Decision-Making in the Cognitive Architecture SiMA

Alexander Wendt, Friedrich Gelbard, Martin Fittner, Samer Schaat, Matthias Jakubec, Christian Brandstätter, Stefan Kollmann
TU Wien, Institute for Computer Technology, Gusshausstrasse 27-29
1040 Vienna, Austria
{wendt, gelbard, fittner, schaat, jakubec, brandstaetter, kollmann}@ict.tuwien.ac.at

***Abstract*—In a cognitive architecture, decision-making is the task that processes information from sensor data and stored knowledge to get appropriate action plans and actuator commands. Its aim is to make a decision in a given situation based upon available options and current goals of the system. In this paper, the decision-making process of the cognitive architecture SiMA is presented. Its unique features are the comprehensive evaluation of options, an application of case-based reasoning, as well as the management of resources by a two-step decision-making process. The implementation is verified through an artificial world implementation of a use case.**

Keywords—cognitive architecture; decision-making; artificial intelligence; case-based reasoning; symbolic computational modeling; causal reasoning; SiMA; ARS;

I. Introduction

The cognitive architecture SiMA (formerly named ARS) [1] stands for "Simulation of Mental Apparatus & Applications". It relies on a holistic functional model of the human mind based on the theory of psychoanalysis. The model conforms to the wide, and generic approach to Artificial General Intelligence (AGI) projects with the same focus on a solution to control complex systems without human interaction.

There is a steadily increasing demand for control systems that can operate in complex environments. Such environments are the AI players in computer games [2], agents in economic agent simulations [3], but also building automation systems [4]. In the project ECABA, SiMA will be adapted to solve tasks that can hardly be solved in a satisfying way by building automation systems of today. There, goals are often ambiguous and contradictory, e.g. to maintain comfort and at the same time reduce energy consumption. Hence, with model predictive control, a system can be perfectly optimized for that model. By using a cognitive architecture, functionality is added to compare different options and to adapt the system at runtime by making use of previous experiences.

The architecture SiMA has been described in several previous papers. However, in this paper, a detailed description of the newly developed decision-making is presented for the first time.

First, a brief description of the general SiMA architecture is presented. Then, SiMA is related to other cognitive architectures. The main part of the paper is to step through the decision cycle by using a simple use case from an artificial world. The execution of that use case finally shows the results of this work.

II. The Cognitive Architecture SiMA

The model is based on two foundations of psychoanalysis [5]. The first principle is resolving conflicts between the motivations (desires) of the system and internal rules. Internal rules represent policies or social behavior. The conflicts have to be resolved, to stay capable of acting. For instance, in building automation, a rule says that comfort must be maintained and at the same time, the system has a desire to lower energy consumption.

The second principle is the separation of functionalities that process unconscious and conscious data. Associations of data in the unconscious processing (upper part of Fig. 1) rely on the similarity of data or simultaneousness of data. Therefore, only simple, but quick comparisons are made. Conscious processing of data is similar to logical or relational data processing in many other architectures (lower part of Fig. 1).

A vital part of the architecture is embodiment. The body defines the desires called *drives* of the system. Therefore, body information is represented by drives (upper left of Fig. 1). The main task of the drive track is to create the drives of the system. A drive consists of an importance, an object, and a default action to satisfy the drive that is executed on the object. Later in decision-making, the drives are the foundation to generate system goals.

Perceived external objects are represented as *percepts* (below the drives of Fig. 1), which then define the *perceived state*. It captures the whole situation.

In a process, which can be called *psychic spreading activation* [6], *memorized states* are activated according to the rules of unconscious data. They are similar to the perceived state. Therefore, they are relevant to the current situation. The memorized states will be the base for the generation of options to the system.

The drives, the perceived state, and activated memorized states are tested by rules (system policies) in the "Super-Ego Rules track" and "Defense Track" (upper right part in Fig 1.). Additionally, emotions are created as a reaction to the applied rules, the drives, and the memorized states. An *emotion* is an evaluation of the internal state of the system.

