A Computational Model for Cognitive Human-Robot Interaction: An Approach Based on Theory of Delegation

Filippo Cantucci
Institute of Cognitive Science and Technology, National Research Council of Italy, (ISTC-CNR)
Rome, Italy
filippo.cantucci@istc.cnr.it

Rino Falcone
Institute of Cognitive Science and Technology, National Research Council of Italy, (ISTC-CNR)
Rome, Italy
rino.falcone@istc.cnr.it

***Abstract*—In this paper we present a cognitive model to support reasoning and decision making on socially adaptive task delegation and adoption. The designed model allows a robot to dynamically modulate to dynamically modulate its own level of collaborative autonomy, by restricting or expanding a received task delegation, on the basis of several context factors as the needs of other users involved in the interaction. We exploit principles underlying theory of delegation, theory of mind and BDI agent modelling, in order to build a decision making system for real-world teaming between autonomous agents.**

The model has been developed by using *JaCaMo* framework, which provides support for implementing multi-agent systems and integrates different multi-agent programming dimensions.

We tested our model in a specific domain on the humanoid robot *Nao*, widely adopted in human-robot interaction applications. The support study has established that the model provides the robot with the ability to modify its social autonomy and to handle possible collaborative conflicts due to the initiative to help the user beyond her/his request.

I. INTRODUCTION

In every-day life, humans cooperate with other humans, in order to gain knowledge, achieve and share goals, following social norms. These are sometimes encoded as laws, sometimes as expectations. A primary research topic in cognitive human-robot interaction is the design of autonomous systems that can interact and cooperate proficiently with humans. Indeed, social robots are becoming part of daily life and are present in a variety of environments, including hospitals [1], offices [2], schools [3], tourist facilities [4] and so on. In these contexts, robots have to coexist and collaborate with a wide spectrum of users not necessarily able (or willing) to adapt their interaction level to the kind requested by a machine: the users need to deal with artificial systems whose behavior must be understandable and effective. To be effective, the interaction between humans and robots should consider not only the ability of the robots but also the human preferences [5]. Robots have to maintain as much as possible a natural and intelligent interaction with humans: they should modulate their level of support interpreting both the contextual situations and the needs of the other agents involved in the cooperation [6], just like humans typically do when they interact with each other. The integration of these kinds of social skills in autonomous robots would naturally lead to a deeper relationship of trust between them and humans. Several cognitive architectures have been proposed [7], [8], [9], everyone with the goal of simulating human's cognitive and behavioral features at different levels of cognition: perception, learning, reasoning, planning, memory and so on. Along with the ability to autonomously elaborate the context information, react to the changes in the environment, make decisions about the task they are expected to carry out by showing some level of proactiveness, robots should integrate the conceptual instruments necessary to transform their autonomy into *social autonomy* [10].

A. Problem and contribution

As claimed in [11], *cooperation* implies the definition of the two complementary mental attitudes of task *delegation* and task *adoption* linking collaborating agents. Delegation and adoption are two basic cognitive ingredients of any collaboration and organization. The notion of autonomy in artificial agents, should integrate different levels of task adoption. Indeed, after receiving a task delegated from the outside, artificial agents should exploit their knowledge about the environment, including other agents are interacting with them, to adjust their own decision, for example by going beyond the delegated task, or (partially or completely) changing it, or again, adopting just a sub-part of it, because the context does not allow a complete task achievement. Theory of delegation, should guide the design of the decision making process of every robot that has to collaborate with humans in daily life.

In summary, the contribution of this research includes:

- the development of a declarative, knowledge-oriented, plan-based computational model that relies on the principles defined in the theory of delegation. The proposed approach provides a robot with an internal representation of itself and the actor involved in the interaction, every one with their own beliefs, goals, plans. In particular, the model is a decision making system where the interaction between the robot and the user is reproduced. Once a user delegates a task to the robot, it can take its decision

about the level of task adoption, on the basis of the environmental context and of the mental states attributed to the human it is interacting with. The presence, in the robot's mind of a self-representation, allows it to have a detailed description of its internal status, its technological limits and consider them in the decision process.
- A support study where the computational model has been tested on a well known robotic platform. The study has shown that the robot was able to adapt its level of collaborative autonomy in adopting a task delegated from the outside. The model has conferred to the robot the capability to go beyond the simple task acceptance and to handle possible collaborative conflicts due to the initiative to help the user beyond its request.

