In-time Explainability in Multi-Agent Systems: Challenges, Opportunities, and Roadmap

Francesco Alzetta[1][0000−0001−5118−2084], Paolo Giorgini[1][0000−0003−4152−9683], Amro Najjar[3][0000−0001−7784−6176], Michael I. Schumacher[2][0000−0002−5123−5075], and Davide Calvaresi[2][0000−0001−9816−7439]

[1] University of Trent, Trento, Italy
{francesco.alzetta,paolo.giorgini}@unitn.it
[2] HES-SO Valais, 3960 Sierre, Switzerland
{davide.calvaresi, michael.schumacher}@hevs.ch
[3] University of Luxembourg, Luxembourg
amro.najjar@uni.lu

Abstract. In the race for automation, distributed systems are required to perform increasingly complex reasoning to deal with dynamic tasks, often not controlled by humans. On the one hand, systems dealing with strict-timing constraints in safety-critical applications mainly focused on predictability, leaving little room for complex planning and decision-making processes. Indeed, real-time techniques are very efficient in predetermined, constrained, and controlled scenarios. Nevertheless, they lack the necessary flexibility to operate in evolving settings, where the software needs to adapt to the changes of the environment. On the other hand, Intelligent Systems (IS) increasingly adopted Machine Learning (ML) techniques (e.g., subsymbolic predictors such as Neural Networks). The seminal application of those IS started in zero-risk domains producing revolutionary results. However, the ever-increasing exploitation of ML-based approaches generated opaque systems, which are nowadays no longer socially acceptable — calling for eXplainable AI (XAI). Such a problem is exacerbated when IS tend to approach safety-critical scenarios. This paper highlights the need for on-time explainability. In particular, it proposes to embrace the Real-Time Beliefs Desires Intentions (RT-BDI) framework as an enabler of eXplainable Multi-Agent Systems (XMAS) in time-critical XAI.

Keywords: eXplainable BDI model; Real-Time Systems; Multi-Agent Systems; eXplainable Autonomous Agents

1 Introduction

The recent advancements in the field of artificial intelligence (AI) are fostering the development of autonomous decision-making processes in systems operating in the real-world. In particular, nowadays, the majority of AI-based systems rely on Machine Learning (ML) approaches.

However, ML-based systems must face the problem of the opacity of sub-symbolic predictors (e.g., neural networks) [27, 6], which is no longer acceptable. Hence, current regulations acknowledge the right for meaningful explanations when automated decisions affect humans' lives [23, 17]. This originated a fervent effort in the so-called eXplainable AI (XAI) community, whose priority is to tackle the opacity of behaviors and results stemming from ML-based systems [24, 4, 35, 1, 20].

Recent studies advocated that multi-agent systems (MAS) offer a coherent yet expressive set of abstractions, promoting *conceptual integrity* in the engineering of complex software systems and serving the purpose of social XAI — constituiting the so-called XMAS [32, 17, 20].

Nevertheless, there is a worrying lack of consideration for the production and delivery time of the explanation. For example, considering devices operating in the real world such as autonomous cars [5], Unmanned Aerial Vehicles (UAVs) [17, 29], and traffic control networks, they are required to deal with a multitude of inputs and variables in highly-dynamic and unpredictable environments — while obliged to comply with Real-Time (RT) constraints. Therefore ensuring the on-time production and delivery of an explanation is crucial.

Real-Time Systems (RTS) are characterized by a plethora of algorithms ensuring compliance with strict-timing constraints [11]. Nevertheless, they require that both the environment and the possible system's interactions with it are predetermined (or predictable) [11]. If the environment is too complex to be thoroughly analyzed, or if it changes considerably, an RTS is not able to autonomously adapt to the new scenario — neither oracle nor one-size-fits-all approaches are possible.

For example, in self-driving cars, reacting in real-time to an unexpected event by promptly braking is necessary but not sufficient. The car should be able to analyze the surrounding environment and evaluate the consequences of its actions. If a deer crosses the road, the possible choice of the car to swerve in a ravine should be a decision taken on the base of a well-defined reasoning process, rather than being merely the result of reactive behavior that aims to avoid the animal. Therefore, there is a need for realizing systems able to base their decisions on their (evolving) knowledge of the world within given temporal bounds.

Calvaresi et al. [14] proposed a solution to enable the real-time compliance of MAS revising their pillars. Nevertheless, to mimic the cognitive behavior of humans using regular MAS is burdensome.

Being inspired by Bratman's theory of human practical reasoning [8], the Belief-Desire-Intention (BDI) model [34] represents one of the most recognized approaches to integrate the desired cognitive abilities in autonomous agents [28]. Furthermore, since the BDI agents' behavior is knowledge-driven – being determined by deliberation over well-structured concepts such as beliefs, goals, and intentions –, the cause-effect relationship that brought to the intended means can be depicted clearly. This paper presents and discusses the still unexplored challenges of designing and developing *Real-Time eXplainable* BDI Multi-Agent

Systems (RTX-BDI-MAS), eliciting *goals*, *challenges*, *opportunities*, possible *application scenarios*, and a *road map* to achieve such a result. Figure 1 schematically represents the view calling for RTX-BDI-MAS.

