Towards a Toolkit for Teaching AI Supported by Robotic-agents: Proposal and First Impressions

Nilson Mori Lazarin[1,2], **Carlos Eduardo Pantoja**[1,2], **José Viterbo**[1]

[1]Institute of Computing - Fluminense Federal University (UFF)
Niterói – Brazil

[2]Federal Center for Technological Education Celso Suckow da Fonseca (Cefet/RJ)
Rio de Janeiro – Brazil

{nilson.lazarin,carlos.pantoja}@cefet-rj.br, viterbo@ic.uff.br

Abstract. *Using hardware platforms combined with other paradigms stimulates learning and has become an apparent trend in education. The Multi-Agent Systems domain studies the use of autonomous entities to solve complex problems. When combined with the Belief-Desire-Intention (BDI), a cognitive model close to human thinking, it can contribute to facilitating the teaching of Artificial Intelligence (AI) and Computational Thinking (CT). This work presents a toolkit to support the teaching of AI supported by BDI intelligent agents using a prototyping approach. Four workshops were held at different levels, and a technological acceptance questionnaire was designed to analyze the proposed toolkit.*

1. Introduction

Artificial Intelligence (AI) concepts have been presented as a motivating element for Computational Thinking (CT) practices in the classroom. The integration between AI and CT tends to cooperate to construct knowledge, simplifying abstraction and algorithmic thinking, especially in high school classes [Caruso and Cavalheiro 2021]. One of the fields of AI is Intelligent Agents, which are virtual or physical entities, autonomous, endowed with cognitive abilities, social ability, and adaptability to adverse situations. Such agents can be part of a Multi-agent System (MAS), where intelligent agents coexist in pursuit of their individual and collective goals, whether mutual or conflicting [Wooldridge 2009].

Over time, cognitive models that try to understand the functioning of human reasoning have been studied in sociology and philosophy. Remarkably, the Belief-Desire-Intention model (BDI) [Bratman 1987] models the functioning of the human psyche through its beliefs, desires, and intentions in practical reasoning. It describes how the experience and knowledge obtained from social interactions and with the world can influence decision-making, reflected in a chain of plans and actions to achieve individual goals. Technologically, there are frameworks for developing intelligent agents that allow applications in different domains to be created, such as community [Pereira dos Santos et al. 2016], biology [Souza de Jesus. et al. 2021], social modeling [Rocha Costa 2022], among others.

The BDI model adds a cognition layer closer to human thinking to software and hardware computational entities. Educationally, a BDI agent-based approach allows abstractions such as beliefs and intentions to be used in constructing computational entities.

It can lead to a natural learning of the process of building intelligent applications, as the student can understand the reasoning from explaining how he does it in the real world.

Teaching supported by hardware platforms (e.g., Arduino, Raspberry Pi) is becoming a technological learning trend that contributes to improving the teaching-learning process, promoting proactive thinking in the classroom, and training students in the search for solutions to real-world problems [Korom and Illés 2022, Novák et al. 2018, Al-Masri et al. 2020]. Allied with this, adopting emerging technologies in the application of project-based learning, such as the Internet of Things (IoT), cyber-physical systems, and ubiquitous computing, among others, contribute to the cohesion and intersection of computing with other areas [Al-Masri et al. 2020]. However, regarding solutions that support embedding MAS in hardware, there is still a lack of an integrated tool that facilitates the teaching and development of applications supported by intelligent agents [Brandão et al. 2021].

Within this scenario, this work aims to present a proposal of a toolkit applied to teaching of AI supported by BDI intelligent agents using a prototyping approach. The approach consists of programming technologies to support the development of Embedded MAS and a teaching methodology mixing theory and practice. Our philosophy relays on the ability to add cognition to collaborative networked devices — *Cognitive Hardware On Network* (Chon). The toolkit comprises an operating system distribution, a development interface, a middleware IoT, and a standard prototype architecture. Five workshops were held to analyze the toolkit and the methodology: three in scientific workshops, one during an undergraduate class in an Information Systems course, and the last during a master's class in Computing. A Technology Acceptance Model (TAM) questionnaire was applied to analyze the participants' responses regarding the acceptance of the proposed toolkit.

This work is structured as follows: some related works are presented in Section 2; Section 3 presents the toolkit proposed in this paper; the experiments and the results of the toolkit evaluation are presented in Section 4; Finally, the conclusions are presented in Section 5.

