

# Machine Learning Algorithms for Cognitive and Autonomous BDI Agents

Ömer Ibrahim Erduran

Department of Computer Science, Goethe University, Frankfurt am Main, Germany

## Abstract

The concept of *cognitive Agents* has its roots in the early stages of multi-agent systems research. In early agent research, the understanding of the term *agent* was referring to *software agents* with basic capabilities of perception and action in a proper environment adding potential cognitive capabilities inside the agent architecture. A fundamental drawback of the concept is the barrier to learning capabilities since the full properties of the agent are hard coded. Over the years, the research in Agent-oriented software engineering has provided interesting approaches with promising results in the interplay between Machine Learning methods and cognitive software agents. Such a combination is realized by an integration process of Machine Learning algorithms into the cognitive agent cycle in the specific architecture. This paper gives an overview of the combination of both paradigms including the applied concepts and architectures for different scenarios. A three-dimensional cube *ML-COG* is introduced to illustrate the integration perspectives for both paradigms. Therefore, the considered literature refers to ML-COG by arranging them into a cube. After a concise literature review, a selection of relevant research questions and open issues is presented as worthwhile to be investigated.

## Keywords

Multi-Agent System, BDI Agent, Machine Learning, Agent-oriented Programming, Cognitive Agents

## 1. Introduction

Over the years, research in autonomous Agents and Multi-agent systems (MAS) has emerged into a multi-disciplinary field with influences from a wide range of related scientific fields containing a plethora of paradigms. Due to the rapid advancement of Machine Learning (ML) algorithms, especially in Deep Learning and Reinforcement Learning, the understanding of agency reflected by the term *agent* has gained different meanings. This circumstance has been pointed out by *Dignum et al.* [1], according to which the different understandings of *agent* could be fundamentally seen on the one side as a concept or on the other side as a paradigm for autonomous software systems. Furthermore, *Shoham* pointed out the fundamental shift from logic-based AI and Knowledge Representation to ML and statistical algorithms [2]. In the viewpoint paper of *Bordini et al.* [3], the authors proclaim a *cognitive era* and investigate the contributions of Agent-oriented Programming (AOP) to future intelligent systems. Specifically, they mention AOP as an approach for the rapid development of cognitive agents which are context sensitive. This means, that for a given scenario or a task that has to be processed,

---

LWDA'22: Lernen, Wissen, Daten, Analysen. October 05–07, 2022, Hildesheim, Germany

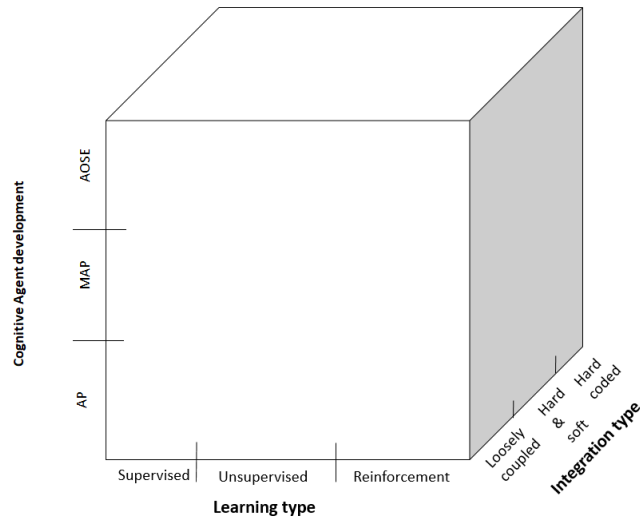
✉ [erduran@cs.uni-frankfurt.de](mailto:erduran@cs.uni-frankfurt.de) (Ö.I. Erduran)

ORCID [0000-0002-1586-0228](https://orcid.org/0000-0002-1586-0228) (Ö.I. Erduran)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)



**Figure 1:** The proposed multidimensional ML-COG Cube

software agents can be applied on large scale being extended or specified with capabilities for a given scenario, e.g. as autonomous vehicle agents for transportation in Mobility or warehouse agents for sorting and packing goods for deliveries. Since the goals and plans as well as the predefined set of possible actions are usually implemented into the agent architecture, it shows a robust behavior in its corresponding environment. This circumstance represents a contrast to the learned behavior in ML. A main disadvantage of ML is the black box component, where the insight into the underlying structure of the learning process can not be seen. That is why the behavior of a learning agent based on *Russel & Norvig*[4] for instance, can not be explained thoroughly, especially in sub-symbolic ML approaches. In Deep Reinforcement Learning (DRL), the learning agent behavior leads to actions, which are also difficult for humans to understand<sup>1</sup>.

