CMAssist: A RoboCup@Home Team

Paul E. Rybski, Kevin Yoon, Jeremy Stolarz, Manuela Veloso

CMU-RI-TR-06-47

October 2006

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

CMAssist is a new effort in our lab, the CORAL research group[1], to study the research issues surrounding human-robot interaction in unmodified indoor home and office environments. Our focus is on methods for interaction between the human and the robot that are similar to natural human-human interactions. In 2006, we placed 2nd out of 11 teams in the first ever RoboCup@Home[2] international human robot competition competition where we have demonstrated our initial efforts.

[1] http://www.cs.cmu.edu/∼coral
[2] http://www.robocupathome.org

Contents

1 Introduction

The field of human-robot interaction (HRI) is developing very rapidly as robots become more capable of operating with people in natural human environments. For robots to be accepted in the home and in workspaces, people will need to be able to interact with them in a natural and familiar fashion. Robotic sensing, cognitive, and actuating capabilities will need to achieve a certain level of complexity such that humans can treat them more as teammates or partners in order for the research community to reach this goal. Such enabling capabilities include the ability to recognize the presence and activities of nearby people, possess a spatial and semantic notion of the shared environment, and understand (a subset of) natural human languages. By allowing robots to behavior and interact more socially with and around people, we believe that they will more readily be accepted by non-technical individuals as part of their daily lives and routines.

CMAssist is a new effort by our lab to study human robot interaction from the standpoint of mobile robotic assistants/partners. To this end, we have developed two mobile testbeds that we are using to experiment with interaction technologies. We are primarily focused on issues that involve interactions directly between a human and a robot and have focused our attentions on human detection, speech understanding, and rudamentary dialog processing. In 2006, we participated in the first ever RoboCup@Home international human robot competition competition where we have demonstrated our initial efforts.

2 System Overview

2.1 Hardware

The CMAssist robots (shown in Figure 1) were initially based on the ER1 mobile robot platform from Evolution Robotics. However, due to limitations in the motors and structural components, mobility is provided by a custom set of DC motors, and additional structural hardware was obtained directly from the company 80/20, which manufacturers the aluminum x-bar materials used for the robot's internal structure. The robots have a CAMEO [6] omnidirectional camera rig mounted on the top of their sensor mast. They are also equipped with Videre Design's STH-MDCS2-VAR variable-baseline stereo camera rig using lenses with an 80 degree lateral field-of-view. Infrared range sensors are also placed around the base of the robots. The ER1 arm (Figure 3) is a 1-DoF gripper actuator and is the only manipulator attachment for the CMAssist robots. Computational power is provided by two Pentium-M laptops running Linux. A third Pentium laptop running Windows relays voice input captured by a Shure wireless microphone to the NAUTILUS[3], natural language understanding system written by the Naval Research Labs (NRL) [7, 4, 5]. A Logitech joystick can be used to assume manual override control of the robot at any time. An emergency stop button provides further safety by cutting the power to the motors.

[3] http://www.nrl.navy.mil/aic/im4/Nautilus.php

Figure 1: The robots of CMAssist 2006: Erwin and Carmela. The CMAssist team placed 2nd at the 2006 RoboCup@Home competition.

