Autonomous Mobile Service Robots *For* Humans, *With* Human Help, and *Enabling* Human Remote Presence

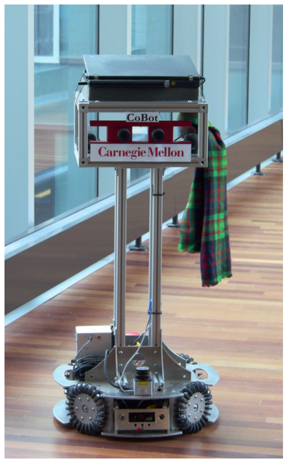
Manuela Veloso, Stephanie Rosenthal, Rodrigo Ventura*, Brian Coltin, and Joydeep Biswas
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

***Abstract*— We present our CoBot mobile service robots, which we aim at being a truly functional contribution to humans in our office environments. The robots perform tasks for humans, proactively assess and ask for the need of human help, and enable human remote presence through mobile telepresence. We introduce and present how we address these multiple roles that humans play with respect to autonomous mobile service robots. We first briefly introduce the CoBot robots, which include robust autonomous localization and navigation. We then focus on presenting how users can request tasks and interact with CoBot, how CoBot reasons about humans that may help it, and how humans can be mobile telepresent on CoBot. CoBot has been reliably and increasingly effectively functioning in our environments for the last two years. We estimate it has autonomously navigated for our task-based tests, in two different buildings, altogether for more than 50km.**

I. INTRODUCTION - CoBot ROBOTS

Many researchers, present authors included, aim at having autonomous mobile robots robustly perform service tasks in our indoor environments. The efforts have been very extensive and successful.[1] We would like to concretely credit two efforts that have more closely motivated our work, namely the Xavier robot at Carnegie Mellon [1] and the RoboCup@Home initiative [2], which provides competition setups for indoor service autonomous robots, with a yearly increasing wide scope of challenges of autonomy and interaction with users.

We follow on those many efforts with the specific goal of concretely deploying such autonomous mobile robots to be tasked by users in our environment. Our environment consists of a nine-floor academic building containing approximately 80 offices per floor in the top four floors. On one floor, for example, there are 35 individual offices for faculty and staff and 44 offices each shared by 2-3 graduate students. The lower floors have fewer offices and are mostly dedicated to classrooms, lounges, a cafe, labs, a three-floor ramp, and a variety of open working areas. We have developed two robots, namely CoBot-1 and CoBot-2, shown in Figure 1. The robots are agile in their navigation due to their omnidirectional bases[2] and can autonomously localize and navigate in an arbitrary office environment, while effectively avoiding obstacles [3]. The robots purposefully include a modest variety of sensing and computing devices, including a vision camera, a Kinect depth-camera, a small Hokuyo LIDAR, a touch-screen tablet, microphones and speakers, as well as wireless communication. In this work, we focus on CoBot-2, which has a laptop with the screen facing forwards, towards the direction of movement, that occupants can use to interact with the robot.


(a) CoBot-1

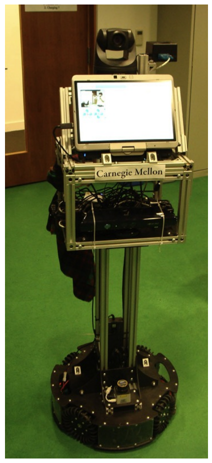

(b) CoBot-2

Fig. 1: Omnidirectional mobile robots for indoor user service.

Following up on our goal to make an agile, inexpensive robot platform with limited onboard computation, CoBot has some limitations connected to its perception, cognition, and action. CoBot has localization uncertainty in large open spaces and also has difficulty perceiving chairs found in common areas resulting in increased navigation time as it attempts to re-localize or avoid these areas entirely. Additionally, CoBot does not have arms or the ability to manipulate objects to push chairs out of the way, press elevator buttons to navigate between floors, or pick up the mail or other objects to give to the building occupants. While the robot can overcome some of these challenges autonomously, CoBot follows a symbiotic human-robot relationship [9] and proactively

This work is partially supported by a grant from the National Science Foundation. The views and conclusions are those of the authors only.