Since over the years research has been done in the considered intersection, the contribution of this survey is to bring together a significant amount of research, where BDI is added with ML methods categorizing them concerning the technical realization as well as the considered ML methods. Furthermore, the survey points out research areas in this intersection worthwhile for deeper investigation. To clarify the corresponding setting of the work handled, we first explain the fundamentals which are considered in this survey. Specifically, we also distinguish the sections of Multi-agent Learning (MAL) which is a prevalent research direction in the combination of agency as a concept and usually considering reinforcement learning algorithms. To structure the literature which we investigate in this survey, we clarify our approach to set the research focus of this survey. As mentioned, the integration of ML and AOP is the core research intersection, where the works that have been done so far are represented in this survey. To the best of our knowledge, this is the first survey, which explicitly considers ML and AOP for the cognitive BDI agent architecture. Furthermore, it comprises mainly the work done in

<sup>1</sup>Here, one can look at the well-known "Move 37" of AlphaGo from DeepMind, mentioned in <https://www.deepmind.com/research/highlighted-research/alphago>, last access: 04/18/2022.

the last two decades. The remainder of this survey is structured as follows: section 2 contains the preliminaries as well as a distinction of the topic handled with other directions to prevent misconceptions. The categorization approach of this survey is handled in section 3, where the ML-COG cube is described in particular (Fig.1). In the main part, section 4, we examine the existing literature presenting different approaches to tackle the challenge of integrating ML and AOP and furthermore categorize the considered works into the ML-COG cube. After the categorization, we present in section 5 the elaborated open challenges and directions that are worthwhile for profound research. Finally, we conclude our survey in section 6.

## 2. Fundamentals

A compact exposition of both paradigms ML and AOP is presented in the following subsections focusing on the main aspects.

### 2.1. Machine Learning

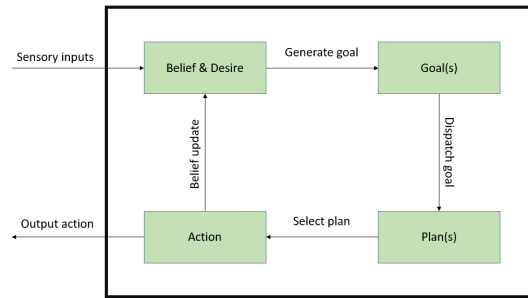
Learning algorithms are data-driven, which means that for specific learning behavior, the algorithm gets exposed to a large data set. Here, the learning process can vary according to the learning objective and furthermore, the setting. In principle, the relevant learning algorithms can be subdivided into 3 categories: *supervised*, *unsupervised* and *reinforcement learning*. All of them have been investigated with respect to the integration into the cognitive agent architecture [5, 6, 7]. In supervised learning, the learning algorithm gets a proper training data set to apply the learning process therefore learning a specific behavior <sup>2</sup>. After the training process, the testing step examines the performance of the learned behavior with a smaller sample from the data set which is not considered during the training phase. In contrast, unsupervised learning considers learning algorithms that are given the objective to find contextual structures in a given data set. Thus, the learning algorithm does not get information about the objective but has to find an underlying structure. In reinforcement learning, a learning agent is considered, that interacts in an environment to learn and perform a specific behavior. Here, the agent itself gets rewarded or punished for its actions in this environment. Based on a reward function, the goal of the agent is to maximize the reward which leads to a specific behavior in the given environment. These 3 learning approaches are also covered in the ML-COG cube and also are objected to in this survey.

### 2.2. BDI agent architecture

Autonomous Agents have been broadly investigated in distributed artificial intelligence. Different applications, where agents come into play, are among others ranging from Negotiation mechanisms and Game Theory to Distributed problem-solving. In Agent Programming, we suppose an internal cognitive architecture based on the *observation, thought, act cycle* that each considered cognitive agent applies during processing in its environment [8]. Starting from the fewer capabilities of a reactive agent that only reacts to senses from the environment, the more

---

<sup>2</sup>Here, *proper* means the suitable choice of a data set for the learning objective.



**Figure 2:** The typical cognitive BDI cycle based on Deljoo et al.[12]

complex cognitive architecture is usually represented by the *Belief, Desire, Intention* - in short BDI - architecture. The BDI model is a goal-oriented practical reasoning system and it has its roots in practical psychology. A pre-version of the BDI model is the *Practical Reasoning System (PRS)*. Bryson[9], for example, presents learning for PRS and cognitive agents based on the cognitive logical model of *Marvin Minsky* [10]. Learning, therefore, has been the main challenge since the beginning of cognitive reasoning systems development. In the Agent literature, there exist multiple variations of the BDI architecture, where one example is depicted in Fig. 2. The agent observes information from the environment, defining its *belief*. The *desires* are derived from the beliefs, indicating the planned behavior of the agent. For each desire, a sequence of *goals* and *plans* as combinations, which are defined, come into play. A single plan can contain multiple *actions*. An action is then executed by the agent in its environment and the beliefs are updated at the same time. A more comprehensive survey that covers the BDI agent architecture and its variations, is examined by *Silva et al.* [11].

