

Explainable Agents as Static Web Pages: UAV Simulation Example

Yazan Mualla^{1(⊠)}, Timotheus Kampik², Igor H. Tchappi^{1,4}, Amro Najjar³, Stéphane Galland¹, and Christophe Nicolle⁵

 ¹ CIAD, Univ. Bourgogne Franche-Comté, UTBM, 90010 Belfort, France {yazan.mualla,igor.tchappi,stephane.galland}@utbm.fr
² Department of Computing Science, Umeå University, 90187 Umeå, Sweden tkampik@cs.umu.se
³ AI-Robolab/ICR, Computer Science and Communications, University of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg amro.najjar@uni.lu

 ⁴ Faculty of Sciences, University of Ngaoundere, B.P. 454, Ngaoundere, Cameroon
⁵ CIAD, Univ. Bourgogne Franche-Comté, UB, 21000 Dijon, France christophe.nicolle@u-bourgogne.fr

Abstract. Motivated by the apparent societal need to design complex autonomous systems whose decisions and actions are humanly intelligible, the study of explainable artificial intelligence, and with it, research on explainable autonomous agents has gained increased attention from the research community. One important objective of research on explainable agents is the evaluation of explanation approaches in humancomputer interaction studies. In this demonstration paper, we present a way to facilitate such studies by implementing explainable agents and multi-agent systems that i) can be deployed as static files, not requiring the execution of server-side code, which minimizes administration and operation overhead, and *ii*) can be embedded into web front ends and other JavaScript-enabled user interfaces, hence increasing the ability to reach a broad range of users. We then demonstrate the approach with the help of an application that was designed to assess the effect of different explainability approaches on the human intelligibility of an unmanned aerial vehicle simulation.

Keywords: eXplainable Artificial Intelligence \cdot Engineering multi-agent systems \cdot Human-Computer Interaction

1 Introduction

Since the lack of interpretability of both black-box machine learning models and complex rule-based systems is a generally-acknowledged socio-technical problem, the research domain of eXplainable Artificial Intelligence (XAI) is gaining

© Springer Nature Switzerland AG 2020

Y. Mualla and T. Kampik—These two authors have contributed equally.

D. Calvaresi et al. (Eds.): EXTRAAMAS 2020, LNAI 12175, pp. 149–154, 2020. https://doi.org/10.1007/978-3-030-51924-7_9

increased attention from researchers of various disciplines. A particularly motivating factor is that emerging laws and regulations, most notably the European Union's GDPR, require that certain decisions of information systems must be humanly interpretable [3]. Recent works in the literature highlighted explainability as one of the cornerstones for building trustworthy responsible AI systems [11,12]. In this context, an obvious research frontier for the autonomous agents and Multi-Agent Systems (MAS) community is the design of explainable intelligent agents [8]. This frontier is explained by the fact that while intelligent agents have been established as a suitable technique for implementing autonomous highlevel control and decision-making in complex AI systems [13], there is still a need for these systems to be understood and trusted by the human users.

Considering that the growth of research on explainable agents is accelerating, contributions that empirically evaluate the proposed explainability approaches are still scarce [2,6]. In this regard, Agent-based Simulation (ABS) fits the requirements to implement such empirical evaluations. ABS is a set of interacting intelligent entities that models and executes, within an artificial environment, the real-world autonomous agents, their relationships, and interactions with the environment [13]. Consequently, ABS can be considered as a natural step forward towards better managing and evaluating the proposed explainability approaches in Human-Computer Interaction (HCI) empirical user studies.

To facilitate more research and bridge the gap between the theoretical proposed explainability approaches on the one hand, and the practical evaluation of such approaches on the other hand, this demonstration paper presents an ABS approach to engineer explainable agents and MAS prototypes for the specific purpose of empirical evaluation in HCI studies. The approach makes use of light-weight web technologies that facilitate rapid prototyping and allow for the deployment of agents and MAS as static web pages.

