A Unified Internal Representation of the Outer World for Social Robotics

Pablo Bustos[1], Luis J. Manso[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
{pbustos,ljmanso,lvcalderita}@unex.es
https://robolab.unex.es/
[2]Dept. Tecnología Electrónica, University of Malaga, Málaga, Spain
{jpbandera,adrigtl,rebeca,ajbandera}@uma.es
www.grupoisis.uma.es/

Abstract. Enabling autonomous mobile manipulators to collaborate with people is a challenging research field with a wide range of applications. Collaboration means working with a partner to reach a common goal and it involves performing both, individual and joint actions, with her. Human-robot collaboration requires, at least, two conditions to be efficient: a) a common plan, usually under-defined, for all involved partners; and b) for each partner, the capability to infer the intentions of the other in order to coordinate the common behavior. This is a hard problem for robotics since people can change their minds on their envisaged goal or interrupt a task without delivering legible reasons. Also, collaborative robots should select their actions taking into account human-aware factors such as safety, reliability and comfort. Current robotic 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 these collaboration tasks. In this paper, we address this problem by proposing and discussing a deep hybrid representation, DSR, which will be geometrically ordered at several layers of abstraction (deep) and will merge symbolic and geometric information (hybrid). This representation is part of a new agents-based robotics cognitive architecture called CORTEX. The agents that form part of CORTEX are in charge of high-level functionalities, reactive and deliberative, and share this representation among them. They keep it synchronized with the real world through sensor readings, and coherent with the internal domain knowledge by validating each update.

Keywords: Social robotics, world internalization, deep representations

1 Introduction

While the economic benefits of robotics in industry are already clear, it is expected that their inclusion in everyday life will have a tremendous impact. The EU's H2010 initiative states that, as human assistants, tomorrow's robots will have the capacity to resolve many of the future economic and social challenges faced by European society, such as aging and well-being. However, to access these new markets and to be competitive, robots have to be dependable, smarter and able to work in closer collaboration with humans. In these scenarios, human-robot interaction (HRI) is now envisioned more as a relationship among companions than a mere master-slave relationship. It can be considered that the complete design of a real co-robot is beyond the scope of what can be achieved technically today. Still, to make progress along this path there are several important issues that are currently being discussed as a way to facilitate the design of new cognitive robotic architectures that will, one day, show real human-robot collaboration (HRC).

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 -e.g. semantic, situational or motor, and the robot has to internalize a coherent representation about the motions, actions and intentions -including abilities and preferences- of the rest of partners. Additionally, a major difficulty in HRC scenarios is that people can 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, trigger appropriate reactions and, ultimately, be socially proactive.

In this paper we will argue that to fully develop HRI, and to pave the way into HRC, a cognitive robotics architecture should use a deep, central representation shared among the agents composing it, which codes information at different levels of abstraction. We will explore here this issue, unfolding the arguments that support it and the design decisions taking during its development.

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* [1], BRICS Robot Scene Graph [2] and RoboCog's InnerModel [3] all appeared in 2013 as a response to the need for an structured, centralized representation of the robot and world kinematics. These 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. The concept of deep representations was first described by Beetz et al. [31] as *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. In a collaborative scenario, we should also consider representation and inference mechanisms for models including the persons bodies, actions, abilities and intentions.

Separately, symbolic and metric representations have been proposed in many different forms and uses. Symbolic knowledge representation have been at the core of AI since its beginnings [4] [5] and cover all forms of relational formalizations such as production rules, frames, schemes, cases, semantic nets, first order logic or situational calculus. At a high level of abstraction, the Robot Learning Language (RoLL) [30] 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) [32]. A symbolic graph structure was proposed in [6] as part of our previous architecture RoboCog [7] and it will be described in later sections. Metric and kinematic representations are commonly used as part of 3D simulators and graphics engines [8] [38] [40].