*Visiting Scholar, Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal supported by the Portuguese Foundation for Science and Technology through the CMU-Portugal Program.

[1]We are aware of the extensive list of references in this area, which we are regretfully not able to discuss and include due to space limitations.

[2]The CoBot robots were designed and built by Michael Licitra, mlicitra@cmu.edu, with the base being a scaled-up version of the CMDragons small-size soccer robots, also designed and built by Licitra. The robots have been running extensively since Spring 2009 (CoBot-1) and since Spring 2010 (CoBot-2) without any hardware failures.

assesses that it needs help and asks for help from humans sometimes to resolve each of these limitations, particularly the physical ones. Humans play therefore a helping role to the robot.

The goal of CoBot as a service robot is to perform tasks to users. Concretely deploying the robots to users requires providing an effective way for users to request tasks to be executed by CoBot. We devise a user-friendly web interface that allows for users to reserve the robots for specific tasks. We present three classes of tasks, namely a *Go-To-Room* task, in which the user requests the robot to go to a specific location at some requested time, the *Transport* task, where the user requests the robot to pick up an object at a specified location, and to deliver it to a drop-off location, While these tasks support our presentation, development, and experiments, our architecture is flexible in the definition of new tasks. Humans play therefore a user role, as requesting services from the robot.

A special task that users may request from the robot is mobile *Telepresence*, where the user can remotely operate the robot from the web. CoBot is equipped with a controllable camera and a rich remote web interface. Users can remotely control and zoom CoBot's camera, directing to their visual point of attention, and drive with either directional commands, by clicking on a point on the floor of the camera image, or by clicking on a point in a map. Humans play therefore a remote presence role enabled by the robot.

In the next three sections, we first present how humans request tasks from the robot and how the robot schedules and executes the tasks with its behavior planner. We then present how the robot reasons about human helpers, in terms of the model of their help, and the use of such models to plan its navigation. And then we briefly introduce how humans can be mobile telepresent on CoBot. We finally discuss some experiments and draw conclusions.

II. EXECUTING TASKS FOR HUMANS

The design of an architecture to address our goal to deploy mobile robots to general users, poses several challenges: (1) the task-request interface should be easily accessible and user-friendly, (2) the scheduling algorithm has to take into account navigation times from one location to another, (3) navigation should be safe and reliable for office environments, and (4) human-robot interaction should be intuitive.

A. Requesting Tasks

To address these challenges, we contribute a *Users to Mobile Robots* (UMR) architecture, which interacts with users in two distinct ways: through a web interface, for managing bookings and following the robots state, and directly with the robots through their onboard user interface.

The web-based booking interface addresses challenge (1), to the extent that web-based booking systems are a widespread and familiar scheme for reserving services, being found in numerous services such as in hotel reservations, car rental, and more recently in ZipCar™. Challenge (2) is addressed by a scheduling agent. This agent verifies feasibility of bookings taking into account the locations requested in the tasks, and is capable of proposing a feasible alternative starting time if not. The robust navigation method used addresses challenge (3). It is based on the Kinect depth camera, capable of effectively navigating in an office environment, while avoiding moving and fixed obstacles [4]. The robot's face-to-face interaction with its users is based on a joint synthetic-voice and touch-screen *onboard user interface*. Messages are both spoken by a synthetic voice and displayed on the touch-screen, while the user can respond to these messages using buttons displayed on the touch-screen. This interface is simple and easy to use, thus addressing challenge (4).