978-1-4673-9606-6/15/$31.00 ©2015 IEEE

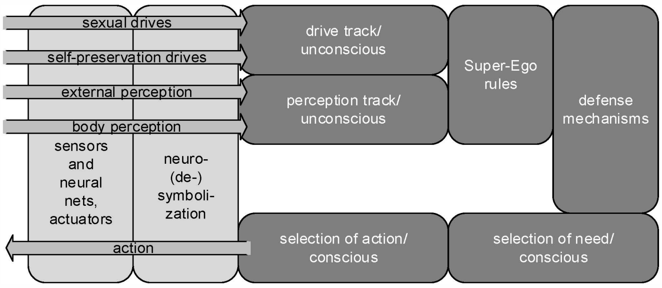

Fig. 1. General view of the SiMA model

Decision-making, which is the main topic of this article, takes place in the "Selection of Need" and "Selection of Action" (lower part of Fig. 1). Here, sequences of memorized states define *episodes*. Episodes can be seen as cases of case-based reasoning [7]. They describe experiences and especially consequences of actions, which are useful for the agent. From the episodes that were generated from activated memorized states, the *options* for action plans of the system are generated. On the other hand, the drives of the system determine the *goals*. The options tell the system what it can do, and the drives tell the system what it wants to do.

Decision-making is executed in two phases. First, the decision is made which options can satisfy the most immediate goals. The purpose is to focus the system resources on only the important options. Then, those options that were selected are analyzed regarding planning. The result of the planning is a possible action. After evaluation of the proposed actions of the options, one option is selected, and the corresponding action is executed. In this paper, the decision-making will be described in detail.

III. Related Work

Every cognitive architecture has its way of implementing decision-making. An overview of cognitive architectures is given in [8], [9] and [10]. Soar [11] and ACT-R [12] have a decision-making process, which is based on rules that are continuously tested on the system inputs. While, in ACT-R, the rules are directly applied to the system state, in Soar, rules only propose operators, which are then applied. Additionally, Soar has a fixed decision cycle, where rules are used to evaluate decisions, to search for long-term memories and to perform sub-goaling. In ICARUS [13], skill hierarchies are extensive to match deducted beliefs. BDI [14] tries to reach desires (goals) by applying beliefs (rules) to percepts, similar to ICARUS. The difference between BDI architectures and SOAR, ICARUS and ACT-R is that BDI is not focused on problem-solving. Instead, behavior templates are matched to perceived situations. In LIDA [15], during the cognitive cycle, attention codelets filter input data by assigning them a certain amount of attention. It is done by forming coalitions with the input of the workspace. The decision is based on the winning coalition with the highest activation. Its content is broadcast to allocate resources for planning.

SiMA can be related to the other architectures. From similarity, LIDA is the most similar because LIDA also models the human mind and with the conscious broadcast, LIDA has similarities to the two-step decision-making process of SiMA. The topics, where SiMA differs from the other architectures is its strong focus on unconscious processes, which keep a policy-like system called "defense mechanisms" for fast, but "second-best" solutions to problems [16]. Further, compared to the architectures above, at the moment a purely case-based decision-making is implemented and not a heuristic search through rules. Finally, SiMA is based on psychoanalysis. There are doubts about whether it is a valid approach or not. An argument for a valid approach is that SiMA has defined most of the components of other architectures e.g. an attentional mechanism. These concepts were directly derived from the theory.

IV. Use Case Description

The desciption of the decision-making is supported by a simple use case. It shall be used to clarify the concepts presented and at the same time, it is a part of the results. In Fig. 2, illustration 1, three objects are present: An agent at the bottom of the illustration controlled by SiMA decision-making, a food source for the top and a stone between the agent and the food source. The agent has a simulated body with homeostasis much like our own.

At the start of the scenario, the main goal of the agent is to find a food source to eat from, i.e. to satisfy the "hunger"-drive. There are other drives as well as a "relax"-drive and "deposit"-drive to excrement consumed food.

The food source is visible to the agent. However, there is a stone between the agent and the food source. As long as the "hunger"-drive is the most important drive, the task is to get to the food source and to eat it.

External perception and available, memorized episodes are used to generate options. Each state of an episode has an associated action that will bring the agent to the next memorized state if everything goes as planned. In this case, there is an episode that tells the system how to go around the stone, to reach the food source. If there were no episode available, the agent would act on instinct and try to go through the stone.

