From an autonomous soccer robot to a robotic platform for elderly care

João Cunha, Eurico Pedrosa, Cristovão Cruz, António J.R. Neves, Nuno Lau,
Artur Pereira, Ciro Martins, Nuno Almeida, Bernardo Cunha, José Luis
Azevedo, Pedro Fonseca and António Teixeira

Transverse Activity on Intelligent Robotics, IEETA / DETI University of Aveiro,
Aveiro, Portugal
{joao.cunha, efp,cac, an, nunolau, artur, ciro.martins, nuno.almeida,
mbc, jla, pf, ajst }@ua.pt

Abstract. Current societies in developed countries face a serious problem of aged population. The growing number of people with reduced health and capabilities, allied with the fact that elders are reluctant to leave their own homes to move to nursing homes, requires innovative solutions since continuous home care can be very expensive and dedicated 24/7 care can only be accomplished by more than one care-giver.

This paper presents the proposal of a robotic platform for elderly care integrated in the Living Usability Lab for Next Generation Networks. The project aims at developing technologies and services tailored to enable the active aging and independent living of the elderly population. The proposed robotic platform is based on the CAMBADA robotic soccer platform, with the necessary modifications, both at hardware and software levels, while simultaneously applying the experiences achieved in the robotic soccer environment.

1 Introduction

Current societies in developed countries face a serious problem of aged population. The growing number of people with reduced health and capabilities, allied with the fact that elders are reluctant to leave their own homes to move to nursing homes, requires innovative solutions since continuous home care can be very expensive and dedicated 24/7 care can only be accomplished by more than one care-giver.

Technology directed to Ambient Assisted Living can play a major role in improving the quality of life of elders, enabling and fostering active aging without leaving their homes. In the context of this scenario, the introduction of a mobile robotic platform could be an asset, by complementing and enhancing the deployed infrastructure. A robot can be a mobile monitoring agent, by providing images from spots that are occluded from the house cameras, as well as helping to reduce the feeling of loneliness that often affects the elderly, when endowed by means of human interaction.

Ever since the first robots were created, researchers have tried to integrate robots in our daily lives. In particular, domestic assistants have been a constant driving goal in the area, where robots are expected to perform full daily chores in a home environment. While some simple forms of domestic robots, such as vacuum cleaner robots, are increasingly becoming part of our everyday life, robots designed for human care are far from commercialization.

Meanwhile, a large number of this type of robots have been developed over decades by academies and research groups. The results and insights obtained through the conducted experiences will undoubtedly shape the care robots of tomorrow in fields such as Face Recognition, Speech Recognition, Sensor Fusion, Navigation, Manipulation, Artificial Intelligence and Human-Robot Interaction to name a few.

This chapter presents the proposal of a robotic platform for elderly care based on a robotic soccer platform, with the necessary modifications, both at hardware and software levels, while simultaneously applying the experiences achieved in the robotic soccer environment.

The Institute of Electronics and Telematics Engineering of Aveiro (IEETA), a research unit of the University of Aveiro, Portugal, have been developing, for many years, a significant activity in the context of mobile robotics. One of the most visible projects that has resulted from this activity is the CAMBADA [1] robotic soccer team .

The CAMBADA project provided vast experience in areas such as Distributed Architectures [2], Machine Vision [3,4,5], Sensor Fusion [6,7], Multi-Robot Cooperation [8,9], to name a few. This experience is reflected in the series of positive results achieved in recent years. The CAMBADA team won the last four editions of the Portuguese National Championship, placed second in the 2010 European Championship and placed third in 2009 and 2010 World Championship while winning the world title in 2008.

There is no better proof of the successful application of soccer robots in home environments than the RoboCup@Home[1] league. This league was created in 2006 from the need to place more emphasis on real world problems, not addressed in robotic soccer [10]. This league is currently the largest league of the RoboCup initiative and includes a vast number of teams that started as soccer teams and then evolved to this robotic paradigm [11].

As stated before, the goal of this chapter is to present a mobile autonomous robot designed to improve the quality of life of an elderly person in a household environment. In order to achieve this goal, the robot should be able to perform important tasks, namely be safe for users and the environment, avoid dynamic and static obstacles, receive information from external sensors, execute external orders, among others.

