *Attached norms or rules*: social constructs contain norms or rules that guide action and prescribe appropriate behavior in a certain context. Norms, can be defined as rules of conduct that constrain self-interested behavior and that are adopted and enforced in an informal setting (Mahoney, P. and Sanchirico, C, 2001).
2. *Written/unwritten (coded/tacit)*: a social construct can be formed and communicated by writing the attached rules down on paper, or they are internalized in actors and with the help of interaction transferred to others (March et al. 2000).
3. *Life span*: every social construct has a starting time, an evolution, monitoring and controlling period and a finishing time (see Liu, 2000, p.67).
4. *Authority, responsibility and control:* according to Fayol (1916), authority can be seen as 'the right to give orders' and responsibility the reliability in obeying to orders. Control can assure actors behave responsible, and can be part of the authoritarian party or a third party. Stamper (2001) argues that authority is necessary to start or finish a social construct. In a community without a clear authority structure, shared authority is a way of making sure others comply to a social construct.

5. *Inheritance or prerequisite of other social constructs*: for example, when preparing a sales contract, sales men refer to their conditions that are registered at the institution of commerce and the registered conditions inherit conditions from public law (see Liu, 2000, p.68).
6. *Scenario*: according to the language action school (Barjis et al.,2001; Goldkuhl & Röstlinger, 2003), there can be a more or less standardized process (scenario) for establishing a social construct between actors; a marriage in the Netherlands has to be prepared by well defined pre-conditions such as pre-marriage in which case the couple has to be registered for at least two weeks and a maximum of a year.
7. *Context*: depending on the context and community (Gibson, 1979), some constructs become active and others are ignored. A policeman at work treats civilians in a different way compared to his children at home, even if they committed the same kind of crime.
8. *Roles and identification*: the actor is given a role or identification, e.g. employer, employee, to make clear the authority, control and rules applied to that role(Liu, 2000).

Social constructs with all its properties are established and maintained by a process of socialization and communication and during their life-span, are monitored by enacting actors, through observation, mentoring, practice, and training (March et al., 2000). Secondly, they create standards of appropriate behavior and stabilization. Thirdly when they are widely known and accepted as legitimate, they are often self-enforcing. And fourthly, they create formal and informal social structures.

In an organization with social constructs the dynamics of social constructs is depending on change of context, entering and exiting actors, on forming, revising, and dying out and can be seen as the survival of the fittest. Some constructs have shown a long experience of successes and are written down on paper, thereby encoding parts of the history and "organizational memory". They accumulate experiences over several generations.

An organization could be defined as a collection of actors that share a collection of social constructs and every actor is taking part in one or more of the social constructs in that organization. However the actor exists out of individual properties and experiences different interactions, perceptions and cognition during its life inside and outside the organization, hence, the actor possesses personal characteristics[1] e.g. attitudes, norms and values. Personal characteristics and experiences create different representations of social constructs in the heads of the actors, thereby creating different representations of the organization as a whole in the mind of every actor. The actors in the simulation have personal characteristics that are defined by the parameters of the cognitive architecture. Based on individual experiences and interaction with other actors, every

1 Compared to social construct we call this personal construct

actor can build up a history of social constructs that are successful or unsuccessful and can act on its environment based on that experience.

Figure 1 shows where social constructs are placed in the context or organization and that social constructs are internalized as shared knowledge, but at the same time can be externalized as for instance contracts, rules or laws. Secondly, there is dynamics of social constructs triggered by language action, signs and context change or interpretation differences.