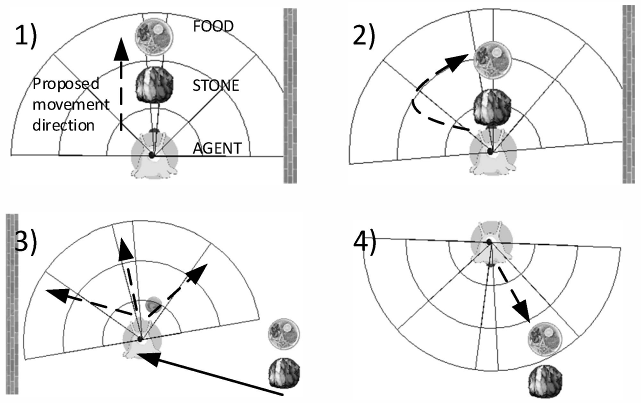

Fig. 2. Four screenshots from the simulator with the agent following different decision paths

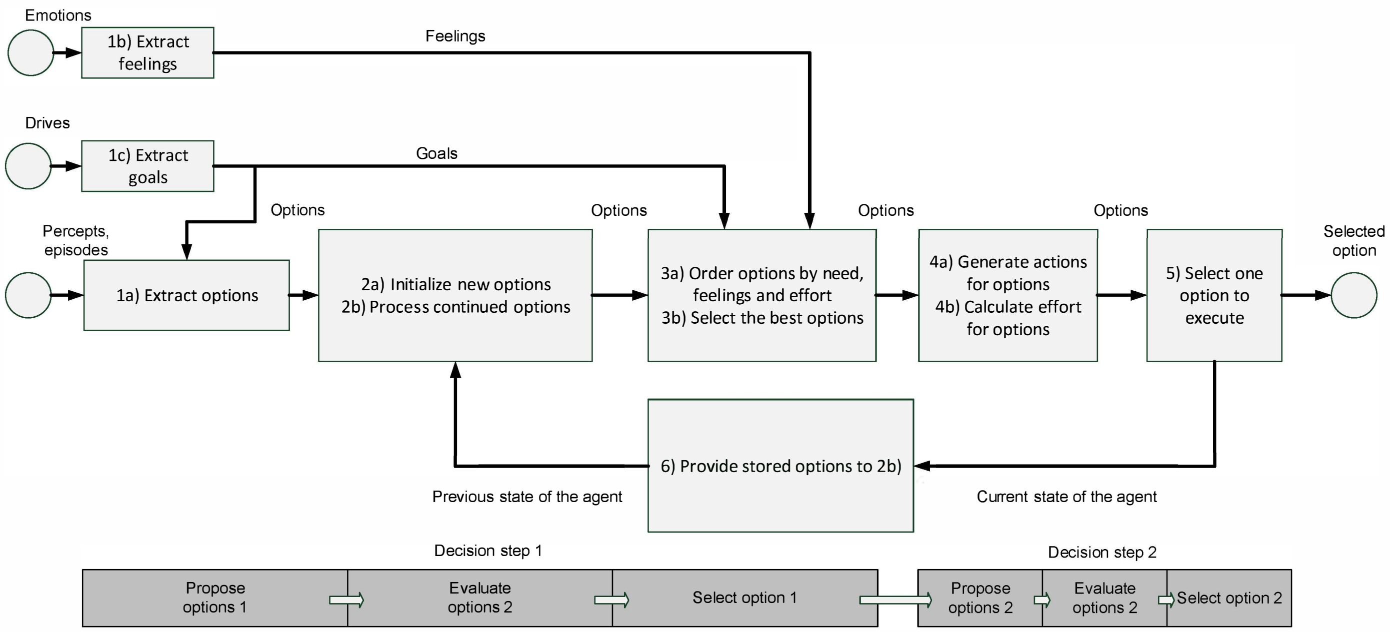

Fig. 3. Decision-making process of SiMA

V. DECISION-MAKING PROCESS

In Fig. 3, one cognitive cycle of the decision-making is described. It is only a part of the complete SiMA architecture. At the output of the cognitive cycle, an action is available.

