Vol-3261
urn:nbn:de:0074-3261-7

WOA 2022
23rd Workshop "From Objects to Agents"

Proceedings of the 23rd Workshop "From Objects to Agents"

Genova, Italy, September 1-3, 2022.

Edited by

Angelo Ferrando *
Viviana Mascardi *

* University of Genova, Department of Informatics, Bioengineering, Robotics and Systems Engineering, Genova, Italy

Table of Contents

Session 1: Human-agent interaction and emotions

Session 2: Symbolic knowledge injection and extraction

Session 3: Trust and autonomy in agent interactions

Session 4: Agent-Based Modeling and Simulation

Session 5: Emotionality and rationality

Session 6: Applications and projects

Session 7: Risk mitigation, robustness and security

2022-11-03: submitted by Angelo Ferrando, metadata incl. bibliographic data published under Creative Commons CC0
2022-11-03: published on CEUR Workshop Proceedings (CEUR-WS.org, ISSN 1613-0073) |valid HTML5|

Can agents talk about what they are doing? A proposal with Jason and speech acts

Valeria Seidita[a], Francesco Lanza[a], Angelo Maria Pio Sabella[a] and Antonio Chella[a,b]

[a]Dipartimento di Ingegneria, Universitá degli Studi di Palermo, Italy
[b]ICAR-CNR National Research Council, Palermo, Italy

Abstract

The dream of building robots and artificial agents that are more and more capable of thinking and acting like humans is growing by the day. Various models and architectures aim to mimic human behavior. In our current research, we propose a solution to make actions and thought cycles of agents explainable by introducing inner speech into a multi-agent system. The reasons that led us to use inner speech as a self-modeling engine raised the question of what inner speech is and how it affects cognitive systems. In this proposal, we used speech act to enable a coalition of agents to exhibit inner speech capabilities to explain their behavior, but also to guide and reinforce the creation of an inner model triggered by the decision-making process through actions applied to the surrounding world and to themselves. The BDI agent paradigm is used to keep the agents rational and with the innate ability to act in a human-like manner. The proposed solution continues the research path that began with the definition of a cognitive model and architecture for human-robot teaming interaction, and aims to integrate the believable interaction paradigm into it.

Keywords

Human-Agent Interaction, Transparency, Jason, Inner Speech

1. Introduction

"What does a person think about before he takes an action?" — This is the question we must ask ourselves in order to reproduce in an agent the decision-making abilities inherent in man. Making profitable decisions is a very complex process that depends on many factors, especially when dealing with collaborative tasks and a highly variable environment. Getting an agent to make a decision in a fruitful way involves a number of challenges.

With collaborative tasks, we refer to human-agent (or human-machine, or even human-robot) interaction. In a collaborative task, humans and agents cooperate and work together to achieve a common goal. The goal is achieved through communication and interaction between agents, agents and humans, and agents and the environment. Let us consider team of humans. The goal is shared, each team member knows the goal, a set of actions (or tasks, i.e., one or more plans) to achieve the goal, his or her own capabilities, probably those of others, and all the known elements of the environment. In complex interactions and tasks, often not everything is known

WOA 2022: 23rd Workshop From Objects to Agents, September 1–2, Genova, Italy
valeria.seidita@unipa.it (V. Seidita); francesco.lanza@unipa.it (F. Lanza); angelomariapio.sabella@community.unipa.it (A. M. P. Sabella); antonio.chella@unipa.it (A. Chella)
0000-0002-0601-6914 (V. Seidita); 0000-0003-4382-6366 (F. Lanza); 0000-0002-4325-759X (A. Chella)

about the environment; not everyone knows everything, but can share knowledge. In addition, each team member's interaction with the environment can cause a change in the environment that can affect another member's actions.

So, when a team member decides to perform an action, he checks if this action is within his capabilities and if no other team member has already performed it or is performing it. If so, he can, or not, perform the action or delegate another team member.

There are many elements that underlie the decision-making process, including knowledge of one's own capabilities, continuous replanning when a goal is not achieved, and interaction with peers to gather information to gain new knowledge. In addition, we must also consider factors, such as the trust one team member has in another, his or her mental states (the inclination or willingness to perform an action), and the knowledge derived from the other's ability to explain the reasons for his or her actions or outcomes.

The interaction between humans and agents raises issues that are still being explored from a design process perspective. If everything about a system is not known a priori, and it evolves and changes during its execution phase, then it is not possible to define the properties of the system at design time and to study the requirements for the system completely. Our idea is based on the creation of a cognitive model and the corresponding agent architecture, whose modules allow structuring the agent's decision-making process, taking into account its internal states. Thus, we have combined aspects of self-modeling and Theory of Mind to integrate the elements of interaction already mentioned. On the implementation level, on the other hand, we have harnessed the power of the BDI agent paradigm [1, 2, 3] and one of the best known and most widely used agent programming language, *Jason* [4, 5].