We will present all the parts involving the development of the robot, focusing on the following topics:

- distributed hardware architecture;

---

[1] www.robocupathome.org

- hardware abstraction from the high-level software;
- machine vision algorithms for recognition of objects present in a home environment;
- sensor and information fusion;
- indoor self-localization;
- automatically modeling of the environment and construction of occupancy maps;
- multimodal human-robot interaction;
- robot control and monitoring.

The main contribution of this chapter is to present new advances in the areas described above, regarding the development of a mobile autonomous robot focused on Ambient Assisted Living, taking the example of the adaptation of a soccer robot from the CAMBADA team, developed in the University of Aveiro, Portugal.

2 The Living Usability Lab for Next Generation Networks

The development of a robotic platform for elderly care is part of a broader project named Living Usability Lab for Next Generation Networks[2]. The project is a collaborative effort between the industry and the academy that aims to develop and test technologies and services that give elderly people a quality lifestyle in their own homes while remaining active and independent.

To enable active aging of the elder population, the different services provided make use of the NGN infrastructure that offers large bandwidth transport technologies suitable for services such as image and voice transmission. This feature allows a thorough analysis of the collected data to ultimately generate knowledge of the observed behaviors. This then allows a feedback on the developed services and technologies and provides an opportunity for improvement.

A Living Lab is not just a set of information services but a complex entity composed by physical spaces and infrastructures (information and communication systems and services, peripheral devices, development tools and methodologies for analysis, specification, evaluation, validation and dissemination of the results) and requires intense involvement of stakeholders (whether they are, for instance, end users, professionals, researchers or students) to allow the research and development, in continuum, of new technologies and services, as described in Fig. 1.

3 System Overview

The robotic platform used is based on the CAMBADA robotic soccer platform. The robot has a conical base with radius of 24 cm and height of 80 cm. The

---

[2] www.livinglab.pt

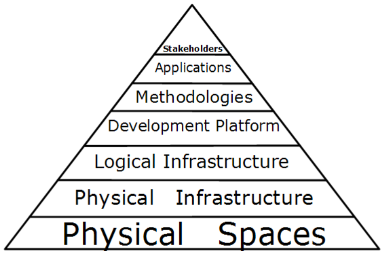


Fig. 1. The conceptual model of the LUL project

physical structure is built on a modular approach with three main modules or layers.

The top layer has the robot vision system. Currently the robot uses a single Microsoft Kinect camera placed on top of the robot pointing forwards. This is the main sensor of the robot. The retrieved information is used for localization and path-planning to predefined goals.

The middle layer houses the processing unit, currently a 13" laptop, which collects data from the sensors and computes the commands to the actuators. The laptop executes the vision software along with all high level and decision software and can be seen as the brain of the robot. Beneath the middle layer, a network of micro-controllers is placed to control the low-level sensing/actuation system, or the nervous system of the robot. The sensing and actuation system is highly distributed, using the CAN protocol, meaning the nodes in the network control different functions of the robot, such as motion, odometry and system monitoring.

Finally, the lowest layer is composed of the robot motion system. The robot moves with the aid of a set of three omni-wheels, disposed at the periphery of the robot at angles that differ 120 degrees from each other, powered by three 24V/150W Maxon motors (Figure 2). With this wheel configuration, the robot is capable of holonomic motion, being able to move in a given direction independently of its orientation.

Following the CAMBADA hardware approach, the software is also distributed. Therefore, five different processes are executed concurrently. All the processes run at the robot's processing unit in Linux.

Inter-process communication is handled by means of a RealTime DataBase (RTDB) [12] which is physically implemented in shared memory. The RTDB is divided in two regions, the local and shared regions. The local section allows communication between processes running in the robot. The shared section implements a Blackboard communication paradigm and allows communication between processes running in different robots. All shared sections in the RTDB are kept updated by an adaptive broadcasting mechanism that minimizes delay and packet collisions.

The processes composing the CAMBADA@Home software are (Figure 3):

a) b)

Fig. 2. CAMBADA@Home hardware system: a) The robot platform. b) Detailed view of the motion system.

