In AAAI 2011 Spring Symposium: Help Me Help You: Bridging the Gaps in Human-Agent Collaboration, Stanford, CA, USA, March 2011.

A framework in which robots and humans help each other

E. I. Sklar[1,4]**, S. L. Epstein**[2,4]**, S. Parsons**[1,4]**, A. T. Ozgelen**[4]**, J. P. Munoz**[1] and **J. Gonzalez**[3]
[1]Brooklyn College, [2]Hunter College, [3]City College, [4]The Graduate Center
The City University of New York (CUNY)
New York, NY USA
contact author: sklar@sci.brooklyn.cuny.edu

Abstract

Within the context of human/multi-robot teams, the "help me help you" paradigm offers different opportunities. A team of robots can help a human operator accomplish a goal, and a human operator can help a team of robots accomplish the same, or a different, goal. Two scenarios are examined here. First, a team of robots helps a human operator search a remote facility by recognizing objects of interest. Second, the human operator helps the robots improve their position (localization) information by providing quality control feedback.

Introduction

This paper reports on a preliminary investigation of collaboration in human/multi-robot teams. We have constructed a framework that is designed to support one or more robots working with a human operator in a dynamic, real-time environment. Control of the robots is shared between the human operator and a software controller, and the locus of control can switch during run-time. Sample studies are presented here to demonstrate the baseline capability of the framework and to drive the next step of our broader research plan.

Our research is motivated by two related application areas: *urban search and rescue* (Murphy, Casper, and Micire 2001; Jacoff, Messina, and Evans 2000; Yanco et al. 2006) and *humanitarian de-mining* (Santana, Barata, and Correia 2007; Habib 2007). In both instances, teams of robots are deployed to explore terrain that is potentially unsafe for humans and to locate targets of interest. In the first case, robots explore an enclosed space, such as a collapsed building, and search for human victims who may be physically trapped. The goal is to locate these victims and transmit their positions to human operators, so that human first responders can remove the victims to safety. In the second case, robots explore an open space, such as a field in a war zone, to search for anti-personnel mines that may be hidden from view. The goal is to locate these mines and transmit their positions to human operators, so that the mines can be disarmed and the area rendered safe for people to traverse.

Both application areas have three fundamental tasks in common. First, a robot must be able to explore a region (traverse and maneuver in the physical space) and *localize* (determine and track its position there). Second, a robot must be able to *recognize* objects of interest, using on-board sensors and possibly augmented intelligence to interpret sensor input. Third, a human operator must be able to communicate with the robots remotely and *strategize* so that the team can accomplish its overall task effectively. In a collaborative system, the human operator should not be overloaded with tasks, and the robots should not be idle; the robots should help the human operator accomplish her goal, and the human operator should help the robots accomplish their goal(s).

Background and related work

Human-Robot Interaction (*HRI*) supports collaborative activities by humans and robots to achieve shared goals. Typical HRI research concentrates on the development of software and/or hardware to facilitate a wide range of tasks. These include robots maneuvering in physical spaces, both those designed for humans (Kang et al. 2005) or unfit for humans (Murphy 2000); people programming complex robots (Sandini, Metta, and Vernon 2007); robots cooperating with human partners (Burke and Murphy 2004; Finzi and Orlandini 2005; Wegner and Anderson 2006). and with other robots (Dias et al. 2004; Lagoudakis et al. 2004; Mataric, Sukhatme, and Ostergaard 2003; Stone and Veloso 1998); and interfaces for communicating with robots (Kaber, Wright, and Sheik-Nainar 2006; Rooy, Ritter, and St Amant 2002). Deployed HRI applications include cleaning (Roomba 2010), helping the elderly (Tyrer et al. 2006), assisting first responders in search and rescue tasks (Crasar 2010), and de-mining in military settings (Freese et al. 2007).