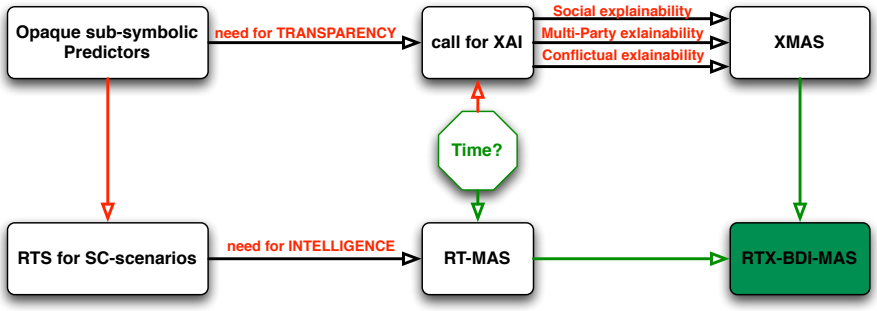

Fig. 1. Needs for RTX-BDI-MAS schematization.

The remainder of the paper is organized as follows. Section 3 analyzes the current challenges in designing Real-Time eXplainable BDI Multi-Agent Systems (RTX-BDI-MAS). Section 4 discusses the main advantages of developing RTX-BDI agents. Section 5 elaborates on possible application scenarios that would benefit from the employment of such agents. Section 6 proposes a road map to design a model for the development of RTX-BDI-MAS. Finally, Section 7 concludes the paper.

2 Background

XAI – Nowadays, most intelligent systems (IS) leverage on *subsymbolic* predictive models. Such a wide adoption is mainly due to the unprecedented data availability, enabling to detect useful statistical information hidden into such data semi-automatically. Nevertheless, most of the ML techniques carry well-known drawbacks. For example, the algorithmic *opacity* – the difficulty for the humans to *understand how* ML-based IS (also referred to as *black boxes*) operate or compute their outputs — is a serious issue if decisions' liability is needed [27]. Current regulations such as the GDPR [38] recognize the citizens' *right to explanation* [23]—implicitly requiring *understandable* IS. Moreover, having an understandable system can boost people's *trust* and *acceptability* —otherwise harmed. To cope with such issues and the normative requirements, the XAI research field has recently emerged, particularly tackling *interpretability* and *explainability* [25].

XMAS – Current XAI solutions are mostly use-case-specific and they help the interpretation of single ML-based algorithms [1]. However, standalone explainable approaches do not satisfy the needs of distributed and inter-connected IS.

For example, IoT systems are characterized by heterogeneous inputs, devices, and data-types concurring in the composition of complex information structures [20]. Hence, in [17], the MAS paradigm has been identified as potential means to *(i)* dynamically provide *interpretations* and *explanations* for opaque systems, *(ii)* ease the integration among different solutions/components for similar tasks or predictors, *(iii)* increase the degree of automation characterizing the development of intelligent systems, *(iv)* support AI and ML-based systems in distributed and decentralized contexts, where data cannot be moved due to technical or legal reasons, and *(v)* introduce and contribute to the social dimension of explainability.

RTS – Computing systems whose behavior correctness depends not only on the value of the computation but also on the time at which the results are produced – providing *soft* and/or *hard* timing guarantees – are known as RTS [36]. In RTS, the tasks models are *periodic*, *aperiodic* or *sporadic*, depending on the regularity of the tasks' activation (i.e., periodic - potentially infinite regular activations, aperiodic - irregularly interleaved, and sporadic - consecutive jobs are separated by minimum and maximum inter-arrival time) [11].

RT-MAS – A *Real-Time Agent* (RTA) extends and embodies a real-time *process*. Similarly to the RTS, the RTA correctness depends on both soundness and delivery time of its outcomes [21]. Enriching the conventional MAS with concepts such as *deadlines*, *precedence*, *priority*, and *constrained resources*, and mechanisms to handle them result in the so-called RT-MAS [12]. RTAs are intended to operate in highly dynamic environments. Thus, they adopt the *Earliest Deadline First* (EDF) mechanism [11] as the local scheduler. Nevertheless, EDF can only handle periodic tasks. Hence, to execute also aperiodic tasks (e.g., in charge of the message exchange), an RTA should combine EDF with a bandwidth reservation mechanism — i.e., the Constant Bandwidth Server (CBS) mechanism [12].

In the context of RT-MAS (similarly to RTS), missing "soft" deadlines may cause performance degradation, and missing "hard" deadlines entails a failure and possibly severe consequences.

Finally, MAS can be considered real-time compliant only if all the agents and their mechanisms (interactions included) operate accordingly [16].