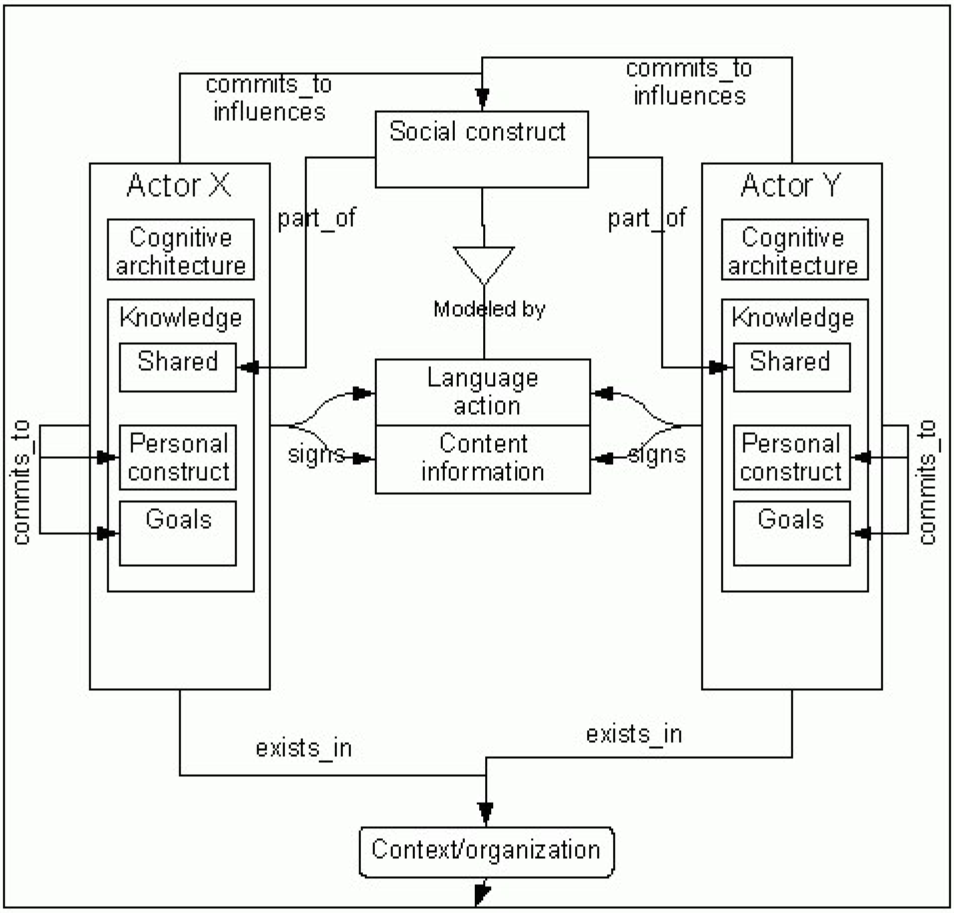

Figure 1: Social constructs, actors and context

In this paper we argue that social constructs are formed by interaction of actors and the process is based on individual learning of each actor involved. Argyris and Schön(1978: 16-17) argue:

> An organization is like an organism each of whose cells contains a particular, partial, changing image of itself in relation to the whole. And like such an organism, the organization's practice stems from those very images. Organization is an artifact of individual ways of representing organization.
>
> Individual members are continually engaged in attempting to know the organization, and to know themselves in the context of the organization. At the same time, their continuing efforts to know and to test their knowledge represent the object of their

> inquiry. Organizing is reflexive inquiry….
>
> [Members] require external references. There must be public representations of organizational theory-in-use to which individuals can refer. This is the function of organizational maps. These are the shared descriptions of the organization which individuals jointly construct and use to guide their own inquiry….
>
> Organizational theory-in-use, continually constructed through individual inquiry, is encoded in private images and in public maps. These are the media of organizational learning.

Hence, the private image is the mental representation of the internalized social constructs, creating a representation of the organization in the mind of the actor and the public map can be compared with externalized social constructs.

For studying organizational behavior of actors that appear to obey a certain social construct, we can build an architecture that is creating mental representations of the actor and is able to learn and forget. The simulation, cognitive architecture and learning principles are explained in the next section.

Simulation, cognitive architecture and modeling the organization

In our research, we make use of a simulation tool to study behavior of organizations and the forming of social constructs. Uses of simulation are diverse and vary from understanding some features of the world to prediction, training and entertainment. Scientists build very simple models that focus on some small aspect of the world by formalizing theories into specifications that can be programmed into a computer. Compared to empirical studies, simulations can be run and rerun many times, with the same environment conditions, and are relatively cheap. Compared to mathematical models, programming languages are more expressive and less abstract than most mathematical techniques. Secondly, programs deal more easily with parallel processes and processes without a well-defined order of actions than systems of mathematical equations. Thirdly, programs are modular, making it possible to change only parts of the program without affecting the remainder. Finally, it is relatively easy to build simulation systems that include heterogeneous actors with different knowledge bases, different capabilities and so on (Gilbert & Troitzsch, 1999).

What do we need to model when we are trying to simulate behavior of organizations? Argyris and Schön (1978) also argue that the organization is a cognitive enterprise, however, does an organization have cognition, or is the cognition of an organization a collection of cognitions of private actors and is it this collection of cognition that makes the organization look like an intelligent creature and having a mind of its own? We approach the organization not as a cognitive enterprise, but rather as an enterprise with cognitive actors. In the construction of a model, we have to start to build the components, before starting to build social constructs or organizational features.

Creating an organizational simulation model demands many capabilities of a software architecture, especially when the organization is seen as a collection of actors that have their own mental representation of the organization as a whole. While in many economic and organizational studies, organizations and the human actors within, are considered as behaving in a fully rational manner without maintaining representations, searching for optimal solutions, we describe organizations and actors as boundedly rational (Helmhout et al. 2003). Bounded rationality is the concept that Simon(1945/1976, p.80) coined to conceptualize the limitations of "perfect" representations of the environment by the human actor and also of the mental processing capacity of the actor. They are boundedly rational, because they do not have a complete representation of the world around them (the ontological argument) and even if they have, they are not able to process all the information and opportunities they encounter.

Considering the fact that actors are boundedly rational and have a mental representation of themselves and the environment, the actors need to be supported by a cognitive architecture. The cognitive architecture enables an actor to perceive, reason, learn, forget and communicate with other actors independently.

Newell(1973b) came with a production-system theory of human cognition. He tested a number of different production systems, concluding with his SOAR theory (Laird, Newell & Rosenbloom, 1987) of human cognition. The rise of the ACT-R system (Anderson, 1976; Anderson & Lebiere, 1998) resulted in a production system that incorporates properties of SOAR, but also includes 'connectionist-similarities' that makes ACT-R a rich, hybrid architecture.

Many architectures of cognition have been developed (Posner, 1989), but the two main-stream architectures are SOAR and ACT-R. For our research and simulation we have chosen the ACT-R architecture, because it makes distinction between procedural and declarative knowledge and has so-called sub-symbolic properties of learning attached to every chunk in memory.

In this paper we are not interested in the exact workings of ACT-R, but explain some basic assumptions of the architecture that are necessary for demonstrating the simulation outcome.. Figure 2 displays the main components of the cognitive architecture ACT-R. The architecture consists of three main modules: the declarative memory, the procedural memory and the goal-stack. The declarative memory is an ordered collection of knowledge chunks stored in a list or tree-structure and can be compared with a database consisting of facts, which are most of the time created through interactions with the outside world (knowledge gathering) or by internal cognition (knowledge creation). The procedural memory contains so-called procedures that are the reactors of

ACT-R. They enable goals to be transformed, and sub-goals and new procedures to be formed. The procedures are responsible for how an actor reacts to certain goals. The goal stack keeps track of the state of the current (sub)goal ((sub)-problem) that needs to be solved and the goal is solved when the goal is released (popped) and the stack is empty. In the simulation, for example, when an actor perceives a chunk that represents another actor, a procedure is triggered and the goal is transformed to the status of "evasion".

