Interdisciplinary Development and Evaluation of Cognitive Architectures Exemplified with the SiMA Approach

Samer Schaat, Alexander Wendt, Stefan Kollmann, Friedrich Gelbard, Matthias Jakubec
(schaat, wendt, kollmann, gelbard, jakubec@ict.tuwien.ac.at)
Institute for Computer Technology, Vienna University of Technology
1040 Vienna, Austria

Abstract

In this paper we show how simple simulation scenarios can be used to develop and test foundational functionalities of cognitive architectures, exemplified with the SiMA architecture. We present an interdisciplinary methodology that considers the challenges in capturing and evaluating basic functionalities of the human mind. In this regard, we structure and concretize assumptions from various disciplines and show how we evaluate their plausibility in a consistent model, using parametrized simulations.

Keywords: Artificial Intelligence; Cognitive architectures; Intelligent agents; Computer simulation

Introduction

The computational approach to examine the human mind provided a powerful methodology of research. When the examination of information processing systems (such as the human mind) is at stake, computer scientists are particularly suitable to contribute their experience. Nevertheless, computer scientists often still approach problems of Cognitive Science in a classical AI way. This is especially the case regarding *interdisciplinary*: instead of concretizing models of the human mind from other disciplines into a consistent and testable form, often own models that suffice computational criteria (such as efficiency) are developed. In this regard computer science stays behind its possibilities in contributing to understand the human mind. A counter-example is to use the computational methodology in an approach of *synthetic psychology* (Braitenberg, 1986). Similarly, computational models often focus on simulating high-level cognition without considering their foundations, such as motivation and emotion. We propose a more *natural approach* in considering the *foundations of cognition* in a unified cognitive architecture that harnesses the possibilities given by computational simulations and is able to provide a unified tool to test assumptions and their relationships to each other. We will use superficially simple simulation scenarios to guide our development and test the resulting model. On the one hand this considers that most of humans' behaviour is covered by every-day capabilities (what Bargh & Chartrand (1999) called the unbearable automaticity of being). On the other hand our experience with the cognitive architecture *SiMA*[1] *(Simulation of the Mental Apparatus & Applications*) (Schaat, Wendt, Jakubec, et al., 2014; Dietrich, et. al., 2014) showed that – especially when the foundations of the human mind are at stake – every-day behaviour is more suitable to analyse the basic functions of the human mind.

[1] ARS (Artificial Recognition System) was renamed to SiMA.

State of the Art

A good overview and classification of cognitive architectures are elaborated in (Duch, Oentaryo & Pasquier, 2008; Langley, Laird & Rogers, 2009; Vernon, von Hofsten & Fadiga, 2010). There, cognitive architectures are classified into three categories: symbolic, emergent, and hybrid architectures. *Symbolic architectures* process high-level symbols like objects or concepts and derive action plans thereof. In *emergent architectures* no symbols are processed but low-level activation signals in a network, for example an artificial neural network, are propagated. Actions emerge out of holistic structures. Emergent architectures are self-organizing and bottom-up structured. *Hybrid* architectures combine characteristics of both, symbolic and emergent architectures.

Prominent examples of symbolic architectures are SOAR, EPIC, ICARUS and NARS. Examples of emergent architectures are IPCA, Cortronics, NuPIC, and NOMAD. ACT-R, CLARION, LIDA, DUAL, Polyscheme, 4CAPS, Shruti, and Novamente can be regarded as hybrid architectures. For a description of these projects see, for instance, Duch, Oentaryo & Pasquier (2008).

ACT-R (Anderson, Bothell, Byrne, Douglass, Lebiere & Qin, 2004), as a member of hybrid cognitive architectures, processes its data with the help of different modules, for example, a module for visual data, a module for motor data (actions), a module for goals. In each processing cycle, production rules are matched against facts in short-term memory. The production rule which produces outcome, which is closest to ACT-R's goals wins.

In SOAR (Laird, Congdon, Coulter, Derbinsky & Xu, 2011), a member of symbolic cognitive architectures, the processing cycle selects operators which fit the current problem and lead to a state which is closer to a desired goal.

Furthermore, LIDA (Faghihi & Franklin, 2012) is a member of hybrid cognitive architectures. In LIDA the *cognitive cycle* activates modules to filter input data, to select actions, and to process the actions. Additionally, LIDA has several built in learning mechanisms.