- **Vision** which is responsible for acquiring the visual data from the Kinect sensor.
- **Sensorial Interpretation - Intelligence and Coordination** is the process that integrates the sensor information and constructs the robot's worldstate. The agent then decides the commands to be applied, based on the perception of the worldstate.
- **Wireless Communications** handles the inter-robot communication, receiving the information shared by other robots and transmitting the data from the shared section of the RTDB.
- **Lower-level communication handler** or hardware communication process is responsible for transmitting the data to and from the low-level sensing and actuation system.
- **Monitor** that checks the state of the remaining processes, relaunching them in case of abnormal termination.

Given the real-time constraints of the system, all process scheduling is handled by a library specifically developed for the task, the *Process Manager* [1].

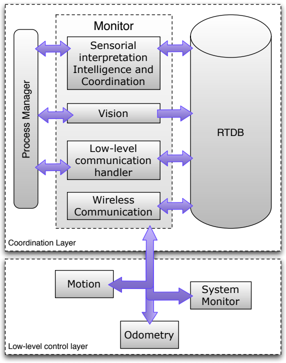
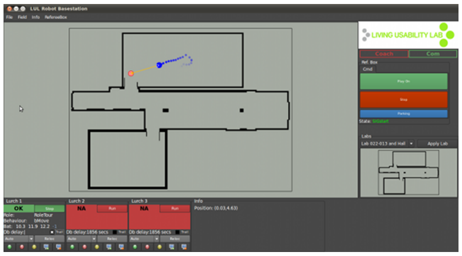


Fig. 3. CAMBADA@Home software architecture.

3.1 Monitoring station

The monitoring station, also known as basestation, has a determinant role both during the development of an autonomous assistant robot capability as well during its application. The basestation is an adapted version of the CAMBADA team basestation [1] taking in consideration a set of requirements that emerge from the development of a service and assistive robot (Figure 4).



State: SIGstart

Lab 022-013 and Hall

Role: RoleTour

Behaviour: bMove

Bat: 10.3 11.9 12.2

Db delay:1856 secs

Position: (0.03,4.63)

Fig. 4. CAMBADA@Home basestation GUI. The GUI is divided in three panes. The lower pane shows the internal state of the robots. The center pane draws the indoor *blueprint* and the robots location. The right pane hold the robots control panel box (e.g. start, stop) and several operational visual flags.

The basestation application provides a set of tools to perform the control and monitoring of the robot. Regarding the control activity, this application allows high level control of the robot by sending basic commands such as *run*, *stop* and *docking*. It also provides a high level monitoring of the robot internal state, namely its batteries status, current role and behavior, indoor self-localization, current destination point, breadcrumb trail, etc.

Furthermore, this application provides a mechanism that can be used to enforce a specific behavior of the robot, for debugging purposes.

4 Perception

Humans rely heavily on vision or vision based abstractions to acknowledge the world, to think about the world and to manipulate the world. It is only logical to empower artificial agents with a vision systems with capabilities similar to the human vision system.

The vision subsystem of this robot is constituted by a single depth sensor, the Kinect, fixed at the top of the robot. It is accessed through the freenect [13] library, and provides a depth and color view of the world. With the depth information, we create a 3D metric model of the environment. Using this model, we then extract relevant environment features for the purpose of localization and navigation, namely the walls and the obstacles, The proposed methods consider that the height and pitch of the camera relatively to the ground plane remain constant, parameters that are set when the system is calibrated.

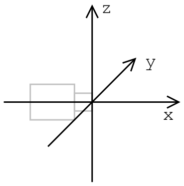


Fig. 5. Camera coordinate system

4.1 Pre-Processing

Both wall and obstacle detection is done using only the depth image, which has 640×480 pixels. The image is subsampled by a factor of 5 in both dimensions, leaving us with a 128×96 image, which has proven to contain sufficient information for wall and obstacle detection. This decision is not based on current time constraints, but was made to account for future project developments.

4.2 Walls

A wall is a structure that delimits rooms. It is opaque and connects the floor to the ceiling [14].

With this simple definition in mind, the approach we follow to identify the walls in the environment is to select all the points with height, relative to the floor, lower than the ceiling height and perform a column-wise search on the remaining points of the image for the point which is farthest from the robot, along the x axis, according to the coordinate system shown in Figure 5. The retrieved points are then used in later stages of processing to allow for robot localization.

4.3 Obstacles

An obstacle is an occupied volume with which the robot can collide.