2. Related works

There are several approaches to promoting CT and teaching AI and agents by adopting hardware and robotic platforms. For example, Frankie [Pimentel et al. 2018] is an intelligent robot used in Educational Robotics activities to present AI concepts and intelligent agents to high school students. The approach is divided into two steps: in the first, students train the robot to recognize pre-established figures; in the second, the robot runs on a playful carpet with several possible destinations, where students must present images for the robot to recognize the path to follow. A motor, a display, an ultrasonic sensor, a line follower, and a camera are used in the hardware layer. The firmware layer uses the Arduino Mega 2560. The interface layer uses serial communication via Python commands. The reasoning layer is hosted on a Single-Board Computer (SBC) Raspberry Pi and uses a neural network to interpret figures that indicate what to do next in the environment.

Robotic platforms that integrate agents, including educational solutions, usually use a four-layer architecture (hardware, firmware, serial communication, and reasoning) to interconnect the reasoning of high-level cognitive models to control physical devices of hardware to try to abstract user intervention in these layers [Pantoja et al. 2016, Onyedinma et al. 2020, Silva et al. 2020]. From the student's perspective, the greater the technical abstractions at the structural level, the tendency for the CT learning process to be optimized. For example, in the case of Frankie, the student needs to carry out manual training directly on Raspberry, requiring prior knowledge of operating systems besides AI structures. Similarly, the student only has access to this system in a wired way when physically connecting a monitor. Ideally, the student should focus on AI and CT in the learning process, and the architectural and technological integration issues should be abstracted during the teaching process.

The Multi-agent System for non-Holonomic Racing (MuSHR) is a full-stack robotic platform applied in AI and Mobile Robot classes at undergraduate and graduate levels, which uses rapid prototyping techniques acting on all layers [Srinivasa et al. 2019]. A VESC Driver hosts the firmware layer. In the Interface layer, messages from the Robot Operating System (ROS) are used, and the reasoning layer is hosted on an Nvidia Jetson Nano. Although the educational prototype is standardized, abstracting some levels of student intervention (hardware and interfacing), technical and specialized knowledge in certain technologies is still required. Highly specialized technologies are excellent candidates for teaching support tools at the undergraduate and graduate levels. When the toolkit is intended to be applied at Elementary, Technical, and High School levels, the need for prior knowledge or the learning curve of such technologies can muddle the AI learning process. It may cause disinterest due to the delay in achieving results or require high student commitment.

Besides, other initiatives for teaching AI and BDI that do not use hardware exist. One example is Edubot [Lima et al. 2018], a serious game that, through the Problem-Based Learning approach, enables the practice of concepts of intelligent agents from the AI class of a bachelor's degree course. In it, the player must program the agent's action logic according to the availability of sensors and logical actuators. The first phase requires the student to program an agent using conditional rules based on the agent's perceptions to avoid contact with moving obstacles in the scenario. The other phase requires the student to map the scenario and build a model where the agent, when finding an object, must return to the initial position, avoiding contact with static obstacles. Edubot abstracts the entire structural part of the student, providing a simulated panel displaying the steps that agents perform dynamically. It allows the student to focus on the development activities of the agent's reasoning layer. However, in practical experimentation with 22 students, a previous study of the syntax of the Python language was recommended, which uses traditional algorithms and object orientation structures.

Finally, using cognitive models like the BDI, an IDE [Burattini et al. 2022] presents a visual programming approach, allowing students with no experience in agent-oriented programming to develop autonomous systems. Each agent is composed of an initial block and a library of plans. The initial block defines the initial beliefs and plans to which the agent is committed. Each agent's plan is constructed and stored in separate blocks, forming the agent's procedural knowledge library. Visual tools based on building the CT by composing blocks have been successful at different educational levels [Queiroz et al. 2019]. Nevertheless, the expected result of applying block approaches to AI thinking supported by agents tends to be positive, but it still lacks tool support to deal with embedded systems.

3. Proposal of a toolkit for teaching AI supported by robotic-agents

This section will present the technological toolkit for teaching AI and intelligent agents embedded in robotic platforms to act at different levels of education. It abstracts the student's intervention at technological levels to facilitate the learning process. However, depending on the educational level, it still leaves the possibility of intervention for the teacher or professor to decide.

