The cognitive architecture of a robotic salesman

Adrián Romero-Garcés[1], Luis Vicente Calderita[2], Jesús Martínez-Gómez[3], Juan Pedro Bandera[1], Rebeca Marfil[1], Luis J. Manso[2], Pablo Bustos[2], and Antonio Bandera[1]

[1] University of Málaga, Spain, 29071
argarces@uma.es
[2] University of Extremadura, Spain, 10003
[3] University of Castilla-La Mancha, Spain, 02071

Abstract. This paper describes a robotics cognitive architecture for social robots named CORTEX. This architecture integrates different levels of abstraction (from basic geometry to high-level predicates) into a unique Deep Space Representation (DSR) that different agents interface. These agents update the contents of the DSR with new data from the outer world, and execute, plan and design behaviours. The design of CORTEX as an unified deep representation allows to fit both the subsymbolic processing and flexibility requirements of robot control. In this paper a first implementation of CORTEX has been integrated into *Gualzru*, a robotic salesman, and tested in real scenarios. Results show that this cognitive architecture allows this robot to adequately execute its use case, and that it has a promising adaptability to achieve new tasks and be used in new scenarios.

Keywords: robotics, cognitive architectures, social robots

1 Introduction

Robotic architectures have improved steadily over the years, evolving as a conglomerate of modules that let the robot negotiate increasingly complex indoor and outdoor environments. As a result, current robot architectures integrate multiple sophisticated algorithms for real-time perceptual, planning and action processing, including 3D object and space recognition, simultaneous localization and mapping, navigation and task planning, manipulation and grasping, action monitoring, human detection and tracking, recognition of human activities, human-robot dialoguing or action monitoring.

Cognitive architectures, on the other side, have centered on general-purpose reasoning and problem solving, and on modeling and measuring human performance in those tasks. Most of them focus on higher-level cognitive processes in non-robotics domains, although there is an increasing number of exceptions [19] [31] [13]. Recent reviews of current cognitive architectures can be found in [20] [7] [40] [1] [5] [18] [42].

It is clear now that both can benefit from each other. For example, cognitive methods would be useful in HRI scenarios where the robot has to use a model of the human to improve its planning decisions, in collaborative problem solving where a human is in the loop, in situations where the existing domain knowledge is not enough and a common sense module using generic knowledge might bring in the needed solution, or to introduce symbolic learning techniques to expand and improve the existing knowledge base. A interesting discussion involving a kitchen-robot cognitive scenario can be found in [2]. Robotic architectures would complement cognitive ones by providing the connection to the real world through continuous, parallel and asynchronous processes, acquisition of new objects, sounds, words, actions and skills, or learning what actions can be applied to them [36].

Robotics cognitive architectures are more recent creatures[4], at least from the point of view of real robots implementing them in complex scenarios [11] [17] [32] [21] [4] [38]. Social and service robotics are pushing even further the need for an integration between cognitive and robotics architectures. These robots have to interact with humans and interpret our world in a similar fashion in order for both to share meanings, common references, attention, context dependent references, etc. Even more, social robots have to maintain a model of the human it is interacting with, and use that model to obtain better plans to act. Thus, the need for integrating both worlds is urgent and many benefits are expected from it. However, there are still many problems, inside each field and in the combination of both. One way to advance in this integration problem is to identify basic features that have to be met at the *interface*. From the side of the cognitive architectures this interface could be initially reduced to four requirements. Quoting Scheutz et al. [37],

> "...there are at least four important commonalities or core commitments among (major) cognitive architectures: (1) percepts are given as discrete complex data types, often corresponding to or representing objects in the environment together with their properties; (2) goals are explicitly represented; (3) behaviors are represented by discrete actions with parameters; and (4) processing occurs in cognitive cycles in which percepts are processed first, followed by internal processing, and subsequent (external or internal) action selection."

This combination of discreteness and sequentiality collides frontally with the parallel, continuous, time-critical processes used in robotics architectures, and that is precisely one of the first challenges for cognitive robotics. In this paper we will describe ongoing work on a new agent-based robotics cognitive architecture, named CORTEX, that is designed to satisfy the four conditions above and also to contribute with two key ideas. The first one is the use of a dynamic graph representing the robot and its environment, that is shared among the agents and codes geometric and symbolic information. The second one is the pervasive use of internal emulation to *imagine* potential curses of action of the robot and its environment [15] [6] [16].

[4] According to [22] the term was first introduced by R. Reiter in 1993.

Agent-based architectures in robotics and AI have been used for a long time [26][23][33] and are common nowadays in cognitive robotics [12] [35] [43] [3] [19]. We use the term *agent* here as and autonomous software module in charge of a well defined functionality, reactive, deliberative of hybrid, that interacts with other agents to enact the complete system. From the point of view of the implementation, the designer can map simple agents to components in the underlying programming framework, and to groups of interconnected components when they endow a more complex function.