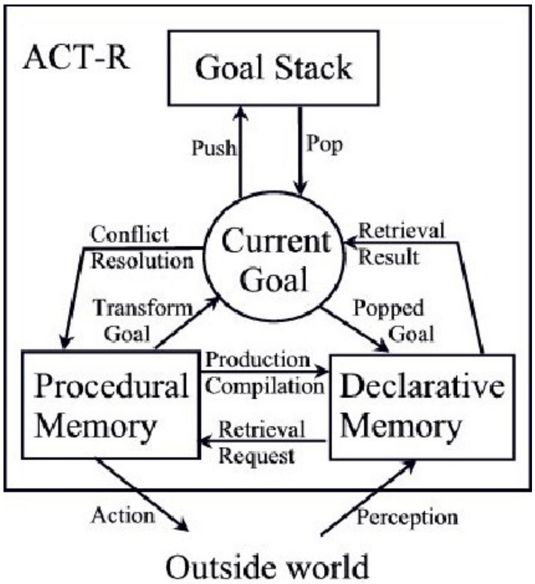

Figure 2: Information flow within ACT-R: a cognitive architecture (source: Anderson & Lebiere, 1998)

The simulation in this paper is about the evolution of social constructs and particularly how these constructs are learned by the actors individually. Therefore we are interested in the learning capabilities of the actor and its cognitive architecture.

ACT-R engine and cognitive learning

ACT-R distinguishes between two types of knowledge creation, creation by perception and creation by cognition. As ACT-R distinguishes also between procedural and declarative memory, learning in this context can be of two types: *learning of procedures* and *learning of declarative knowledge*.

Declarative Learning

Declarative learning can take the form of creating new knowledge or strenghtening existing knowledge. The first type of declarative learning is called chunking: creating new chunks in memory, based on perception or internal cognition. In the simulation, the other actor is perceived and stored in a chunk and is used by procedures that calculate change of distances between the other actor and itself, and because this type of learning is a discussion about perception and vision, it is not discussed in this paper. The second type is depending on time-dependent properties of the chunk and is characterized by sub-symbolic learning

Declarative *sub-symbolic learning* is based on base-level activation B_i: the estimate of how likely a chunk is to match a production, depending on the frequency and recency with which that chunk has been used. Hence, the more often a chunk is retrieved from memory, the more likely the chunk will be re-selected.

Declarative *sub-symbolic learning* is also based on associative strengths: depending on the associative strengths and the frequency with which a chunk is observed, some chunks will be more strongly connected to particular goal requests than others. This type of learning approximates connectionism, in the sense that certain associations or paths from procedure (and goal) to chunks in declarative memory are stronger than others (see Anderson & Lebiere, 1998 for details).

The purpose of the simulation is not to focus on declarative learning. However, as base level learning B_i is one of the most important and commonly used formulas of ACT-R, we will introduce this here. Base level learning (see equation 1.1) is an estimation of the log odds (of all presentations of a chunk in history) that a chunk will be used and is defined as:

$$B_i = \ln\left(\Sigma_{j=1}^{n} t_j^{-d}\right) + \beta$$ *Base-level learning equation (1.1)* where,

t_j	=>	represents the time-difference (t_{now} - $t_{presentation}$) when the chunk was represented in memory,
d	=>	is the decay rate
β	=>	the initial activation

The base-level learning is describing a logarithmic power function, approximating the Power Law of Forgetting when no chunks are presented and the Power Law of Learning when many consecutive chunks are presented.

The parameters of the base-level learning equation can give every actor or even every chunk different rates of decay, thereby allowing the actor to forget when no representations of a chunk take place and enable it to remember when representations take place. The repetition of the times-table at primary schools is a good example of putting the Power Law of Learning to work.

Procedural Learning

While declarative chunks are selected based on matching and highest activation, procedures are selected on the basis of expected gain or utility. When there is a goal to be solved, one or more procedures can be candidates to solve the problem. However, given that ACT-R is a serial processor, the procedure with the highest expected gain or utility is selected.

The utility of a procedure (see equation 1.2) is defined as:

$$U = P * G - C + \sigma$$ *Utility equation(1.2)*

where,

P => learning probability

G => goal value

C => cost of using the procedures

σ => stochastic noise variable

G is the importance that ACT-R attaches to achieving a particular goal. It is possible to assign every procedure a different g-value that influences the ACT-R engine in selecting a particular procedure. However in most ACT-R simulations, the same goal-value is assigned to all procedures.

The probability *P* (of learning) depends on two sub-probabilities: *q*, the probability of the procedure working successfully, and *r*, the probability of achieving the goal if the procedure works successfully. *P* is defined as:

$$P = q * r$$ *Probability of Learning Equation (1.3)*

where,

q => probability of the procedure working successfully

r => probability of achieving the goal if the procedure works successfully

Cost C (see equation 1.2) reflects the amount of time necessary for a procedure to complete its task. The noise σ is added to the utility to create non-deterministic behavior, otherwise with deterministic behavior, the simulations of ACT-R would give no room for other procedures that have (almost) similar parameters to fire.

In this paper we are especially interested in procedural learning parameters q and r, because they are of the most influence on behavior in the simulation.

As with chunks, there are two types of learning regarding procedures: procedural symbolic learning and procedural sub-symbolic learning. In procedural symbolic learning, ACT-R distinguishes between specialization and generalization. While specialization is used for problems that are recurring often and become routinized, generalization is used in analogy problems, the situations where problems that look similar can be solved by generalized procedures. Until now, ACT-R has no stable and well-defined solutions for the problem of this type of procedural learning. Although there are two types of procedural learning, we focus on the procedural sub-symbolic learning here.

