Inner speech: a mechanism for self-coordinating decision making processes in robotics

A. Romero-Garcés[1], A. Hidalgo-Paniagua[2], P. Bustos[3], R. Marfil[1], and A. Bandera[1]

[1] Universidad de Málaga http://www.uma.es
{argarces, rebeca, ajbandera}@uma.es
[2] SCALIAN, home page: https://www.scalian.com
alejandro.hidalgo@scalian.com
[3] University of Extremadura http://www.uex.es
pbustos@uex.es

Abstract. The experience of inner speech is a common one for humans, playing a relevant role in generating spontaneous responses to the context but also in regulating how we think and behave. Intimately tied to our sense of self, the inner speech provides, via a mechanism of internalization, a running monologue of thoughts. This monologue plays a basic role in relevant aspects related, for instance, with our ability for structuring, regulating, and shaping our activities. In this paper, we emphasize this specific aspect of the inner speech and run some examples in the CORTEX software architecture, where semantic tokens (words and sentences) are employed not only for internalizing the world but also for regulating the decision making activities.

Keywords: Inner speech, cognitive architecture, inner world

1 Introduction

Different theories have formulated the inner language in people as a form of private speech, which we can summarize as the silent production of words in one's mind. In psychology, there is currently an open discussion about the precise nature of this inner speech [4, 8, 7]. For authors such as Gregory [8], the inner speech is reactive, i.e. it occurs as a spontaneous and uncontrolled response to the context the person finds herself in. It is then the responsible of the inner generation of more complex, elaborated perceptions. The production of these words or sentences are not under our control, and do not involve any effort. On the other hand, other authors, although agree with this proposal that some inner speech utterances are reactive, argue that not all of them are. Thus, in general, they recognize the importance of inner speech in the self-regulation of cognition and behaviour [7].

For this last school of thought, we usually engage in inner speech when we are deliberating about what to do in a specific scenario. This inner speech then plays a cognitive and self-regulatory role in the control of one's own behavior.

Briefly, as we deliberate, we are aware, and sentences, or at least words, in natural language come to our minds [7]. In this way, egocentric language, whose aim is to internalize and guide thought and behavior, helps us to elaborate a plan with respect to the activity we perform, and helps us to mental orientation and conscious understanding.

With the idea of implementing self-awareness in robots currently gaining relevance [15, 2], the aim of mimicking the concept of inner speech is particularly interesting. To achieve this goal, we must not to confuse inner speech with other inner, non-verbal experiences [3]. Any solution that will not be supported by symbols is not an inner speech instance. On the other hand, the topic is inline with the Language of Thought Hypothesis (LOTH) proposed by Fodor [6]. The LOTH proposes that thinking occurs in a mental language. This mental language (Mentalese) shares with spoken language to be organized in meaningful words that can be combined into sentences. The meaning of these sentence depends on the meanings of its component words and how they are combined. That is, simple concepts are combined in organized ways according to certain rules of grammar to create thoughts.

The goal of encoding information related to perceptions and actions as semantic tokens was one of the driving forces behind the design of the CORTEX cognitive robotic architecture [1, 13]. The central core of CORTEX is the Deep State Representation (DSR), a graph-based representation where all information coming from the agents in the architecture is annotated. A relevant part of the information in this DSR describes the context, but also the robot's intents and current activities, using semantics tokens. In this paper, we extend the CORTEX architecture to implement some kind of inner speech for self-organizing the robot's behaviour. Through words and sentences, our proposal is able to structure an inner dialogue, that allows to coordinate the activity of several decision makers.

The rest of the paper is organized as follows: Section 2 provides some details about related work, putting the focus on the relevance of inner speech as one of the mechanisms involved in self-regulation. Section 3 describes an instantiation of the robotics cognitive architecture CORTEX for the intralogistic robot and how the symbolic information in the DSR can be considered an internal monologue that allows the robot to self-organize its actuation in the outer world. Section 4 briefly introduces the real robot and scenario used for testing. Some key performance indicators obtained from our evaluation are provided. Finally, Section 5 draws some conclusions and introduces our future work.