The paper is organized as follows: section 2 describes the theoretical models underlying our approach and the software framework used for its implementation. Section 3 focus on the description of the computational model; section 4 illustrates a support study where a real robot cooperated with humans in a specific domain; section 5 is dedicated to conclusions and future works.

II. BACKGROUND

We briefly introduce the theory beyond our computational model and the software framework used for its implementation.

A. BDI Agents

BDI agents [12] are one of the most popular models in agent theory [13]. Originally inspired by the theory of human practical reasoning developed by Michael Bratman [14], BDI model focuses on the role of intentions in reasoning and allows to characterize agents using a human-like point of view. Very briefly, in the BDI model the agent has *beliefs*, information representing what it perceives in the environment and communicates with other agents, and *desires*, mean states of the world that the agent might to accomplish. The agent deliberates on its desires and decides to commit to one of them: committed desires become *intentions*. To satisfy its intentions, it executes *plans* in the form of a course of actions or subgoals to achieve. The behaviour of the agent is thus described or predicted by what it committed to carry out. An important feature of BDI agents is the property to react to changes in their environment as soon as possible while keeping their proactive behaviour.

B. Levels of adoption about the delegated task

As mentioned above, delegation and adoption are two basic ingredients of any collaboration and organization. Typically cooperation works through the allocation of some task τ (or sub-task), by a given agent A, the *client*, to another agent B, the *contractor*, via some "request" (offer, proposal, announcement, etc.) meeting some "commitment" - bid, help, contract, adoption and so on [11]. The task τ, the object of delegation, can be referred to an action α or to its resulting goal state g. By means τ we will refer to the action/goal pair $\tau = (\alpha, g)$. For a complete theoretical overview of the delegation theory we refer to [11]. Let's focus on a deep level of cooperation, where the contractor can adopt a task delegated by the client, at different levels of effective help. In the theory of delegation, various levels of contractor's adoption are individuated:

- **Sub help**: The contractor satisfies just a sub-goal of the delegated task,
- **Literal help**: the contractor adopts exactly what has been delegated by the client,
- **Over help**: the contractor goes beyond what has been delegated by the client without changing the clients plan,
- **Critical help**: the contractor satisfies the relevant results of the requested plan/action, but modifies that plan/action,
- **Critical-Over help**: the contractor realizes an over help and in addition modifies the plan/action,
- **Critical-Sub help**: the contractor realizes a sub help and in addition modifies the plan/action,
- **Hyper-critical help**: the contractor adopts goals or interests of the client that the client itself did not take into account (at least, in that specific interaction with the contractor): by doing so, the contractor neither performs the action/plan nor satisfies the results that were delegated.

It is important to underline that we are considering collaborative robots, i.e. robots that have as their main goal the positive collaboration with the user (client).

C. JaCaMo Framework

JaCaMo [15] is a framework for multi-agent programming that integrates three different multi-agent programming levels: agent-oriented (AOP), environment-oriented (EOP) and organization-oriented programming (OOP). Every mentioned level is associated to three well-known existing platforms that have been developed for years, separately:

- ***Jason*** [16], a powerful *AgentSpeak(L)* [17] interpreter for BDI agents programming,
- **CArtAgO** [18] for programming shared environment artifacts,
- **$\mathscr{M}$oise** [19] for programming multi-agent organizations.

JaCaMo framework provides a powerful tool for implementing our computational model, in terms of:(i) the capability to represent the mental states of the real actors involved in the interaction as BDI agents;(ii) the possibility for agents of the computational model to exchange information;(iii) the possibility to implement a shared environment where can be mapped the skills of the real robot. Each of these features allowed us to reproduce the real interaction in the decision-making system of the robot. The development of our computational model has been based mainly on the first two platforms, Jason and CArtAgO. We do not exclude, in the future, to exploit $\mathscr{M}$oise in order to introduce organizational rules or constraints among the agents that populate the computational model.

