Lessons Learned From Modelling Situated Cognitive Agents Interacting With a Dynamic Environment

Lukas Seiling*, Maximilian Plitt*, Paul Schweidler*, Oliver Klückmann*, Lilian Befort, Robin Konczir, Severin Reuter, Kai Preuss, Sebastian Wiese and Nele Russwinkel
Department of Psychology and Ergonomics, Marchstraße 23
10587 Berlin, Germany

*** Authors contributed equally**

Abstract

The study of knowledge representations and reasoning problems faced by a cognitive agent interacting with a dynamic and incompletely known world is relevant to cognitive robotics and understanding complex cognition and related fields. The paper introduces four cognitive agents that were modeled in a student project with specific requirements. The cognitive architecture ACT-R was used to model flexible agents that interact with objects in a grid field with only a limited field of view. Long-term planning is not possible here: the meaning of objects needs to be discovered and the field explored to find the goal as quickly as possible. The project demonstrates how the four agents learn from interactions and what information needs to be kept available to flexibly decide in unpredictably occurring situations. All four agents are shortly described in more detail. The project covers on a small scale some aspects that are crucial for autonomous agents in a simple game environment. The four agents are faced with 15 challenge environments that need to be explored and managed. The challenge performance results show that a higher number of productions does not necessarily lead to better performance.

Keywords: mental representation; situated cognition; embodied cognition; situated agent; interactive learning

Introduction

Everyday we cope with new challenges, experience time pressure when we try to complete our tasks and need to flexibly react to changes and new information. Normally we do not have time to evaluate different options, but have to instantly find a good solution and act accordingly. Therefore, in such situations our cognitive system not just relies on independent, well-elaborated rational processes, but often depends on the situation and context in which cognition occurs. Embodied cognition holds that cognitive processes are deeply rooted in the body's interactions with the world (e.g. Wilson, 2002). The research field of embodied cognition is still widely heterogeneous, but there are some distinct characteristics of embodied cognition most researchers agree upon. Embodied cognition is situated, cognition is time-constrained, and whenever possible we offload cognitive work onto the environment. According to Levesque and Reiter (1998), cognitive robotics is the study of knowledge representation and reasoning problems faced by an autonomous robot (or agent) in a dynamic and incompletely known world. This leads to the research question what cognitive mechanisms are central to build a cognitive agent that is able to cope with such an environment and is still to accomplish its goals. Situated agents in dynamic environments are good test beds to investigate different implementations of such mechanisms because of their required abilities to flexibly manage changes in the environment, explore unknown objects and to handle novel challenges. In the long run this kind of research is relevant to learn more about complex cognition. According to (Funke, 2010) complex cognition deals with all mental processes that are used by an individual for deriving new out of given information, with the intention to make decisions, solve problems, and plan actions. This assumes an active and goal-directed information processing by an agent that is able to perceive its environment and to use its memory. In a complex situation, the result is more than the sum of perceptual, learning, and memory processes. In this sense perception can be seen as part of a higher structure. The context delivers the meaning which is not only given by itself but in combination with other events and objects. In addition this kind of research has the potential to develop good solutions for cognitive robotics or human-robot collaboration.

The aim of this paper is to explore these questions within a simple task environment. We want to show four different realizations of such an agent for the same requirements and task environments within the cognitive architecture ACT-R (Anderson, 2000). These examples can support other researchers faced with similar task requirements to reason about ways to build such an agent. Usually each modeler starts with their own idea, therefore such model challenges are useful to explore a wider field of possible implementations and an evaluation thereof. Furthermore, we hope to contribute information to the question of how situated cognition can be realized with a cognitive architecture and what possible architectural developments are promising in order to address such research fields. In a student project of about 2 months, the given task was to develop a situated agent that should use mainly cognitive plausible mechanisms to deal with several challenges.

Agent requirements

The game (a grid field with several colored object on its tiles, see Figure 1) required the interactive agent to

1. find out what object on the grid represents the agent and the color it has

2. search for the goal that has to be approached. The goal is not visible at the beginning

3. find out what object with what color shows what kind of effect on contact (obstacle, add points, deduct points), in other words to explore the environment and infer the best way towards the goal

4. cope with constantly appearing and disappearing objects (due to a *fog of war* mechanism, only objects in immediate surroundings are shown) and make decisions based on a mental representation of the environment.