There are three main categories of control architectures for human-robot systems (Goodrich and Schultz 2007): *fully autonomous*, where robots make decisions and control their actions on their own; *directly controlled*, where robots are driven by human operators; and *mixed-initiative* (Carbonell 1971; Horvitz 1999), where robots share decision making with human users. Mixed-initiative systems reflect recent trends within the HRI community toward *socially intelligent* interfaces (Breazeal and Scassellati 2002; Dautenhahn 2007) in which the aim is for robots and humans to respond to each other naturally.

We focus on mixed initiative architectures and highlight several approaches. *Adjustable autonomy* in a human-robot system permits dynamic transfer of control from human to robot and vice versa (Goodrich et al. 2001; Scerri, Pynadath, and Tambe 2002). *Collaborative control* offers a dialog-based architecture in which decisions are "discussed" and made in real-time (Fong, Thorpe, and Baur 2003). Other mixed-initiative systems have implemented an affect-based architecture (Adams, Rani, and Sarkar 2004) and used statistical techniques to infer missing information in human-robot communication (Hong, Song, and Cho 2007).

Within mixed-initiative human-robot teams, *collaboration* is an open area of research. One of the primary issues for human-robot teams is that "...robots elicit emergent behavior wherein individual robots follow simple coordination rules, without any explicit teamwork models or goals. This breaks down when a team includes people because the robots can't explain their actions and their role as a team player" (Nourbakhsh et al. 2005).

In response to this concern, we have designed a framework which includes an *intelligence engine* that can offer explanations for actions and provide a means to blend decisions from the human operator with those of the system. Our intelligence engine is based on *FORR* (FOr the Right Reasons), a cognitively-plausible architecture that models the development of expertise (Epstein 1994). FORR is predicated on the theory that good decisions in complex domains are best made by a mixture of experts (*Advisors*). Each Advisor is a resource-bounded procedure that represents a single rationale for decision making. FORR provides a common knowledge store (set of *descriptives*) that Advisors reference as necessary and use in different ways. The FORR architecture is domain-independent, but the knowledge and procedures it acquires are domain-specific. To date, FORR has supported applications for game playing (Epstein 2001), simulated pathfinding (Epstein 1998) and constraint solving (Epstein, Freuder, and Wallace 2005).

We hypothesize that FORR can be particularly useful in the human/multi-robot scenarios we address here, where a variety of information and a variety of perspectives on that information may contribute to a correct decision. Our system design incorporates Advisors as *software agents* (Wooldridge 2002) that offer opinions about what to do under certain circumstances. Some are very simple, while others are more complex. Thus, opinions from a broad spectrum of Advisors are interleaved with opinions from the human operator. The intelligence engine blends these opinions in real-time. This engine will also be able to learn whose advice (including that of the human operator) is most effective for given tasks under particular circumstances. The design of this intelligence engine, discussed in the next section, represents the first application of FORR to a physical robot.

Framework

Our test arena, shown in Figure 1, is a physical space that contains color-coded landmarks to guide the robots. The arena is divided into 7 regions: 6 rooms and a hallway. (A map for it appears on the right side of Figure 3.) A human team member (*operator*) is remotely located, positioned physically away from the test arena so that her only view of the space is via camera images sent to her by the robots.

Figure 1: Robot's test arena

We have designed a software framework that employs a multi-layer architecture, illustrated in Figure 2. In the *robot layer*, robots make autonomous decisions about individual low-level tasks . In the *agent layer*, agents make recommendations about tasks that require shared intelligence. In the *human layer*, a human operator provides high-level input. Each box in the figure is a process, described below.