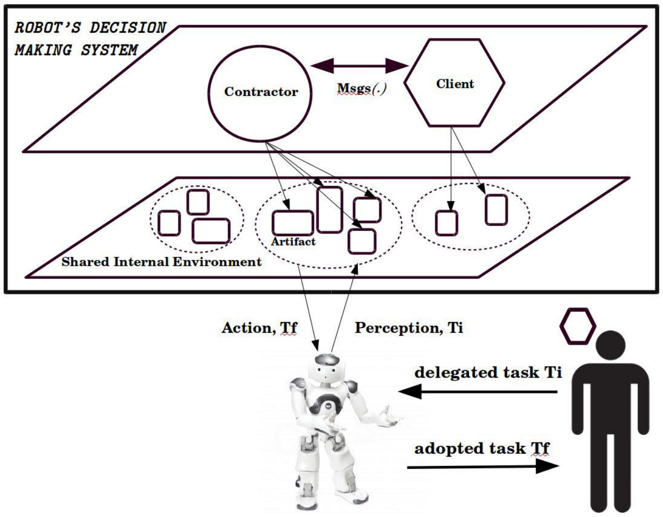

Fig. 1. Computational model overview

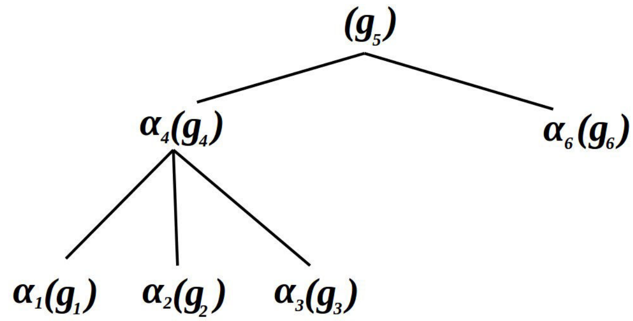

Fig. 2. composed plan example

III. DESCRIPTION OF THE COMPUTATIONAL MODEL

In this section we illustrate the conceptual ingredients of the implemented computational model. The main goal is to make an artificial agent able to autonomously adapt its level of collaborative autonomy, when it adopts a task delegated from a human client. We refer to the real artificial agent as a robot that is interacting with humans. We exploit the formalism provided by JaCaMo, in particular by Jason for the agent programming and CArtAgO for the environment programming.

When a user delegates a task τ_i to the robot, the task τ_f that the robot decides to achieve, can match with the delegated one or not. The level of τ_i adoption depends on the robot's ability to map in its decision making system:

- an high-level description of the perceived current state of the environment,
- a self-representation in terms of intentional system,
- the mental states of the other real agent involved in the interaction.

The capability of an autonomous agent to meta-represent itself and other agents and reason about their beliefs, goals, plans, intentions is known as *Theory of Mind* [20].

A. Conceptual ingredients of the model

The computational model (Figure 1) can be considered a multi-agent system which provides the robot with a theory of mind.

In particular, the model is populated by two categories of agent:

- the *Contractor*,
- the *Client*.

Agents belonging to the first category define a self-representation of the robot, with their own mental attitudes, while agents belonging to the second one, define a representation of the human clients, involved in the interaction with the robot, with their mental states. Please note that when we refer to *Client* and *Contractor*, we always indicate the mental representations, in the model, of the interacting real agents. Notice that the system can potentially be equipped by several versions of the robot itself, with different mental states. These versions could correspond to different contractor agents in the robot's decision making system. We could define, for example, a ”lazy” robot version, or a really proactive version, by giving the different descriptions of their set of cognitive ingredients. At this stage of the work we have considered just one self-representation of the robot, choosing a version in which it has the goal to provide more help than delegated every time that the contextual factors allow it.

Generally speaking, an agent's cognitive state can be described as a set of beliefs, goals and plans. A belief β is a grounded first-order logic formula encoding the information perceived from the environment, attributed to other agents, or provided by the communication with other agents. Further knowledge can be generated, in term of new beliefs, reasoning on simple beliefs through complex rules. A goal g is the state of affairs that an agent wants to achieve. An agent achieves a goal, matching to the intention it commits to pursue, by implementing a plan π, defined as part of its own plan library Π, which establishes the know-how of the agent. According to practical reasoning principles, plans are courses of actions or sub-goals the agent has to carry out before achieving the ”top-level goal”.

