

Incorporating a Reusable Human Robot Interface with an Auction Behavior-Based Robotic Architecture

Bradford A. Towle Jr.

Department of Computer Science and Engineering
College of Engineering, University of Nevada, Reno
Reno, NV, USA
towle@cse.unr.edu

Monica Nicolescu

Department of Computer Science and Engineering
College of Engineering, University of Nevada, Reno
Reno, NV, USA
monica@cse.unr.edu

Abstract - Service robots have the potential of improving the quality of life and assist with people's daily activities. Such robots must be capable of performing multiple tasks and schedule them appropriately while interacting with people over long periods of time. In addition, the robots have to deal with potentially unknown users, handle requests that may have (critical) time constraints and perform in dynamic environments while effectively addressing all the requests received. This paper demonstrates the use of the Auction Behavior-Based Robotic Architecture (ABBRA) in order to develop effective service robots. The proposed approach has the following contributions: i) it enables long-term operation of robots and their interaction with known and unknown users, ii) it handles multiple user requests while dealing with potentially critical time constraints, iii) it provides a reusable interface based on ABBRA, which can run on multiple platforms and iv) it supports flexible interactive capabilities such as requesting that the user wait in order to complete a time sensitive task. The proposed system was demonstrated on two physical robotic platforms: the Adept Mobile's Pioneer 3Dx™ and the Segway RMP®.

Keyword-component; Robotics; Integrating HRI to Architecture; Behavior Arbitration; Auction Mechanism; Behavior-Based;

I. INTRODUCTION

While robotic technologies have greatly advanced in recent years, wide spread deployment of service robots has still not been achieved. Given the complexity of robotic challenges, a lot of research is focused on developing robots that perform a single task very well. Such robots simply shutdown after the task is complete and their tasks often have limited interaction with humans, as robots usually view the humans as obstacles more than opportunities for interaction. However, service robots need to possess long-term autonomy and continuously be prepared for to interact with people in the environment. In addition, service robots must also remain alert for human interaction while performing an assigned task. This means the robots must interact with multiple, different people for a prolonged period of time while still keeping a high level of performance.

This paper presents several contributions. By applying a previously developed robot control architecture (ABBRA) to a human-robot interaction (HRI) problem, it allows for long term robot operation and interaction with known and unknown users. The robot can receive and handle multiple user requests and, using ABBRA, will handle them in the most efficient manner. If there are any time constraints on the requests,

ABBRA allows the robot to handle them appropriately. In addition, the auction-based robotic architecture is integrated with a reusable HRI interface. The components for the interface were chosen based on the control architecture and can run on multiple robotic platforms demonstrating the reusability of the interface. The system also provides for flexible interactive capabilities: if the robot is engaged in a time critical task, the robot requests a new user to wait until the task is handled. Along with this contribution, the robot can deal with a user that is not interacting with it correctly. Two robotic platforms were used to validate the system: Adept Mobile's Pioneer 3Dx™ and the Segway RMP®.

This paper is divided into the following sections: Section II provides the motivation and general approach, Section III gives the related works, Section IV provides the design and methodology, Section V presents the results from a proof of concept test, and Section VI concludes the paper.

II. MOTIVATION AND RELATED WORK

Currently, approaches to HRI come from two different schools of thought: task-centric [1, 2] and human-centric [2] design. Task-centric HRI methods typically have robots that interact in the context of a specialized application such as search and rescue [3-6], specialized interfaces or teleoperation [7]. This school of thought often considers various scenarios for robotic applications [8, 9] and deals with the application requirements for task completion when designing the HRI modules. This can result in specialized systems where the interaction is unique to the situation. Examples include space exploration [10] and search and rescue applications [3-6]. It is also important to mention that for specific applications a teleo-operated system is more appropriate [7]. Application-centric design pursues new interfaces for humans and robots to interact in order to improve the performance of the robotic system for the specific application. These novel interfaces include haptic devices [11], novel environmental indicators [12], gesture recognition [13, 14], tangible interaction [15], and vocal interaction [16, 17]. Because this area focuses on novel interaction, very little focus is given to the reusability of the human robot interface.

