Deep Representations for Collaborative Robotics

Conference Paper · December 2016

DOI: 10.1007/978-3-319-50862-7_14

7 authors, including:

Luis J. Manso
Aston University

Pablo Bustos
Universidad de Extremadura

Juan Pedro Bandera
University of Malaga

Adrián Romero-Garcés
University of Malaga

Deep Representations for Collaborative Robotics

Luis J. Manso[1], Pablo Bustos[1], Juan P. Bandera[2], Adrián Romero-Garcés[2], Luis V. Calderita[1,2], Rebeca Marfil[2], and Antonio Bandera[2](✉)

[1] RoboLab Group, University of Extremadura, Cáceres, Spain
{lmanso,pbustos,lvcalderita}@unex.es
[2] Dept. Tecnología Electrónica, University of Malaga, Málaga, Spain
{jpbandera,adrigtl,rebeca,ajbandera}@uma.es
https://robolab.unex.es/
http://www.grupoisis.uma.es/

AQ1

Abstract. Collaboration is an essential feature of human social interaction. Briefly, when two or more people agree on a common goal and a joint intention to reach that goal, they have to coordinate their actions to engage in joint actions, planning their courses of actions according to the actions of the other partners. The same holds for teams where the partners are people and robots, resulting on a collection of technical questions difficult to answer. Human-robot collaboration requires the robot to coordinate its behavior to the behaviors of the humans at different levels, e.g., the semantic level, the level of the content and behavior selection in the interaction, and low-level aspects such as the temporal dynamics of the interaction. This forces the robot to internalize information about the motions, actions and intentions of the rest of partners, and about the state of the environment. Furthermore, collaborative robots should select their actions taking into account additional human-aware factors such as safety, reliability and comfort. Current cognitive systems are usually limited in this respect as they lack the rich dynamic representations and the flexible human-aware planning capabilities needed to succeed in tomorrow human-robot collaboration tasks. Within this paper, we provide a tool for addressing this problem by using the notion of deep hybrid representations and the facilities that this common state representation offers for the tight coupling of planners on different layers of abstraction. Deep hybrid representations encode the robot and environment state, but also a robot-centric perspective of the partners taking part in the joint activity.

Keywords: Deep representations · Cognitive robots · Agent-based robotic architecture

1 Introduction

In order to engage humans in interactions, the new generation of robots should be able to emanate responses at human interaction rates and exhibit a proactive behaviour. This proactive behaviour implies that the internal architecture

K. Amunts et al. (Eds.): BrainComp 2015, LNCS 10087, pp. 1–15, 2016.
DOI: 10.1007/978-3-319-50862-7_14

of these robots should not only be able to perceive and act, but also to perform off-line reasoning. Cognition is the ability that allows us to internally deal with information about the external world and, hence, this ability is subject to the existence of an internal representation of this information. Classical cognitive systems posit an inner realm richly populated with internal tokens that stand for external objects and states of affair [27]. These internal representations, however, are not valid to generate predictions or reasoning. Recent works suggest that cognitive architectures cannot work on a passive, bottom-up fashion, simply waiting to be activated by external stimuli. Instead, these architectures must continuously use memory to interpret sensory information and predict the immediate future. These predictions about the outer world can be used to actively drive the resources to relevant data in top-down modes of behaviour, allowing an efficient and accurate interpretation of the environment [27,37].

The concept of deep representations was clearly described by Beetz et al. [33]: *representations that combine various levels of abstraction, ranging, for example, from the continuous limb motions required to perform an activity to atomic high-level actions, subactivities, and activities.* This definition is however provided in a paper where the robot performs its activities alone. If a collaborative robot has to cooperate with a human partner as a work companion, it should be endowed with the abilities to consider its environmental context and assess how external factors could affect its action, including the role and activity of the human interaction partner in the joint activity. Efficient collaboration not only implies a common plan for all involved partners, but also the coordination of the behavior of each agent with those of the other ones, i.e. to gain a joint intention. This coordination should be simultaneously addressed at different levels of abstraction, and to correctly satisfy it, the robot has to internalize a coherent representation about the motions, actions and intentions of the rest of partners. Additionally, a major difficulty in human-robot collaboration (HRC) scenarios is that people cannot only exhibit a rather non-deterministic and unstable behavior, but they also tend to perceive current robots as slow and unintelligent. These factors difficult HRC. To overcome them, the robot should continuously try to guess their partners' goals and intentions, triggering appropriate reactions -i.e. being socially proactive.