The choice of high-level agents, the most general functionalities that comprise the architecture, is an open and debated issue directly related to the current models of intelligence. Even from the more restricted point of view of a service robot that has to operate in a constrained but real-world domain, it looks like that any choice of loosely coupled high-level functions is dependent on the defining domain. If the domain changes, one immediate effect is that the coupling among agents tightens, since new information has to travel among them. For example, given two *Navigation (N)* and *PersonDetector (P)* agents being part of a robotic architecture, we can have N to localize the robot in a map, plan paths to target locations and navigate those paths avoiding dynamic obstacles, and P to detect people, extract an articulated outline of their silhouette, track it, incorporate it as an instance of a generic model of a person and create beliefs about her basic intentions. These two high-level agents work without communicating -loosely coupling- and provide updated symbols to a deliberative agent, which will use them to reason on the mission at hand, and send commands back to the agents. But if we want N to react differently whether the obstacle is an inanimate object or a person, so N can elicit specific human-aware avoiding behaviors, then they have to communicate. One solution is to let the deliberative agent cope with this by extending its set of rules. The other one is to let both agents share a common context and reprogram N to do human-aware navigation when the person steps in. The first solution is the *conscious* one, where the new skill has to be activated by the deliberative agent. The second one would be an *unconscious* skill triggered automatically by the recognition of the situation ahead. Note that , if properly programmed, this new behavior could also be inhibited by the deliberative agent. We want our architecture to allow for both possibilities, by manual programming now, and by automatic learning in the near future.

Our first step in this direction is the definition of an graph-based internal representation of the robot and its environment shared by the agents. This approach in cognitive robotics is not new and the graph is formally equivalent to other knowledge representations like productions, schemes or frames. In fact, the graph can be translated online to PDDL so deliberative agents can read it. The main challenges posed by this shared object are, (i) the coherent combination of geometric and symbolic information in one structure; (ii) the policy followed by agents to edit the graph; and (iii) the mechanism to efficiently republish and resynchronize the changing graph among them. These issues will be addressed in Section 2.

The second design choice in CORTEX takes us to a less paved path in the arena of robotic cognitive theories. The basic idea has been proposed and defended as the Simulation Theory of Cognition [16][15] [15] [9]. This theory advocates the use of an internal model to represent reality, and to anticipate and simulate the outcome of emulated actions. The first step towards a simulation centered architecture is the existence of an internal model rich enough to compute the outcome of actions at different levels of time and space resolution. Note that this time and space dimensions apply also to different levels of abstraction, from the kinematic model of the robot, to the symbolic definition of objects and actions. This idea of the brain continuously exploring future courses of action is widely accepted in cognitive Neuroscience, and is applied at many abstraction levels: from the short timestamp of a spinal cord reflex, to the anticipatory perception of a distant approaching car using a priori information recovered from episodic memory in combination with low frequency visual and auditive indicators. Also, the mirror neuron theory is closely related to the notion of emulation [28], as well as the theory about the theory of mind humans might have about others [14]. Emulation in CORTEX is performed with different techniques depending on the level of abstraction. Current work in this line is considering three levels, (i) basic control loops as differential equations, (ii) geometric simulation using graph-scenes, and (iii) symbolic search using planning algorithms. These three levels will share, nevertheless, the graph representation so all agents can benefit of the results.

In the rest of the paper we describe in more detail CORTEX and a recent implementation of it the salesman robot Gualzru, built for a consortium of Spanish companies working in the advertisement and technological sectors.

2 From RoboCog to CORTEX

Our previous proposal for a robotics cognitive architecture, RoboCog [30][25], followed the schema of a three-tier architecture [8] with a layer behaviour generation agents, an executive agent in charge of monitoring the current plan and a deliberative agent wrapped around the planning and learning architecture PELEA[10]. RoboCog, as well as CORTEX, uses RoboComp [24] as the underlying component-oriented programming framework.

In RoboCog, the behaviour agents and the deliberative layer communicate through two shared data objects, a kinematic tree representing the geometric short-term state of the robot and the environment -*InnerModel*, and a graph maintaining a symbolic representation of the robot, its environment and the current plan -*AGM*. Both data objects are required for flexible, re-targetable robot architectures [35]. The two objects are complementary, and together represent the robot belief about itself and the world. The functioning of the architecture can be easily explained if we picture it as a large dynamical system. Starting in a quasi-stationary state, the perceptual modules try to keep the internal representation synchronized with the world. But when a new mission is requested, a plan is generated and injected into the symbolic graph. This alteration creates a disequilibrium to which the whole system reacts trying to restore the initial balance. This view resembles the idea of the difference-engine as proposed by Minsky [27]

However, the division into symbolic and geometric levels, although initially quite useful, turned out to be a major obstacle in the evolution of RoboCog towards handling more demanding social robotics scenarios. The main reasons are (i) the complexity to update and access both representations separately; (ii) the difficulty in maintaining a tight division between both abstraction levels; (iii) the difficulty in maintaining a coherent common state when both data objects are accessed separately; (iv) the difficulty in achieving a highly efficient access to data when two or more accesses have to be made for each query; and (v) the difficulty in introducing new concepts or new attributes to existing concepts when they must exist in both graphs;

As a response to the problems found in RoboCog, we started the design of CORTEX, the robotics cognitive architecture described in this paper. The core of this architecture is a new, integrated, dynamic multi-graph object that can hold several levels of abstraction, from basic geometry and sensorial state to high-level symbols and predicates describing the state of the robot and the environment during a short time lapse. This object is called Deep State Representation (DSR).