The toolkit aims to make the learning process of the concepts of autonomous and cognitive agents practically and tangibly feasible. It includes a GNU/Linux distribution that integrates several solutions to abstract technological dependencies; a development environment; an agent-oriented programming language adapted to control hardware; a communication infrastructure using IoT, and standardized models of low-cost prototypes.

In this scenario, the student can use a computer with the development environment for programming cognition and send it remotely to the prototypes without needing a physical wired connection or disassembly. With the toolkit's support, the student can program the interaction between the prototypes using the IoT. However, the students do not need to know the technical details. The student can focus on the reasoning layer with standardized prototype models. Furthermore, if want, the student can program the firmware layer too. It allows flexibility of the toolkit for various educational purposes, supporting the production of CT in AI and the programming of robotic platforms. Figure 1 presents the interconnection of all tools in an educational environment.

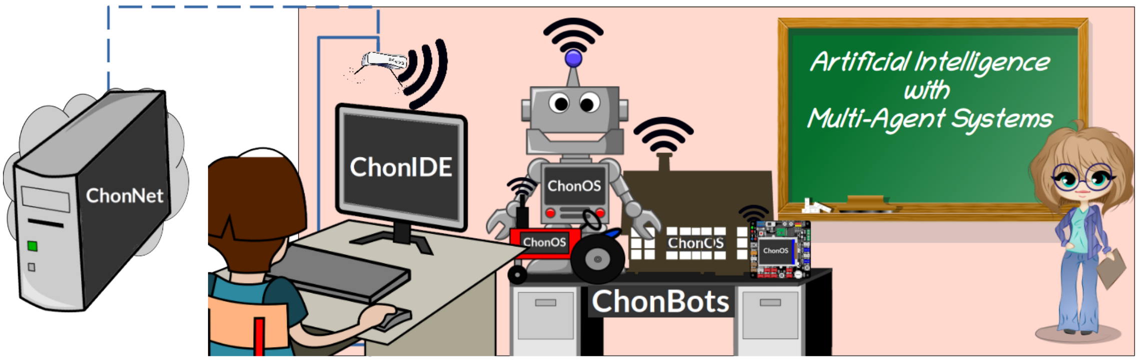

Figura 1. Use of Chon toolkit for practical teaching of intelligent agents.

Using the development environment (ChonIDE), the student can program the reasoning of the prototypes (ChonBots) remotely through a wi-fi connection. These prototypes can collaborate to perform complex tasks through the IoT network (ChonNet). The hardware of the prototypes is standardized to facilitate the manipulation and learning of AI. It executes a specific-purpose operating system to support the development of these activities (ChonOS). Technologically described below are each of the components of the toolkit:

- ChonBots: *Chon - Basic Prototypes* (`http://bots.chon.group/`) are low-cost prototype models for Embedded MAS. They comprise an SBC, a prototyping platform, sensors, and actuators. Each model has a list of supported actions so that a robotic agent (thinking layer) could manage the prototype. In addition, the schematics from the hardware layer, the source code from the firmware layer, and the library from the interface layer are provided. Figures 2a, 2b and 2c present the schematic, a 2WD prototype built, and the list of actions still supported. The prototype had light, distance, and line-follower sensors. The actuators are LED, buzzer, and 2WD robotic platform.

(a) Schematic. **(b) Prototype.** **(c) Supported Actions and Perceptions.**

Figura 2. Schematic, prototype, actions and perceptions of the ChonBot 2WD.