To illustrate the functionality of the UMR architecture, we present next a running example of the booking and execution of a task requested by a user:

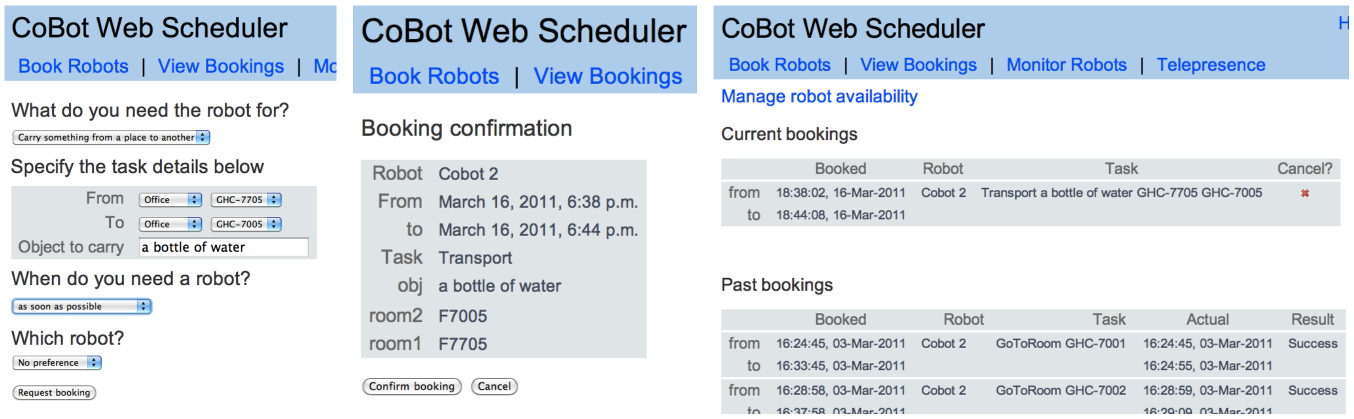
1) At 6:35PM, the user uses a web browser to request a robot to transport a bottle of water, from room 7705 to room 7005, as soon as possible (Figure 2a);
2) The web interface proposes to book CoBot-2 since it is available, estimating its arrival at the pick up location (7705) at 6:38PM (Figure 2b);
3) After confirmation, CoBot-2 starts executing this task immediately: it navigates to room 7705, while displaying and speaking the message "Going to 7705 to pick up a bottle of water and bring it to 7005" on the *onboard user interface*;
4) Upon arrival to 7705, CoBot-2 displays and speaks the message "Please place a bottle of water on me to deliver", and awaits someone to click the 'Done' button displayed on the touch-screen;
5) Once this button is pressed, the robot starts navigating to room 7005;
6) upon arrival to 7005, CoBot-2 displays and speaks the message "Please press 'Done' to release me from my task", and awaits the user to press the 'Done' button;
7) Once this button is pressed, the task is considered successfully executed, and the robot navigates back to its home location.

After the task has been booked, the user can check the booking on the web (Figure 2c), and cancel it if necessary. During the task execution, the user can follow the progress of the robot navigation, either on the map view (Figure 4a) or through the camera view (Figure 4b).

B. Executing Tasks

After a task is scheduled, the executing manager agent sends the robot-specific scheduled task set to the corresponding robot manager agent to execute. The robot's Behavior Interaction Planner plans the sequence of action to complete each task.

Typically, task planners plan only the autonomous actions to complete a task and a separate dialog manager interacts with humans to receive the task requests. However, a robot cannot always perform its actions autonomously and relies on humans in the environment to help it complete tasks. Additionally, as a robot performs actions, humans in the


CoBot Web Scheduler

Book Robots | View Bookings | Mo

What do you need the robot for?

Carry something from a place to another

Specify the task details below

From	Office	GHC-7705
To	Office	GHC-7005
Object to carry	a bottle of water	

When do you need a robot?

as soon as possible

Which robot?