The application-centric design method is often based on a specific application or research initiative and results in a specialized approach to human-robot interaction. This means that once the robot is started, it will perform a certain task

until completion and then turn off. This does not allow for a prolonged period where humans can interact with the robot and often only people with robotics expertise can effectively work with the robots. This design philosophy often considers high performance to be more important than the ease of interacting with the robot. Therefore, a steep learning curve can accompany an interface designed from application requirements.

The second broad area of HRI consists of socially aware or emotional robotics. Approaches in this field could be seen as relying on a specialized interface, which is designed from a human-centric perspective [2]. This philosophy takes into account the needs of the human user before considering the application or performance of the system. Instead, this area of research focuses on what class of users will be interacting with the system [18] and what emotional needs for the user are required. Similar to application-centric design, this philosophy also leads to specialized applications. However, these applications are focused mostly on the specifics of the interaction with the users, instead of the completion of a certain task. For example, robotic therapy has become an emerging field. Originally, care providing robotics was limited to aiding the elderly and disabled [19]. However, now robots can provide care and therapy for autistic children [20], rehabilitation patients [21], and even provide an element of psychological care [22]. Another relevant field is that of robotic pets [23], where researchers observe the affects of children playing with robotic animals versus their biological counter-part. Another major application is to analyze how robots react to human emotions [24-26] and how humans emotionally react to robots [27-31]. The primary focus of this research is to promote emotional, social and psychological acceptance of robots in society.

The CoBot research project is most similar to our work [32], with a major focus on planning and navigation for indoor service robots. The CoBot robot uses a collaborative control schema, with which the robot requests help from the user if it detects that it cannot complete the desired goal [33, 34]. ABBRA uses an auction behavior-based system for dynamic task allocation for individual robots. This allows the robot to determine which task is most important to run at that moment based on several constraints, including the possibility of accepting or denying an interaction with a human based on its current time constraints. This is in contrast with the approach used by CoBot, whose scheduling algorithm creates a conflict-free plan, which frees the robot from dealing with interruptions from users [35]. Furthermore, CoBot does not deal with direct interaction with people, but instead follows a schedule that was generated from requests coming from the web. This work was also extended to explore interesting research regarding robotic teamwork and planning [36]. In one approach [37] robots provide tours and decide who should do what in their tour based on previous knowledge. However, robots are not designed to handle un-cooperative users or direct interaction with people, apart from following the robot on the tour. In contrast, the work described in this paper deals

with direct and sometime unexpected interaction with human users.

Several existing approaches mention of the need for reusability in HRI interfaces. However, these papers focus on reusability in one of two methods: either they refer to the need of reusing either the robotic architecture [38, 39] or the component of an HRI interface [40]. Both of these philosophies are good, however, in these cases reusability not the main concern of the work

III. APPROACH

The research in this paper addresses the human robot interaction problem from the perspective of integrating a control architecture with an HRI interface. The Auction Behavior-Based Robotic Architecture (ABBRA) has already demonstrated that it can provide a robust action selection mechanism for service robots [41-43].

ABBRA allows behaviors to compete for control of an actuator. This is accomplished by each behavior collecting metrics from the environment and participating in an auction. Each behavior is assigned an activation level based on the metrics it submitted. The auction cycle is running continuously, thus allowing ABBRA to handle changes and noise in the environment and handle new requests being added during runtime. The robot can also handle tasks with time constraints and determine if a task has enough time to finish and how important the behavior is to run at that moment. ABBRA is designed to be independent of specific applications instead; it is designed to arbitrate between sets of generic behaviors. This makes ABBRA an ideal base to design a generic interface. For more information on the ABBRA architecture see [41-43].