2 Related work

Inner speech is an area of interest in artificial intelligence and machine consciousness [10]. In the earlier work by Steels [18], the process of reentering generated speech as speech understanding (re-entrant loop) was used for pushing language but also its underlying meaning towards greater complexity. Re-entrancy was then mainly proposed for checking the intelligibility of an utterance in their own

reception systems, being linked to the ability for generating complex grammars in natural language. In this work, the relevance of the inner speech is restricted to the area of natural language understanding or generation. Moreover, as Clowes [4] pointed out, having inner speech a conscious component, the construction of grammatical sentences is however usually considered an unconscious activity. On the other hand, Clowes [4] emphasizes the role of inner speech for organizing consciousness, regulating and shaping ongoing activities, and driving attention. This self-regulation model was evaluated in several experiments in which groups of agents had to execute different tasks [5].

Inner imagery and grounded inner speech appear as the relevant items in the 'consciousness test' by Haikonen [9]. Thus, a machine will be conscious if it has these phenomena without being pre-programmed, can describe their contents, and recognize them as its own results.

Chella [3] proposes a cognitive architecture for inner speech implementation in a Pepper robot, based on the Standard Model of Mind proposed by Laird et al. [12]. Considering their relevant ties, this proposal focuses on inner speech as a mechanism for reaching self-awareness in robotics. As in our work, they assume that the robot has linguistics abilities.

Large Language Models (LLMs) have demonstrated to be useful for dealing with problems in computational linguistics (speech recognition, natural language generation, machine translation...). But recently, they have also shown their ability for managing a rich internalized knowledge about the world [14], and even for answering questions that appear to require some degree of reasoning [11]. Considering these new capabilities of LMMs, Huang et al [10] extend these models for becoming an interactive problem solver and serving as reasoning models combining multiple sources of feedback. Thus, in the Inner Monologue (IM), the actuation of the robot is self-adapted considering the feedback from a language-based scene description and the one provided by a human user that the robot is cooperating with. As in our case, the actions to be executed are chosen from a set of pre-trained robotic skills. If these skills are mapped in the IM to textual descriptions that can be invoked by the language model, in our approach they are mapped to graph structures that can be detected by the agents in CORTEX. If the IM chains together perceptions, robotic skills and human feedback in a shared language prompt [10], our approach chains perceptions, robotic actions and human requests in the DSR.

3 Inner speech in the CORTEX architecture

Figure 1 shows an instantiation of the CORTEX architecture for the domain of robotic intra–logistics. This is the scenario that we will use for demonstration in Section 4. Briefly, the goal of the robot is to manage roll containers in the store, satisfying the petitions from a team of human pickers.

The central item in the architecture is the DSR. The DSR is a multi-labelled directed graph that holds symbolic and geometric information within the same structure. Figure 2 shows one simplified example. It works as a working memory.

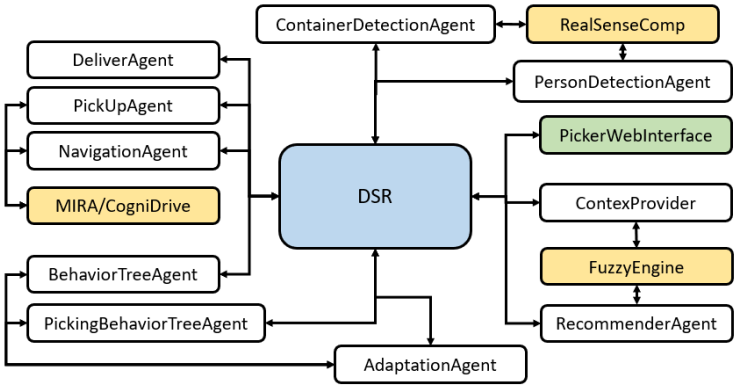


Fig. 1: Instantiation of the CORTEX architecture in the domain of robotic intralogistic.