Symbolic and metric representations have been separately proposed in many different forms and uses. Symbolic knowledge representation have been at the core of AI since its beginnings [21,23] and cover all forms of relational formalizations such as production rules, frames, schemes, cases, first order logic or situational calculus. At a high level of abstraction, the Robot Learning Language (RoLL) [32] could be used for learning models about human behaviour and reactions, joint plan performance or recognizing human activity. Also, human models have been employed by the Human-Aware Task Planner (HATP) [34]. A symbolic graph structure was proposed in [14] as part of our previous architecture RoboCog [6]. On the other hand, metric and kinematic representations are commonly used as part of 3D simulators and graphics engines [39,40]. However, the concept of deep representations [33] implies an unified, hierarchical organization of the knowledge that ranges from the symbolic layer to the motor one, mapping abstract concepts to, or from, geometric environment models and sensor data structures of the robot. The inclusion of a detailed physical layer on the representation will allow the robot to solve naive physics problems, which cannot be performed based on abstractions, using temporal projection [26]. The presence of a detailed representation of the spatial state of the problem is also required in the work of Wintermute: *... actions can be simulated (imagined) in terms of this concrete representation, and the agent can derive abstract information by applying perceptual processes to the resulting concrete state* [35]. The use of a situational representation of the outer world to endow the robot with the ability to understand physical consequences of their actions can be extended, in a collaborative scenario, to support proactive robot behaviors. This possibility has been addressed in the LAAS Architecture for Autonomous Systems proposed by Ali et al. [36].

The rest of the paper is organized as follows: First, Sects. 2 and 3 present arguments that support the former claims. Section 4 describes the functioning of our proposed architecture as a set of agents interacting through the deep state representation. Section 5 briefly presents several application scenarios where the world model is currently been tested. Conclusions and future work are drawn at Sect. 6.

2 Agency

The concept of agent in Computer Science and Artificial Intelligence is rather broad and their varied meanings cover most of what a program can do. Franklin [10], after a through review of many proposals, synthesizes an autonomous agent as *a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.* We propose here a similar definition of the term,

> A computational entity in charge of a well defined functionality, whether it be reactive, deliberative of hybrid, that interacts with other agents inside a well-defined framework, to enact a larger system.

When several of these agents are somehow interconnected to enact a higher-level function, they are called Agent-based Architectures. In robotics and AI, they have been used for a long time [13,22] being Minsky's Society of Mind [17], probably the most famous one.

From the point of view of the implementation, we map agents to software components in a one-to-many or one-to-one policy. To complete the definition, a software component is a program that communicates with other programs inside a well defined framework. Components are usually created as an instance of a formal component model [4,5,25]. Note that there is not much difference between agents and components and in many contexts both are interchangeable. However, we use component here in the more restricted sense of being a program, rather than a more general functional abstraction, like agents.

The CORTEX architecture is implemented using the component-oriented robotics framework RoboComp [15]. The choice of a agent-based architecture responds to computational simplicity and elegance. Agents are functional units that can be easily combined to form in a given structure. They can be defined recursively as made up of other simple agents, and there is always a rather simple connection to the underlying software components. This is the first reason why CORTEX is an agent-based architecture.

In CORTEX, higher-level agents define the classic functionalities of cognitive robotics architectures, such as navigation, manipulation, person perception, object perception, dialoguing, reasoning, planning, symbolic learning or executing. These agents operate in a goal-oriented regime [23] and their goals can come from outside through the agent interface, and can also be part of the agent normal operation. For our needs, we want agents to be autonomous and obedient, at the same time. Autonomous to provide opportunistic behavior so non-planned events in the environment can be detected, and obedient so when new goals arrive to the system, all task-oriented agents start working to achieve the current sub goal. Thus, regarding the kind of function they perform, agents can be anything in the reactive-deliberative spectrum, although in this paper and to simplify the exposition we will refer to them whether as deliberative or reactive. This flexibility in the internal organization of the agents, as the building blocks on CORTEX, is the second reason why we have chosen an agent-based architecture.

Communication among agents define the structural part of the architectures. In cognitive robotics architectures, instead of a search for a correct model of human intelligence, what is explored is the design space of embodied intelligence [24]. We adopt here the broad view that these systems encompass two main flows of information. First, a deliberative one, in which agents must provide a symbolic description of the robot and the world to the deliberative agents, so they can reason about facts and plan a course of actions to achieve the current goal. These actions are sent back to the non-deliberative agents as local goals. Second, a reactive one, in which non deliberative agents interact to perform a sort of multi-modal behavior. These behaviors can be triggered by external goals or by a recognizable situation ahead. Note that other horizontal functions of the architecture such as memory and learning are left here to the internal functioning of the agents and the representational mechanisms that they share.