BDI Agents – BDI-based agents are characterized by *beliefs*, *goals*, and *plans*. Beliefs represent the agent's knowledge about itself and the surrounding environment. Goals represent states of the world the agent wants to bring about. Plans are the means by which the agent can act to achieve its goals. BDI agents are well suited in unpredictable scenarios requiring dynamic decision-making due to their ability to choose the best plan to achieve a goal, given their current beliefs. In most BDI-based approaches, the process of repeatedly choosing and executing plans is called the *agent's reasoning cycle* [7].

3 Challenges

AI/ML-based systems are progressively pervading safety-critical application scenarios. Therefore, the needs for explainability and time-predictable behaviors blend in *demanding real-time production and delivery of the explanations.*

RTS and RT-MAS are able to comply with strict timing constraints. Yet, RTS show only predetermined behaviors — limitation overcome by RT-MAS. Nevertheless, since both RTS and RT-MAS (as-is) are incapable of performing "explicit reasoning" to explain their conduct, they cannot be considered XAI-compliant. BDI agents and XMAS can both make autonomous decisions dynamically. In BDI, the agent's reasoning cycle offers intrinsically a cause-effect explanation regarding its decisions, while XAMS can explain ML-based system (i.e., generate a symbolic representation of subsymbolic knowledge). Nevertheless, both BDI and XMAS lack the main property of RTS and RT-MAS: make decisions, and therefore act, in time.

Therefore, none of the existing approaches taken individually allows to produce explanations complying with time constraints and dynamically adapting to the environment in which they operate. Despite several studies attempted to combine the RTS and MAS [37, 19] and other made their way through [12], to the best of our knowledge no previous attempt of providing in time explanations can be mentioned.

Moving towards the definition of a model that integrates RT-MAS and XMAS properties and capabilities, requires to address several questions. In particular:

*What is the impact of **RT-compliance** on XMAS?*
Depending on the application domain, the consequences generated by a given explanation can vary significantly. Thus, having a predictable delivery time of the explanation will play a crucial role. Such a requirement entails the development of mechanisms ruling all the behaviors (including the ones generating the explanations) in a real-time manner. Considering the social dimension of explainability (i.e., goal-driven XAI [17, 20]), the whole process might require several interactions between explainer and explainee. Current approaches neglect the converging time of conveying an explanation —condition unacceptable under RT assumptions. To overcome such a limitation is crucial. Hence, the agent should be aware of the costs of generating an explanation (both resources and time-wise) and act accordingly — even if it will come on the expenses of performance, explanation's granularity, or quality.

In particular, a first step for metrics and mechanisms-revision in BDI agents should involve inevitably a structural revision of the architecture. Thus, the notion of time itself can also play a direct role within a given explanation (e.g., impossible to complete safely *plan-A*, so emergency switch to *plan-B*) — **clearly performed in time**.

*Which **Architecture** should be adopted?*
The architecture of a software agent identifies the fundamental components

that allow the agent to make decisions (henceforth producing explanations) taking into account the temporal constraints typical of RTS. In literature, there are two approaches: layered and integrated architectures. In layered architectures, the real-time and the cognitive functionalities are separated, acting on different layers, and each one relies and depends on the behavior of the other. This approach involves two loosely coupled sub-systems, so it allows an easier design. However, since the deliberative layer does not act in real-time, such a system cannot guarantee compliance with hard real-time constraints. Concerning integrated architectures, instead, Musliner et al. [31] claim that a hybrid system can be obtained by embedding either AI into a real-time system or real-time reactions into an AI system. In the former case, AI computations are forced to meet deadlines like any other real-time task, while in the latter the deliberation techniques will be short-circuited in favor of a real-time reflexive action. An eXplainable system should always provide at least a basic motivation for its decisions, so an RTX-BDI architecture should integrate AI processes (including explanations) into an RTS. By doing so, the level of detail of the explanation can depend on the time the agent has to provide it.

*Which **Algorithms** have to regulate the behavior of the agent?*
A crucial point consists in the definition of the algorithms and techniques that enable the scheduling of the agent's activities and comply with strict timing-constraints. As discussed previously, due to the diversity of their original purposes, MAS, XMAS, and RTS rely upon significantly different mechanisms that need to be revised and modified to allow them to cooperate. For instance, while the concept of a real-time task can easily be mapped with a BDI action, problems arise when the BDI agent should manage the different types of tasks (periodic, aperiodic, and sporadic) typical of RTS [11]. Indeed, such a characterization is neither considered in the agent-oriented paradigm, nor in the current state of the art of XAI [1], preventing the use of pure real-time theories and their application in the context of XAI and MAS with a 1-to-1 mapping. Therefore, when designing an RTX-BDI agent, a revision of actions and plans becomes necessary to take these diversities into consideration. A similar analysis has been done by Calvaresi et al. in [16], which suggests that a mapping between Jade's *behaviours* and the real-time task models is possible. Nevertheless, the BDI model has higher abstraction levels, thus requiring a more complex mapping.