ABBRA is designed to run on a generic robot platform, with no particular need for specific hardware. This design feature, coupled with the hardware abstraction provided by the Robotic Operating System (ROS) [44], allows ABBRA to run on multiple platforms. Since the robotic control architecture is a common factor between robots it is an ideal place to start designing a reusable HRI interface. Instead of defining specialized tasks or a set of social awareness requirements, the building blocks used to create the interface were derived from the basic components of ABBRA (Such as activation level, behavior metrics, and task management). After these components were defined, a more human-centric presentation was considered. This allowed the interface to be designed with the same generalities that were built into ABBRA and the capability of running on different robotic platforms.

Along with reusability, the ability to handle time constraints was a capability that was desired for the system. The interaction with users is handled as a process identical to those used for executing service tasks. Since the architecture dynamically determines which process to run, if critical time constraints exist, the robot may not choose the human interaction request as the highest priority. When this occurs the robot will ask the user to please wait. Unfortunately, this means the robot must deal with non-cooperating users [45] or

a user that does not interact with the robot appropriately. Just because the robot asks the user to wait doesn't mean they will. Should the robot encounter a non-cooperating user it will attempt to avoid the person but stop if it cannot avoid a collision.

This work makes several assumptions. First, is that in order to implement a reusable interface, all potential robotic platforms using the proposed approach should be running ABBRA as control architecture for the robots. Second, the robot in the proposed system is autonomous except for the times when the user is requesting tasks or information. Therefore, once the user makes his/her selection the robot will continue without the need of human intervention. Third, the user is not interested in raw data (sensory or motor control). In some robotic applications, such as space exploration, the user will be interested in every piece of data collected. However, for service robots it is more likely that the user is more interested in having the robot complete tasks than analyzing raw data. Although issues regarding emotional and socially aware robotics have significant importance in HRI, this paper will focus more on the practical approach in which the user is not as interested in connecting emotionally to the robot, but rather in getting the robot to complete a desired task. However, several socially aware concepts were used in the interface to aid the robot in better communicating with the user. No assumption is made regarding the cooperation of the user, except for common sense when dealing with machinery.

IV. DESIGN AND METHODOLOGY

To enable scalability, portability and to rely on users knowledge of operating graphical user interfaces, a GUI was chosen to enable a user's interaction with the robot. The GUI was implemented using the PyQt library, which can run on both Windows and Linux platforms. This is useful since most mobile devices can run stripped down versions of Linux, which means a GUI that can run under Linux can run on most tablets and even a few smart phones. Thus, the interface was designed to be functional across multiple robotic platforms with multiple robotic tasks because the same underlying control architecture, ABBRA is used. This architecture, via ROS, can run on a wide variety of robotic platforms. Since reusability was a high priority, primary components of the architecture were used as the building blocks for the GUI. ABBRA has provides the following functionality: the addition of new tasks, the competition of currently running tasks, and the collection of the overall system status. The interface thus contains three major components, mapped to this functionality: i) input, ii) system status, and iv) facial expression. The input portion of the interface is only displayed when the user requests an interaction by looking directly at the camera. If a person is in front of the camera, the robot will detect the face (using Haar Feature-based Cascade Classifier for Object Detection in OpenCV [46]) and know that a user is requesting to interact with it. The face detection system considers that a face of an interested person is in the field of view if the bounding box surrounding the face is larger than a threshold. This is to remove influences from

people farther away from the robot. This simple method for starting the interaction provides a proof of concept that the interface can be used by different people.

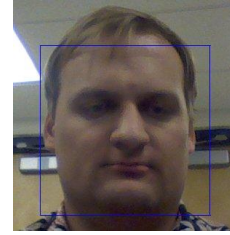


Figure 1: The face detector tells the robot the user wishes to interact.

When the robot detects a face and determines it has time to interact with the user the GUI will display all tasks capable of being performed by the robot on the left side of the screen and allow the user to either run or cancel the task depending on its status.