In the present work, we go a step further and make a very early step for extending what we have done so far to give the agent the ability to reason aloud, i.e., to externalize its reasoning and thus make its actions comprehensible to humans.

Our proposal is based on the recognition that humans are able to talk to themselves, either aloud or in a quiet voice, in order to improve their understanding of what they are doing, to self-regulate their behavior, and to increase their knowledge of what surrounds them by using the feedback that inner speech gives them. We propose to combine the concept of inner speech with the concept of speech act to obtain a tool that allows us to move from the cognitive model to the agent architecture and then to the actual implementation. Moreover, this idea provides the input to explore the possibility of modifying the selection functions of the reasoning cycle of the *Jason* interpreter[4]. We then propose a first validation through a simple example in which we have implemented inner speech through speech acts.

The rest of the paper is organized as follows: in section 2, we describe the work done so far in this context and then, in sections 3 and 4, we give an overview of the concepts of inner speech and speech acts with the aim of highlighting their characteristics and the motivation for using them; in section 5, we illustrate the model we propose; in section 6 the validation scenario of the model with its code is given; and finally, in section 7 some conclusions are drawn.

2. Decision Process in Human-Agent Interaction

Adaptivity, autonomy, and proactivity are the qualities or abilities that every human being must possess in order to be able to decide at any time what actions to take. In collaborative scenarios, humans or agents (or robots), must have the ability to make appropriate decision-making, select an action or decide whether to perform it by itself or delegate it to someone else.

In [6] we proposed an architecture for modeling a human-robot team system.

The proposed architecture uses the elements underlying human-agent interaction. It is based on the standard model of the mind [7], and considers the classical MAPE (Monitoring, Analyse, Process, Execute) process [8]. To this, we added the modules for representing autonomous and adaptive interactions. We considered that the inputs to the reasoning process are motivations. Motivations are the heart of the decision process and the elements that initiate it. They contains all the information and processes that represent the inner world of an agent, his beliefs, desires, intentions, knowledge, and skills, but also norms, rules, emotions, the degree of trust in the other, and everything else that can serve as input for action (see Figure 1). We consider both external (objects, resources, ...) and internal (state of minds, emotions, ...) environments.

From a theoretical point of view, it is possible to model situations such as the justification of one's actions or the anticipation of actions, important for decision making and replanning. Anticipation is the process that allows generating a "current situation" i.e., the state of the world, from a selected possible action. It allows imagining the outcome of an action and actually implementing it only if its outcome is consistent with the post-conditions of the selected goal. In [9, 10] we proposed a prototype for part of the described architecture and an implementation using the BDI paradigm and *Jason* language, respectively.

In the first work, we paid special attention to the module of anticipation and robot knowledge enhancement. Basically, we developed a tool to look into the robot's mind while it is performing its collaborative tasks. In the second work, we proposed a variation of the reasoning loop that lies below the *Jason* interpreter to incorporate general motivation into the language and its management in the same way that *Jason* does in managing beliefs.

Finally, in [11] we used the trust model developed by Falcone and Castelfranchi [12, 13] in conjunction with the practical reasoning theory to build a model for the robot's decision. The idea was to extend the deliberative process and the belief base representation to allow the robot to decompose a plan into a series of actions. Each action is directly linked to the knowledge useful for its execution and is computed by a function

$$J \leftarrow \mathit{justify}(\alpha_i, B_{\alpha_i}) \tag{1}$$

In this way, the robot creates and maintains a model of "itself" and can justify the results of its actions. Justification is a key outcome of the application of self-modeling capabilities, as well as a useful means of improving trust interactions. It takes place in the execution phase, at the beginning of the justification cycle, a *Jason* agent updates all its beliefs and intentions, determines its desires, and selects some of them that become intentions to which it assigns a plan (line 1 to line 8 in Figure 2). After that, the agent usually processes the stack of actions to decide which one to execute. At this point, we have added a new function

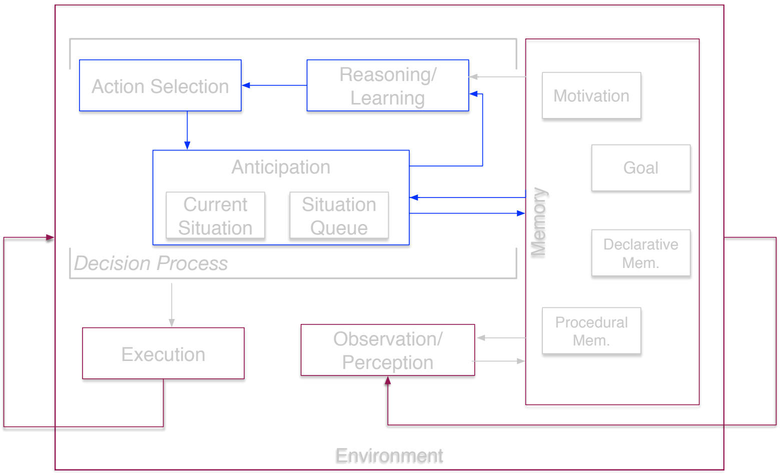


Figure 1: The abstract architecture for creating human-agent (robot) interaction based on human-human interaction in a team working together in the same rapidly changing environment to achieve a known and common goal.