Formally, the plan library belonging to an agent in the computational model

$$\prod = \prod^{d} \cup \prod^{a} \tag{1}$$

is a collection of Π^d *composed plans* and Π^a *abstract plans*. Composed plans (Figure 2) represent complex hierarchical goals that decompose into other complex sub-goals g_i or actions α_i. This results in a graph representation in which edges denote plan decomposition and root nodes in the graph correspond to goals or complex actions. Typically the lowest decomposition level is formed by *elementary actions*, which, in the case of a robot, match with its elementary perception and action capabilities, for example object detection, face recognition, object grasping, moving toward a point in the

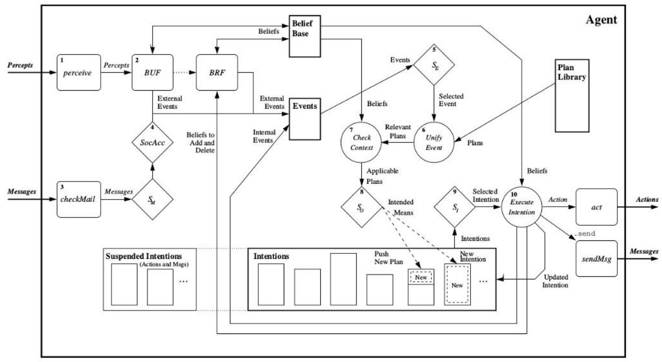

Fig. 3. Jason agent reasoning cycle [16]

space and so on. Instead, abstract plans are plans which can be specialized.

A plan for achieving g_i , can be written according to the Jason formalism:

$$+!g_i : c_i \longleftarrow b_i \tag{2}$$

An agent operates by means of its own *reasoning cycle* (Figure 3); through that, it can update its beliefs base, achieve goals by selecting plans whose context c_i are matching with the current state of the interaction, described through the beliefs. The agent acts with respect to the body b_i of the selected plan, which is the course of actions/sub-goals needed for achieving the goal g_i. The reasoning cycle can be extended and customized, for implementing a specific reasoning logic. Notice that is possible to write several *relevant plans* with the same goal to achieve, but different contexts or bodies. Relevant plans become *applicable plans*, if their context is a logical consequence of the agent's belief base.

In addition to the plans for achieving goals, an agent can trigger plans for reacting to every change in its belief base, corresponding to a change in the current state of the world. The Jason's formalism for plans used for reacting to environment changes is:

$$+!\beta_j : c_j \longleftarrow b_j \tag{3}$$

In this way an agent implements the two fundamental aspects of reactiveness and proactiveness: the agent has goals which it tries to achieve in the long term, while it can react to changes in the current state of the world. Finally, an important feature of Jason platform is the capability to integrate a speech-act based communication [21], which enables knowledge transfer between agents.

The Client and the Contractor in the computational model can exploit a shared environment, programmed in CArtAgo, which is a collection of *artifacts*. Artifacts are entities modelling services and resources for supporting agents activities. Artifacts have the main property to link the low-level control part of the robot with its high-level decision making system. Indeed, the robot is provided with its own APIs, for collecting data from sensors, and acting in the real world. APIs can be

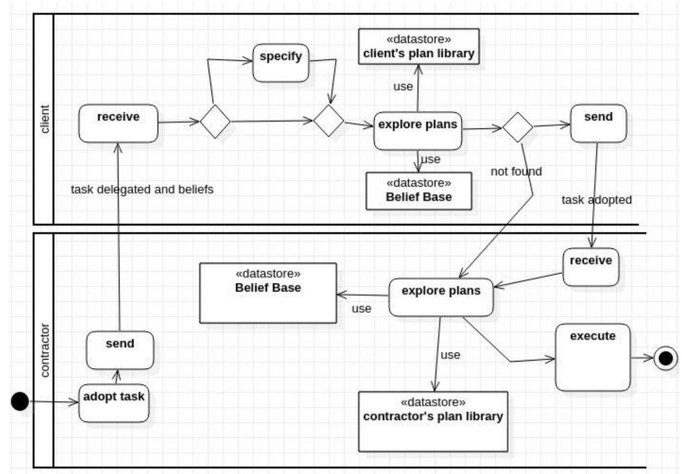

Fig. 4. Goal recognition strategy

wrapped in specific artifact's functionalities, which become an abstraction of the elementary actions the robot can perform in the real world. The contractor agent representing the robot, can exploit elementary actions to update its beliefs base or to carry out complex goals or actions. The possibility to equip the robot with a self-representation and a model of other agent involved in the interaction, is really powerful and introduces a further important feature which can lead its decision process: a human-like description of itself.