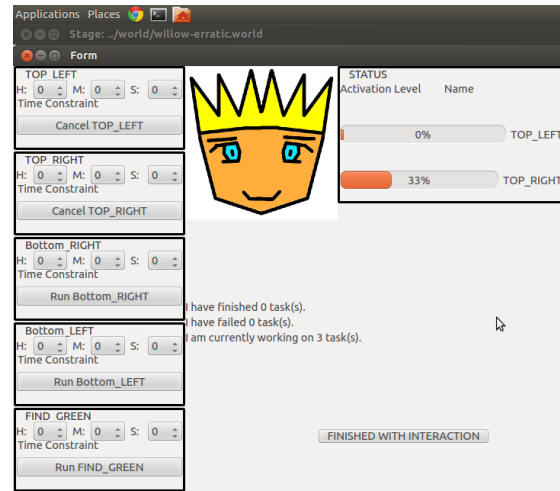


Figure 2: Interface in interactive mode- The left panels are available tasks the robot can run (or are currently running). The panel on the right displays the active tasks and their activation levels. In this example the robot can run five different tests.

Each task has its own screen area and provides three drop down menus along with a button to run the tasks. If the task is currently running, the button will be used for cancellation of the task. The three drop down menus allows the user to select a time constraint for the tasks they wish the robot to run. The need for a time constraint was also derived from ABBRA, allowing every potential task to have a time constraint. Eventually, this interface could be run on a smart phone or tablet (using a text box can cumbersome for these devices), thus drop down menus were used to input the time constraints. After a task is requested by the user, the label on the button for that task will change signifying press if the user wants to cancel the task. This gives the user complete control of invoking and cancelling tasks that the architecture allows. Ideally, a configuration file for each specific task could be used for the creation of custom parameters besides time

constraints. However, for the purposes of this paper only the time constraint was used as a user input parameter.

The primary metric for ABBRA is a behavior's activation level, which is used to determine which tasks should control a specific actuator. This metric is displayed as a status bar with the appropriate task name next to it on the right side of the screen. This displays what the priority of the tasks should be from the robots perspective. By displaying the activation level as a status bar, the user could automatically know what task the robot thought was most important. The status box would be visible when the robot is moving or when the robot is in an interactive mode. This means the user does not have to be interacting with it to see this information.



Figure 3: Non-interactive mode

If the activation level does not provide enough information to the user, the mouse tooltip, text displayed when you leave the mouse over an object, will provide more information about the specific task.

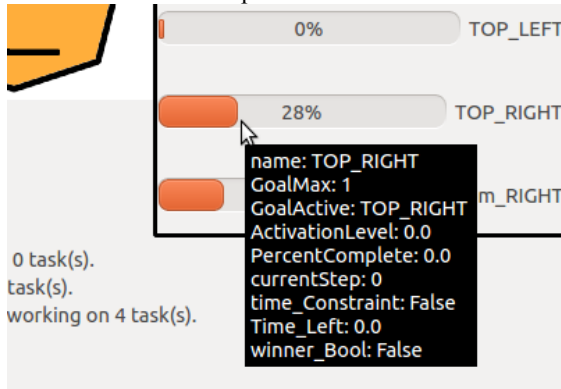


Figure 4: Tooltip providing more detailed information.

This demonstrates that the interface is capable of relaying low-level information if the human user desires it.

The third section of the interface is the facial expression, which is displayed in the middle of the GUI. This consists of a picture portraying a facial expression with a message underneath it. Humans can understand facial expressions faster than they can comprehend large amounts of data scrolling on a terminal. Therefore the GUI takes behavior data from the robot's current set of running behaviors and determines what facial expression is more relevant for that data. The goal is not to introduce emotion into the robotic system for human acceptance, but rather to portray the status

of the system through facial expressions. A large amount of work has already demonstrated that mimicking a face with cartoonish features allows human acceptance because it avoids the "uncanny valley" [47-49].