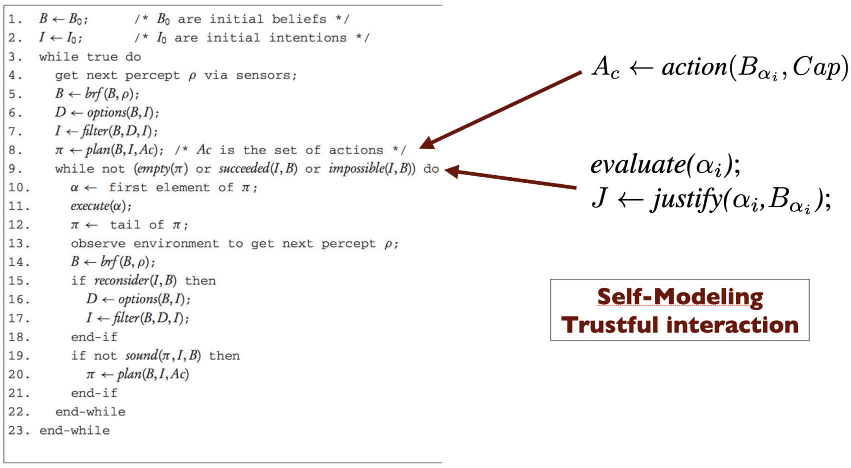


Figure 2: The algorithm for agent's practical reasoning and the extension for adding self-modeling and trustful interactions [4, 11].

$$A_c \leftarrow action(B_{\alpha_i}, Cap) \tag{2}$$

that allows us to associate part of the belief base with the capabilities of the robot, which in this way is able to justify its actions through the function $\mathcal{J}$.

The idea we are now proposing is to extend this even further and give the agent the ability to activate what is called inner speech, the ability to speak to oneself by making oneself the object of thought. The implementation is realized using the instrument of speech acts, its theory is perfectly consistent with the psychological theory of inner speech. To our knowledge, there is no approach in the literature, especially in the field of agents and robotics, that combines inner speech with speech acts both conceptually and in terms of implementation.

3. Inner Speech

The term inner speech has several connotations in the literature. It is a concept developed and used in psychology. It is often associated with the concept of self-awareness and self-consciousness [14, 15]. Self-awareness is "the ability to become the object of one's attention" [16] and takes into account three possible elements: the social environment, the physical world, and the self. Self-awareness also includes the ability to direct attention to one's mental state, i.e., perceptions, feelings, attitudes, intentions, emotions, etc. Morin was the first in [17] to link the concept of inner speech to the ability to think about oneself.

Inner speech is a concept used mainly in cognitive development and executive function theory. It is a way of reflecting on one's own experiences. People generally reflect on their own experiences in different ways [18]. Inner speech plays an important role in self-regulation of cognition and behavior, so it is used in people for both data collection and behavior regulation, and is also considered a motivational tool.

According to Morin, inner speech is the activity of silent self-talk. There are many other synonyms or equivalents, with some differences between adults and children, for example, self-talk, internal dialog, private speech or egocentric speech, or self-verbalization. The latter two activities generally refer to the behavior of children who comment on their own actions without caring whether they are understood or not. What interests us from a technical point of view is that inner speech serves self-direction, self-regulation, problem solving, planning, and memory. Another important point we want to keep in mind is that inner speech serves information retrieval. Or rather, self-information, as it is called by Morin and Everett. A person who needs to perform an action may talk to himself. This conversation serves him to identify data and processes.

During interactions, the use of inner speech or loud inner speech can be a way to make cognitive and decision-making processes transparent. And it is also a way to perform verbal mediation with oneself to support certain activities. Often, especially with children, inner speech is seen or used as an accompaniment or constant commentary on what they are doing. This is exactly what we want to do. In this paper, we take our cue from [19, 20] and propose to give an agent the ability to comment aloud on what it is doing.

4. Speech acts

The multi-agent paradigm provides a set of functionalities that enables agents to speak with other agents. Communication between agents changes the state of the world of every single agent that receives a new message. In general, this behavior may be a matter of the speech act theory proposed by Searle [21] and Austin[22].