DSR's underlying structure is a multi-graph in which edges connecting nodes can have several labels. In its current implementation the graph is read from disk when the network of components that conform the robot software is deployed, an updated later as the robot evolves in its environment. The graph contains a symbolic description of the domain, where nodes hold attributes and edges hold predicates. The initial value of these attributes might be undefined and are updated during robot execution by the agents, both perceptive and deliberative. Embedded in this graph, DSR hold a metric, kinematics description of the robot and of its near environment. Both representations, together, define the current knowledge of the robot about itself and its surroundings. All agents are automatically subscribed to this graph and receive updates whenever something changes, through the underlying communications middleware provided by RoboComp. Perceptual agents can edit the graph by changing the truth value of symbolic predicates, or by updating the numerical value of metric attributes. This actions (*non-structural changes*) can be done without external permissions and is the responsibility of the agent to inject coherent values. On the other hand, *structural changes* in the graph can also be done but require an explicit validation by the *Executive* agent. This validation is essentially a model checking reasoning step that has to be done in order to guarantee the integrity and coherence of the global representation. Finally, when the deliberative agents, PELEA, accept and initiate a new mission, and a plan is obtained, this plan is written to the graph as sequence of *desired* action/states pairs which are recognizable by the action generation agents. These agents take this action commands as internal goals and try to satisfy them.

As was advanced in Section 1, through this graph perceptual agents can transmit a discrete, symbolic state of its perceptive and motor activities to their deliberative companions, can accept symbolic description of local goals and the planning and monitoring agents can proceed in cognitive cycles as the plan is executed and the world representation updated. Furthermore, the DSR graph is a shared global state among the agents that facilitates the coding of new skills that depend on cross information among agents. Figure 2 shows the organization of modules in the CORTEX architecture. It can be depicted as a cylindrical surface split in several vertical slices, the agents, which are the old modules now standing at equal positions. In the center of the cylinder, DSR holds the current internal belief of the robot, affecting and being affected by all the modules, and always struggling for an equilibrium that the reality fades away.

The benefits of CORTEX and its unified representation can be highlighted in the following example: a robot that should interact with people using two different channels, speech and a touch screen. A software component is required to deal with each of these channels, but they affect the same symbolic data (phrases perceived or transmitted). Changes in these data affect the execution of the use case, so the planner has to be involved. Fig. 1 shows how RoboCog and CORTEX deal with the reception of a new speech response. In RoboCog, it is necessary to communicate the change to the Executive, so it can update the AGM (symbolic representation). Once the AGM is updated it has to update the geometric representation, and it has also to inform both the Conversational and the TouchScreen agents (but after both representations have been correctly synchronized). If the TouchScreen agent receives different data while these changes are propagated (red zone in Fig. 1.a), it may send different updates to the Executive and loose synchronization with the rest of the architecture. Thus, additional inter-agent connections are usually required (green arrows) to reduce this possibility. On the other hand, CORTEX can use non-structural changes to allow both agents synchronize through its subscription to the DSR, while the Executive itself can also update the execution of the use case simply by checking this inner representation (Fig. 1.b). As depicted, while both architectures obtain the same results, CORTEX allows for more efficient and robust solutions.

An initial implementation of CORTEX has been tested under the Spanish ADAPTA project, which was formed in 2012 as an industrial consortium to explore new technologies in personalized, dynamic advertising in public spaces. Among these technologies, social robotics appears as one of the most attractive solutions. To explore this possibility, a new robot was designed and built for the project. *Gualzru*, acts as a salesman that moves in a large shopping area and tries to convince potential clients to follow it to an interactive advertising panel. The external appearance of *Gualzru* has been carefully chosen for its purpose (see Fig. 4.a). Its sensors, actuators and algorithms are also adequate to safely navigate through daily life environments, and its behaviour is controlled by the new CORTEX architecture. In this paper we will use this implementation as an example of the characteristics of CORTEX and will evaluate its performance.

In the next Sections, some additional details of the implementation of CORTEX for the ADAPTA project are described, the use case that was the target of

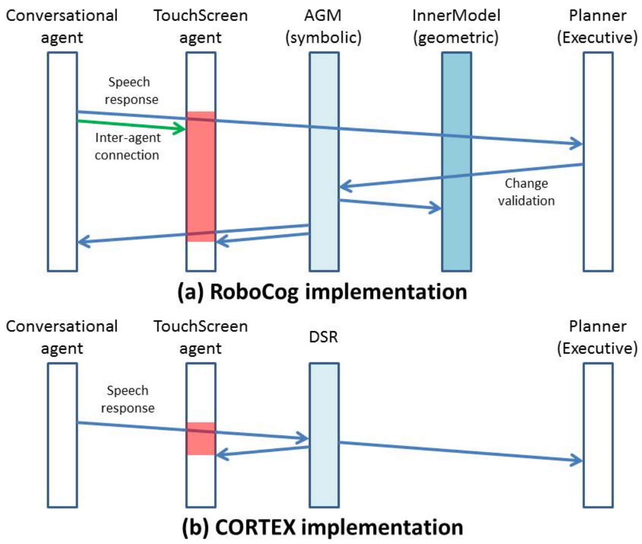

Fig. 1. Using speech and a touch screen to interface the robot. An implementation example: (a) RoboCog solution; (b) CORTEX solution.

