A Multi-level Explainability Framework for Engineering and Understanding BDI Agents

JAAMAS Track

Elena Yan
Mines Saint-Etienne
Saint-Etienne, France
elena.yan@emse.fr

Samuele Burattini
Alma Mater Studiorum, University of Bologna
Cesena, Italy
samuele.burattini@unibo.it

Jomi F. Hübner
Federal University of Santa Catarina
Florianópolis, Brazil
jomi.hubner@ufsc.br

Alessandro Ricci
Alma Mater Studiorum, University of Bologna
Cesena, Italy
a.ricci@unibo.it

ABSTRACT

As the complexity of software systems rises, the ability to provide explanations of system behavior has become a desirable property for any Artificial Intelligence based system, including autonomous multi-agent systems. While explainability is mainly explored to increase trust and understanding for end-users, it is also an interesting property from a software engineering perspective, supporting developers and designers in the debugging and validation phases. To address the different needs and expertise of these roles, we propose a framework that generates explanations at multiple levels of abstraction, enabling both the engineering and the understanding of BDI agents.

KEYWORDS

Explainable Agents; Explainability; BDI Agents

ACM Reference Format:
Elena Yan, Samuele Burattini, Jomi F. Hübner, and Alessandro Ricci. 2026. A Multi-level Explainability Framework for Engineering and Understanding BDI Agents: JAAMAS Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/ZIQG2381

1 INTRODUCTION

Explainability, i.e., the ability of a system to provide explanations of its behavior, has emerged as a desired property for Artificial Intelligence (AI) systems [18, 29]. Among these, multi-agent systems play a prominent role [28]. As intelligent agents become more autonomous and integrated in real-world systems, their ability to explain their behavior and the reasoning behind their decisions is essential in order to be trusted by humans [3, 15, 17].

Explainability has been explored not only from an end-user perspective to understand and trust the system (e.g., [7, 11, 14, 16, 27]), but also from an engineering perspective [22, 26], e.g., developers can use explanations to spot and debug runtime problems (e.g., [1, 13, 25]), or designers can use explanations to validate the system behavior according to the system requirements (e.g., [2, 8, 24]). When adopting an engineering perspective, explainability is strongly influenced by the agent architecture and programming language, as explanations need to be expressed in terms of their underlying abstractions. As a result, explanations should be tailored to different roles and needs.

We propose a *multi-level explainability framework* that presents explanations at three levels of abstraction: (i) Implementation Level, based on agents' programming language abstractions and targeting developers, (ii) Design Level, based on agents' design architecture abstractions and targeting designers, and (iii) Domain Level, based on the domain knowledge and targeting end-users. We select the BDI architecture [6, 20] as a reference abstraction to build explanations for the Design Level and Jason [5] programming language for the Implementation Level. We discuss how the Domain Level can be supported through use cases [9], user and system stories [23].

Explanations are generated from execution logs that capture relevant agent events, and the knowledge about the semantics underlying the specific level. These events are mapped from lower levels to higher levels and presented as narratives, with concepts depending on the level of abstraction. The explanation of an event is given by another set of events associated with its causal link. Users can request explanations at any level of abstraction, depending on the desired granularity. We present a prototype that supports explanation at Implementation and Design Levels, leaving the Domain Level for future work.

This paper summarizes the contribution published in *Autonomous Agents and Multi-Agent Systems* [30].

2 DIFFERENT ROLES, DIFFERENT LEVELS OF EXPLANATION

Different user roles require explanations expressed at the appropriate levels of abstraction. *Developers* require explanations addressing the inner workings and technological aspects of the MAS. *Designers* require explanations addressing the agent system's principles and architecture to validate the system requirements. *Domain experts* and *end users* require explanations related to the domain requirements, abstracting from the agent's architecture and technology.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/ZIQG2381

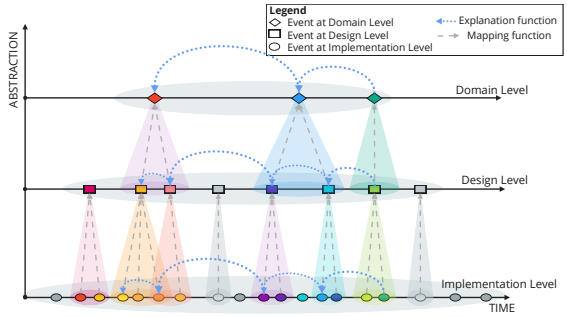

Figure 1: The idea of multiple levels of abstraction for the explanation of a (computing) system.