However, in deliberative decision-making, this action must not be an external action that is immediately performed in the environment. Instead, it can be an internal action that only modifies the internal state of the system. An example of an internal action is to "think" about something, which means that the software will perform some analysis of an option. Therefore, an external action can be executed after several internal actions have been executed.

In Table 1, the three internal actions "CALL_MEMORY", "ACT_ANALYSIS" and "FOCUS_MOVEMENT" are executed before the external action "TURN_LEFT" is finally executed. Because the option originates from an episode ("ACT_DRIVE"), in this case first the memory is recalled for supplementary information, and then the episode is analyzed and then the agent sets is an internal focus on the perceived area in front of him, to notice possible obstacles. Now, the agent is ready to execute the external action and to turn left. For each type of option, the system can use different processes to get to external actions. The different types of options will be described later on.

An option that originates from the perceived state ("PERCEPTDRIVE") does not need any internal action "CALL_MEMORY" and therefore, this process looks different.

The handling of options is done according to teleo-reactive planning [17]. Each type of option has a state machine, which handles that particular option until the system goal has been reached or the option is no longer interesting for the system. The system must, therefore, keep track of the state of the option. If an option has reached a certain state, then the next state is pursued. States are reached by executing internal actions. If an internal action fails or if the next state cannot be achieved, the option remains in the latest valid state. An example can be seen in Table 1, where the state after "CALL_MEMORY" is a state that will make the system to perform an analysis of the option by executing the internal action "ACT_ANALYSIS".

In the example in Table 1 ("cycle" 13-16), the state machine of the option consists of four states. First, the three internal actions and then an external action. Because the external action changes the state of the environment, the state of the option is reset, and the process starts new. In the current implementation, the state machine has been manually defined. In a future system, it would be better if the state machine is learned and saved as procedural knowledge.

Because of the usage of options, they contain meta-data about themselves. As mentioned in the previous section, each option is processed in a state machine. Therefore each option has to know, which process steps have been executed, the results of them, for the system to know what to do next with that option. The meta-data also consider results of certain analyzes.

Previously memorized states were activated through a spreading activation in the "Perception track" of Fig. 1. The memorized states are fragments of episodes, which are relevant to the current perceived state. They form the main source of options of what the system can do. Besides of the episodes, options are generated from the instincts, which are predefined actions based on the perceived state.

TABLE I. ORIGINS OF SELECTED OPTIONS AND EXECUTED ACTIONS

Cycle	Goal Source	Action
11	PERCEPTDRIVE	FOCUS_ON_GOAL
12	PERCEPTDRIVE	FOCUS_MOVEMENT
13	**ACTDRIVE**	**CALL_MEMORY**
14	**ACTDRIVE**	**ACT_ANALYSIS**
15	**ACTDRIVE**	**FOCUS_MOVEMENT**
16	**ACTDRIVE**	**TURN_LEFT**
17	ACTDRIVE	ACT_ANALYSIS

The third source of options is a search behavior based on the drives. In Table 1, the sources of options are labeled with “PERCEPTDRIVE” for instinct options from perception and “ACTDRIVE” for options from episodes.

The inputs of decision-making are the following: Emotions, drives, the perceived state activated episodes. It is to be noted that episodes consist the activated memorized states. For each memorized state, its episode is activated as input to decision-making.

In the following, the decision-making process is played through step by step. The process is illustrated in In Fig. 3.

A. Propose Options 1 in the First Cycle

The first step of the two-step process is to prepare the input data for further usage in decision-making. First, in 1a) of Fig. 3 options are extracted from the episodes, the perceived state, and the drives.

An episode consists of memorized states. Meta-data are associated with an episode about well a certain percept in a memorized state of an episode can satisfy a certain drive. For instance, in Fig. 4, a “Cake” satisfies the “hunger”-drive. The cake may also satisfy other drives like the “bite”-drive. Therefore, from the meta-data of the episode, several options are extracted. However, each option is still associated with the original episode. The episode is necessary for later process steps to extract the information on how the drive is satisfied.

Further, the option contains information of a scale [0, 1] how well the drive was satisfied the time as the episode was recorded. It can be seen as the system reward. While a “Cake” satisfies a drive perfectly with 1.0, “broccoli” may only satisfy this drive to a range of 0.5.