The **robot** and the **roll container** are geometrical entities, linked to the **world** node (a specific anchor providing the origin of coordinates) by a rigid transformation. At the same time that we can compute the metric relationship between **robot** and **roll container** ($RT^{-1} \times RT''$), this **roll container** can be located close to the **robot**, and hence, the robot can launch the procedure for picking it up. In parallel, an agent can annotate that the **robot** is_not_detecting **people**. Features as the level of the **battery** are annotated as properties of the specific node linked to the **robot**. In this example, most of the nodes are present when we woke up the robot. Other ones, such as the **roll container**, are added by specific low-level perceptive modules. The verbs, encoded in the arcs (e.g. is_not_detecting) are updated by this same set of modules,

3.1 Perception and Action

In this scenario, the software architecture employs ten agents that are connected to the DSR. Robot localization and navigation is addressed by the MIRA/CogniDrive software from Metralabs. This stack is connected to the NavigationAgent, which is the responsible for updating all the needed information in the DSR. For instance, this agent monitors the battery status or the presence of close obstacles. The PersonDetection and the ContainerDetection agents use an Intel RealSense D435i RGBD camera for detecting and localizing people and roll containers in the environment. Some context items are fuzzified (e.g. this is the case for the **trolleypayload**, encoded into {EMPTY, LOW, MEDIUM, HIGH, FULL}). This enriches the dialogue (looking at the DSR, we can now for instance read that the **trolleypayload** is **FULL**).

More elaborated concepts related to non-functional properties such as safety are also generated and updated in the DSR at runtime. A fuzzy logic framework was designed for managing these high-level perceptions. The ContextProvider agent takes specific context information from the DSR and provides this to

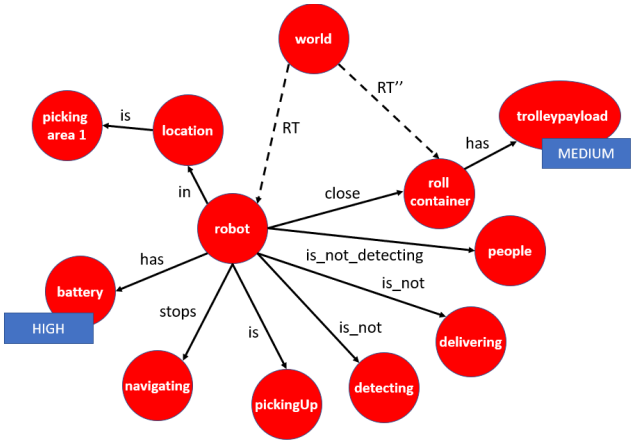
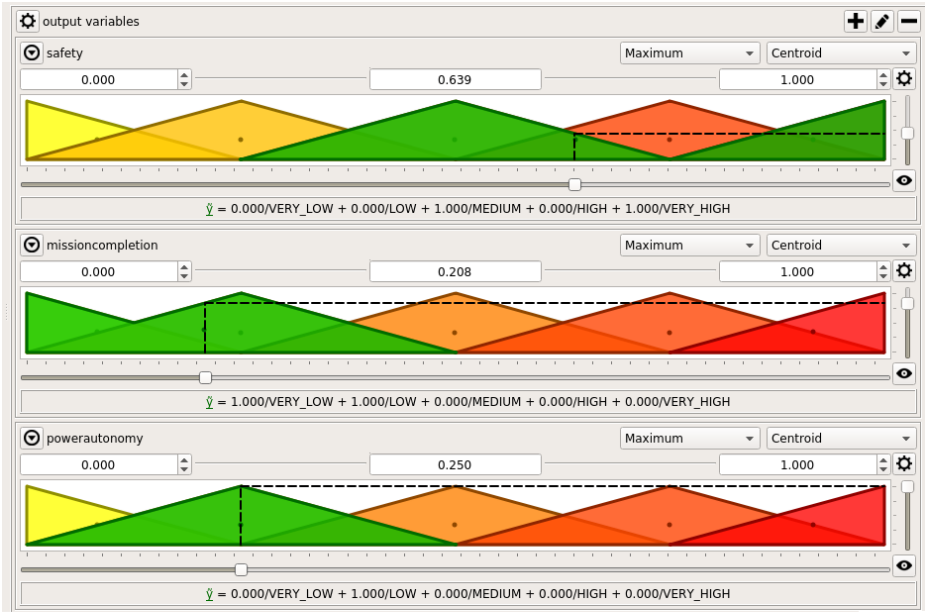