In ICARUS (Langley, Choi & Trivedi, 2011), facts about the environment and objects are called *percepts* and *beliefs*, and rules are called *skills*. Skills are applied to percepts and beliefs in order to reach ICARUS's goals.

SiMA Approach

Rated according to the scheme sketched above the SiMA architecture is a hybrid one. It defines *three layers*, where the lowest one comprises the neural activities, i.e. the sensor and actuator activities (see Figure 1, the leftmost block). The second layer has to build neurosymbols from the neural input and in the other direction neural actuator signals from the symbolic results of the topmost layer, the psyche, which is understood as a symbol processing machine.

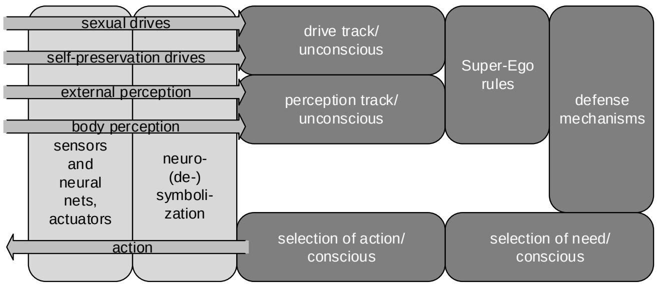

Figure 1: Overview of the SiMA model.

Some specific key features define the SiMA approach. First and foremost it is a *functional model*, i.e. it follows a generative approach with the focus on describing functions that generate behavior instead of building a behavior model. This enables a generic and flexible model. Another feature is the *layered description* of human information processing. The principle here is to use appropriate means of description for different aspects of a systems, e.g. the neuronal layer should be described with other means than the psychic layer. In developing such a model we use a *holistic and unified approach*, which considers a consistent and coherent description of all key aspects of human information processing. The consideration of these key features is only possible by following a *bionic* and hence interdisciplinary approach.

The impetus for SiMA was the challenge, to design a control unit able to cope with ambiguous situations, such as the security monitoring of an airport or the cooperation of a robot with human co-workers. The artificial system should have the same "*feeling*" for a situation as a human would have. The only way to gain this could be the bionic approach. So a holistic theory of the human mental apparatus as the control unit of the human body (Solms, 2009) was needed to work as the *blueprint* for the SiMA model. The basis was found in Freud's metapsychology. Freud came up with two major structuring concepts for the psyche, the first and the second topographical model. In the first model the distinction is made between the primary process, where data are handled totally unconsciously according to the pleasure principle, and the secondary process with preconscious and conscious, also rational, data treatment, where additionally the reality principle gets observed. From the point of view of computer science it clearly is a data model, while Freud's second topographical model is a function model. It distinguishes between the functions of the Id, the treatment of bodily needs, the Super-Ego, the demands from being a social creature, and the ego, which has to mediate between the other two. This abstract theory is concertized in the SiMA project as a basis, which is extended by contemporary theories from various disciplines, such as Damasio's (2003) theory of emotions.

Case-driven Agent-based Simulation

The challenges in capturing the functionality of the human mind in an interdisciplinary collaboration using computational simulations pose special requirements on the methodology in developing and evaluating the SiMA model. The question here is, how to translate assumptions about human mind's functioning from other disciplines in a deterministic and testable simulation model? In the SiMA project *case-driven agent-based simulation* (Schaat & Dietrich, 2014; Bruckner, Gelbard, Schaat et al., 2013) is developed. This methodology guides interdisciplinary collaboration in finding the required functions and data for a simulation model of the human mind. A combination and adaption of *casuistry, agent-based simulation and use-case driven requirements engineering* proved suitable to cope with the challenges of interdisciplinary collaboration and the evaluation of models of the human mind. Amongst others these challenges are the restricted accessibility of the human mind, interdisciplinary knowledge translation, the complexity in explaining and evaluating models of the human mind.

The first step in case-driven agent-based simulation is the analysis of the model requirements. Here, we use a casuistic approach, where the behavior and underlying assumed psychic processes in a concrete case, e.g. a hungry agent perceiving a food source and another agent, are described in a narrative way. But our experience in interdisciplinary collaboration showed that a textual concretization and structuration is needed to use such exemplary case as a basis for further development of a causal and deterministic model. Overall, the procedure of case-driven agent-based simulation consists of following steps (also see Figure 2).

