Messenger BDI: An Human and Cognitive Agent Communication approach over KQML

Gustavo Saldanha[1], **Mateus Façanha**[1], **Kauã Santos**[1], **Tielle Alexandre**[2]
Bruno Freitas[1,2], **Nilson Lazarin**[1,2], **and Carlos Pantoja**[1]

[1]Federal Center for Technological Education Celso Suckow da Fonseca (Cefet/RJ)
Rio de Janeiro, RJ – Brazil
[2]Institute of Computing - Fluminense Federal University (UFF) – Niterói, RJ – Brazil

chon@grupo.cefet-rj.br

Abstract. *While embedding Belief-Desire-Intention (BDI) agents in resource-constrained devices enables autonomous operations, effective Human-Agent Interaction (HAI) for real-time supervision remains challenging due to complex, low-level communication barriers. This work proposes a chat-based web interface facilitating seamless human interaction with distributed BDI Multi-agent Systems (MAS). By allowing users to intuitively select speech-act performatives, the system automatically generates and delivers valid KQML messages to embedded agents. A proof-of-concept involving a simulated BDI-controlled UAV executing flight instructions validates the approach's feasibility and effectiveness in supporting human-in-the-loop supervision for distributed, resource-constrained MAS.*

1. Introduction

Intelligent agents are autonomous computational entities capable of making decisions and acting independently, based on their perceptions of the environment, individual knowledge, and specific goals. A Multi-Agent System (MAS) is a group of computational agents that exist within the same system and compete or collaborate to achieve common or conflicting goals (Wooldridge 1999). To exhibit intelligent behavior, these agents are commonly designed using cognitive models that represent their mental capabilities and reasoning processes (Sichman et al. 1992). One of these models is the Belief–Desire–Intention (BDI) (Bratman et al. 1988), which formalizes an agent's decision-making process in terms of *beliefs* – information it holds about the world; *desires* – objectives it wants to achieve; and *intentions* – commitments to plans of action toward those objectives (De Silva et al. 2020). BDI-based MAS have been widely adopted across several domains (Cardoso and Ferrando 2021), and recent studies have demonstrated that they can also be embedded into resource-constrained devices operating in pervasive environments, perceiving and acting in real-world contexts (Lazarin et al. 2024).

Ensuring effective coordination between agents in a MAS depends on structured and semantically interpretable interaction mechanisms. These mechanisms allow agents to coordinate their actions through standardized languages that define shared message formats and interaction protocols. Among them, the Knowledge Query and Manipulation Language (KQML) (Finin et al. 1994), grounded in speech act theory, enables agents to issue requests and exchange information. When integrated with IoT middleware such as ContextNet (Endler et al. 2011), which supports large-scale and context-aware exchanges among distributed mobile and embedded entities, such semantic infrastructures enable agent mobility and coordination between agents from different MAS's across pervasive networks (Jesus et al. 2021). Moreover, recent implementations integrating KQML-based communication with IoT middleware infrastructures demonstrate that embedded MAS can sustain decentralized coordination and interoperable interaction across distributed and resource-constrained environments (Lira et al. 2025; Freitas et al. 2025a).

In addition to Agent-to-Agent (A2A) coordination, Human–Agent Interaction (HAI) plays an important role when embedded BDI-based MAS are deployed at the edge – such as in service robots, autonomous vehicles, and IoT controllers – by relating autonomous decision-making to human supervision. In these contexts, supervision constitutes a key ethical and safety principle, ensuring that human operators retain meaningful control over systems capable of autonomous reasoning and action. The HAI allows adaptive collaboration in which agents interpret human intentions and communicate their internal states or decisions in an explainable way, thereby improving transparency and trust (Lewis 1998; Flathmann et al. 2019). In resource-constrained or intermittently connected environments, effective HAI also supports human-in-the-loop (HITL) supervision through lightweight yet reliable communication mechanisms, allowing agents to receive guidance or updates under limited network infrastructure (Lazarin et al. 2025). This capability is particularly useful for agents operating in real-world environments, where human feedback is often required to adjust plans, ensure safety, or recover from unexpected situations.

However, to ensure human supervision in MAS, it is necessary both a communication architecture and an interface for the human . Furthermore, the MAS must have levels of intervention and control that allow the human to exercise proportional authority in the face of the risks and potential impact of the system, including redirecting ongoing actions or suspending operations. The lack of this capacity could, for example, compromise the MAS's ethical behaviour in some scenarios.