From the needs of these different roles, we propose the following levels of abstraction in generating the explanations:

- *Implementation Level*, which generates explanations closely related to the technology used to implement MAS (e.g., specific agent-oriented programming language);
- *Design Level*, which generates explanations focusing on the principles and architectures adopted to design the system (e.g., specific cognitive architecture), abstracting from the low-level implementation details; and
- *Domain Level*, which generates explanations focusing on the functional and non-functional requirements of the system as defined with stakeholders and domain experts, dealing with domain-specific knowledge and insights.

This is the underlying idea of our *multi-level explainability framework*, a framework that builds explanations of BDI agents at multiple levels of abstraction.

3 BUILDING NARRATIVES AT MULTIPLE LEVELS OF ABSTRACTION

A traditional way to collect information about the running system is by using logs to capture the events related to the agent's reasoning and behavior. From these events, we can generate sentences of the agent's behavior using the abstractions at a certain level. We refer to a *narrative*, a story of the agent's behavior composed of a sequence of events that are presented in natural language sentences. An *explanation* of a certain event is given by a subset of such a narrative composed of events that cause the event to explain. Users can select the suitable level of abstraction, read the narrative, ask, and receive explanations of specific events. In order to build narratives at multiple levels above the system logs (Fig. 1), we identify the following elements at each level: (i) a *set of events* that encapsulates the agent's behavior based on the abstraction level, (ii) an *explanation function* that defines the causal link among events of one level, and (iii) a *set of mapping functions* that defines the relations from lower-level events to build higher-level events.

At the Implementation Level, we choose Jason [5] as the reference agent-oriented programming language. The relevant events composing the narrative at this level concern the Jason agent's reasoning cycle and are built directly from the logs of the system. The events concern the perception of the environment, reception of messages, updates to the belief base and plan library, selection of a plan, goal lifecycle (i.e., created, suspended, removed), intentions lifecycle (i.e., created, waiting, suspended, removed), and actions (i.e., triggered, failed, finished). A first trivial explanation function can be based on goals, i.e., events are caused and explained by goals [12]. The explanation of an event is linked with the closest goal that can be identified by the same intention.

At the Design Level, we choose the BDI cognitive architecture [20], which refines Newell's knowledge level [19] by defining agents having knowledge and goals, with practical reasoning concepts [6]. Events at this level capture the agent's beliefs, goals, and intentions that are mapped and abstracted from the Implementation Level. To show the flexibility of different explanation functions at different levels and to exploit the semantics of the BDI architecture, we define the explanation function as the closest related event that causes the event to explain (e.g., an action executed is caused by a new intention, and a new intention is caused by a new goal).

The Domain Level requires further integration with other methodological approaches (e.g., use cases or user stories) to incorporate domain knowledge and capture system requirements at a higher level of abstraction. Following the established schema for describing use cases [9], domain events can be identified and generated accordingly. These events enable domain experts and end-users to validate system behavior against the use cases or system requirements and to request explanations when misalignments are detected [24].

4 CONCLUSION AND FUTURE WORK

The key idea of the contribution is to frame explainability for BDI agents at multiple levels, enabling stakeholders playing different roles (e.g., developers, designers, end-users) to request explanations about the (agent) system behavior at the proper level of abstraction. Among the necessary steps to validate the idea, an open-source prototype of the framework has been developed for the Implementation and Design Levels[1], along with an example to showcase the functionalities of the tool. The architecture includes a logger component, attached to each Jason agent to collect the logs, and a narrative generator component exposing a web interface to process the logs and build narratives.

