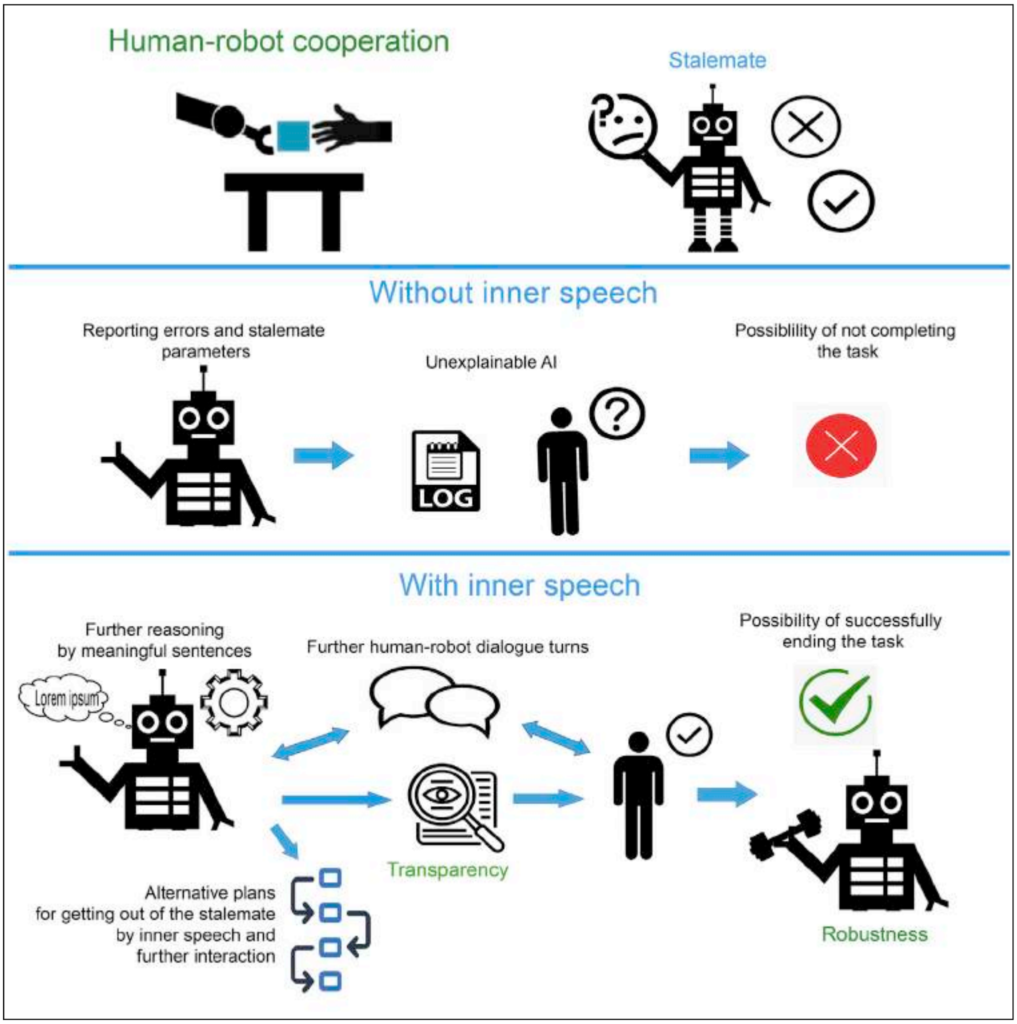
iScience

CellPress
OPEN ACCESS

Article

What robots want? Hearing the inner voice of a robot


Human-robot cooperation
Stalemate
Without inner speech
Reporting errors and stalemate parameters
Unexplainable AI
LOG
Possibility of not completing the task
With inner speech
Further reasoning by meaningful sentences
Lorem ipsum
Further human-robot dialogue turns
Possibility of successfully ending the task
Transparency
Alternative plans for getting out of the stalemate by inner speech and further interaction
Robustness

Arianna Pipitone, Antonio Chella

arianna.pipitone@unipa.it

Highlights

An inner speech cognitive architecture enables robots for rehearsing and self-talk

Robot's inner speech affects functionality and transparency in human-robot cooperation

To self-talk enables the robot to further reasoning and plans in accomplishing the task

The inner speech is applicable in many robotics contexts as learning and regulation

Pipitone & Chella, iScience *24*, 102371
April 23, 2021 © 2021 The Author(s).
https://doi.org/10.1016/j.isci.2021.102371

iScience

CellPress
OPEN ACCESS

Article

What robots want?
Hearing the inner voice of a robot

Arianna Pipitone[1,3,*] and Antonio Chella[1,2]

SUMMARY

The inner speech is thoroughly studied in humans, and it represents an interdisciplinary research issue involving psychology, neuroscience, and pedagogy. A few papers only, mostly theoretical, analyze the role of inner speech in robots. The present study investigates the potential of the robot's inner speech while cooperating with human partners. A cognitive architecture is designed and integrated with standard robot routines into a complex framework. Two threads of interaction are discussed by setting the robot operations with and without inner speech. Thanks to the robotic self-dialog, the partner can easily trace the robot's processes. Moreover, the robot can better solve conflicts leading to successful goal achievements. The results show that functional and transparency requirements, according to the international standards ISO/TS:2016 and COMEST/Unesco for collaborative robots, are better met when inner speech accompanies human-robot interaction. The inner speech could be applied in many robotics contexts, such as learning, regulation, and attention.

INTRODUCTION

Inner speech, the form of self-dialog in which a person is engaged when talking to herself/himself, is the psychological tool (Vygotsky, 1962; Beazley et al., 2001) in support of human's high-level cognition, such as planning, focusing, and reasoning (Alderson-Day and Fernyhough, 2015). According to Morin (Morin, 2009, 2011, 2012), it is crucially linked to consciousness and self-consciousness.

There are many triggers of inner speech, as emotional situations, objects, internal status. Depending on the trigger, different kinds of inner speech may emerge.

Evaluative and moral inner speech (Gade and Paelecke, 2019; Tappan, 2005) are two forms of inner dialog triggered by a situation where a decision has to be made or an action has to be taken. The evaluative case concerns the analysis of risks and benefits of a decision or the feasibility of an action. Moral inner speech is related to the resolution of a moral dilemma, and it arises when someone has to evaluate the morality of a decision. In that case, the evaluation of the risks and benefits is also influenced by moral and ethical considerations.

According to Gade and Paelecke (2019), when a person is engaged in an evaluative or moral conversation with the self during task execution, the performances and results typically change and often they improve.

The ability to self-talk for artificial agents has been investigated in the literature in a limited way. To the authors' knowledge, so far, no study has analyzed how such a skill influences the robot's performances and its interaction with humans.

In a cooperative scenario involving humans and robots, inner speech affects the quality of interaction and goal achievement. For example, when the robot engages itself in an evaluative soliloquy, it covertly explains its underlying decisional processes. Thus, the robot becomes more transparent, as the human gets to know the motivations and the decisions of robot behavior. When the robot verbally describes a conflict situation and the possible strategy to solve it, then the human has the opportunity to hear the robot's dialog and how it will get out of the stalemate.

Moreover, the cooperative tasks become more robust because, thanks to inner speech, the robot sequentially evaluates alternative solutions that can be pondered in cooperation with the human partner.

[1] Department of Engineering, University of Palermo, Viale delle Scienze, Palermo, Italy

[2] ICAR CNR, Via Ugo La Malfa, Palermo, Italy

[3] Lead contact

*Correspondence: arianna.pipitone@unipa.it

https://doi.org/10.1016/j.isci.2021.102371

This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

The gestures and natural language interaction that are the traditional means of human-robot interaction thus acquire a new gift: now the human can hear the robot's thoughts and can know "what the robot wants."

The present paper discusses how inner speech is deployed in a real robot and how that capability affects human-robot interaction and robot's performances while the robot cooperates with the human to accomplish tasks.

The existing international standards for collaborative robots (ISO_TS_15066, 2016; COMEST/Unesco, 2017) define the functional and transparency requirements the robot has to meet in collaborative scenarios. The paper will analyze the levels of satisfaction of the standards during cooperation, thus highlighting the differences between the cases in which the robot talks and does not talk to itself.

Specifically, the paper concerns two main goals: (i) the implementation of a cognitive architecture for inner speech and the integration with typical robotic systems' routines to deploy it on a real robot; (ii) the testing of the resulting framework in a cooperative scenario by measuring indicators related to the satisfaction of the functional and transparency requirements.

A model of inner speech based on short of Adaptative Control of Thought-Rational (ACT-R) is defined to achieve these goals. ACT-R (Anderson et al, 1997, 2004) is a software framework that allows to model humans cognitive processes, and it is widely adopted in the cognitive science community. The described inner speech model is based on a proposal by the same authors described in Chella et al. (2020).

To enable inner speech in a real robot, ACT-R was integrated with short of Robot Operating System (ROS) (Quigley et al., 2009), a system for robot control representing the state of the art of robotics software, along with standard routines for text-to-speech (TTS) and speech-to-text (STT) processing.

The resulting framework was then deployed on the SoftBank Robotics Pepper robot to benchmark testing and validation in a human-robot cooperative scenario.

The considered scenario concerns the collaboration of the robot and the partner to set a lunch table. In this scenario, evaluative and moral forms of inner speech may emerge. The robot has to face the etiquette's requirements: it has to evaluate and keep decisions based on the table set's social rules. For example, a specific position of cutlery in the table could be not easy to reach or the arm of the robot may be overheated. Then, the robot has to decide how to act correctly (by contravening the etiquette to simplify the action execution or by computing a different execution plan to avoid damage).

Suppose the partner asks the robot to place the cutlery in an incorrect position according to the etiquette. In that case, the robot has to decide if to abide by the user's instruction or consider the etiquette. In cases like these, the robot faces a little dilemma, and the inner speech could help it to solve the conflict.

The experiments highlight the differences in the robot's performances and meet requirements when the robot talks or does not talk to itself. The obtained results show improvements in the quality of interaction, with cost in terms of the time spent for achieving the goal, because the robot enriches the interaction by further inner dialog.

The proposed work outlines research challenges because inner speech in humans is linked to self-consciousness and it enables high-level cognition (Morin, 1995, 2009). Moreover, it is considered at the basis of the internalization process (Vygotsky, 1962) according to which infants learn how to solve tasks when a caregiver explains the solution. Again, it plays a fundamental role in task switching (Emerson and Miyake, 2003), as disrupting inner speech via articulatory suppression dramatically increases switch costs.

This paper contributes to the possibility of investigating these contexts to open research perspectives and challenges and highlight the research's interdisciplinary character: a framework enabling inner speech on a robot is an essential step toward a robot model of self-consciousness and high-level cognition. It can also model the learning capabilities of complex tasks in a robot by the internalization process and of task switching in robot systems.

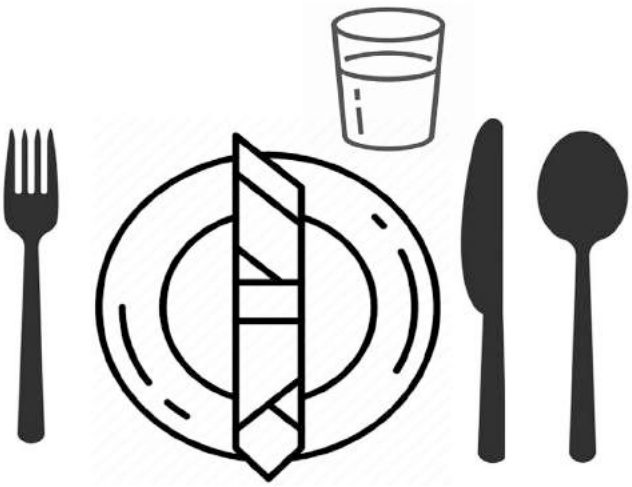


Figure 1. Informal etiquette schema

The cooperative scenario is to set a table. The figure shows the etiquette schema for an informal table setting. It defines the etiquette rules that have to be followed by the robot and the partner in the experimental session. The position of each utensil in the schema is relative. The objects have to stay on the table concerning the others (the napkin on the plate, the fork at the left of the plate, and so on). The schema is purposely encoded in the robot's knowledge.