No preference

Request booking

(a) book a robot

CoBot Web Scheduler

Book Robots | View Bookings

Booking confirmation

Robot	Cobot 2
From	March 16, 2011, 6:38 p.m.
to	March 16, 2011, 6:44 p.m.
Task	Transport
obj	a bottle of water
room2	F7005
room1	F7705

Confirm booking | Cancel

(b) confirmation

CoBot Web Scheduler

Book Robots | View Bookings | Monitor Robots | Telepresence

Manage robot availability

Current bookings

	Booked	Robot	Task	Cancel?
from	18:38:02, 16-Mar-2011	Cobot 2	Transport a bottle of water GHC-7705 GHC-7005	✖
to	18:44:08, 16-Mar-2011			

Past bookings

	Booked	Robot	Task	Actual	Result
from	16:24:45, 03-Mar-2011	Cobot 2	GoToRoom GHC-7001	16:24:45, 03-Mar-2011	Success
to	16:33:45, 03-Mar-2011			16:24:55, 03-Mar-2011	
from	16:28:58, 03-Mar-2011	Cobot 2	GoToRoom GHC-7002	16:28:59, 03-Mar-2011	Success
to	16:37:58, 03-Mar-2011			16:29:09, 03-Mar-2011	

(c) view bookings

Fig. 2: Screenshots of the web interface, showing (a) the web interface to perform a booking, (b) the confirmation screen containing the start and (estimated) end times, and (c) the list of current and past bookings performed by the user.

environment may want to know what robot's goals are. Our Behavior Interaction Planner therefore reasons about a robot's incapabilities [5] and human interest in the robot and plans for both human interactions in addition to the autonomous actions. As it executes the plan, it reports back to the server a descriptive message for online users to follow the robot progress in the web interface.

We define actions and interactions that are required to complete a task along with their preconditions and effects. For `ask` interactions, for example, there are no preconditions, the robot `speaks` the defined text, and the effect is the required human response (*e.g.* clicking a 'Done' button on CoBot's user interface). For `navigate` actions, the precondition is that the robot `speak` aloud its new goal to humans in the area, the robot then sends a desired location to the *navigation* module, and the effect is that the robot is in the location that it should navigate to. The separate *navigation* module controls the low level motor control and obstacle avoidance for navigation. Any other actions needed for a task can be defined similarly.

Given a new task, the robot plans the sequence of actions necessary to complete it. For example, in the $\text{Transport}(s, l_p, l_d, m)$ task, the Behavior Interaction Planner plans the following sequence of actions (illustrated in Figure 3) at start time s: `navigate` to location l_p, `ask` for the object m, `navigate` to l_d, and `ask` for task completion confirmation.

The Behavior Interaction Planner can also plan for a robot's incapabilities. For example, if CoBot (with no arms) must navigate between different floors of the building, this requires not only `navigate` actions, but also human interaction to ask for help with pressing buttons and recognizing which floor the robot is on. In these cases, the Behavior Interaction Planner plans:

- `navigate` to elevator,
- `ask` for help pressing the up/down button,
- `navigate` into the elevators,
- `ask` for help pressing the floor number and recognizing that floor,
- `navigate` out of the elevator,
- `navigate` to goal

Upon arriving at goal locations, the robot may also need help picking up objects and plans for these additional `ask` interactions accordingly.

III. HUMANS HELP AS OBSERVATION PROVIDERS

Unlike oracles modeled in OPOMDPs [6], humans in the environment are not always available or interruptible [7], may not be accurate [8], and they may have a high cost of asking or interruption [9]. We formalize these limitations within the POMDP framework. In particular, we will model the probability of a robot receiving an observation from a human in terms of the human's availability and their accuracy to reduce the uncertainty of the robot. A similar formulation can be achieved for increasing capabilities.

1) Location: We assume that humans are located in a particular known location in the environment, and can only help the robot from that location. When the robot is in state s it can only ask for help from the human h_s in the same state. As a result of taking the ask action `ask`, the robot receives an observation o from the human.