This work proposes an architecture that enables communication between humans and BDI agents.The proposed approach allows the human user to select a speech-act performative (e.g., *tell*, *achieve*, *ask-one*) and automatically generate a KQML message, which is then transmitted to a BDI agent. For this, we adopt a web architecture using the WebSocket protocol to mediate human-agent messages, where a client-server instance of the ContextNet middleware (Endler et al. 2011) is created to format messages in KQML and send them to agents and vice versa. Also, we create a web-based chat interface for humans to exchange commands with an agent within a MAS. The MAS is programmed using an extended version of Jason (Pantoja et al. 2023), which allows agents to communicate via ContextNet.

For the proposed approach, a proof-of-concept implementation has been developed and integrated into a resource-constrained devices simulator (Freitas et al. 2025b). An Unmanned Aerial Vehicle (UAV) controlled by a MAS receives and executes human instructions during flight. The UAV can receive messages from the chat and adjust its behavior in response to human supervision. This experiment demonstrates the feasibility of integrating a chat-based communication interface to control distributed, resource-constrained devices using BDI agents.

This paper is structured as follows: in Section 2, we present related works; in Section 3, we propose our chat-based approach. Section 4 shows the qualitative evaluation proof-of-concept; Section 5 demonstrates the reproducibility of this work; and finally, Section 6 presents the conclusion.

2. Related works

We conducted a mapping review using the PRISMA guideline to investigate the state of the art in human-agent communication in BDI-based MAS, focusing on interfaces, usability, and supporting technologies for HAI mediated by chat and supported by KQML. The objective was to identify approaches, challenges, and solutions from the literature that could guide the development of a conversational interface integrated with communication structures among agents. The review focused on studies on communication in distributed systems, message-exchange mechanisms in BDI agents, and interface proposals that facilitate interaction with intelligent agents. The searches were conducted in two academic databases: Scopus and Web of Science, using the following search string: “Human-agent” AND (“interface” OR “chat” OR “interaction” OR “communication” OR “collaboration” OR “teaming” OR “cooperation”) AND (“BDI” OR “Belief-Desire-Intention”).

The criteria to select relevant studies included explicitly addressing HAI; presenting a solution, architecture, or analysis involving BDI agents; mentioning protocols, frameworks, or communication middleware; and discussing interface, usability, or conversational aspects. After applying the inclusion and exclusion criteria, 31 works in Scopus and 65 in Web of Science were identified. These studies were filtered by reading the titles and abstracts and by assessing methodological completeness, clarity of objectives, quality of the results, and contribution to understanding HAI in MAS.

- Scopus (31 papers): In the first step, 5 papers were removed as they were not full research papers, and 6 were identified as duplicates. From the remaining 20, 7 were excluded for not being related to the research area. Subsequently, 3 papers were excluded due to a lack of full-text access. Finally, 5 papers were removed for failing to adhere to the specific theme, leaving 5 selected papers.
- Web Of Science (65 papers): The first step eliminated 27 papers (18 unrelated to the area and 9 duplicates). A deeper analysis of the remaining 38 led to the exclusion of 30 additional works considered unrelated to the core theme. Of the remaining 8, 2 were excluded due to access restrictions and 1 due to non-adherence, resulting in 5 selected papers.

The final selection of 10 papers was analyzed by reading the titles, abstracts, and full texts, and by assessing methodological completeness, clarity of objectives, quality of results, and contribution to understanding HAI in MAS. The identified literature can be divided into two main strands. The first gathers studies focused on BDI architectures, models, and symbolic communication protocols. This group supports the communication infrastructure. The first work (Lazarin et al. 2025) validates the exchange of KQML messages between humans and BDI agents. Although it uses email protocols (IMAP/SMTP) to demonstrate the feasibility of encapsulating agent messages, our work uses a synchronous WebSocket+IoT middleware approach to allow human-agents communication. Another work (Johnson and Johnson 2001) details in how interaction and collaboration among agents, organizations, and humans should be structured to achieve common goals in a multi-agent environment. Another related work is Chat-BDI (Gatti et al. 2025) that proposes a BDI chat, which integrates Large Language Models (LLM) to translate natural language into KQML messages. It allows humans to interact with BDI agents, acting as an interpreter between natural language and KQML. Our work differs by enabling direct interaction with distributed and resource-constrained embedded MAS.

Regarding the infrastructure, one work (Song and Piggott 2003) contributes to the modeling of message-handling systems to ensure communication integrity in complex distributed systems. On the other hand, interface frameworks abstract complexity, serving as a reference for the integration architecture proposed in this work (Urlings et al. 2005). A recent work (Barange et al. 2014) focuses on task-oriented conversational behavior, proposing models that directly map BDI beliefs and intentions into dialogues. Agents and humans can interact in natural language, but the MAS is not geographically distributed, and all agents and humans are in the same system. Finally, one work (Berger and Pesty 2005) addresses the construction of conversational languages, discussing how AI can process natural language to make communication with the agent more fluid and less dependent on rigid commands.