Bordini et al. [23] described the speech act and how it works in an agent communication module. The basic principle of the speech act theory lies in the meaning of *language*. The principle can be summarized by assuming language as action. An artificial agent uses utterance to inform other agents about changes or to exchange novel information that concerns the surrounding world. The utterance produced by an agent changes, effectively, the state of the world in terms of beliefs, desires, and intentions owned by a hearer agent.

A multi-agent system supports basic speech act in its communication module through the adoption of agent communication languages [24], such as the *Knowledge Query and Manipulation Language* (KQML) [25, 26] or FIPA [27, 28, 29]. For instance, as denoted in [23], speech acts in *Jason* may be used through the usage of a set of internal action that let agents communicate one with the other.

Accordingly with Bordini [23], the BDI framework *Jason* is able to perform illocutionary speech act (what was done), also know as performative speech act. This is possible thanks to the set of internal actions owned by each agent.

In *Jason*, a basic illocutionary speech act may be send from a sender agent to a receiver using the *.send* internal action. Depending on what the agent wants to communicate, it can use different types of illocutionary speech act or performative. *Jason* provides a set of semantic performative to let an agent communicate. The *performatives*' name follow the KQML standard [30, 31]. Given the list of all possible illocutionary speech acts, then, the point is to exploit them in order to let the agent speak to itself, to show the capacity for inner speech. In this way, inner speech lets the agent's mental state change, as well as the agent's intention and behavior. Moreover, inner speech, by definition, makes the agent's decision-making process explainable.

5. Agents are able to explain themselves: the proposed approach

In this paper, we extend the concept of justification on the outcome of the action to the entire action, from selection to execution, and use the results of reasoning to update the agent's knowledge base and explain to the outside world what it is doing.

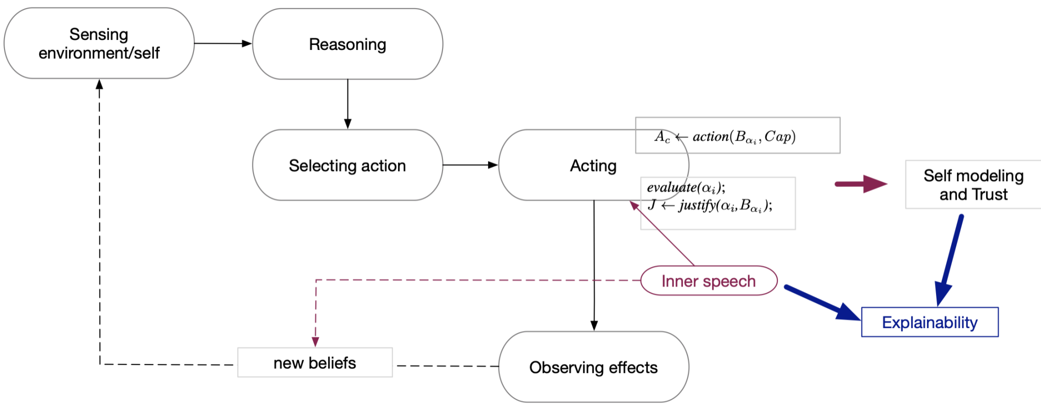


Figure 3: Key ideas for implementing inner speech in agents.

In Figure 3 we show our proposed behavioral view of the modules in the architecture shown the Figure 1. We did not include a start and end point because we want to represent the continuous reasoning cycle of an agent. Agents reasons by computing input from the environment, both external and internal, and select actions from the set provided by the designer. Once the action is selected, the two functions mentioned in section 2 are activated to create a rationale for the actions. Then, the agent observes the effects of its actions and the resulting changes in the external world, and updates its beliefs through perception. We added the inner speech module at the point where we use the $\mathcal{J}$ function. Before and after the execution of each action, the agent starts its inner speech and generates new beliefs, in addition to those generated by observing the effects on the environment.

From a technological point of view, the BDI agent paradigm and the *Jason* interpreter reasoning cycle are well suited for developing the parts we focused on in Figure 3. At runtime, a BDI agent consists of the Belief set, the Plan set, the Intention set, the Event set, and Action as well as the selection functions managed by the interpreter: S_E, S_O, and S_I [4].

Events can be external and internal. External events are generated by the perception of the environment and correspond to the deletion or addition of beliefs. Internal events are generated by the agent when executing a plan and they do not affect the environment. Thus, events are the ones most closely associated with the execution phase. Some examples are the addition of achievement or test goals.

Our idea is to handle internal events and link them with speech act to generate an explanation for agents' actions.