- ChonOS: *Chon - Operational System* (`http://os.chon.group/`) is a specific-purpose GNU/Linux distribution for Embedded MAS. It uses Jason Embedded, an extended version [Bordini and Hübner 2006] of an AgentSpeak(L) language [Rao 1996] interpreter, to enable agents to control embedded systems, communicability, and mobility over an IoT infrastructure. This distribution facilitates device configuration; loading the firmware and reasoning; and Embedded MAS management (start, stop, or debug). All these functionalities are carried out remotely without disassembling or turning off the prototype. In the first run, ChonOS starts a Wi-Fi Hotspot where the teacher has access to the configuration panel to define a name for the prototype and include it in a nearby wireless local network.
- ChonIDE: *Chon - Integrated Development Environment* (`http://ide.chon.group/`) is a development environment for the firmware and reasoning layers of the Embedded MAS. The student can develop, send, start, or stop the execution of reasoning and develop, compile, or deploy the firmware [Souza de Jesus et al. 2022]. The development environment can be installed and executed on a computer, transmitting the software for execution on the prototype, or accessed directly via a mobile phone or browser.
- ChonNet: *Chon - Network Services* (`http://net.chon.group/`) is a layer of cloud services that implements a dynamic device name update architecture [Lu and Liu 2009] and a context services architecture for collaborative, pervasive, and large-scale applications [Endler et al. 2011]. ChonNet facilitates the management and development of the prototypes because even if it moves through different networks, the connection is made by mnemonics (robot name) instead of a network address that can change each time it is used. Furthermore, it enables communication protocols [Souza de Jesus. et al. 2021] between intelligent agents from different Embedded MAS, as it assigns a universal identifier, allowing IoT communication inside or outside the local network.

All the technical details of the toolkit are abstracted from the student, who needs to focus only on learning AI and their allowed reflection on how cognitive models thought of in human thinking can be used to generate computational entities with the autonomous capacity to control. It is possible to explore teaching methodologies applied at different educational levels by simplifying access to technical tools. At levels such as undergraduate and graduate, it is evident, given the diverse previous educational experiences, that there is an application and facilitation of learning. However, a smaller learning curve is expected on another educational level, given the possibility of the student performing the CT using a cognitive model close to human thinking.

3.1. AI teaching approach supported by robotic agents

A practical approach contributes to transforming theoretical concepts into examples of easy assimilation, stimulates learning, increases commitment, and improves interaction in the classroom [Fernandes et al. 2018]. This section presents a method that seeks to enable the teaching of AI supported by using intelligent agents in a practical approach.

Figure 3 presents the proposed approach with six steps, from the conceptualization of AI and IoT to the development of integrated cognitive systems. Each step of the approach is described below:

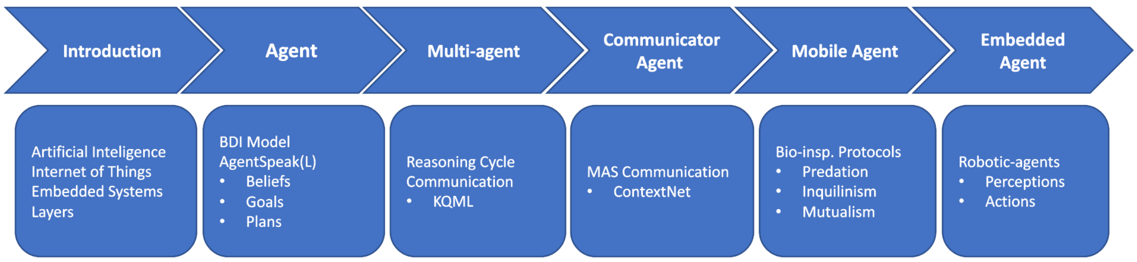

Figura 3. Teaching AI supported by robotic agents.

1. Introduction: Initially, introductory concepts about the different paradigms that enable the development of projects for embedded cognitive systems and their layers are presented. The practical activity in this step involves installing ChonIDE on the student's computer and running the *hello world*. The objective is for the student to learn to use the development environment, consult the MAS logs, and inspect the mind of a BDI agent.
2. Agent: The second step presents concepts about intelligent agents, BDI architecture, and the AgentSpeak(L) language. The hands-on activity involves programming a reactive agent. The goal is for the student to be able to manipulate the beliefs, goals, and plans of a BDI agent.
3. Multi-agent: This step presents the concepts of the BDI agent's reasoning cycle and the communication between agents of the same system. The practical activity involves programming a MAS with two agents that share beliefs and plans. The objective is for the student to know the BDI agent's communication.
4. Communicator: The fourth step presents concepts about communication between different multi-agent systems through IoT communication middleware. The practical activity, preferably in pairs, involves programming two MAS who share beliefs and plans. The objective is for the student to know the communication between different MAS.