Fig. 2: Example of the DSR. Edges labelled as close or is_not denote logic predicates between nodes. On the other hand, edges starting at **world** and ending at **robot** and **roll container** are geometric and they encode a rigid transformation (RT and RT'' respectively) between them. Geometric transformations can be chained or inverted to compute changes in coordinate systems (see text).

the FuzzyEngine. These context items are fuzzified and employed in a fuzzy inference engine for estimating, in this case, metrics related with safety, mission_completion and power_autonomy (basically, they are the non-functional properties considered in this intralogistic scenario). Figure 3(Top) provides a snapshot of estimation of these perceptions. The robot can then internalize that the **safety** is **MEDIUM - VERY HIGH**, the **mission_completion** is **VERY LOW - LOW**, and the **power_autonomy** is **LOW**. This situation is the unconscious result of a context situation (see Figure 3(Bottom)). All this corpus of information, related with responses to changes on the context, constitutes in our approach the reactive voice of inner speech [7].

But the novelty here is to use this framework for involving inner speech in the 'episodes of deliberation' [7]. At deliberative level, the nominal course of action is encoded in the robot using two Behaviour Trees, designed using the BehaviourTree.CPP library by Davide Faconti[4]. In Figure 1, the modules in charge of executing these BTs are the BehaviourTreeAgent and the PickingBehaviourTreeAgent. Following the strategy described in [16], both BTs implement a nominal course of action (i.e. the set of actions to be sequentially executed when all is going fine) as its main branch, but are extended with the variability expressed in alternative branches that we define at design-time. Figure 4 shows the BT executed by the BehaviourTreeAgent. In the right side of the BT, under the Switch3 node, we have three possible robot's behaviour. The PickDeliverCharge is the nominal one: the robot moves to a picking area for picking up a roll con-

[4] https://github.com/BehaviorTree/BehaviorTree.CPP




```
1   if robotbump is PRESSED then safety is VERY_LOW
2   if peopleinroom is EMPTY then safety is HIGH
3   if peopleinroom is LOW then safety is HIGH
4   if peopleinroom is MEDIUM then safety is MEDIUM
5   if peopleinroom is HIGH then safety is LOW
6   if peopleinroom is FULL then safety is VERY_LOW
7   if trolleypayload is FULL then safety is VERY_LOW
8   if trolleypayload is HIGH then safety is LOW
9   if trolleypayload is MEDIUM then safety is MEDIUM
10  if trolleypayload is LOW then safety is HIGH
11  if trolleypayload is EMPTY then safety is VERY_HIGH
12  if objectdelivered is DELIVERED then missioncompletion is VERY_HIGH
13  if objectdelivered is not DELIVERED then missioncompletion is VERY_LOW
14  if objectdetected is IDENTIFIED then missioncompletion is HIGH
15  if objectdetected is not IDENTIFIED then missioncompletion is LOW
16  if batterylevel is VERY_LOW then missioncompletion is VERY_LOW
17  if batterylevel is LOW then missioncompletion is LOW
18  if batterylevel is MEDIUM then missioncompletion is MEDIUM
19  if batterylevel is HIGH then missioncompletion is HIGH
20  if batterylevel is VERY_HIGH then missioncompletion is HIGH
21  if objectpicked is PICKED then missioncompletion is VERY_HIGH
22  if objectpicked is not PICKED then missioncompletion is VERY_LOW
23  if goal is REACHED then missioncompletion is VERY_HIGH
24  if goal is not REACHED then missioncompletion is VERY_LOW
25  if batterylevel is VERY_LOW then powerautonomy is VERY_LOW
26  if batterylevel is LOW then powerautonomy is LOW
27  if batterylevel is MEDIUM then powerautonomy is MEDIUM
28  if batterylevel is HIGH then powerautonomy is VERY_HIGH
29  if batterylevel is VERY_HIGH then powerautonomy is VERY_HIGH
30  if trolleypayload is FULL then powerautonomy is VERY_LOW
31  if trolleypayload is HIGH then powerautonomy is LOW
```

Fig. 3: (Top) Evaluation of high-level perceptions using fuzzy logic: (safety, mission_completion and power_autonomy; and (Bottom) Rules set for determining the current high-level perceptions