The second group analyses chats as an interface, focusing on user perception and acceptance. A recent work (Chang et al. 2024) analyzes the dynamics of companionship between humans and AI, highlighting the importance of consistency and empathetic response for maintaining engagement. Another work (Ltifi 2023) discusses trust, demonstrating that interfaces that emulate semi-human characteristics (such as well-designed chatbots) significantly increase the system's credibility with the user. Users' perceptions and behaviors when interacting with chatbots are also investigated, validating that conversational interfaces reduce cognitive barriers and facilitate technology adoption (Chin et al. 2024). Finally, a survey (Goel and Ganatra 2021) on the future of chatbots points out that integrating lightweight interfaces with robust server architectures supports the model of connecting a simple web interface to a complex network of agents.

The preliminary analysis of these works revealed that although there are solutions with technical infrastructure, they do not explore BDI agents and KQML-based communication, and, consequently, studies exploring the effectiveness of conversational interfaces with agents and embedded agents are rare. Most of the works explore natural language in centralized and simulated MAS. Since the nature of embedded MAS distributes reasoning to the edge of AI systems without depending on centralised infrastructure, KQML is a viable option once it specifies how agents can verbalise their messages using AgentSpeak, for instance. Thus, this mapping indicates that human-BDI-agent interaction remains an underexplored field. In this work, we seek to combine a conversational interface to BDI agents aiming to reduce technical details in collaboration with intelligent agents.

3. The HAI Communication Interface

This work proposes a new approach to facilitate communication between humans and BDI agents. The core idea of this approach is to decouple the human from network configuration and middleware-specific details, offering a conversational interaction platform with devices embedded with MAS. For this, we propose a web-based architecture (Fig. 1) that abstracts the construction and handling of KQML messages, enabling communication with agents embedded in those devices. The communication with the MAS is mediated by a specialized communication-oriented agent that serves as a bridge between the human interface and the IoT infrastructure, interacting with the ContextNet middleware to route messages between users and intelligent agents.

In this architecture, the user interacts via a web interface (front-end) that translates the chosen illocutionary force and propositional content into a KQML message with specific commands and parameters, and forwards it to the cloud server. The Messenger BDI offers a chat-based environment in which the user selects a performative (i.e., tell, askOne, askAll, achieve, and tellHow) and provides a logical expression representing the intended information or request, as shown in Figure 2. The system then translates this input into the KQML communication structure: `<identifier, user_uuid, performative, agent_uuid, message>`. The server maintains a persistent connection with this client, ensuring continuous asynchronous communication. Then, the Cloud Server routes the message to the corresponding agent on the target Edge AI device, which is embedded with a full MAS. The agent's response follows the reverse path, returning to the user interface transparently.