An important requirement for the architecture is to facilitate the transition between deliberatively controlled behaviors and autonomous behaviors, so there can be an overall improvement of task achieving performance. Either by hand coding or by automatic learning, the way the architecture interconnects agents must facilitate the incremental creation of autonomous, efficient and reliable skills. Let's examine now a simple situation in which two agents have to interact to complete a task. The agents are a *Manipulation* agent and an *Object-Perception* agent. Within the deliberative flow, to locate and grasp an object both agents must be coordinated by an *Executive* agent following a previously computed plan. The basic steps could be,

1. DetectTarget(t)
2. MoveHand(t)
3. Grab(t)

A plan like this, when executed by the action and perceptual agents, although it might succeed, will perform poorly. The reason is that there are many assumptions being made about the robot and the world that will result in a slow, failure prone and rigid behavior. It would be similar to someone following low-level orders to do some new manipulation task. The solution is assuming that the target is static, the position of the target obtained by the camera and the position to where the hand will arrive are the same, i.e. perfect calibration, that the arm movement is very precise, that there is not uncertainty in the target position as detected by the camera, among others. In real scenarios with uncertainty in the measurement process, low cost mechanics and errors in the kinematic calibration of the body-arm-head ensemble, a much more reliable method would be to use a two-fold sequence. First, a sort saccadic arm movement takes the hand to a zone close to the target, then, a visual servo loop re-positions the hand in the reference system of the camera, thus canceling all calibration errors between the camera and the hand [11,12]. This is a clear example of a skill that requires a fluid interchange of information between *Object-Perception* and *Manipulation*. Note that the visual perception of the hand would also correspond to *Object-Perception*. Adapting the former sequence of steps,

1. DetectTarget(t)
2. MoveHand(t)
3. InitiateVisualServo()
4. Grab()

MoveHand() is now an autonomous skill that involves both agents and requires and intense interchange of information between them. The robotics cognitive architecture must be prepared to incorporate these kind of changes. The communication channels among agents must be flexible enough to allow for these kind of reorganizations. Note that even if the agents would have been chosen differently, so there were no need to transfer information externally, there will be always other situations in which different agents would need to communicate[1]. So the claim here is that a smart decision for an architecture would be to think of some mechanism that could communicate information among agents without much restrictions and still, be flexible enough to allow modifications, whether by the programmer, or automatically through learning algorithms. This is the third reason why CORTEX is built as a set of loosely coupled agents whose communication channels can be chosen in many different ways. We want to anticipate the possibility of future improvements that will come in the near future, and the design space provided by an agent-based architecture with regard to how agents communicate is large. In particular, and as a solution seeking flexibility, we propose here that they communicate through a shared data structure, as explained in Sect. 3.

[1] Obviously we are discarding here the one big agent-doing-everything case.

3 The Deep State Representation

The Deep State Representation (DSR) is a graph structure that, within CORTEX, holds the representation of the robot and its environment. It is not the first time that a graph structure is used for this purpose. However, to our knowledge, the first works that proposed a graph as an internal representation for a robotics architecture focused only in geometric data. ROS' transform library, *tf* [9], BRICS Robot Scene Graph [3] and RoboCog's InnerModel [20] all appeared in 2013 as a response to the need for such a structure: a centralized representation of robot and world kinematics. Even tough those constructions are important advances towards better robotic architectures, a richer, and deeper representation was needed to hold the complete set of beliefs of the robot. In CORTEX, the graph structure of the DSR holds symbolic and geometric data, and is accessed by all agents during their operations. In fact, the DSR is the only means for the agents to communicate. Figure 1 shows a small DSR graph with multiple labeled edges representing heterogeneous attributes.