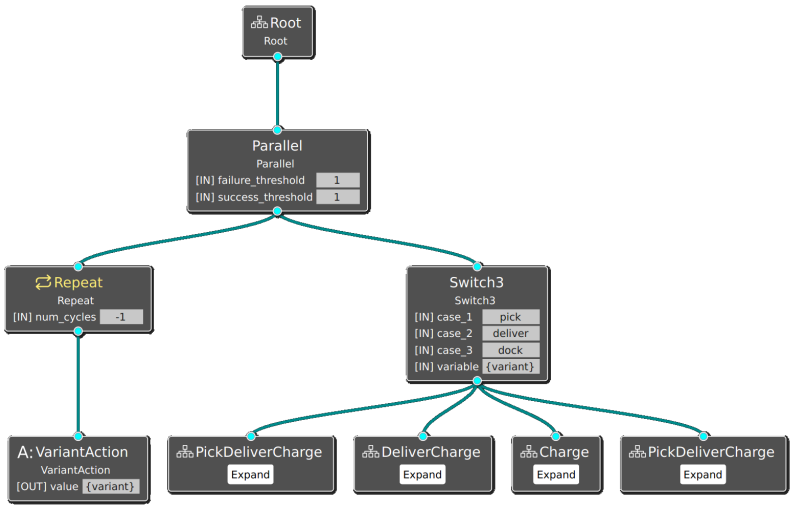


Fig. 4: Behaviour Tree encoding the behaviour of the robot at the higher level (see text)

tainer, delivering in a second area, and then returns to the charging area. The second branch forces the robot to deliver the roll container in the Delivery area (e.g. because the container is already full), and the third one commands the robot to abort the mission and goes to the charging area because battery level is very low. The branch to be chosen depends on an input command, the variant value, which is captured by the VariantAction node. Changing this value (pick, deliver, dock), we can modify the course of action at runtime.

The PickingBehaviourTreeAgent executes a BT that encodes the algorithm for allowing the robot to approach and pick up a roll container. There is also a nominal behaviour and alternatives, encoded as we have shown for the main BT in Figure 4. The commands emanated from both BT Executors, which will be simultaneously active, must be coordinated, and also correctly synchronized with the information reflecting the context dynamics. Being all the information about perceptions and actions refreshed in the DSR, this coordination is addressed by the inner dialogue that automatically emerges. We detail an example in Section 3.2. In this situation, the robot is deliberating and engage itself in a inner speech. Contrary to the reactive speech, the inner speech involving the BT Executors is performed for a reason (satisfying a mission) and it involves trying (and sometimes failing). It then meets the criteria demanding an event to be qualified as an action [8, 7].

Finally, Figure 1 also shows two additional agents. The PickUp and the Deliver agents are responsible of lifting up and down the roll container once it has been detected and localized.

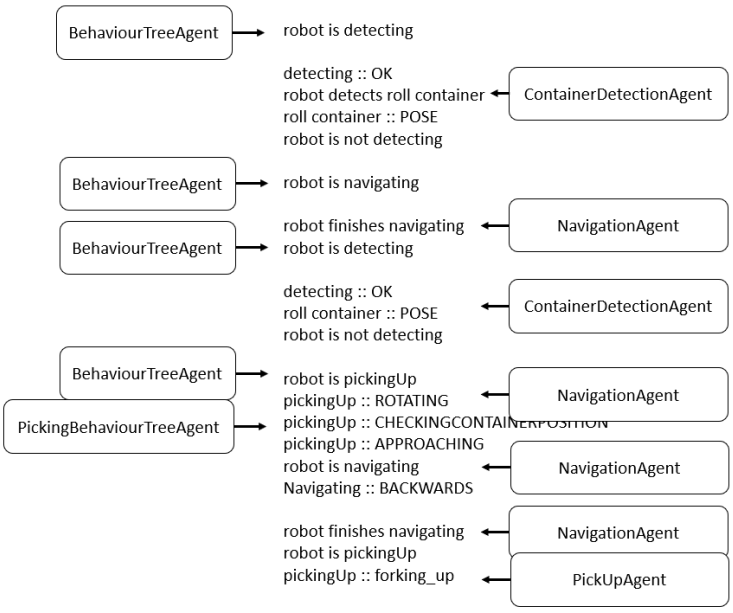


Fig. 5: Inner dialogue in the DSR.