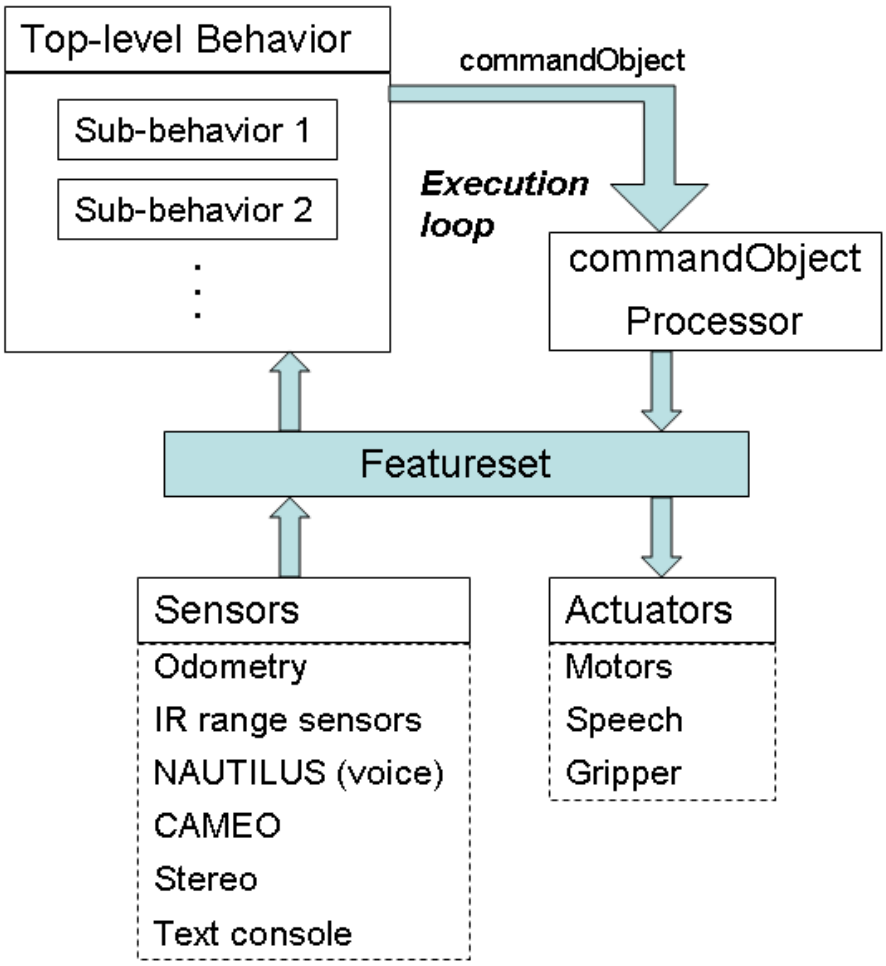


Figure 2: CMAssist Software Architecture

2.2 Software Architecture

Figure 2 provides a high-level view of the software architecture. The computationally-intensive vision modules (CAMEO and stereo vision) were developed in C++ and the NAUTILUS speech processing software was written in CommonLISP. The object-oriented nature of Python made it the language of choice for development of the high-level modular behaviors. The behaviors were also the most frequently modified and tested component of the CMAssist system making the interpreted programming lan-

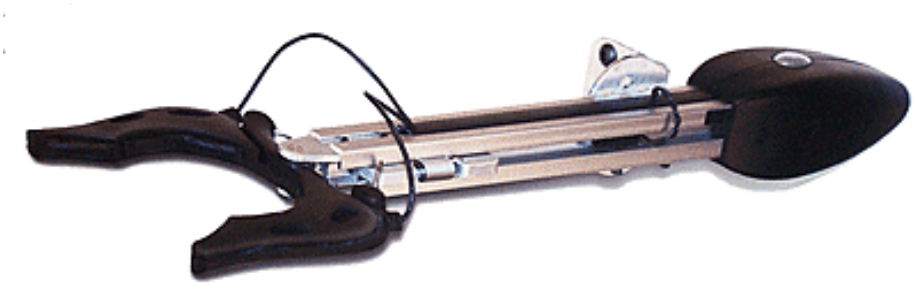


Figure 3: ER1 Gripper Arm

guage a sensible choice. The following sections describe in further detail the components shown in Figure 2.

2.2.1 Sensors and Actuators

Both sensors and actuators are typically created as servers so that multiple clients can share resources.

A single Python server connected to the motor board acts as both a sensor - delivering **odometry** information - and actuator - receiving linear and rotational velocity commands for the **differential drive motors**. **Infrared range sensors** placed around the base of the robot also relay information through this server and are used as virtual bumpers.

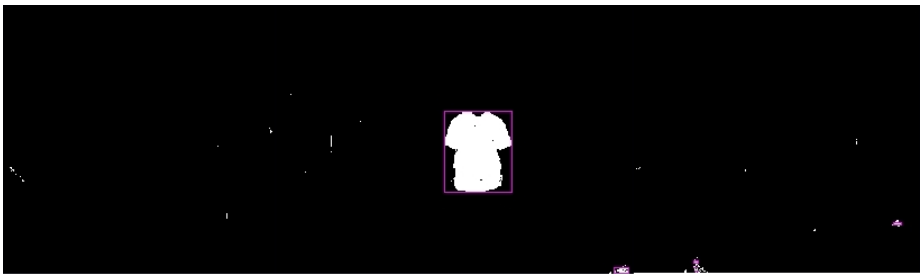
The **CAMEO** omnidirectional camera rig is primarily used for object detection using color histograms [1]. These color histograms are manually trained on desired targets (including people). The image locations of matching objects are streamed over a dedicated color channel port. Up to 16 color channels can be created. See Figure 4. The front camera is also used for landmark identification by visually recognizing objects with clusters of SIFT [3] features (Figure 5).