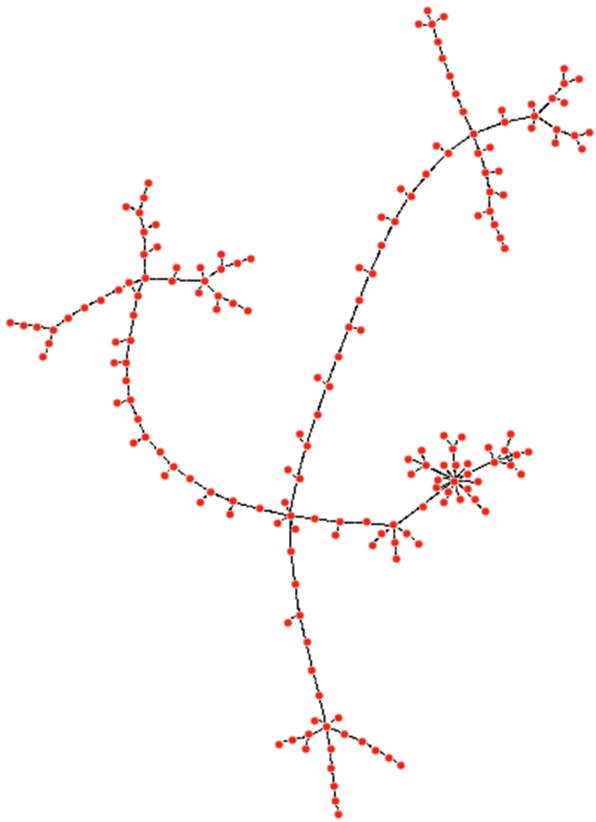


Fig. 1. Full graph of the DSR.

The idea of a shared representation among agents has its roots in several classical papers [8,18,19] that developed the concept of the blackboard architecture. Later, Hayes-Roth [2] extended this idea into a complete control architecture. As A, Newell himself put it,

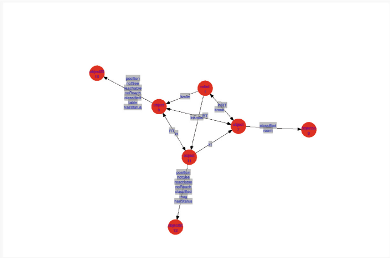
> Metaphorically we can think of a set of workers, all looking at the same blackboard: each is able to read everything that is on it, and to judge when he has something worthwhile to add to it. This conception is just that of Selfridge's Pandemonium [19]: a set of daemons, each independently looking at the total situation and shrieking in proportion to what they see that fits their natures...

In the original blackboard systems, agents where conceived more as problem solvers, heterogeneous experts that contribute to the overall problem in a hybrid planned-opportunistic way. They communicate through a shared structure where goals, sub goals and problems state were incrementally updated. In CORTEX, agents solve not only deliberative tasks but also perceptual, motor and behavioral, so their communication needs are somewhat different. Nevertheless, we gather some ideas from these architectures [7,16] and also others from graph theory and distributed databases.

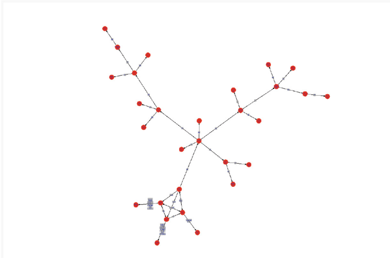
3.1 Why Is the DSR a Graph?

The first reason to use a graph in CORTEX is because all internal information defining the state of the robot and its beliefs about the environment, can be stored according to a predefined structure. That structure is a model of how sensor data can be interpreted and organized. As general data structures, can hold any relational knowledge composed of discrete elements and relations among them. In this broad category fall almost all symbolic knowledge representation methods including frames, schemes, production rules and cases, and also the geometric knowledge that the robot has to maintain about itself and the environment. This geometric knowledge includes instances of the types of objects recognizable in the world like i.e. chairs, tables, cups or generic obstacles of undefined form. Also human bodies and its parts like arms, heads, legs, etc. All these parts are kinematically related through 3D transformations forming a scene-tree.

A second reason is that the graph can be made to evolve under some generative rules. Assuming that the type of nodes and edges are predefined, the graph can evolve by inclusions or deletions of parts, causing structural changes. Also it can evolve by changing the value of the attributes stored in nodes and edges. The structural changes can be regulated by a generative grammar that defines how the initial model can change. A typical example would be that of the robot entering a new room and, after exploring it, it would add a new node to the graph. The grammar would impede the new node to be connected to something else but the corresponding door, and maybe it would be oriented parallel to one of the walls of the proceeding room. So graphs gives us the capacity needed



(a) Initial world model in DSR with the robot and the room.



(b) A person enters the room, and when detected by the *Person* agent, inserted in the DSR.

Fig. 2. Illustration of various worldModel State. The symbol robot has been constrained to one symbol for explanatory reasons.

to store objects and their relations, and combined with a grammar to control its evolution, gives us a coherent growing model. Figure 2 shows how the graph changes when a person enters the scene. In the left side only the robot and the rooms are represented. In the right side, a person enters the room and the graph incorporates her as sub graph correctly related to the existing structure and with symbolic attributes denoting what is known about her.

A third reason to use a graph structure is the possibility of translating it into a PDDL instance. Depending on what is stored in the graph and the PDDL version this procedure has certain restrictions but it allows a direct use of start of the art planning algorithms that otherwise would have required an important additional effort. Further details on how this translation is done can be found in [14].