3.2 Self-coordinating the robot's behaviour

The robot receives the command of collecting a roll container from a specific picking area. The BehaviourTreeAgent drives the robot to this area and then launches the detection procedure. Figure 5 illustrates a simplified view of the inner speech (emphasizing the relevant items for this case) in the DSR and the agents involved in annotating the information. The information coming from the BT Executors (e.g. requiring to detect the container) is complemented with the linguistic (e.g. adding a roll container to the DSR when this is detected) and geometrical (e.g. pose of the container) data provided in a reactive manner by the perception modules . When the BehaviourTreeAgent annotates that the **robot** is **pickingUp**, the PickingBehaviourTreeAgent knows that it must take the control of the robot.

The execution in Figure 5 is satisfying the requirements of a nominal use case. But it is relevant to note that the agents in the architecture are not asleep, waiting to be awakened by the deliberative modules. On the contrary, they are always running. Figure 6 shows a situation where people is detected close to the roll container. This information is annotated in the DSR by the PersonDetectionAgent. In this situation, The FuzzyEngine returns a low value for the safety, which is updated in the DSR by the RecommenderAgent. Taken into account this data, the AdaptationAgent decides that it is needed to change the course of action, and sends an indication (a variant value) for changing the executed branch to the BehaviourTreeAgent. A second fuzzy inference engine in the FuzzyEngine module quantifies what the best action to conduct is. The BehaviourTreeAgent should

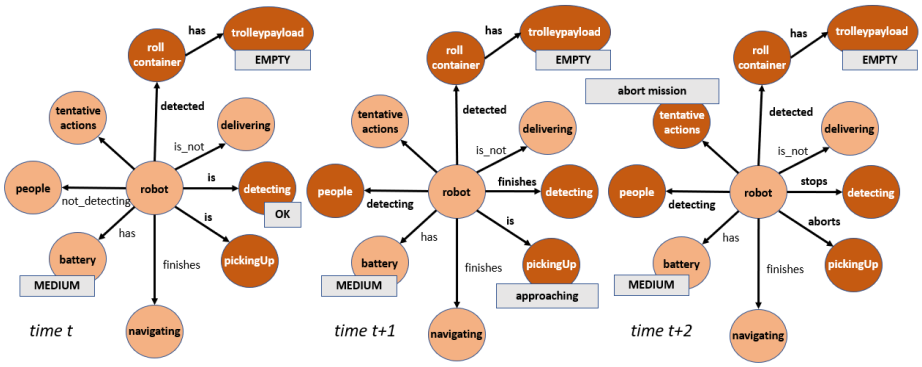


Fig. 6: The nominal execution is aborted because people is detected very close to the roll container. The figure shows a simplified view of the evolution of the DSR.

then stop the current action and executes the proposed new action, updating the DSR for maintaining the coordination among agents. All the inner dialogue narrating this 'episode of deliberation' can be traced by checking the linguistic terms annotated in the DSR. In Section 4, we document another example of self-adaptation.

4 Experimental evaluation

For correctly moving roll containers in the retail scenario, we have employed the CARY robot from MetraLabs GmbH. Each container can be moved by the robot from one picking position to another one if this is required by a human operator, and will be delivered if there is not a new petition or when the roll container is full. For detecting the roll container, the robot uses a camera placed in its frontside. However, for fine approaching, the robot uses a laser range finder placed in its backside (at the front of the fork, very close to the ground). For navigation, it uses the four laser range finders on the front and the lidar and camera placed in its frontside.