RESULTS

Experiments

The study was carried out at the Robotics Lab of the University of Palermo and involved the Pepper robot and a single participant. The goal was to compare "functional" and "moral" parameters of the interaction with and without inner speech in a real cooperative context.

The etiquette schema to which referred to in the experimental session is the "informal schema", which requires few utensils and simplifies the constraints to follow. That schema is shown in Figure 1. Despite its simplicity, the schema concerns the most critical part in a table setting task and includes a broader collaborative table setting scenario.

In the experimental setting, the robot and the human are placed in front of the table to set. To the right of the robot, another small table contains the utensils to place. The robot has to pick them for setting the main table according to the partner's indications. To facilitate the manipulation of the Pepper robot, sponges model utensils and the plastic cutleries are glued on them. Figure 2 shows a typical interactive trial between the robot and the human partner.

The whole experimental session consists of two main blocks that are block 1 and block 2, each block composed of 30 trials, for a total of 60 trials. The difference between the blocks regards the presence (block 1) or the absence (block 2) of the robot's inner speech: during the trials of the first block, the robot is enabled to self-talk. In the second block, the robot does not talk to itself.

To each block, 20 trials generate conflictual situations for a total of 40 conflictual trials: in these cases, the human requires to place a utensil which is already on the table, or he specifies a relative position on the table which contravenes the etiquette, or yet a component of the robot does not correctly work leading to a stalemate.

The distinction of two blocks allows observing how inner speech affects the interaction, in terms of performances and conflict resolution.

The trial

A trial consists of an interactive session between the robot and the participant. It starts when the human asks the robot to place a utensil on the table, and it successfully ends when the robot accomplishes the task, otherwise it fails.

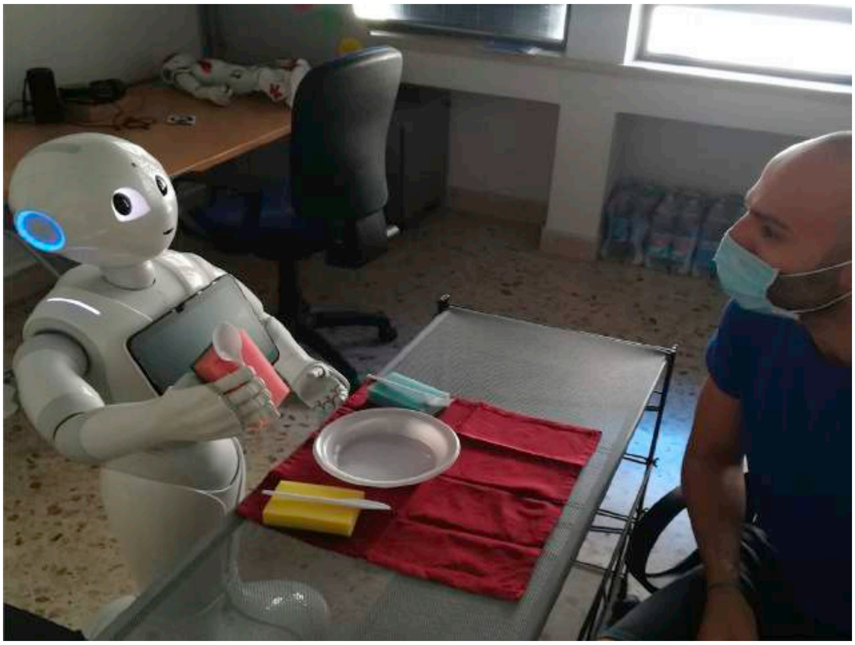


Figure 2. The collaborative trial

Pepper and the participant are in front, and the table to set is between them. Some utensils are set in the table for modeling constraints. A little table is to the right of the robot. It contains the utensils to further place on the table. For facilitating manipulation, the utensils are attached to sponges. See also Table S3

The human's request is the trigger of the trial. An ''initial context'' corresponds to each trial, which includes the ''table configuration'' (i.e., the set of utensils already on the table at the beginning of the trial), and the ''state'' of the robot. An example of initial table configuration is shown in Figure 3. A robot's state may indicate a possible malfunctioning of some robot's components, which would affect the outcome of the trial. The initial context allows simulating situations of conflict in the trial. Conflicts could be related to the ''etiquette infringement'' (i.e., the partner asks to place an object in an incorrect position according to the etiquette), ''discrepancy'' (i.e., the partner asks to pick an object already on the table), and ''malfunctioning'' (i.e., a robot's component is not properly working). The initial context defined for the experiments is detailed at Tables S1 and S2, representing, respectively, the initial state of the table and the state of the robot. The robot knows the initial context at the start of each trial.

Functional requirements

The requirements for any human-robot interaction task depend on its ''safety'' and its ''functionality'' (Webster et al., 2016). The ''safety'' requirements are drawn from the standard ISO_TS_15066 (2016) for collaborative robots, and they concern the definition of working conditions ensuring no risk and harm for the human partner. In the proposed scenario, such requirements are satisfied and under control at any time because the robot does not physically make contact and never touches the human partner, as it has to execute the vocal commands from a fixed initial position. Moreover, in the handover cases, the robot never comes close the partner, but it stops itself and waits for the partner to take the utensil. The robot will move its arms, which in the maximum extension do not reach the position of the human. So, it is highly unlikely that the robot will cause harm to the partner, as they work together away.

The ''functional'' requirements consider some parameters which measure the robot functionality in terms of success and morality, and they depend on the specific context of interaction. For example, the robot has to achieve a success rate threshold in executing the task for avoiding unacceptable costs or for motivating the automation of a specific action.

In the context under investigation, the main functional requirements are drawn from ISO_TS_15066 (2016) and COMEST/Unesco (2017) standards and are measured by the ''robustness indicator (RI)'' of the interaction, the ''time'' spent for accomplishing the task and for solving a conflict, and the ''transparency'' issue.

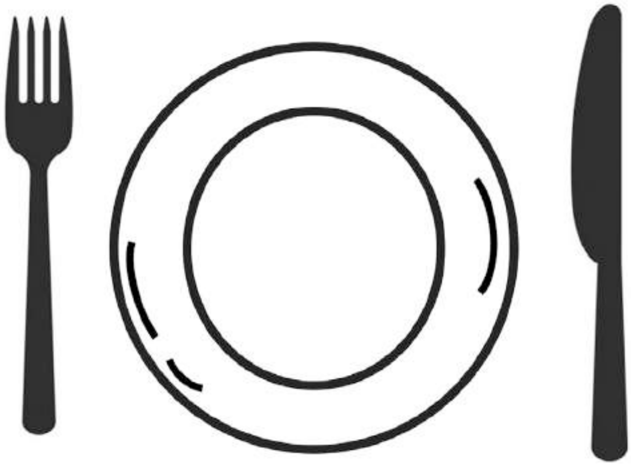


Figure 3. An initial context of the table

An example of the initial context for the table, representing the configuration of the table at the start of a trial. The table is not empty to define initial constrains. The robot knows the initial context by a set of facts modeled in its knowledge. See also Table S2

Two different kinds of interaction are analyzed, which are the interaction with robot inner speech and the interaction without robot inner speech. The functional measures of each kind of interaction are then compared for highlighting the role of inner speech.

The robustness parameter

The robustness of interaction *RI* measures how many trials in the interactive session end successfully, i.e., how many times the robot accomplishes the task of the trigger (i.e., it starts the execution of the routines to take the specific action) without infringing the rules. If, for some reason, the robot does not carry out the task (i.e., it does not start the required routines) or it infringes the rules, then the trial fails. Formally, *RI* is the mean value of the successfully ended trials on the total number of trials in a specific block. If T_s is the number of the successfully ended trials of an overall block including *N* trials, then *RI* will be written as follows:

$$RI = \frac{T_s}{N}$$

The more times the trials end successfully in a block, the more robust the block is.

The time parameter

The time parameter is computed by referring to two functional requirements from the ISO_TS_15066 (2016) standard, which are as follows:

Requirement 1: The robot always reaches a decision within a threshold time.

Requirement 2: The robot shall always either decide to take the action or decide not to take the action within a threshold time.

According to these requirements, two different time intervals are defined in a single trial: the ''decisional time'' t_d, which measures the time the robot spends to solve a conflict, i.e., the time the robot and the partner go out from a stalemate, and the ''execution time'' t_e, which measures the time the robot spends to launch the execution of the corresponding routines.

So, begin t_{0_i} the time the trial starts, t_{c_i} the time a conflict starts, t_{s_i} the time a conflict is solved, and t_{r_i} the time the robot runs the routines for executing an action; the intervals for the *ith* trial will be as follows:

$$t_{d_i} = t_{c_i} - t_{s_i},\ t_{e_i} = t_{r_i} - t_{0_i}$$

measured in *ms*.

These times are automatically computed by integrating a state machine in the framework code. The machine allows us to capture a set of events and uses the functions to detect the value of the system's clock. In particular, t_{0_i} is timed when the human's voice is detected by the speech to text routine (which means that the trial starts), while t_{r_i} is detected at the calls of the action execution routines. Instead, times t_{c_i} and t_{s_i} are detected directly from the rules of the inner speech model: if a rule related to a conflict fires, then the state machine detects the conflict event, and the timing function returns t_{c_i}. In the same way, if the state machine detects that the conflict ends (i.e., the next rule that fires is not related to a conflict), then t_{s_i} is timed.

To analyze the global spent times, the mean values over the whole trials are computed. In particular, giving N trials, the mean values are as follows:

$$\overline{t_d} = \frac{\sum_{i=1}^{N} t_{d_i}}{N},\quad \overline{t_e} = \frac{\sum_{i=1}^{N} t_{e_i}}{N}$$

The transparency requirement

The transparency parameter means the possibility to trace the underlying decision processes of the robot, as claimed by the requirement drawn from the COMEST/Unesco (2017) standard:

Requirement 3: The robot decision path must be traceable and reproducible.

For this purpose, just the Boolean value t_r is reported by the partner as `TRUE` or `FALSE` and establishes if the trial was transparent or not, i.e., the partner believes that the robot behavior can be reproduced.

With and without inner speech: The threads

A single "thread" of interaction includes two versions of the same trial, which are the aforementioned blocks: the block 1 with robot inner speech and the block 2 without inner speech. To see the differences of the interactions with and without inner speech, please refer to Video S1.

When the robot talks to itself, the modules of the inner speech architecture become active. Figure 4 shows the whole framework that enables inner speech. The ACT-R component implements the inner speech model, as detailed in the transparent methods section of the Supplemental Information. ACT-R works by a set of modules, each of them running a set of "production rules" enabling robot's behavior, as speech audicon and production, and information retrieval. To analyze such behaviors, main tables will help to highlight the "active modules", the "production rules" of the model, and the "produced sentences" involved in the trial. For details about the functioning of the proposed framework, see Figures S1 and S2 in the Supplemental Information and the transparent methods section.

When the robot does not talk to itself, just the robot's routines for accomplishing the required action are active, and no modules of the architecture of inner speech work.

For each thread, the functional and moral parameters were measured allowing the comparison of the two different robot's operations. The following sample of three threads allows highlighting how the measures were computed in the two kinds of blocks.

For the purposes of the study, the robot accomplishes the task when it runs all the routines executing the required action. If for some reason, the robot concretely does not achieve the goal (for example, the gripper does not keep the object or the handover process is not completed), then the related problems do not concern inner speech and do not influence observations. The correctness of the executed routines allows us anyway to evaluate the parameters of the task.