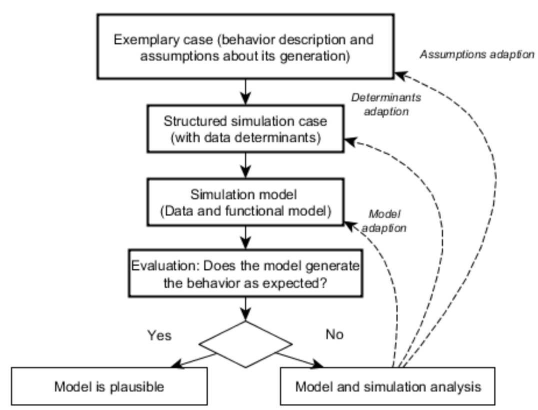

Figure 2: Case-driven agent-based simulation.

Exemplary Case

The exemplary case is a *narrative description* of a concrete case that *demonstrates assumptions* in an exemplary form, e.g. regarding motivations and decision making in a concrete internal and external situation.

The exemplary case primarily serves as a *platform* for interdisciplinary collaboration that facilitate the discussion between researchers, which often use different approaches and vocabularies. Hence, the usage of a concrete case supports *bridging the disciplines* and enhances the understanding.

The exemplary case at hand (called "Adam is hungry") describes a simplified gent-object interaction. The initial situation is given by hungry Adam, the agent with the SiMA architecture, Bodo, a passive agent, and a Viennese Schnitzel as a food source. The exemplary case describes abstractly how Adam's motivations, represented by drives, get in conflict with perception and social norms. And how mediating psychic processes finally decide his actions. In short, Adam is confronted with choosing to eat, share, give the Schnitzel or even beat Bodo. But under which external and internal conditions does he choose the respective alternative actions? A deterministic description is needed. Generally, to use the exemplary case as a point of departure for model development, some criteria must be considered. These are especially the *explication of assumptions* and requirements, and the consideration of a consistent and *deterministic description* with a concrete focus (e.g. motivation and decision making). Therefore the exemplary case is transformed into, structured *simulation case*.

Simulation Case

A focus in analyzing and transforming the exemplary case into a simulation case is an analysis of the data that determine the agent's behavior. Following a functional approach we also focus on how a change in these data determinants would lead to a behavioral change. We distinguish *four groups of determinants*: the agent's experience, personality factors (as simplifications of memories and body functionalities), the environmental state, and the agent's initial internal state (given by drives and emotions).

The simulation case for the described exemplary case is sketched in Figure 3, with the standard scenario of eating the Schnitzel, and the alternative scenarios of beating Bodo, and sharing or giving the Schnitzel to Bodo. As sketched, the personality factor "neutralized intensity", which indicates the strength of the defense and secondary process (see below), plays a key role in the selection of the scenario.

The transformation into a structured simulation case follows use-case-based requirements analysis in software engineering. Data determinants represent pre-conditions, the description of an agent's final internal state and selected action represent post-conditions. For the standard scenario, eat, the inner processes that generate the post-conditions from these pre-conditions are described step-by-step. We also have to track and justify every possible behavior of the exemplary case (e.g. share, beat). That is, for the alternative scenarios we only describe how the change of data determinants would lead to an alternative behavior.

Overall, this structuration enables a fine-grained requirements analysis, the development of a causal model and its evaluation.

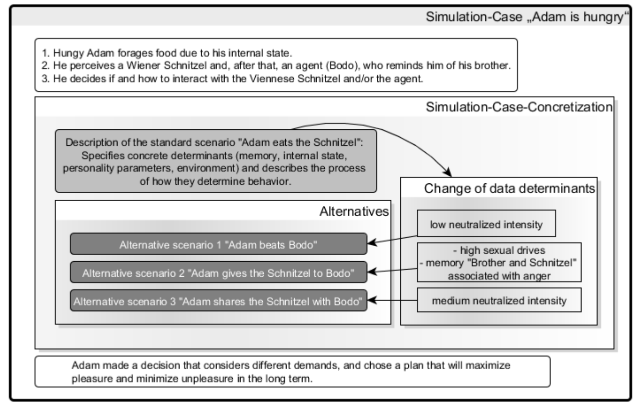

Figure 3: Simulation case "Adam is hungry".