The goal is to move onto the goal tile as fast as possible, preferably with a high score. At the end of the course, 15 novel challenge fields were provided and the agents were tested in order to explore and evaluate their respective performance and flexibility in unknown environments. Prior knowledge for all agents was:

- the color of the goal is green,
- the agent starts on the top row (color is unknown),
- the goal is below the agent, in the lower half of the grid,
- possible movements are left, right, up, down; these are restrained by a yellow bounding box ,
- movements towards a colored tile can have three different consequences (blocked movement, pass and win points, and lose points),
- the effect of object colors are randomized each trial, except for the green goal object.

The main cognitive skills required are therefore to learn about the environment through interactions, to explore the grid in order to find the goal, to draw inferences, to gather information and hold this information in mental representations and to make decisions based on available information. In the following sections, the different cognitive agents will be introduced, and for each agent it will be explained how and where different aspects of information are gathered and represented (for instance by chunk representations in some buffer, e.g. goal or imaginal; or production rules) and how this knowledge is used in specific situations. Then it will be sketched out how the four requirements mentioned above are realized for the different agents. Lastly the main benefits and weaknesses of each agent are discussed and performance in some situations is described.

Some requirements are realized similarly across the agents, such as identifying itself through an action and checking for the object that moved. Since it is known that the agent starts in the top row, all visible objects are encoded and movement is initiated. When a change is registered by the visual module, the color of the moved object is stored as a mental representation in either the goal or imaginal buffer. Most agents also initiate object tracking via the visual module to keep their agent representations in focus. To facilitate self-localization and an understanding of the grid's dimensions, the agents make use of geometric data of the objects and borders for simple heuristics, such as moving towards the center of the grid. This data is also stored as part of the agent's mental representation of the task.

Models

Speedy

Orients itself according to sub-goals. Bonus points are collected when close. Acts pre-attentively and therefore quickly.

Relevant information for the agent is stored in the imaginal buffer including its current position, movement intention as well as its specific color. At a later point, the colors of additional tiles are stored in the imaginal buffer according to their meaning. After successful self-identification the agent starts searching for the goal. To reach the goal tile as quickly as possible, a strategy of subgoals is pursued. Subgoals represent specific waypoints the agent tries to reach. Information regarding the agent's current subgoal is stored in the goal buffer. The goal buffer also contains the minimal and maximal x and y coordinates of the grid field, representing the borders and the distance to the subgoal. Since the goal object is located in the lower part of the grid field, the agents tries to reach its first subgoal, which is directly in the middle of the grid field in order to explore the space where the goal could be located.

Searching the environment First, the agent routine searches its visible field for the green goal and follows its subgoal. The adjacent tile in the agents movement direction is checked. In case of an object of unknown color, the object gets evaluated. Otherwise a movement according to the object's meaning is executed. An unknown object is tested by the agent moving onto it. Object meanings are evaluated by using the visual-location module that searches for an appearing text (score). In case a red text appears, it is examined for "+" and "-" signs and accordingly the bonus or malus chunk stored in the imaginal buffer is filled. In case no text appears, the color is inferred to be an obstacle. For obstacles or malus objects, movement direction is changed. Bonus tiles are collected whenever in the vicinity.

Locating the goal As soon as the agent gets close to a subgoal location, the subgoal chunk gets updated. Further subgoals in order to find the goal are pursued, namely reaching the bottom-left corner and reaching the right grid border at three quarters of the grid's height. The green goal object is detected with a pop-out effect due to the high utility of the search production. As soon as the goal is recognized, its location is stored as the new subgoal.

Decision making and problem solving Even if the agent's heuristic is to move directly towards its subgoal, the walking path is not implemented as a straight line. Steps are chosen randomly, whereas the movement in subgoal direction is prioritized. If there are obstacles or malus objects on the path, they are avoided. In case the movement direction is changed