The C++ **stereo vision** server was built on top of SRI's Small Vision System and is used primarily for obstacle avoidance. A top-down local occupancy map is built from the planar projection of 3D range data. In combination with a particle filter, this data is used for localization. The detection of prominent "blobs" (marked with a white '+' in Figure 6) is also used to aid in person tracking.

The stereo vision server streams a compressed form of the information it gathers. In the top-down occupancy map that is generated, the camera is positioned at the top-center looking down towards the bottom of the image (Figure 6(right)). Two bars (30cm in width) drop down from the top and stop at the closest obstacle. The distance of these two bars from the top of the image give a measure of the distance to the obstacles the robot is approaching and can be used by an obstacle avoidance behavior in deciding which way to turn.

The stereo server is also responsible for maintaining an estimate of the robot's pose in the environment. A particle filter [2] employing a ray-tracing beam model is used

(a) Color blob search on color image



(b) Binary image displaying color blob segments

Figure 4: CAMEO color segmentation.

along with an *a priori* known map of the environment for localization.

Speech recognition is achieved through Naval Research Labs **NAUTILUS** [7, 4, 5] natural language understanding system. Built on top of IBM's ViaVoice, grammar and vocabulary definitions constrain the utterances that NAUTILUS is permitted to recognize. This increases the chances of proper utterance recognition and reduces the likelihood of recognizing a phrase that was never said. A CommonLISP voice server wraps the NAUTILUS software and outputs recognized text to all listening clients.

Besides voice, simple text is CMAssist's other major mode of command input. The **text console** client is regarded as a sensor in the framework of this system and can be used to give any commands that can be given by voice in addition to other low-level commands.

The main method by which the robot communicates with its human users is through **speech**. Cepstral's (www.cepstral.com) Text-to-Speech software is wrapped in a Python speech server which utters text sentences sent to it by high-level behaviors. Erwin is endowed with the Cepstral David voice while Carmela speaks with the Cepstral Callie voice.

2.2.2 Featureset: Sensor-Actuator interface

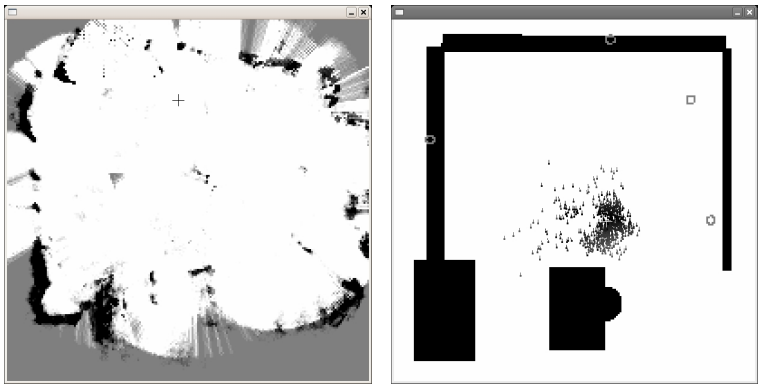
The *featureset* (written in Python) is a contained module that provides a single interface to all sensors and actuators for any behavior that may need to use them. It takes as input a configuration file that specifies the IPs and ports of all the sensor/actuator

(a)

(b)

Figure 5: By recording its SIFT features into a database, the cabinet in (a) can be identified in other images like (b) where it is marked with a white line.

Figure 6: Stereo vision: color image from stereo image pair (left), calculated depth map (center), projected top-down occupancy map (right).



(a) (b)

(c)

Figure 7: (a) The environment as seen by the stereo camera. (b) Particle filter localization using an *a priori* map before a landmark is observed. (c) The same particle filter estimate after a landmark is observed and an update occurs.

server/clients. It then creates servers or clients that connect to them and internally stores sensor data as it is streamed from the servers so that data requests from the Featureset are unblocked.