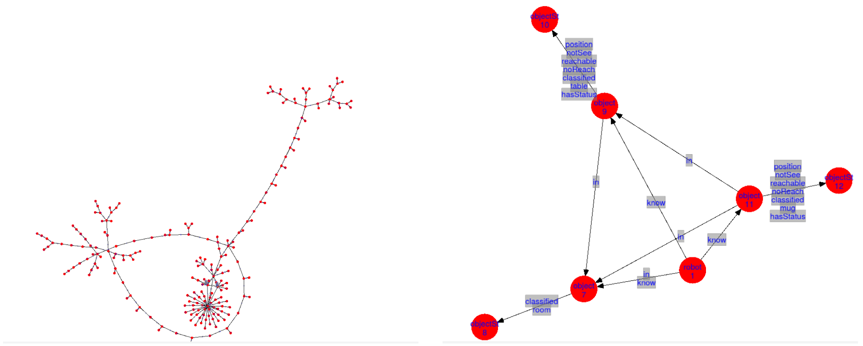
However, the concept of deep representations implies an unified, hierarchical organization of the knowledge that ranges from the symbolic layer to the motor one, mapping abstract concepts to, or from, geometric environment models and sensor data structures of the robot. The presence of a detailed representation of the spatial state of the problem is also required in the work of S. 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* [33]. 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. (2009).

The rest of the paper is organized as follows: Section 2 presents arguments and examples that support the former claims. Section 3 presents an application scenario where the world model is currently been tested. Conclusions and future work are drawn at Section 4.

2 The Deep State Representation

CORTEX is an agent-based new cognitive robotics architecture designed as an evolution of our former RoboCog architecture that provides the agents with a shared, hybrid representation of the robot's belief about itself and its environment. This graph-like structure is called DSR and can be accessed by all agents during their operations. DSR is the only means for the agents to communicate among them. Figure 1 shows a small DSR graph with multiple labeled edges representing heterogeneous attributes.

The idea of a shared representation among agents has its roots in several classical papers [9] [10] [11] that developed the concept of *blackboard architecture*. Later, Hayes-Roth [12] extended this idea into a complete control architecture. 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



(a) A complete DSR graph

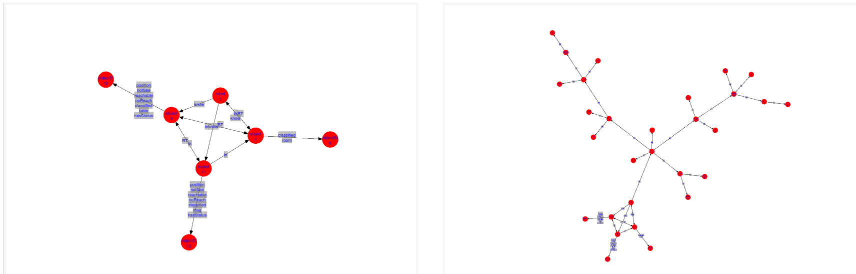
(b) Zoom in showing predicates and node names

Fig. 1. The DSR graph at different levels of resolution, showing metric and symbolic properties.

solve not only deliberative tasks but also perceptual, motor and behavioral ones, so their communication needs are somewhat different. Nevertheless, we gather some ideas from these architectures [13] [14] and also others from graph theory and distributed databases [43]. We present now some arguments supporting the need of a graph representation if the robot's inner beliefs.

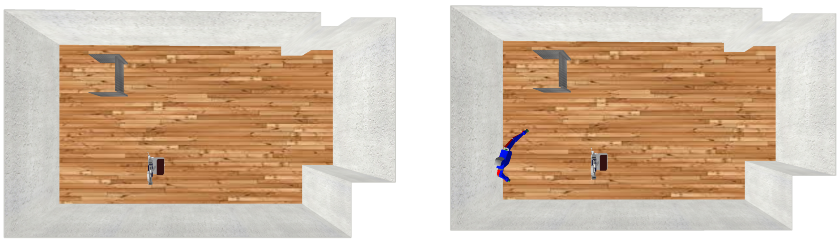
The first reason to use a graph in CORTEX is because all internal information defining the state of the robot and its beliefs about the environment can be stored according to a *generic structure*. As generic data structures, graphs can hold any relational knowledge composed of discrete elements and relations among them. In this broad category falls 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 the 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 have to be oriented parallel to one of the walls of the proceeding room. So graphs give us the capacity needed to store objects and their relationships and, combined with a grammar, a means to control its evolution to produce a growing model coherent with some initial domain knowledge. 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. Thus, the graph is not only a means of storage but a way to articulate information coming from sensors and processed by agents. Once in the graph, information can be accessed and interpreted by other agents.



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

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



(c) Graphic representation of the geometric view.

(d) Graphic representation of the geometric view when a person is inserted in the DSR.