Another challenge concerns the scheduler employed by the agent. Indeed, most of the state-of-the-art agent platforms adopt best-effort approaches that are not able to handle the system behavior in worst-case scenarios [16]. In such approaches, computational times and deadlines do not have a role in deciding about the execution of the next task. This prevents the agent from controlling the generation and communication of explanations and the interleaving with potentially non-time-critical tasks. To realize real-time explainable agents is essential that they rely on a real-time compliant local scheduler [15].

However, there are no real-time schedulers suitable to be implemented, as they are, in agents running in an open environment (hence flexible enough to

deal with sudden changes in priorities). Indeed, some additional mechanisms must be implemented to allow the agent to manage the dynamic activation of tasks having arrival times unknown a-priori. An agent acting in real-time must be able to establish which goals to prioritize when it cannot achieve all of them, a common situation when considering time as a limited resource. While the concept of priority is central in RTS, in agent-based systems it is often neglected, assuming that all goals will be eventually reached by the agent.

Similarly, the problem of choosing among plans designed to achieve the same goal must be addressed. Indeed, in a real-case scenario, to adapt to different circumstances agents usually have different ways to achieve their goals. Then, the agents should be able to elaborate trade-offs by selecting plans that allow them to balance between the number of goals achieved and the efforts or resources required to achieve them. A very delicate and critical challenge that characterizes XMAS – worsened by the introduction of real-time – concerns how the agent should/must behave when the execution of an intention fails (e.g., generating or communicating an explanation do not converge before a critical deadline occurs). The strict schedule typical of RTS leaves little room for the execution of unforeseen, unboundable, and alternative tasks, which are needed to perform backtracking or to try a different way to achieve the goal. In a multi-agent system, the agents have to exchange information among them, negotiate, and cooperate. Therefore, it is necessary to define interaction techniques that allow real-time communication and cooperation between agents.

How should the system be ***Validated****?*
Once such a system is designed and implemented, the problem regarding how it should be validated arises. Indeed, in literature few studies tried to achieve the same goal [1, 12], hence there are no significant results to compare with. However, interesting insights can be obtained by evaluating particular properties: for instance, comparing performances time-wise respect to state-of-the-art MAS (and XMAS if any) frameworks, or measuring flexibility by analyzing the behaviors of such a system and real-time ones in unpredictable scenarios. Moreover, implementing explainable mechanism will already play a crucial role within the validation stage itself. In particular, such a mechanism can enable meaningful and more understandable debugging phases eliciting values, roles, and dynamics of internal (possibly hidden/opaque) parameters.

Figure 2 summarizes the components entangled with each of the challenges discussed above: architecture (AR), algorithms (AL), and validation (VA).

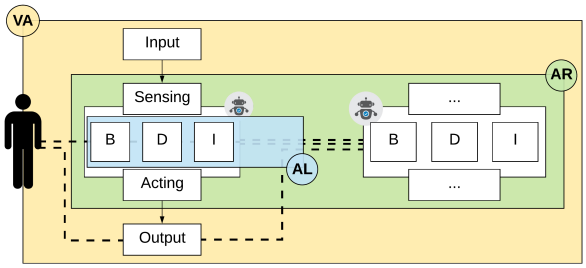

Fig. 2. Graphical representation of the components and the respective challenges. The dashed lines represent the information exchange between entities.

4 Opportunities

Realizing the RTX-BDI-MAS model allows to merge the properties and advantages of classical RTS (i.e., systems characterized by bounded response times and no deadline miss) and XMAS (i.e., systems able to generate symbolic representations of subsymbolic information) and extend them with the BDI systems' capability of making decisions and adopting dynamic behaviors in response to the changes in the environment in which they operate.

RTX-BDI-MAS can be particularly useful in safety-critical and unpredictable domains, such as autonomous driving, telerehabilitation, personal coaching, and air traffic control. In these scenarios, the systems involved need to adopt algorithms that allow them to behave correctly (and in time) in case an unforeseen event occurs. Moreover, due to their safety-related requirements, the use of symbolic AI in the reasoning process is mandatory, as the uncertainty given by statistical AI and ML-based systems – which can still be used to solve specific sub-problems – may lead to catastrophic consequences. Furthermore, in the last decade, safety-critical systems are increasingly composed of different (possibly distributed) components interacting with one another — strengthening the choice of MAS as underlying paradigm.

On Explainability – Transparency and understandability are broadly known to be the main factors calling for XAI [1]. Nevertheless, enabling explicit reasoning *about* and *in* time can be a key enabler for crucial desiderata such as expressing systems, agents, and robots' reasons, capabilities, and limits to their end-user [26]. Hence, time can play a prominent role in the decision-making process, whether a plan is chosen or dropped. The claimed system *accountability* [2] cannot be achieved regardless of *transparency* and *time*. Conversely, a system is not able to perform in *real*-world application scenarios predictably (or properly at all) — since the humans' interactions are inherently entangled with the concept

of time. Finally, RTX-BDI-MAS can facilitate the tuning of the explanation's granularity, enhancing the efficiency and the response time of the system.