2.2.3 Behaviors

Behaviors compose the intelligence of CMAssist. Based on sensor data a behavior receives via the featureset, it decides how to drive the actuators in a way that best accomplishes its intended purpose.

A behavior takes as input the *featureset* and a *commandObject*. The commandObject contains instructions on how to drive the actuators. Depending on the behavior, this object may be modified before it is returned. In addition to returning the commandObject, every behavior also outputs a *result* status flag that indicates its status to its parent behavior. Within every execution loop, the commandObject returned by the top-level behavior is processed by a *commandObjectProcessor* that in turn makes the appropriate changes to the featureset (e.g. sending a velocity command). Direct writes to the featureset are not permitted within the behaviors themselves. This keeps the state of the robot static within an execution loop

The behavior architecture is modular in that behaviors can execute other behaviors as sub-behaviors. The manner in which multiple behaviors are combined is at the discretion of the behavior developer. Within the top-level behavior, however, a subsumption scheme prioritizes the commands given by the different sub-behaviors. The obstacle avoidance behavior, for instance, would have higher priority than the person-following behavior when it comes to driving the motors. Another behavior that sends a phrase to the speech server, while of lower priority, would not be overridden by either of the previous behaviors since they are driving different actuators.

Finite State Machines (FSM) are also a common mechanism of activating sub-behaviors when their execution in a defined sequence makes more sense than a priority queue.

These are a couple of examples of the behaviors in the CMAssist repertoire:

The **Follow** behavior searches the CAMEO image stream for the largest rectangular blob with the general proportions of a human torso and maintains a specified distance to the perceived object. Stereo data is used to obtain accurate range distance as well as to add robustness: the robot does not attempt to follow a color blob if it is not perceived by the stereo sensors.

The **ObstacleAvoidance** behavior is the highest priority behavior. If the current motion trajectory will result in a collision with obstacles detected by either the stereo-derived occupancy map or the IR sensors, this behavior overwrites it with a command that maneuvers the robot around the obstacle in a manner that minimizes its deviation from the current trajectory.

3 Task Training

For a robot to be general useful in a home or office environment it would be ideal if it could be taught new tasks when necessary. To this end we implemented a task training system for the CMAssist robots, which can be taught either by dictation or imitation.

We define a task as a set of instructions to be executed by the robot. These instructions can be anything from the set of commands the robot understands, including other tasks. Therefore it is possible for tasks to become arbitrarily complex as a new task can refer to multiple previously-trained tasks.

Tasks are stored in text files with each task step written in the same form as the command would be spoken. This allows one to easily review stored tasks as well as to create new ones if so desired.

3.1 Training by dictation

To begin training the robot by dictation, the user says, "When I say *x*", where *x* is a phrase (eg. "dinner is ready"), and then optionally "do the following." Anything the user says after this will be considered a command, with the robot giving feedback if it does not understand. When finished training, the user can either say "Thank you," ending the training behavior, or he/she can ask "Is that understood?" The latter case will result in the robot repeating the task sequence. The user then states whether the robot correctly recorded the steps, and, if it did not, repeats the sequence of steps the robot is supposed to carry out.

3.2 Training by imitation

Sometimes it may be more desirable to teach by "showing," having the robot take in contextual information when recording the task rather than by "saying". To start training by imitation, the user says, "Let me show you what to do when I say *x*," again where *x* is some phrase. The robot then invokes the Follow behavior (see above) and follows the person around. Whenever the person says something, it records what he/she said and where, reasoning that what was said must be said at that location (i.e. location is the context). If the robot does not understand what was said, it notifies the user. This continues until the user says "thank you," terminating the follow and training behaviors. When executing a task trained in this manner, the robot goes to each location it had heard the user say something and repeats whatever was said.

4 Robocup@Home

RoboCup@Home is a new league in the RoboCup competition which is a radical departure from the traditional robot soccer leagues. Instead of playing a game in this competition, the robots must perform a series of tasks that challenge them to interact effectively with humans in a natural indoor home environment. The competition is broken into two parts, each of which has a distinct focus. The RoboCup@Home competition is useful because it provides a standard domain on which to evaluate the us-

ability and reliability of robots that will ultimately be used in the home. The RoboCup philosophy is to help foster research by providing a standardized platform on which to test problems.