2 Explainable Agents as Static Web Pages

2.1 Motivation

The proposed approach to explainable agent prototyping and simulation development is to implement ABS as static web front ends. These web pages can be easily deployed to any device that serve or render web pages, and shared with a broad audience, for example as web links. From a technology perspective, the approach makes use of the *JS*-son library [5], which allows the creation of Belief-Desire-Intention (BDI) agents, as well as agents with other reasoning loops and MAS in a higher-level programming language with little learning and technology overhead. We argue that the proposed approach has the following advantages:

Ease of deployment. Because the program code consists only of static files that are to be provided by a web server, it can be deployed in a straightforward manner, without the need of a complicated installation routine or the requirement to have extensive permissions on the target server. In particular, the program files can be moved to traditional static file servers (*e.g.*, via

FTP upload or via the upload feature of a content management system), or integrated into light-weight developer operations-oriented tools and services $(e.g., \text{ continuous deployment to GitHub pages on push to a git repository's master branch).$

- **Reach.** The explainable ABS can be easily shared with any potential human user who can access the Internet with a recent web browser. While for running HCI studies, oftentimes video *vignettes* are created to allow for easier sharing [1], using such vignettes i) severely limits interactive features and ii) does not allow for convenient updates (minor changes, for example to the user interface, require re-recording the video(s)).
- **Scalability.** Because all program code is executed by the client, in particular by the browser of the corresponding end-user, applications developed with the proposed approach scale well; the server merely needs to provide (by all practical means: small) static files, which means researchers can host the applications essentially free of costs.

2.2 Architecture

The proposed approach is based on the following architecture design (depicted in Fig. 1). A MAS (or, in simple scenarios, a single agent) is engineered to run encapsulated in a web page. Note that single-user interactions/human-in-theloop approaches are possible, and even multi-user interactions can be realized with light-weight real-time communication technologies such as Web Real-Time Communications (webRTC) [4], albeit with a minimally invasive integration of server-side technologies. The state of the environment and all agents it contains is exposed to a User Interface (UI) manager component that processes the state and makes it available to the following components:

- A grid world displays the "physical state" of the environment, *i.e.* the position of agents and artifacts.
- A state table or tree-like structure provides an overview of relevant information that is not obvious from the grid world representation, *i.e.*, hidden properties like goals and internal states of agents.
- A **notification system** informs users about important events, *e.g.*, when agents diverge from their expected behavior. Notifications are displayed as visually invasive alerts that overly the rest of the user interface.
- Interaction controls (only in non-study mode, to not distract study participants) allow users to switch between different simulation modes and adjust simulation parameters.

3 UAV Simulation Example

The use of ABS for Unmanned Aerial Vehicles (UAVs) is gaining more interest in complex civil application scenarios where coordination and cooperation are necessary [7]. To provide a running example, let us describe how the approach we



Fig. 1. Explainable Agent-based Simulation architecture

introduce in this paper can be used in a UAVs simulation scenario¹. The study evaluates how different explainability approaches affect human intelligibility of a UAV delivery simulation. The simulation is provided in three modes, which represent different paths through the explanation generation process.

- **Basic mode.** It displays the current state of all agents, including their current goal (target destination and mission type) in a table-like overview that updates in real-time.
- Adaptive filter mode. It aggregates the most important information across agents; *i.e.*, users do not need to scan the table for relevant information, but can see at a glance which agents perform missions according to their expectations, and which agents are in possibly problematic states ("stranded", uncoordinated). When an agent enters such a state, an alert with the agent's ID and goal information is generated.
- Contrastive mode. Alerts are constructed using an *implicitly counterfactual* explanation scheme, following the structure Agent A is doing P [instead of Q] because of C, where P is the current behavior, Q is the presumably expected behavior, and C is the execution condition. This means [instead of Q] is implied by the alert and hence dropped from the text.

¹ The developed application has been used for a follow-up of the explainable agents HCI study presented by Mualla *et al.* [9, 10].