Evaluation

After developing the model (see sections below for an overview) it is tested using the *simulation case as a test plan*. In particular we parameterize the simulation according to the scenario's data determinants and observe if the functions generate and data determine behavior as expected (see Chapter Calibration and Simulation). We do not only validate the behavior, but also how the behavior is generated and determined, e.g. how emotions and drives evolve and influence the agent's decision. If the agent behaves unexpected or the data visualization indicate wrong assumptions, based on an analysis on different levels, we have to conduct another iteration of the procedure (see f*eedback cycles a, b, and c* in Figure 2). Possibility *a* and *b* indicate that the inputs form other disciplines (psychoanalysis, neuroscience) are interpreted and transformed wrongly or implicit requirements emerge during implementation (implementing a model helps to understand and specify it). Possibility c may be caused by inconsistencies in an underlying theory or between different theories. This feedback helps to sharpen theories from other disciplines precisely.

This evaluation methodology enables us to test our *model's predictability and plausibility*; in particular, the validity of the case's assumptions and if the specified data determine the expected behavior (change).

Primary Process

In SiMA, the primary process represents unconscious data processing. It is characterized through fast and immediate processing of data that is close to sensor values. Its logic can be well described by the rules that apply on associations between data structures. There are two rules of the forming of associations: similarity and simultaneousness. This means that things that are similar are likely to form associations as well as objects that occur at the same time or within a short time frame.

The inputs of the primary process are defined by the homeostasis of the body, the body perception and the external perception. Homeostatic values are symbolized into *drive tensions*, which are a mean for intensity of bodily need. In the *Drive Track* in Figure 1, *drives* are created from the drive tension and extended with a drive object and a drive aim. The *drive object* is the external object, which is able to satisfy a drive and the *drive aim* is the action that has to be taken to satisfy it.

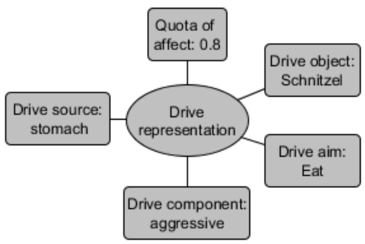

Figure 4: Drive representation.

External perception and body perception are symbolized and define the input of the *Perception Track* in Figure 1. Here, based on perceived features, internal representation (so-called percepts) of objects are inferred. Through the property of simultaneousness, these objects form a *perceived image* that represents the current situation. Through the property of similarity, similar stored situations as *stored images* are then activated. In the stored images, memorized emotions are associated that reminds the system of a certain emotional state. Together with the drives, they generate the current emotions (Schaat, 2013) of the system that will be used later on in decision making.

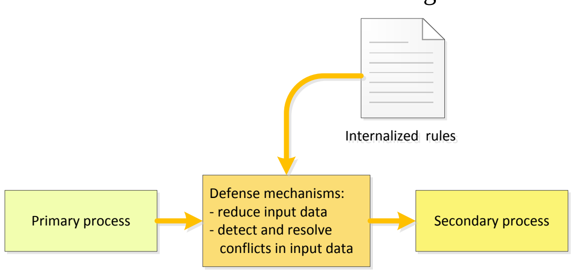

Figure 5: Overview of defense mechanisms' function.

At the verge of the primary process to the secondary process are the defense mechanisms located (see Figure 5). Defense mechanisms are a kind of filter mechanism. The two tasks of the defense mechanisms are, firstly, to reduce the data which flow from the primary process to the secondary process and, secondly, to detect and resolve conflicts in input data. In order to process the first task (data reduction), the data are assessed by emotions and the focus of attention is set on specific data with a high level of activation. To process the latter task, to detect a conflict, the defense mechanisms have access to an internalized rule base, the *Super-Ego-Rules*. And in order to, eventually, resolve a conflict, the defense mechanisms can repress input data or alter them before the defense mechanisms pass them on to the secondary process. Which defense mechanism is chosen, depends on personality factor "conflict tension" (severeness of a conflict) and the stage of development of the personality of the software agent.