Procedural sub-symbolic learning: probability q and r

The parameters that estimate the learning probability P of the selection of a procedure are probability *q* and *r* that describe the success ratio of procedure usage. They reflect the history of successes and failures of a production. Probability *q* is the success-ratio of executing the procedure directly; it keeps track of the immediate successful execution of the condition and action side of a procedure. Probability *r* is the success-ratio calculated over the achievements of completing a goal, after all the subgoals are solved and the goal of the current the goal level is achieved and popped. In other words, all goals that follow the current goal have to be fulfilled successfully to accomplish the current goal. For example, consider three doors with door 1 being locked, door 2 being unlocked, and door 3 being unlocked and having a present. The main goal is to get the present. The subgoal is to open a door and get the present. There are three procedures available to open the door, “open door 1”, “open door 2” and “open door 3” and one procedure “get the present”. Assuming that the procedure “open door 1” is selected, an immediate failure (q and r are affected) of opening the door is accounted for. Next, procedure “open door 2” is selected and the subgoal, “open door”, succeeded (parameter q of “open door 2”: a success is added). However, the procedure “get the present” fails. In that case, parameter r of “open door 2” is affected (failure) and parameter q and r of “get the present” are affected (both failure). Next, procedure “open door 3” is selected, the opening of the door succeeded and the present found, i.e. the main goal is achieved (a success is accounted for q and r of both, “open door 3” and “get the present”.. In this case, the actor has learned that door 1 is locked, door 2 has no present and door 3 has the present. After repeating this experiment with the same conditions a couple of times, the actor learns that “open door 3” is the most successful procedure.

The probability to learn is defined in the following manner(see equation 1.4):

$$q, r = \frac{Successes}{Successes + Failures} = \frac{S}{S+F} \quad \textit{Probability Learning Equation(1.4)}$$

From the formula, it is clear that with initialization, either S >= 1 or F => 1. The default initialization of the procedural parameters is to give S a value of 1 and F a value of 0, i.e. an optimistic view of success of the procedure.

Event Discounting

In calculating successes, failures and efforts thus far, no time component has been included. That means that right now, events and efforts in the past would be equally weighted as the ones experienced at present. This means that the actor would be locked into a path-dependent behavior and would have no freedom to choose from procedures that failed in the past. However, people are more aware of the impact of present experiences than experiences encountered in the past. Therefore ACT-R uses functions to discount the effect of past experiences by implementing a power-decaying function that is similar to the base-level learning equation. The formulas for discounting successes and failures can be seen in equation 1.5:

$$Successes \vee Failures = \Sigma_{j=1}^{m} t_j^{-d} \quad \textit{Success / Failure Discounting Equation(1.5)}$$

where,

m => number of successes or failures

t_j => time difference, now – occurrence_time of the success / failure

d => success decay rate (default: 0.5)

The formula allows the user to give different decay rates for successes and failures. For example if the decay ratio of failures is lower than successes then ACT-R tends to remember negative experiences more quickly than positive experiences.

Learning from interaction: extending the ACT-R architecture

In order to enable actors to learn from interaction in a multi-actor environment, the ACT-R architecture has to be enhanced and redesigned in several ways. First of all, ACT-R is designed as a single threaded program for interaction between a user and a computer. Secondly, because ACT-R is not a multi-actor environment, there is no communication protocol and no simulation environment.

The single ACT-R program is replaced with a client / server system of communicating programs or threads, one for each actor and one for the simulated physical environment. The memory structure of ACT-R has been rebuild in order to enable new memory access structures that are necessary for interactive behavior like perception, communication and social behavior. A message parser for sending and interpreting messages together with a perception handler and position handler has been added to extend the ACT-R architecture.

Based on this multi-actor system design, we try to explain the evolution of an internal construct based on past learned experiences with other actors..

Simulating the evolution of an (internalized) social construct

In this section we explain how an (internalized) social construct with its attached norm(s) and rules evolve over time. A norm in a social construct can be influenced by other norms, but in the simulation presented here, we keep things simple by only focusing on one simple social construct with one simple norm.

The simulation describes a two-dimensional environment in which actors can move and have to pass one another on the right or left side. In countries like England and the Netherlands, these rules have evolved differently and have been institutionalized by the enforcement of law, as passing on the left or passing on the right side. In our simulation, we want to study how this norm evolves over time and if a preference for either passing left or right is established between two actors. We illustrate how one of these norms (as part of a social construct) becomes preferred over the other by the process of cognitive learning.

We initially analyse the given condition using repeated games theory.(see Axelrod, 1984) This theory models norms / social constructs as equilibrium strategy choice.

Figure 3 describes a coordination game model in which two Nash equilibriums are present.

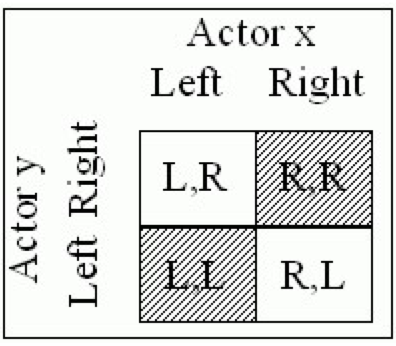

Figure 3: Norm selection as a coordination game (Picker, 1997)

Payoffs are the highest when both actors pass on the right side or on the left. Hence when selecting a side to pass, both selecting right, and both selecting left is a Nash equilibrium. We can extend the model with individual experiences and memory of past experiences. We can use a simple form of learning, partially based on the utility function of ACT-R. We define the following formula:

$$Utility = \frac{1+S}{1+S+F}$$

Every actor remembers its score and it adapts its strategy in the field over time (see figure 4). At t = 0, the actor has no preferences and in this case selects randomly, actor X left and actor Y right. Both actors receive a penalty of F = 1 and the utility of that selection of strategy changes to ½ . Now the selection is exactly the opposite direction, in which actor X selects right and actor Y left. Another penalty creates a field in which both strategies are valued again as equal. Suppose that both, actor X and actor Y, decide to pass right. Now both are credited with a success and a Nash equilibrium has been established.