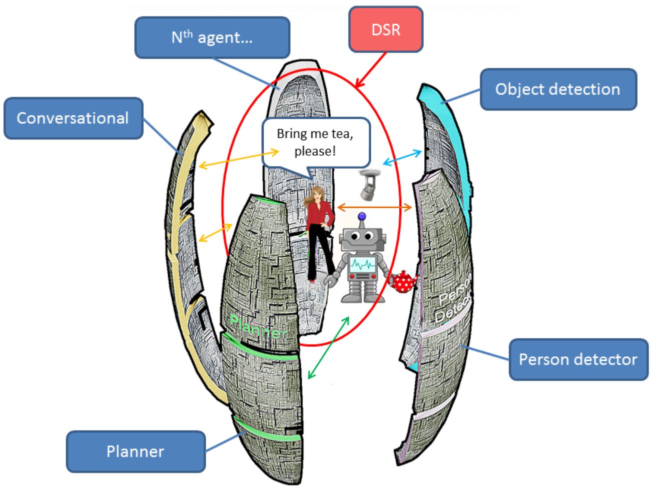

Fig. 2. Overview of the CORTEX architecture depicted as quasi-cylindrical shape sliced in agents with the shared representation occupying the central volume.

the project and some experimental results, including user interviews about the HRI personal experiences.

3 Agents in Gualzru

This section describes the different agents that have been used for the CORTEX implementation integrated in *Gualzru* (Fig. 3).

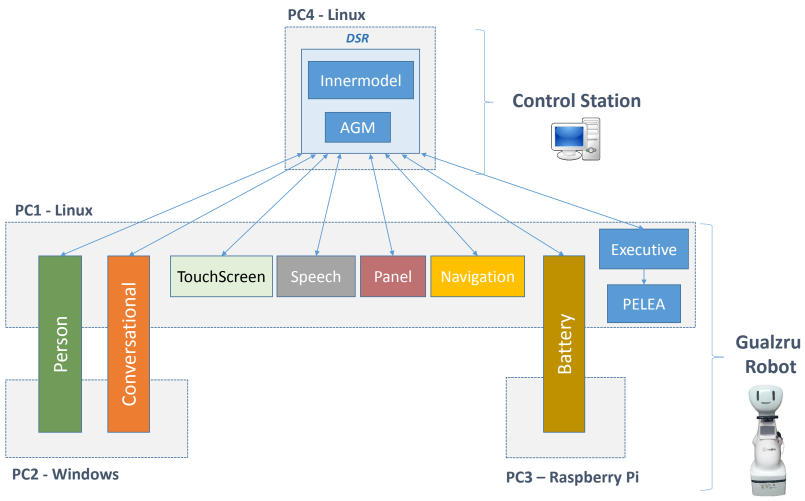

Fig. 3. Overview of the agents used in Gualzru

3.1 Person

The agent Person is responsible for detecting and tracking the potential customer during the execution of the whole process. Internally, the agent is formed by several components. One of them runs a Microsoft's Kinect based program in a separated embedded computer, making use of the multi-language, multi-OS functionality of RoboComp. The agent also detects the face of the person when she is positioned close enough and classifies it according to gender and age. To this end, the component uses Local Binary Patters [39] as features to feed a cascade of Support Vector Machine classifiers [34].

When the agent detects a new person, it injects a set of nodes in the DSR graph coding her position, gender and age. When the robot approaches the person and the body detector is no longer able to track the silhouette, the face detector and tracking components of the agent continue updating the represented person in the DSR. This way, we are solving several interesting problems in this specific situation, the (re)identification of an world element, the part-to-whole association problem caused by limited visibility of sensor, the context problem since the person is automatically available to other agents and the grounding problem since the person is being tracked and (re)identified continuously and assigned to the same high-level symbol.

3.2 Conversational

The Conversational agent has the goal of getting an affirmative answer from the user to the proposal of going to the panel. While this goal is reached, the robot should detect when the user is requesting some information from it. If (a) this information is related to the advertisement, and (b) the robot has some knowledge about the requested information, it would provide the user with such information.

Each phrase from the user involves a two stages process: transcription and comprehension. The transcription is carried out in the Windows embedded computer (PC2, see Fig. 3) taking advantage of the Microsoft Kinect Speech SDK. The language model is generated though a n-grams procedure [46], using online corpus as COLA [45]. Relying on a Bag-of-Words representation of the transcription of the input utterances, the agent categorizes each user sentence by using a Naive Bayes classifier trained from real scenarios. The most probable sentence category (e.g. question about distance to the panel) will determine the phrase of the robot.