On Time-awareness – To reason *about* and *in* time, RTX-BDI-agents have to embody a real-time scheduler. Thus, it is possible to ensure that every decision-making-process is executed respecting the time constraints, while the integration of temporal concepts in the BDI model, such as computational time and deadline, allows the agent to make these decisions considering also the time as a finite resource. Since the agents are able to take their decisions on the base of the time required to execute an intention, designers can also tune the agents to prioritize optimal solutions time-wise or output-wise, or to identify a *feasible* balance of the two. Such a tuning allows the designers to specify the desired Quality of Service (QoS), configuring the agent to be more reactive or more reflective, depending on the desired behavior. It is worth noticing that this only affects the choices made by the agent (i.e., which goals they commit to and which intentions they execute), without breaking any real-time boundary.

On Ease of design – Compared to distributed RTS and XMAS, RTX-BDI-MAS provides a more natural design phase. Indeed, the developer has to design the single components (beliefs, desires, plans, and tasks) without the burden of establishing the rules and behaviors that operate the run-time execution of the device. Since the BDI model is based on the human practical reasoning theory, the design of such components is very intuitive. The reasoning cycle of a BDI agent, indeed, is similar to our way of thinking: we perceive a change in the environment (change in agent's beliefs) which can make us desire to achieve some goals (instantiation of agent's desires), and we reason about the actions to take to satisfy those desires (generation of intentions through means-end reasoning).

On Robustness – With respect to RTS, RTX-BDI-MAS grant more robustness in open environments, allowing real-time software to promptly and adequately deal with system failures. Indeed, since RTS are designed to work in controlled and predefined environments, the possibility of having system failures is excluded a-priori (unless hardware failures handled with devices redundancy). In general, RTS only manage overloads, i.e., the system can lower the band to fit the tasks, if this does not cause losing deadlines or important information [11]. Conversely, when feasible, RTX-BDI-MAS allows the re-planning and rearrangements on-the-fly by reconsidering their goals and intentions.

5 Application scenarios

In general, an RTX-BDI architecture allows us to build systems able to perform autonomous actions in time, reasoning not only in a self-interested way but coordinating with all the other agents, possibly leveraging on symbolic reasoning. This is particularly valuable for systems in which decision-making processes are needed, but reducing at the minimum the human error and increasing the acceptance and understanding of the system's behaviors are the cornerstones.

Summarizing, the possible scenarios in which a system operates can be classified in general-purpose or non-safety-critical (NSC), XAI-critical and non-safety-critical (XNSC), safety-critical (SC), dynamic safety-critical (DSC), and dynamic XAI and safety-critical (XDSC). Figure 3 organize the types of system per the most appropriate scenario, highlighting the evolution of the efficiency (according to the resources allocated), predictability, and the capability of producing explanations of opaque subsymbolic predictors.

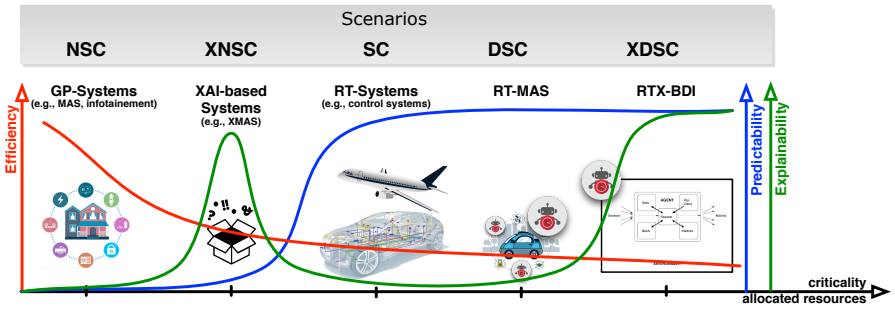

Fig. 3. Systems classification per efficiency, predictability, explainability, resources allocated, scenarios and criticality.

Besides the expected lack of efficiency (given the resources allocated and the need for ensuring timing guarantees) RTX-BDI agents are envisioned to be equipped with both XAI and RTS capabilities, while keeping unaltered the social abilities typical of MAS.

To highlight the relevance of RTX-BDI-MAS in XDSC, we briefly elaborate on two envisioned scenarios.

Telerehabilitation – Most of the telerehabilitation systems are expected to be used without the direct supervision of any medical staff. The majority of the proposed systems leverage on wearable distributed sensing [9, 14]. Such systems provide real-time monitoring and feedback, storing the data generated during the therapy sessions for the (long-term) trend-analysis. For simple exercises, such approaches are effective. Nevertheless, depending on the joint(s) to rehabilitate, the therapy might be more complex (from both physical and cognitive viewpoints). A first step to enable telerehabilitation for more demanding therapies is presented in [13], where the authors developed a semantic model for RT-MAS enabling more elaborated – yet real-time compliant– interactions among the wearable sensors. Along this path, we envision that RTX-BDI-MAS can empower the telerehabilitation systems by providing not only real-time monitoring and feedback but also providing in-time explanations. In particular, it would enrich the coaching capability of the system and provide better (possibly more understandable) support to the patients dealing with complex exercises, which

have higher chances of causing late or wrong movements — thus hurting the patient and jeopardizing the beneficial effects of the therapy.