The obstacle detection method is similar to the one used for wall detection. To detect the obstacles, we reject the points that have an height, relative to the floor, greater than the robot height plus a margin to account for noise, or that belong the floor. The point is considered to be floor if it's height is lower than a threshold, method that proved to be good enough to cope with the noise. We then perform a column-wise search on the remaining points of the image for the point closest to the robot along the x axis.

At the end of this process we have 128 obstacle points that can be used for occupancy belief update.

The algorithms described process the images captured by the Kinect depth camera and extract the visible walls and obstacles. Using solely depth information the robotic agent is able to detect walls without having to perform color calibration or without being susceptible to natural lighting conditions (Figure 6).


a)

b)

c)

Fig. 6. Kinect vision system: **a)** The image captured by the Kinect rgb camera. **b)** The same image captured by the Kinect depth camera. **c)** The 2D vision of the extracted information of the depth camera, the blue points are walls and the red points are obstacles. The robot is placed in the bottom center of the image.

5 Localization

For indoor localization, we successfully adapted the localization algorithm used by the Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture (CAMBADA) team to estimate a robot position in a robotic soccer field. The algorithm was initially proposed by the Middle Size League (MSL) team Brainstormers Tribots [15].

The Tribots algorithm [15] constructs a FieldLUT from the soccer field. A FieldLUT is a grid-like data structure where the value of each cell is the distance to the closest field line. The gradient of the FieldLUT values represents the direction to the closest field line. The robot detects the relative position of the field line points through vision and tries to match the seen points with the soccer field map, represented by the FieldLUT. Given a trial position, based on the previous estimate and odometry, the robot calculates the matching error and its gradient and improves the trial position by performing an error minimization method based on gradient descent and the RPROP algorithm [16]. After this optimization process the robot pose is integrated with the odometry data in a Kalman Filter for a refined estimation of the robot pose.

To apply the aforementioned algorithm in a indoor environment the concept of white line was replaced with the walls. In an initial phase, given a map of the environment, usually the building blueprints, a FieldLUT list is created that contains all possible configurations of seen walls for different positions. By testing a grid of points over the map, a new FieldLUT is created, and added to the FieldLUT list, when a new configuration of seen walls is detected. As the robot moves through the environment, it dynamically loads the FieldLUT corresponding to its position, which should consist of the walls seen in that part of the map.

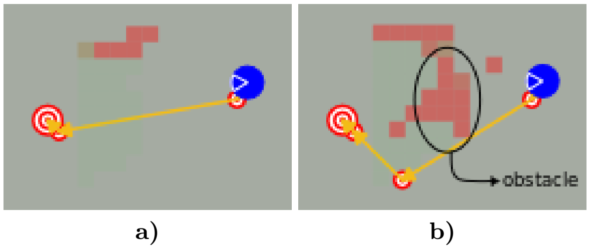
The need to use a set of FieldLUTs instead of a single FieldLUT for the entire map arises from the local minimum problem inherent to gradient descent algorithms. Since the walls in a domestic environment have an associated height which is naturally higher than the robot, from a given point in the environment there is usually a set of walls that are out of the line-of-sight of the robot. This scenario doesn't occur in a robotic soccer field where the lines are co-planar with the field. Therefore using a single FieldLUT could match the wall points extracted from the captured images to unseen walls, resulting in erroneous self-localization.

The described method does not solve the initial localization problem. This is solved by applying the visual optimization process on different trial positions evenly spaced over the known map. To reduce the search space of the initial localization, the initial heading of the robot is given by a digital compass.

6 Navigation

The robotic agent receives a sequence of goal points to go to in a patrolling manner. As it arrives the goal point it decomposes its path to the next goal point in intermediate goal points. The robot navigates between intermediate goal points in a straight line.

The considered metric map is a discrete two-dimensional occupancy grid. Each cell-grid (x, y) has an associated value that yields the believed occupancy. The navigation plan is calculated by a path-finding algorithm supported by the probabilistic occupancy map of the environment. Because of the dynamic nature of the environment the robotic agent uses D* Lite [17], an incremental path-finding algorithm that avoids full re-planning in face of changes in the environment. This methodology enables obstacle avoidance seldom based on incremental path planning (Figure 7).



a) **b)**

Fig. 7. Incremental path planning: **a)** Path planned to achieve the goal point. **b)** A new obstacle appeared in the path of the robot resulting in a re-planned path adjusted to the changes of the environment.