Just as options are extracted from episodes, options are extracted in a similar way from the perceived state. The perceived state is also associated with meta-data about which percepts can satisfy certain drives.

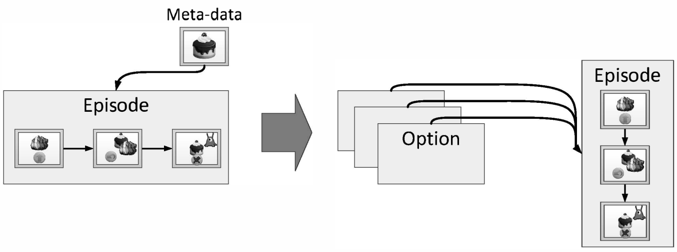

Fig. 4. Extraction of options from an episode

Options are also directly extracted from the drives. The purpose is to provide the agent with options for percepts, which are neither present in the perceived state or in the episodes. The agent shall have the possibility to search for those percepts.

In the episodes, besides of the meta-data about drive satisfaction, also the emotional state of the agent is stored as a memorized feeling. If there were a hostile agent in the use case, there would be an episode that tells the agent about an experience with the hostile agent. The internal state of the agent would be stored as a *memorized feeling*. Options that originates from feelings tell the agent to react to external stimuli. This is a major difference to drives, which only fulfill bodily needs and cannot be used to avoid something that has been perceived.

In this use case, three options are extracted, and their applications are shown in Fig. 2. First, in “2)”, an option is extracted from an episode about how to go around a stone. Second in “3)” an option is extracted from a drive, in order to make it possible to search for the desired percept. Third in “4)”, an option is extracted from the percept “Cake”, which would fulfill the “hunger”-drive.

In 1b) of Fig. 3, the current state of feelings is extracted from the emotions. Feelings are a description of the current mood and influence the selection of options. Each feeling stands for a certain preference behavior. For instance, if the system has the feeling “anxiety”, the preferred behavior is to exploit known solutions. On the other side, if the system has the feeling “confidence”, behaviors are favored that prefer exploratory actions.

In 1c) of Fig. 3, system goals are extracted from drives. The need to consume a food source would become the wish to reach it.

B. Evaluate Options 1 in the First Cycle

In step 2a) of Fig. 3, the previously extracted options are initialized. It means that they are set to an initial state of the state machine.

Options also have an importance to the system. The importance mainly consists of the reward in terms of the fulfillment of a drive or acting according to a feeling and the effort to fulfill the purpose of the option. The effort can be counted negatively. The sum of those components provides the overall importance. However, a first evaluation is performed at this stage, in order to exclude non-relevant options. For instance, percepts like a “Cake”, which are desirable to fulfill drives, the effort is higher to reach it if the “Cake” is far away than if it is near. Another example is the options that were extracted from drives. The effort is higher to search for something, where its location is not known than to go to something that is in the perceived state. In order not to waste resources on unreachable options, a simple effort analysis is performed for each new option.

Step 2b) will be described later in the next cognitive cycle, as it handles options, which have already been processed once and are not new to decision-making.

C. Select Option 1 in the First Cycle

In step 3a) of Fig. 3, the first selection starts. All options are evaluated and sorted according to the following criteria: the possibility to fulfill goals (from drives), the response to the current state of feelings and effort to reach it.

It is clear that the more important the drives are, the more important are the options that offer fulfillment of those drives.

The current state of feelings plays a role in the selection of options. Situations that require a reaction to external stimuli will boost the importance of those options, which can fulfill that purpose. For instance, if the agent feels “anxiety”, options that bring the agent away from that situation are favored. In the use case of this paper, the option that takes the agent around the stone is favored because the last time the agent did that, the solution was successful and the agent felt joy.

One or more of the options with the highest importance are selected in 3b) of Fig. 3. The more system resources available, the more options can be processed during one cycle regarding possible actions. If the resources are strongly limited, only one option is selected. If there are plenty of resources, several options can be selected for deeper analysis in the planning and generation of actions.