Secondary Process

The secondary process is responsible for the preconscious/conscious processing of data. Its main task is to take a decision about an action based on the inputs from the primary process. However, different to the primary process, more extensive associations of data structures are possible. Data structures are extended with a word, making it possible to communicate the information to a received outside of the system. Also, temporal and hierarchical associations may be used, making it possible to order things. At the beginning of the secondary process, activated stored images, which were independent in the primary process are formed into sequences called *acts*. Acts define events and the actions necessary to be taken to get from an event to another.

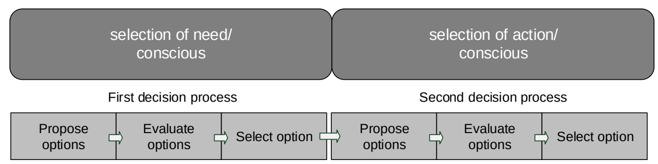

Figure 6: Decision making in the secondary process.

Decision making in SiMA can be divided into two stages as seen in Figure 6. In each stage, similar process steps are taken: first to limit the number of options, considering all available options. Second, the options passing the first stage have access to more system resources and one of them is finally selected.

The first step in the decision making process is to extract the possible options that the system can develop and act on. It is the start of the *Selection of Need* track of Figure 1 and Figure 6. This is done through the creation of *possible goals* ("propose options" in Figure 6) from the acts or from perception. Drives from the primary process become *drive wishes*, which are one of the motivations to do something in the system. They define the desired external object, the preferred action and the importance to reach it. Emotions are transformed into feelings that can also be used to emphasize or to avoid certain situations.

After a general initialization with a basic effort analysis, the possible goals are evaluated regarding the possibility to fulfill a certain drive wish, an emotional state, and social rules (evaluate goals" in Figure 6). Based on the available system resources one or more possible goals are selected for further processing ("select option" in Figure 6).

The selected possible goals are the options that the system has at the moment. In the *Selection of Action* track of Figure 1 and Figure 6, possible action plans are generated and evaluated for each of them. Then, one option is selected and executed.

Decision making of the secondary process is a deliberative process in contrast to the primary process.That is, the options of the system can be processed during multiple cycles without any external actions. The system can reason about several options in sequence before taking a decision. Internal actions are used to perform analysis of options and to execute queries to the knowledge base that modify the internal state of the possible goal.

Calibration and Simulation

The modules in SiMA encapsulate functionalities of the human mind and are developed independently, following a black-box approach. Meaningful integration tests for these modules require a level of knowledge about module interaction, which is not available, due to the high number of modules, parameterization options and their functional structure. Therefore we keep integration testing to a minimum in favor of system testing, using exemplary cases. Calibration is performed in various steps on each scenario. First the environmental situation (Adam, Bodo, and the Schnitzel) is modelled as the most basic layer of calibration. Next, the drive situation is modelled and memories are created to match memorized actions to drives, according to the simulation case description. Where needed, the defense mechanisms are modelled and harmonized with the drive situation. Lastly, the acts are modelled and associated to the memorized actions. Each step could, and often did, require previous steps to recalibrate to allow modeling according to the description. This resulted in a calibration strategy similar to a waterfall model with feedback.

Simulation Results

As mentioned, we validate our model via test scenarios in the MASON simulation framework. This chapter summarizes the results of these simulations. We compare the agent's behavior and internal state to the expectations defined in the simulation case. The internal state of the agent is checked via data visualizations.

The simulation scenarios are designed to show the capabilities and impacts of the functional modules. Exemplary case 1 is focused on the primary process, specifically the interaction between perception, drive state and defense mechanisms. The secondary process focusses on following the memorized action sequences. In each scenario, the agent can choose between four plans: EAT, BEAT, GIVE and SHARE[2]. The initial environmental situation is also shared among scenarios. The blue lines indicate sight ranges, the green Agent will be referred to as Adam, the red agent as Bodo and the round shape between them as Viennese Schnitzel.