In the initial technical challenge, the robots must demonstrate their abilities to exist in human environments and interact naturally with people. Tasks that the robots can attempt to demonstrate include following humans, navigating the environment, understanding natural human language, and manipulating small objects. These tasks were essentially boolean where a team either succeeded or failed at the task. Points were awarded evenly to each of the succeeding teams out of 1000 total points possible for that round. Thus, if 2 teams succeed, each would receive 500 points. In addition to the pre-set tasks, the competitors can elect to participate in an open challenge whereby they demonstrate a novel aspect of HRI to a jury of technical judges. In the open challenge event, all teams are ranked by relative performance. The scores are assigned by assigning the top-ranking team more points out of the 1000 total points than the other teams, the second-place team gets more points than the third place team, and so on and so forth.

The five competitors that received the highest scores in the initial technical round advanced to the final round. In the final round, the robots are required to demonstrate an interesting HRI capability to a panel of judges and are then evaluated on the aesthetics, ease of use, and overall quality of the robot and its software. CMAssist placed second in the final. The following sections describe the tests of the competition and how CMAssist fared in each.

4.1 Follow Human Test

This test has two phases. In the first phase the robot is to follow a team member by any means necessary (including putting a beacon of some sort on the team member). In the second phase the robot must follow a judge who wears nothing other than his or her normal clothes.

CMAssist successfully completed the first phase of the test, following a team member wearing a bright blue shirt. However, we were unable to complete the second phase, as the judge was wearing a shirt of similar color to other objects in the environment. CMAssist was one of seven teams to succeed in phase 1 of this challenge and earned 142.9 points. No teams succeeded in phase 2 of this test.

4.2 Navigation Test

This test also had two phases. In both phases, the robot had to successfully navigate between different points in the environment using natural language commands like "Go to the couch" as input. In the first phase these commands could be typed into a keyboard, and the teams chose the waypoints. However, in the second phase the robot had to understand voice commands, and the judges chose the waypoints.

CMAssist successfully completed the first phase using speech input, but failed to complete the second phase when the robot failed to detect an obstacle and collided with it. CMAssist was one of three teams to succeed in phase 1 of this challenge and earned 333.3 points. No teams succeeded in phase 2 of this test.

4.3 Manipulation Test

In this test the robot must successfully manipulate an object such as a newspaper, can, or door. The newspaper was to be grabbed from the floor and carried a few meters, the can was to be grabbed from a refrigerator, and the door was to be opened, driven through and shut.

CMAssist was the only team that attempted this test, though it was intended only as a proof of concept as to how one might approach retrieving a newspaper. The robot successfully drove up to and grabbed the newspaper, but given the extremely limited capabilities of the ER1 gripper arm attachment, the newspaper was folded in a certain way to facilitate grasping and dragged along the ground. CMAssist was the only team to succeed at phase 1 of this test and earned 1000 points. No team succeeded in phase 2 of this test.

4.4 Open Challenge

For the open challenge event, in which teams demonstrate a particular HRI capability of their robot to technical judges, CMAssist chose to show the robots' task training through imitation capability (see Task Training section). The robot was shown what to do when "dinner is ready" was said, driving to various parts of the house announcing to people to come to dinner. When told to, it successfully completed the trained task. CMAssist finished first in this challenge and earned 444 points.

4.5 Final Challenge

At the end of the technical challenge, CMAssist led with 1920 points. The next highest score in the technical challenge was 698 points. The top five teams from the technical challenge advanced on to the finals.

In the final challenge, robots again showed an aspect of HRI, but to a panel of judges where the abilities were judged purely on non-technical means. That is, aesthetics, ease of use, price point, and time to commercialization were categories that the robots were judged by. Additionally, all points are reset to zero for each of the different teams that actually make it to the final round. Thus, the final rankings are made solely based on the scores achieved by the teams in their final round performance.

For this event, CMAssist exhibited a scenario in which the robot was pre-trained to act as a butler or valet, answering the door and offering entertainment to an arriving guest while the host was busy finishing preparing a meal in the kitchen. CMAssist received second place in the final.