A fourth reason to support the choice of graphs is the facility to visualize its contents. Graph's contents can be displayed in multiple ways using available 3D technology and that is a crucial feature to debug the code of the agents, specially when interacting among them. In CORTEX, visualization of DSR is done using the open source 3D scene-graph OpenSceneGraph, OSG [1] and a class implementing the observer pattern that keeps DSR and OSG synchronized. The DSR graph can be drawn in different ways. The geometric nodes and edges are drawn as a normal 3D scene, using the meshes and 3D primitives that can be stored as attributes in DSR. The symbolic relations can be drawn as an independent graph or as a superimposed structure on its geometric counterpart.

3.2 DSR Formalization

DSR is a multi-label directed graph which holds symbolic information as logic attributes related by predicates. These are stored in nodes and edges respectively. Also, DSR holds geometric information as predefined object types linked by 4×4 homogeneous matrices. Again, these are stored in nodes and edges respectively. With DSR, the hand of the robot can be at a 3D pose and, at the same time, it can be *close_to_the_door knob*, being this a predicate computed by measuring the distance between the hand and the knob, in the graph representation. Note that this distance could also had been measured with more precision by direct observation of both the knob and the hand once they are inside the frustum of the robot's camera but, at the end, that information would be stored in the graph and propagated to the other agents.

As a hybrid representation that stores information at both metric and symbolic level. The nodes store concepts that can be symbolic, geometric or a mix of them. Metric concepts describe numeric quantities of objects in the world that can be structures like a three-dimensional mesh, scalars like the mass of a link, or lists like revision dates. Edges represent relationships among symbols. Two symbols may have several kinds of relationships but only one of them can be geometric. The geometric relationship is expressed with a fixed label called "RT". This label stores the transformation matrix between them. A formal definition of DSR can be given as a multi-label directed graph $G = (N, E)$ where N represents the set of nodes $\{n_1, ...n_k\}$ and E the set of edges $\{e_1....e_r\}$.

$$G = (V, E) \text{ where } E \subseteq N \times N, uv \neq vu (\text{without loops } vv) \tag{1}$$

According to its symbolic nature, edges properties are:

1. For each pair $e = uv$ the inverse does not exist $e = uv \neq e^{-1} = vu$
2. For each pair $e = uv$, e can store multiple values
3. The set of e is defined as $L = \{e_1, ...e_r, (l_1, l_2, ...l_s)\}$ where $l_i \neq l_j$

According to its geometric nature and the properties of the transformation matrix RT, the characteristics of geometric edges are:

1. For each pair $e = uv = RT$, e is unique
2. For each pair $e = uv = RT$, define the inverse of e as $e^{-1} = vu = RT^{-1}$

Therefore the kinematic chain $C(u, v)$ is defined as the path between the nodes u, v and an equivalent transformation $RT*$can be computed by multiplying the equivalent transformations corresponding to the sub paths from each node to their closest common ancestor. Note that sub path from the common ancestor to v will be obtained multiplying the inverse transformations.

This geometrical relations are showed in Fig. 3.

4 CORTEX Internal Organization

The functioning of the architecture as a set of agents interacting through the DSR 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, updating parts of it as things change. 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. In the process of going back to normality the system extends its internal model, capturing more details of

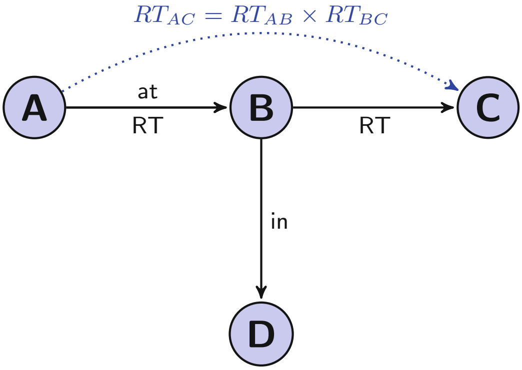


Fig. 3. Unified representation as a multi-labeled directed graph. Edges are labeled "at" and "in" denoting logic predicates between nodes. Also, edges between A,B and B,C have a geometric type of label, "RT" that codes a rigid transformation between them. Geometric transformations can be chained or inverted to compute changes in coordinate systems.

the external world. The new knowledge is used in the next perturbation. Furthermore, the idea of opportunistic control by which agents can write in the blackboard is driven not only as a result of solving a local goal but triggered by internal events. A typical situation for a visual perceptive agent would be to configure, search and track types of objects specified in the current plan, but these agents are also in charge of other secondary *unconscious* tasks like obstacle detection for navigation or simply, novelty detection if in a well known environment. In this cases the agent would inject the percepts into the shared graph so the information can be used somewhere else.