3.3 Speech

The Speech agent is in charge of generating the synthesized speech. It reads the phrases to say from a symbolic attribute attached to the robot geometric node in the DSR. This attribute must have been previously updated by the Conversational agent. The open-source Festival [41] solution is employed as text to speech tool.

3.4 TouchScreen

The TouchScreen agent was introduced as a means to reinforce the speech recognition system. Due to limitations in current sound recording and ARS technology in ope, reverberating or crowded open spaces, a complementary mediun of HR communication was needed. This agent, (i) reads the robot phrases determined by the Planner from the DSR and shows them in the touch screen located on the chest of Gualzru, and (ii) users can answer the robot by touching the screen. The functionality has proven very useful in noisy environments where it is difficult to hold a conversation.

3.5 Panel

The Panel agent is in charge of selecting an adequate product for the client, according to her estimated gender and age. It obtains its information from a local database of products indexed by people features that can be updated by the owner of the global advertisement system.

3.6 Battery

This agent is in charge of monitoring the battery level of *Gualzru*. If the battery level goes below a certain safety threshold, the state of the robot is accordingly updated in the DSR.

3.7 High-level Planner: PELEA and the Executive

This deliberative agent, composed by two components, is the responsible for the high-level planning of the whole system, integrating planning and re-planning, monitoring and learning abilities. The core of this agent is the PELEA framework that is explained in detail here [10]. It is composed by the following sub-modules:

Planning and Learning

- An Execution module in charge of the interaction between PELEA and the environment.
- A Monitoring module that checks the plan progress during its execution.
- A Decision Support module that produces new plans using the high-level planner. A new plan is elaborated when it receives a problem from the Monitoring Module (planning) or when the Monitoring module indicates a discrepancy between the observed and expected states (re-planning).
- A Learning module that infers knowledge from the experience gathered by the high-level planner during the plan execution. This knowledge is employed to increase the efficiency of the PELEA agent in future executions of the use case.

Executive The second module of the High-level Planner agent is the Executive, that acts as a bridge between the DSR and the PELEA framework. It is has two main goals. On one hand, the Executive is responsible for coordinating the action of the plan and it is aware about changes in the DSR. Thus, when a module produces a change for the DSR (e.g. when a person is detected or lost), the Executive reacts trying to keep the inner representation coherent. Using the domain knowledge, the second goal of the Executive is to check if this new model of the representation is valid. This validation occurs only when it implies a structural change in the DSR. If the validation is accepted, the whole DSR changes according to the new proposal.

3.8 Navigation

The Navigation agent is in charge of the robot motion and localization in the operations area. For localization, a specialized component uses a small set of AR tags, AprilTags [29], distributed at strategic places. This component computes the required geometric transformations to update the position of the robot in the DSR, which is then propagated to the rest of the agents. The navigation component uses an evolution of the R-ORM [47] algorithm, a reactive method that allows the robot to reach a target avoiding possible obstacles. This modified algorithm is able to divide the problem of reaching a certain objective into a set of sub-problems, in which sub-objectives are set, updated and reached sequentially.

4 Gualzru

Fig. 4 depicts the final version of the *Gualzru* robot that has been developed for the ADAPTA project. Most elements in this robot are similar to the ones of the prototype detailed in Romero-Garcés et al. [34]. However, several modifications have been addressed for this final version of *Gualzru*.

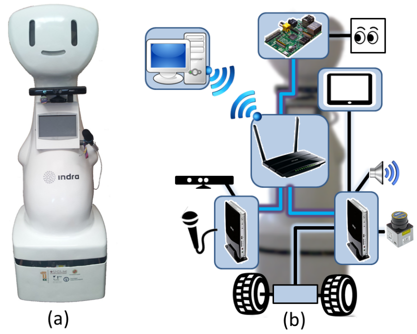

Fig. 4. (a) Gualzru, the robotic salesman; and (b) Hardware structure

Fig. 4.a shows the external appearance of the robot. The tactile screen and the speakers have been replaced by new devices and the cover has been polished and painted. Fig. 4.b presents the internal hardware structure where a router-switch connecting the four main elements is shown:

– An embedded computer that runs the core of CORTEX. This computer is connected to (i) the laser range finder used for navigation, (ii) to the motor controllers that move the robot, and (iii) to the speakers and to the tactile screen installed in the chest of the robot. This screen will be used to display the phrases uttered by the robot and to provide an alternative to the audio channel to communicate with the robot.

- An embedded computer connected to the Kinect® sensor, and to the microphone used to capture the voice of the potential client. Some components of the Conversational and the Person agents run in this computer.
- A Raspberry Pi® device, that is in charge of controlling the eye motion of *Gualzru* and monitoring its battery level.
- An additional external computer. It is in charge of showing the internal representation and the state of the plan. It looks like a "control panel" since It can run interface modules, that allow a technician to monitor the robot and the data stored in CORTEX, and to take direct control over Gualzru if required (e.g. dangerous situation or incorrect behaviour).

5 Experiments: testing the cognitive architecture in real use cases

The proposed CORTEX architecture has been tested in *Gualzru* working in several real daily-life scenarios. In this Section we first describe the use case and then discuss the results obtained.