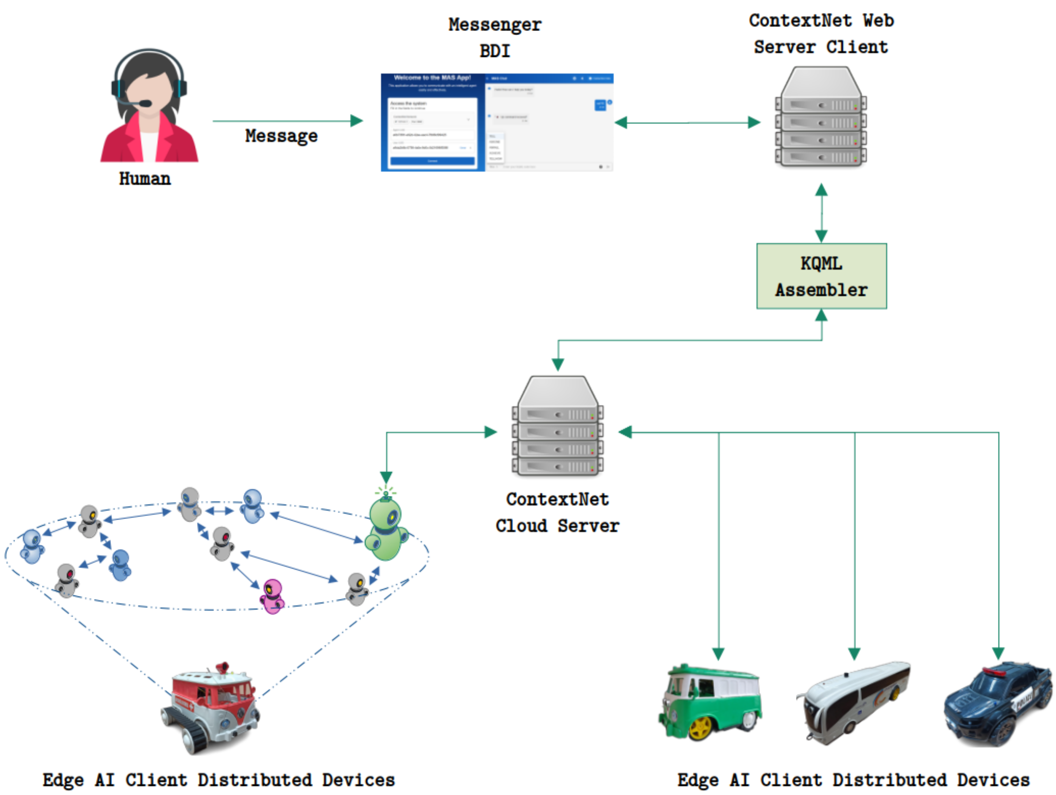

Figure 1. The architecture of the human–agent communication system.

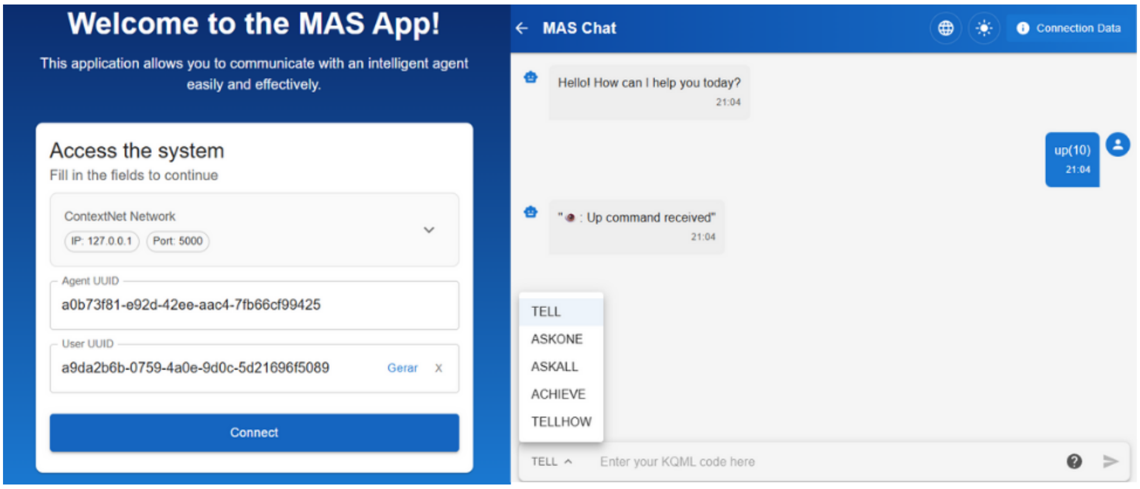

Figure 2. The Messenger BDI chat user interface.

The main contribution of this architecture is to allow humans to interact with distributed IoT MAS. A human can supervise or control an embedded AI device from a distance using our proposed chat, which leverages the IoT architecture in which a server cloud instance manages distributed clients. Besides, since these clients have a MAS on the edge of the system, they can deliberate and operate without relying on the cloud server side.

Instead of handling protocol syntax or network details, the user only provides three parameters:

- IP and Port: the internet address of the ContextNet gateway;
- Agent UUID (Universally Unique Identifier): the universal identifier of the target agent;
- User UUID: the universal identifier of the sender.

Then, the interface establishes a WebSocket connection with the server, which formats the user input into KQML messages and forwards them to ContextNet. Those responses are returned through the same channel and displayed to the user. Auxiliary mechanisms, such as parameter preloading via QR Code or custom URLs, are supported to facilitate configuration.

4. Proof of concept

We extended the Drone agent scenario presented by Santos et al. (Santos et al. 2025) to validate human capabilities for interacting with and controlling MAS using the proposed Messenger BDI. The system consists of an Embedded MAS controlling a 3D UAV model within the Webots simulator, featuring motor control and integrated GPS navigation. In this scenario, the UAV functions simulate a microcontroller: the agent transmits high-level commands (internal actions) for the UAV to execute, and the UAV sends back environmental percepts to the agent.

In this way, the UAV's embedded MAS has an interface agent (Pantoja et al. 2016) (named *Eye*), a communicator agent (named *Teletela*), and a Jason agent (Bordini and Hübner 2006) named *Navigator*. The *Teletela* agent acts as the bridge for external communication via ContextNet. It operates as a direct interface, awaiting commands from the human operator (via Messenger BDI) and forwarding them to the *Navigator*. Conversely, it provides feedback to the human once the command has been processed by the *Navigator*. Meanwhile, *Navigator* manages the navigation logic based on beliefs updated by *Teletela*. It dispatches instructions to *Eye* only when the agent is not busy, retaining the human authority to abort ongoing operations. The *Eye* agent is responsible for flying the UAV by sending *act* commands to the controller when it receives a message from *Navigator*. It stops sending a specific command when the UAV reaches a specified GPS location. This communication is possible because a simulated port was created to connect the agent to the UAV. In this way, the *Eye* agent connects to Webots (Michel 1998) via the */dev/ttyExogenous0* port. Simulated serial ports are used as a means to substitute real hardware with simulators without changing the agent's behavior (Freitas et al. 2023). The MAS running is illustrated in Figure 3.