To complete the picture of how CORTEX is organized, Fig. 4 shows a very schematic sketch of the main agents and their connection through the DSR.

- The central part of the figure represents the DSR. It is enclosing all the Executive, a module in charge of managing the inclusion of new changes on the DSR or publishing the new DSR to the agents.
- In the figure, the squared boxes that surround the inner representation represent networks of software components -agents. They encode complete robotics functionalities -e.g. navigation, conversation, planning, etc- and share information about the state through the representation. The current picture shows the instantiation of CORTEX within Gualzru.
- Finally, the boxes on the top part of the figure enclose action modules (e.g., for moving the robot or speech a phrase). The boxes on the down part enclose perception modules (e.g. for capturing a laser scan or the battery level). The WinKinectComp module is a specific component that provides information taken from the Kinect sensor from Microsoft (skeletons, joins, faces...) and speech transcriptions. All these modules encode the Hardware Abstraction Layer (HAL).

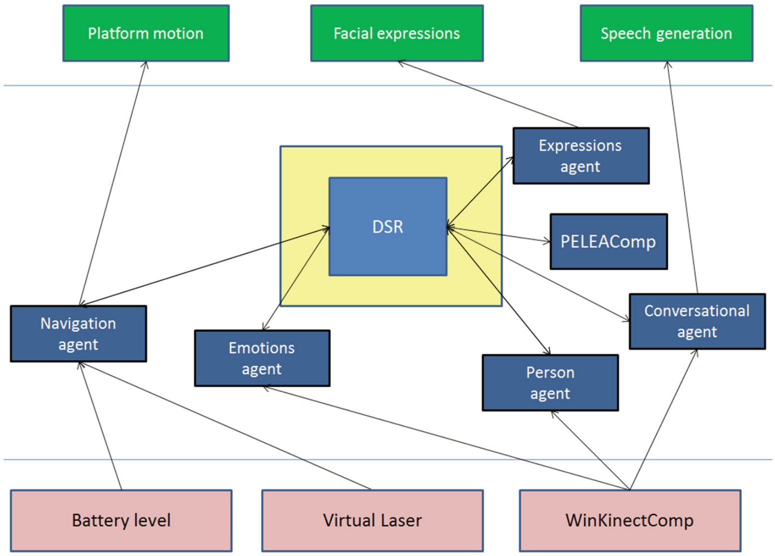


Fig. 4. Overview of the shared world model and its location within the cognitive architecture. The picture shows the proposed framework within the Gualzru scenario described at Sect. 5 (see text for details).

5 Experimental Results

As an initial validation of CORTEX and DSR in a real robot interacting with humans we tested the ideas on Gualzru [30]. Gualzru is a salesman robot that works autonomously in crowded scenarios and has to step out when a potential client passes by. Gualzru will approach the customer and start a conversation trying to convince her to go to an interactive sales panel. If the robot succeeds, it will walk the person to the panel and then will start a new search.

In previous versions of the robot, we found that some synchronization problems were caused by having a fragmented internal representation. The robot used two separate graphs, one for the kinematic state and one for symbolic attributes and predicates. There several architectures that keep these representations separated [6] and it is a reasonable choice since both hold different contents and require different types of processes. However, in complex interaction scenarios, an integrated representations simplifies synchronization making all data available to all agents at the same time. The introduction of DSR in Gualzru solved those problems and agent communications worked flawlessly.

Other important drawback was related to its limited conversational abilities. These limitations greatly affects its performance. Speech recognition is hard to solve in noisy, crowded scenarios in which even people find difficulties in understanding each other (see Fig. 5). It is also difficult to understand what the robot is saying. To improve the ability of the robot to communicate within this scenario, we have added a tactile screen on the robot. This screen displays what

Fig. 5. The Gualzru robot interacting with people at the University of Malaga.

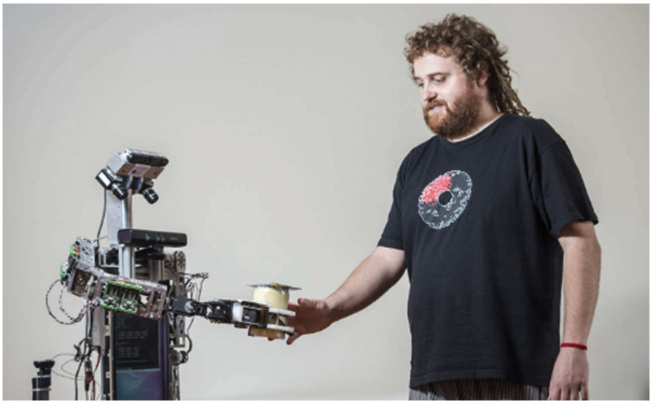


Fig. 6. Shelly is closing one mission.

the robot is saying. It allows the person to answer to the robot by touching the desired response on the screen. All this information is shared among agents using the symbolic annotations added to DSR. It is important to note that these concepts can be updated by the agents at interaction rates. The ADAPTA project started on 2012, and different representations were evaluated. The last trials on September 2015 allowed to deeply test that the new representation is able to engine the whole architecture at human interaction rates.