5. Mobile: Later, the concepts of mobile agents and Open MAS are presented. The practical activity, preferably performed in pairs, involves programming two MAS that allow agents to enter and exit. The objective is for the student to know bio-inspired communication protocols for agents.
6. Embedded: Finally, concepts about robotic agents, actions, and perceptions of the environment are presented. The practical activity preferably carried out in pairs, involves installing or configuring ChonOS and programming two MAS, one capable of handling the prototype (previously built) and another that requests the execution of tasks. The objective is to enable the student to program embedded cognitive systems.

4. Workshops

We performed five workshops to evaluate the toolkit use and the proposed approach. Three are offered as a mini-course, and two during a regular course. The first workshop was held during the *XVI Workshop-School on Agents, Environments and Applications* (WESAAC 2022), held at UFSC in Blumenau. The audience was mostly made up of specialists in the field of MAS. The objective was to analyze the acceptance of using the proposed toolkit for Embedded MAS and gather suggestions for improvements. ChonIDE was in the alpha version during the first edition, allowing only the upload and deployment of the reasoning and firmware layers. It was still necessary to code the MAS in an additional editor, zip the project folder and send it to the prototype.

The objective of the last four workshops was to analyze the improvements made (ChonIDE beta version) in the toolkit and gather feedback from students. The second workshop was held during a Scientific Workshop at UTFPR in Ponta Grossa. The audience was primarily undergraduate and graduate students with experience in MAS. The third workshop was held during a class on Multi-agent Systems at the Institute of Computing at UFF. The audience consisted of graduate students in their first contact with MAS. The fourth workshop was held during a class on Specialist Systems for the bachelor's degree in Information Systems at Cefet/RJ. The audience consisted of undergraduate students in their first contact with MAS. The last workshop was at IFSP in Campos do Jordão with undergraduate, technical, and high school students and too last-grade students in elementary school, everyone in their first contact with MAS.

4.1. Acceptance of the proposed tollkit

A test using the Technology Acceptance Model (TAM) was applied to evaluate the use of the toolkit. The questionnaire[1] was performed anonymously and online after the completion of the workshop. We sought to analyze the participant's profiles on their performance level, familiarity with the agent-oriented paradigm compared to other paradigms, and experience in development. We sought to assess: acceptance of the command and firmware layer management functionalities; the usability perception of the toolkit; and finally, the intention to use the toolkit.

The results of the profile are as follows: 100% of the interviewees knew programming and the object-oriented paradigm; 75% knew of the agent-oriented paradigm; 87% already had some experience with embedded systems; finally, 63% already had experience with multi-agent and embedded multi-agent systems.

[1] `http://csbc23.chon.group` - presentations and photos of the workshops are also available.

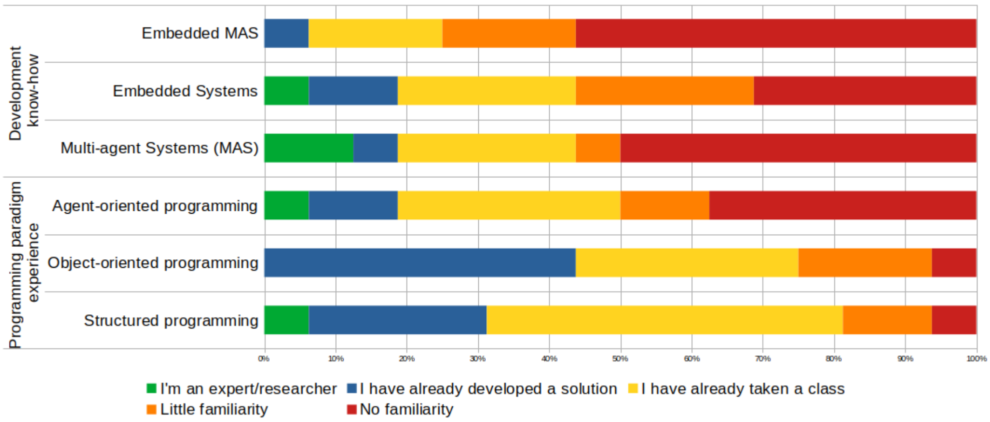

Figura 4. Level of experience in programming paradigm and development know-how of workshop participants.

About the toolkit functionalities, the respondents: 100% strongly agreed that the UPLOAD functionalities, Boards and libraries management, compilation, and firmware deployment facilitate the development; 100% agreed that MAS code editor, Mind Inspector, and MAS Log make development easier. 85% agree that remotely starting and stopping MAS facilitates development.