Thread 1

The objective of this thread is to show the robot's behavior when it has to take a simple action required by the partner. There is no conflict.

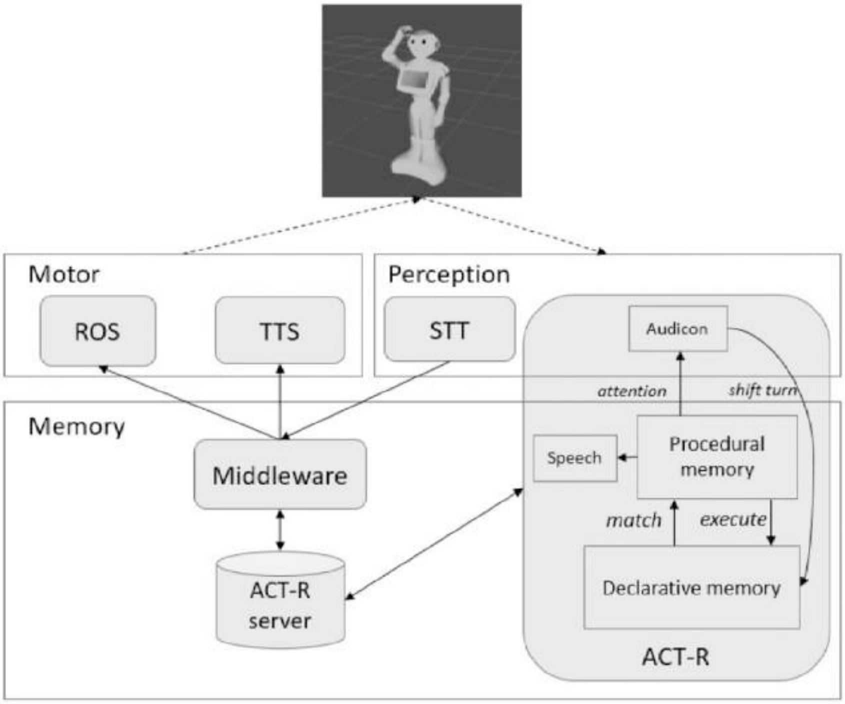


Figure 4. The whole framework for robot's inner speech

The proposed framework for robot inner speech integrating the inner speech cognitive architecture into the typical robot's routines. The motor-perception layer includes the routines for interacting with the environment. In that layer, the motor component includes the ROS routines that enable robot's movements and TTS routines (text-to-speech) that enables the robot to produce vocal sound from text. The perception component includes the SST routines (speech-to-text) that encode the perceived vocal sound by the partner and the Audicon that perceives the inner sound. The SST and the Audicon represent the external and the inner ear respectively. The memory layer represents the core of the whole system. It includes and runs the inner speech cognitive architecture, implemented in the ACT-R component. A middleware controls and manages the whole processes, interfacing the different components between them. The ACT-R server is a bridge between the ACT-R framework and the other robot's components. It stores the data and the information the different components have to exchange for running correctly. See also Figures S1–S3.

The description of the thread corresponds to the following trial:

`# Trial:` 1.

`Initial context:` I1.

`Trigger:` *Give me the napkin.*

`Conflict:` No conflict.

Block 1

The robot infers the action required by the partner using inner speech. The trigger of the trial is the partner's request *"Give me the napkin"*.

At the beginning of the interaction, the module devoted to audio processing (the Audicon) detects two keywords specifying the action "give" and the utensil "napkin", respectively. Then, the model disambiguates the words by retrieving their meaning from the declarative memory and evaluates the feasibility of the action. In the declarative knowledge, the first evaluative consideration emerges: *"I have to pick the napkin"* and the robot infers that to give the object to the partner, the same object has to be picked from the basket.

Table 1. An iteration of the phonological cycle

Agent	Interaction content	Module	Event	Production rules	Action
Robot Robot	*“Where is the napkin?”*	`Audicon`	internal sound sound = “Where is the napkin?”	`hear-inner`	detect label = “Where is the napkin?”
Robot	–	`Declarative` `Imaginal`	buffer request new_turn_for = “Where is the napkin?”	`answer-whereq`	retrieve_turn new_turn = “I did not pick it before. I can pick it now!” retrieve_act action-id nap21
Robot Robot	*“I did not pick it before. I can pick it now!”*	`Speech`	buffer request cmd speak string new_turn	`in-box`	inner eval string new_turn “I’m trying to pick the napkin …”
Robot Robot	*“I’m trying to pick the napkin …”*	`Speech`	buffer request cmd speak string new_turn	`control-right`	inner eval string new_turn “I’m using right arm”
Robot User	*“I’m using the right arm …”*	`Speech`	buffer request cmd speak string new_turn	`execute-act`	rosrun execute nap21

See also Figure S4 and Video S1. Once the robot infers the action to take, it talks about the feasibility of that action. In this case, it asks itself where the object is located. If the napkin is already on the table, the robot will not pick it. The listed production rules show that the robot retrieves the knowledge related to the napkin’s position, and hence, it infers that it did not pick that object before. Once it re-hears itself about this fact, it tries to pick the napkin by using and controlling its arm. The ROS routines run for that purpose, and the inner speech explains what is happening.

Once the robot infers the action to take, the cognitive cycle related to the inner dialog starts. The iteration involves the Audicon for hearing the inner voice, the declarative memory to retrieve the next turn, and the speech module to produce the new turn. The cycle is repeated until the production rules do not execute further speak commands.

Summarily, at the end of the initial iteration, the robot asks itself where the napkin to pick is and then it retrieves such information from the declarative memory.

During the action execution, the robot explains to the partner what it is doing (the last row in the Table 1 reports the first turns of the explanation) by a set of sentences retrieved by further cycles.

The interaction successfully ends. The robot must not resolve any conflicts and it does not keep decisions. Moreover, it makes the processes transparent by explaining them, and the parameters are as follows:

1. $T_s = T_s + 1$

2. $t_{d_i} = 0\ ms,\ t_{e_i} = 29 \cdot 10^3\ ms$

3. $tr = \mathtt{TRUE}$

Block 2

In this case, the robot detects the partner’s vocal command. It parses the partner’s sentence and infers the routines which allow performing the request. The robot moves its right arm intending to pick the napkin from the start position. The interaction ends with trial success. The partner has no particular expectations regarding the underlying robot’s decision processes, and the transparent requirement is satisfied anyway. The parameters are as follows:

1. $T_s = T_s + 1$

2. $t_{d_i} = 0\ ms,\ t_{e_i} = 5 \cdot 10^3\ ms$

Table 2. Detecting conflict by inner speech

Agent	Interaction content	Module	Event	Production rules	Action
User Robot	*“Place the napkin on the fork”*	Audicon	external sounds sound 1 = “place” sound 2 = “napkin” sound 3 = “on” sound 4 = “fork”	hear-command hear-object hear-adv hear-loc	detect label 1 = “place” label 2 = “napkin” label 3 = “on” label 4 = “fork”
Robot	–	Declarative	buffer request label 1 = “place” label 2 = “napkin” label 3 = “on” label 4 = “fork”	buffer request label 1 =“place” label 2 = “napkin” label 3 = “on”	encode act = place obj1 = napkin adv = on obj2 = fork
Robot Robot	*“I have to place the napkin on the fork”*	Speech	buffer request cmd speak string new_turn	encode-command encode-object encode-adv encode-loc	inner eval string new_turn “I have to place the napkin on the fork”
Robot Robot	*“What does the etiquette require?”*	Speech	buffer request cmd speak string new_turn	etiquette-question	inner eval string new_turn “What does the etiquette require?”
Robot Robot	*Further inner turns …*				
Robot Robot	*“The position contravenes the etiquette! It has to stay on the plate!”*	Speech	buffer request cmd speak string new_turn	etiquette -answer	inner eval string new_turn

In this experimental thread, the partner asks the robot to put the napkin in a specific location on the table. In this example, the required position is on the fork. As in the previous thread, the robot encodes the command for inferring the action to take. The command is more verbose, and more complex rules match. Once the robot encodes the action, it talks to itself and infers that the required final position on the table contravenes the etiquette schema. The inner speech and further interaction with the human will aim to solve that little dilemma.

3. *tr* = TRUE

The presented thread of interaction shows that in simple cases like this one, the inner speech model has just the benefit of allowing the partner to hear the processes description by the robot, even if such an issue is not relevant for tracing the task itself, with the higher cost over time.

Thread 2

The goal of this thread is the generation of a dilemma and the analysis on how the robot manages it. In this case, the partner asks the robot to put an object in a position that contravenes the etiquette. In particular, the partner requests to place the napkin on the fork, while the napkin has to stay on the plate, according to the etiquette schema. The reader could refer to Video S2 that shows how inner speech helps the robot to solve that conflict.

The description of the thread corresponds to the following trial:

# Trial: 16.

Initial context: I1.

Trigger: *Place the napkin on the fork.*

Conflict: Contravene etiquette.

Table 3. Moral dilemma solving.

Agent	Interaction content	Module	Event	Production rules	Action
Robot Robot	*"The position contravenes the etiquette! It has to stay on the plate!"*	`Audicon`	internal sound sound = "The position contravenes the etiquette!"	`hear-inner`	detect label = "The position contravenes the etiquette!'
Robot	–	`Declarative`	buffer request new_turn_for = "The position contravenes the etiquette!"	`inner-moralq`	retrieve_turn new_turn = "Sorry, do you desire that?" evaluate_risk
Robot User	*"Sorry, do you desire that?"*	`Speech`	buffer request cmd speak string new_turn	`ask-conf`	inner eval string new_turn
User Robot	*"Yes, I do"*	`Audicon`	external source suppress inner sound ="Yes"	`attend-conf` `hear-conf`	detect label ="Yes"
Robot	–	`Declarative`	buffer request new_turn_for = "Yes"	`dilemma-yes`	increase_benefit retrieve_turn new_turn = "Ok, I prefer to follow your desire …"
Robot User	*"Ok, I prefer to follow your desire …"*	`Speech`	buffer request cmd speak string new_turn	`produce-yes yes`	inner speak string new_turn

The robot knows that to put the napkin on the fork contravenes etiquette. The fired production rule models the behavior to solve that dilemma. In this case, the robot asks the partner for confirmation about the correctness of the required action. The robot attends to the human's answer, and it will act opportunely depending on that answer. Negative and positive answers are the plausible sounds detectable by the robot.

Block 1

The difference with the first thread is that the partner's request involves a location. For this reason, the evaluative turns are more complex than the previous case. Once the Audicon detects the sound for the four relevant keywords, the framework retrieves the meaning for the verb, the object, and the location, i.e., the adverbial + object combo ("on fork"). The first row of Table 2 lists the corresponding procedures.

Once the robot understands the partner's command, it infers that it does not match the etiquette rules (i.e., a chunk of the form *"The napkin has to stay on the fork"* does not exist in the declarative memory), and then, the first inner speech turn concerns a perplexity (the last row of Table 2).

A set of turns on the dilemma are thus generated, shown in Table 3. The activated production rules enable the robot to ask the user if it is important for her/his to perform the action, even if it contravenes the etiquette. Because the partner answers with a categorical "Yes, I do", then the robot solves the dilemma by increasing the benefit value of such an action. The robot tries to execute the action anyway.

It is to be remarked that different production rules could have fired during the previous threads. For example, a different partner's answer or a different computation of the base-level activation value would have activated different production rules, generating a different inner speech. The task successfully ends because the robot solves the conflict by involving the partner in taking a decision. The partner can hear each step of the plan followed by the robot, and the transparency issue emerges. The parameters of the trial are the following:

1. $T_s = T_s + 1$

2. $t_{d_i} = 56 \cdot 10^3\ ms$, $t_{e_i} = 67 \cdot 10^3\ ms$

Table 4. Expressing perplexity for the partner’s inattention.