5 Future Work

The CMAssist research effort has many goals on the horizon. A recently acquired Hokuyo URG-04LX laser range finder remains to be integrated into the robots. Both the laser and SIFT markers will be used for improved localization.

We are also in the process of expanding the dialogue capabilities of the robot. Task training is being extended to cover conditional clauses so that the robot could be told

Figure 8: RoboCup@Home arena: kitchen view 1

to do something only when some environmental condition is satisfied (e.g. “If Jeremy is there, say hi to Jeremy”).

In an effort to make interaction with our robots more intuitive and pleasant, both aesthetically and functionally, we are collaborating with members of Carnegie Mellon’s School of Design.

6 Acknowledgments

We would like to thank the Naval Research Labs for developing the NAUTILUS natural language understanding system and for their assistance in getting it to work on the CMAssist platforms.

References

[1] J. Bruce and M. Veloso. Fast and accurate vision-based pattern detection and identification. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taiwan, May 2003, to appear.

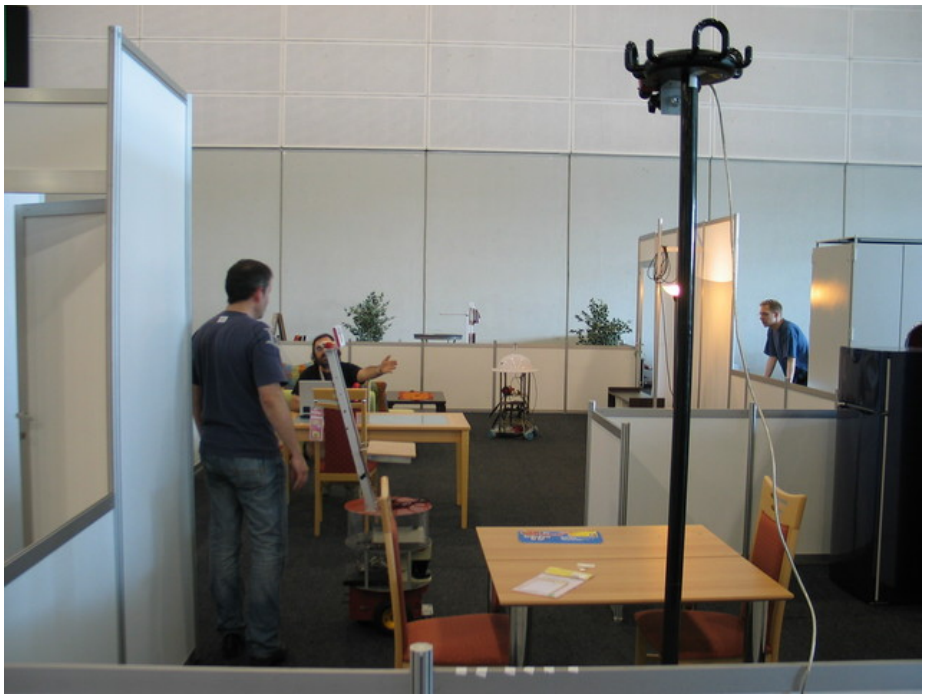


Figure 9: RoboCup@Home arena: kitchen view 2

[2] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pages 1322–1328, 1999.

[3] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1150–7, 1999.

[4] D. Perzanowski, A. Schultz, W. Adams, and E. Marsh. Goal tracking in a natural language interface: Towards achieving adjustable autonomy. In *Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation: CIRA '99*, pages 208–213, Monterey, CA, November 1999. IEEE Press.

[5] D. Perzanowski, A. Schultz, W. Adams, E. Marsh, and M. Bugajska. Building a multimodal human-robot interface. *IEEE Intelligent Systems*, 16(1):16–21, January/February 2001.

[6] P. E. Rybski, F. de la Torre, R. Patil, C. Vallespi, M. M. Veloso, and B. Browning. Cameo: The camera assisted meeting event observer. In *Proceedings of the 2004*

Figure 10: RoboCup@Home arena: living and dining rooms

IEEE International Conference on Robotics and Automation, New Orleans, April 2004.

[7] K. Wauchope. Eucalyptus: Integrating natural language input with a graphical user interface. Technical Report NRL/FR/5510-94-9711, Naval Research Laboratory, 1994.

Figure 11: RoboCup@Home participants