The robot was deployed in a real retail store (Eroski) sited in Casarabonela (Malaga, Spain) in February 2022. The shop is approximately 300 square metres in size, with narrow corridors (close to 2 metres). The deployment was easily conducted: we need to capture the map of the shop and then divide up the space into picking areas. In each of these areas, we set a pair of observation poses and a delivering pose. When a human picker asks for a roll container in a picking area, the robot will try to deliver it in the delivering pose. On the other hand, when the operator asks for a container to be removed, the robot will search for it in the picking area from the observation points. If it fails to search from the first point, it will try to search from the second. Figure 7 shows an example of

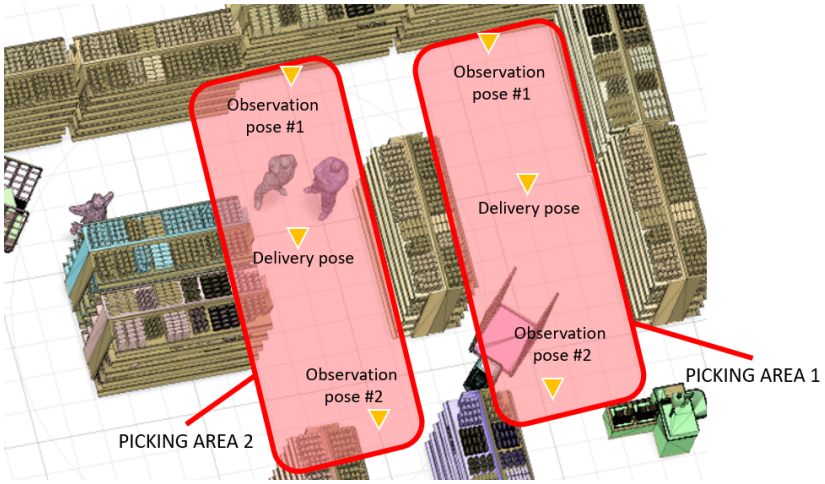


Fig. 7: Layout of the space: picking areas and relevant poses [17].

distribution of areas and relevant poses. Figure 8 provides some snapshots of the robot moving in the shop.

Next, we detail one mission. The robot is asked to move to one of the picking areas for picking up a roll container and deliver it to a second area. The robot arrives to the observation pose #1 and starts detecting the roll container. The container is detected and the robot pick it ups. The inner dialogue is the same that the one illustrated in Figure 5. Now, the BehaviourTreeAgent asks the robot to move the container to a second picking area. However, the robot detects that the roll container is loaded with its maximum payload. The inner dialogue is shown in Figure 9. It illustrates how the changes in the context are considered and they can modify the course of action. When the payload is FULL, the situation is considered unsafe by the robot, and the RecommenderAgent recommends to execute the default deliver action (move the roll container to the deliver area instead of delivering it to the next requested picking point). The AdaptationAgent sends a deliver value to the BehaviourTreeAgent, which aborts the current execution and annotates in the DSR that the robot should move to the default deliver area.

5 Conclusions and future work

The robotic software architecture CORTEX has been widely used in several projects, and has been endowed into robots working in real scenarios. The idea of annotating all the information coming from the agents in the architecture in a central graph-based representation has been a direct step towards real composability and has eased configurability. The information in this representation can be geometric or symbolic. And being this symbolic information the result

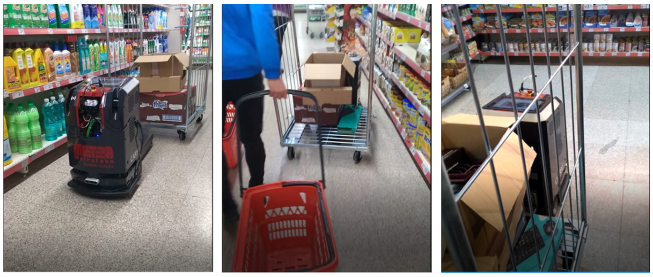


Fig. 8: The CARY robot moving a roll container through the Eroski store.

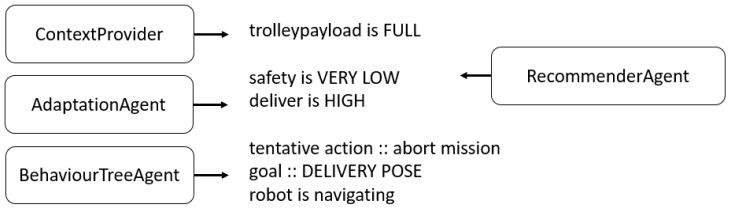


Fig. 9: The inner speech verbalizes that the context has changed and the system takes the decision of moving the roll container directly to the Delivery area.

of our natural manner of encoding a problem-solving procedure, it has been unconsciously structured using natural language. This can be traced back to our previous work in other robotic domains, where there is already a body of topics or ontology that has been maintained and updated over time. A process of internalization thus emerges, in which an instrument of thought has gone from being used by researchers to define how to interact with the environment or solve problems to determining how the robot organises itself internally to deal with these same issues.