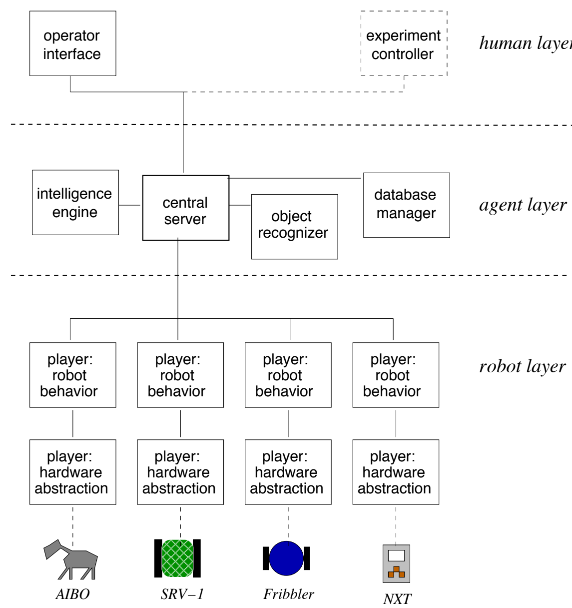

Figure 2: System architecture

The human layer provides for an operator and a controller. The *operator interface* is shown in Figure 3. The right half of the window shows a bird's eye view that indicates the position of each robot in the space. The upper left region contains a "robot's eye view" of the environment. The lower left region contains manual controls that the human can use to drive one robot at a time. The *experiment controller* is used by a human who is not part of the team. This component activates pre-designed experimental conditions. It can, for example, simulate the loss of a robot at run-time. The person who uses the experiment controller has no contact with the human operator.

The agent layer includes both reasoning and data manage-

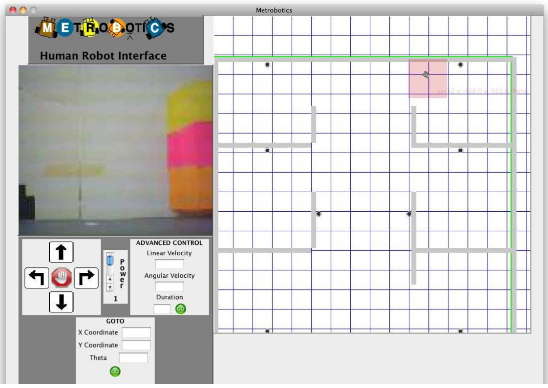

Figure 3: Human interface

ment. The *intelligence engine* is the mechanism that supports learning and collaborative decision making, described later in this section. The primary functions of the *central server* are message passing and bookkeeping. It keeps track of the instantiated processes and of the robots that are connected to the system. The *object recognizer* helps the team identify and locate objects in the environment. *Landmarks* are fixed, pre-defined entities in the robot's physical environment that provide cues for localization. The markers in each of the regions in the robot's environment have different color-codings, so the robot can determine which region it is in. *Objects* are variable items in the environment that are not landmarks. Objects may be labeled with the help of the human operator. Finally, the *database manager* logs system activity. It collects experimental data and maintains a database of known objects and other shared data structures (e.g., a map).

The bottom, robot layer is built on an open source project called *Player/Stage*[1] (Vaughan and Gerkey 2007). Player/Stage provides a modular client/server framework for robotics programming that includes a simulator and allows for unified control of multiple robot platforms. An abstract client class contains high-level robot control functionalities and is extended to support the needs of a particular application. Hardware-specific drivers, implemented as servers, contain low-level sensor and actuator control functions. In our framework, the abstract client is implemented in the *robot behavior* sub-layer, which handles perception, including image acquisition, localization and low-level decision making for each robot. A platform-specific server (or driver) is implemented in the *hardware abstraction* sub-layer, which communicates directly with the robot hardware. We have adapted Player drivers for four different robot platforms that span a range of heterogeneous capabilities. (See Table 1.)