D. Propose Options 2 in the First Cycle

Similar to LIDA [15], the selected option(s) have access to the resources of the planning in step 4a) of Fig. 3. From here, the options are more comprehensively analyzed. First, it has to be found out what to do with the options. This is dependent on the current state of the meta-data of the options. It means that the state of the option is checked, in order to decide the next step according to the state machine for this option type. Here, an internal or external action is proposed that will alter the state of the option.

E. Evaluate Options 2 in the First Cycle

In 4b) of Fig. 3, all proposed actions are then evaluated regarding their effort to execute and in detail how certain actions could satisfy a particular drive. In the case of an agent that shall reach a food source, which is located behind a stone, the effort of walking through the stone is much higher than going around it.

F. Select Option 2 in the First Cycle

In in 5) a ranking by importance is done. In the use case, the act of going around the stone is selected to be executed. However, it may be the case that after evaluation of the proposed actions, none of the options are good enough to fulfill. This fact is first known after the analysis of an option and not at the time an option is selected in the first step of the decision process. Therefore, there is the possibility to define that an option, which action shall be executed by the system needs at least a particular importance to be selected. If none of the options are worth fulfilling, then other options have to be taken into account. In that way, the selection of the best option may take several cognitive cycles.

The action of the one selected option is then executed. This option and the other options that did pass the first selection mechanism (“Select Option 1”) are saved in a short-term memory in step 6) of Fig 3. Then, they and especially, their states are available in the next cognitive cycle.

In case that the proposed action is an external action, it is performed in the environment. Otherwise, the internal action will be executed in the system first in the succeeding cognitive cycle.

G. Propose Options 1 in the Second Cycle

In the succeeding cognitive cycle in step 2b) of Fig 3, the content of the short-term memory is merged with incoming new possible goals that were described earlier. At this stage, some options have been processed further in their state machines. They are updated with the information of the new options that have been generated in step 2a).

The internal action of the previously selected option is executed in step 2b). For instance, in the use case, if the internal action was “ACT_ANALYSIS”, the service that offers such an analysis is started and processes the episode of the option. In that way, the internal state of the option is changed. The “episode-analysis”-service demands some system resources and are therefore only applied to the selected option.

H. Further Cognitive Cycles

As long as the previously selected option is important enough compared to other options, it will be processed in the state machine until its purpose has been fulfilled. Because the “option selection 1” allows more than one option to be stored in the short-term memory, the execution of options can be interrupted and resumed, if more important options have to be fulfilled. It allows the system to consider new external influence in each cognitive cycle as new options are always evaluated together with the currently processed options.

VI. Results

Decision-making of the SiMA agent was implemented in JAVA as an agent in the MASON framework [5]. The use case described is very simple, yet it demonstrates the basic functionality of decision-making in SiMA. The use case itself is a part of the results. Table 1 is a part of the simulator results, and it shows which internal actions are necessary to execute, in order to perform an external action.

In Fig. 2, illustration 2), as the agent gets to the stone, “cycle” 13 is reached in Table 1. Because the content of an episode perfectly matches the perceived state, the option to satisfy the “hunger”-drive is selected for this episode (“ACTDRIVE” in Table 1). The episode tells the agent how to get around the stone. In illustration 3) of Fig. 2, the agent was manually put somewhere else on the map. Now, no options were available from the perceived state or from any episode. Instead, the selected option originated from the “hunger”-drive. It triggered a search behavior. In illustration 4) of Fig. 2, after searching a while, the “Cake” is visible once again. From there, the option that originates from the perceived state (“PERCEPTDRIVE” in Table 1) is selected, because it is better to go to a “Cake” in the vision than to search for another one.

The shown use case was a very simple one, but the purpose was to demonstrate the decision-making process. More impressing are the results of [1], [5] and [16], which all rely on this decision-making functionality.

VII. Conclusion

The purpose of the project SiMA is to create a model of the human mental apparatus. It incorporates a strong focus on unconscious processes like the generation of drives and evaluation of percepts by emotions. The pre-conscious process of decision-making has been described in this paper. It uses a two-step, multi-cycle decision-making first to filter relevant from non-relevant options. Then, those options are evaluated in detail. Finally, one option is selected. The action of this option is executed.