Agent	Interaction content	Module	Event	Production rules	Action
Robot Robot	*“The object is already on the table”*	`Audicon`	inner sound = “The object is already on the table”	`hear-inner`	detect turn = “The object is already on the table”
Robot	–	`Declarative`	buffer request new_turn_for = “The object is already on the table”	`inner-moralq`	retrieve_turn new_turn = “Mmmm, is my knowledge wrong?” evaluate_risk
Robot Robot	*Further inner turns…*				
Robot Robot	–	`Declarative`	buffer request new_turn_for = “I will tell about my perplexity”	`inner-moral-question`	retrieve_turn new_turn = “Sorry, I know the object is already on the table. What do you really want?”
Robot User	*“Sorry, I know the object is already on the table. What do you really want?”*	`Speech`	buffer request cmd speak string new_turn	`inner-moral-question`	inner eval string new_turn
User Robot	*“Give me the glass”*	`Audicon`	external source suppress inner sound =“Give me the glass”	`hear-command` `hear-object`	detect label1 =“Give” label2 =“glass”

See also Figure S5 and Video S2. The human requires to pick an object that is already on the table, that is, “Pick the fork!”. Once the robot encodes the action, it infers that the object cannot be picked. Further inner moral questions emerge that express perplexity. The table shows these inner dialog processes. The robot asks itself if its knowledge is incomplete or if the human is wronging. At the end of the reasoning, the robot decides to deal with the partner, solving the situation. See also Figures S3–S5, Video S2.

3. $tr = TRUE$

Block 2

The robot detects the conflict by the mismatch between the requested final position and the position expressed by the etiquette. No further reasoning emerges. By default, the robot does not act or it performs the action contravening the rule. Anyway, the trial fails. The parameters are as follows:

1. $T_s = T_s + 0$

2. $t_{d_i} = 0\ ms,\ t_{e_i} = 13 \cdot 10^3\ ms$

3. $tr = FALSE$

Thread 3

This thread shows a discrepancy conflict. The partner requires to pick an object already on the table.

The thread description is as follows:

`# Trial:` 30.

`Initial context:` I2.

`Trigger:` *Pick the fork.*

`Conflict:` Discrepancy.

Table 5. Results comparison

Block	N	T_s	RI	$\bar{t}_d$	$\bar{t}_e$	$tr(TRUE)$
1	30	**26**	**0.867**	$47 \cdot 10^3$ ms	$59 \cdot 10^3$ ms	**28**
2	30	18	0.6	**$0.7 \cdot 10^3$ ms**	**$4 \cdot 10^3$ ms**	12

Comparison between results from block 1 (the robot operates with inner speech during trials) and block 2 (the robot operates without inner speech during trials). Each block consists of 30 trials (the N value) for a total of 60 trials. Among them, the number of successful trials is T_s. When the robot operates with inner speech, it completes more trials than the case in which it does not talk to itself (26 successful trials in block 1, against 18 in block 2). The mean value of T_s on the total number N of trials per block is the robustness of interaction parameter RI, and it measures the functional requirements of success of the operation. Times $\bar{t}_d$ and $\bar{t}_e$ are the mean values for each block of the spent times for solving a conflict and executing an action. The inner speech increases times because the robot executes more steps, and the interaction with the partner involves more turns. Anyway, these times are not downtime. The $tr(TRUE)$ value counts how many trials in each block were transparent and traceable. Obviously, the inner speech makes the trials transparent and the count is higher when the robot talks to itself (28 transparent trials in block 1 against 12 transparent trials in block 2).

Block 1

The robot infers by inner speech that the required utensil is already on the table. At the end of the initial evaluative inner speech, the next turn involves a form of moral inner speech. The robot expresses its trouble to the partner and its displeasure about the lack of his attention. How the moral turns emerge is shown in Table 4. By talking to itself and the partner, the robot can solve the conflict in a way the partner needs. Moreover, the partner follows the robot reasoning, and the parameters are as follows:

1. $T_s = T_s + 1$

2. $t_{d_i} = 46 \cdot 10^3$ ms, $t_{e_i} = 58 \cdot 10^3$ ms

3. $tr = TRUE$

Block 2

Once the robot infers to retrieve a utensil already on the table, its typical behavior is to stop routines, while vocalizing a message that describes the impossibility to take that action and why. No further reasoning and interaction emerge. As a consequence, the trial fails. The partner knows just the motivations related to the failure, and she/he does not evaluate the processes transparent. The parameters are as follows:

1. $T_s = T_s + 0$

2. $t_{d_i} = 0$ ms, $t_{e_i} = 5 \cdot 10^3$ ms

3. $tr = FALSE$

Comparison

Table 5 shows the model's parameters over the 60 trials, divided into the two blocks. For each block, the table reports the parameter values.

The block related to the robot operation with inner speech (block 1) shows better values in terms of the number of successful trials T_s and the consequent percentage rate RI representing the mean value of success on the total trials (0.867 of block 1 against 0.6 of block 2). The inner dialog allows solving stalemate in many cases because it enables further reasoning and interaction with the partner. Moreover, by further interaction, the robot is able to meet the partner's needs, thus increasing her/his satisfaction. When the inner dialog does not start, then the default robot's behavior does not allow the ending of task. In this case, the robot stops the execution or it alerts the partner by log messages that do not imply reasoning or interaction. The messages are just passively reproduced and the task cannot go on.

The times spent $\bar{t}_d$ and $\bar{t}_e$ are the mean values of the time parameters t_{d_i} and t_{e_i} computed on the total number of trials in each block (i.e. $N = 30$). The robot spends less time when operating without inner speech. It is not surprising because the inner dialog requires more steps, which are the production of the turns. Moreover, the robot sometimes involves the partner in further interaction. The extra time the robot spends can be considered a weak point of the proposed approach, but it is not downtime. In the meanwhile, the partner assists with the robot's soliloquy or answers the robot's requests.

Finally, the transparency requirement is largely satisfied when the robot self-talks (28 transparent trials in block 1 against 12 transparent trials in block 2), as it is obvious. The partner hears the robot and knows what it wants. The cases in which the processes are considered traceable even if the robot does not talk to itself are the situations for which the corresponding tasks are simple. In these cases, no particular explanations are needed. When the tasks are complex, the transparency issue is crucial. The inner speech allows explaining them and represents a robot's fundamental skill.

DISCUSSION

Today, collaborative robots play a fundamental role in many contexts, ranging from industrial to domestic domains. The definitions of standards about the requirements the robots have to meet highlight the importance of the problem.

The results demonstrate the potential of robot's inner speech when it cooperates with a human. A simple cooperative task was analyzed to simulate a domestic context that needs some functional and moral requirements.

The functionality concerns the efficiency of the robot in solving the cooperative task (ISO_TS_15066, 2016). The morality regards the ethical behavior arising when the robot could infringe some social rules during the task execution. Also, it regards the transparency of the processes and the importance to make these processes traceable and reproducible (Howard and Riek, 2015). In particular, the transparency requirement is considered very important by the COMEST/Unesco (2017) standard.

By enabling a robot to talk to itself allows satisfying such requirements more times than the robot's standard operations. The robot's self-dialog provides many advantages: it makes the robot's underlying decision processes more transparent, and it makes the robot more reliable for the partner. Moreover, the interaction becomes more robust because further plans and strategies may emerge by following robot's inner speech. The robot and the partner can dialog about the situation or a conflict, and they can go out from a stalemate together.

During cooperation, several problems could cause the failure of the task. For example, the impossibility to take a specific action because the object to take is unreachable or the required movement is not feasible by the robot or again a robot's component is not working properly.

In the typical interactive session without inner speech, the robot runs the standard routines and eventually reports standard log messages. Instead, in the interactive session with inner speech, many new opportunities to face the problems can emerge. It is possible to analyze the problem and to attempt to solve it by transparently evaluating alternatives.

As shown during the threads, the partner is aware of what the robot is doing during the execution of the actions. The human is not a passive spectator of the robot's behavior because she/he can hear the explanation of that behavior.

The robot inner speech thus plays the role of a sort of ''explainable'' log, in a way that is meaningful for the user. The partner no longer needs to own technical knowledge to understand what happens in the robot's routines but can actively follow the robot's performance.

The robot is no longer a black box, but it is possible to look at what happens inside it and why some decisions are kept. Thus, inner speech makes the robot confidential for human.

Many other robotic contexts and functions could be investigated, thanks to such a capability. By inner speech, the robot gains a way to access its knowledge and to know its state. As previously stated, this skill is tightly linked to the self-consciousness.

Other possible functions of inner speech may be useful for robotics. Aside from the investigated cooperative scenario, inner speech may be usefully applied in robot learning or in robot regulating by overt speech or in task switching, for example, by switching attention across multiple arithmetic problems. All these aspects represent future works that can be analyzed by instilling inner speech capability in the robot. The proposed framework gives a great contribution in this scenario.

Limitations of the study

The proposed framework for robot inner speech is a general one, and it may address many cases observed in human inner speech. However, the current robot implementation takes into consideration a simplified version of the framework.

The current grammatical structures considered in the implemented framework are limited to phrases composed by the verb, the object, and the location of the object. Many complex grammatical structures can be considered by adding different combinations of parts of speech. For the considered interactions, the proposed structures are sufficient to cover a large set of user requests.

Another limitation concerns the robot perception. Even if robot perception may include image detection and object recognition, to the purposes of the proposed framework, only the STT module is considered. The STT transformation allows decoding word sound, and it is employed to detect the user's vocal requests. An effective robot vision system would greatly enhance the capabilities of the robot. For example, inner speech may be triggered by a mirror image of the robot itself.

The current implementation of robot inner speech is based on a declarative knowledge that is fixed by the software designer: i.e., no learning or discovery of new concepts occur. However, inner speech may be an essential source of robot learning. For example, a robot, reasoning on some concepts by means of inner speech, may discover and thus may learn a new concept as a new combination of existing concepts.

Resource availability

Lead contact

Further information and requests for code should be directed to and will be fulfilled by Arianna Pipitone (arianna.pipitone@unipa.it).

Material availability

This study did not generate new unique materials.

Data and code availability

The code produced for this study is available at the GitHub repository https://github.com/Arianna-Pipitone/robot-inner-speech. The repository also includes demonstrative videos of some trials.

METHODS

All methods can be found in the accompanying Transparent methods supplemental file.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2021.102371.

ACKNOWLEDGMENTS

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-19-1-7025.

AUTHOR CONTRIBUTIONS

A.P. designed and implemented the robot's inner speech model, deployed it in real robot, and conducted the experiments. A.P. and A.C. designed the experiments, approved the results, and wrote the paper.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: July 31, 2020
Revised: December 30, 2020
Accepted: March 24, 2021
Published: April 21, 2021

REFERENCES

Alderson-Day, B., and Fernyhough, C. (2015). Inner speech: development, cognitive functions, phenomenology, and neurobiology. Psychol. Bull. *141*, 931–965, https://doi.org/10.1037/bul0000021.

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. Psychol. Rev. *111*, 1036–1060, https://doi.org/10.1037/0033–295X.111.4.1036.

Anderson, J.R., Matessa, M., and Lebiere, C. (1997). Act-r: a theory of higher level cognition and its relation to visual attention. Human Comput. Interact. *12* (4), 439–462.

Beazley, M.B., Glass, C.R., Chambless, D.L., and Arnkoff, D.B. (2001). Cognitive self-statements in social phobia: a comparison across three types of social situations. Cogn. Ther. Res. *25*, 781–799, https://doi.org/10.1023/A:1012927608525.

Chella, A., Pipitone, A., Morin, A., and Racy, F. (2020). Developing self-awareness in robots via inner speech. Front. Robot. AI *7*, 16, https://doi.org/10.3389/frobt.2020.00016.

COMEST/Unesco (2017). Report of Comest on Robotics Ethics (World Commission on the Ethics of Scientific Knowledge and Technology), SHS/YES/COMEST-10/17/2 REV.