2) Availability: The *availability* of a human in the environment is related to both their presence and their interruptibility [10]. We define availability α_s as the probability that a human provides a non-null observation o in a particular state s:

$$0 \le \alpha_s \le 1 \tag{1}$$

If there is no human available in particular state, $\alpha_s = 0$. A human provides observations with probability

$$p(o \neq o_{null} | s, \texttt{ask}) = \alpha_s \tag{2}$$

and would provide no observation o_{null} otherwise

$$p(o_{null} | s, \texttt{ask}) = 1 - \alpha_s \tag{3}$$

Receiving the o_{null} is equivalent to receiving no observation or timing out waiting for an answer. This is to ensure that $\sum_o p(o|s, \texttt{ask}) = 1$.

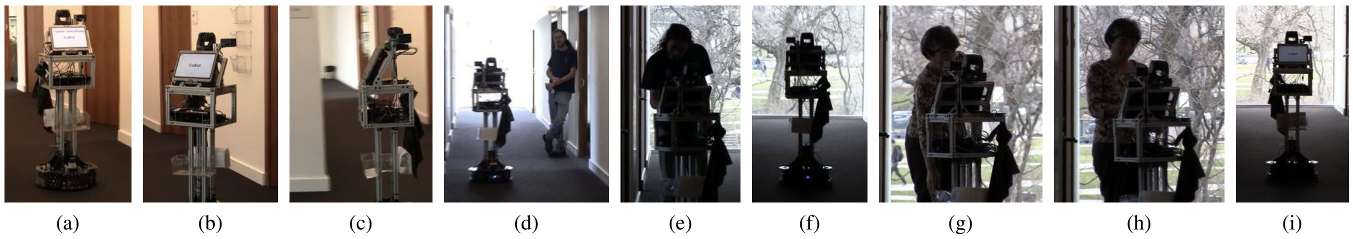


Fig. 3: (a,b,c) After CoBot-2 receives a *Transport* task request, it autonomously navigates to the location l_p to pick up a bottle of water and take it to location l_d. (d,e) Upon arriving to l_p, CoBot-2, asks a person to place the bottle of water and afterwards press 'Done'. (f,g) Then, CoBot-2 navigates to location l_d to deliver the bottle of water. (h,i) When the user presses 'Done', CoBot-2 navigates back to its home location. (The complete video is submitted with this paper.)

3) Accuracy: The non-null observation o that the human provides when they are available depends on their *accuracy* η. The more accurate the human h_s, the more likely they are to provide a true observation o_s. Otherwise, h_s provides observations $o_{s'}$ where s' are states near s in the transition graph.

Formally, we define the accuracy η_s of h_s as the probability of providing o_s compared to the probability they provide any non-null observation $o \neq o_{null}$ (their availability α_s).

$$\eta_s = \frac{p(o_s|s, \texttt{ask})}{\sum_{o \neq o_{null}} p(o|s, \texttt{ask})} = \frac{p(o_s|s, \texttt{ask})}{\alpha_s} \tag{4}$$

4) Cost of Asking: It is generally assumed that supervisors are willing to answer an unlimited number of questions as long as their responses help the robot. However, in active learning, there is a *cost of asking* in terms of the time it takes for them to answer the question and the cost of interrupting them to limit the number of questions asked.

Let λ_s denote the cost of asking for help from h_s. These costs vary for each person, but are assumed to be known before planning. The cost for querying the human if they answer with a non-null observation $o \neq o_{null}$ is

$$R(s, \texttt{ask}, s, o_s) = -\lambda_s \tag{5}$$

However, if the person is not available to hear the question or provide a response, there is no expected cost.

$$R(s, \texttt{ask}, s, o_{null}) = 0 \tag{6}$$

Our reward structure has consequences that affect policy solutions. In particular, the robot does not receive negative reward when it tries unsuccessfully to ask someone for observations so it can afford to be riskier in who it tries to ask rather than incurring a higher cost of asking someone who is more available.