We defined distinct navigation states and transition commands to ensure the Embedded MAS operation inside WebBot's environment. These commands modify the controller's internal state, triggering specific instruction sets for each step, enabling embedded agents to manipulate the Drone. The initial state **Landed** states that the Drone is on, not

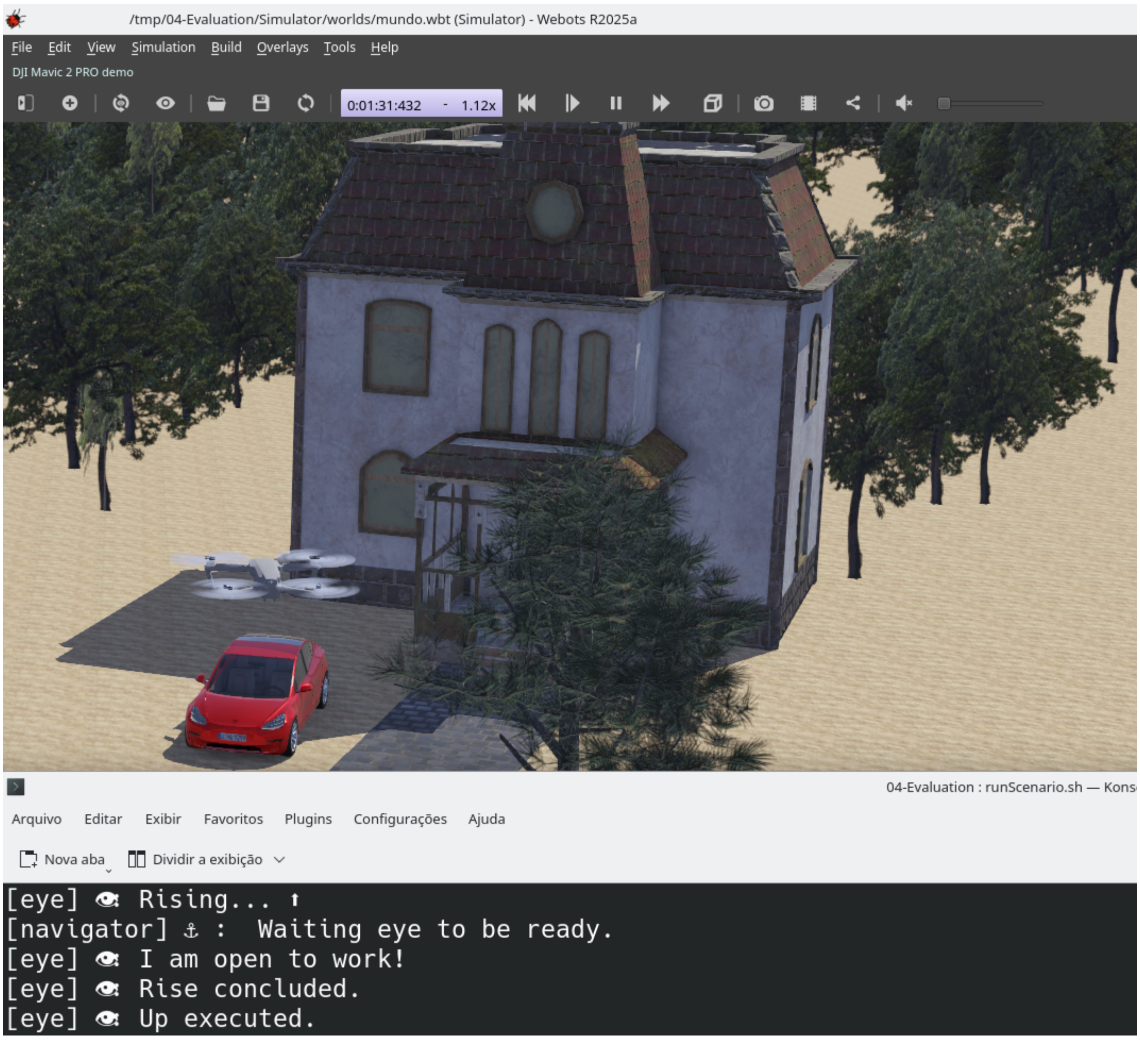

Figure 3. The Drone controlled by a MAS into Webots.