### 2.3. Integration type

As a third dimension in the ML-COG cube (Fig. 1), the integration type denotes, in which form both paradigms ML and AOP are deployed during the architectural design and implementation phase of the intelligent agent system. It ranges from a fully designed and programmed agent behavior to the end where the agent behavior is fully trained with ML techniques. If the learning algorithm is implemented into the BDI architecture, we consider it as *Hard*. If the learning algorithm is modular, it is called *loosely coupled*. Consequently, a combination of both is called *Hard&soft*<sup>3</sup>.

## 3. ML-COG cube categorization scheme

The integration of ML into the BDI architecture as two distinct paradigms is the core area that we consider in this survey. To provide a clear view of this intersection with the corresponding published works, we set up a categorization scheme. The rationale for this scheme is based on a problem-solution order since we focus on an integration problem, which can be seen

<sup>3</sup>Another interesting scale representation distinguishing between learned and hard-coded behavior is introduced by Ricci, A. in his talk "Agent Programming in the Cognitive Era: A New Era for Agent Programming?", EMAS 2021.

equally as an implementation problem in AOP. Displayed as a cube structure, we present the ML-COG cube to classify the considered research. In its basic features, ML-COG comprises 3 main dimensions. The first dimension, which is defined as the *cognitive agent development*, is reflected on the  $y$ -axis. In this dimension, we distinguish between different agent development approaches leaning on the fundamental literature of Multi-agent research based on *Wooldridge* [8]. The development of cognitive agent architecture ranges from a single agent approach in Agent Programming (AP) [13], to multiple agents interacting with each other in Multi-Agent Programming (MAP)[14] up to Agent-oriented Software Engineering (AOSE)[15], where the whole Software engineering process for MAS is covered including organizational roles and the interacting environment. All of the 3 approaches in the  $y$ -axis are constricted to BDI agents. In the second dimension, we accordingly envisage the ML perspective, which is reflected in the  $x$ -axis. Here, we differentiate between the learning approach, which can be in general divided into *supervised learning*, *unsupervised learning* and *reinforcement learning*. Since the core of this survey is the integration of both ML and AOP, we focus on adding both dimensions together accordingly by investigating the different approaches. Therefore, we distinguish in the third dimension the type of integration, represented in the  $z$ -axis. Here, we distinguish a loosely coupled integration, a hard-coded, and a combination of soft and hard-coded integration types. Throughout the survey process, we assign each considered work and approach. It is important to note that, we constrict the surveyed literature mainly to the approaches, where ML is considered for BDI agents. Thus, we have to neglect prominent works where Learning is investigated into other types of cognitive architectures like *SOAR* or *Act-R* <sup>4</sup>. From this starting point, we went through the publications cited in the considered works. Due to space constraints, we consider specific representative works for ML-COG. We also apologize to the authors whose work we had to omit due to space constraints.

## 4. Literature Review

Based on the approach explained in section 3, this survey covers works that combine ML for AOP, especially considering the BDI architecture. The literature collection is processed by selecting works where ML approaches are applied to BDI agents, i.e. the learning algorithm is integrated into the BDI cycle. We examined a plethora of works neglecting approaches, where ML is though considered but not for BDI agents. One work mentioned before is from *Bordini et al.* [3], where the literature is examined with respect to Artificial Intelligence in general for BDI agents. The mentioned work considers ML approaches but is not limited to. Whereas in this survey, the focus solely lies on ML for BDI agents. Based on the fact, that they cover a broader range of the literature spectrum, they do not go into detail for specifically mentioned works that are also subject to this survey. In this survey, the ML paradigm and BDI architecture are opposed, and thus, we explain the related literature in this more specific context considering the introduced categorization scheme. In this survey, we filter in section 5 challenges concerning ML and AOP.

---

<sup>4</sup>Interested readers are referred to *Broekens et al.*[16] and *Nason & Laird* [17].