Autonomous vehicle and robots – in the case of fully autonomous multi-vehicles or robots, the compliance with strict-timing constraints is imperative (e.g., to avoid collisions). However, complex interactions (e.g., negotiations) are increasingly pervading the robotic and autonomous vehicles worlds. In our vision, establishing an agreement might soon leverage on the explanation of ML-based predictors or of complex behaviors intertwined with subsymbolic information. In the case of semi-autonomous vehicles [5], the system might be required to present timely explanations to the driver to undertake a given (possibly time-critical) decision, which requires the human's approval. Finally, in UAVs search and rescue scenarios [30], UAV teams need to cooperate to achieve common goals. In such a case, identifying the responsibility of each UAV is crucial. Hence, it can enable to ensure efficient collaboration or, in a case of failure, to trace the underlying reasons and assign responsibilities — both to improve the future system's performance and to held involved parties accountable. Once again, employing RTX-BDI-MAS would bridge the advantages of the two worlds (i.e., XMAS and RT-MAS).

6 Road map

This section presents the four phases need to formalize an RTX-BDI-MAS model.

PH1: *First formalization of the RTX-BDI model.*
To guarantee the properties of RTS, the BDI structure needs to be revised to consider the necessary real-time notions, such as *priority*, *deadline*, and *worst-case execution time*. Redesigning desires, plans, intentions, and actions by involving such elements would allow the integration of a real-time scheduler in the reasoning cycle of the agent, providing real-time guarantees in both the deliberative and executive processes. Moreover, it is necessary to identify the tasks model and the dynamics necessary to perform predictably the intra-agent explainability [20].

PH2: *Definition of policies to handle plan failures.*
Plan failure management, as discussed in section 3, is probably one of the most challenging problems to be solved in real-time compliant MAS. Indeed, due to the high dynamism of the scenarios in which XMAS are expected to operate, the robustness typically obtained by RTS is very difficult to be achieved. Moreover, if the system is composed by a growing number of elements, also the possible failures of other agents must be taken into consideration and managed. The robustness of the system can be further improved by developing a real-time compliant selection function able to avoid conflicts between intentions, similarly to what is done in [39]. Such a work, by performing pseudo-random simulations of different interleavings of the plans, looks for an optimal interleaving of the actions that will allow the agent to achieve the largest number of goals. This

approach helps in minimizing the possibility of plan failures, but to be applicable in RTX-BDI-MAS it has to be redesigned to consider real-time compliance. Finally, besides the effects that it might imply, the failure of an explanations might entail several factors (e.g., lack of a common ontology, unknown state of mind of the explainee, and possible lack of time to complete the interaction necessary for the entire knowledge-transfer). To avoid, or understand, the reasons standing behind a failure, specific mechanisms need to be developed to setup effective "possibly personalized" explanations.

PH3: *The definition of the interaction techniques.*
`PH1` and `PH2` allow the development of single RTX-BDI agents. When the system scales from single to multi-agent settings, interaction techniques and protocols are required to allow the agents of the RTX-BDI-MAS to communicate (henceforth explain), negotiate, and cooperate. Although a standard for MAS communication already exists (i.e., FIPA Agent Communication Language (ACL) [22]), it lacks of several fundamental mechanisms crucial to handle multi-step explanations and RT-compliance. Indeed, FIPA ACL does not provide a way to manage either the network load and messages status (e.g., bounding congestion and delivering times is not possible), nor the in/out message queues. Furthermore, broadcasting (particularly useful when the information of a sensor should be exploited by many components) is difficult to be achieved. To overcome such limitations, a communication middleware able to guarantee bounded-time delays must be employed. In [18], the authors identify the Real-Time Publish-Subscribe (RTPS) as viable technology (already adopted by the Data Distribution Service (DDS) systems in aerospace domains [33]).

PH4: *The implementation of a prototype for verification and validation*
The last phase regards the development of an RTX-BDI-MAS prototype, which must be used to verify and validate the model. The evaluation can be done in a simulated or real environment. The implementation of a simulator allows a better, safer, and cheaper analysis of the systems' behavior — since it acts in a controlled environment. However, deploying the system in real-world devices represents a more significant validation. Indeed, in real-world scenarios, the adaptability of the system is stressed.