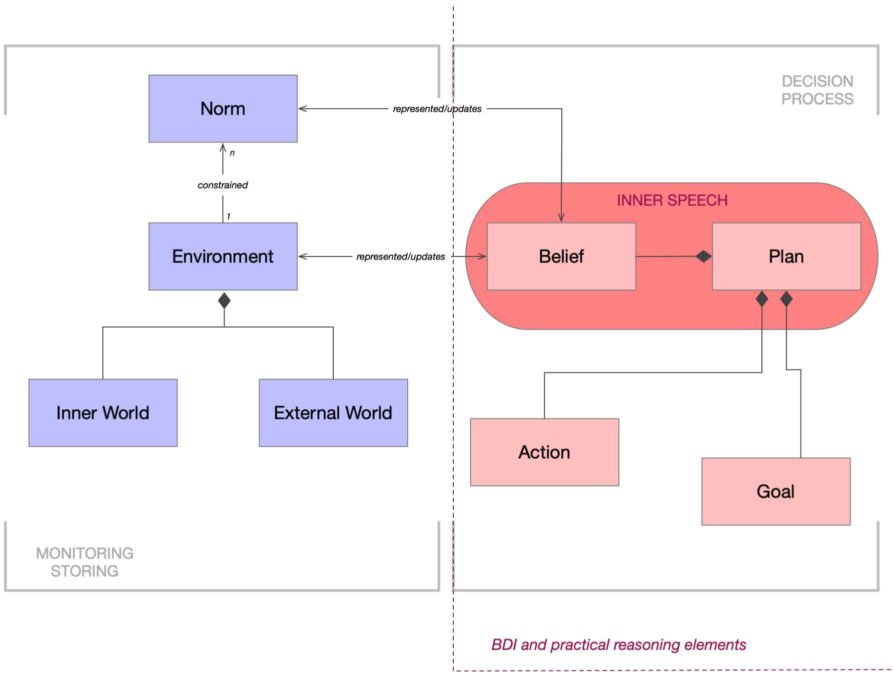


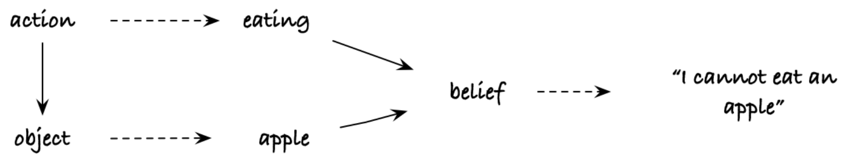
Figure 4: Design abstraction for human-agent interaction and inner speech.

The plan represents how the agent should act to achieve a certain goal, taking into account the belief, the action to be performed and the goal to be achieved. Therefore, plans are the elements that make inner discourse possible.

The idea we had from the theoretical point of view finds its natural implementation in speech acts. In an agent system, communication is based on the theory of speech acts [21], briefly described in the section 4. Since inner speech is a way of thinking about oneself, we claim that under certain conditions a speech act can address the agent itself. We assume that each agent in each plan, in addition to taking actions to achieve the team's goals, can send a message to itself. This message aims at changing its beliefs or goals, while making this message available to the outside world. In the work we propose in this paper, the communication with the outside world is done through the console; in the near future, a real dialogue between agent and human will be implemented.

The elements in Figure 4 show what is described above. The operating environment, whether internal or external, can be represented by the agent's beliefs. The environment is governed by norms, which are also represented by beliefs. Beliefs are the main elements that make up a plan, along with actions and goals. The left part of the figure shows the elements that allow us to instantiate the monitoring and storage modules of the architecture in Figure 1. On the right side, on the other hand, we can instantiate the reasoning module, which is based on the logical foundation of the BDI paradigm and practical reasoning. If we had used a paradigm different from BDI, these elements would look different, but the logical basis would not change. The advantage we have gained from our choice is that we can easily connect the theoretical model with its practical implementation by using the *Jason* interpreter's reasoning cycle.

Through beliefs, we can link actions to the environment and thus have or maintain knowledge about what actions can or cannot be performed and on what objects this can occur.



For example, consider to have an object *apple*, an action *eating*, and the relation between them realizes a norm of "not", then the agent can infer that it cannot eat the apple. Making the representation more complete, we can obtain the agent saying that it cannot eat the apple because it is an agent (or because it is not a human) or other more complex thought. It would depend only on how much complete is the agents' knowledge and its representation.

Concretely, then, we can change the practical reasoning cycle in Figure 2 by the following actions, which replace lines 10 and 11.

foreach α_i **do**
    evaluate(α_i);
    $R \leftarrow$ *rehearsal*(α_i,B_{α_i},D);
    $update(B,D)$;
    $J \leftarrow$ *justify*(α_i,B_{α_i});
end

Algorithm 1: The algorithm for extending the practical reasoning with the inner speech.

For each action, the agent evaluates the pre- and post-conditions, and then a function we call *rehearsal* is implemented to reason about actions, beliefs, and goals, i.e., desires in the case of

BDI agents. With the result of the reflections, the agent updates the knowledge base and then justifies its actions.