There are external as well as internal actions. While the external actions alter the environment, the internal actions alter the state of the selected option. As compared to reactive decision-making systems, a deliberative decision-making allows the system to "think" by not executing external actions in each cognitive cycle. From the perspective of software engineering, internal actions offer a proper way of accessing services, e.g. certain algorithms that are added at run-time.

For each type of option, a state machine is defined on how to handle them over several cognitive cycles. These state machines define which actions can follow a certain state. At the moment, options have three sources. Each of them uses an own state machine that has been predefined. In the future, the composition of services and actions could be learned instead and stored in an ontology. In that way, the state machines could be based on the context of the episodes. The main means of reasoning is case-based reasoning at the moment.

Currently, options as solutions for achieving system goals are handled in a forward-chaining manner. It means that options are extracted from a current situation and that an action is executed in the direction of the goal state. Future work should define concepts for backward-chaining, too, where the starting point is the goal state. The usage of services for options-analysis and the composition of the state machines for each option type would allow this type of reasoning in the system.

Finally, the work at hand is the first conceptualization and implementation of deliberative decision-making in SiMA. It has taken SiMA one step closer to the functionality of the human mental apparatus.

References

[1] S. Schaat, K. Doblhammer, A. Wendt, F. Gelbard, L. Herret, D Bruckner. A Psychoanalytically-Inspired Motivational and Emotional System for Autonomous Agents, in proceedings of IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, 2013, pp. 6647-6651.

[2] J. E Laird, M. van Lent. Human-level ai's killer application: Interactive computer games. AI Magazine, AAAI, Volume 22 No. 2, 2001.

[3] R. Lye, T. Peng Lung, S. A. Cheong. Understanding agent-based models of financial markets: A bottom–up approach based on order parameters and phase diagrams. Physica A: Statistical Mechanics and its Applications, Volume 391, Issue 22, 2012 pp. 5521–5531.

[4] ECABA - Energy-efficient Cognitive Autonomous Building Automation, http://sima.ict.tuwien.ac.at/projects/ecaba/, TU Wien, SiMA homepage, accessed on 2015-07-28

[5] D. Bruckner, F. Gelbard, S. Schaat, A. Wendt: Validation of cognitive architectures by use cases, in proceedings of Industrial Electronics (ISIE), 2013 IEEE International Symposium on, 2013.

[6] A. Wendt, S. Schaat, F. Gelbard, C. Muchitsch, D. Bruckner. Usage of Spreading Activation for Content Retrieval in an Autonomous Agent, in proceedings of IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, pp. 6670-6675, Vienna, Austria, 2013.

[7] M. M. Richter, O. R. Weber. Case-based reasoning. A Textbook, 2013.

[8] W. Duch, R. J. Oentaryo, M. Pasquier. Cognitve Architectures: Where do we go from here?, Frontiers in Artifcial Intelligence and Applications, vol 171, 2008, pp. 122-136.

[9] D. Vernon, C. von Hofsten, L. Fadiga. A Roadmap for Cognitive Development in Humanoid Robots. http://www.vernon.eu/publications/10_Vernon_vonHofsten_Fadiga_COSMOS_Extracts.pdf (accessed 02/2015), 2010.

[10] P. Langley, J. E. Laird, S. Rogers. Cognitive architectures: Research issues and challenges. Cognitive Systems Research, 10(2):141–160, 2009.

[11] J. E. Laird. Extending the Soar Cognitive Architecture. Proceeding of the 2008 conference on Artificial General Intelligence. Division of computer science and engineering, University of Michigan, 2008.

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

[13] P. Langley, D. Choi. A Unified Cognitive Architecture for Physical Agents. Computational Learning Laboratory, Center for the Study of Language and Information, Stanford University, Stanford, CA 94305 USA, 2006.

[14] S. Gottifredi, M. Tucaty, D. Corbatta, A. J. Garcia, G. R. Simari. A BDI architecture for high level robot deliberation. Inteligencia Artificial, 46:74–83, 2010.

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

[16] S. Schaat, A. Wendt, M. Jakubec, F. Gelbard, L. Herret, D. Dietrich. ARS: An AGI Agent Architecture. In Proceedings of Artificial General Intelligence, Springer, 2014, pp. 155-164.

[17] N. Nilsson. Teleo-reactive programs for agent control. arXiv preprint cs/9401101, 1994.