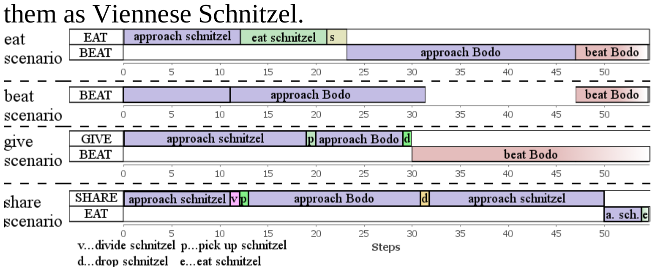

Figure 8: Behavior sequences in the simulation case.

[2] Plans are written in capital letters to distinct them from actions.

Standard scenario eat

The first chart in Figure 8 describes the behavior of Adam during the standard scenario (eat the Schnitzel). The first column shows Adam's current plan. The combination of high hunger drive (see Figure), the perception of a Schnitzel and the memorized satisfaction for eating Schnitzel, make Adam initially follow the plan to EAT. After the Schnitzel was consumed, Adam switches to the BEAT plan, as it fits the new perception (Bodo; no Schnitzel) and new drive state. Figure visualizes the changes in Adams drive state in detail. Adam starts with high hunger. While he eats, the hunger drops, since eating the Schnitzel changes Adam's body state which the drives represent. In time, the hunger subsided below the sexual drives, which started out low but steadily increase. The stamina drives (Figure 9 in blue and cyan) represent Adam's need for relaxation and changed in response to Adams exhaustion while approaching the Schnitzel (first two peaks) and Bodo (third peak).

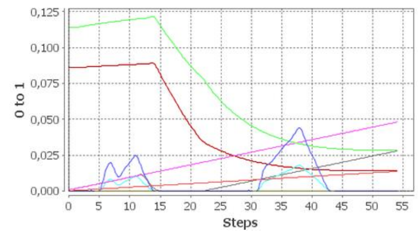

Figure 9: Drives: hunger in green (aggressive) and red (libidinous).

Alternative scenario beat

This scenario differs from the standard scenario in its initial drive state. Adam starts with higher, faster increasing sexual drives and low hunger. The BEAT plan is memorized with the highest satisfaction for the sexual drives and is associated with the current emotional state (see Figure 10). Beating reduces the anger and causes a short peak of joy.

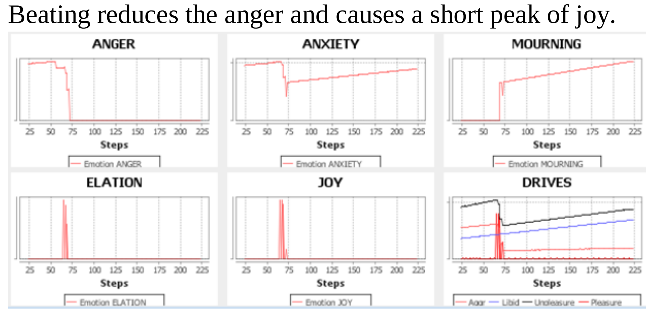

Figure 10: Emotional state in the beating scenario.

Alternative scenarios give and share

These scenarios use a defense mechanism to alter Adam's behavior away from the current drive demands. This is achieved by the drive mechanism "sublimation", which changes the valuations of the possible actions associated to the hunger drive, away from their memorized satisfaction values. Due to their similarities they are discussed together. The third chart in Figure 8 shows Adam's behavior during the give scenario and the fourth chart shows Adam's behavior during the share scenario.

In both scenarios, the drive situation is similar to the standard scenario, with the hunger drives dominating. A super-ego rule is introduced to create a conflict between high hunger drives and perceiving Bodo with the Schnitzel.

The conflict is resolved by subliming the action eat, which is chosen due to its high memorized satisfaction for the hunger drive (see description of standard scenario eat), with the action give or share. In both cases Adam will follow the corresponding plan, as the sublimed action now promises the highest satisfaction for the hunger drive.

Lessons learned

The calibration of drives makes for a good point of departure for calibrating the SiMA model. One possible problem at this stage is that the drive situation also influences the perception. Calibrating the perceived reality is normally done straight forward by directly specifying the simulated world, but extreme drive states influence the selection of memories for perception, since the agent also activates memories that are that satisfy current drives. After modelling the base behavior, the defense mechanism can be used to fine tune the agent. They can alter the influence other primary process modules have on the decision. In the current simulations, the defense was only used to model alternative behavior in situations that match certain criteria (e.g. sharing when Bodo is next to the food source). More elaborated uses of this tool are thinkable as the defense is capable of influencing the impact other primary process modules have on the decision (e.g. fine tuning the drive state via super-ego rules). This ability to influence, and even alter, the results of other modules, also makes the defense harder to calibrate, since changes in this module may affect many others and vice versa.