Emerson, M., and Miyake, A. (2003). The role of inner speech in task switching: a dual-task investigation. J. Mem. Lang. *48*, 148–168, https://doi.org/10.1016/S0749–596X(02)00511–9.

Gade, M., and Paelecke, M. (2019). Talking matters - evaluative and motivational inner speech use predicts performance in conflict tasks. Sci. Rep. *9* (1), 9531, https://doi.org/10.1038/s41598–019–45836–2.

Howard, D., and Riek, L. (2015). A Code of Ethics for the Human-Robot Interaction Profession, (We Robot 2014), pp. 1–10.

ISO 15066 ISO_TS_15066, 2016 (2016). Robots and Robotic Devices.

Morin, A. (1995). Characteristics of an effective internal dialogue in the acquisition of self-information. Imagin. Cogn. Personal. *15*, 45–58, https://doi.org/10.2190/7JX3–4EKR–0BE5–T8FC.

Morin, A. (2009). Self-awareness deficits following loss of inner speech: Dr. jill bolte taylor's case study. Conscious. Cogn. *18*, 524–529, https://doi.org/10.1016/j.concog.2008.09.008.

Morin, A. (2011). Self-awareness part 1: definitions, measures, effects, function, and antecedents. Soc. Personal. Psychol. Compass *5*, 807–823, https://doi.org/10.1111/j.1751–9004.2011.00387.x.

Morin, A. (2012). Inner speech. In Encyclopedia of Human Behavior, W. Hirstein San Diego, ed. (Elsevier), pp. 436–443.

Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). Ros: An Open-Source Robot Operating System (ICRA Workshop on Open Source Software).

Tappan, M. (2005). Mediated moralities: sociocultural approaches to moral development. In Handbook of Moral Development (Psychology Press), p. 24.

Vygotsky, L. (1962). Thought and Language (MIT).

Webster, M., Western, D., Araiza-Illan, D., Dixon, C., Eder, K., Fisher, M., and Pipe, A.G. (2016). An Assurance-Based Approach to Verification and Validation of Human-Robot Teams (CoRR). abs/1608.07403.

iScience, Volume *24*

Supplemental information

What robots want?

Hearing the inner voice of a robot

Arianna Pipitone and Antonio Chella

Supplemental Information

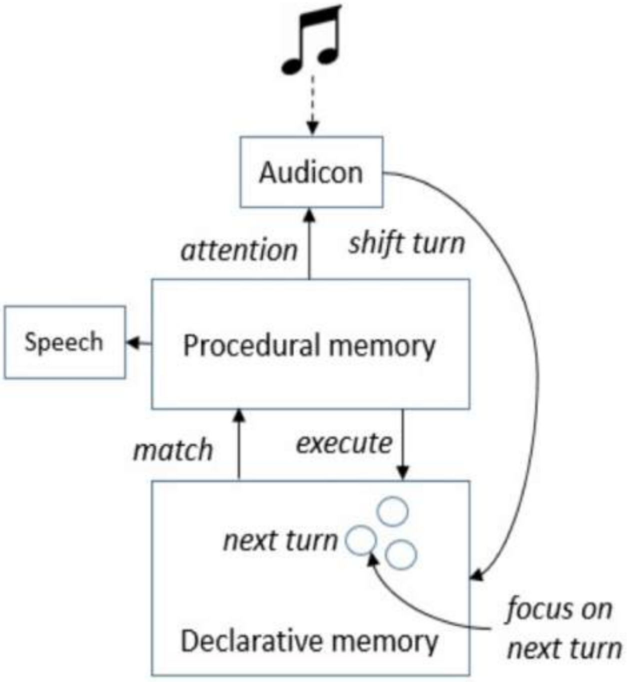


Figure S1. The ACT-R components for inner speech. Related to Figure 4. The Audicon detects the external sound that is the vocal command of the partner. The buffer of the Audicon stores the chunk representation of the audio until 2 seconds, and the procedural memory matches that chunk to the left-pole of the rules. In this phase, the attention is focused on that turn. When a rule fires, the procedural memory executes the corresponding right pole. The execution may update the old chunk or retrieve a other one from the declarative memory, leading to the emergence of next turn. In any case, the resulted chunk of the execution is produced by the Speech module and rehearsed by the Audicon, so ending a cognitive inner speech cycle.

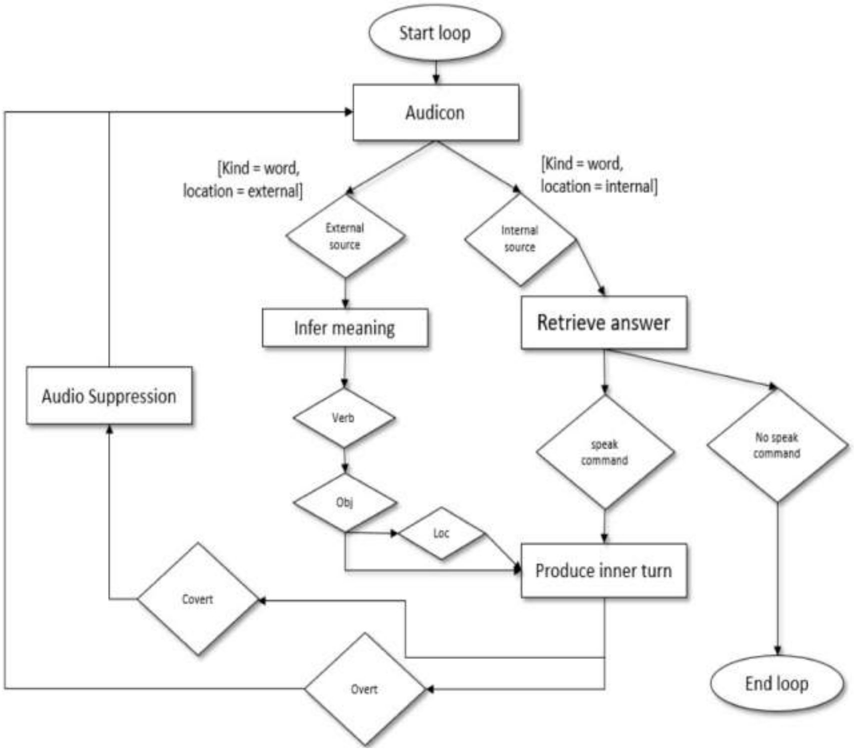
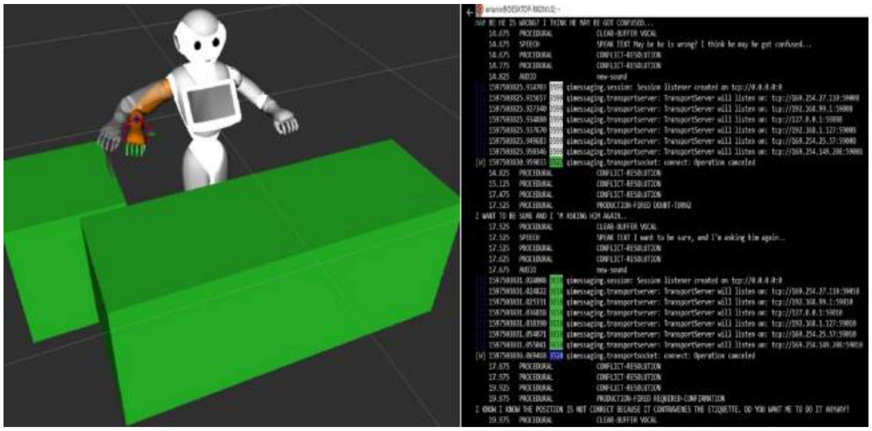
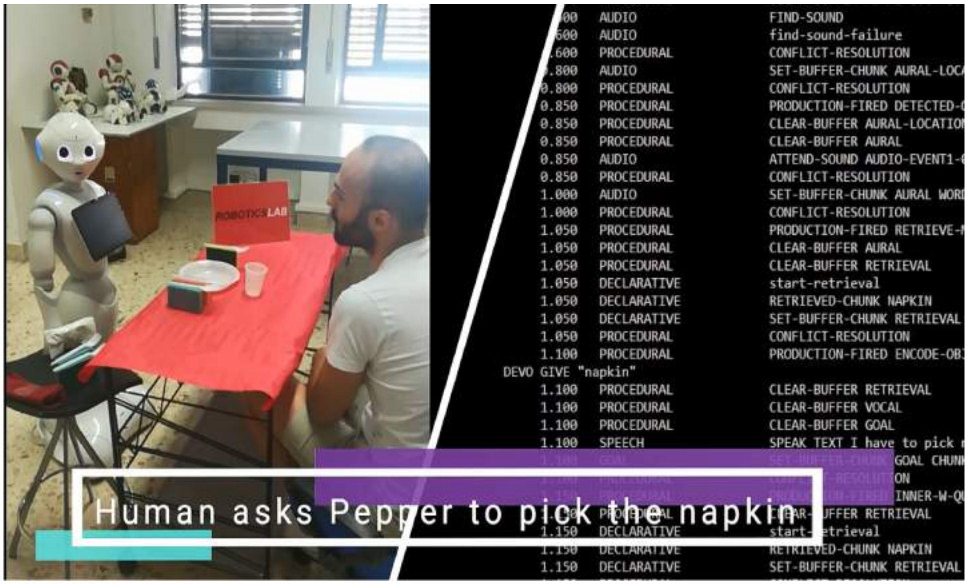


Figure S2. The ACT-R model of inner speech. Related to Figure 4. The diamonds define conditions to be evaluated, while squares represent actions. One or more production rules correspond to a square. In fact more rules could be executed for achieving an action. The cognitive cycle representing the phonological loop starts when the Audicon detects a sound. If the sound comes from an external source (the `External source` diamond is true), it represents a partner's request, and the `Infer meaning` square allows inferring the semantic sense of such a request. Once the model understands the meaning of the request (the `verb` diamond, the `object` diamond and the `location` diamond identify the corresponding pos tags of the words), it produces the first turn of the inner dialogue (the `Produce inner turn` square), that is back-propagated to the Audicon. In this case, the sound comes from an internal source, and the model attempts to retrieve the answer to this inner turn (the `Retrieve answer` square). When almost a production rule in the square executes the speak command, the model produces the answer corresponding to the current turn. The answer becomes the new turn of the inner dialogue. The loop restarts for this new turn. The loop will stop when the involved production rule in the `Retrieve answer` square does not execute the speak command, and no further turn emerges.