For the research presented in this paper a simple proof of concept was used for visual feedback from the controller. The robot would use five primary facial expressions representing the overall status of the robot. These facial expressions represented upset, sad, unhappy, normal (straight faced), happy, cheerful.



Figure 5: Above from left to right - upset, sad, unhappy, normal (straight faced), happy, cheerful

For proof of concept, simple values taken from the architecture itself were used to calculate the facial expression for the robot. These values included number of active tasks, the number of tasks a robot can have active at once, the number of completed task and the number of failed tasks. The stress and the confidence of the robot were calculated as follows

$$Stress = 1 - \frac{N_{at}}{N_t} \quad (1)$$

N_{at} = Numer of active task

N_t = Total Number of tasks robot can do at once

$$Confidence = \frac{N_c}{N_c + N_f} \quad (2)$$

N_c = Number of completed tasks by the robot

N_f = Number of failed tasks

$$E_v = \text{round} \left(\left(\frac{(Stress + Confidence)}{4} \right) * 10 \right) \quad (3)$$

E_v = Emotional value between Inetegers [0:5]

The resulting value would then be mapped directly to the facial expression where 0 corresponds to upset and 5 corresponds to being cheerful. There is one other facial expression used when the robot asks the user to wait while it finishes a tasks, and when the user simply leaves the robot interface and does not interact with it within 45 seconds.

These facial expressions allow the interface to quickly send a message to the human user, instead of sending pages and pages of scrolling text. Beneath the facial expression is the general message. Most of the time it will display the number of tasks completed, running, and failed. However, should the robot need to ask the user to wait or if another important message occurs, the robot will display the message as well.

V. RESULTS

The testing phase consisted of a complex scenario demonstrating interactions with humans, requests for

achieving of goals, and people interrupting the robot's performance. These scenarios provided a proof of concept for designing the HRI GUI from the architecture. The facial expression simply allows the user to have an immediate understanding the status of the robot [47-49]. The experiments have been performed with both a Pioneer and a Segway RMP robot.

The environment had three objectives with known location that the robot could be requested to reach. These objectives were identified as the yellow, orange and red goal, marked with colored paper in the UNR Computer Science and Engineering building. A fourth objective consisted of finding and investigating a bright green object with an unknown location, which was identified as the green goal. Two users interacted with the robot, one of which is an author of the paper and another, an undergraduate student with the Computer Science and Engineering Department. The second user only needed brief instructions on using the interface after which she was able to fully interact with the robot.

The robot used a map of the engineering building on campus. It used the AMCL module from ROS [50] to keep localization and the Move_Base module [51] to plan routes to objectives with known locations.



Figure 6: The map of the scenario used to test the HRI module

For the testing scenario, a user initiates testing at the start location (lower right corner in *Figure 6*) and requests the robot to move to the yellow goal (Basic goal request). The robot is allowed to accomplish the goal without interruption. Once the robot reaches the yellow goal, it stops and waits for the user to initiate another interaction (demonstrating continuing, long term operation). This time the user requests the robot to move to the orange goal. However, shortly after the robot begins moving, the user attempts to interrupt it (demonstrating interruption of a task with non-critical time constraint). Since the orange goal is not time critical it allows the user to interact with it. This demonstrates that the robot will allow the user to interrupt it when it is not pressed with a time critical task. The user then request the robot find and investigate a green object (green goal). The robot then reaches the green and orange goal. The user will then request an interaction with the robot and ask it to go to the red goal with a 5-minute time constraint. This time constraint was critical enough to not allow the user to interrupt it. Shortly after the robot starts moving toward the red goal, the user will interrupt it and try to request an interaction (User interaction request when the robot has a time critical tasks). The robot asks the user to wait and then avoids the user and continues on to the red goal. This demonstrates

that the robot can deny interaction with the user in time critical situations. Once the robot reaches the red goal, the user attempts to interact with the robot but does not assign any new task to the robot. The robot will then time out the interaction, demonstrating that it can deal with users who do not actually want to request a service.