A. HOP-POMDP Formalization

To briefly review, POMDPs are represented as the tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{O}, \Omega, T, R\}$ of states $\mathcal{S}$, actions $\mathcal{A}$, observations $\mathcal{O}$ and the functions:

- $\Omega(o, s, a) : \mathcal{O} \times \mathcal{S} \times \mathcal{A}$ - observation function, likelihood of observation o in state s after taking action a
- $T(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ - transition function, likelihood of transition from state s with action a to new state s'
- $R(s, a, s', o) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{O}$ - reward function, reward received for transitioning from s to s' with action a and observation o

We define the HOP-POMDP as a POMDP for a robot moving in the environment with humans, and then discuss differences between humans as observation providers and noisy sensors.

Let HOP-POMDP be $\{\Lambda, \mathcal{S}, \alpha, \eta, \mathcal{A}, \mathcal{O}, \Omega, T, R\}$. where

- Λ - cost of asking each human
- α - availability for each human
- η - accuracy for each human
- $\mathcal{A} = A \cup \{\texttt{ask}\}$ - autonomous actions and a query action
- $\mathcal{O} = O \cup \{\forall s, o_s\} \cup o_{null}$ - autonomous observations, an observation per state, and a null observation
- $T(s, a_{ask}, s) = 1$ - self-transition for asking actions

Specifically, let h_s be the human in state s with availability α_s, accuracy η_s, and cost of asking λ_s. Our observation function Ω and reward function R reflect the limitations of humans defined in Equations 1-6. The remaining rewards, observations, and transitions are defined as with any other POMDP.

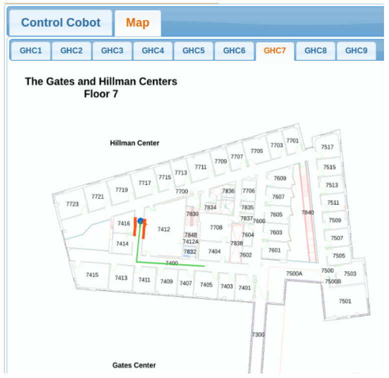
B. Plan Execution

The best HOP-POMDP policy is one in which the robot takes actions that result in low uncertainty or takes actions that leave it in states with a high possibility of a human reducing its uncertainty. As a result, the robot may plan longer paths to navigate in the hallways, but the robot is more likely to navigate with low uncertainty. With lower uncertainty, the robot will navigate faster to its goal locations [9]. Additionally, if the robot is taking paths with a high likelihood of human availability, it can ask these same people to help increasing its capabilities (*e.g.,* pressing elevator buttons).

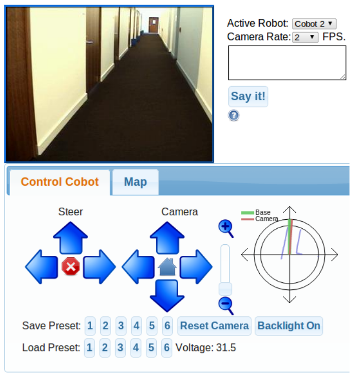
IV. TELEPRESENCE

In addition to performing tasks fully autonomously, users may control CoBot-2 in a semi-autonomous telepresence mode as a *Telepresence* task. In telepresence mode, live sensory information and camera images are streamed and displayed directly in the user's web browser.

The telepresence interface, shown in Figure 4, displays the camera image, the text-to-speech interface, and the controls



(a) map view


Active Robot: Cobot 2
Camera Rate: 2 FPS.
Say it!
Control Cobot
Map
Steer
Camera
Base
Camera
Save Preset: 1 2 3 4 5 6 Reset Camera Backlight On
Load Preset: 1 2 3 4 5 6 Voltage: 31.5

(b) telepresence view

Fig. 4: Screenshots of the telepresence interface, showing (a) the map view of CoBot-2 location with its navigation path, and (b) the robot's camera view, together with camera and robot motion controls.

for both the robot navigation and camera pan-tilt-zoom settings. The telepresence interface provides three control modalities with increasing levels of autonomy, allowing the user to joystick the robot, and select a destination point on the image or on a map. In all modalities, the robot autonomously avoids obstacles. In addition to controlling the robot with the interface buttons, users may click directly on the image to point the camera or to navigate the robot to the point clicked. The interface map displays the robot's current location and orientation, and highlights detected obstacles to help the user to navigate safely. The user may click on the map to send the robot autonomously to a location. We have found that users utilize all of these control modalities depending on the situation.