5.1 The ADAPTA use case

A deep description of the use case that determines the behaviour of the robot can be addressed in previous contributions [34]. Basically, *Gualzru* waits by an advertising panel and targets any person in its surroundings. The robot will move towards the person in a short displacement (3-4 meters maximum), that finishes when the robot reaches the person at a social distance (1.5 meters). Then, the robot introduces itself while classifying the person into a group (using gender and age parameters). Once the classification process finishes, the robot chooses a Product Topic to offer and tries to convince the person to follow it to the panel. The robot is also able to answer certain questions about several topics: its name and condition, location of the panel, requested service time, price of the service and extended information requirement. In addition to these topics, it also detects when the user accepts or rejects its invitation, using voice detection or a touch screen.

The robot can withdraw from the interaction, once started, due to two causes: (i) the person shows no interest in the robot; and (ii) the person rejects its invitation to approach the advertising panel. On the other hand, if the person agrees on going to the panel area, *Gualzru* accompanies her to the destination.

5.2 Analysis of the results

Fig. 5 shows the robot while executing the use case in real scenarios. In these experiments, thirty-three people who interacted with the robot were asked to fill a questionnaire after the experience. The questionnaire is designed as a Likert scale, although it uses six levels, from 0 to 5, to remove the neutral option (middle point). Table 1 shows the results for the questions related to the performance of

Table 1. Questionnaire results (33 tests)

Question	$\bar{x}$	σ	$\bar{x}_{prev}$	σ_{prev}
2.1 Have you understood what the robot told you?	4.27	1.23	3.57	1.28
2.2 Do you think the robot understood you?	3.72	1.28	2.7	1.37
3.1 Did the robot get blocked?	1.42	1.71	1.39	1.72
3.2 Was the interaction natural?	3.06	1.22	3.1	1.11
3.3 Was the conversation fluent?	3.06	1.32	2.85	1.22
3.4 Did the robot seem to be tele-operated?	1.27	1.48	0.87	1.44
3.5 Was the touch screen useful for the interaction?	4	1.58	-	-
4.1 Did you enjoy the experiment?	4.63	0.54	4.31	0.88
4.3 Would you like to repeat?	4.51	0.75	4.28	1.32
4.4 Would you recommend it to other people?	4.69	0.58	4.52	0.86

the cognitive architecture. See [34] for a complete description of the main test scenario, questionnaire procedure and the results obtained without using the CORTEX architecture and the touch screen panel. As depicted in the table, the subjective experience of people interacting with the robot improved thanks to the use of the touch screen panel to reinforce speech interaction.

Fig. 5. *Gualzru* robot during the execution of the experiments in real scenarios.

The analysis of the execution of the use case, on the other hand, revealed occasional synchronization issues. Even after careful design, latency fluctuations and spurious interlocking may appear when running the 20 software components in four computers. This situation has been greatly improved with the introduction of the DSR, which makes us think that most of the problems derived from the original separation in two graphs. When the DSR was set, and after a slightly greater effort in the design phase to set the required symbolic attributes, their location in the kinematics tree and their update policy, most of the problems vanished.

Once these questions were solved, the overall system usage was quite reliable and robust. The highly distributed character of the architecture raises the interesting question of whether the self reallocation of processes to optimize hardware and communications usage across the set of on board computers would be a useful research line to pursue. Maybe the underlying middleware needed to perform that on line re allocations among computers would penalize more that the benefits obtained.

6 Conclusions

This paper discusses the design of CORTEX and an initial implementation of it on the service robot Gualzru. CORTEX is a cognitive architecture that unifies different levels of abstraction into a common representation. This implementation of CORTEX has been tested in a real scenario, and it has proven to be adequate for complex, social robotics applications. The architecture is efficient, flexible and robust.

As an ongoing research, other design options and are being currently explored by our research groups to improve CORTEX. The implementation of DSR has still issues to solve related to scalability. With more complex scenarios, the graph has to hold the geometry of many objects -meshes- belonging to a known map, so collision detection among objects can be calculated for planning purposes. Another possibility, even more demanding, would be to inject into the graph the raw data read by the sensors, i.e. laser and RGBD cameras, so different agents like *Navigation* or *ObjectPerception* could build and reuse meaningful percepts more easily. In this situation, parts of the graph would have update periods as low as 30 ms, while other would remain unchanged for seconds. To cope with the necessary bandwidth to keep all agents and internal components synchronized a partial updating strategy would have to be used. If we also take into account that agents can manipulate local copies of the DSR for emulation purposes before publishing an update, maybe collaborative repository technologies like *git* could be also inspirational.

Future work will focus on completing a running version of CORTEX, in which DSR will be tested in more challenging domains. Also, we will continue with the development of emulations using the DSR with different simulation engines, allowing CORTEX not only to represent the reality, but also to test new behaviours through virtual executions and internal evaluations.

Acknowledgment

This paper has been partially supported by the Spanish Ministerio de Economía y Competitividad TIN2012-TIN2012-38079-C03 and FEDER funds, and by the Innterconecta Programme 2011 project ITC-20111030 ADAPTA.