Fig. 2. Two states of the DSR graph, before a person enters the room (a,c), and after she is detected and inserted in the DSR, (b,d).

A third reason to use a graph structure is the possibility of translating it into a *PDDL* instance. There are certanin restrictions that depend on what is stored in the graph and the PDDL version used, but it allows a direct use of start of the art planning algorithms that otherwise would have required an important effort. Further details on how this translation is done can be found in [6].

A forth 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 this is a crucial feature to debug the code of the agents, specially when interacting among them. In CORTEX, visualization of the DSR is done using the open source 3D scene-graph OpenSceneGraph, OSG [15] 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.

An additional reason to use a graph is because it is possible to *share* it efficiently among the agents using different techniques. The goal is to provide the agents with a mechanism to modify the graph and propagate that modification to all others. As long as this is achieved, all agents will have access to the global represented state and will be able to use it as a broad context to select the best possible action. There are several options that can be analyzed:

– A first option is to use an existing graph database server running as an agent and use its API to modify and query the graph. The database should allow for multi-graphs with a variable number of attributes in nodes and edges and work with low latency and high throughput. Also, the model checking functionality that filters candidate updates would have to be coded outside the server. We have not made tests with currently available graph databases like Neo4j[41] or Sparksee[42] but we expect to be a reasonable option if some latency is allowed.
– A better solution in this line would be a database with a notification service, so changes were automatically propagated to a set of clients. There is at a least one open source database that we now of that provides this capability, RethinkDB [43], but it is a document oriented database and conversion between database types and agents language types will penalize the overall process.
– A second option is to use a communications middleware -Ice in RoboComp- that provides a publication-subscription service. Using a set of topics, all agents would publish their changes and all would receive the updates. In this distributed solution, the graph would not have a central store, but it would exist as a set of local copies. This solution needs a synchronization mechanism distributed in all the agents to guarantee the global coherence of the graph, similar to the ones used in collaborative editing [16] or BASE databases[17][18].
– A third option would be to let the agents push partial or global updates on the graph to a known server agent. This agent would process the updates to guarantee the global coherence of the graph and would publish the new versions back to the agents. A similar solution was proposed in [19] for a distributed scene-graph to be used in shared virtual reality scenarios. Also, this approach is similar to the one used in modern code repositories, such as Git. Each agent works with a local copy of the graph while new updated versions are arriving by subscription to the server. The local management of this flow is responsibility of the local agent until it decides to push the changes up to

the server. After that it receives a confirmation that the changes are valid or a denying response with the error. This is the solution currently implemented in CORTEX. Performance is more than enough for our current needs and comparative tests will be done when the other implementations be available.

In the next section we present a brief formalization of DSR in its current state.

2.1 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 $4x4$ 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 be 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 have to 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 $\{e_1....e_r\}$. An edge e joining the nodes u and v will be expressed as $e = uv$.

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

According to its nature, the properties of symbolic edges are:

1. Given a symbolic edge $e = uv$, we cannot infer the inverse $e^{-1} = vu$
2. A symbolic edge $e = uv$ can store multiple values
3. The set of e is defined as $L = \{e_1, ...e_r, (l_1, l_2, ...l_s)\}$ where $l_i \neq l_j$

On the other hand, according to its geometric nature and the properties of the transformation matrix RT, the characteristics of geometric edges are:

1. For each geometric edge $e = uv$, e is unique
2. For each geometric edge $e = uv = RT$, we can 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. These geometrical relations are showed in Figure 3.

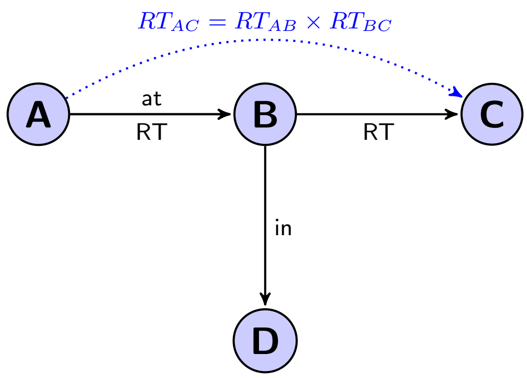


Fig. 3. Unified representation as a multi-labeled directed graph. Edges are labeled "at" and "in" denoting logic predicates between nodes. Also, edges between A,B and B,C have a geometric type of label, "RT" that codes a rigid transformation between them. Geometric transformations can be chained or inverted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