In the next section, we give a simple example with a speech act in a well-known scenario developed in the *Jason* online tutorial. This example is used to validate the theoretical approach and verify the use of speech acts for message return and external communication.

6. Implementing Inner Speech with a BDI Example

In the example, a collaborative task between agents is proposed, and no human is present. However, this does not affect the validation, since we want to check here the possibility that an agent activates its inner speech. The example was taken from the examples on *Jason*'s reference page[1].

The scenario, adapted for our purposes, involves collaboration and communication between four agents to complete a construction. The main purpose is to find resources of different types on the map, mine them and bring them to the agent in charge of the construction, called the *Builder*. The Builder agent needs three types of resources to achieve its goal, and works with three Collector agents to do so. Each Collector searches for a resource and can bring it along. Whenever a Collector agent finds a resource that the Builder needs, it picks it up and brings it to him. When a particular type of resource is no longer needed or when the constructions have finished, the Builder informs the collectors. The environment is represented through a simple grid representation and collectors are able to perceive whether or not a resource is present in a cell.

The Builder agent specifies the resource to search for, which exists in at most five per type. The Collector agents start searching for the first resource and stop when they have found five. In the original example, this control is centralized to the Builder agent and the Collectors do nothing but execute what is requested by the Builder. To validate the inner language module, we made some changes in this example and moved the resource control to the Collectors. In this way we can insert the *rehearsal* function at the moment the Collector agent finds a resource that is not of the type requested by the Builder or is no longer needed.

In the following algorithms we show an excerpt of the implemented code for the Collector agent.

The first portion of the code concerns the search for resources and four different cases may occur. In the first two cases, the agent keeps moving since it has not yet found any resources or has found a resource of a type not required. In the second case, the agent finds a resource that it believes is needed (because the collected quantity is not yet sufficient). The simplest case is seen in line 3, where the agent evaluates the triggering event and then changes its beliefs by a speech act directed to itself. The result of the execution can be seen in Figure 5, where the console contains a message that refers to that specific belief.

When the agent is supposed to continue with the search, it triggers the next plan (the last case) which involves extracting the found resource and transporting it to the builder. After

[1]http://jason.sourceforge.net/wp/examples/, this example was originally written by Rob Clarke and Andy Buck as 2nd coursework for the Multi-Agent Systems Module run in 2004-2005 at the University of Durham, U.K. , later edited by Rafael Bordini.

which it may continue with the extraction or the research.

+!check_for_resources : not found(_) $\leftarrow$ $move\_to(next\_cell)$.
+!check_for_resources : found(R) & enough(R) $\leftarrow$ $move\_to(next\_cell)$.
+!check_for_resources : found(R) & **not** resource_needed(R)
  $\leftarrow$ $.my\_name(Me)$;
    .send(Me, *tell*, resource_needed(R));
    !check_for_resources.
+!check_for_resources : found(R) & resource_needed(R)
  $\leftarrow$!$stop\_checking$;
    !take(R, builder);
    !continue_mine.

Algorithm 2: Part of the code that shows the plans used for searching resources.

1: +enough(R)[builder] : found(R) & resource_needed(R)
  $\leftarrow$ $.my\_name(Me)$;
    .send(Me, *untell*, [resource_needed(R), found(R)]);
    .drop_all_desires;
    !continue_mine.
2: +enough(R)[builder] : resource_needed(R)
  $\leftarrow$ $.my\_name(Me)$;
    .send(Me, *untell*, resource_needed(R)).

Algorithm 3: Part of the code that shows the plans that manage the receipt of notification by the builder of a resource type that is no longer needed.

The second part of the code concerns receiving a message from the Builder when it has collected enough of a particular resource. In this case, the agent must distinguish between two different circumstances. When it has found a resource of the type specified by the Builder, it performs internal reasoning to remove everything related to that resource from the belief base, except for the new knowledge. It may also abort the task it is performing to continue the search. Otherwise, it may be sufficient to update its own Belief Base.

In this example, we only show the existing mapping between the psychological concept of inner speech and the communication agent module through the use of the speech act. The example presents a complex structure including the environment in which the implemented methods can be interwoven with the *rehearsal* function. Next step, we will include the *rehearsal R* function for supporting the environment's artifacts and plan library revision.