References

1. Comparative Table of Cognitive Architectures - BICA Society, http://bicasociety.org/cogarch/architectures.htm
2. Beetz, M., Stulp, F., Radig, B., Bandouch, J., Blodow, N., Dolha, M., Fedrizzi, A., Jain, D., Klank, U., Kresse, I., Maldonado, A., Marton, Z., Mösenlechner, L., Ruiz, F., Rusu, R.B., Tenorth, M.: The Assistive Kitchen - A demonstration scenario for cognitive technical systems. In: Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN. pp. 1–8 (2008)
3. Benjamin, D.P., Plaza, P., York, N.: ADAPT : A Cognitive Architecture for Robotics An Implementation of ADAPT. Forum American Bar Association pp. 337–338 (2001)
4. Chella, A., Frixione, M., Gaglio, S.: A cognitive architecture for robot self-consciousness. Artificial intelligence in medicine 44(2), 147–54 (Oct 2008), http://www.ncbi.nlm.nih.gov/pubmed/18715770
5. Chong, H.Q., Tan, A.H., Ng, G.W.: Integrated cognitive architectures: A survey. Artificial Intelligence Review 28(2), 103–130 (2007)
6. Diamond, A., Holland, O., Marques, H.G.: The role of the predicted present in artificial and natural cognitive systems. In: Proceedings of the Second Annual Meeting of the BICA Society. Arlingron, Virgina, USA (2011)
7. Duch, W., Oentaryo, R., Pasquier, M.: Cognitive Architectures: Where do we go from here? Frontiers in Artificial Intelligence and Applications 171, 122–136 (2008)
8. Gat, E.: On three-layer architectures. In: Kortenkamp, D., Bonnaso, R.P., Murphy, R. (eds.) Artificial Intelligence and Mobile Robots. AAAI Press (1998)
9. Grush, R.: The emulation theory of representation: motor control, imagery, and perception. The Behavioral and brain sciences 27(3), 377–96; discussion 396–442 (Jun 2004), http://www.ncbi.nlm.nih.gov/pubmed/15736871
10. Guzman, C., Alcazar, V., Prior, D., Onaidia, E., Borrajo, D., Fdez-Olivares, J., Quintero, E.: PELEA: a Domain-Independent Architecture for Planning, Execution and Learning. Proceedings of ICAPS'12 Scheduling and Planning Applications woRKshop (SPARK) pp. 38–45 (2012)
11. Haazebroek, P., van Dantzig, S., Hommel, B.: A computational model of perception and action for cognitive robotics. Cognitive processing 12(4), 355–65 (Nov 2011)
12. Haber, A., Sammut, C.: A Cognitive Architecture for Autonomous Robots. Advances in Cognitive Systems 2, 257–276 (2013), http://cogsys.org/journal/volume-2
13. Hanford, S.D.: A Cognitive Robotic System based on the SOAR cognitive architecture for mobile robot navigation, search and mapping missions. Ph.D. thesis, Pennsilvania State University, USA (2011)
14. Hiatt, L., Trafton, J.: A cognitive model of theory of mind. . . . International Conference on Cognitive . . . (2010), http://iccm-conference.org/2010/proceedings/papers/Hiatt.pdf
15. Holland, O., Goodman, R.: Robots with internal models. Journal of Consciousness Studies 10(4), 1–45 (2003), http://www.dipankarhome.com/Journal of Consciousness Studies.pdf
16. Holland, O., Goodman, R.: Robots With Internal Models A Route to Machine Consciousness? Journal of Consciousness Studies 10(4-5), 77–109 (2003), http://www.ingentaconnect.com/content/imp/jcs/2003/00000010/F0020004/1348