B. Decision making strategy

As analyzed above, the contractor represents a bridge between the real world and the computational model and allows the latter to have an high-level description of the perceived environment. Instead, the client has the main function to support the decision about τ_i adoption level. The client is profiled by exploiting a classical approach to User Modelling [22] which can be applied to its cognitive ingredients: beliefs, goals and plans are mapped with respect to the domain in which the robot is operating. While beliefs and goals of a client represent the mental state attributed to the user, its reasoning cycle implements a logic that makes the robot able to reason about goals of the current interlocutor. In practice, modifying the reasoning cycle means to adapt the architectural components shown in figure 3. For τ_f computation, we implemented a *context-dependent* plan recognition [23] strategy relying on:

- representing real agents of the interaction, in the robot's mind, included the robot itself,
- the capability of the agents in the computational model to share their mental states between them by speech-act communication functionalities,
- the possibility to abstract real actions in a shared simulated environment available to the agents.

Figure 4 shows the activity diagram of the strategy used by the robot to adapt its level of task adoption. Once the interaction starts and the user delegates τ_i to the robot, the first step of τ_f calculation is to activate the contractor into the computational model. This agent, with its own initial beliefs, triggers a plan for adopting the initial task τ_i

$$+!adoptTask(\tau_i, U) : true \longleftarrow send(U, \tau_i, R_{bb}). \qquad (4)$$

The contractor has the intention to adopt the task τ_i delegated by the user U. The plan's body allows the contractor to send to the agent U, τ_i and the beliefs stored in its belief base R_{bb}. At this point, the decision process is temporarily moved into the client U. The task τ_i could be completely specified by the user or the user could delegate to the robot a task in which some entity is not declared. For example he/she could delegate the goal ”put the red object on the table” or ”put an object on the table”. In this case the robot has to reason about the task specification, on the basis of the user profile represented by the client's beliefs, goals and plans. Already at this stage, the robot shows the capability to provide more help than delegated, requested by the task specification. Once τ_i is completely specified, the client agent exploits its reasoning cycle to explore the plan library in order to find at least a plan of which τ_i represents a top-level goal or a sub-goal to achieve before accomplishing a complex one. Once found, plans related to τ_i are selected. Their context is checked with respect to the current state of the world (remember that the client agent can reason about beliefs sent by the contractor agent too) and the belief attributed to the client representation. Once found an applicable plan among them, the client sends to the contractor the task τ_f, associated to the selected plan. τ_f can match with τ_i or not. This strategy allows the real robot to potentially extend its proactivity realizing an over-help, or at least a literal help. Notice that the ”action” that the client performs in the model is to send to the contractor the message carrying in τ_f. The plan for sending τ_f is:

$$+!finalTask(\tau_f) : true \longleftarrow send(Contractor, \tau_f) \qquad (5)$$

The final decision about τ_f the implementation is up to the contractor again, which tries to execute a plan. On the basis of the current state of its belief base, the contractor chooses, among the relevant plans, the one applicable to the context. The context of every plan in the contractor's library takes into account the beliefs describing the capabilities of the robot itself and its internal status. If an applicable plan exits, then τ_f becomes the final task to pursue: the selected plan can match or not with the one attributed to the client and the robot can satisfy τ_f modifying or not the plan of the user: in the first case it will implement a literal or an over help; in the second one it will implement critical or over-critical help. If the robot does not have the resources to execute the task calculated, it will execute a sub-task of τ_f, implementing a sub-help or critical sub-help. If a plan for achieving τ_f does not exist, the robot starts an interaction with the user.