#### 4.1. BDI and Decision trees

One of the first works mentioning ML approaches for BDI agents explicitly is from *Guerra-Hernández et al.* [5, 18, 19] where the plan selection process is investigated by applying logical *decision trees*. As a typical Supervised Learning approach, this method is integrated into the BDI cycle by adding the decision tree into the interpreter of the agent, transforming the selected plans into intentions. *Phung et al.* [20] apply decision trees for BDI agents using a learning-based framework, where the learning component is added to the BDI cycle. The agent processes its past experience to adapt it to the current behavior with respect to background knowledge. The result of the learning algorithm is then added to the beliefs of the agent. In the work of *Airiau et al.* [21], the BDI agent is investigated to learn from past experience by preventing failed plan executions. In an initial step, the relation of *goals* and *plans* is represented by means of *goal-plan trees*. A goal-plan tree contains the defined goals and their corresponding plans of a BDI agent, leading to a hierarchical tree structure with goals and possible sub-plans. In the thesis of *Singh* [22], the plan selection step in the BDI cycle is tackled with different approaches. Multiple works related to the author are therefore considered. The works of *Singh et al.* [23, 24] build upon the previous paper [21] and add *context conditions* for the plan selection process in form of decision trees. In common, a context condition is a Boolean function that needs to be predetermined during the implementation phase. It is attached to each plan and describes the conditions and whether a plan is useful to a corresponding goal in a specific situation. Focusing on the learned behavior, the decision tree is built up for each considered plan in the agent's library. Each tree, therefore, leads to a decision of whether the plan will be successful or fail with a probability score. A further extension of this work is from *Singh et al.* [25], where plan selection considering changing dynamics is investigated. A confidence measure function for the degree of *stability of plans* is presented with respect to execution traces and the environment of the agents. The resulting weights are added to the plans denoting the success of being applied for a corresponding goal.

#### 4.2. BDI and Reinforcement Learning

The thesis of *Feliu* [26] considers the application of Reinforcement Learning (RL) for generating plans in BDI agents without relying on earlier knowledge. The author covers some related works concerning BDI and learning, which are also objects of this survey. Related to this setting, where RL is applied for BDI are from *Pereira & Dimuro* [27, 28] and *Chang* [29]. In these works, BDI and RL are combined by integrating learning into the BDI cycle. The work of *Qi & Bo-ying* [30] investigates a combination of RL and BDI for robot soccer simulation. Here, RL is considered as a feedback process by using the Q-Learning algorithm for the simulation steps. The learning algorithm is not integrated into the BDI architecture but processes the outcome of the BDI agent's action. Another approach in the same setting is presented by *Wan et al.* [31] where a BDI agent is extended with Q-Learning in AgentSpeak, which is the language definition instance the agents use to communicate with each other. More specifically, the plan library is improved by the Q-learning decision algorithm in an uncertain environment. What they found out is, that in state space exploration, which is the obligatory step in RL, the communication of AgentSpeak slowed down. For faster convergence, Deep Reinforcement Learning (DRL)

seems to be a suitable approach. The latter is also mentioned in section 5. Action selection based on rules is a challenge inside this area which is tackled by *Broekens & Hindriks* [16]. In this work, the authors use RL for the *rule selection step*, which slightly differs from the action selection process. In the typical RL setting, the learned behavior is the corresponding action. In contrast in this work, an internal uninstantiated rule is selected during the learning process. They consider the *GOAL* agent programming language. The relevant components for learning are reflected in the states, which are built up with a set of rules of the agents and the number of active goals. The considered state representation seems to be an initial version for learning but is capable to deliver interesting results for rule selection. The learning process takes place inside the agent architecture.

Initial works of combining elements of the RL setting with *partial observability* has been investigated by *Rens et al.* [32]. Here, the authors combine the BDI architecture with the *Partially observable Markov decision process (POMDP)* plan approach providing initial results by considering small experimental settings. They argue in favor of a more complex simulation environment. For this approach, *Chen et al.* [33] integrate the POMDP into the planning phase of the BDI architecture by considering *AgentSpeak*. *Nair & Tambe* also investigate in [34] the concept of POMDP for the BDI paradigm. They consider Multi-agent teaming by POMDP and *team-oriented programming*. Another work concerning this specification is from *Rens & Moodley* [35], where the reward-maximizing approach of POMDP and the management of multiple goals in BDI systems are combined. These works open up opportunities for investigating RL and BDI in Multi-agent settings. *Bosello & Ricci* [36] extend the BDI architecture with RL. They consider *SARSA* algorithm for the decision-making of the agent. A Low-level learning approach represented in the *BDI-FALCON* agent architecture, which is presented in *Tan et al.* [37, 38]. At its lowest level, BDI-FALCON contains a reactive learning module based on *Temporal Difference Learning (TD)*, an RL algorithm that estimates a value function of state-action pairs  $Q(s, a)$  that indicates the learning step of the system. Two other modules contain the BDI-native components like goals and plans which are sent to the low-level RL environment. *Karim et al.* [39] propose an approach, where learning with a high level of abstraction by a BDI agent is connected to a low-level RL environment, based on BDI-FALCON [37]. Resulting in a hybrid architecture, the BDI agent generates plans that are derived from the RL environment. *Norling* [40] integrates the Q-Learning algorithm into the BDI cycle to learn rules for pathfinding in a grid world. It is evaluated in a simple grid environment. *Subagdja & Sonenberg* [41] also integrate Q-Learning algorithm into the BDI agent cycle. They introduce *meta-level* plans which are considered for monitoring, the reasoning step, and the monitoring of plans executed. *Badica et al.* [42] apply TD for BDI agents. Considering a grid scenario, they define the agent's actions as well as specific states representing the corresponding goals. *Singh & Hindriks* investigate in [43] the Q-Learning algorithm for adaptive behaviors in autonomous BDI agents.