Figure 2 displays the simulation in an interactive test mode that allows for the manipulation of simulation parameters through the user interface². (In the study application, UI controls where hidden and simulation parameters were set via the simulation's URL parameters to avoid end-user distraction and interference).

After all agents' states have been collected, explanations, *i.e.* summaries that give an overview of the agents' beliefs, are generated. For this, the chosen modes (Basic, adaptive filter, or contrastive) will determine the processes to be executed.



Fig. 2. Explainable UAV simulation with UI controls.

4 Conclusion

In this demonstration paper, we have shown how explainable agent simulations can be deployed as static web pages. The presented approach serves as an example of how light-weight tools with a small development, deployment, and operations footprint can be utilized to: i rapidly develop agent/ABS prototypes in a widely-used higher-level programming language and ii roll-out these prototypes and simulations at scale to large and diverse user groups, in particular for the purpose of empirical validation. As future research, from an engineering perspective, it can be considered as valuable to extend the JS-son library, which forms the foundation of this demonstration, with additional, generically useful abstractions for implementing explainable reasoning-loop agents. For this, components that this work implement can be extracted and merged into JS-son. From HCI and XAI perspectives, it can be considered as interesting to extend

² The source code of the simulation is available at https://github.com/TimKam/uavxai-simulation/. A running simulation system (with UI controls activated) is available at http://s.cs.umu.se/51x65y.

the simulation to allow for human-in-the-loop feedback that helps improve the explanations over time.

Acknowledgments. This work was partially supported by the Regional Council of Bourgogne Franche-Comté (RBFC, France) within the project UrbanFly 20174-06234/06242. It was also partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

- Aguinis, H., Bradley, K.J.: Best practice recommendations for designing and implementing experimental vignette methodology studies. Organ. Res. Methods 17(4), 351–371 (2014)
- Anjomshoae, S., Främling, K., Najjar, A.: Explanations of black-box model predictions by contextual importance and utility. In: Calvaresi, D., Najjar, A., Schumacher, M., Främling, K. (eds.) EXTRAAMAS 2019. LNCS (LNAI), vol. 11763, pp. 95–109. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30391-4_6
- Goodman, B., Flaxman, S.: European union regulations on algorithmic decisionmaking and a "right to explanation". AI Mag. 38(3), 50–57 (2017)
- Johnston, A.B., Burnett, D.C.: WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web. Digital Codex LLC, St. Louis, MO, USA (2012)
- Kampik, T., Nieves, J.C.: JS-son-A minimal JavaScript BDI agent library. In: 7th International Workshop on Engineering Multi-Agent Systems (EMAS 2019) (2019)
- Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artif. Intell. 267, 1–38 (2019)
- Mualla, Y., Bai, W., Galland, S., Nicolle, C.: Comparison of agent-based simulation frameworks for unmanned aerial transportation applications. Procedia Comput. Sci. 130, 791–796 (2018)
- Mualla, Y., et al.: Agent-based simulation of unmanned aerial vehicles in civilian applications: a systematic literature review and research directions. Future Gener. Comput. Syst. 100, 344–364 (2019)
- Mualla, Y., Najjar, A., Kampik, T., Tchappi, I.H., Galland, S., Nicolle, C.: Towards explainability for a civilian UAV fleet management using an agent-based approach (2019). arXiv:1909.10090 cs.AI
- Mualla, Y., Tchappi, I.H., Najjar, A., Kampik, T., Galland, S., Nicolle, C.: Humanagent explainability: an experimental case study on the filtering of explanations. In: 12th International Conference on Agents and Artificial Intelligence (2020)
- Preece, A.: Asking 'Why' in AI: explainability of intelligent systems-perspectives and challenges. Intell. Syst. Account. Finance Manag. 25(2), 63–72 (2018)
- Rosenfeld, A., Richardson, A.: Explainability in human-agent systems. Auton. Agents Multi-Agent Syst. 33, 1–33 (2019)
- Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. 10(2), 115–152 (1995)