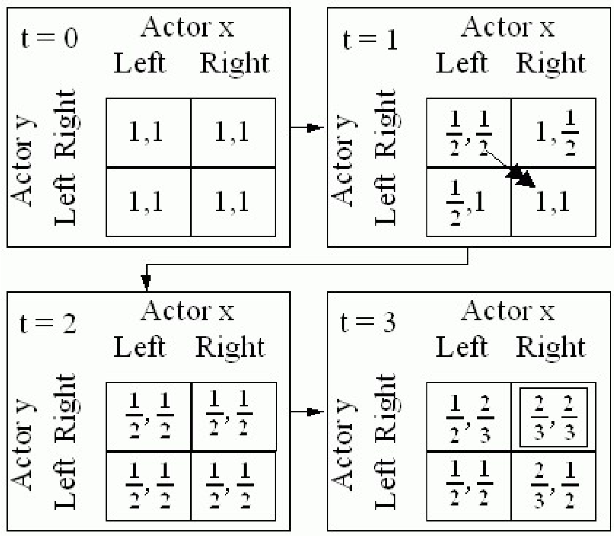

Figure 4: Nash equilibrium development over time

In the above pattern of behavior, actor X and Y learn over time that one of the strategies, passing both on the left, or on the right provides the highest payoff.

The application of game-theory to learning behavior clarifies the pattern of behavior of the actors for the simulation. In the ACT-R simulation, the utility calculation is much more complex. Noise is added to account for some variance in decision making.

In the simulation experiment, the actors are given an equal personal construct, meaning that the initialization parameters of the actors are equal; equal decay rates, equal utility-preferences, equal motivation value to solve the goals and equal procedural, declarative memory and noise distribution function. The personal constructs are identical to make sure that the simulation outcome is based on interaction and not on difference of personalities. If for example we would give actor X compared to Y a higher decay rate, it would forget its past quicker than Y and probably would behave differently, however in this experiment we are not testing difference between behavior of individual actors.

The simulation includes two actors, each on an X,Y grid, traveling from coordinates 100,100 to 200,100 and back and the other way around for the other actor. In that case, the actors meet up with each other, and decide to pass left or right. In figure 5, you see a screen shot where actor “ayse” decided to go left and actor “martin” decided to go right. Based on their perception, they notice if their evasive behavior was successful or not.

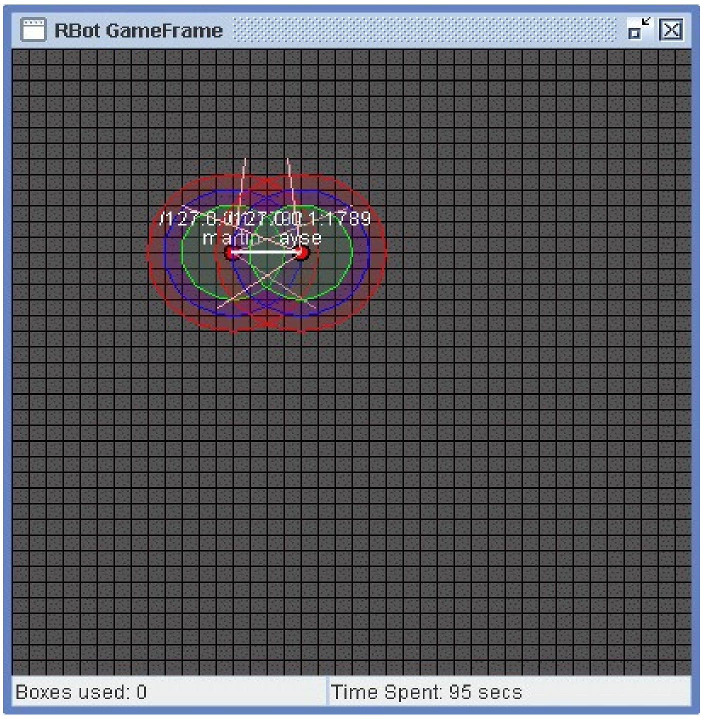

Figure 5: Game Frame of two actors trying to pass each other

The actors learn in an indirect way, meaning, they evade right or left and then see if their evasion was successful or not. Based on this experience, they update their productions and use this experience in the next encounter.

As shown in the game theoretical approach, when actors choose both the same strategy, they pass successful and start to self-enforce that particular strategy and leaving no opportunity for the other strategy to emerge. In the simulation, we encountered the same behavior when actors chose the successful strategy, but we found an initially unstable behavior when actors began with a non-cooperative move. Apparently, the experience of immediately success gives the actors the intention to choose the same strategy again, reinforcing successful behavior and immediately get a lock-in, but selecting initially the wrong strategy, the actors need time to find out what strategy is successful. In this experiment setup, we are interested in the simulation where the bother actors initially choose te wrong strategy and have to decide on experience which strategy is acceptable for both actors.

By initially selecting the colliding strategy, the actors behave like hopping from left to right and from right to left[2]. After some collisions, noise gives the actor freedom of escaping from hopping and after some time settling into the successful strategy, either left or right.

