Agent Programming in the Cognitive Era

JAAMAS Track

Rafael H. Bordini
PUCRS
rafael.bordini@pucrs.br

Amal El Fallah Seghrouchni
Sorbonne Université
Amal.Elfallah@lip6.fr

Koen Hindriks
Vrije Universiteit Amsterdam
k.v.hindriks@vu.nl

Brian Logan
University of Nottingham
b.s.logan@uu.nl

Alessandro Ricci
University of Bologna
a.ricci@unibo.it

ABSTRACT

It is claimed that, in the nascent 'Cognitive Era', intelligent systems will be trained using machine learning techniques rather than programmed by software developers [10]. A contrary point of view argues that machine learning has limitations, and, taken in isolation, cannot form the basis of autonomous systems capable of intelligent behaviour in complex environments [14]. In this paper, we argue that the unique strengths of Belief-Desire-Intention (BDI) agent programming languages provide an ideal framework for integrating the wide range of AI capabilities necessary for progress towards the next-generation of intelligent systems.

ACM Reference Format:
Rafael H. Bordini, Amal El Fallah Seghrouchni, Koen Hindriks, Brian Logan, and Alessandro Ricci. 2021. Agent Programming in the Cognitive Era: JAAMAS Track. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 BDI AGENTS

The predominant approach in agent programming is inspired by the Belief-Desire-Intention (BDI) model [4]. In BDI agents [16] the behaviour of an agent is specified in terms of beliefs, goals, and plans. *Beliefs* represent the agent's information about itself, the environment, and other agents. *Goals* represent a desired course of action or state of the environment the agent is trying to bring about. *Plans* are the means by which the agent can achieve its goals. Plans are typically predefined by the agent developer, and consist of primitive actions that directly change the state of the environment and subgoals which are in turn achieved by subplans. At run-time, an *interpreter* updates the agent's beliefs and goals in response to messages and sensory information from the agent's environment (percepts), and manages the agent's intentions. An *intention* is a future course of action the agent is committed to carrying out. In practice, an intention is often implemented as a stack of partially instantiated plans, the execution of which is expected to achieve a (top-level) goal or respond to the change in the agent's beliefs (typically reflecting perceived changes in the environment or new information communicated by other agents). The interpreter is also responsible for choosing which intention to execute and for executing steps in the plan forming the top of the intention.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

BDI-based agent programming languages offer significant advantages for developing autonomous systems [13], including more rapid development [2], context sensitive, robust behaviour [21], and, critically, greater intelligibility and verifiability compared to agents where programs are learned [5, 6, 12, 15, 19]. Moreover, a BDI-based approach also facilitates the integration of a wide range of symbolic, stochastic and sub-symbolic AI techniques [8, 20, 22]. In [3], we briefly survey the historical development of BDI-based programming languages, and review previous work on integrating AI techniques (including machine learning) into BDI agents. In this paper, we briefly discuss some of the key open research problems and possible future research directions identified in [3]. In particular, we identify two ways AI techniques may be integrated into a BDI agent architecture, outline the challenges of engineering a BDI-based AI integration framework, and highlight some opportunities and open research challenges in this area. Although we focus on the unique strengths of the BDI approach, we believe our proposals are applicable, at least in part, to other approaches to AOP, particularly those languages in the broader cognitive agent tradition, e.g., GOAL [9], SOAR [11], that have similar logic-based or declarative "roots".

2 INTEGRATING AI INTO BDI AGENTS

Two main architectural strategies to integrate AI with BDI-based agent programming languages can be identified: (i) *AI as a service* (i.e., exogenous case), in which the AI is packaged as a separate, independent component, either running within the same (agent) system, or in a distributed fashion accessed through the network; and (ii) *AI embedded into agents* (i.e., endogenous case), in which AI components or techniques are used to augment or replace elements of the standard BDI architecture and/or interpreter cycle. In reality, these two strategies form the ends of a spectrum of possibilities, giving rise to a range of 'hybrid' approaches, in which some AI components are exogenous while others are endogenous. However, in the interest of brevity, we focus on purely exogenous and purely endogenous strategies here.

In the *AI as a service* approach, AI components such as external image/speech recognition systems, text-to-speech services, document analysis capabilities, etc., are modelled as part of the agents' *application environment* [17]. An agent program accesses and exploits the AI capability by means of (external or internal) actions and percepts. From an agent development perspective, this is conceptually similar to the way in which an agent program interacts with the underlying agent hardware/platform and the agent's environment. The agent developer is responsible for integrating the AI service into the agent program: invoking the appropriate service(s) at the appropriate point in the execution of the agent, and writing code to exploit the expanded set of percepts (represented as beliefs) made available by the AI service. Such an approach has the advantage of being easy to integrate into existing agent platforms and development methodologies, while at the same time facilitating the development of sophisticated applications. For example, a 'personal assistant' agent involving text or speech interaction that previously required specialist development expertise [23] will increasingly fall within the scope of an average agent developer.