This work opens several research directions. Our current focus is on BDI and Jason agents, but the modular approach to build narratives enables the reuse of the framework with other BDI-based agent technologies by adapting the Implementation Level abstractions and their mapping to the Design Level. Furthermore, the framework can be mapped to non-BDI, sub-symbolic, or generative LLM agents as well. On top of these agents' technologies, it would be useful to create a level that acts as a cognitive neck [21] to facilitate the explanation by exploiting cognitive concepts. Moreover, the current work focuses only on the agent dimension, generating explanations for individual agents. In the broader MAS community, there are other dimensions [4, 10], i.e., environment, interaction, and organization, that can also be explored to build explanations at multiple levels. This enables richer narratives that capture the interactions between agents, actions with possible failures in the environment, and the structural, functional, and normative aspects of the organization.

[1]Logger component: https://github.com/yan-elena/agent-logging and a narrative generator component: https://github.com/yan-elena/agent-explanation

REFERENCES

[1] Tobias Ahlbrecht. 2023. An algorithmic debugging approach for belief-desire-intention agents. *Annals of Mathematics and Artificial Intelligence* 92, 4 (May 2023), 797–814. https://doi.org/10.1007/s10472-023-09843-4

[2] Ahmad Alelaimat, Aditya Ghose, and Hoa Khanh Dam. 2023. Mining and Validating Belief-Based Agent Explanations. In *Explainable and Transparent AI and Multi-Agent Systems*, Davide Calvaresi, Amro Najjar, Andrea Omicini, Reyhan Aydogan, Rachele Carli, Giovanni Ciatto, Yazan Mualla, and Kary Främling (Eds.). Springer Nature Switzerland, Cham, 3–17. https://doi.org/10.1007/978-3-031-40878-6_1

[3] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. 2019. Explainable Agents and Robots: Results from a Systematic Literature Review. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1078–1088. https://doi.org/10.5555/3306127.3331806

[4] Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, and Alessandro Ricci. 2019. Dimensions in programming multi-agent systems. *The Knowledge Engineering Review* 34 (2019), e2. https://doi.org/10.1017/S026988891800005X

[5] Rafael Bordini, Jomi Hübner, and Michael Wooldridge. 2007. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. Vol. 8. John Wiley & Sons, Hoboken, NJ. https://doi.org/10.1002/9780470061848

[6] Michael Bratman. 1987. *Intention, Plans, and Practical Reason*. Cambridge: Harvard University Press, Cambridge, MA. https://doi.org/10.2307/2185304

[7] Joost Broekens, Maaike Harbers, Koen Hindriks, Karel Bosch, Catholijn Jonker, and John-jules 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 8*, Vol. 6251. Springer, Berlin, Heidelberg, 28–39. https://doi.org/10.1007/978-3-642-16178-0_5

[8] Álvaro Carrera, Carlos A. Iglesias, and Mercedes Garijo. 2013. Beast methodology: An agile testing methodology for multi-agent systems based on behaviour driven development. *Information Systems Frontiers* 16, 2 (July 2013), 169–182. https://doi.org/10.1007/s10796-013-9438-5

[9] Alistair Cockburn. 2000. *Writing Effective Use Cases* (1st ed.). Addison-Wesley Longman Publishing Co., Inc., USA.

[10] Yves Demazeau. 1997. Steps towards multi-agent oriented programming. In *First International Workshop on Multi Agent Systems*. Boston, Mass.

[11] Louise A. Dennis and Nir Oren. 2021. Explaining BDI Agent Behaviour through Dialogue. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 429–437. https://doi.org/10.5555/3463952.3464007

[12] Maaike Harbers, Karel van den Bosch, and John-Jules Meyer. 2010. Design and Evaluation of Explainable BDI Agents. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2010, Toronto, Canada, August 31 - September 3, 2010*, Jimmy Xiangji Huang, Ali A. Ghorbani, Mohand-Said Hacid, and Takahira Yamaguchi (Eds.). IEEE Computer Society Press, Toronto, Canada, 125–132. https://doi.org/10.1109/WI-IAT.2010.115

[13] Koen V. Hindriks. 2012. Debugging Is Explaining. In *PRIMA 2012: Principles and Practice of Multi-Agent Systems*, Iyad Rahwan, Wayne Wobcke, Sandip Sen, and Toshiharu Sugawara (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 31–45. https://doi.org/10.1007/978-3-642-32729-2_3

[14] D. N. Lam and K. S. Barber. 2005. Comprehending Agent Software. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (The Netherlands) *(AAMAS '05)*. Association for Computing Machinery, New York, NY, USA, 586–593. https://doi.org/10.1145/1082473.1082562