### 4.3. Literature categorization overview

In table 1, we have listed a selection of the research works handled in this survey and classified them with respect to the ML-COG dimensions neglecting mentioned other surveys in the previous section. For future research in this area, the open research challenges in Section 5 can be considered. For the sake of clarity, we set up the columns reflecting the dimensions of

**Table 1**

Overview of selected surveyed literature based on ML-COG

| Work                            | SA/MA | Learning            | Integration     | Objective               |
|---------------------------------|-------|---------------------|-----------------|-------------------------|
| Airiau et al., 2009 [21]        | SA    | Decision Tree       | Hard            | Plan selection          |
| Badica et al., 2012 [42]        | SA    | TD-Learning (RL)    | Hard            | RL-BDI agent            |
| Bosello & Ricci, 2019 [36]      | SA    | SARSA (RL)          | Hard & Soft     | RL-BDI agent            |
| Broekens et al., 2012 [16]      | SA    | Model-based RL      | Hard            | Rule selection          |
| Feliu, 2013 [26]                | SA    | Q-Learning (RL)     | Hard & Soft     | Plan generation         |
| Heinze et al., 1999 [44]        | SA    | CLARET (SL)         | Loosely coupled | Plan recognition        |
| Hernandez et al., 2004 [5]      | SA/MA | Decision Tree       | Hard            | Plan execution          |
| Karim et al., 2006-a [39]       | SA    | RL                  | Loosely coupled | Plan execution          |
| Karim et al., 2006-b [45]       | SA    | RL                  | Loosely coupled | Plan execution          |
| Lokuge & Alahakoon, 2007 [46]   | SA    | KAM                 | Hard & Soft     | Intention selection     |
| Norling, 2001 [47]              | SA    | RPDM                | (?)             | Decision Making         |
| Norling, 2004 [40]              | SA    | RPDM/Q-Learning     | (?)             | Decision Making         |
| Phung et al., 2005 [20]         | SA    | Decision Tree       | Hard & Soft     | Experience learning     |
| Qi et al., 2009 [30]            | SA/MA | Q-Learning          | Hard & Soft     | Decision Making         |
| Rodrigues et al., 2022 [6]      | SA    | Deep Neural Network | Hard & Soft     | Decision making         |
| Singh et al., 2010-a [24]       | SA    | Decision Tree       | Hard            | Plan selection          |
| Singh et al., 2010-b [23]       | SA    | Decision Tree       | Hard            | Plan selection          |
| Singh et al. 2011 [25]          | SA    | Decision Tree       | Hard            | Plan selection          |
| Singh & Hindriks, 2013 [43]     | SA    | Q-Learning          | Hard & Soft     | BDI-RL agent            |
| Subagdja & Sonenberg, 2005 [41] | SA    | Q-Learning          | Hard            | Learning Plans /Actions |
| Tan et al., 2011 [37]           | SA    | TD-Learning         | Loosely coupled | Plan selection          |
| Wan et al., 2018 [31]           | SA    | Q-Learning (RL)     | Hard            | RL-BDI agent            |

ML-COG. The second column contains the information, whether a Single-Agent (SA) or a Multi-Agent (MA) setting is considered. In the cube, SA is placed into AP and MA is placed into MAP. In addition, the last column *Objective* contains the contribution goal of the corresponding work. Note, that for the dimension *Cognitive agent development*, we distinguish between Single-agent (SA) and Multi-agent(MA) development. In the context of the survey, it is equally to AP and MAP, as explained in section 3. Note, that we have left out the works from the table, where learning approaches are not explicitly implemented or executed <sup>5</sup>.