moving, and waiting for a command from the agent. The only available command in this state is for the agent to ask to take off by sending an up command. After takeoff, the Drone automatically enters **hover** mode, where it remains in the air until further commands. The **hover** state is a central state from which all commands originate and return. The Drone can go to the **right**, **left**, **up**, **down**, **forward**, and **backward**. After every command, the Drone returns to the **hover** idle state. To guarantee controlability and mobility, we chose a 5-second waiting delay to reflect the distance travelled. Additionally, when the agent needs to land the Drone, it sends the 'land' command to the Drone. Then it goes to a state **landing**, where it either goes down or waits for a cancel command from the agent. So, the Drone can go down for 5 seconds until it reaches the ground, or return to the landing state to verify whether there is a cancelling command from the agent. Figure **??** shows the Finite State Machine of the Drone's states.

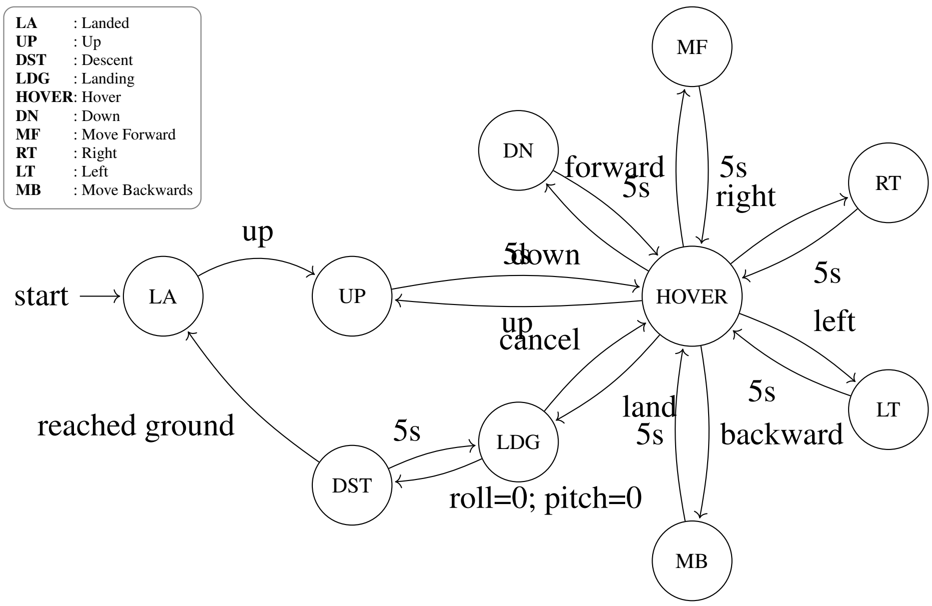

Figure 4. The Finite State Machine representing all states available for controlling the Drone.

To illustrate the integration between the MAS and the Messenger BDI, consider a scenario involving a human operator, named Sergeant, issuing instructions via the Messenger BDI chat interface to a Drone. In the first case, the Sergeant sends a command to the Drone to lift to 10 meters. So, the Sergeant inputs `up(10)`. Since the Drone is currently grounded, it autonomously injects a `takeOff` routine before executing the ascent, which initiates the embedded motors and performs the `UP` command. Internally, each increment forms a continuous feedback loop in which the controller oscillates between `HOVERING` and `UP` states to maintain mechanical stability until the requested altitude is reached.

However, in a simulated urgent operational context, the Sergeant declines to wait for the ascent to complete and modify the ongoing commands. So, the Sergeant queues a sequence of commands: `down(3)` to descend, `left(4)` to move left, `right(8)` to move right, `forward(6)` to move forward, and `backward(5)` to move backwards, followed by a termination `land` command. The *Navigator* acts as a buffer, receiving these commands and dispatching them to the *Eye* only when it is not busy, which sends the low-level state transitions to the controller. It ensures that neither the embedded system nor the *Eye* is overloaded by the asynchronous input. Finally, the Sergeant decides to abort the sequence nears completion by sending a `cancel` command, the pending `land` instruction is instantly abolished from the Navigator's queue. The MAS halts the current trajectory in the next cycle and holds its position, effectively resetting the agent to an idle state, awaiting the next mission directive.

5. Reproducibility

To ensure the reproducibility of this work, the source code of the Embedded MAS, the simulator implementation, the Messenger BDI (front-end and back-end services), and the complete source code are publicly accessible in separate repositories for the web client and server components. These repositories provide documentation and support for installation, experimentation, and further development.[1]

6. Conclusion

This work presented an architecture to enable communication between humans and MAS. The Messenger BDI abstracts KQML message formatting and ContextNet routing, allowing users to interact with agents by using a chat-like interface. As a proof-of-concept, we proposed that a human control an UAV embedded with a MAS in the Webots simulator. The results demonstrated that the MAS could interpret and execute human commands. Despite the inherent latency of distributed execution, the UAV successfully performed navigation tasks by relying on internal stabilization mechanisms and intermediate hovering states. The experiments validated the feasibility of using conversational interfaces to control and supervision autonomous BDI systems.