Robot behaviors. Each robot platform can perform a set of simple behaviors, which are implemented in the *hardware abstraction* sub-layer and are invoked by the *robot behavior* sub-layer. Examples are listed in Table 2. The first column

[1] `http://playerstage.sourceforge.net/`

platform	*sensing*	*locomotion*	*communication*
SRV-1/ARM `www.surveyor.com`	camera	tracked	radio
"Fribbler" (= Scribbler: `www.parallax.com` + Fluke: `www.roboteducation.org`)	camera	wheeled	bluetooth
AIBO ERS-7 `www.sonyaibo.net`	camera	legged	wireless
Mindstorms NXT `mindstorms.lego.com`	sonar	wheeled /tracked	bluetooth

Table 1: Robot platform capabilities

names each behavior. Values in the second column indicate the expected duration of each behavior. Values in the third column indicate the behavior category: *motion*, *perception*, *communication* or *localization*. Motion behaviors are invoked with the expectation that they will continue for a predicted (relative) amount of time. (Conditions may arise, however, that require changing the expected motion duration dynamically.) Sense tells the robot to read its sensor values, such as capturing a camera image. Mark tells the robot to remember its current location as a waypoint that can be used for future navigation, and atMark tells the robot to check if it is at a waypoint.

behavior	*duration*	*category*
Forward	$[short \ldots long]$	motion
Backward	$[short \ldots long]$	motion
turnLeft	$[short \ldots long]$	motion
turnRight	$[short \ldots long]$	motion
Stop	—	motion
Sense	—	perception
Mark	—	localization
atMark	—	localization

Table 2: Examples of low-level robot behaviors

Intelligence engine. All complex decision making is handled in the *intelligence engine*, orchestrated by a multiagent society of Advisors, based on the FORR architecture. Each Advisor agent has a single intention. Its task is to recommend behaviors that it predicts will (eventually) fulfill that intention, based on its beliefs about the current state of the world and the capabilities of the robot that it is advising. An Advisor is not a behavior—rather, an Advisor considers all possible behaviors (e.g., those listed in Table 2) and provides *comments* on the appropriateness of each at the given moment. A robot consults its Advisors and selects its behavior accordingly.

Advisors are divided into three tiers. *Tier-1* Advisors recommend low-level, emergency (reactive) behaviors (in the spirit of Brooks (1986)). *Tier-2* Advisors recommend multistep behaviors that may address subgoals. *Tier-3* Advisors are heuristics. We implement tier-1 Advisors in the *robot behavior* sub-layer. Advisors in tiers 2 and 3 are implemented in the *intelligence engine*. The Advisors are organized hierarchically. Tier-1 Advisors are executives whose advice is always followed as soon as it is given. If no tier-1 advice is given, then decisions are deferred to tier 2 (and, if need be, to tier 3) for further deliberation. In practice, most Advisors belong to tier 3.

Tier-1 Advisors are crucial when a condition arises that jeopardizes a robot's safety. For example, if a robot detects that it is about to drive off the edge of a cliff, then the it should stop immediately. Tier-1 Advisors are event-driven and execute in a predetermined order—as soon as one comments, its advice is taken immediately. For example, the tier-1 Advisor Halt comments in response to sensor input that indicates the robot is in a dangerous position and should not move any further. Halt advises that the robot execute the Stop behavior; and the robot does so immediately. Other examples of tier-1 Advisors include: Manual, which comments when the human operator takes control; MadeIt, which comments after a robot receives sensor input indicating that it has arrived at a waypoint; and Enforcer, which drives a multi-step behavior to execute its next step.

Tier-2 Advisors respond to pre-specified situations with a sequence of robot behaviors. Each tier-2 Advisor has a boolean *trigger* that recognizes a situation and a *sequence builder* that constructs a sequence of behaviors to deal with that situation whenever the Advisor's trigger returns true. Tier-2 Advisors do not validate the correctness of their sequences; rather, they respond to a known situation quickly with a sequence of behaviors intended to address it. For example, the tier-2 Advisor ResolveConflict triggers when the robot's sensory information disagrees with its beliefs. ResolveConflict instructs the robot to move slightly and then collect sensor input again. Other examples of tier-2 Advisors include GoHome, which triggers when the robot's energy level is low and instructs the robot to go to a recharging station; and Roundabout, which triggers when the robot recognizes an obstacle and causes the robot to go around it.