## 5. Open research challenges

The research done so far in the intersection of ML and AOP provides many different applications, where some of which have been elaborated on in the previous section. The ML-COG cube is in an ongoing development process. It will be extended with finer-grained categorization scales thus being subject to future research. Since the categorization process follows the presented dimensions, we point out the following application areas, which are picked due to their technical proximity as well as based on the contributions and potential limitations in the investigated literature. Therefore, we list the following areas for future research:

1. Communication protocols and Emergent communication
2. Cognitive decision making and Learned behavior
3. Goal-level learning

<sup>5</sup>A ? entry denotes, that the implementation type is not clearly classifiable.



The overall aim is to provide a high level of abstraction with the usage of learning-based components. The areas 1 and 2 are intentionally formulated each with two extremes, indicating the different approaches to agent development in the programming phase. The first area ranges from predefined communication protocols like *AgentSpeak* [48] which is considered in a MAS over to emergent communication [49] in learning-based agents interacting with each other. Current research in emergent communication provides RL algorithms in Multi-agent settings to encourage agents to communicate with each other based on single and collective rewards. This area is important, especially in MAS where reliable communication leads to efficient coordination and cooperation. In table 1, one can see that nearly all works focus on the single-agent setting. The shift to MAS is therefore a crucial step in inspecting the behavior of BDI learning agents interacting with each other. A combination of learning-based communication with initial rules represents such a combination approach. The advantage overall is a better explainable learned behavior and thus the corresponding actions of the agents [50]. In the second area, we distinguish rather different agent types which are commonly considered in MAS as it is presented by *Russel & Norvig* in [4]. Decision-making is the essential step agents process to reach their goals successfully. The research in *Multi-agent Learning* based on RL algorithms has already covered a broad range of settings starting from single-agent settings to MAS settings with different applications [51]. Here, we see future work in the MAS settings based on cognitive decision-making based on the BDI architecture. Works covered in this survey already provide solution for the single-agent setting [36, 37]. Furthermore, the usage of Neural networks is rather a rarity and one recent work combining BDI agents and Neural networks are from *Rodrigues et al.* [6]. As a third area, we see learning at the Goal level as a novel approach to connecting ML and BDI. In the surveyed literature, learning at the Plan level is predominantly tackled but no Goal-level learning is presented. Sub-symbolic learning methods, like Neural Networks, could be therefore considered.

## 6. Conclusion

Learning methods in MAS differ from the traditional ML process since the autonomous and flexible behavior of the agents is considered and furthermore, interacting in a complex and dynamic environment. This survey aims to get in the lane at the intersection of ML and AOP by comprising the relevant work done over the years in the field. In ML research, the term *agent* is predominantly considered as a concept rather than an existing instance with explicit developed cognitive capabilities as it is in software agents [1]. Such a form of disambiguation also influences the contextual understanding of our work. In spite of the fact that this intersection is based on different approaches, cognitive software agents have not been considered sufficiently in ML research and therefore represent a relevant direction for current research. The analysis of such an integration process will lead to better insight into the functioning of learned behaviors in a cognitive framework.