V. Discussion and Conclusion

We conduct experiments with CoBot every day all the time, to test different types of tasks, the planning using models of human help, and demonstration telepresence tasks. In particular, we have found that CoBot performs very efficiently with a linear relation between the distance traveled and the time to execute a task. Humans on the path of the robot like to interact with the robot, blocking and unblocking the robot's path after the robot explicitly requests "Please excuse me!" The robot safely navigates around walls, chairs,

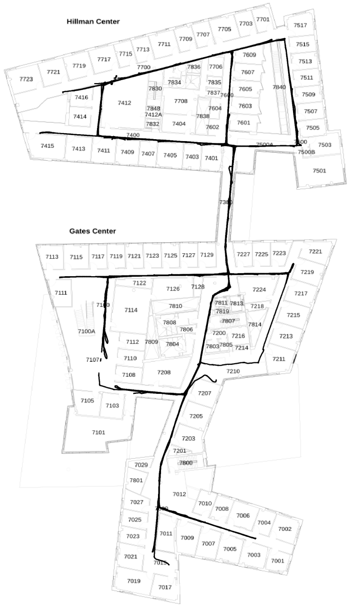


Fig. 5: Union of all trajectories traveled by CoBot-2 on the 7th floor of the Gates-Hillman Center.

and people throughout the entire office environment. Figure 5 shows all the trajectories traveled by CoBot-2 during experiments with 41 tasks (21 Transport, and 20 Go-to-Room), on the 7th floor of the building involving all 88 offices of that floor, and spanning almost all navigable space on the floor.

In this paper, we focused on three different roles that humans present to our mobile service robots, when moving in indoor office environments. Such service robots perform tasks for humans, may need help from humans, and enable humans to be remotely telepresent.

References

[1] R. Simmons, R. Goodwin, K. Z. Haigh, S. Koenig, and J. O'Sullivan, "A layered architecture for office delivery robots," in *Proceedings of the first international conference on Autonomous agents (AGENTS'97)*, 1997, pp. 245–252.
[2] U. Visser and H.-D. Burkhard, "RoboCup: 10 years of achievements and future challenges," *AI Magazine*, vol. 28, no. 2, pp. 115–132, Summer 2007.
[3] J. Biswas and M. Veloso, "WiFi localization and navigation for autonomous indoor mobile robots," in *IEEE International Conference onRobotics and Automation (ICRA)*, May 2010, pp. 4379–4384.
[4] ——, "Depth camera based indoor mobile robot autonomy," in *submission*, 2011.
[5] S. Rosenthal, J. Biswas, and M. Veloso, "An effective personal mobile robot agent through symbiotic human-robot interaction," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1, 2010, pp. 915–922.
[6] N. Armstrong-Crews and M. Veloso, "Oracular pomdps: A very special case," in *ICRA '07*, 2007, pp. 2477–2482.
[7] M. Shiomi, D. Sakamoto, K. Takayuki, C. T. Ishi, H. Ishiguro, and N. Hagita, "A semi-autonomous communication robot: a field trial at a train station," in *HRI '08*, 2008, pp. 303–310.
[8] S. Rosenthal, A. K. Dey, and M. Veloso, "How robots' questions affect the accuracy of the human responses," in *The International Symposium on Robot-Human Interactive Communication*, 2009, pp. 1137–1142.
[9] S. Rosenthal, J. Biswas, and M. Veloso, "An effective personal mobile robot agent through a symbiotic human-robot interaction," in *AAMAS '10*, 2010, pp. 915–922.
[10] J. Fogarty, S. E. Hudson, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. C. Lee, and J. Yang, "Predicting human interruptibility with sensors," *ACM ToCHI*, vol. 12, no. 1, pp. 119–146, 2005.