17. Journal, I., Vol, H.R., Scienti, W., Company, P., Kawamura, K., Ii, A.P., Bodenheimer, R.E., Sarkar, N., Park, J., Clifton, C.A., Spratley, A.W., Hambuchen, K.A., Soft, I., Control, A.: A parallel distributed cognitive control system for a humanoid robot. International Journal of Humanoid Robotics 1(1), 65–93 (2004)
18. Kingdon, R.: A review of cognitive architectures. Tech. rep., ISO Project Report. MAC 2008-9 (2008)
19. Laird, J., Kinkade, K., Mohan, S., Xu, J.: Cognitive Robotics Using the Soar Cognitive Architecture. In: 8th International Workshop on Cognitive Robotics. pp. 46–54 (2012), http://www.aaai.org/ocs/index.php/WS/AAAIW12/paper/download/5221/5573
20. Langley, P., Laird, J.E.J., Rogers, S.: Cognitive architectures: Research issues and challenges. Cognitive Systems Research 10(2), 141–160 (2009), http://linkinghub.elsevier.com/retrieve/pii/S1389041708000557 http://www.sciencedirect.com/science/article/pii/S1389041708000557
21. Lemaignan, S., Ros, R., Mosenlechner, L., Alami, R., Beetz, M.: ORO, a knowledge management platform for cognitive architectures in robotics. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems pp. 3548–3553 (Oct 2010), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5649547
22. Levesque, H., Lakemeyer, G.: Cognitive Robotics. In: Handbook of Knowledge Representation (2008)
23. Maes, P.: The agent network architecture (ANA). SIGART Bulletin 2(4), 115–120 (1991)
24. Manso, L., Bachiller, P., Bustos, P., Calderita, L.: RoboComp: a Tool-based Robotics Framework. In: SIMPAR, Second International COnference on Simulation, Modelling and Programming for Autonomous Robots (2010)
25. Marfil, R., Calderita, L.V., Bandera, J.P., Manso, L.J., Bandera, A.: Toward Social Cognition in Robotics : Extracting and Internalizing Meaning from Perception. In: Workshop of Physical Agents WAF2014. pp. 1–12. No. June, Leon, Spain (2014)
26. Minsky, M.: The Society of Mind. Simon & Schuster (1988)
27. Minsky, M.: The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind. Simon & Schuster (2006)
28. Nelissen, K., Luppino, G., Vanduffel, W., Rizzolatti, G., Orban, G.A.: Observing others: multiple action representation in the frontal lobe. Science (New York, N.Y.) 310(5746), 332–336 (2005)
29. Olson, E.: AprilTag: A robust and flexible visual fiducial system. Proceedings - IEEE International Conference on Robotics and Automation pp. 3400–3407 (2011)
30. P Bustos, J Martnez-Gómez, I Garca-Varea, L Rodrguez-Ruiz, P Bachiller, L Calderita, LJ Manso, A Sánchez, A Bandera, J.B.: Multimodal Interaction with Loki. In: Workshop of Physical Agents. pp. 1–8. Leon, Spain (2013), http://ljmanso.com/files/Multimodal_interaction_with_Loki.pdf
31. Puigbo, J.y., Pumarola, A., Tellez, R.: Controlling a General Purpose Service Robot By Means Of a Cognitive Architecture. In: Artificial Intelligence and Cognition Workshop (2013)
32. Puls, S., Graf, J.: Cognitive Robotics in Industrial Environments. In: Inaki, M. (ed.) Human Machine Interaction - Getting Closer. INTECH (2012), http://www.intechopen.com/source/pdfs/26326/InTech-Cognitive_robotics_in_industrial_environments.pdf
33. Rao, A.S., Georgeff, M.P.: BDI Agents: From Theory to Practice. Icmas pp. 312–319 (1995)

34. Romero-Garcés, A., Calderita, L.V., Martínez-Gómez, J., Bandera, J.P., Marfil, R., Manso, L.J., Bandera, A., Bustos, P.: Testing a fully autonomous robotic salesman in real scenarios. In: IEEE International Conference on Autonomous Robots Systems and Competitions. pp. 1–7. Vilareal, Portugal (2015)
35. Sammut, C.: When Do Robots Have to Think? Advances in Cognitive Systems 1, 73–81 (2012)
36. Scheutz, M.: Computational mechanisms for mental models in human-robot interaction. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8021 LNCS(PART 1), 304–312 (2013)
37. Scheutz, M., Harris, J., Schermerhorn, P.: Systematic Integration of Cognitive and Robotic Architectures. Advances in Cognitive Systems 2, 277–296 (2013)
38. Sloman, A., Wyatt, J., Hawes, N., Chappell, J., Kruijff, G.J.: Long term requirements for cognitive robotics. In: Cognitive Robotics Papers from the 2006 AAAI Workshop Technical Report WS0603. pp. 143–150. No. McCarthy (2006), http://www.aaai.org/Papers/Workshops/2006/WS-06-03/WS06-03-022.pdf
39. T. Ojala, M. Pietikäinen, D.H.: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: 12th IAPR International Conference on Pattern Recognition (ICPR 1994). pp. 582–585 (1994)
40. Taatgen, N.: The past, present, and future of cognitive architectures. Topics in Cognitive Science pp. 1–23 (2010), http://onlinelibrary.wiley.com/doi/10.1111/j.1756-8765.2009.01063.x/full
41. Taylor, P., Black, A.W., Caley, R.: The Architecture of the Festival Speech Synthesis System. In: The Third ESCA Workshop in Speech Synthesis. pp. 147–151. Jenolan Caves, Australia (1998), http://www.isca-speech.org/archive/ssw3/index.html
42. Thórisson, K.R., Helgasson, H.P.: Cognitive Architectures and Autonomy: A Comparative Review. Journal of Artificial General Intelligence 1(-1), 1–30 (2012)
43. Wei, C., Hindriks, K.V.: An agent-based cognitive robot architecture. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7837 LNAI, 54–71 (2013)
44. http://www.ros.org
45. Hofland, Knut and Jørgensen, Annette Myre and Drange, E and Stenstr. A Spanish spoken corpus of youth language, Corpus Linguistics. 2005.
46. Frederick Jelinek, Statistical methods for speech recognition, MIT Press. 1997.
47. David Chamorro and Vázquez Martín, Ricardo. R-ORM: relajación en el método de evitar colisiones basado en restricciones. X Workshop de Agentes Físicos, Cáceres, España, 2009.
48. https://git-scm.com/