About the perception of the use: 100% strongly agree that using the toolkit facilitates the development process, and 100% agree that the usability is easy, and they learned to use ChonIDE easily. About the intention of use: 87% strongly agree that they would use the toolkit daily; Finally, 100% strongly agree that they would recommend using it.

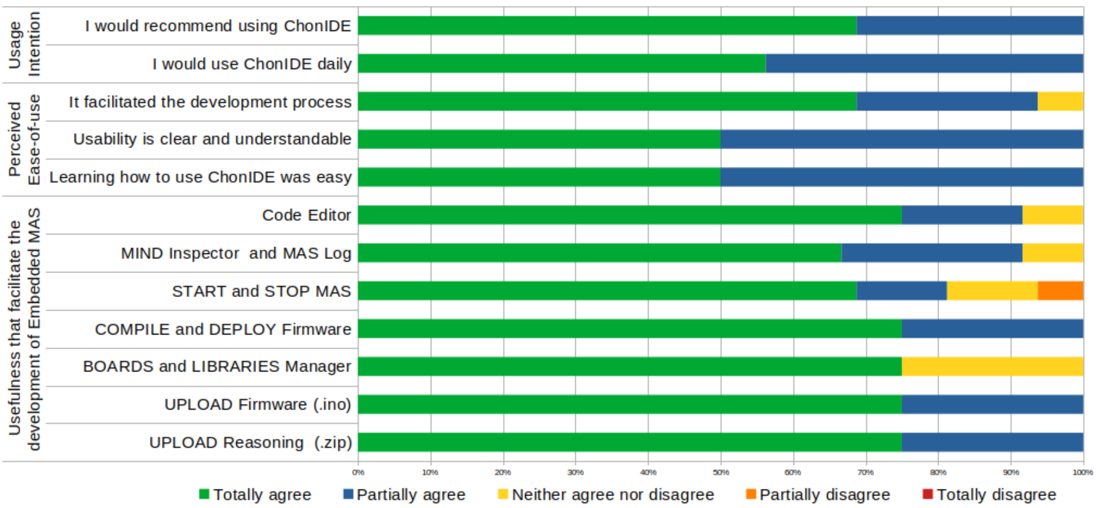

Figura 5. Acceptance results about the proposed tool kit.

5. Conclusions

This paper presented a set of tools for teaching AI using intelligent agents embedded in robotic platforms. The toolkit makes it possible to abstract the need for student intervention in technological parts. The toolkit consists of a GNU/Linux distribution, an IDE, an IoT communication service, a teaching methodology, and low-cost prototype models.

Currently, the toolkit has been validated by users with experience in agent-oriented programming and by undergraduate and graduate users. The results were positive, and it is believed that this practical approach can facilitate teaching AI to students at other levels of education.

Future works may evaluate the use of the toolkit in teaching AI in high school/technical students' classes. In addition, can address educational aspects and teachers' perceptions of using the toolkit. Another possibility is evaluating the tool's usability integrated with other classes, such as robotics and networks.

Referências

Al-Masri, E., Kabu, S., and Dixith, P. (2020). Emerging hardware prototyping technologies as tools for learning. *IEEE Access*, 8:80207–80217.

Bordini, R. H. and Hübner, J. F. (2006). BDI Agent Programming in AgentSpeak using *Jason*. In Toni, F. and Torroni, P., editors, *Computational Logic in Multi-Agent Systems*, pages 143–164, Berlin, Heidelberg. Springer Berlin Heidelberg.

Brandão, F. C., Lima, M. A. T., Pantoja, C. E., Zahn, J., and Viterbo, J. (2021). Engineering approaches for programming agent-based iot objects using the resource management architecture. *Sensors*, 21(23).

Bratman, M. (1987). *Intention, Plans, and Practical Reason*. Cambridge: Cambridge, MA: Harvard University Press.

Burattini, S., Ricci, A., Mayer, S., Vachtsevanou, D., Lemee, J., Ciortea, A., and Croatti, A. (2022). Agent-Oriented Visual Programming for the Web of Things. Online.

Caruso, A. and Cavalheiro, S. (2021). Integração entre pensamento computacional e inteligência artificial: uma revisão sistemática de literatura. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação*, pages 1051–1062, Porto Alegre, RS, Brasil. SBC.