Conclusion and Future Work

We showed how a narrative exemplary case that exemplifies assumptions and requirements for a cognitive architecture is structured into a deterministic description and test plan. The resulting model is evaluated in simulations, where we showed how the interplay of various decision factors, such as drives, emotions, social norms, results in an adapted decision for an agent's internal and external state. Using a functional model we emphasized on how behavioral change is generated without changing the model.

The presented evaluation using simulation cases can be regarded as a calibration that validates the transformation of the SiMA model into a software implementation. However, such calibration is only a first step to demonstrate the model's consistency and plausibility, with the next step being to observe if unspecified parameters would generate comprehensible results without model adaptions.

In the end such methodology is a premise to apply the model in a specific domain, e.g. as a decision support tool for marketing strategies (as currently approached), where we will additionally test our model against empirical data.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). *An integrated theory of the mind*. Psychological Review, 111(4):1036–1060.

Bargh, J.A., & Chartrand, T.L., (1999). *The unbearable automaticity of being*. American psychologist, 54(7): 462-479.

Braitenberg, V., (1984). *Vehicles: Experiments in synthetic psychology*, MIT Press, Cambridge.

Bruckner, D., Gelbard, G., Schaat, S., & Wendt, A. (2013). *Validation of cognitive architectures by use cases*. Proceedings of Industrial Electronics (ISIE), IEEE International Symposium on.

Damasio, A.R., (2003). *Looking for Spinoza: Joy, Sorrow, and the Feeling Brain*. Harvest Books.

Duch, W., Oentaryo, R. J., & Pasquier, M. (2008). *Cognitive Architectures: Where do we go from here?*, Frontiers in Artifcial Intelligence and Applications, vol 171, pp. 122-136.

Dietrich, D., et al. (2014). *Naturwissenschaftliches, psychoanalytisches Modell der Psyche*. Technical Report II, Vienna University of Technology, Institute of Computer Technology, Austria.

Faghihi, U., & Franklin, S. (2012). *The Lida model as a foundational architecture for AGI*. In Theoretical Foundations of Artificial General Intelligence (pp. 103–121). Springer.

Laird, J. E., Congdon, C. B., Coulter, K. J., Derbinsky, N., & Xu, J. (2011). *The Soar user's manual version 9.3.1*. Technical report. Computer Science and Engineering Department, University of Michigan.

Langley, P., Choi, D., & Trivedi, N. (2011). *Icarus user's manual*. Technical report, Institute for the Study of Learning and Expertise 2164 Staunton Court, Palo Alto.

Langley, P., Laird, J. E., & Rogers, S. (2009). *Cognitive Architectures: Research Issues and Challenges*, Cognitive Systems, 10, 141-160.

Schaat, S., Doblhammer, K., Wendt, A., Gelbard, F., Herret, L., & Bruckner, D., (2013). *A Psychoanalytically-Inspired Motivational and Emotional System for Autonomous Agents.* Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society.

Schaat, S., Wendt, A., Jakubec, M., et al. (2014). *ARS: An AGI Agent Architecture*. In B. Goertzel, L. Orseau, J. Snaider (Eds.). Artificial General Intelligence. Proceedings of the 7th International Conference, AGI (pp. 155-164). New York, USA: Springer.

Schaat, S., & Dietrich, D., (2014). *Case-Driven Agent-Based Simulation for the Development and Evaluation of Cognitive Architectures.* Proceedings of the 26th Benelux Conference on Artificial Intelligence (pp. 73-80).

Solms, M. (2009). *What is the 'Mind'? A Neuro-Psychoanalytical Approach*. In Dietrich D., Fodor G., Zucker G., & Bruckner D. (Eds.). Simulating the Mind (pp. 115-122). Wien, Austria: Springer.

Vernon, D., von Hofsten, C., & Fadiga, L. (2011). *A Roadmap for Cognitive Development in Humanoid Robots*. Springer, Berlin.