## References

- [1] V. Dignum, F. Dignum, Agents are dead. long live agents!, in: Proceedings of the 19th AAMAS, 2020, pp. 1701–1705.
- [2] Y. Shoham, Why knowledge representation matters, *Communications of the ACM* 59 (2015) 47–49.
- [3] R. H. Bordini, A. El Fallah Seghrouchni, K. Hindriks, B. Logan, A. Ricci, Agent programming in the cognitive era, *Autonomous Agents and Multi-Agent Systems* 34 (2020).
- [4] S. J. Russell, P. Norvig, *Artificial Intelligence: a modern approach*, 3 ed., Pearson, 2009.
- [5] A. G. Hernández, A. El Fallah-Seghrouchni, H. Soldano, Learning in BDI Multi-agent Systems, in: J. Dix, J. Leite (Eds.), *Computational Logic in Multi-Agent Systems*, volume 3259, Springer Berlin Heidelberg, 2004, pp. 218–233.
- [6] R. Rodrigues, R. A. Silveira, R. D. Santiago, A Mediator Agent based on Multi-Context System and Information Retrieval, *ICAART 2022* (2022).
- [7] O. I. Erduran, M. Minor, L. Hedrich, A. Tarraf, F. Ruehl, H. Schroth, Multi-agent Learning for Energy-Aware Placement of Autonomous Vehicles, in: 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, Boca Raton, FL, USA, 2019, pp. 1671–1678.
- [8] M. J. Wooldridge, *An introduction to multiagent systems*, 2nd ed ed., John Wiley & Sons, Chichester, U.K, 2009.
- [9] J. Bryson, Cross-paradigm analysis of autonomous agent architecture, *Journal of Experimental & Theoretical Artificial Intelligence* 12 (2000) 165–189.
- [10] M. Minsky, Logical Versus Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy, in: *AI Magazine Volume 12 Number 2*, 1991.
- [11] L. d. Silva, F. Meneguzzi, B. Logan, BDI Agent Architectures: A Survey, in: Proceedings of the Twenty-Ninth IJCAI, Yokohama, Japan, 2020, pp. 4914–4921.
- [12] A. Deljoo, T. M. van Engers, L. Gommans, C. T. de Laat, et al., What is going on: Utility-based plan selection in bdi agents., in: *AAAI Workshops*, 2017.
- [13] Y. Shoham, Agent-oriented programming, *Artificial Intelligence* 60 (1993) 51–92.
- [14] R. H. Bordini, M. Dastani, J. Dix, A. El Fallah Seghrouchni (Eds.), *Multi-Agent Programming: : Languages, Tools and Applications*, Springer US, 2009.
- [15] A. Sturm, O. Shehory, *Agent-Oriented Software Engineering: Revisiting the State of the Art*, Springer Berlin Heidelberg, 2014, pp. 13–26.
- [16] J. Broekens, K. Hindriks, P. Wiggers, Reinforcement Learning as Heuristic for Action-Rule Preferences, in: R. Collier, J. Dix, P. Novák (Eds.), *Programming Multi-Agent Systems*, volume 6599, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 25–40.
- [17] S. Nason, J. E. Laird, Soar-RL: integrating reinforcement learning with Soar, *Cognitive Systems Research* 6 (2005) 51–59.
- [18] A. Hernandez, A. El Fallah-Seghrouchni, H. Soldano, Distributed learning in intentional bdi multi-agent systems, in: Proceedings of the Fifth Mexican International Conference in Computer Science, 2004. ENC 2004., 2004, pp. 225–232.
- [19] A. G. Hernandez, A. E.-F. Seghrouchni, H. Soldano, Bdi multiagent learning based on first-order induction of logical decision trees, in: *Intelligent Agent Technology: Research and Development*, World Scientific, 2001, pp. 160–169.

- [20] T. Phung, M. Winikoff, L. Padgham, Learning Within the BDI Framework: An Empirical Analysis, in: Knowledge-Based Intelligent Information and Engineering Systems, volume 3683, Springer Berlin Heidelberg, 2005, pp. 282–288.
- [21] S. Airiau, L. Padgham, S. Sardina, S. Sen, Enhancing the Adaptation of BDI Agents Using Learning Techniques, *International Journal of Agent Technologies and Systems* 1 (2009) 1–18.
- [22] D. Singh, Learning plan selection for BDI agent systems, Ph.D. thesis, RMIT University, 2011.
- [23] D. Singh, S. Sardina, L. Padgham, S. Airiau, Learning Context Conditions for BDI Plan Selection, *AAMAS* (2010).
- [24] D. Singh, S. Sardina, L. Padgham, Extending BDI plan selection to incorporate learning from experience, *Robotics and Autonomous Systems* 58 (2010) 1067–1075.
- [25] D. Singh, S. Sardina, L. Padgham, G. James, Integrating learning into a bdi agent for environments with changing dynamics, in: *Twenty-Second IJCAI*, 2011.
- [26] J. Feliu, Use of Reinforcement Learning for Plan Generation in Belief-Desire-Intention (BDI) Agent Systems, Ph.D. thesis, University of Rhode Island, Kingston, RI, 2013. doi:10.23860/thesis-feliu-jose-2013.
- [27] D. R. Pereira, L. V. Goncalves, G. P. Dimuro, Constructing BDI Plans from Optimal POMDP Policies, with an Application to AgentSpeak Programming, *Conferencia Latinoamericana de Informatica, CLEI* (2008) 11.
- [28] D. R. Pereira, G. P. Dimuro, Um algoritmo para extração de um plano bdi que obedece uma politica mdp otima, in: *Anais do Workshop-Escola de Sistemas de Agentes para Ambientes Colaborativos, Pelotas, PPGINF/UCPel*, 2007.
- [29] S. Chang, Simulacao de ambientes multiagente normativos, *Workshop-Escola de Sistemas de Agentes para Ambientes Colaborativos WESAAC* (2007).
- [30] G. Qi, W. Bo-ying, Study and Application of Reinforcement Learning in Cooperative Strategy of the Robot Soccer Based on BDI Model, *International Journal of Advanced Robotic Systems* 6 (2009) 15.
- [31] Q. Wan, W. Liu, L. Xu, J. Guo, Extending the BDI Model with Q-learning in Uncertain Environment, in: *International Conference on Algorithms, Computing and Artificial Intelligence*, ACM, Sanya China, 2018, pp. 1–6.
- [32] G. Rens, A. Ferrein, E. Van Der Poel, A BDI Agent Architecture for a POMDP Planner, *9th International Symposium on Logical Formalization of Commonsense Reasoning: Commonsense* (2009).
- [33] Y. Chen, K. Bauters, W. Liu, J. Hong, K. McAreavey, L. Godo, C. Sierra, Agentspeak+: Agentspeak with probabilistic planning, *Proc. of CIMA* (2014) 15–20.
- [34] R. Nair, M. Tambe, Hybrid BDI-POMDP Framework for Multiagent Teaming, *Journal of Artificial Intelligence Research* 23 (2005) 367–420.
- [35] G. Rens, D. Moodley, A hybrid POMDP-BDI agent architecture with online stochastic planning and plan caching, *Cognitive Systems Research* 43 (2017) 1–20.
- [36] M. Bosello, A. Ricci, From Programming Agents to Educating Agents – A Jason-Based Framework for Integrating Learning in the Development of Cognitive Agents, in: L. A. Dennis, R. H. Bordini, Y. Lespérance (Eds.), *Engineering Multi-Agent Systems*, volume 12058, Springer International Publishing, Cham, 2020, pp. 175–194.