In conclusion, by exploiting the plan recognition technique already described, the robot can identify possible goals/plans of the user, which do not necessarily match with the delegated task. They can be goals outstanding the delegated task, because the real agent decided it can adopt the task at a different level of help. However, there is a trade-off between pros and cons in extending the level of task adoption; possible conflicts can emerge when the robot provides less or more help than delegated. Conflicts can arise for several reasons [11]. For now, we just start from the assumption that the user appreciates the collaborative initiative of the robot, but sometimes the robot can make a mistake in classifying the user it is interacting with, because of its limited perceptive skills. As we will see in the next section, the computational model stem this limitation without losing its ability to go beyond the task delegated by the user.

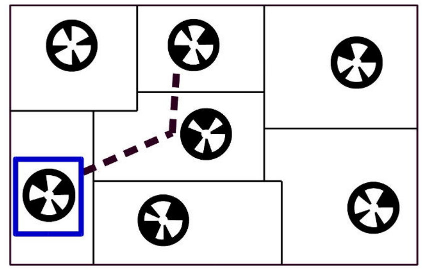

Fig. 5. Interactive map

IV. Experimental setup and application scenario

Our computational model has been tested on a well known robotic platform: the humanoid robot Nao [24]. We figured a scenario where the Nao robot serves as an ”infoPoint assistant” that could help people to get information about restaurants, museums, historical monuments to visit and nightclubs, in the city of Rome. We choose this domain for three main reasons: first of all, as mentioned in the introduction, tourism and hospitality companies have started to adopt robots and AI services in the form of chatbots, robot-concierge, self-service information/check-in/check-out systems and so on; second, this domain allowed us to make experiments with a real robot by overcoming the technological limitations related to the robotic platform (grasping issues, navigation issues). Furthermore, robot as touristic assistant can figure several possible scenarios, of which providing information is only a part.

Through the use of a simple interactive map (figure 5), the robot shows to the user where the requested point of interest (POI) is placed and indicates the path to the destination. It suggests the less busy way (dashed path), starting from the infoPoint (marked landmark) to the POI. The map is partitioned in zones, encoded by landmarks that Nao can easily recognize and associate to integers (e.g. 68, 80, 107). Every point of interest is associated to a particular area of the city populated by restaurants, museums and so on. The map is interfaced to a specific artifact exploited by the contractor agent to make it accessible. POIs are described in the belief base of the contractor through expressive annotations. For instance, to a restaurant can be associated a tuple of the form *restaurant(name, category, location, capacity, target, state)*, where *category* describes the restaurant's typology, *state* indicates if it's open or closed, *target* the audience

	τ_f if $0.7 \leq Acc_s \leq 1.0$
Q1	$enjoyTheCity : c_1 \longleftarrow findRestaurant(laSoraLella, 68, Typical); findPlaceToVisit(araPacis, 68, historical).$
Q2	$enjoyTheCity : c_1 \longleftarrow findRestaurant(laParolaccia, 68, Typical); findPlaceToVisit(araPacis, 68, historical).$

	τ_f if $0.0 \leq Acc_s < 0.4$
Q1	$enjoyTheCity : c_1 \longleftarrow findRestaurant(laSoraLella, 68, Typical); findPlaceToVisit(SantaCecilia, 68, church).$
Q2	$enjoyTheCity : c_3 \longleftarrow findRestaurant(AngoloDelVino, 68, Typical); findMuseumToVisit(MuseoDal, 68, art).$

	τ_f if $0.4 \leq Acc_s < 0.7$
Q1	$enjoyTheCity : c_1 \longleftarrow findRestaurant(laSoraLella, 68, Typical); findPlaceToVisit(piazzaTrilussa, 68, square).$
Q2	$enjoyTheCity : c_1 \longleftarrow findRestaurant(Otello, 68, Typical); findPlaceToVisit(piazzaTrilussa, 68, square).$

TABLE I
TASK ADOPTION RESULTS

for whom it is addressed (e.g. singles, couples, groups) and *capacity* if it is small, big or medium.