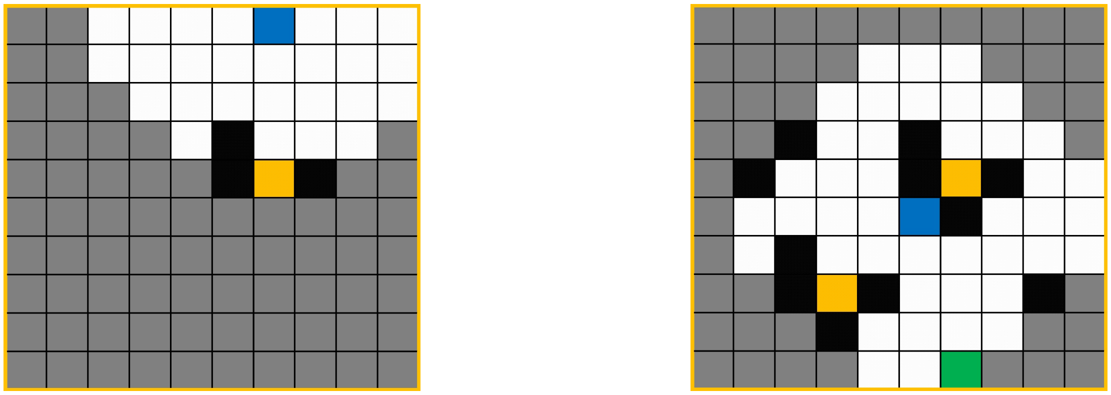

Figure 1: Examples of two situations in the same environment. The area in the proximity of the agent is ”visible” to it, anything beyond is invisible, i.e. covered by a *fog of war*. On the left, the agent is in a starting position, with 4 objects in 2 colors, but not the goal directly visible. Black tiles are obstacles while yellow tiles are bonus fields, although in the beginning the meaning of these colors is unknown to the agent. On the right, the agent advanced to a point where the goal tile is now visible. Note that exploration revealed ”traps” to the agent, enticing it to enter a dead end of obstacles.

the blocked location is stored in the imaginal to prevent the agent approaching the same object again. This mechanism is helpful to free the agent from triangular ”traps” or to walk around obstacles blocking the path.

Strengths and weaknesses Since the agent perceives all objects only pre-attentively, the speed to reach the goal tile is maximized. This is supported by the fact that it does not take bonus points into consideration unless they are located directly on the path. Therefore, the resulting score is low compared to the other agents. If the goal tile is spotted but not reachable, the agent will be stuck at an obstacle and cannot free itself due to its goal-directed heuristics (with a total of 41 productions). During testing, the agent was able to complete 12 out of 15 challenges. This represents an overall satisfying performance with special regard to the enormous speed with which Speedy solves the challenges. However there is still room for improvement, including priority collection of bonus objects for higher scores as well as the ability to recover from entrapment. As soon as the agent spotted the goal object with its path blocked, the agent gets stuck. Therefore, a possible option would be to change the subgoal in case the distance to this subgoal is not reduced within several steps.

Forest

Represents the current goal by separate state, intention and searching slots in the goal chunk. Unknown tiles are sought to be identified.

This agent attends a random tile with an object in the top row of the grid. When an object is located pre-attentively the agent will try to exercise a movement. After the movement the agent is attending the same location again to check whether the color of the tile has changed. In case of a color change - the agent was found. In the other case a different color will be perceived and the same process is repeated. This color information is stored in the goal-chunk in order to have a sustained awareness of itself.

Searching the environment Before executing a move the agent checks whether the next targeted tile is blocked by another object. If the object is unknown the agent will try to move on that tile. By that the classification will start. If the color of the desired field changes to the color of the agent the object will be classified as a bonus or malus tile. If the color of the desired fields does not match - the object is classified as an obstacle. When the agent encounters an object, it will retrieve a chunk from the declarative memory. In case of an obstacle or malus tile it will try to avoid that tile and in case of a bonus tile it will try to move onto it.

Locating the goal With the analyzed and calculated grid the agent has a good starting point and orientation to use its first heuristic - make your way to the middle of the grid. Since the agent is aware that the main goal will be in the lower half of the grid. After reaching it desired location the agent will use another heuristic. This heuristic is based on a waypoint system by exploring the left and the right side, making its way to the bottom of the grid. In the routine a high utility production ensures that a visible green tile will be prioritized as the new main goal.

Decision making and problem solving In general the agent’s processes are organized in routines as visualised in Figure 2A. Inside of the main routine a hierarchic structure is used to ensure that distinct routines are available at certain times. For example the agent checks before every movement whether the green main-goal is visible or the adjacent field is an unknown or known object. A wide variety of 53 productions were used to solve other problems on its way to the green tile. That included several escape mechanisms inside the decide-action routine to get around the obstacles. A specific goal-chunk is used to hold slots of the actual state, an intention and a searching slot. Each slot had a distinct assignment: the state-slot was used to guide the agent through the heuristics, the intention-slot was used to remember the loca-