7. Discussions and Conclusions

Interest in systems capable of self-adaptive and self-aware capabilities is growing rapidly in these years. Equipping robots or agents with cognitive capabilities is certainly the next breakthrough in the field of artificial intelligence, and more and more scientists are talking about machines capable of behaving like humans. In this paper we present a possible solution

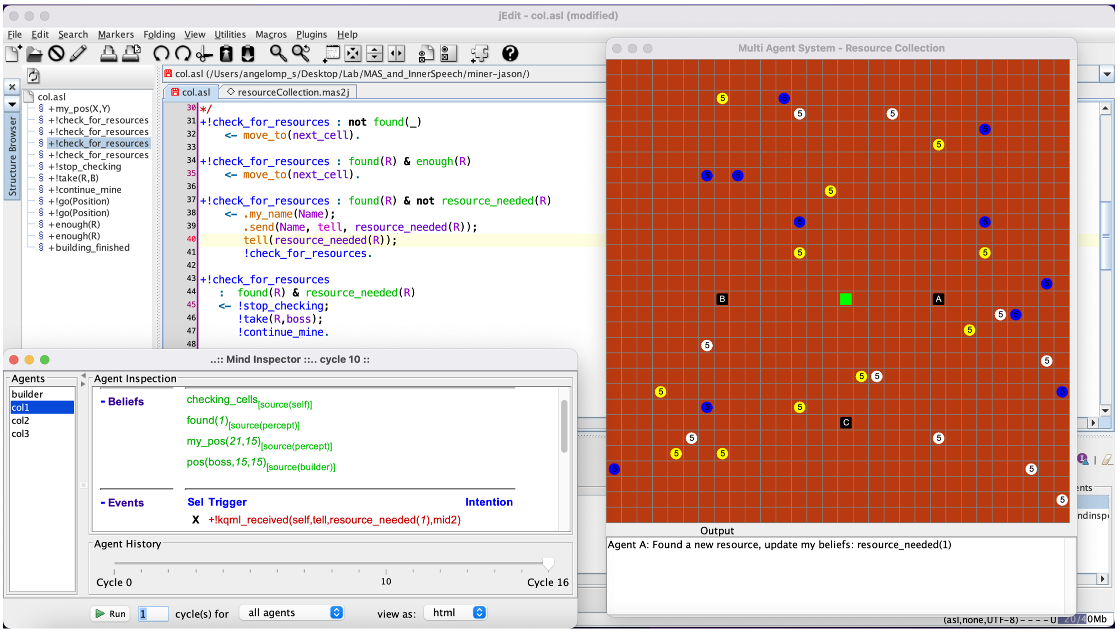


Figure 5: A screenshot of the plans running for the Collector agent.

to endow agents, robots and intelligent systems, with internal language capabilities. The deliberations a person makes before taking an action or making a decision are a key moment for adaptive and autonomous behavior, especially when working in teams. Humans have developed the ability to put themselves at the center of their thinking and to activate what is known as inner discourse to regulate and control their behavior. The idea we present in this article is a preliminary approach to use the concept of speech act to implement inner speech in agents. From a technological point of view, in moving from theory to implementation, we experimented with the agent's BDI technology and obtained good results in terms of simplicity of handling some specific design abstractions. We then proposed a validation of the theoretical approach through a simple example of a multi-agent system developed in *Jason*.

The proposed approach builds on and extends our previous work on endowing robots with the ability to justify the results of their actions. The ability to explain what is done, and why, is the focus of our work, which aims to create agents that are reliable, explainable, and believable. We believe that the method we use and the cognitive model that underlies all of our work can also incorporate other elements such as emotions, mental states, and even moral and ethical values into the reasoning process. This will be the topic of our future work.

The advantage we drew from the choices we made was that we could easily connect the theoretical model to its practical implementation by using the *Jason* interpreter's reasoning cycle. However, a challenge that we highlighted during our work, and that will be the subject of our future work, is that the transition from the cognitive model to the implemented model requires a careful and precise methodological approach to the design.

Another element to be considered is the completeness and accuracy of the agent's knowledge representation. Indeed, it is necessary that the elements of the environment are correctly associated with all the actions that can be performed and the rules for their activation. Ontologies, which can be found in the literature and for which a process of conversion to belief is necessary, can help in this regard, and their integration will be part of our future work.

Finally, we plan to further validate the proposed approach on a more complex scenario, e.g., the table setting, and complement it with a rigorous methodological approach.

Acknowledgments

The research leading to these results is in the frame of the project awarded by number FA9550-19-1-7025 - Air Force Office of Scientific Research.

References

[1] A. S. Rao, M. P. Georgeff, et al., BDI agents: from theory to practice., in: ICMAS, volume 95, 1995, pp. 312–319.

[2] A. S. Rao, Agentspeak (l): BDI agents speak out in a logical computable language, in: European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Springer, 1996, pp. 42–55.

[3] M. Georgeff, A. Rao, Rational software agents: from theory to practice, in: Agent technology, Springer, 1998, pp. 139–160.

[4] R. H. Bordini, J. F. Hübner, BDI agent programming in agentspeak using jason, in: International Workshop on Computational Logic in Multi-Agent Systems, Springer, 2005, pp. 143–164.