In the short term, the AI as a service approach offers the opportunity to develop significantly more capable agent applications within the standard BDI model. Sensing and behaviour that would previously have required specialist programming can be easily integrated into existing agent programming platforms and exploited by application developers. Moreover, exploiting such capabilities does not require any modifications to standard agent development methodologies. However, while the opportunities offered by such an approach are significant, further research is required to determine whether current environment-oriented frameworks such as CArtAgO [18] and EIS [1] provide the support needed from a software engineering perspective for the effective integration of a broad and diverse range of AI services, for example those requiring large amounts of sub-symbolic information, or those services that return essentially "control information" (e.g., which plan to adopt, which intention to progress). Moreover, the need to explicitly invoke a service by an action in a plan in this approach may make it difficult to integrate techniques such as machine learning, except in the form of pre-trained components.

In the *AI embedded into agents* approach, the aim is to raise the level of abstraction of agent programming by increasing the basic 'competence' of the agent language or platform. For example, AI components may be used to induce appropriate context conditions for plans, to learn which applicable plan is most appropriate in a given situation, or to manage potential conflicts between intentions. As the competence of the agent increases, the role of the agent developer changes from programming exactly what the agent will do in all situations to providing more strategic information, heuristics, advice, social knowledge (e.g., norms), etc. about what the agent should (or should not) do in a given situation, leaving the details of the implementation of the strategy to the agent. This style of development has some similarities with the 'training' of systems advocated in the 'cognitive computing' approach, and one can envision hybrid architectures in which some behaviours are learnt, while others, perhaps those with a supervisory or high-level decision making role, are 'programmed'.

For many applications, particularly those involving interaction with humans, there are standard (often codified) ways of approaching a task that must be followed for safety, regulatory, quality control, or other reasons. While machine learning (or other AI techniques such as planning) can be used to adapt the behaviour of the agent, e.g., to a particular user or to generate a novel implementation of a high-level action, the critical aspects of the agent's behaviour are required to fall within a particular envelope or follow a particular pattern. We believe BDI-based agent programming approaches enhanced with AI techniques offer a fruitful framework for such 'controlled adaptation', as the structuring of BDI agent programs in terms of goals and plans allows different (sub)goals to be associated with differing degrees of adaptation in a natural way. For example, the means used to achieve some goals may be precisely specified by developer-supplied plans, while the means used to achieve other goals may be learned, or synthesised at run-time using first-principles AI planning techniques.

3 FUTURE RESEARCH DIRECTIONS

There is a very large space of such extended BDI architectures. One way to structure future research in this area is to explore where and how AI techniques can be embedded *within the BDI model itself.* It seems plausible that the BDI approach of determining which plan to adopt based on the agent's current beliefs should apply equally to the problem of selecting an appropriate deliberation strategy given the agent's current state. Such an approach has the potential to transfer the intelligibility of the intention-driven BDI model to the embedding of AI.

Key to this approach to embedding AI into BDI agent programs is determining what features of intentions and the context of their execution are necessary to implement the BDI cycle for a particular application. This includes such issues as when a goal should be adopted or dropped, which plan to use to achieve a goal, when the plan should next be executed, etc., but also many other issues relating to the social context of the agent, e.g., the expectations of humans and other agents, the prevalent norms, ethics, and values and how these should determine the behaviour of an agent. The fact that, in the BDI model, these issues must always be addressed in the context of other intentions with potentially differing characteristics is a unique strength of the BDI approach, in that it necessitates the adoption of a holistic view of the problem of developing autonomous intelligent systems.

Alongside the purely architectural questions of how the overall problem of intelligent behaviour should be broken down (e.g., what are the most appropriate components/APIs), work in this direction gives rise to a range of new research problems centred around the notion of *bounded adaptation*. How should the split between programmer-determined fixed or canonical behaviours and agent-determined adaptations of these behaviours (e.g., refinements, or implementations of (very) high-level actions, etc.) be characterised? What development methodologies and verification approaches can be used to specify and certify the behaviour of agents that integrate significant AI capabilities into their decision making? This can be seen as establishing a new strand of research exploring *hybrids* of the programming-based, learning-based, and model-based approaches to developing AI capabilities identified in [7]. We believe future BDI-based agent programming approaches will offer a fruitful framework for such controlled adaptation.

REFERENCES

[1] Tristan M. Behrens, Koen V. Hindriks, and Jürgen Dix. 2011. Towards an environment interface standard for agent platforms. *Ann. Math. Artif. Intell.* 61, 4 (2011), 261–295.

[2] Steve S. Benfield, Jim Hendrickson, and Daniel Galanti. 2006. Making a strong business case for multiagent technology. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006).* ACM, 10–15.

[3] Rafael H. Bordini, Amal El Fallah Seghrouchni, Koen Hindriks, Brian Logan, and Alessandro Ricci. 2020. Agent programming in the cognitive era. *Autonomous Agents and Multi-Agent Systems* 34, 37 (2020).