3 Experimental Results

As an initial validation of CORTEX and DSR in a real robot interacting with humans, we tested these ideas in Gualzru [26]. Gualzru is a salesman robot that works autonomously in crowded scenarios and has to step out when a potential client passes by. He 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 separated graphs, one for the kinematic state and one for symbolic attributes and predicates, as many current robotic architectures [20]. Agents injecting data in both graphs at different rates and expecting changes to occur under timeout restrictions, caused unpredictable behavior very hard to debug. This not well understood complexity caused a steady decrease of productivity in the project, to a point where it was difficult to go on. The substitution of the graphs by the new integrated DSR considerably improved the working conditions again. Not all problems were gone and new debugging and monitoring tools are still needed, but communication among agent started to work flawlessly and more complex behaviours are now being explored.

A simple example shows how DSR enables the coordination of several agents in a primitive HR collaboration scenario. DSR provides the agents a common context with multi-modal, semantically distant information, to take the correct decisions. When a person entered the robot's field of view, the *Person* agent would inject a simplified skeleton in the kinematic graph at the right position relative to the floor. Perceptive updates on this representation were performed smoothly as long as the person remained in view and all agent could access that cognitive object. At the same time, other agent *Dialog* was trying to maintain a conversation with the person following steps of a plan hold in the symbolic graph. This agent will keep talking under the condition that the person is paying attention, which is computed as a simple function of some person-robot relational parameters such as presence, proper distance, and face and eyes correctly detected. Those parameters were also being computed by the first agent *Person* and injected as attributes in the symbolic graph.

The existence of an integrated representation also helps to the redesign of the software architecture. For instance, one important drawback of Gualzru 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 Figure 4). 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. The screen is controlled by a specific agent, but shares information with the rest of agents on the framework, such as the Dialog one. Thus, it was easy that this screen displayed what the robot is saying. The screen also allows the person to answer to the robot by touching the desired response on the screen. This information, although captured by the Screen agent, is injected in the graph and made available to the rest, so the agent in charge of the ASR/Dialog can use it to complete missing data. It is important to note that these concepts can be updated by the agents at interaction rates.

The ADAPTA project, which gave birth to Gualzru and the advertisement scenario, started in 2012 and many versions and options of the current DSR have been evaluated since then. The last demonstration tests will be held in October 2015 a will show us if the DSR graph is able to the sustain the whole architecture at human interaction rates.

4 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

- solving the synchronization problem;
- endowing the full kinematic tree with symbolic information; and

Fig. 4. The Gualzru robot interacting with people at the University of Malaga.

– 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. Also, the whole graph is kept synchronized among the agents by using an efficient publishing mechanism.

Future work will focus on injecting raw data directly in the graph and let the agents build on it more abstract representations. The processing schema that we propose admits the inclusion of active perception strategies by mixing top down -planned- and bottom up -reactive- trends through the agents interaction with the DSR. Also, we plan to exploit the hierarchical structure in the graph to optimize the communication mechanism by, for example, allowing temporal subscriptions to specific parts of the representation -e.g. the person or the robot arm. It is also needed to evaluate the computational effort associated to the management of graphs such as the one in Figure 1. Although initially the number of nodes/arcs may be relatively small, the inclusion of raw data in the leaves, of new spatial structures discovered during navigation or new predicates relating logical attributes, might introduce delay or throughput problems affecting the overall performance.