To enable deployment and testing of the RT-BDI MAS, a framework equipped with an intuitive graphical user interface and comprehensive analysis tools is needed. Extending any of the most recognized and supported agents frameworks in the literature, such as JACK [10], JADE [3], and Jason [7] is not feasible nor effective. Indeed, among the main limitations hampering such a way it is possible to mention *(i)* they are based on Java, thus incapable of guaranteeing any real-time compliance and *(ii)* they rely on general-purpose algorithms (e.g., round-robin and first-come-first-served) neglecting elements such as *time*, *utilization*, and *deadlines* core of any real-time compliant algorithm [15, 12], and *(iii)* lack of means to extract symbolic knowledge from subsymbolic data. Coupling explainability and visualization would boost the user's understanding of the underlying system. Furthermore, in case explainability provides a deep view of the

inner-mechanism, it allows the user/developer to predict the outcomes demonstrated by the system when the input parameters change. Thus, the system can be validated under different settings, always allowing clear system assessment and understanding.

7 Conclusion

This paper discussed challenges and opportunities of modeling and developing explainable and RT compliant MAS based on the BDI cognitive architecture. This preliminary analysis shows that such a system can enhance the reasoning and decision-making processes of applications that have to comply with strict real-time constraints, while providing transparency, and promoting trust. More precisely, it allows us to exploit the structure of BDI to easily design explainable RTS able to dynamically adapt to the uncertainties that characterize open environments. Moreover, using BDI fosters adaptive and user-friendly explainability, which enables end-users (or other agents in the system) to understand the system behavior and modify it in case a need arises. Nevertheless, several complex challenges must be faced. The main ones concern the system's type of architecture, its mechanisms, and behavior policies.

References

1. Anjomshoae, S., Najjar, A., Calvaresi, D., Främling, K.: Explainable agents and robots: Results from a systematic literature review. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. pp. 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems (2019)
2. Baldoni, M., Baroglio, C., Boissier, O., May, K.M., Micalizio, R., Tedeschi, S.: Accountability and responsibility in agent organizations. In: International Conference on Principles and Practice of Multi-Agent Systems. pp. 261–278. Springer (2018)
3. Bellifemine, F., Poggi, A., Rimassa, G.: Jade–a fipa-compliant agent framework. In: Proceedings of PAAM. vol. 99, p. 33. London (1999)
4. Besold, T.R., Uckelman, S.L.: The what, the why, and the how of artificial explanations in automated decision-making. CoRR **abs/1808.07074**, 1–20 (2018), http://arxiv.org/abs/1808.07074
5. Biondi, A., Nesti, F., Cicero, G., Casini, D., Buttazzo, G.: A safe, secure, and predictable software architecture for deep learning in safety-critical systems. IEEE Embedded Systems Letters pp. 1–1 (2019)
6. Biondi, A., Pazzaglia, P., Balsini, A., Di Natale, M.: Logical execution time implementation and memory optimization issues in autosar applications for multicores. In: International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS) (2017)
7. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason, vol. 8. John Wiley & Sons (2007)
8. Bratman, M.: Intention, plans, and practical reason, vol. 10. Harvard University Press Cambridge, MA (1987)

9. Buonocunto, P., Giantomassi, A., Marinoni, M., Calvaresi, D., Buttazzo, G.: A limb tracking platform for tele-rehabilitation. ACM Transactions on Cyber-Physical Systems **2**(4), 1–23 (2018)
10. Busetta, P., Rönnquist, R., Hodgson, A., Lucas, A.: Jack intelligent agents-components for intelligent agents in java. AgentLink News Letter **2**(1), 2–5 (1999)
11. Buttazzo, G.C.: Hard real-time computing systems: predictable scheduling algorithms and applications, vol. 24. Springer Science & Business Media (2011)
12. Calvaresi, D.: Real-time multi-agent systems: challenges, model, and performance analysis (2018)
13. Calvaresi, D., Calbimonte, J.P.: Real-time compliant stream processing agents for physical rehabilitation. Sensors **20**(3), 746 (2020)
14. Calvaresi, D., Marinoni, M., Dragoni, A.F., Hilfiker, R., Schumacher, M.: Real-time multi-agent systems for telerehabilitation scenarios. Artificial Intelligence in Medicine **96**, 217 – 231 (2019). https://doi.org/https://doi.org/10.1016/j.artmed.2019.02.001
15. Calvaresi, D., Marinoni, M., Lustrissimini, L., Appoggetti, K., Sernani, P., Dragoni, A.F., Schumacher, M., Buttazzo, G.: Local scheduling in multi-agent systems: getting ready for safety-critical scenarios. In: Multi-Agent Systems and Agreement Technologies, pp. 96–111. Springer (2017)
16. Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., Buttazzo, G.: The challenge of real-time multi-agent systems for enabling iot and cps. In: Proceedings of the International Conference on Web Intelligence. pp. 356–364. ACM (2017)
17. Calvaresi, D., Mualla, Y., Najjar, A., Galland, S., Schumacher, M.: Explainable multi-agent systems through blockchain technology. In: Explainable, Transparent Autonomous Agents and Multi-Agent Systems - First International Workshop, EXTRAAMAS 2019, Montreal, QC, Canada, May 13-14, 2019, Revised Selected Papers. pp. 41–58. Springer, Berlin Heidelberg (2019). https://doi.org/10.1007/978-3-030-30391-4_3
18. Calvaresi, D., Schumacher, M., Marinoni, M., Hilfiker, R., Dragoni, A.F., Buttazzo, G.: Agent-based systems for telerehabilitation: strengths, limitations and future challenges. In: Agents and Multi-Agent Systems for Health Care, pp. 3–24. Springer (2017)
19. Carrascosa, C., Bajo, J., Julián, V., Corchado, J.M., Botti, V.: Hybrid multi-agent architecture as a real-time problem-solving model. Expert Systems with Applications **34**(1), 2–17 (2008)
20. Ciatto, G., Calegari, R., Omicini, A., Calvaresi, D.: Towards XMAS: explainability through multi-agent systems. In: Proceedings of the 1st Workshop on Artificial Intelligence and Internet of Things co-located with the 18th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2019), Rende (CS), Italy, November 22, 2019. pp. 40–53. CEUR Worhsop Proceedings, Sun SITE Central Europe, RWTH Aachen University (2019), http://ceur-ws.org/Vol-2502/paper3.pdf
21. Dragoni, A., Sernani, P., Calvaresi, D.: When rationality entered time and became a real agent in a cyber-society. vol. 2280, pp. 167–171 (2018)
22. FIPA: FIPA ACL message structure specification. FIPA agent communication language specifications (2002), http://www.fipa.org/specs/fipa00061
23. Goodman, B., Flaxman, S.: European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine **38**(3), 50–57 (2017). https://doi.org/10.1609/aimag.v38i3.2741
24. Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., Giannotti, F.: A survey of methods for explaining black box models. ACM Computing Surveys (CSUR) **51**(5) (Jan 2019). https://doi.org/10.1145/3236009