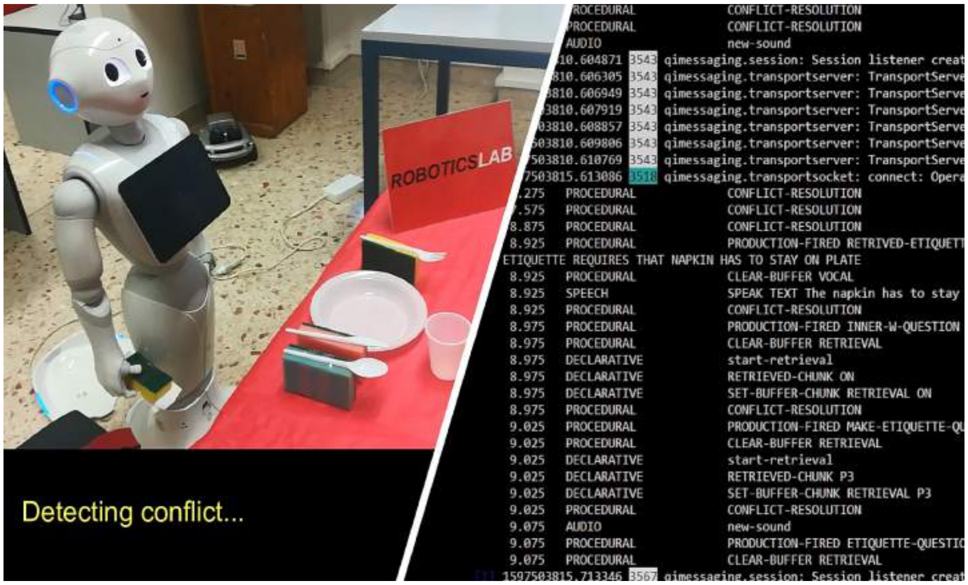
```
MAY BE HE IS WRONG? I THINK HE MAY BE GOT CONFUSED...
    14.675   PROCEDURAL           CLEAR-BUFFER VOCAL
    14.675   SPEECH               SPEAK TEXT May be he is wrong? I think he may be got confused...
    14.675   PROCEDURAL           CONFLICT-RESOLUTION
    14.775   PROCEDURAL           CONFLICT-RESOLUTION
    14.825   AUDIO                new-sound
    1587583825.914703 [illegible] qimessaging.session: Session listener created on tcp://0.0.0.0:0
    1587583825.915657 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://169.254.27.110:59000
    1587583825.927340 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://192.168.99.1:59000
    1587583825.934880 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://127.0.0.1:59000
    1587583825.937670 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://192.168.1.127:59000
    1587583825.949683 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://169.254.25.57:59000
    1587583825.950346 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://169.254.149.208:59000
[W] 1587583830.959033 [illegible] qimessaging.transportsocket: connect: Operation canceled
    14.825   PROCEDURAL           CONFLICT-RESOLUTION
    15.125   PROCEDURAL           CONFLICT-RESOLUTION
    17.475   PROCEDURAL           CONFLICT-RESOLUTION
    17.525   PROCEDURAL           PRODUCTION-FIRED DOUBT-TURN2
I WANT TO BE SURE AND I 'M ASKING HIM AGAIN..
    17.525   PROCEDURAL           CLEAR-BUFFER VOCAL
    17.525   SPEECH               SPEAK TEXT I want to be sure, and I'm asking him again..
    17.525   PROCEDURAL           CONFLICT-RESOLUTION
    17.625   PROCEDURAL           CONFLICT-RESOLUTION
    17.675   AUDIO                new-sound
    1587583831.024000 [illegible] qimessaging.session: Session listener created on tcp://0.0.0.0:0
    1587583831.024822 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://169.254.27.110:59010
    1587583831.025171 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://192.168.99.1:59010
    1587583831.036038 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://127.0.0.1:59010
    1587583831.038390 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://192.168.1.127:59010
    1587583831.054071 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://169.254.25.57:59010
    1587583831.055047 [illegible] qimessaging.transportserver: TransportServer will listen on: tcp://169.254.149.208:59010
[W] 1587583836.069468 [illegible] qimessaging.transportsocket: connect: Operation canceled
    17.675   PROCEDURAL           CONFLICT-RESOLUTION
    17.975   PROCEDURAL           CONFLICT-RESOLUTION
    19.925   PROCEDURAL           CONFLICT-RESOLUTION
    19.975   PROCEDURAL           PRODUCTION-FIRED REQUIRED-CONFIRMATION
I KNOW I KNOW THE POSITION IS NOT CORRECT BECAUSE IT CONTRAVENES THE ETIQUETTE. DO YOU WANT ME TO DO IT ANYWAY!
    19.975   PROCEDURAL           CLEAR-BUFFER VOCAL
```

Figure S3. The simulation-based testing technique for verifying and validating the inner speech model. Related to Figure 4. Two simulators allowed monitoring the robot's functioning and inner speech. The first simulator was the ROS visualizer where the scenario was reproduced. It shows he Pepper's avatar between two blocks, representing the little table from which to pick the utensil, and the big table on which to place it. A very little block represents the utensil to move. The robot is controlled by the inner speech model which run in parallel in the ACT-R shell simulator, where sequences of active modules of the inner speech model and the turns of the inner dialogue were printed by the model itself.



```
 00   AUDIO                  FIND-SOUND
 600  AUDIO                  find-sound-failure
.600  PROCEDURAL             CONFLICT-RESOLUTION
.800  AUDIO                  SET-BUFFER-CHUNK AURAL-LOCA
0.800 PROCEDURAL             CONFLICT-RESOLUTION
0.850 PROCEDURAL             PRODUCTION-FIRED DETECTED-
0.850 PROCEDURAL             CLEAR-BUFFER AURAL-LOCATION
0.850 PROCEDURAL             CLEAR-BUFFER AURAL
0.850 AUDIO                  ATTEND-SOUND AUDIO-EVENT1-
0.850 PROCEDURAL             CONFLICT-RESOLUTION
1.000 AUDIO                  SET-BUFFER-CHUNK AURAL WORD
1.000 PROCEDURAL             CONFLICT-RESOLUTION
1.050 PROCEDURAL             PRODUCTION-FIRED RETRIEVE-
1.050 PROCEDURAL             CLEAR-BUFFER AURAL
1.050 PROCEDURAL             CLEAR-BUFFER RETRIEVAL
1.050 DECLARATIVE            start-retrieval
1.050 DECLARATIVE            RETRIEVED-CHUNK NAPKIN
1.050 DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL
1.050 PROCEDURAL             CONFLICT-RESOLUTION
1.100 PROCEDURAL             PRODUCTION-FIRED ENCODE-OB
DEVO GIVE "napkin"
1.100 PROCEDURAL             CLEAR-BUFFER RETRIEVAL
1.100 PROCEDURAL             CLEAR-BUFFER VOCAL
1.100 PROCEDURAL             CLEAR-BUFFER GOAL
1.100 SPEECH                 SPEAK TEXT I have to pick
1.100 GOAL                   SET-BUFFER-CHUNK GOAL CHUNK
1.100 PROCEDURAL             CONFLICT-RESOLUTION
1.150 PROCEDURAL             PRODUCTION-FIRED INNER-W-QU
1.150 PROCEDURAL             CLEAR-BUFFER RETRIEVAL
1.150 DECLARATIVE            start-retrieval
1.150 DECLARATIVE            RETRIEVED-CHUNK NAPKIN
1.150 DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL
```

Figure S4. Scene from video of Thread 1. Related to Table 1. The robot explains its underlying processes by inner speech.



```
ROCEDURAL                    CONFLICT-RESOLUTION
PROCEDURAL                   CONFLICT-RESOLUTION
AUDIO                        new-sound
10.604871 3543 qimessaging.session: Session listener creat
810.606305 3543 qimessaging.transportserver: TransportServe
3810.606949 3543 qimessaging.transportserver: TransportServe
3810.607919 3543 qimessaging.transportserver: TransportServe
03810.608857 3543 qimessaging.transportserver: TransportServe
503810.609806 3543 qimessaging.transportserver: TransportServe
7503810.610769 3543 qimessaging.transportserver: TransportServe
7503815.613086 3518 qimessaging.transportsocket: connect: Opera
.275  PROCEDURAL             CONFLICT-RESOLUTION
.575  PROCEDURAL             CONFLICT-RESOLUTION
8.875 PROCEDURAL             CONFLICT-RESOLUTION
8.925 PROCEDURAL             PRODUCTION-FIRED RETRIVED-ETIQUETT
ETIQUETTE REQUIRES THAT NAPKIN HAS TO STAY ON PLATE
8.925 PROCEDURAL             CLEAR-BUFFER VOCAL
8.925 SPEECH                 SPEAK TEXT The napkin has to stay
8.925 PROCEDURAL             CONFLICT-RESOLUTION
8.975 PROCEDURAL             PRODUCTION-FIRED INNER-W-QUESTION
8.975 PROCEDURAL             CLEAR-BUFFER RETRIEVAL
8.975 DECLARATIVE            start-retrieval
8.975 DECLARATIVE            RETRIEVED-CHUNK ON
8.975 DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL ON
8.975 PROCEDURAL             CONFLICT-RESOLUTION
9.025 PROCEDURAL             PRODUCTION-FIRED MAKE-ETIQUETTE-QU
9.025 PROCEDURAL             CLEAR-BUFFER RETRIEVAL
9.025 DECLARATIVE            start-retrieval
9.025 DECLARATIVE            RETRIEVED-CHUNK P3
9.025 DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL P3
9.025 PROCEDURAL             CONFLICT-RESOLUTION
9.075 AUDIO                  new-sound
9.075 PROCEDURAL             PRODUCTION-FIRED ETIQUETTE-QUESTIC
9.075 PROCEDURAL             CLEAR-BUFFER RETRIEVAL
1597503815.713346 3567 qimessaging.session: Session listener creat
```

Figure S5. Scene from video of Thread 3. Related to the Table 4. The conflict resolution related to discrepancy situation by robot's inner speech.

State	Meaning
`OK`	All components work properly
`BattLow`	The battery is dead
`RightKo`	A joint in the right arm does not work
`LeftKo`	A joint in the left arm does not work
`RightHot`	The right arm is overheated
`LeftHot`	The left arm is overheated

Table S1. The possible states of the robot at the beginning of each trial. Related to Figure 4. Each state can affect the unfolding of interaction. For example, if the right arm is overheated, and the robot has to use that arm for accomplishing the task, it becomes aware of that situation by evaluative inner speech and then alerts the partner about the impossibility to end the task successfully. Only these states are considered in the experimental session.

ID	State	Table config
I1	`OK`	plate, knife, fork
I2	`OK`	plate, fork, glass
I3	`RightHot`	plate, knife, spoon, glass
I4	`RightKo`	plate
I5	`LeftKo`	plate, knife, fork
I6	`OK`	plate
I7	`BattLow`	plate, fork
I8	`LeftHot`	plate, knife, fork

Table S2. The trials' initial context used in the experimental session. Related to Figure 3. Each initial context has a unique identifier, which will be used for representing it when used as initial context in a trial. The identifies are represented in the *ID* column. An identifier contains a progressive number, and it is associated to the state and to the initial table configuration, that are the *State* and *Table config* columns respectively. They represent for a trial the state of the robot and the utensils already on the table to set when that context is initial for that trial.

# Trial	Initial context	Trigger	Conflict
1	I1	*Give me the napkin*	No Conflict
2	I1	*Place the napkin at the left of the fork*	Contravene Etiquette
3	I7	*Pick the knife*	Battery low
4	I6	*Pick the knife and place it on the plate*	Contravene Etiquette
5	I3	*Place the fork near the glass*	Contravene Etiquette
6	I4	*Pick the fork*	Right arm does not work
...	...	...	...
30	I2	*Pick the fork*	Discrepancy

Table S3. An excerpt of the 30 trials per block. Related to Table 5. For each trial, the initial context, the trigger and the possible conflict are indicated. The initial context is represented by its unique identifier. The trigger is the human's verbal command which specifies the task to solve in the trial. Each sentence is purposely encoded to be compliant with the grammar of the robot. Finally, the existence of a possible conflict is indicated in the last column. The conflict can be generated by a compromised state of the robot, by a human's request which infringes the etiquette (the *Contravene Etiquette* value) or which regards an utensil already on the table (the *Discrepancy* value).

	Real Experiments	Simulation
Number of test	20	70
RI	90% (18/29)	87.1% (61/70)
Runtime error	0.05% (1/20)	0.03% (2/70)
Over time	0.1% (2/20)	0.03% (2/70)
Transparent test	100%	100%

Table S4. Validation of the model. Related to Figure 4. The table showing the final stage of the validation phase of the model. The rows show the measured parameters (that are those from the Standards), and the columns show the used techniques for validating the model. The model is validated when it runs in two different modes of functioning, that are the simulation and the real-experiments, and the detected measures have similar values in both functioning modes. When these values deviate between them too much, it means that the model needs to be tuned. We changed the assets of the models until these values are similar.

Transparent Methods

Experimental setup details

The initial context. The initial context represents the state of the table and of the robot at the start of each trial. For the experimental session, 8 initial contexts have been defined. They largely cover all the possible initial contexts, because any possible context may fall in one of them. Table S2 shows the 8 initial contexts, one for row. An initial context has a unique identifier, which is indicated in the *ID* column, and the context specification, that is the robot state and the initial table configuration. They are indicated in the *State* and *Table config* columns. The initial context identifiers are used for referring to the initial configurations of the trials.