The CORTEX architecture was also endowed within Shelly, an autonomous mobile robot that, contrary to the Gualzru robot, has two arms, with 7 DoFs and a final end-effector. The initial aim is to locate and grasp objects. Figure 6 shows how Shelly is bringing one cup to the user. All the robot's activity (room and table perception, speech recognition and command identification, cup localisation and grasping, etc.) is performed without human supervision. The total number of software components is now greater than 20, all of them organised on agents that are connected to the DSR. This remains as the only way for interchanging information among agents. The robot is able to correctly analyse the context and solves the commanded missions.

6 Conclusions and Future Work

This paper has presented our proposal for internalizing a deep state representation of the outer world. After testing the previous approaches in very demanding scenarios, the unified representation arises as our final approach for endowing the full kinematic tree with symbolic information; and providing the geometric information to the high-level planner. The unified representation is currently interfaced by a set of task-related networks of agents, which will provide broad functionalities such as navigation, dialog or multi-modal person monitoring. The current implementation guarantees that the agents are able to feed the unified representation with new geometric models or symbolic concepts, and that the data stored in the representation is kept synchronized with the real world by updating actions performed by different agents.

Future work will focus on exploiting the hierarchical structure encoded within the DSR. This will allow the agents to subscribe to specific parts of the representation (e.g. the person or the robot arm). It is also needed to evaluate the computational times associated to the management of our graphs. Although it is clear that the number of nodes/arcs is relatively small, it must be noted that this graph is currently shared with all the agents on the architecture.

Acknowledgments. This paper has been partially supported by the Spanish Ministerio de Economía y Competitividad TIN2015-65686-C5 and FEDER funds, and by the Innterconecta Programme 2011 project ITC-20111030 ADAPTA.

References

1. http://www.openscenegraph.com
2. Hayes-Roth, B.: A blackboard architecture for control. Artif. Intell. **26**(3), 251–321 (1985)
3. Blumenthal, S., Bruyninckx, H., Nowak, W., Prassler, E.: A scene graph based shared 3D world model for robotic applications. In: 2013 IEEE International Conference on Robotics and Automation, pp. 453–460, May 2013. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6630614
4. Brugali, D., Scandurra, P.: Component-based robotic engineering (Part I). IEEE Robot. Autom. Mag. **16**, 84–96 (2009)

5. Brugali, D., Shakhimardanov, A.: Component-based robotic engineering (Part II): systems and models. IEEE Robot. Autom. Mag. **17**(1), 100–112 (2010)
6. Calderita, L.V., Bustos, P., Suárez Mejías, C., Fernández, F., Bandera, A.: THERAPIST: towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children. In: 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, vol. 1, pp. 374–377 (2013). http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6563970&tag=1
7. Corkill, D.D.: Blackboard systems. AI Expert **6**(9), 40–47 (1991)
8. Erman, L.D., Hayes-Roth, F., Lesser, V.R., Reddy, D.R.: The Hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty. ACM Comput. Surv. **12**(2), 213–253 (1980)
9. Foote, T.: tf: The transform library. In: 2013 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), pp. 1–6. Open-Source Software workshop (2013)
10. Franklin, S., Graesser, A.: Is It an agent, or just a program?: A taxonomy for autonomous agents. In: Müller, J.P., Wooldridge, M.J., Jennings, N.R. (eds.) ATAL 1996. LNCS, vol. 1193, pp. 21–35. Springer, Heidelberg (1997). doi:10.1007/BFb0013570
11. Gratal, X., Romero, J., Bohg, J., Kragic, D.: Visual servoing on unknown objects. Mechatronics **22**(4), 423–435 (2012). http://dx.doi.org/10.1016/j.mechatronics.2011.09.009
12. Hutchinson, S., Hager, G., Corke, P.: A tutorial on visual servo control. IEEE Trans. Robot. Autom. **12**(5), 651–670 (1996). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=538972
13. Maes, P.: The agent network architecture (ANA). SIGART Bull. **2**(4), 115–120 (1991)
14. Manso, L.J.: Perception as stochastic sampling on dynamic graph spaces. Ph.D. thesis, Universidad de Extremadura (2013)
15. 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)
16. McManus, J.W.: Design and analysis of concurrent blackboard systems. Ph.D. thesis, College of William and Mary, Virginia (1992)
17. Minsky, M.: The Society of Mind. Simon & Schuster, New York (1988)
18. Newell, A.: Some problems of basic organization in problem-solving programs. Technical report, RAND Corporation (1962)
19. Oliver, G.S.: Pandamonium: a paradigm for learning. In: Proceedings of the Symposium on the Mechanization of Thought Processes, pp. 511–529 (1959)
20. Bustos, P., Martnez-Gomez, J., Garca-Varea, I., Rodrguez-Ruiz, L., Bachiller, P., Calderita, L., Manso, L.J., Sanchez, A., Bandera, A., Bandera, J.: Multimodal interaction with loki. In: Workshop of Physical Agents, Leon, Spain, pp. 1–8 (2013). http://ljmanso.com/files/Multimodal_interaction_with_Loki.pdf
21. Poole, D., Mackworth, A.: Artificial Intelligence: Foundations of Computational Agents. Cambridge University Press, Cambridge (2010). http://artint.info/index.html
22. Rao, A.S., Georgeff, M.P.: BDI agents: from theory to practice. In: ICMAS, pp. 312–319 (1995)
23. Rusell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Pearson, Upper Saddle River (2009)