This paper describes how to employ this inner speech for synchronizing the execution of deliberative modules. In our examples, the dialogue is simple and we could argue that the same result could be obtained by substituting the annotated words by specific commands or messages. However, all this inner speech is here available to the robot. A stream of sentences such as "the container is FULL" or "the battery level is LOW" can be obtained from the DSR. We are currently working on using this inner stream as a source of input data for a Natural Language Processing framework. This source can complement the external data coming from a speech recognition system, allowing the robot to use all the linguistic information in the DSR for augmenting the interaction with human users. Future work will also focus on testing more complex decision makers based on Automated Planning.

Acknowledgements

This work has been partially developed within SA3IR (funded by EU H2020 ESMERA Project, Grant Agreement 780265), the project RTI2018-099522-B-C4X, funded by the Gobierno de España and FEDER funds, and the B1-2021_26 project, funded by the University of Málaga.

References

1. Bustos, P., Manso, L., Bandera, A., Bandera, J., García-Varea, I., Martínez-Gómez, J.: The cortex cognitive robotics architecture: Use cases. Cognitive Systems Research 55, 107–123 (2019)
2. Chella, A., Cangelosi, A., Metta, G., Bringsjord, S.: Editorial: Consciousness in humanoid robots. Frontiers in Robotics and AI 6 (2019)
3. Chella, A., Pipitone, A., Morin, A., Racy, F.: Developing self-awareness in robots via inner speech. Frontiers in Robotics and AI 7 (2020)
4. Clowes, R.: A self-regulation model of inner speech and its role in the organisation of human conscious experience. Journal Consciousness Studies 14, 59–71 (2007)
5. Clowes, R., Morse, A.F.: Scaffolding cognition with words. In: Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems. pp. 101–105 (2005)
6. Fodor, J.A.: The Language of Thought. New York: Thomas Y. Crowell (1975)
7. Frankfort, T.: Action and reaction: the two voices of inner speech. Teorema 16(1), 51–69 (2022)
8. Gregory, D.: Are inner speech utterances actions? Teorema 39(3), 55–78 (2020)
9. Haikonen, P.: Robot brains: circuits and systems for conscious machines. John Wiley & Sons (2007)
10. Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K., Ichter, B.: Inner monologue: Embodied reasoning through planning with language models (2022), https://arxiv.org/abs/2207.05608
11. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners (2022), https://arxiv.org/abs/2205.11916
12. Laird, J.E., Lebiere, C., Rosenbloom, P.S.: A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. Ai Magazine 38(4), 13–26 (2017)
13. Marfil, R., Romero-Garces, A., Bandera, J.P., et al: Perceptions or actions? grounding how agents interact within a software architecture for cognitive robotics (2020)
14. Petroni, F., Rocktäschel, T., Lewis, P.S.H., Bakhtin, A., Wu, Y., Miller, A.H., Riedel, S.: Language models as knowledge bases? CoRR abs/1909.01066 (2019)
15. Reggia, J.A.: The rise of machine consciousness: Studying consciousness with computational models. Neural Networks 44, 112–131 (2013)
16. Romero-Garcés, A., Freitas, R.S.D., Marfil, R., et al.: Qos metrics-in-the-loop for endowing runtime self-adaptation to robotic software architectures. Multimed Tools Appl 81, 3603–3628 (2022)
17. Romero-Garcés, A., Hidalgo-Paniagua, A., González-García, M., Bandera, A.: On managing knowledge for mape-k loops in self-adaptive robotics using a graph-based runtime model. Applied Sciences 12(17) (2022)
18. Steels, L., et al.: Language re-entrance and the 'inner voice'. Journal of Consciousness Studies 10(4-5), 173–185 (2003)