To understand the behavior, we want to look inside the actor's memory and see some explanations; how can the behavior of escaping and reinforcement be explained and is this behavior reinforced in such a way that selecting the other strategy becomes unfavorable.

Figures 6 till 13 give the results of the experiment that is starting with a collision, first no particular strategy is chosen, as well as right move as left move are preferred. However after some first trials, the actors are preferring one strategy (based on interaction), and after a while so strongly, that the actors only choose the left strategy, and the utility difference between right and left becomes so significantly large, that the actors are locked-in into the left passing strategy.

The successes and failures, ratio and utility of actor X and Y, figure 6, 8, 10 and 12, indicate that many interactions reinforce the left strategy and causing the actor to behave in that fashion all the time, as long as punishment is less heavy than reward.

2 In real life, probably everyone has had some experience of walking through a corridor and has had some collisions with people coming towards you.

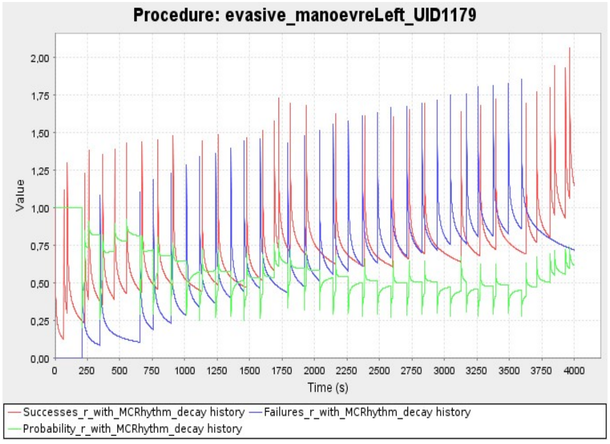

Figure 6: Actor X: Success Failure, Ratio Left Move

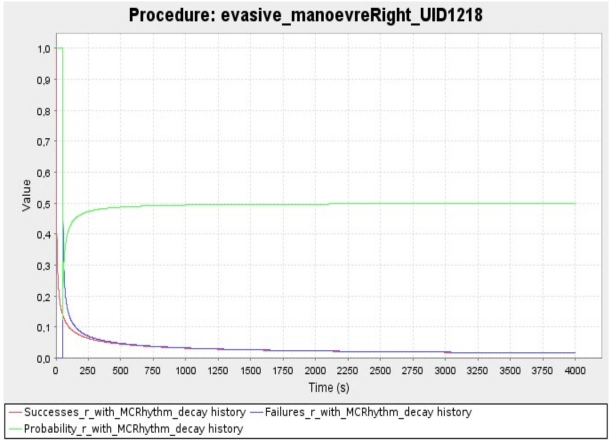

Figure 7: Actor X: Success Failure, Ratio Right Move

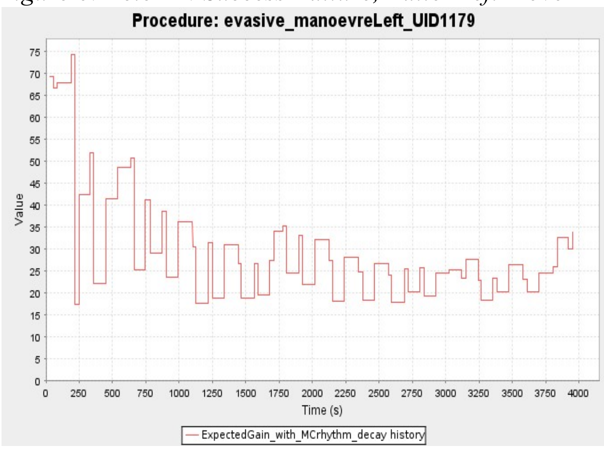

Figure 8: Actor X: Utility Left Move

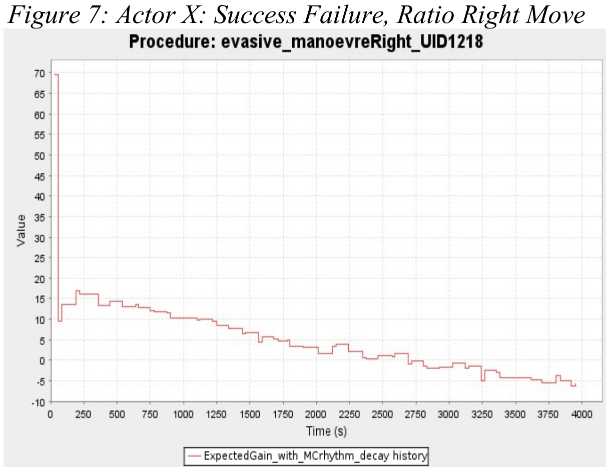

Figure 9: Actor X: Utility: Right Move

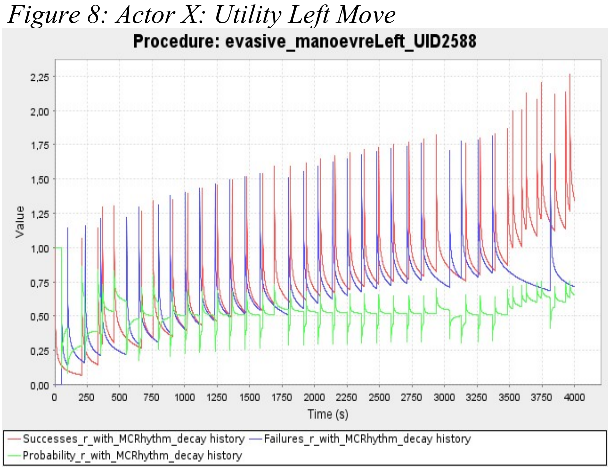

Figure 11: Actor Y: Success, Failure, Ratio Left Move

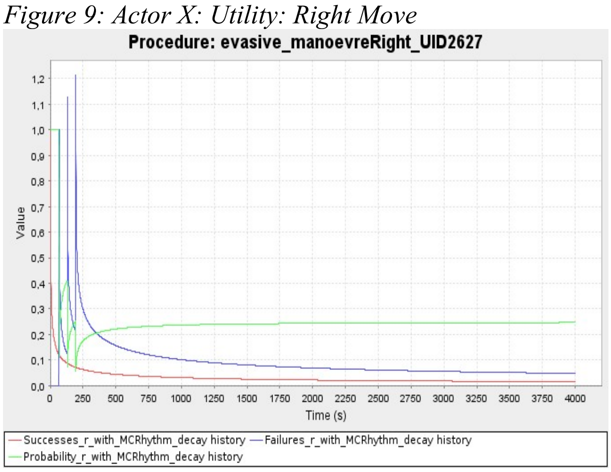

Figure 10: Actor Y: Success, Failure, Ratio Right Move

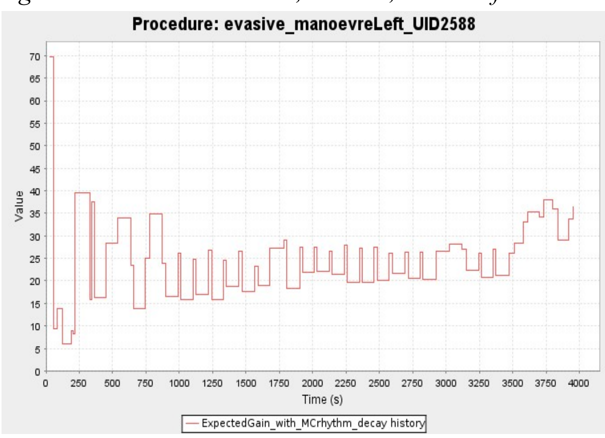

Figure 12: Actor Y: Utility Left Move

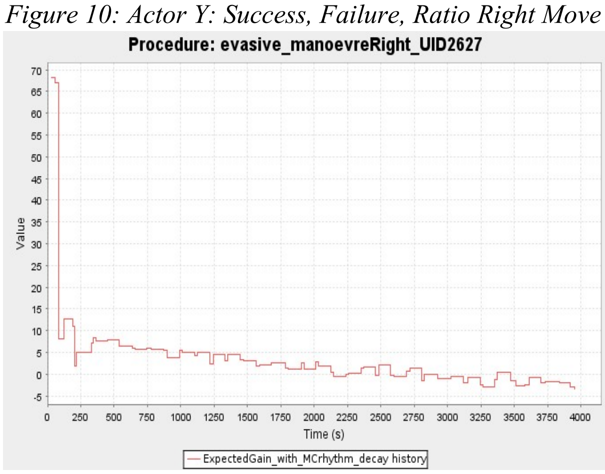

Figure 13: Actor Y: Utility Right Move

According to the behavior of the simulation, if we do not look inside the actors, we could conclude to see some start of "formalized" behavior, in that sense, that actors seem to have a mutual agreement. However, this mutual agreement is based on many interactions and reinforcement of the successful strategy and this agreement exists not on paper but inside the head of the actor.

Assuming that a social construct has evolved, based on cognitive learning, the properties of the social construct can be analyzed as follows:

1. *Attached norms or rules*: the social construct that is formed, is based on one enforced rule, choosing the successful strategy, in this case left.