[15] Pat Langley, Ben Meadows, Mohan Sridharan, and Dongkyu Choi. 2017. Explainable Agency for Intelligent Autonomous Systems. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017*, Satinder Singh and Shaul Markovitch (Eds.). AAAI Press, San Francisco, California, USA, 4762–4764. https://doi.org/10.1609/aaai.v31i2.19108

[16] Avleen Malhi, Samanta Knapic, and Kary Främling. 2020. Explainable Agents for Less Bias in Human-Agent Decision Making. In *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, Davide Calvaresi, Amro Najjar, Michael Winikoff, and Kary Främling (Eds.). Springer International Publishing, Cham, 129–146. https://doi.org/10.1007/978-3-030-51924-7_8

[17] Bertram F. Malle. 2004. *How the Mind Explains Behavior: Folk Explanations, Meaning, and Social Interaction*. The MIT Press, Cambridge, MA. https://doi.org/10.7551/mitpress/3586.001.0001

[18] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38. https://doi.org/10.1016/j.artint.2018.07.007

[19] Allen Newell. 1982. The Knowledge Level. *Artificial Intelligence* 18, 1 (1982), 87–127. https://doi.org/10.1016/0004-3702(82)90012-1

[20] Anand S Rao, Michael P Georgeff, et al. 1995. BDI agents: from theory to practice.. In *Proceedings of the First International Conference on Multiagent Systems*, Victor R. Lesser and Les Gasser (Eds.), Vol. 95. The MIT Press, Cambridge, MA, 312–319.

[21] Alessandro Ricci, Stefano Mariani, Franco Zambonelli, Samuele Burattini, and Cristiano Castelfranchi. 2024. The Cognitive Hourglass: Agent Abstractions in the Large Models Era. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2706–2711.

[22] Sebastian Rodriguez, John Thangarajah, and Andrew Davey. 2024. Design Patterns for Explainable Agents (XAg). In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024*, Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.). ACM, Richland, SC, 1621–1629. https://doi.org/10.5555/3635637.3663023

[23] Sebastian Rodriguez, John Thangarajah, and Michael Winikoff. 2021. User and System Stories: An Agile Approach for Managing Requirements in AOSE. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1064–1072. https://doi.org/10.5555/3463952.3464076

[24] Sebastian Rodriguez, John Thangarajah, and Michael Winikoff. 2023. A Behaviour-Driven Approach for Testing Requirements via User and System Stories in Agent Systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1182–1190. https://doi.org/10.5555/3545946.3598761

[25] Michael Winikoff. 2017. Debugging Agent Programs with Why? Questions. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems* (São Paulo, Brazil) *(AAMAS '17)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 251–259. https://doi.org/10.5555/3091125.3091166

[26] Michael Winikoff. 2024. Towards Engineering Explainable Autonomous Systems. In *Engineering Multi-Agent Systems - 12th International Workshop, EMAS 2024, Auckland, New Zealand, May 6-7, 2024, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 15152)*, Daniela Briola, Rafael C. Cardoso, and Brian Logan (Eds.). Springer, Cham, 144–155. https://doi.org/10.1007/978-3-031-71152-7_9

[27] Michael Winikoff, Galina Sidorenko, Virginia Dignum, and Frank Dignum. 2021. Why bad coffee? Explaining BDI agent behaviour with valuings. *Artificial Intelligence* 300 (2021), 103554. https://doi.org/10.1016/j.artint.2021.103554

[28] Michael Wooldridge. 2009. *An introduction to multiagent systems*. John wiley & sons, USA.

[29] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. 2019. Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. In *Natural Language Processing and Chinese Computing*, Jie Tang, Min-Yen Kan, Dongyan Zhao, Sujian Li, and Hongying Zan (Eds.). Springer International Publishing, Cham, 563–574. https://doi.org/10.1007/978-3-030-32236-6_51

[30] Elena Yan, Samuele Burattini, Jomi Fred Hübner, and Alessandro Ricci. 2025. A multi-level explainability framework for engineering and understanding BDI agents. *Autonomous Agents and Multi-Agent Systems* 39, 1 (2025), 9. https://doi.org/10.1007/S10458-025-09689-6