As future works, we indicate refining the command execution lifecycle, expanding support for additional robotic platforms, exploring more autonomous decision-making behaviors within the MAS, and enhancing communication mechanisms with the MAS. Similarly, research on human–agent teamwork in collaborative environments shows that the development of Shared Mental Models and the inclusion of multimodal (verbal and non-verbal) channels significantly improve coordination and performance (Hanna and Richards 2014). Concurrently, future work includes improving natural language understanding and large language models for chat-based interfaces.

From an ethical point of view, Messenger BDI can be seen as an initial advance in promoting human participation in MAS decision-making by allowing individuals to send commands to be executed by the system. However, in certain contexts, it may be necessary for interventions to be processed immediately by the MAS, so issues related to prioritization and levels of human oversight still need to be investigated. Future studies can explore different supervisory approaches (e.g., HOTL, HITL) in different contexts.

References

Barange, M., Kabil, A., De Keukelaere, C., and Chevaillier, P. (2014). Task-Oriented Conversational Behavior of Agents for Collaboration in Human-Agent Teamwork. In *PAAMS 2014*. Springer. DOI: 10.1007/978-3-319-07551-8_3.

[1]Available at: `https://papers.chon.group/WESAAC/2026/MessengerBDI/`

Berger, A. and Pesty, S. (2005). Towards a Conversational Language for Artificial Agents in Mixed Community. In *CEEMAS 2005*. Springer. DOI: 10.1007/11559221_4.

Bordini, R. H. and Hübner, J. F. (2006). BDI Agent Programming in AgentSpeak Using Jason. In *CLIMA 2005*. Springer. DOI: 10.1007/11750734_9.

Bratman, M. E., Israel, D. J., and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(3):349–355. DOI: 10.1111/j.1467-8640.1988.tb00284.x.

Cardoso, R. C. and Ferrando, A. (2021). A Review of Agent-Based Programming for Multi-Agent Systems. *Computers*, 10(2):16. DOI: 10.3390/computers10020016.

Chang, Y., Li, Z., Lee, S., Park, J., and Yun, H. (2024). Exploring Human-AI Companion Dynamics: The Interplay between Chatbots and Users. In *PACIS 2024 Proceedings*. AIS Electronic Library. Available at: `https://aisel.aisnet.org/pacis2024/track19_userbeh/track19_userbeh/1`.

Chin, H., Zhunis, A., and Cha, M. (2024). Behaviors and Perceptions of Human-Chatbot Interactions Based on Top Active Users of a Commercial Social Chatbot. *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW2):1–28. DOI: 10.1145/3687022.

De Silva, L., Meneguzzi, F., and Logan, B. (2020). BDI agent architectures: a survey. In *IJCAI'20*. ACM. URL: https://dl.acm.org/doi/abs/10.5555/3491440.3492124.

Endler, M., Baptista, G., Silva, L. D., Vasconcelos, R., Malcher, M., Pantoja, V., Pinheiro, V., and Viterbo, J. (2011). ContextNet: Context Reasoning and Sharing Middleware for Large-Scale Pervasive Collaboration and Social Networking. In *PDT '11*, New York, NY, USA. ACM. DOI: 10.1145/2088960.2088962.

Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML as an agent communication language. In *CIKM '94*, pages 456–463, Gaithersburg, Maryland, United States. ACM Press. DOI: 10.1145/191246.191322.

Flathmann, C., McNeese, N., and Canonico, L. B. (2019). Using Human-Agent Teams to Purposefully Design Multi-Agent Systems. *Proceedings of the HFES Annual Meeting*, 63(1):1425–1429. DOI: 10.1177/1071181319631238.

Freitas, B., Lazarin, N., and Pantoja, C. (2023). Uma Proposta de Emulador de Portas Seriais para Sistemas Multiagentes Embarcados. In *Anais do XVII WESAAC*, pages 55–66, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/wesaac.2023.33437.

Freitas, B., Lazarin, N., Pantoja, C., Jatoba, N., Pinto, V., and Viterbo, J. (2025a). A Migration Control Approach to Improve Security for Open MAS in IoT Networks. In *PAAMS 2025*. Springer. DOI: 10.1007/978-3-032-07638-0_21.

Freitas, B., Lazarin, N., Pantoja, C., and Viterbo, J. (2025b). Integrating Simulated Exogenous Environments to Support the Learning Process of the Embedded MAS Approach. In *WEI 2025*. SBC. DOI: 10.5753/wei.2025.9115.