24. 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
25. Wang, A.J.A., Qian, K.: Component-Oriented Programming. Wiley, Hoboken (2005)
26. Kunze, L., Dolha, M.E., Guzman, E., Beetz, M.: Simulation-based temporal projection of everyday robot object manipulation. In: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011) (2011)
27. Clark, A.: An embodied cognitive science? Trends Cogn. Sci. **3**(9), 345–351 (1999)
28. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press, Cambridge (1997) AQ2
29. Calderita, L.V., Manso, L.J., Bustos, P., Suárez-Mejías, C., Fernández, F., Bandera, A.: THERAPIST: towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children. In: JMIR Rehabil Assist Technol (2014)
30. Romero-Garcés, A., Calderita, L.V., González, J., Bandera, J.P., Marfil, R., Manso, L.J., Bandera, A., Bustos, P.: Testing a fully autonomous robotic salesman in real scenarios. In: Conference: IEEE International Conference on Autonomous Robot Systems and Competitions (2015)
31. Manso, L.J.: Perception as stochastic sampling on dynamic graph spaces. Ph.D. dissertation, University of Extremadura, Spain (2013)
32. Kirsch, A., Kruse, T., Msenlechner, L.: An integrated planning and learning framework for human-robot interaction. In: 4th Workshop on Planning and Plan Execution for Real-World Systems (held in conjunction with ICApPS 09) (2009)
33. Beetz, M., Jain, D., Msenlechner, L., Tenorth, M.: Towards performing everyday manipulation activities. Robot. Autonom. Syst. **58**, 1085–1095 (2010)
34. Alami, R., Chatila, R., Clodic, A., Fleury, S., Herrb, M., Montreuil, V., Sisbot, E.A.: Towards human-aware cognitive robots. In: AAAI-06, Stanford Spring Symposium (2006)
35. Wintermute, S.: Imagery in cognitive architecture: representation and control at multiple levels of abstraction. Cogn. Syst. Res. **1920**, 129 (2012)
36. Ali, M.: Contribution to decisional human-robot interaction: towards collaborative robot companions, Ph.D. thesis, Institut National de Sciences Appliquées de Toulouse, France (2012)
37. Holland, O.: The future of embodied artificial intelligence: machine consciousness? In: Iida, F., Pfeifer, R., Steels, L., Kuniyoshi, Y. (eds.) Embodied Artificial Intelligence. LNCS (LNAI), vol. 3139, pp. 37–53. Springer, Heidelberg (2004). doi:10.1007/978-3-540-27833-7_3
38. Manso, L.J.: Perception as stochastic sampling on dynamic graph spaces. Ph.D. thesis, University of Extremadura, Spain (2013)
39. http://wiki.ros.org/urdf
40. https://www.khronos.org/collada/

Author Queries

Chapter 14

Query Refs.	Details Required	Author's response
AQ1	Please confirm if the corresponding author is correctly identified. Amend if necessary.	
AQ2	References [28, 29, 31, 38] are given in the list but not cited in the text. Please cite them in text or delete them from the list.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	/ through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	/ through letter or through characters	new character / or new characters /
Change to italics	— under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	/ through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	/ through character or where required	
Reduce space between characters or words	between characters or words affected	