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 *Proceedings of the Workshop on Posters and Demos Track*, PDT '11, New York, NY, USA. ACM.

Fernandes, M., Santos, C., Souza, E., and Fonseca, M. (2018). Robótica educacional uma ferramenta para ensino de lógica de programação no ensino fundamental. In *Anais do XXIV Workshop de Informática na Escola*, pages 315–322, Porto Alegre, RS, Brasil. SBC.

Korom, S. and Illés, Z. (2022). Methods of teaching programming with raspberry pi. In *2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–5.

Lima, T. M. d. S., Menezes, R. d. F. d. S., Filho, A. O. B., Viana, D., and Bottentuit Junior, J. B. (2018). Edubot: um estudo prático de aprendizagem baseada em problemas no contexto de agentes inteligentes e jogos sérios. *Revista Tecnologias na Educação*, 27.

Lu, Z. and Liu, J. (2009). Design and implementation of dynamic domain name system based on bind. In *2009 International Symposium on Computer Network and Multimedia Technology*, pages 1–4.

Novák, M., Kalová, J., and Pech, J. (2018). Use of the arduino platform in teaching programming. In *2018 IV International Conference on Information Technologies in Engineering Education (Inforino)*, pages 1–4.

Onyedinma, C., Gavigan, P., and Esfandiari, B. (2020). Toward campus mail delivery using bdi. *Journal of Sensor and Actuator Networks*, 9(4).

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 Baldoni, M., Müller, J. P., Nunes, I., and Zalila-Wenkstern, R., editors, *Engineering Multi-Agent Systems*, pages 136–155, Cham. Springer International Publishing.

Pereira dos Santos, F., Adamatti, D., Rodrigues, H., Dimuro, G., De Manuel Jerez, E., and Dimuro, G. (2016). A multiagent-based tool for the simulation of social production and management of urban ecosystems: A case study on san jerónimo vegetable garden - seville, spain. *Journal of Artificial Societies and Social Simulation*, 19(3):12.

Pimentel, C. S., Queiroz, R. L., Lima, P. M. V., and Ferrentini Sampaio, F. (2018). Projeto Frankie: uma proposta para o ensino de Inteligência Artificial na Educação Básica. In *Nuevas Ideas en Informática Educativa*, volume 14, pages 493–498, Santiago.

Queiroz, R., Sampaio, F., and dos Santos, M. (2019). DuinoBlocks4Kids: Using Free Technology and Low-Cost Materials for the Exercise of Computational Thinking in Primary School via Programming and Educational Robotics. *Revista Brasileira de Informática na Educação*, 27(02):167.

Rao, A. S. (1996). Agentspeak(l): Bdi agents speak out in a logical computable language. In Van de Velde, W. and Perram, J. W., editors, *Agents Breaking Away*, pages 42–55, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rocha Costa, A. C. d. (2022). Tpo: A type system for the architecture of agent societies. In Alechina, N., Baldoni, M., and Logan, B., editors, *Engineering Multi-Agent Systems*, pages 292–311, Cham. Springer International Publishing.

Silva, G. R., Becker, L. B., and Hübner, J. F. (2020). Embedded architecture composed of cognitive agents and ros for programming intelligent robots. *IFAC-PapersOnLine*, 53(2):10000–10005. 21st IFAC World Congress.

Souza de Jesus, V., Lazarin, N. M., Pantoja, C. E., Manoel, F. C. P. B., Alves, G. V., Ramos, G., and Viterbo, J. (2022). Proposta de uma IDE para Desenvolvimento de SMA Embarcados. In *Anais do XVI Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações (WESAAC 2022)*, Blumenau. UFSC.

Souza de Jesus., V., Pantoja., C. E., Manoel., F., Alves., G. V., Viterbo., J., and Bezerra., E. (2021). Bio-inspired protocols for embodied multi-agent systems. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,*, pages 312–320. INSTICC, SciTePress.

Srinivasa, S. S., Lancaster, P., Michalove, J., Schmittle, M., Summers, C., Rockett, M., Smith, J. R., Chouhury, S., Mavrogiannis, C., and Sadeghi, F. (2019). MuSHR: A low-cost, open-source robotic racecar for education and research. *CoRR*, abs/1908.08031.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John wiley & sons.