2. *Written/unwritten*: in the experiment, the rule is unwritten, internalized, however, the "traffic law" can be easily written down and that is exactly what most countries have done in the world, however the step of forming an externalized social construct is not simulated in this paper..
3. *Life span*: The life span in the simulation is infinite for the construct that survives, however with a more dynamic field, with entering and exiting actors, the life-span is uncertain.
4. *Authority*: authority is not formally present. Authority could be seen as being shared over the community, with the actors that are the longest in the system, authorizing others to behave like they do. There is no authority like a policeman fining the actors for incorrect behavior, but there is some kind of indirect control. As long as the actors are passing successful, they don't need to correct the other. But suppose one actor is deviating from the strategy, the other actor will immediately cause the actor to fail, and trying to "negotiate" again about what strategy should win.
5. *Inheritance or prerequisite of other social constructs*: there is no inheritance of other constructs, we only can say that the social construct inherits its norm from internal evolving productions.
6. *Scenario*: because the social construct has to develop itself, there is not a pre-defined scenario.
7. *Context*: clearly, the social construct develops and becomes active only during the time the actor perceives another actor, meaning that as long there is a context present with many actors, the social construct can reinforce itself.
8. *Roles and identification:* the roles in the simulation are for both actor similar, one actor has no privileges and no other role than the other.

Based on the presence of properties of the social construct and its frequent use, we can conclude that there is a form of social construct that is internalized in the mind of the actors and that behavior in this case is based on cognitive learning.

Secondly, we can conclude that organizational learning in this experiment is a form of (situated) interactive learning. Actors adapt based on anticipated other actors' signs and try to impose their behavior onto others by signaling their behavior.

Thirdly, the number of interactions within a community have to be high enough to be able to enforce certain type of behavior, because actors forget what was (not) successful behavior.

Fourthly, a lock-in could occur, in the case when actors after enough times of reinforcement, create a mindset that becomes resistant to change and the only way to change is probably a dramatic shock such as entrance of actors that have a different mindset, exiting of experienced actors or enforcing a punishment system like an institution. .