Acknowledgments

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

References

1. T. Foote, “tf: The transform library,” in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, ser. Open-Source Software workshop, 2013, pp. 1–6.
2. S. Blumenthal, H. Bruyninckx, W. Nowak, and E. Prassler, “A scene graph based shared 3d world model for robotic applications,” *2013 IEEE International Conference on Robotics and Automation*, pp. 453–460, 5 2013.
3. I. G.-V. L. R.-R. P. B. L. C. L. M. A. S. A. B. J. B. P Bustos, J Martnez-Gómez, “Multimodal interaction with loki,” in *Workshop of Physical Agents*, 2013, pp. 1–8.
4. S. Rusell and P. Norvig, *Artificial Intelligence: A Modern Approach (3rd Edition)*. Pearson, 2009.
5. D. Poole and A. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, 2010.
6. L. J. Manso, “Perception as stochastic sampling on dynamic graph spaces,” Ph.D. dissertation, 2013.
7. L. V. Calderita, P. Bustos, C. Suárez Mejías, F. Fernández, and a. Bandera, “Therapist: Towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children,” *7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, vol. 1, pp. 374–377, 2013.
8. “Sdf xml definition format.”
9. S. Oliver G., “pandamonium : A paradigm for learning,” in *Proceedings of the Symposium on the Mechanization of Thought Processes*, 1959, pp. 511–529.
10. A. Newell, “Some problems of basic organization in problem-solving programs,” Tech. Rep., 1962.
11. L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, “The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty,” *ACM Computing Surveys*, vol. 12, no. 2, pp. 213–253, 1980.
12. B. Hayes-Roth, “A blackboard architecture for control,” *Artificial Intelligence1*, vol. 26, no. 3, pp. 251–321, 1985.
13. J. W. McManus, “Design and analysis of concurrent blackboard systems,” Ph.D. dissertation, 1992.
14. D. D. Corkill, “Blackboard systems,” *AI Expert*, vol. 6, no. 9, 1991.
15. “http://www.openscenegraph.com.”
16. M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski, “Convergent and commutative replicated data types,” *Bulletin of the European* , no. 104, 2011.
17. V. Balegas, C. Ferreira, R. Rodrigues, M. Shapiro, N. Preguic, and M. Najafzadeh, “Putting consistency back into eventual consistency,” in *EuroSys'15*, 2015.
18. D. Pritchett, “Base: An acid alternative,” *Queue*, no. June 2008, 2008.
19. M. Naef, E. Lamboray, O. Staadt, and M. Gross, “The blue-c distributed scene graph,” in *Proceedings - Virtual Reality Annual International Symposium*, 2003, pp. 275–276.

20. S. Lemaignan, R. Ros, L. Mosenlechner, R. Alami, and M. Beetz, "Oro, a knowledge management platform for cognitive architectures in robotics," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3548–3553, 10 2010.
21. Hofland, K., Jørgensen, A.M., Drange, E. and Stenstr. A Spanish spoken corpus of youth language, Corpus Linguistics. 2005.
22. Frederick Jelinek, Statistical methods for speech recognition, MIT Press. 1997.
23. 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.
24. https://git-scm.com/
25. 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)
26. 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)
27. Manso L.J.: Perception as stochastic sampling on dynamic graph spaces, Ph.D. dissertation, Univ. of Extremadura, Spain (2013)
28. Tomasello, M., Carpenter, M., Call, J., Behne, T., Moll, H.: Understanding and sharing intentions: The origins of cultural cognition. Behavioral and Brain Sciences 28 (5), 675-691 (2005)
29. Bauer, A., Wollherr, D., Buss, M.: Human-robot collaboration: a survey, Int. Journal of Humanoid Robotics (2007)
30. 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 ICAPS 09) (2009)
31. Beetz, M., Jain, D., Msenlechner, L., Tenorth, M.: Towards performing everyday manipulation activities. Robotics and Autonomous Systems (2010)
32. Alami, R., Chatila, R., Clodic, A., Fleury, S., Herrb, M., Montreuil, V., Sisbot, E.A.: Towards human-aware cognitive robots, AAAI-06, Stanford Spring Symposium (2006)
33. Wintermute, S.: Imagery in cognitive architecture: Representation and control at multiple levels of abstraction, Cognitive Systems Research 1920, 129 (2012)
34. Ali, M.: Contribution to decisional human-robot interaction: towards collaborative robot companions, PhD Thesis, Institut National de Sciences Appliquées de Toulouse, France (2012)
35. Clark, A.: An embodied cognitive science?, Trends in Cognitive Sciences 3(9)(1999)
36. Holland, O.: The Future of Embodied Artificial Intelligence: Machine Consciousness?. Embodied Artificial Intelligence, 37-53 (2004)
37. L. J. Manso: Perception as Stochastic Sampling on Dynamic Graph Spaces, PhD Tesis, University of Extremadura, Spain (2013)
38. http://wiki.ros.org/urdf
39. http://www.existentialprogramming.com/
40. https://www.khronos.org/collada/
41. http://neo4j.com/
42. http://sparsity-technologies.com/
43. http://rethinkdb.com