Tier-3 Advisors are heuristics that suggest a single robot behavior. For example, the tier-3 Advisor Reorient comments when the robot is approaching a sensed obstacle and advises the robot to turn slightly to the right or left. Other examples of tier-3 Advisors include TravelForward, which advises the robot to move forward when its goal is in that direction; and Skim, which advises the robot to move along the edge of a wall that the robot is near. Because each tier-3 Advisor represents a particular perspective, they are likely to disagree. Traditionally, in a FORR-based system, each Advisor's comment has a *strength* associated with it that indicates the Advisor's degree of preference for or opposition to a particular robot behavior. Disagreements among Advisors are resolved by *voting*, which tallies the total strength for each behavior and then selects the one with the largest total strength. Weights are learned for the Advisors over time. Then, voting multiplies the strength of each comment by the weight of the Advisor that made it. Alternative mechanisms to combine comments are a current research focus.

Input from the human operator is provided through the mixture of Advisors. The Manual tier-1 Advisor gives the human operator direct control of a robot. The human can also provide recommendations through tier-2 or tier-3 Advisor agents. Such recommendations are not given special priority; they are considered along with the (system-provided) Advisors at the corresponding level. This approach to a mixed initiative system allows the human operator's influence in the system to vary dynamically as needed.

Sample studies

Here we describe two sample studies conducted with our framework using SRV-1 robots. First, a team of robots helps a human operator search a remote facility by recognizing objects of interest. Second, the human operator monitors the localization process, with the eventual aim of helping the robots improve the accuracy of their position estimation.

Object recognition. We have run two sets of tests in which the human operator controls robots in the arena and attempts to recognize objects of interest. The first set of tests employed a $1human : 1robot$ ratio, and each of three different human operators ran five test cases. The second set of tests employed a $1human : 2robot$ ratio, and each of two different human operators ran five test cases. For all test cases, another person (not the operator) placed the robot(s) and an object of interest somewhere in the arena (in locations unknown to the operator). The goal was for the operator to drive the robot(s) to find the object. In all cases, the operator was eventually successful. The tests recorded the total amount of time taken to find the object and the number of regions visited. The average amount of time spent per region was calculated, as a basis for comparing the efficiency of different test runs. For the two-robot tests, additional measurements were made: the amount of time that each robot was driven and the number of times each robot was driven.

Table 3 contains the results from these tests. In both cases, only the Manual Advisor was used. Thus, in the two-robot case, only one robot was active at a time. Note that the system seems to be more efficient when there is only one robot. We surmise that this is because the human operator's attention is split between two robots. We anticipate improvement in this when the second robot is fully autonomous, one aim of our current work.

operator	*total time (min)*	*number of rooms covered*	*time per room (min/room)*
first set of tests: $1human : 1robot$			
A	2.26 (0.74)	4.60 (0.89)	0.48 (0.09)
B	2.48 (0.52)	4.80 (0.84)	0.52 (0.06)
C	2.16 (0.97)	4.00 (1.22)	0.52 (0.11)
overall	2.30 (0.72)	4.47 (0.99)	0.51 (0.09)
second set of tests: $1human : 2robots$			
A	2.99 (1.79)	4.20 (1.48)	0.67 (0.38)
B	2.65 (0.99)	4.00 (0.71)	0.66 (0.20)
overall	2.82 (1.38)	4.10 (1.10)	0.67 (0.28)

average (and standard deviation) over 5 runs.

Table 3: Sample study results: Object recognition.

Table 4 compares the amount of time each robot was driven in the 2-robot tests, as well as the number of times that the operator switched from one robot to another. Adding the amount of time each robot was used in a test run gives a more accurate comparison with the values in Table 3. For example, operator A drove both robots on average a total of 2.91 minutes, as compared to 2.99 minutes on average for the overall test; the difference (0.08 minutes) is the amount of time that the operator was driving neither robot. A likely explanation is that the human operator lost time when she switched focus from one robot to another. Further study will examine these differences.

operator	*robot1 (min)*	*switches*	*robot2 (min)*	*switches*	total time
A	0.48 (0.67)	0.40 (0.55)	2.43 (1.65)	1.00 (0.00)	*2.91*
B	1.09 (0.40)	1.60 (0.55)	1.30 (0.68)	1.40 (0.55)	*2.39*

average (and standard deviation) over 5 runs.