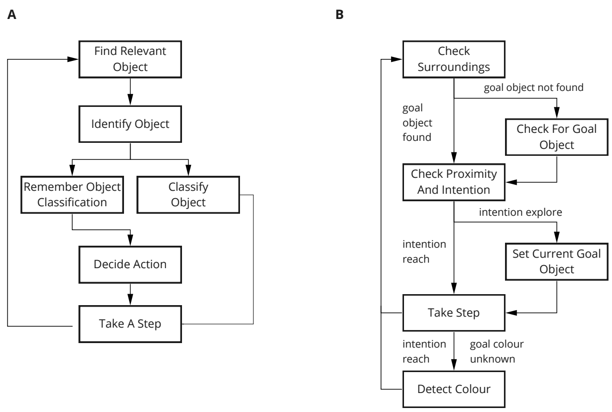

Figure 2: Flowcharts of the goal-finding routines of the agents named “Forest” (A) and “Intell-agent” (B) after self- and board-detection.

tion the agent was moving to and the searching-slot gave it the actual state of the routine.

Strengths and weaknesses Our agent was comparatively fast when moving through the presented environments. The approach was to keep it simple and to process everything pre-attentively. This meant less costs in terms of time. On the other hand the agent was not able to solve neither complex nor medium difficulty levels, because avoiding movements were not context sensitive. A key aspect for improvement would be the agent’s capacity to remember where it is coming from.

Ms. Captain Curious

Explores the environment randomly. Actively seeks out novel tile colors. Identifying all encountered tiles is prioritized over reaching the goal.

Ms. Captain Curious was developed to be a lightweight model employing only a few heuristics that are generally useful, instead of specialized strategies for different cases. The model does not build up a lot of permanent knowledge. Most of the relevant information is retrieved from the visual module when needed and then temporarily stored in the goal or imaginal buffer. In the following, some of the heuristics, or routines, are described followed by a brief look at the overall performance of the model.

Searching the environment The localization of the agent relative to the grid was necessary to determine a suitable direction for the next moves - instead of steering the agent against the border, an unexplored direction was preferred. The heuristic was to find the borders of the field and move in the direction of those borders that are away the farthest. Since only the eight fields adjacent to the spaceship were visible at any given time, movements were necessary to explore the grid. The agent then started moving from the current quadrant to the one diametrically opposite. Only when touching borders or unknown objects, the direction was reversed or re-evaluated. On an empty grid, the movement pattern would thus resemble a billiard ball rolling unchecked from edge to edge across a billiard table.

Locating the goal In order to find out the meaning of the different colored objects, the agent had to be steered on unknown object tiles to then memorize the consequence (point gain, point deduction or obstacle) and subsequently deal with the tile types differently (seek, avoid). This curious, minimal learning behavior can be described as ”Whenever you see a novel object, check it out and remember the consequence”. This primacy of curiosity was eponymous for the agent. In case the colors of obstacles or malus fields have already been determined they were avoided. Whenever there was a known object on a field in the intended direction, a new direction of movement was selected randomly. This is no long term planning strategy, only a consideration of the next step.

Decision making and problem solving After each single movement, the agent checks whether the goal or any known or unknown object is in sight, and if there is a contact with the border. The checks have different priorities: Identifying the color of bonus tiles has the highest priority. Once the color of the bonus tile has been identified, approaching the target is set above the tile identification. Finally it is evaluated whether the spaceship is at a border, if necessary the direction of movement is reversed. Additionally, further abstract procedural patterns were identified and made explicit, which, for example, governed the handling of obstacles and malus fields or determined the hierarchy of routines.

Strengths and weaknesses Overall, the agent performed comparatively well, solving most of the challenges within the given time constraints with a total of 83 productions. Yet, since an initial decision was to design an agent that moves around a lot rather than thoughtfully weighing each of its decisions, wall-like rows of obstacles posed serious problems: When an unknown object or the goal was situated behind a row of obstacles, the agent would move towards it, resulting sometimes in the agent getting irresolvably stuck. It was serendipity, provoked by the random selection of movement directions, that sometimes helped the agent to circumvent those problem situations nonetheless.

Intell-Agent

Uses a 12 tile diamond-shaped visual representation for reasoning. Intentions determine goal pursuit behaviour. Preference to collect visible bonus points followed by reaching the goal.

The Intell-agent represents its environment as a diamond-shaped field consisting of 12 tiles (see Figure 3). This information, the agent’s current x- and y-position as well as its last move, are saved in a “vision”-chunk in the imaginal buffer consisting of 15 slots. Consequently it is able to detect and avoid immediately adjacent triangular traps, which are token (malus and obstacles) in an disadvantaged formation.