The environment is represented in an occupancy map implemented by the OctoMap library [18]. Although OctoMap is capable of creating 3D occupancy maps, the environment is projected onto the ground plane, thus constructing a 2D occupancy map of the environment. Each change in the environment, tracked by the occupancy map, is reflected in the corresponding cell-grid value used in the path-finding algorithm.

The perceived obstacles (x, y) points (section 4) are used to update the occupancy map. For each obstacle (x, y) point the corresponding (x, y) node of the occupancy map is updated by casting a ray from the robot current point to the target node, excluding the latter. Every node transversed by the ray is updated as free and the target updated as occupied. However a *maxrange* value is set to limit the considered sensor range. If an obstacle is beyond the *maxrange*, only the nodes transversed up to *maxrange* are updated while the remaining nodes remain unchanged, including the target node. However, due to the limited vertical field of view of the Kinect sensor, target nodes where the Kinect sensor can not *see* the floor are updated as occupied without making any assumption about the occupancy of the closer nodes. This is made to prevent the *freeing* of nodes at the Kinect sensor vertical blind region.

7 Human-Robot Interaction

Spoken language is a natural way – possibly the most natural - to control and process human-robot interaction. It has some important advantages: eyes and hands free; communication from a distance, even without being in line of sight; no need for additional learning for humans.

Therefore, we integrated in our mobile service robot some interaction facilities by means of three spoken and natural language processing components: an Automatic Speech Recognition (ASR) component to process the human requests (in form of command-like small sentences), a Text-to-Speech (TTS) component to generate more natural responses from the robot side, and a semantic framework (dialog manager) to control how these two components work together.

The requirements for this spoken and natural language interaction system result from the rulebook of the RoboCup@Home competition. An example of a use-case is the *Follow Me* task where the robot is asked to follow user. In this use-case two command-like sentences are needed: "[Robot's Name] follow me" and"[Robot's Name] stop follow me".

According the use-cases the following requirements for our speech-based interaction system are defined:

- The speech recognition component should be speaker independent, have a small vocabulary, and be context dependent and robust against stationary and non-stationary environmental noise.
- The speech output should be intelligible and sound natural.
- The dialog manager system should be mixed-initiative allowing both robot and user to start the action, provide or ask for help if no input is received or incorrect action is recognized, and ask for confirmation in case of irreversible actions.

In terms of hardware two types of input systems are being tested: a robot mounted microphone and a microphone array framework with noise reduction and echo cancellation. To deal with the high amount of non-stationary background noises and background speech usually present in these interaction environments, a close speech detection framework is applied in parallel to noise robust speech recognition techniques.

Speech recognition is accomplished through the use of CMUSphinx, an Open Source Toolkit for Speech Recognition project by Carnegie Mellon University. Additionally, we are testing speech recognition results obtained by using the Microsoft Speech SDK. For this propose both speaker dependent and speaker independent profiles are being trained, and a specific grammar for command interaction defined, with each command-like sentence preceded by a predefined prefix (robot's name).

For robot speak-back interaction and user feedback, external speech output devices (external speakers) will be used. The speech synthesis component will be implemented by means of a concatenative system for speech output. For that propose, we are testing the Microsoft Speech SDK and the FESTIVAL Speech Synthesis system developed at the Edinburgh University. We are trying to implement some adaptation features like using the information on distance from robot to user to dynamically change the output volume, and changing the TTS rate from normal to slower according to user's age.

These tools are integrated in the Olympus framework [19] (Figure 8), developed at Carnegie Mellon University, which provides a domain independent voice based interaction system. This framework provides the Ravenclaw dialog manager [20], which interacts with an application back-end, in this case the robot software architecture, to perform predefined tasks such as the "Follow me" example.

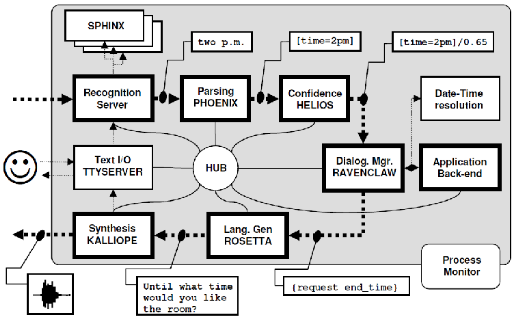


Fig. 8. The Olympus framework.