The robot can interact with different kind of users: for instance, it can give information to tourists and citizens. Since our goal is to demonstrate the flexibility of the computational model, without loss of generality we leverage on a simplified user encoding, based on colors and numbers. Tourists are encoded with a red shirt and citizens with a green one. Moreover, people can have different mental states, depending on their characteristics and attributes, i.e. the age, the marital status and so on. In our case study we exploited the marital status in order to classify the interlocutor as i) single, ii) in couple, iii) with family and iv) in group. The marital status is represented by a number on the shirt: 1 for singles, 2 for couples, 3 for families and 4 for groups. In conclusion, the robot can perceive the user as, for instance, a single citizen, or a tourist on holiday with his own family. The robot can make mistake in perceiving the user. For mapping this perceptive process into the model, two beliefs, in the contractor agent, are updated when the robot detects the user:

$$userCategory(U_c, Acc_c) \text{ and } maritalStatus(S, Acc_s)$$

The first one indicates if the user is a tourist or a citizen, the second one indicates its marital status. The robot classifies the user's attributes with a certain *accuracy*, expressed by Acc_c and Acc_s. We conducted a test in which the robot could interact with tourists or citizens with different marital status. Hereinafter we describe the scenario where the robot interacts with a tourist who is single and asks it to achieve the result to find a restaurant. Moreover, we took in consideration the case where the robot was able to correctly recognize the user as a tourist, but it could classify its marital status at different levels of accuracy Acc_s. The user asks to the robot:

- Q_1: "I would like to go to La Sora Lella restaurant"
- Q_2: "I would like to go to eat something in Trastevere"

Questions imply two different τ_i delegation:

- Q_1: $findRestaurant(\text{"}LaSoraLella\text{"}, 68, \text{"}Typical\text{"})$
- Q_2: $findRestaurant(\text{""}, 68, \text{""})$

In the plan library of the agent representing the real user, a plan π_1 is present which has the result to enjoy the city, eating in a restaurant and visiting a monument:

$$\begin{aligned}\pi_1 :& enjoyTheCity : c_1 \longleftarrow \\ & findRestaurant(Name, Location, Category); \\ & findPlaceToVisit(Name, Location, Category).\end{aligned}$$

This means that the robot attributes this plan to the user and maps it in the client agent. Notice that, in the client's plan library, can be attributed several plans with the same goal of enjoying the city, but different contexts and bodies. Last, the robot choses the relevant plan to execute as depicted in section III.

Table 1 shows the level of τ_i adoption related to the situation described above. In all cases where the delegation is univocal (Q_1), the robot can go beyond the delegation, without changing the client's plan (over-help). When the delegation is vague (Q_2) the robot is still able to extend its help: indeed, it can use the few task specifications in order to find a restaurant which better adapts to the user, by considering the accuracy which it has been classified. For example, when $0.0 \leq Acc_s < 0.4$ the robot exploits the "stereotype" of a tourist representation in its decision making system and chooses a typical restaurant (typically a tourist wants eat in typical restaurants) targeted for couples instead of single people. Vice versa it chooses a restaurant targeted for singles when it is almost sure that the user is effectively single ($0.7 \leq Acc_s \leq 1.0$). Finally, when it cannot distinguish singles from couples, it chooses a restaurant suitable for a generic target audience. Notice that, when the robot does not find any monument to visit, it still does more than delegated, by finding a museum to visit, instead of a monument: it realizes an over help and in addition it modifies the plan attributed to the user (over-critical help).

V. CONCLUSIONS AND FUTURE WORKS

In this paper we presented a cognitive model which integrates the concept of adjustable social autonomy as a basis for an effective human-robot interaction. Exploiting the notions of task delegation, adoption and the theory of mind, the computational model has proven to be really adaptive and flexible, giving to the robot the capability to adjust its level of help on the basis of several dimensions of the cooperation. The computational model is knowledge-dependent, but domain-independent: the agent's mental state can be extended, in order to make it applicable across a number of domains and real situations.

Since the computational model can be exploited in order to build robots that have as their main goal the positive collaboration with the user, the next step of our work will be to introduce the concept of trust in the model. The notion of trust is strictly related to delegation. More precisely, delegation is the result of a complex mental state, described as a set of beliefs, goals and decisions: in one word, trust. A possible strategy to integrate trust in the computational model could be exploit the third multi-agent programming dimension, the organizational one, in order to define a set of behavioral constraints that the agent belonging to the computational model adopts when they reproduce the real interaction. Moreover, considering that specifying plans in the representation of the real actor can be a limit, we aim at introducing of a more dynamic approach for plan selection, more adapt to complex and uncertain real scenarios. Finally we aim at introducing some form of learning in order to improve the ability of the robot to reason about other agent's behaviors, goals, beliefs and decide what level of task adoption it will be necessary and more adapt to entire context of the cooperation.