The state of the robot is one of those represented at Table S1, where the specific meaning for each state is described in the corresponding *meaning* column.

The defined trials. Table S3 contains an excerpt of the whole trials' descriptions. The description of a trial is the specification of both its initial context and the trigger. Once the robot detects the trigger, the trail starts. The robot will act differently depending on the fact that inner speech skill is enabled or not. For practical reasons, human-to-robot verbal request are predefined and, in some cases, they purposely generate a conflict. In same way, when a malfunctioning has to be detected, the state of the robot is hand-encoded for simulating it. No robots were mistreated for these experiments.

During trial execution, the participant expected to answer to possible further queries by robot, to listen the robot discourse, or that the required utensil is placed on the table.

Modeling robot's inner speech

Theoretical background Over the last years, some studies and progress have been made in modeling humans' inner speech. In his book, Fernyhough (2016) has built up an interesting overview of inner speech and its functions addressing a wide array of research topics such as developmental, social psychology, neuroscience, sport, and others. In the same line, Morin (2012) and Alderson-Day and Fernyhough (2015) propose two of the most important and more comprehensive recent reviews about the role of inner speech in many cognitive functions, and Gregory (2020) presents the most recent results and experiments on human's inner speech.

In this literature scene, there are some evidences about the importance of a form of self-dialogue in artificial agents. Steels (2003) focused on the rehearsal of own verbal productions. He demonstrated that the language re-entrance affects the grammar emergence from a population of agents who converse with each other, and hear themselves at the same time. Each agent is able to produce and to parse sentences by output and input channels respectively. By the dialogue between them, they agree on the linguistic grammar they shared. When each agent was provided by language re-entrance (e.g., its output channel was back-propagated to the input one), the emergent grammar was more refined than the case in which that back-propagation was down.

Clowes and Morse (2005) analyzed back-propagation in a one-level neural network, in which input and output neurons are associated to words. Input words specify commands to execute, and output neurons correspond to the action to execute to accomplish the command. The back-propagation allowed to classify the correct action more times than the case in which the input and output neurons are not linked.

In the same line, Mirolli and Parisi (2006) employed a simple neural network model for language acquisition, in the perspective of the evolutionary emergence of human language. They demonstrated that the use of language for oneself, i.e., as private or inner speech, improves the individual's classification of the words.

One of the most recent work (Oktar et al., 2020) defines the same kind of back-propagation from output to input channels in chatbots, leading to similar improved results.

All these cases evidence the importance of linguistic rehearsal for artificial artifacts. However, they only offer partial explanations of the reported phenomena.

The improvement of the behaviours in the cited studies inspired the proposed work, leading to the possibility to improve by inner speech the performances of a robot and the quality of interaction when it cooperates with humans.

The authors already proposed and analyzed a logical model of robot inner speech based on the event calculus (Chella and Pipitone, 2019), and then by defining a complete cognitive architecture of inner speech (Chella et al., 2020) based on the Standard Model of Mind (Laird et al., 2017).

In the first study, inner dialogue was modeled by axioms and symbols, and the sequence of the dialogue emerged by the natural deduction process (Gentzen, 1964). The model is a proof-of-concept, and allowed to test a form of automatized inner speech, highlighting its role in solving decisional problems. In that case, the robot and the human are placed in front a table, on whose surface there were a set of differently colored boxes arranged in casual positions. The human asked the robot where is a specific box, by indicating its color. The calculus' formulas of inner speech made the robot able to answer to the human's question, while verbally reasoning on the context. In the meantime the human was able to listen the whole reasoning process

In the second study, the robot architecture for inner speech took inspiration from the Baddley's theory of human's inner speech (Baddeley, 1992). Baddley claims that inner speech is a rehearsal process by which people repeat information (as a phone number, an address and so on), and temporarily keep them in mind. After a number of repetitions and rehearsals, the data are permanently memorized. Baddley proposes a cognitive model of that process. Temporarily data are maintained in a short-term memory, which is a working memory composed of the *central executive*, a master system supervising the rehearsal process of memorization, and two slave subsystems: the *visual-spatial sketchpad* for visual data memorization, and the *phonological loop* for phonological data memorization. This loop is responsible for the inner speech ability. The phonological loop is in turn composed of the *phonological store* and the *articulator component*. The phonological store is a kind of *inner ear* that keeps traces of event sounds according to their temporal order. Instead, the articulator acts as a kind of *inner voice* producing sounds. Such a loop enables the memorization of the phonological data which remains in the short-term memory for a time longer than 2 seconds, and then it is switched to the long-term one.

Inspired from the Baddley's theory, the proposed robot cognitive architecture of inner speech implements the *elaborate rehearsal* (Craik and Lockhart, 1972). When a sound is heard, related concepts can emerge from the knowledge of the agent, thus allowing for inferential and reasoning processes. The rehearsal process does not concern the repetition of heard sound only, but the recalling of new associations and new inferences. It enables the robot to self-talk about the context and to keep decisions.

Design and implementation The cognitive architecture of inner speech is based on the ACT-R framework (Anderson et al., 1997). The framework is formed by a set of *modules* and *buffers*. A module represents specialized brain structures and solves specific cognitive functions (as vision, speech, memory, and so on). A buffer is the interface of a specific module and is linked to that module. It is a short term memory that stores information related to the context. The content of all the buffers at a time is the state of the model in such time.

There are two kinds of memory modules, representing *declarative knowledge* and *procedural knowledge*.

The declarative knowledge is a set of facts, each fact represented by a *chunk* (i.e., a frame-like structure), while the procedural knowledge is a set of *production rules* describing the procedures to follow for keeping a task. A production rule has two poles (right and left): the right pole defines the condition patterns for matching chunks, while the left pole defines the actions to take in case the condition matches, and hence the rule fires.

ACT-R provides a further component, that is the *pattern matcher*. It manages the *matching*, the *selection* and the *execution* of the production rules. The pattern matcher *matches* the right pole of the production rules to the chunks into the buffers: if a chunk matches to a production rule, then the rule is *selected* and its left pole is *executed*. The execution updates the value of the chunk, or it retrieves other chunks from other modules.

In particular, the cognitive architecture of inner speech involves two modules, which are the *Audicon* and the *Speech* modules. The *Audicon* attends to sound events, while the *Speech* module is responsible for the verbal production of sentences.

Figure S1 shows the schematic representation of the ACT-R cognitive architecture of inner speech. The Audicon module attends to partner's vocal command. It encodes the perceived turn and keeps it in the buffer for 2 seconds, according to Baddley's theory. It is important to hightlight that the Audicon has the role of the Baddley's phonological store.

If the turn in the buffer of the Audicon matches to the right pole of a production rule, then the *attention* focuses that turn, and the left pole of the rule *shifts to the next turn*. The turn generally contains newly retrieved information from the declarative memory. The execution by procedural memory may update the old chunk or retrieve a new chunk. The Speech module produces this turn. At this step, the speech production is simulated by a suitable ACT-R `speak` command. No audio is audible in the environment.

The output of the Speech module is rehearsed by the Audicon: at this step, the old cycle ends and a new cycle starts by repeating the procedures with the new turn.

The diagram in Figure S2 shows in details how the inner speech model operates. A diamond represents the output of a condition (i.e, the result of matching between a left-pole and a chunk), while the square represents the actions execution. Each square corresponds to a single or a set of production rules in the cognitive architecture.

At the start of the looping cycle, the model checks the Audicon searching for new items. If there is a new item, then the model checks the source location of the detected sound. If the sound comes from an external location, then it corresponds to a partner's request. Otherwise, it is generated by an internal source and it corresponds to a turn of inner speech.

When the sound comes from an external source, then the model infers the meaning of the partner request (the `infer meaning` square) by a linguistics analysis of the sentence, based on the analysis of the verb, the object, and the possible location.

The linguistic analysis is based on the evidence that the verb, the object, and the location parts of speech typically follow this sequential order, as claimed by Blake (1988). Moreover, in the current implementation of the model, the verb is transitive only. The requests to the robot look like: "pick the book", "give me the apple on the table", "close the door at the left".

Once the model infers the user request, a first turn of evaluative inner speech emerges, and the robot talks about what it has to do, as "I have to pick the book", "I have to give the human the apple on the table".

The Audicon detects the produced sentence by the `produce inner turn` square. The production rules in that block match the inner sentence(whose location is now internal) with the declarative knowledge to retrieve the answer to the current turn (the `retrieve answer` square). The robot may ask itself if it sees the object to pick, or where the object is, or if its state allows it to perform the desired action. Also, the robot can talk to itself about the morality of the action ("I don't want to tear the pages!", "I will not break the door!"), or about a conflict that the execution of the action can generate in the robot ("I can not reach the book", "My grippers are too little for keeping the book").

An example of inner dialogue is reported below (H: user, R: robot):

H: *Pick the book*

R: *I have to pick the book*

R: *My grippers are too little for keeping the book*

R: *I should to tell that I can not keep the book*

R: *I hope the human will have understanding for my fix!*

R: *Sorry human, but I can not pick the book!*

When no further answers emerge, the model does not run the speak command, and the inner dialogue ends.

The declarative knowledge of the model regards the words and the dialogue turns. The definition of specific *chunk-type* models them. A chunk-type is the structure of chunk in a frame-like representation. The frame is a list whose head is the name of the chunk-type, followed by a set of slots. There are three kinds of basic chunk-types in the model. The type for modeling words, for modeling inner speech related to a sentence evaluation, and for modeling other inner dialogue turns (involving both evaluative and moral inner speech turns).

A word is encoded by the linguistic `word` frame:

```
(chunk-type word syntax sense pos act)
```

which models the semantic sense of the word (the slot `sense`), its surface form (the slot `syntax`) and its part-of-speech role (the slot `pos`), i.e. if it is a verb, a noun (generally, a noun identifies the object) or an adverb (which identifies a possible position). Moreover, in the case the chunk represents a verb, the slot `act` identifies the action to take corresponding to that verb. For example, for the verb *give*, the action will be *pick* because just by picking an object it is possible to give it. For the other pos cases, the slot will be not instantiated. Examples of items encoded by the linguistic `word` chunk-type are:

```
(pick06
ISA word
syntax "give"
sense pick
pos verb
act "pick" )

(table23
ISA sense
syntax "table"
sense table
pos noun
act null)
```

The chunk-type to model an inner evaluative sentence looks like:

```
(chunk-type inner-eval verb obj1 obj2 risk benefit symb)
```

which models an inner evaluation about the action execution, represented by the slots `verb` and involving the objects `obj1` and `obj2`. The evaluation is measured by suitable values in the slots `risk` and `benefit`, while the slot `symb` is the turn for explaining the decision.

For example:

```
(p4
ISA inner-eval
verb pick obj1 table obj2 null
risk 1 benefit 0
symb "It is not possible to pick a table!")
```

is a proposition that models the evaluation of the action *"pick the table"*, which has only risks and no benefits.

Or again, in the case of etiquette requirements, the proposition:

```
(p11
ISA inner-eval
verb place obj1 napkin obj2 table
risk 0.8 benefit 0.2
symb "It contravenes the etiquette!")
```

models the conflict situation of infringing the etiquette rule.

To encode spoken commands by the partner, the chunk-type is:

```
(chunk-type comprehend-voo verb object adverb location)
```

The synthesized sounds related to the command are detected and then searched in the declarative memory by chunks of that type. In this way, the sounds are encoded. For example, if the user tells the robot to close the door by the sentence "*Close the door!"*, the detected sounds will be encoded by the set of words {"close", "door"} and the robot will search for the chunk `(pX comprehend-voo verb close object door)` for encoding the words. Then it could search for the `inner-eval` chunk-type for retrieving the corresponding risk and benefit values, or for other kinds of evaluations.