[4] M. E. Bratman, D. J. Israel, and M. E. Pollack. 1988. Plans and Resource-Bounded Practical Reasoning. *Computational Intelligence* 4, 4 (1988), 349–355.

[5] Joost Broekens, Maaike Harbers, Koen V. Hindriks, Karel van den Bosch, Catholijn M. Jonker, and John-Jules Ch. Meyer. 2010. Do You Get It? User-Evaluated Explainable BDI Agents. In *Multiagent System Technologies, 8th German Conference, MATES 2010, Leipzig, Germany, September 27-29, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6251)*, Jürgen Dix and Cees Witteveen (Eds.). Springer, 28–39.

[6] Louise A. Dennis, Michael Fisher, Matthew P. Webster, and Rafael H. Bordini. 2012. Model Checking Agent Programming Languages. *Automated Software Engineering* 19, 1 (2012), 5–63.

[7] Hector Geffner. 2010. The Model-Based Approach to Autonomous Behavior: A Personal View. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-2010)*. AAAI Press, 1709–1712.

[8] Alejandro Guerra-Hernández, Amal El Fallah Seghrouchni, and Henry Soldano. 2004. Learning in BDI Multi-agent Systems. In *CLIMA 2004 - 4th International Workshop on Computational Logic in Multi-Agent Systems (Lecture Notes in Computer Science, Vol. 3259)*. Springer, 218–233.

[9] Koen V. Hindriks. 2009. Programming Rational Agents in GOAL. In *Multi-Agent Programming: Languages, Tools and Applications*, Amal El Fallah Seghrouchni, Jürgen Dix, Mehdi Dastani, and Rafael H. Bordini (Eds.). Springer, 119–157.

[10] John E. Kelly and Steve Hamm. 2013. *Smart Machines: IBM's Watson and the Era of Cognitive Computing*. Columbia University Press.

[11] J. E. Laird, A. Newell, and P. S. Rosenbloom. 1987. SOAR: An Architecture for General Intelligence. *Artificial Intelligence* 33 (1987), 1–64.

[12] John Bruntse Larsen. 2018. Agent Programming Languages and Logics in Agent-Based Simulation. In *Modern Approaches for Intelligent Information and Database Systems, - 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018, Extended Posters (Studies in Computational Intelligence, Vol. 769)*, Andrzej Sieminski, Adrianna Kozierkiewicz, Manuel Nunez, and Quang-Thuy Ha (Eds.). Springer, 517–526.

[13] Brian Logan. 2015. A Future for Agent Programming. In *Engineering Multi-Agent Systems: Third International Workshop, EMAS 2015, Istanbul, Turkey, May 5, 2015, Revised, Selected, and Invited Papers*. Springer, 3–17.

[14] Gary Marcus. 2018. Deep Learning: A Critical Appraisal. *CoRR* abs/1801.00631 (2018). arXiv:1801.00631 http://arxiv.org/abs/1801.00631

[15] Emma Norling. 2004. Folk Psychology for Human Modelling: Extending the BDI Paradigm. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)* (New York, New York). IEEE Computer Society, 202–209.

[16] A. S. Rao and M. P. Georgeff. 1991. Modeling Rational Agents within a BDI-architecture. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*. 473–484.

[17] Alessandro Ricci, Michele Piunti, and Mirko Viroli. 2011. Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems* 23, 2 (2011), 158–192.

[18] Alessandro Ricci, Michele Piunti, Mirko Viroli, and Andrea Omicini. 2009. Environment Programming in CArtAgO. In *Multi-Agent Programming, Languages, Tools and Applications.*, Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni (Eds.). Springer, 259–288.

[19] David Scerri, Sarah L. Hickmott, and Lin Padgham. 2012. User understanding of cognitive processes in simulation: a tool for exploring and modifying. In *Winter Simulation Conference, WSC '12, Berlin, Germany, December 9-12, 2012*, Oliver Rose and Adelinde M. Uhrmacher (Eds.). WSC, 240:1–240:12.

[20] Changyun Wei and Koen V. Hindriks. 2013. An Agent-Based Cognitive Robot Architecture. In *Programming Multi-Agent Systems: 10th International Workshop, ProMAS 2012, Valencia, Spain, June 4-8, 2012. Revised Selected Papers*, Mehdi Dastani, Jomi F. Hübner, and Brian Logan (Eds.). Springer, 54–71.

[21] Michael Winikoff, Lin Padgham, James Harland, and John Thangarajah. 2002. Declarative & Procedural Goals in Intelligent Agent Systems. In *Proceedings of the Eighth International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002*. Morgan Kaufmann, 470–481.

[22] Yuan Yao and Brian Logan. 2016. Action-Level Intention Selection for BDI Agents. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*. IFAAMAS, 1227–1235.

[23] Neil Yorke-Smith, Shahin Saadati, Karen L. Myers, and David N. Morley. 2012. The Design of a Proactive Personal Agent for Task Management. *International Journal on Artificial Intelligence Tools* 21, 1 (2012).