Figure 7: The users with the Pioneer robot and the Segway RMP

The experiment used the Pioneer robot. The results demonstrate that the architecture and the HRI interface worked as expected, as the robot appropriately handled the task requests and the interactions with the. The robot received two false positives on the face detection, but the overall interaction went as expected.

Figure 8 shows the order in which the tasks were handled by the robot (as given by the auction mechanism in ABBRA), including the tasks representing the interaction with the users.

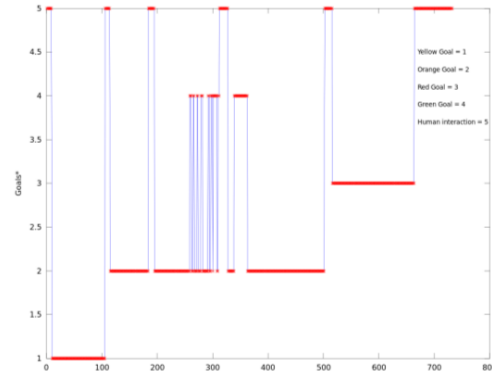


Figure 8: The winning task over time for Pioneer test.

When the scenario begins, the first task that requests control is the human interaction with the robot. The user requests the robot to go to the yellow goal and allows it to complete the task. The human then requests another interaction and tells the robot to go to the orange goal. The user then interrupts the robot again and requests the task of finding a green goal. Given that the orange goal had a known location, the auction mechanism chooses to pursue the orange goal. However, around time step 260 the robot detects the green target with an unknown location and the control is switched to finding the green goal. After reaching the green target, the robot switches back to going to the orange location. During this time there are two false positives from the face detection module, which did not affect the overall behavior of the robot. There is also some oscillation due to noise from the blob-tracker, which also did not affect the robot's performance. The reason this oscillation did not affect the performance is because ABBRA

can handle large amount of noise from the environment and still maintain a high level of performance. This noise in the environment demonstrates that the auction mechanism for behavior selection in ABBRA is also robust to noise in the environment. As a last task, the user requests the red goal with a time constraint. Notice that no human-robot-interaction occurred after this even though it was attempted. After the red goal is finished, the robot is left in an interactive state until it times out and continues its wait for a new request for service.

The second test was run on the Segway RMP and followed the same testing scenario. The robot performed as expected, except for a few false positives with the face detection and some oscillation between the green and orange goals, which were again handled gracefully by the robot's control architecture.

Figure 9 details the winning behaviors over time for the Segway run. In this experiment, similar to the run with the Pioneer, the robot does not allow the user to interrupt the red goal because it has a time constraint.

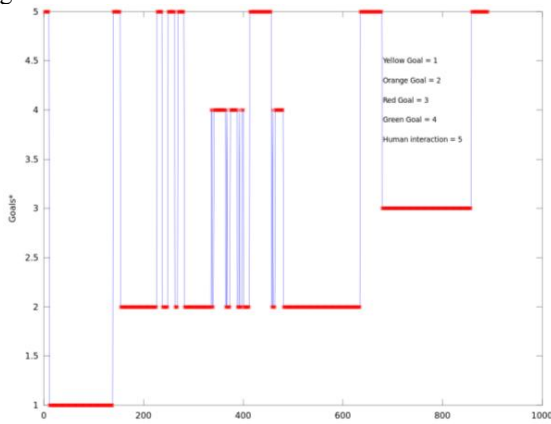


Figure 9: The winning task over time for Segway test.

VI. CONCLUSION

This paper has demonstrated the use of the Auction Behavior-Based Robotic Architecture in developing effective service robots. ABBRA enables long-term operation and interaction with users, while being integrated with a reusable interface that can run on multiple robot platforms. ABBRA allows robots to handle multiple user requests and provides high quality performance for task selection based on what task is most critical to finish first. This includes requesting the user to wait for a time critical task to finish before interacting. The experimental evaluation also shows that the robot can handle a scenario where the user does not cooperate. The proposed system was demonstrated on two different robotic platforms. The results show the potential of this architecture for developing service robots that can operate over extended periods of time in the presence of people in dynamic environments.