- [37] A.-H. Tan, Y.-S. Ong, A. Tapanuj, A hybrid agent architecture integrating desire, intention and reinforcement learning, *Expert Systems with Applications* 38 (2011).
- [38] A.-H. Tan, Falcon: a fusion architecture for learning, cognition, and navigation, in: 2004 IEEE International Joint Conference on Neural Networks (IEEE), volume 4, 2004, pp. 3297–3302 vol.4.
- [39] S. Karim, L. Sonenberg, A.-H. Tan, A Hybrid Architecture Combining Reactive Plan Execution and Reactive Learning, in: *PRICAI 2006: Trends in Artificial Intelligence*, volume 4099, Springer Berlin Heidelberg, 2006, pp. 200–211.
- [40] E. Norling, Folk psychology for human modelling: Extending the bdi paradigm, in: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2004, pp. 202–209.
- [41] B. Subagdja, L. Sonenberg, Learning plans with patterns of actions in bounded-rational agents, Springer-Verlag (2005) 30–36.
- [42] A. Badica, C. Badica, M. Ivanovic, D. Mitrovic, An Approach of Temporal Difference Learning Using Agent-Oriented Programming, in: *20th International Conference on Control Systems and Computer Science*, IEEE, 2015, pp. 735–742.
- [43] D. Singh, K. V. Hindriks, Learning to Improve Agent Behaviours in GOAL, in: *Programming Multi-Agent Systems*, volume 7837, Springer Berlin Heidelberg, 2013, pp. 158–173.
- [44] C. Heinze, S. Goss, A. Pearce, Plan recognition in military simulation: Incorporating machine learning with intelligent agents, in: *Proceedings of IJCAI-99 Workshop on Team Behaviour and Plan Recognition*, Citeseer, 1999, pp. 53–64.
- [45] S. Karim, B. Subagdja, L. Sonenberg, Plans as Products of Learning, in: *2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, IEEE, Hong Kong, China, 2006, pp. 139–145.
- [46] P. Lokuge, D. Alahakoon, Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents, *European Journal of Operational Research* 177 (2007) 1985–2015.
- [47] E. Norling, Learning to notice: Adaptive models of human operators, in: *Second International Workshop on Learning Agents*, Citeseer, 2001.
- [48] R. H. Bordini, J. F. Hübner, M. Wooldridge, *Programming multi-agent systems in AgentSpeak using Jason*, John Wiley & Sons, 2007.
- [49] M. Noukhovitch, T. LaCroix, A. Lazaridou, A. Courville, Emergent communication under competition, in: *Proceedings of the 20th International Conference on AAMAS*, International Foundation for Autonomous Agents and Multiagent Systems, UK, 2021, p. 974–982.
- [50] J. Broekens, M. Harbers, K. Hindriks, K. v. d. Bosch, C. Jonker, J.-J. Meyer, Do you get it? user-evaluated explainable bdi agents, in: *German Conference on Multiagent System Technologies*, Springer, 2010, pp. 28–39.
- [51] S. Gronauer, K. Diepold, Multi-agent deep reinforcement learning: a survey, *Artificial Intelligence Review* 55 (2022) 895–943.