25. Gunning, D.: Explainable artificial intelligence (XAI). Funding Program DARPA-BAA-16-53, Defense Advanced Research Projects Agency (DARPA) (2016), http://www.darpa.mil/program/explainable-artificial-intelligence
26. Hellström, T., Bensch, S.: Understandable robots-what, why, and how. Paladyn, Journal of Behavioral Robotics **9**(1), 110–123 (2018)
27. Lipton, Z.C.: The mythos of model interpretability. ACM Queue **16**(3) (May–Jun 2018), https://dl.acm.org/citation.cfm?id=3241340
28. Logan, B.: An agent programming manifesto. International Journal of Agent-Orientated Software Engineering (2017)
29. Mualla, Y., Najjar, A., Daoud, A., Galland, S., Nicolle, C., Shakshuki, E., et al.: Agent-based simulation of unmanned aerial vehicles in civilian applications: A systematic literature review and research directions. Future Generation Computer Systems **100**, 344–364 (2019)
30. Mualla, Y., Najjar, A., Galland, S., Nicolle, C., Haman Tchappi, I., Yasar, A.U.H., Främling, K.: Between the megalopolis and the deep blue sky: challenges of transport with uavs in future smart cities. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. pp. 1649–1653. International Foundation for Autonomous Agents and Multiagent Systems (2019)
31. Musliner, D.J., Hendler, J.A., Agrawala, A.K., Durfee, E.H., Strosnider, J.K., Paul, C.: The challenges of real-time ai. Computer **28**(1), 58–66 (1995)
32. Omicini, A., Zambonelli, F.: MAS as complex systems: A view on the role of declarative approaches. In: Declarative Agent Languages and Technologies (DALT), LNAI, vol. 2990, pp. 1–17 (2004). https://doi.org/10.1007/978-3-540-25932-9_1
33. Pardo-Castellote, G.: Omg data-distribution service: Architectural overview. In: 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings. pp. 200–206. IEEE (2003)
34. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a bdi-architecture. KR **91**, 473–484 (1991)
35. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: 22nd ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining (KDD'16). pp. 1135–1144. ACM, San Francisco, CA, USA (22–26 Aug 2016). https://doi.org/10.1145/2939672.2939778
36. Stankovic, J.A., Ramamritham, K. (eds.): Tutorial: Hard Real-time Systems. IEEE Computer Society Press, Los Alamitos, CA, USA (1989)
37. Vikhorev, K., Alechina, N., Logan, B.: The arts real-time agent architecture. In: International Workshop on Languages, Methodologies and Development Tools for Multi-Agent Systems. pp. 1–15. Springer (2009)
38. Voigt, P., von dem Bussche, A.: The EU General Data Protection Regulation (GDPR). A Practical Guide. Springer (2017). https://doi.org/10.1007/978-3-319-57959-7
39. Yao, Y., Logan, B.: Action-level intention selection for bdi agents. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. pp. 1227–1236. International Foundation for Autonomous Agents and Multiagent Systems (2016)