REFERENCES

- [1] A. Steinfeld *et al.*, "Common metrics for human-robot interaction." pp. 33-40.
- [2] J. A. Adams, "Critical considerations for human-robot interface development." pp. 1-8.
- [3] M. Baker *et al.*, "Improved interfaces for human-robot interaction in urban search and rescue." pp. 2960-2965 vol. 3.
- [4] J. Casper, and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 33, no. 3, pp. 367-385, 2003.
- [5] R. R. Murphy, "Human-robot interaction in rescue robotics," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 2, pp. 138-153, 2004.
- [6] J. L. Burke *et al.*, "Moonlight in Miami: Field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise," *Human-Computer Interaction*, vol. 19, no. 1-2, pp. 85-116, 2004.
- [7] T. Fong, C. Thorpe, and C. Baur, "Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays, and remote driving tools," *Autonomous Robots*, vol. 11, no. 1, pp. 77-85, 2001.
- [8] H. A. Yanco, and J. L. Drury, "A taxonomy for human-robot interaction." pp. 111-119.
- [9] H. A. Yanco, and J. Drury, "Classifying human-robot interaction: an updated taxonomy." pp. 2841-2846 vol. 3.
- [10] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431-447, 2006.
- [11] S. Katsura, and K. Ohnishi, "Human cooperative wheelchair for haptic interaction based on dual compliance control," *Industrial Electronics, IEEE Transactions on*, vol. 51, no. 1, pp. 221-228, 2004.
- [12] C. M. Humphrey, and J. A. Adams, "Compass visualizations for human-robotic interaction." pp. 49-56.
- [13] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, no. 2, pp. 151-173, 2000.
- [14] J. M. Rehg, and T. Kanade, "Digiteyes: Vision-based hand tracking for human-computer interaction." pp. 16-22.
- [15] C. Guo, and E. Sharlin, "Exploring the use of tangible user interfaces for human-robot interaction: a comparative study." pp. 121-130.
- [16] C. L. Sidner, C. Lee, and N. Lesh, "The Role of Dialogue in Human Robot Interaction," *Mitsubishi Electric Research Laboratories*, 2003.
- [17] A. Billard, K. Dautenhahn, and G. Hayes, "Experiments on human-robot communication with Robota, an imitative learning and communicating doll robot."
- [18] J. Scholtz, "Theory and evaluation of human robot interactions." p. 10 pp.
- [19] P. T. Szemes *et al.*, "Guiding and communication assistant for disabled in intelligent urban environment." pp. 598-603 vol. 1.
- [20] K. Dautenhahn, and I. Werry, "A quantitative technique for analysing robot-human interactions." pp. 1132-1138 vol. 2.
- [21] Z. Bien *et al.*, "Integration of a rehabilitation robotic system (KARES II) with human-friendly man-machine interaction units," *Autonomous robots*, vol. 16, no. 2, pp. 165-191, 2004.
- [22] A. V. Libin, and E. V. Libin, "Person-robot interactions from the robopsychologists' point of view: the robotic psychology and robototherapy approach," *Proceedings of the IEEE*, vol. 92, no. 11, pp. 1789-1803, 2004.
- [23] G. F. Melson *et al.*, "Robots as dogs?: children's interactions with the robotic dog AIBO and a live Australian shepherd." pp. 1649-1652.
- [24] T. Shibata, T. Tashima, and K. Tanie, "Emergence of emotional behavior through physical interaction between human and robot." pp. 2868-2873 vol. 4.
- [25] M. A. Salichs *et al.*, "Maggie: A robotic platform for human-robot social interaction." pp. 1-7.
- [26] C. Bartneck, and J. Forlizzi, "A design-centred framework for social human-robot interaction." pp. 591-594.