Table 4: Robot usage comparison.

Monitoring localization. It is important for the robots to localize accurately, here, to know where they are in the arena. This task is addressed in the robot behavior layer using four standard steps. Figure 4 contains an example. First, raw camera images are captured by the robot's camera (Figure 4a). Second, the images are analyzed for "blobs" of color (4b). Third, the color blobs are matched with landmarks from the dictionary of known objects (4c). Fourth, a particle filter estimates the robot's location (4d), based on the landmarks it sees, as well as the landmarks it has seen recently and the motion behaviors it has executed recently. The end result is a *confidence* measure and an estimated *pose* (x, y, θ) for the robot, where *x* and *y* are planar coordinates and θ is the robot's angle of orientation. Errors in any of the four steps can compromise localization accuracy.

Table 5 shows a representative set of results from our second sample study, during which we monitored localization and collected data over a 13-minute period. The first column contains the actual number of landmarks in the raw image captured in step 1 of the localization process. The second column contains the number of color blob groups detected in step 2. The next four columns contain statistics about step 3 of the process: the number of landmarks found by the process, the number that were recognized correctly, the number that were missed, and the number that were identified incorrectly. The "conf" column contains the confidence value returned by the particle filter in step 4. The final column ("quality") contains an outside observer's assessment of the quality of the localization process. This observer (not the operator) was able to see both the robot in the arena and the particle filter result (e.g., Figure 4d).

Ideally, the quality should correlate with the confidence, but it does not. There are cases when a confidence between 0.36 and 0.39 corresponds to fair, good and poor localization quality. There is some relation between the localization quality and the results of the 3rd step in the localization process. Missing a landmark is less of a problem than incorrectly identifying one. This information can be monitored by the human operator, by watching the images and analysis shown in Figure 4 in real-time during a run. Current work involves giving the human operator the ability to provide quality control feedback to the system, to help the robot localize more accurately.

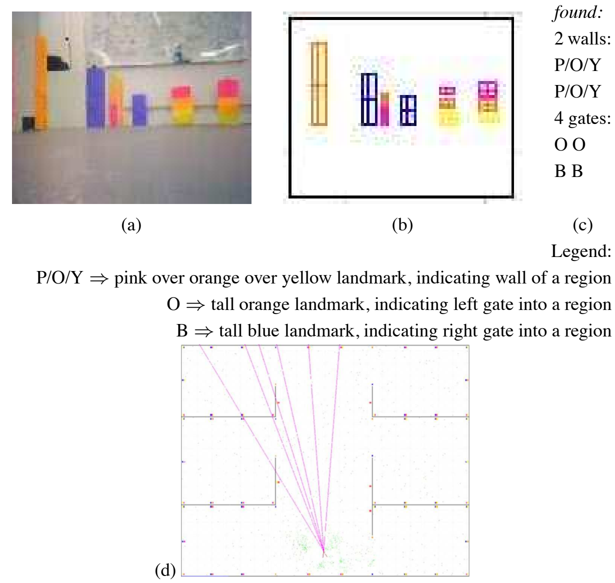

Figure 4: Localization process.

step 1 num	*step 2* blobs	*step 3* found	right	miss	wrong	*step 4* conf	*resulting quality*
4	4	3	3	1	0	0.470	good
6	6	6	5	0	1	0.369	good
3	3	3	3	0	0	0.290	good
3	3	3	3	0	0	0.459	fair
3	3	3	3	0	0	0.363	fair
4	4	2	2	2	0	0.260	fair
4	4	3	3	1	0	0.385	poor
5	5	5	4	0	1	0.309	poor
6	6	5	4	1	1	0.302	poor

(see text for explanation)

Table 5: Sample study results: Monitoring localization.