[5] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, A. Santi, Multi-agent oriented programming with jacamo, Science of Computer Programming 78 (2013) 747–761.

[6] A. Chella, F. Lanza, V. Seidita, A cognitive architecture for human-robot teaming interaction, in: Proceedings of the 6th International Workshop on Artificial Intelligence and Cognition, Palermo, 2018.

[7] J. E. Laird, C. Lebiere, P. S. Rosenbloom, A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics, Ai Magazine 38 (2017) 13–26.

[8] J. R. Anderson, The architecture of cognition, volume 5, Psychology Press, 1996.

[9] A. Chella, F. Lanza, V. Seidita, Human-agent interaction, the system level using jason, in: Proceedings of the 6th International Workshop on Engineering Multi-Agent Systems (EMAS 2018), Stockholm, 2018.

[10] V. Seidita, C. Diliberto, P. Zanardi, A. Chella, F. Lanza, Inside the robot's mind during human-robot interaction, in: 7th International Workshop on Artificial Intelligence and Cognition, AIC 2019, volume 2483, CEUR-WS, 2019, pp. 54–67.

[11] C. Castelfranchi, A. Chella, R. Falcone, F. Lanza, V. Seidita, Endowing robots with self-modeling abilities for trustful human-robot interactions, in: 20th Workshop" From Objects to Agents", WOA 2019, volume 2404, CEUR-WS, 2019, pp. 22–28.

[12] C. Castelfranchi, R. Falcone, Trust theory: A socio-cognitive and computational model, volume 18, John Wiley & Sons, 2010.

[13] R. Falcone, C. Castelfranchi, Trust dynamics: How trust is influenced by direct experiences and by trust itself, in: Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. Proceedings of the Third International Joint Conference on, IEEE, 2004, pp. 740–747.
[14] A. Morin, A neurocognitive and socioecological model of self-awareness, Genetic, social, and general psychology monographs 130 (2004) 197–224.
[15] P. J. Silvia, T. S. Duval, Objective self-awareness theory: Recent progress and enduring problems, Personality and social psychology review 5 (2001) 230–241.
[16] S. Duval, R. A. Wicklund, A theory of objective self awareness. (1972).
[17] A. Morin, Possible links between self-awareness and inner speech theoretical background, underlying mechanisms, and empirical evidence, Journal of Consciousness Studies 12 (2005) 115–134.
[18] L. E. Berk, R. A. Garvin, Development of private speech among low-income appalachian children., Developmental psychology 20 (1984) 271.
[19] A. E. Winsler, C. E. Fernyhough, I. E. Montero, Private speech, executive functioning, and the development of verbal self-regulation., Cambridge University Press, 2009.
[20] L. S. Vygotsky, M. Cole, Mind in society: Development of higher psychological processes, Harvard university press, 1978.
[21] J. R. Searle, J. R. Searle, Speech acts: An essay in the philosophy of language, volume 626, Cambridge university press, 1969.
[22] J. L. Austin, How to do things with words, Oxford university press, 1975.
[23] R. H. Bordini, J. F. Hübner, M. Wooldridge, Programming multi-agent systems in AgentSpeak using Jason, volume 8, John Wiley & Sons, 2007.
[24] G. K. Soon, C. K. On, P. Anthony, A. R. Hamdan, A review on agent communication language, Computational Science and Technology (2019) 481–491.
[25] T. Finin, R. Fritzson, D. McKay, R. McEntire, Kqml as an agent communication language, in: Proceedings of the third international conference on Information and knowledge management, 1994, pp. 456–463.
[26] T. Finin, R. Fritzson, D. P. McKay, R. McEntire, et al., Kqml-a language and protocol for knowledge and information exchange, in: 13th Int. Distributed Artificial Intelligence Workshop, 1994, pp. 93–103.
[27] P. D. O'Brien, R. C. Nicol, Fipa—towards a standard for software agents, BT Technology Journal 16 (1998) 51–59.
[28] M. T. Kone, A. Shimazu, T. Nakajima, The state of the art in agent communication languages, Knowledge and Information Systems 2 (2000) 259–284.
[29] F. Bellifemine, A. Poggi, G. Rimassa, Developing multi-agent systems with a fipa-compliant agent framework, Software: Practice and Experience 31 (2001) 103–128.
[30] Y. Labrou, T. Finin, A semantics approach for kqml—a general purpose communication language for software agents, in: Proceedings of the third international conference on Information and knowledge management, 1994, pp. 447–455.
[31] J. Mayfield, Y. Labrou, T. Finin, Evaluation of kqml as an agent communication language, in: International Workshop on Agent Theories, Architectures, and Languages, Springer, 1995, pp. 347–360.