REFERENCES

[1] D. Casey, O. Beyan, K. Murphy, and H. Felzmann, "Robot-assisted care for elderly with dementia: is there a potential for genuine end-user empowerment?" *The Emerging Policy and Ethics of Human Robot Interaction*, 2015.

[2] M. M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal, "Cobots: Robust symbiotic autonomous mobile service robots." p. 4423, 2015.

[3] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and F. Tanaka, "Social robots for education: A review," *Science Robotics*, vol. 3, no. 21, p. eaat5954, 2018.

[4] S. Ivanov, C. Webster, and K. Berezina, "Adoption of robots and service automation by tourism and hospitality companies," *Revista Turismo & Desenvolvimento*, vol. 27, no. 28, pp. 1501–1517, 2017.

[5] B. Lubars and C. Tan, "Ask not what ai can do, but what ai should do: Towards a framework of task delegability," *arXiv preprint arXiv:1902.03245*, 2019.

[6] S. Rossi, F. Ferland, and A. Tapus, "User profiling and behavioral adaptation for hri: a survey," *Pattern Recognition Letters*, vol. 99, pp. 3–12, 2017.

[14] M. Bratman, *Intention, plans, and practical reason*. Harvard University Press Cambridge, MA, 1987, vol. 10.

[7] L. Kajdocsi and C. R. Pozna, "Review of the most successfully used cognitive architectures in robotics and a proposal for a new model of knowledge acquisition," in *2014 IEEE 12th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2014, pp. 239–244.

[8] P. Ye, T. Wang, and F.-Y. Wang, "A survey of cognitive architectures in the past 20 years," *IEEE transactions on cybernetics*, no. 99, pp. 1–11, 2018.

[9] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, and R. Alami, "Artificial cognition for social human–robot interaction: An implementation," *Artificial Intelligence*, vol. 247, pp. 45–69, 2017.

[10] R. Falcone and C. Castelfranchi, "The human in the loop of a delegated agent: The theory of adjustable social autonomy," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 31, no. 5, pp. 406–418, 2001.

[11] C. Castelfranchi and R. Falcone, "Towards a theory of delegation for agent-based systems," *Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp. 141–157, 1998.

[12] A. S. Rao, M. P. Georgeff, *et al.*, "Bdi agents: from theory to practice." in *ICMAS*, vol. 95, 1995, pp. 312–319.

[13] M. Wooldridge and N. R. Jennings, "Agent theories, architectures, and languages: a survey," in *International Workshop on Agent Theories, Architectures, and Languages*. Springer, 1994, pp. 1–39.

[15] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi, "Multi-agent oriented programming with jacamo," *Science of Computer Programming*, vol. 78, no. 6, pp. 747–761, 2013.

[16] R. H. Bordini and J. F. Hübner, "Bdi agent programming in agentspeak using jason," in *International Workshop on Computational Logic in Multi-Agent Systems*. Springer, 2005, pp. 143–164.

[17] A. S. Rao, "Agentspeak (l): Bdi agents speak out in a logical computable language," in *European workshop on modelling autonomous agents in a multi-agent world*. Springer, 1996, pp. 42–55.

[18] A. Ricci, M. Piunti, M. Viroli, and A. Omicini, "Environment programming in cartago," in *Multi-agent programming*. Springer, 2009, pp. 259–288.

[19] J. F. Hübner, J. S. Sichman, and O. Boissier, "Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels," *International Journal of Agent-Oriented Software Engineering*, vol. 1, no. 3/4, pp. 370–395, 2007.

[20] D. Premack and G. Woodruff, "Does the chimpanzee have a theory of mind?" *Behavioral and brain sciences*, vol. 1, no. 4, pp. 515–526, 1978.

[21] J. R. Searle, *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969, vol. 626.

[22] E. Rich, "User modeling via stereotypes," *Cognitive science*, vol. 3, no. 4, pp. 329–354, 1979.

[23] S. Carberry, "Techniques for plan recognition," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 31–48, 2001.

[24] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "The nao humanoid: a combination of performance and affordability," *CoRR abs/0807.3223*, 2008.