The goal buffer is relevant for representing the agents’ states and intentions. States describe the various stages the agent passes through as part of its goal-finding routine as detailed below. Intentions specify the overall tactic of approaching the set goal. The minimal and maximal x and y coordinates of the grid field are also contained in the goal buffer and represent the borders of the grid field. Lastly, the goal is responsible for upholding information about kind and position of the current goal. The Intell-agents goal-finding routine is structured in five stages shown in Figure 2B.

Searching the environment The visual representation is updated based on the agents current position and its last motion. The visual-location module is used to check for tiles on which the agent does not have any information yet. Tiles outside of the grid field are marked so that they won't be moved on.

Locating the goal The agent checks if the green goal object is visible. If so, its location is stored in the declarative memory. In case a chunk with the goal's location can be retrieved, the agent directly proceeds to the next stage. The agent's current position and the goal position are then compared to determine if the agent is next to the current goal. Based on its intention at that stage, it will then either skip the next stage and move directly towards the current goal (intention: reach) or enter the goal setting stage (intention: explore).

During goal setting, the agent sequentially goes through a list of priorities, only moving onto the next one in case the previous one does not apply.

1. The agent's top priority is to find out which color represents the bonus field. Therefore, if a tile with an unknown color is detected and the bonus color is unknown its position is set as the current goal and its intention is set to "reach".

2. If however, the bonus color is known, reaching bonus tiles becomes the highest priority. The agent tries to find the visual-location of the nearest tile with the bonus color. If successful, the bonus tile becomes the current goal and its intention is set to "reach".

3. If the bonus color is known but no bonus object is visible, the agent looks for the green goal object. In case the green goal object is not visible, the agent tries to retrieve the goal position from the declarative memory. If successful, the green object's position will be set as the current goal and the intention is set to "explore".

4. If the green tile is neither visible nor retrievable, the agent applies a searching heuristic. Depending on current position of the agent relative to the grid field boundaries, the agent will set its current goal to the center of the playing field (if in the top half of the playing field), the lower left corner (after the middle was successfully reached) and eventually the lower right corner (after the left corner was successfully reached).

Decision making and problem solving The agent tries to reduce either its vertical or horizontal distance to the current goal. If this is not possible, it will move onto an object that it has not yet visited. This behaviour is facilitated using 'breadcrumbs' which are placed on tiles' representations within the imaginal buffer that were previously visited. Such tiles are avoided by the agent. All breadcrumbs are removed whenever any goal is reached.

When the agent seeks to identify the meaning of different colors, it does so by moving on to a tile with unknown color and checking for a visual location with red text. If none is detected the tile color is saved as the obstacle color. If red text is perceived, the agent's movement is evaluated. Given the agent has not moved, the color of the current goal is saved as malus color. Otherwise the agent has successfully found out the bonus color.

Strengths and weaknesses The strategy resulting from this cycle is to maximize the score during level completion. This can lead to prolonged run times and sometimes even unexpected behaviour, for example when the agent moves away from the goal just to collect another bonus. Overall the performance of the agent was satisfactory, but due to its tendency to collect every bonus and the fairly high complexity of the model (with a total of 226 productions) the agent is sometimes quite slow and exceeds the time limit of some challenges.

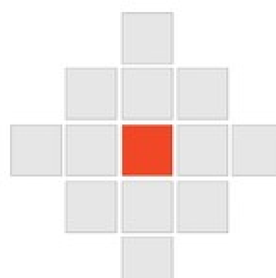

Figure 3: Intell-Agent's twelve tile representation of the agent's environment. Each tile was indexed and could carry the values "empty", "breadcrumb", "out of bounds" and the various object colors.

Performance Review

All four agents tried to cope with a partly unknown environment and tried to apply complex cognition as defined by (Funke, 2010). Thus the agents used mental processes for deriving new information out of given information, with the intention to make decisions, solve problems, and plan actions. They had to perceive their environment and use their memory. Not all mechanisms are cognitively plausible and are sometimes oversimplified, yet the agents were able to flexibly handle new environments and settings and find their way to the goal. Although the task was the same and required the same cognitive skills for fulfilling the four requirements, the implementation of the individual agents varied a lot. The number of productions is a first indication for this, with a range from 41 to 226 productions - with the lowest production agent still performing fairly well.