Summary

We have reported on the design of a framework for investigating collaboration in human/multi-robot teams. Two sample studies, conducted with a preliminary version of our framework, were presented. One focuses on a team of robots that helps a human operator recognize objects in a remote environment. The other focuses on a human operator who helps a robot localize more accurately. Current work includes completing tier-1 Advisors and implementing tier-2 and tier-3 Advisors to support autonomous robot behaviors and collaborative decision making. We are increasing the size of the robot team from 2 members to 10 and conducting experiments with heterogeneous robot teams. In the future, a supervised learning process will allow the human operator to help the robots learn to localize more accurately.

Acknowledgments

This work was supported by the National Science Foundation under #CNS-0851901 and #CNS-0520989, and by CUNY Collaborative Incentive Research Grant #1642.

References

Adams, J. A.; Rani, P.; and Sarkar, N. 2004. Mixed initiative interaction and robotic sys. In *Wkshp on Supervisory Control of Learning and Adaptive Sys, Tech Rept WS-04-10*.

Breazeal, C., and Scassellati, B. 2002. Robots that imitate humans. *TRENDS in Cognitive Sciences* 6(11):481–487.

Brooks, R. A. 1986. A robust layered control system for a mobile robot. *IEEE RAS Jrnl* 2(1):14–23.

Burke, J. L., and Murphy, R. R. 2004. Human-robot interaction in usar technical search: Two heads are better than one. In *Int Wkshp Robot & Human Interactive Comm*, 307–312.

Carbonell, J. R. 1971. Mixed-initiative man-computer instructional dialogues. Technical Report 1971, BBN.

Crasar. 2010. access oct 20. `crasar.org`.

Dautenhahn, K. 2007. A Paradigm Shift in Artificial Intelligence: Why Social Intelligence Matters in the Design and Development of Robots with Human-Like Intelligence. *50 Years of AI Jrnl* LNCS 4850:288–302.

Dias, M. B.; Zlot, R.; Zinck, M.; Gonzalez, J. P.; and Stentz, A. 2004. A versatile implementation of the traderbots approach for multirobot coordination. In *Proc of Intell Aut Sys*.

Epstein, S. L.; Freuder, E. C.; and Wallace, R. J. 2005. Learning to support constraint programmers. *Computational Intelligence* 21(4):337–371.

Epstein, S. L. 1994. For the right reasons: The forr architecture for learning in a skill domain. *Cognitive Science* 18(3):479–511.

Epstein, S. L. 1998. Pragmatic navigation: Reactivity, heuristics, and search. *AIJ* 100(1–2):275–322.

Epstein, S. L. 2001. *Learning to Play Expertly: A Tutorial on Hoyle*. Nova Science. 153–178.

Finzi, A., and Orlandini, A. 2005. Human-Robot Interaction Through Mixed-Initiative Planning for Rescue and Search Rovers. *Advances in AI (AIIA)* LNCS 3673:483–494.

Fong, T.; Thorpe, C.; and Baur, C. 2003. Multi-Robot Remote Driving With Collaborative Control. *IEEE Trans on Industrial Electronics* 50(4):699–704.

Freese, M.; Matsuzawa, T.; Oishi, Y.; Debenest, P.; Takita, K.; Fukushima, E. F.; and Hirose, S. 2007. Robotics-assisted demining with gryphon. *Advanced Robotics* 21(15).

Goodrich, M. A., and Schultz, A. C. 2007. Human-robot interaction: a survey. *Found & Trends in HCI* 1(3):203–275.

Goodrich, M. A.; Olsen, D. R.; Crandall, J. W.; and Palmer, T. J. 2001. Experiments in Adjustable Autonomy. In *IJCAI Wkshp on Autonomy, Delegation, and Control: Interaction with Autonomous Agents*.

Habib, M. K. 2007. Humanitarian Demining: Reality and the Challenge of Technology. *Jrnl of Adv Robotic Sys* 4(2).