8 Conclusions

This paper presents the necessary adaptations to enable a mobile robotic platform, based on robotic soccer, to perform daily tasks in a indoor environment, more specifically to an elderly care context. Namely, solutions related to perception in unstructured environments, such as household environments, indoor localization, safe navigation and human-robot interaction were discussed.

In future developments the authors are developing solutions to add on the obtained results so far. Specifically, we are researching on detection and tracking of people in the environment, while applying the developed navigation algorithms, to safely follow a person through a home, or even inverting the roles and provide guidance in a safely manner.

Acknowledgements

This work is part of the COMPETE - Programa Operacional Factores de Competitividade and the European Union (FEDER) under QREN Living Usability Lab for Next Generation Networks (http://www.livinglab.pt/).

References

1. DFBR: 2. In: DFBR. I-Tech Education and Publishing, Vienna, Austria (In Vladan Papic (Ed.), Robot Soccer, 2010)
2. DFBR: Dfbr. In: Proc. of the 12th IEEE Conference on Emerging Technologies and Factory Automation, ETFA2007, Greece (2007) 973–980
3. DFBR: Dfbr. Mechatronics (2010 (in press))
4. DFBR: Dfbr. In: Proc. of the 4th Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA-2009. Volume 5524 of Lecture Notes in Computer Science., Póvoa do Varzim, Portugal, Springer (2009) 80–87
5. DFBR: Dfbr. In: Proc. of the RoboCup 2007. Volume 5001 of Lecture Notes in Computer Science., Atlanta, USA, Springer (2007) 417–424
6. DFBR: Dfbr. In: RoboCup 2009: Robot Soccer World Cup XIII. Lecture Notes in Artificial Intelligence, Springer (2009)
7. DFBR: Dfbr. Mechatronics (2010 (in press))
8. DFBR: Dfbr. In: Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBOTICA'2008, Aveiro, Portugal (2008) 27–32
9. DFBR: Dfbr. Mechatronics (2010 (in press))
10. van der Zant, T., Wisspeintner, T.: Robocup x: A proposal for a new league where robocup goes real world. In Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y., eds.: RoboCup 2005: Robot Soccer World Cup IX. Volume 4020 of Lecture Notes in Computer Science., Springer (2005) 166–172
11. van der Zant, T., Wisspeintner, T.: Robocup@home: Creating and benchmarking tomorrows service robot applications. In Lima, P., ed.: Robotic Soccer, Vienna: I-Tech Education and Publishing (2007) 521–528
12. DFBR: Dfbr. In Aykanat, C., Dayar, T., Korpeoglu, I., eds.: ISCIS. Volume 3280 of Lecture Notes in Computer Science., Springer (2004) 876–886

13. libfreenect homepage: (Available `http://openkinect.org/wiki/Main_Page`) Accessed in February 2011.
14. Moradi, H., Choi, J., Kim, E., Lee, S.: A real-time wall detection method for indoor environments. In: IROS. (2006) 4551–4557
15. Lauer, M., Lange, S., Riedmiller, M.: Calculating the perfect match: An efficient and accurate approach for robot self-localization. In Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y., eds.: RoboCup. Volume 4020 of Lecture Notes in Computer Science., Springer (2005) 142–153
16. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the rprop algorithm. In: Proceedings of the IEEE International Conference on Neural Networks. (1993) 586–591
17. Koenig, S., Likhachev, M.: D* Lite. In: Proceedings of the AAAI Conference of Artificial Intelligence (AAAI), Alberta, Canada (2002) 476–483
18. Wurm, K.M., Hornung, A., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In: Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, Anchorage, AK, USA (2010) Software available at `http://octomap.sf.net/`.
19. Bohus, D., Raux, A., Harris, T.K., Eskenazi, M., Rudnicky, A.I.: Olympus: an open-source framework fro conversational spoken language interface research. In: proceedings of HLT-NAACL 2007 workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology. (2007)
20. Bohus, D., Rudnicky, A.I.: The ravenclaw dialog management framework: Architecture and systems. Computer Speech & Language **23** (2009) 332–361