Discussion / further research

This paper showed that organizational patterns can emerge solely out of interactions between actors that develop a similar way of thinking. The cognitive learning approach explains that all organizational behavior is based on behavior of the individual and that it are the observers (the actors themselves) who declare that there is an organization and signal the existence of this organization to the outside world.

According to many researchers (March et al., 2000) organizations encode their behavior into formal rules that guide the members to behave in a defined way. The simulation shows interactive learning between two actors, and that an agreement can be reached, but what happens when more actors enter the field, exit the field or have different learning capabilities?

The experiment gives some insight into the behavior of cognitive plausible actors in a social interaction, however we are aware of the fact that the simple experiment has to be supported with similar social empirical data to explain that what is happening on the cognitive (symbol) level is causing changes in the organizational (social) level and the other way around.

In the future we want to put more actors in the field and make many interactions with different actors possible, in order to find explanations for, for example (1) the stabilization time of a norm, (2) if clusters are forming (3) trying to introduce a manager (police man) that controls the norm and what the span of control is for the police man to make sure the system is still following the norm.

Another interesting experiment is to have a group of actors with an already evolved stabilized strategy and explore what critical mass (with what kind of properties) of other actors that have preference for a different strategy is necessary to convince the first group to adopt the new strategy..

Finally, this paper showed an example of cognitive learning of an interaction pattern between two actors, but a more complicated task environment could be designed in which other problem-spaces play a role. The purpose of the research is to create a generic simulation toolkit that can be applied in many fields where cognitive plausible actors could be applied to help for example decision making, militaristic games, planning and so on.

References:

Anderson, J.R. (1976). Language, Memory and Thought. Hillsdale, NJ: Erlbaum Associates.

Anderson, J.R. & Lebiere,C. (1998) *The atomic components of thought.* Lawrence Erlbaum Associates, Mahwah.

Axelrod, R., (1984) *The evolution of cooperation*. New York: Basic Books

Argyris, C., & Schön, D. (1978) *Organizational learning: A theory of action perspective,* Reading, Mass: Addison Wesley

Barjis, J., Dietz J. L. G., & Liu, K., 2001 – "Combining the DEMO methodology with semiotic methods in business process modelling". In: K. Liu, R. J. Clarke, P. B. Andersen & R. K. Stamper, *Information, organisation and technology: Studies in organisational semiotics* (pp. 213-246). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Fayol H. 1916/1918. Administration industrielle et generale. *Bulletin de la Sociéte de l'Industrie Minerale.* Dunod: Paris

Gazendam, H.W.M, (2003), *Models as coherent sign structures.* In: *Dynamics and change in organizations: Studies in organizational semiotics 3,* editors: H.W.M.Gazendam, R.J.Jorna & R.S.Cijsouw, pp. 183-213, Boston: Kluwer.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston, MA: Houghton Mifflin.

Gigerenzer, G., & Selten, R. (Eds.). (2001). *Bounded rationality: The adaptive toolbox*. Cambridge, MA: The MIT Press.

Gilbert, N. & Troitzsch, K.G., (1999), *Simulation for the social scientist.* Open University Press, Buckingham, Philadelphia

Goldkuhl, G., & Röstlinger. 2003 - "Towards an integral understanding of organizations and information systems: Convergence of three theories". In: H. W. M. Gazendam, R. J. Jorna & R.S. Cijsouw (Eds.), *Dynamics and change in organizations: Studies in organizational semiotics* (pp. 133-161). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Helmhout, J.M., Gazendam, H.W.M. and Jorna R.J.J.M., (2003), *Social constructs and boundedly rational actors.* Proceedings Organizational Semiotics 6, Reading UK.

Helmhout, Gazendam & Jorna, (2004), *The Role of Organizational Semiotics and Social Constructs in the Social Awareness of Simulated Cognitive Plausible Actors,* In proceedings Organizational Semiotics 7, 2004,Portugal: Setubal.

Laird, J.E., Newell, A. & Rosenbloom, P. (1987). SOAR: An architecture for general intelligence, *Artificial Intelligence, 33,* pp. 1-64

Liu, K., (2000), *Semiotics in information systems engineering.* Cambridge, England: Cambridge University Press.

Mahoney, P. & Sanchirico, C. *Competing Norms and Social Evolution: Is the Fittest Norm Efficient?* Working Paper, No 00-15, University of Virginia, School of Law, January 2001

March, J.G., Schulz, M. & Zhou, X., (2000), *The Dynamics of Rules: Change in written Organizational Codes.* Stanford University Press, Stanford: California

Mintzberg, H.(1979) *The structuring of Organizations.* Englewood Cliffs, N.J. 1979

Morgan, G. (1986). *Images of organization.* Sage Publications inc., California

Newell, A.,(1973b), You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In Chase, W.G., editor, *Visual Information Processing.* Academic Press, New York.

Picker, R.C. (1997), *Simple Games in a Complex World: A generative Approach to the Adoption of Norms*. http://www.law.uchicago.edu/Picker/aworkingpapers/norms.html University of Chicago.

Posner, M.I.(Ed.).(1989) *Foundations of cognitive science.* Boston(Mass): MIT Press.

Simon, H.A. (1976). Administrative behavior: A study of decision-making processes in administrative organization (3rd ed.). New York: The Free Press. (Original work published 1945).

Stamper, R.K., (2001) Organisational semiotics: Informatics without the computer? In K. Liu, R.J.Clarke, P Bogh Andersen, & R.K. Stamper(Eds.), *Information, organisation and technology: Strudies in organisational semiotics* (pp. 117-171). Boston, MA: Kluwer Academic Publishers

Von Wright, G.H(1963) Norm and Action – A Logical Enquiry, Routledge, London