Hong, J.-H.; Song, Y.-S.; and Cho, S.-B. 2007. Mixed-initiative human-robot interaction using hierarchical bayesian networks. *IEEE Trans on Sys, Man & Cyber* 37(6).

Horvitz, E. 1999. Principles of mixed-initiative user interfaces. In *Proc of CHI*.

Jacoff, A.; Messina, E.; and Evans, J. 2000. A standard test course for urban search and rescue robots. In *PerMIS Proc*.

Kaber, D. B.; Wright, M. C.; and Sheik-Nainar, M. A. 2006. Multimodal interface design for adaptive automation of a human-robot system. *Int Jrnl of Hum-Comp Studies* 64.

Kang, S.; Lee, W.; Kim, M.; and Shin, K. 2005. ROBHAZ-rescue: rough-terrain negotiable teleoperated mobile robot for rescue mission. In *Safety, Sec & Rescue Robotics Wkshp*.

Lagoudakis, M.; Berhault, M.; Koenig, S.; Keskinocak, P.; and Kelywegt, A. 2004. Simple auctions with performance guarantees for multi-robot task allocation. In *Prof of IROS*.

Mataric, M.; Sukhatme, G.; and Ostergaard, E. 2003. Multi-robot task allocation in uncertain environments. *Autonomous Robots*.

Murphy, R. R.; Casper, J.; and Micire, M. 2001. Potential tasks and research issues for mobile robots in robocup rescue. In *Robot Soccer World Cup IV, LNAI 2019*. Springer.

Murphy, R. R. 2000. Marsupial and shape-shifting robots for urban search and rescue. *IEEE Intell Sys* 15(2):14–19.

Nourbakhsh, I. R.; Sycara, K.; Koes, M.; Yong, M.; Lewis, M.; and Burion, S. 2005. Human-Robot Teaming for Search and Rescue. *Pervasive Computing* January-March.

Roomba. 2010. access oct 20. `www.irobot.com`.

Rooy, D.; Ritter, F.; and St Amant, R. 2002. Using a simulated user to explore human-robot interfaces. In *ACT-R Wkp*.

Sandini, G.; Metta, G.; and Vernon, D. 2007. The iCub Cognitive Humanoid Robot: An Open-System Research Platform for Enactive Cognition. *50 Years of AI Jrnl* LNCS 4850:359–370.

Santana, P. F.; Barata, J.; and Correia, L. 2007. Sustainable Robots for Humanitarian Demining. *Int Jrnl of Advanced Robotic Sys* 4(2):207–218.

Scerri, P.; Pynadath, D. V.; and Tambe, M. 2002. Why the elf acted autonomously: Towards a theory of adjustable autonomy. In *Proc of AAMAS*.

Stone, P., and Veloso, M. 1998. Communication in domains with unreliable, single-channel, low-bandwidth communication. In *Proc of IJCAI*.

Tyrer, H.; Alwan, M.; Demiris, G.; He, Z.; Keller, J.; Skubic, M.; and Rantz, M. 2006. Technology for successful aging. In *Proc Eng in Med and Bio Society*.

Vaughan, R. T., and Gerkey, B. 2007. Really Reusable Robot Code and the Player/Stage Project. In Brugali, D., ed., *Software Engineering for Experimental Robotics*. Springer.

Wegner, R., and Anderson, J. 2006. Agent-based support for balancing teleoperation and autonomy in urban search and rescue. *Int Jrnl of Robotics and Automation* 21(2).

Wooldridge, M. 2002. *An Intro to Multi-agent Sys*. Wiley.

Yanco, H.; Baker, M.; Casey, R.; Keyes, B.; Thoren, P.; Drury, J. L.; Few, D.; Nielsen, C.; and Bruemmer, D. 2006. Analysis of Human-Robot Interaction for Urban Search and Rescue. In *Safety, Sec & Rescue Robotics Wkshp*.