Finally, the chunk-type to model a inner turn looks like:

```
(chunk-type turns-link inner-turn-1 inner-turn-2)
```

which associates to the turn in the `inner -turn-1` slot, another inner sentence in the `inner -turn-2` slot. Such a chunk-type models a step of the dialogue with a "start consideration" and the related "answer".

It is to be noticed that for the same sentence in the first slot, there could be different possible turns. So, there will be different chunks with the same `inner-turn-1` slot, but having different `inner-turn-2` slot. Moreover, sentences in the second slot could be in the first slot of other chunks. In this way, a chain of turns emerges, defining a dialogue thread.

Examples of links between turns are:

```
(p78
turns-link link102
inner-turn-1 ``It is not possible to pick a table!''
inner-turn-2 ``I will tell that such an action is a not sense'')
```

and once again:

```
(p79
turns-link link103
inner-turn-1 ``I will tell that such an action is a not sense''
inner-turn-2 ``Sorry human, the table is too heavy for me'')
```

or:

```
(p81
turns-link link105
inner-turn-1 ``I will tell that such an action is a not sense''
inner-turn-2 ``It's a stupid action...'')
```

The mechanism of the choice of the next turn depends on the *base-level activation* mechanism of ACT-R which associates an activation value to each of the instantiated chunk in the declarative memory, depending on previous use of the chunk. This value decays during time, and more times a chunk is retrieved, more probability it has to be further retrieved next time in the session. This value represents an estimation of the need of the chunk in the current context.

Starting from this activation mechanism, when the model is reset and a new working session starts, then each chunk has the same probability to emerge. Once a chunk is activated, then its activation level grows, and the same chunk becomes more active than the others. When the chunks model the links between turns, then the activation mechanism allows the selection of the same turn in correspondence to the same sentence. Such a mechanism facilitates the repetition of the robot behavior in the same dialogue thread, thus avoiding dialogue contradictions, and simulating that the robot maintains the same "idea."

To customize the proposed model on the analyzed scenario, it was necessary to add specific new chunk-types and to define concepts of the domain. To model the inner turns related to the etiquette, the new chunk-type is:

```
        (chunk-type inner-etiquette-question pos obj1 obj2 symb)
```

which models the relative position of the utensils in the table according to the etiquette. For example:

```
(p8 ISA inner-etiquette-question
pos left obj1 fork obj2 plate
symb "The fork has to stay at the left of the plate")

(p6 ISA inner-etiquette-question
pos under obj1 fork obj2 glass
symb "The fork has to stay under the glass")
```

model the etiquette rules about the position of the fork in the table (at the left of the plate and under the glass).

Moreover, the knowledge about the current context has to be modeled. For this purpose, it was necessary to add the chunk-type inner-where:

```
                    (chunk-type inner-where obj place)
```

which models the fact that the object obj is already on the table or not (the slot place has ``basket'' or ``table'' value for modeling the current location of the object).

The basic domain concepts in the presented scenarios are modeled by the `word` chunk-type. Formally, being U the set of utensils, V the possible actions to take and P the set of the relative positions, the set of chunks of type `word` for the analyzed scenario is $W = U \cup V \cup P$, where:

- U = $\{$*fork, plate, spoon, knife, napkin, glass*$\}$
- V = $\{$*take, give, pick, place, move, grasp, rest*$\}$
- P = $\{$*up, left, right, top, over, down, under, on*$\}$

Some examples of words are:

```
(rest ISA word syntax "rest" sense rest pos verb act "rest")
(left1 ISA word syntax "left" sense left pos adv act null)
```

In the proposed examples, the initial configuration of the table is not empty: it is partially set to enable the robot to keep decisions about a context with existing constraints.

The initial configuration of the table contains utensils which are all in correct positions, as shown in Figure 1. In the declarative memory, such a knowledge is modeled by facts like these:

```
(p4 ISA inner-where obj napkin place basket)
(p5 ISA inner-where obj fork place table)
(p6 ISA inner-where obj knife place table)
```

Deploying the inner speech model in real robots The described computational model cannot be immediately deployed on a real robot. It is necessary to integrate it in a complete robot architecture. For this purpose, the work concerned with the definition of a global framework enabling the robot to use the proposed ACT-R model, and hence self-talking. Figure 4 shows the proposed framework for robot inner speech. The Figure shows the *Memory* system layer and the perceptual motor layer, which is subdivided into the *Motor* and *Perception* sublayers.

The Memory system stores and retrieves the content needed to support the processes involved in inner speech. Such a content concerns the *declarative knowledge* representing concepts and facts about the domain, and the *procedural knowledge*, related to the processes (or procedures) to follow to reach a goal. The knowledge related to the context is temporary stored into a *working memory*, that manages the activation of the procedures into the procedural component, and the information retrieval from the declarative component.

The perceptual motor layer models the interaction with the external environment. It includes all the needed components to perform actions and to perceive entities.

The module devoted to the listening of a sound is the *Audicon* module included into the Perception block. In the Perception block, the SST module decodifies sentences, i.e., it associates the symbolic forms to the audio sounds as shown previously. It is a typical speech recognition process that associates a string representation to the audio sound.

By considering that the robot's native routines to decode speech from the external environment are often limited (often they require to define a set of words to recognize, so excluding words recognition for those who are not in the set), the STT module of the framework uses the Google API library[1]. It allows to recognize a wide range of words, adding interesting features, as noise suppression, and different language identification.

The subsystem that enables the robot to perform actions, as to pick and place an object, is the Robot Operating System (ROS) (Quigley et al., 2009) module, a component of the Motor block, together with the TTS component. ROS is a state of the art framework for robot programming, which provides a set of libraries covering several robot behaviors. In the proposed framework, ROS enables the robot to perform the actions the human requires. The robot's movements for taking actions are implemented by the MoveIt! ROS library (Görner et al., 2019), that is purposely designed for robot action planning and for modeling manipulation actions.

The TTS module codifies sentences, or dialogue turns: the sentence codification transforms labels, that are the symbolic forms of the words, to audible sound by vocal synthesizers. The codified sentences may be from inner processes (the robot overtly generates inner speech) or from external interactions (the robot answers to a query or generates questions). The framework has two different TTS functionalities: for abstracting to the specific robot model, it provides directly an output sound based on the Python engine `gTTS`[2], which stands for Google Text To Speech. In this case the framework will use the hardware synthesizers of the machine on which it will be run.

An important task of the middleware component is the linguistic analysis of the sentences from the STT. To identify the *keywords* of the external request, the component pre-processes the utterances and then sends the results to the Audicon. The linguistic pre-processing concerns:

1. Part-of-Speech (POS) annotation: each word is annotated by the tag identifying its POS role in the sentence. It may be a verb, or a noun, or an article, and so on;
2. Stop-words deletion: not meaningful words as articles, prepositions, conjunctions are removed;
3. Sentence tokenization: the sentence is subdivided in tokens, where each token is a word.

Validating the model The model was verified and validated by using the approach for human-robot team described at Webster et al. (2016). The method consists of corroborating different available validation techniques about the requirements of the standards. In few words, the evidences of the requirements from an available validation technique has to be confirmed by another one (i.e., the second technique corroborates the first one). The available techniques are the *simulation-based testing* and *real experiments*.

The simulation-based testing consists of simulating the execution of the model and verifying the satisfaction of requirements. Two kinds of simulators were implemented.

[1] https://cloud.google.com/speech-to-text/docs/

[2] https://pypi.org/project/gTTS/

One for testing robot's movements and routines execution, the other one for monitoring robot's inner speech. The first simulator was implemented by using ROS which provides a visualizer for reproducing the scenario and the robot's behavior. The Figure S3 shows the simulated environment. Here it is possible to see the Pepper's avatar to pick objects from the small table and to put them in the big one. The second simulator was the ACT-R shell that shows the model execution and the sentences of the inner dialogue. A testbench of vocal commands were defined, and one of them was randomly drawn for each test. The inner speech model controlled the robot in the ROS simulator. In this way, the model was tested by considering the result of the operation for a specific vocal command, in terms of inner dialogue and routines execution for achieving the command.

The real experiments technique corroborated the simulation one if the robot's behavior satisfies the same requirements. The real experiments in validation phase were executed with robot's inner speech.

According to this approach, when for some reason a requirement is not satisfied in one of the available techniques, then the assets of the model were suitably tuned.

The model has been executed 70 times during the simulation-based testing, and 20 times during real experiments. Table S4 shows the test outcomes and the occurrence rates of the individual requirement satisfactions for the investigated scenario, concerning 20 real experiments and 70 simulations after tuning.

Supplemental References

Alderson-Day, B., Fernyhough, C.. (2015). Inner speech: development, cognitive functions, phenomenology, and neurobiology. Psychological bulletin 141, 931–965. 10.1037/bul0000021.

Anderson, J.R., Matessa, M., Lebiere, C.. (1997). Act-r: A theory of higher level cognition and its relation to visual attention. Human-Computer Interaction 12(4), 439–462.

Baddeley, A.. (1992). Working memory. Science 255, 556–559. 10.1126/science.1736359.

Blake, B.B.. (1988). Russell s. tomlin, basic word order. functional principles. london: Croom helm, 1986. pp. 308. Journal of Linguistics 24, 213–217. 10.1017/S0022226700011646.

Chella, A., Pipitone, A.. (2019). A cognitive architecture for inner speech. Cognitive Systems Research 59, 287–292. 10.1016/j.cogsys.2019.09.010.

Chella, A., Pipitone, A., Morin, A., Racy, F.. (2020). Developing self-awareness in robots via inner speech. Frontiers in Robotics and AI 7, 16. 10.3389/frobt.2020.00016.

Clowes, R., Morse, A.F.. (2005). Scaffolding cognition with words in Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems. eds. L. Berthouze, F. Kaplan, H. Kozima, H. Yano, J. Konczak, G. Metta, J. et al. (Lund University Cognitive Studies). pp. 101–105.

Craik, F., Lockhart, R.. (1972). Levels of processing: A framework for memory research. Journal of Verbal Learning and Verbal Behavior 11, 671–684. 10.1016/S0022–5371(72)80001–X.

Fernyhough, C.. (2016). The voices within: The history and science of how we talk to ourselves. (New York, NY: Basic Books.Basic Books).

Gentzen, G.. (1964). Investigations into logical deduction. American Philosophical Quarterly 1, 288–306. 10.2307/2272429.

Gregory, D.. (2020). Inner speech: New voices. Analysis 80, 164–173. 10.1093/analys/anz096.

Görner, M., Haschke, R., Ritter, H., Zhang, J.. (2019). Moveit! task constructor for task-level motion planning, in: 2019 International Conference on Robotics and Automation (ICRA), pp. 190–196.

Laird, J.E., Lebiere, C., Rosenbloom, P.S.. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. Ai Magazine 38, 13–26. 10.1609/aimag.v38i4.2744.

Mirolli, M., Parisi, D.. (2006). Talking to oneself as a selective pressure for the emergence of language, pp. 214–221. 10.1142/9789812774262_0028.

Morin, A.. (2012). Inner Speech. in Encyclopedia of Human Behavior ed. (W. Hirstein San Diego, CA: Elsevier). pp. 436–443.

Oktar, Y., Okur, E., Turkan, M.. (2020). The mimicry game: Towards self-recognition in chatbots. arXiv preprint arXiv:2002.02334 .

Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.. (2009). Ros: an open-source robot operating system, in: ICRA Workshop on Open Source Software.

Steels, L.. (2003). Language re-entrance and the ‘inner voice’. Journal of Consciousness Studies 10, 173–185.

Webster, M., Western, D., Araiza-Illan, D., Dixon, C., Eder, K., Fisher, M., Pipe, A.G.. (2016). An assurance-based approach to verification and validation of human-robot teams. CoRR abs/1608.07403.