Table 1: Aggregated Agent Model Results

Agent	# of productions	Completed (out of 15)	Avg. # of Moves	Avg. Time (s)	Avg. Score
Speedy	41	12	26.75	14.184	1145
Forest	53	4	24.10	29.50	80
Ms. Cap. Curious	83	13	39.46	39.78	1180
Intell-agent	226	13	26.61	62.04	1177.2

In Table 1 the different performances are listed and it becomes apparent that most agents were able to manage most of the challenges - some of which were quite complex and difficult to solve. Most importantly lessons were learned on what was helpful to model such an agent having to cope with a dynamic and incompletely known world, i.e. what kind of knowledge representations were crucial and how reasoning problems were solved.

The following insights were gained from modelling situated cognitive agents interacting with the introduced dynamic environment. First, relevant information about the agent (self representation such as colour, once identified) and non-changing information about the environment such as its size are held available, usually in the goal buffer. Further relevant information about the current context, acquired information about the meaning of objects and information about the goal location are also stored, usually in the imaginal buffer. Second, sub-goals or identifying different phases of the task are helpful to find flexible ways to solve specific problems or to have access to specific productions. Third, as speed is important, it is essential to identify what information is relevant and would take too much time to retrieve often. This information should be available in one of the buffers. Information that is only necessary infrequently can be retrieved from memory. Also pre-attentive visual processes were used whenever possible and cognitively plausible, such as for self-localization or when searching for specific information, consequently saving time. Forth, strategies that are too rigid usually lead to situations where agents get stuck and have difficulties to free themselves. Furthermore, keeping information about past movement of the agent or what areas have already been searched are greatly helpful.

Discussion

The main lessons learned by the students were **(1)** to develop a better understanding of what cognitive plausible mechanisms really are, where difficulties lie and how to change the usual approach to this kind of task. Still, parts of the agents show computational aspects rather than cognitive, but time was restricted for the project. **(2)** The second lesson learned was to realize how important it is to use detailed task analysis and visualizations of model structure for group communication while modelling. **(3)** The third point was that pre-attentive visual processes are sometimes sufficient for simple localization and checking purposes of the agent.

Lessons learned regarding the architecture used

ACT-R offers a lot of structures that are helpful to model flexible and learning agents in task environment such as these. Debugging and handling its output was a challenge sometimes. Visual grouping or perception of a "field" and identifying borders was difficult to realize in a cognitively plausible way. Additional visual support would be a very helpful component for research on self-sufficient agents.

The project nicely showed the aspects and requirements that Kurup and Lebiere (2012) listed for high-level cognition in robotics. **(1)** Represent, integrate and use large amounts of knowledge: it needs to be carefully considered what information really needs to be stored and where. Storing the whole grid field in our example would have slowed down the agents, so this was not done. Rather, the students tried to find ways to solve the most important problems with as little stored information as possible, since human participants would also not store all tiles in the grid. **(2)** Learning patterns: in this project, the agents learned the meaning of objects by interacting with them and adjusted their planning accordingly. There was not enough time to learn from difficult situations and obstacle patterns, which would be highly interesting and potentially address **(3)** Problem solving and reasoning. **(4)** Flexible, adaptive, dynamic, and real-time behavior was shown by the agents. Explicit encoding of visual objects was prevented as much as possible in order to not lose valuable time. The agents were also able to flexibly cope with newly appearing objects, an unseen goal and different environments - therefore long-term planning was not possible. The last requirement, **(5)** Interact with humans in a natural way, would require a more refined approach.

This type of challenges for cognitive agents, as mentioned earlier, offers intriguing test beds to explore how flexible models based on a cognitive architecture really are and how much such approaches could add to existing agent approaches. Especially the topic of mental representations (e.g. Clark & Grush, 1999) is crucial in such unpredictable environments and for adaptive and complex behavior. This potential should be explored in more detail.

Acknowledgments

The authors would like to thank their respective project partners and all other participants of the 2020/21 winter semester course "*Applied Cognitive Modeling*" at the Technische Universität Berlin.

References

Anderson, J. R. (2000). *Learning and memory: An integrated approach* (2nd ed.). Wiley.

Clark, A., & Grush, R. (1999). Towards a cognitive robotics. *Adaptive Behavior*, *7*(1), 5-16.

Funke, J. (2010). Complex problem solving: A case for complex cognition? *Cognitive processing*, *11*(2), 133–142.

Kurup, U., & Lebiere, C. (2012). What can cognitive architectures do for robotics? *Biologically Inspired Cognitive Architectures*, *2*, 88–99.

Levesque, H., & Reiter, R. (1998, March). High-level robotic control: Beyond planning. A position paper. In *AIII 1998 Spring Symposium: Integrating Robotics Research: Taking the Next Big Leap.*

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin & Review*, *9*(4), 625–636.