Gatti, A., Mascardi, V., and Ferrando, A. (2025). ChatBDI: Think BDI, Talk LLM. In *Proceedings of the 24th AAMAS*, page 2541–2543, Richland, SC. IFAAMAS. URL: https://dl.acm.org/doi/10.5555/3709347.3743930.

Goel, P. and Ganatra, A. (2021). A Survey on Chatbot: Futuristic Conversational Agent for User Interaction. In *ICSPC 2021*, pages 736–740, Coimbatore, India. IEEE. DOI: 10.1109/ICSPC51351.2021.9451763.

Hanna, N. and Richards, D. (2014). Human-agent teamwork in collaborative virtual environments. In *AAMAS 2014*, pages 1735–1736. IFAAMAS. URL: `https://scholar.google.com/scholar?cluster=14588271285554598075`.

Jesus, V. S. d., Pantoja, C., Manoel, F., Alves, G., Viterbo, J., and Bezerra, E. (2021). Bio-Inspired Protocols for Embodied Multi-Agent Systems. In *ICAART 2021*, pages 312–320. SciTePress. DOI: 10.5220/0010257803120320.

Johnson, P. and Johnson, H. (2001). Interaction, collaboration and communication: A human-computer interaction perspective. In *ACCOLADE: ARCHITECTURE, COLLABORATION, DESIGN*, pages 43–58, NETHERLANDS. DELFT. URL: https://scholar.google.com/scholar?cluster=3372667578355900311.

Lazarin, N. M., de Souza, J. P. B., Pantoja, C. E., Alexandre, T., and Viterbo, J. (2025). Human and BDI-Agent Interaction via KQML Messages over IMAP and SMTP. In *PAAMS 2024*. DOI: 10.1007/978-3-031-70415-4_15.

Lazarin, N. M., Pantoja, C. E., and Viterbo, J. (2024). Dealing with the Unpredictability of Physical Resources in Real-World Multi-agent Systems. In *Agents and Artificial Intelligence*, pages 48–71, Cham. Springer. DOI: 10.1007/978-3-031-55326-4_3.

Lewis, M. (1998). Designing for Human-Agent Interaction. *AI Magazine*, 19(2):67. DOI: 10.1609/aimag.v19i2.1369.

Lira, L., Serafim, A., Lazarin, N., Freitas, B., and Pantoja, C. (2025). BusAI: An BDI-agent-based Urban Transport Information System. In *WESAAC 2025*, pages 137–144. SBC. DOI: 10.5753/wesaac.2025.37550.

Ltifi, M. (2023). Trust in the chatbot: a semi-human relationship. *Future Business Journal*, 9(1):109. DOI: 10.1186/s43093-023-00288-z.

Michel, O. (1998). Webots: Symbiosis Between Virtual and Real Mobile Robots. In *VW 1998*. Springer. DOI: 10.1007/3-540-68686-X_24.

Pantoja, C. E., Jesus, V. S. d., Lazarin, N. M., and Viterbo, J. (2023). A Spin-off Version of Jason for IoT and Embedded Multi-Agent Systems. In *Intelligent Systems*, pages 382–396. Springer Nature Switzerland. DOI: 10.1007/978-3-031-45368-7_25.

Pantoja, C. E., Stabile, M. F., Lazarin, N. M., and Sichman, J. S. (2016). Argo: An extended jason architecture that facilitates embedded robotic agents programming. In *Engineering Multi-Agent Systems*, pages 136–155, Cham. Springer International Publishing. DOI: 10.1007/978-3-319-50983-9_8.

Santos, K., Costa, D., Lazarin, N., Freitas, B., and Pantoja, C. (2025). The BIG Agent: The Utilization of Embedded Multi-Agent Systems in Unmanned Aerial Vehicles for Autonomous Operations. In *WESAAC 2025*. DOI: 10.5753/wesaac.2025.37556.

Sichman, J. S., Demazeau, Y., and Boissier, O. (1992). When can knowledge-based systems be called agents? In *SBIA'92*, volume 9, pages 172–185, Rio de Janeiro. SBC. https://scholar.google.com.br/scholar?cluster=12989357293789094729.

Song, I. and Piggott, P. (2003). Modelling Message Handling System. In *AI 2003*, pages 53–64. Springer. DOI: 10.1007/978-3-540-24581-0_5.

Urlings, P., Sioutis, C., Tweedale, J., Ichalkaranje, N., and Jain, L. (2005). A future framework for interfacing BDI agents in a real-time teaming environment. *JNCA*, 29(2-3). DOI: 10.1016/j.jnca.2004.10.005.

Wooldridge, M. (1999). Intelligent Agents. In *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 27–77. MIT Press, Cambridge, Mass.