- [27] R. C. Arkin *et al.*, "An ethological and emotional basis for human-robot interaction," *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 191-201, 2003.
- [28] J. Goetz, S. Kiesler, and A. Powers, "Matching robot appearance and behavior to tasks to improve human-robot cooperation." pp. 55-60.
- [29] M. L. Walters *et al.*, "The influence of subjects' personality traits on personal spatial zones in a human-robot interaction experiment." pp. 347-352.
- [30] P. H. Kahn *et al.*, "What is a human?-toward psychological benchmarks in the field of human-robot interaction." pp. 364-371.
- [31] C. Bartneck, and J. Forlizzi, "Shaping human-robot interaction: understanding the social aspects of intelligent robotic products." pp. 1731-1732.
- [32] M. Veloso, "CoBot Robots," <http://www.cs.cmu.edu/~coral/projects/cobot/>.
- [33] S. Rosenthal, M. Veloso, and A. K. Dey, "Is Someone in this Office Available to Help Me?," *Journal of Intelligent and Robotic Systems*, vol. 66, no. 1, pp. 205, 2012.
- [34] S. Rosenthal, and M. Veloso, "Mobile Robot Planning to Seek Help with Spatially-Situated Tasks."
- [35] M. Veloso *et al.*, "Symbiotic-Autonomous Service Robots for User-Requested Tasks in a Multi-Floor Building," 2012.
- [36] C. Agüero, and M. Veloso, "Transparent multi-robot communication exchange for executing robot behaviors," *Highlights on Practical Applications of Agents and Multi-Agent Systems*, pp. 215-222, 2012.
- [37] A. Hristoskova, M. Veloso, and F. De Turck, "Personalized Guided Tour by Multiple Robots through Semantic Profile Definition and Dynamic Redistribution of Participants," *Proceedings of the 8th International Cognitive Robotics Workshop at AAAI-12, Toronto, Canada*, 2012.
- [38] Z. Kulis *et al.*, "The distributed control framework: a software infrastructure for agent-based distributed control and robotics." pp. 1329-1336.
- [39] C. Martin *et al.*, "A new control architecture for mobile interaction-robots." pp. 224-229.
- [40] J. Xavier, and U. Nunes, "Reusable Software Components in a Human Robot Interface."
- [41] B. A. Towle Jr, and M. Nicolescu, "Applying dynamic conditions to an auction behavior-based robotic architecture," *Int'l Conf. Artificial Intelligence (ICAI'11)*, vol. Volume 1, no. July 18-21, 2011, pp. 6, 2011.
- [42] B. A. Towle, and M. Nicolescu, "Fusing multiple sensors through behaviors with the distributed architecture," in 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Salt Lake, Utah, 2010, pp. 115-120.
- [43] B. Towle, and M. Nicolescu, "Real-world implementation of an Auction Behavior-Based Robotic Architecture (ABBRA)," in 2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), 2012.
- [44] W. Garage, "ROS | Willow Garage," <http://www.willowgarage.com/pages/software/ros-platform>.
- [45] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Safety evaluation of physical human-robot interaction via crash-testing." pp. 217-224.
- [46] O. CV, "Haar Feature-based Cascade Classifier for Object Detection," http://opencv.willowgarage.com/documentation/python/objdetect_cascade_classification.html.
- [47] M. Blow *et al.*, "The art of designing robot faces: Dimensions for human-robot interaction." pp. 331-332.
- [48] F. Delaunay, J. de Greeff, and T. Belpaeme, "Towards retro-projected robot faces: an alternative to mechatronic and android faces." pp. 306-311.
- [49] K. Dautenhahn *et al.*, "KASPAR—a minimally expressive humanoid robot for human-robot interaction research," *Applied Bionics and Biomechanics*, vol. 6, no. 3-4, pp. 369-397, 2009.
- [50] ROS.org, "AMCL," <http://www.ros.org/wiki/amcl>.
- [51] ROS.org, "MOVE_BASE," http://www.ros.org/wiki/move_base.