

# An Agent-Based Architecture for an Adaptive Human-Robot Interface

Kazuhiko Kawamura, Phongchai Nilas, Kazuhiko Muguruma,  
Julie A. Adams, and Chen Zhou  
Center for Intelligent Systems  
Vanderbilt University  
Nashville, Tennessee 37235

{kaz.kawamura;phongchai.nilas;hiko.muguruma;julie.a.adams;chen.zhou}@vanderbilt.edu

## Abstract

*This paper describes an innovative agent-based architecture for mixed-initiative interaction between a human and a robot that interacts via a graphical user interface (GUI). Mixed-initiative interaction typically refers to a flexible interaction strategy between a human and a computer to contribute what is best-suited at the most appropriate time [1]. In this paper, we extend this concept to human-robot interaction (HRI). When compared to pure human-computer interaction, HRIs encounter additional difficulty, as the user must assess the situation at the robot's remote location via limited sensory feedback. We propose an agent-based adaptive human-robot interface for mixed-initiative interaction to address this challenge. The proposed adaptive user interface (UI) architecture provides a platform for developing various agents that control robots and user interface components (UICs). Such components permit the human and the robot to communicate mission-relevant information.*

## 1. Introduction

Human-robot interaction is handicapped by the fact that the human must be familiar with the detailed robotic system hardware and software. Furthermore, most interfaces require the user to learn the interface tools—i.e., how to “control” the robot. These issues arise due to the differences in the manner that humans and robots represent the world. Robots use quantitative metrics, while humans use qualitative descriptions, such as “on your right” and “near the white chair.” Furthermore, the interaction has largely been a monolog in which the human commands the robot, rather than a collaborative and dynamically changing bi-directional interaction.

Tasking a team of robots can be very complicated and time-consuming. The larger the robotic team, the larger the number of individual interactions required to control the team and the higher the probability of failure or error. In order to minimize failure, interactions should be multidirectional, occurring not only between the human and the robot, but also between the robots.

This paper describes our efforts to develop an adaptive UI architecture for mixed-initiative interaction between the human operator and a team of robots. Section 2 presents related research while Section 3 provides the background on the adaptive UI architecture design. Section 4 provides an update on the current graphical user interface, while Section 5 describes the current status of this work. Finally, Section 6 provides the conclusions and future work.

## 2. Related Research

There are two areas of relevant related research. The first is supervisory control and the second is adaptive user interfaces. Both are discussed in this section.

### 2.1 Supervisory Control

Great effort has been devoted to the development of supervisory control interfaces [4, 7, 15]. Supervisory control is described as the concept in which control is performed by an intelligent controller under the supervision of a human instead of the human performing direct manual control. Supervisory control of a mobile robot requires a human supervisor to provide high-level commands to the mobile robot [10, 15]. Supervisory control may be necessary when the human and the robot are

geographically separated or when a single human supervises a large number of robots.

Robot supervisory control is usually achieved via human-robot interaction through a user interface (UI) [8, 11, 14, 16]. Murphy [14] presents a system that provides cooperative assistance for the supervision of semi-autonomous robots. The system allows communication between a remote robot, a human supervisor, and the intelligent system. Fong [8] developed an advanced vehicle teleoperation system for a Personal Digital Assistant (PDA) that provides human-robot collaborative control. Terrien [16] describes a remote-driving interface that contains sensor fusion displays and a variety of command tools.

## 2.2 Adaptive User Interfaces

An Adaptive User Interface is a customization technique in which the interface components are partially configured by the system. The system is intended to assist the user with the configuration process based upon the user's specifications (e.g. user's preferences and context situations). Previous work in this area has studied how the interface can be adapted to the user's profiles and preferences. For example, Cheshire [6] employed explanation based learning to build a user preference model using a GUI to control the manner in which battle information is presented to the user. Ardissono [2] developed an adaptive UI for on-line shopping that presents the product categories based upon a user profile.

While past work in robotics research has predominately focused on issues such as supervisory control and teleoperation, relatively few robotics systems are equipped with user interfaces that adapt based on the user preferences, current mission, and actual robot situation. This work differs from previous research in human-robot interfaces in three fundamental ways. First, the system architecture is based on a multi-agent system that should permit porting to any mobile robot platform with a variety of sensors. Second, the distributed, agent-based UI should provide Event Triggered adaptation where one agent generates an event to initiate the adaptation of the other agent. Third, the architecture should also allow bi-directional human-robot interaction.

## 3. Multi-Agent Based Human-Robot Interface Architecture

In order to provide a human-robot interface that can adapt to the current context and mission stages, a multi-agent based adaptive user interface architecture was developed.

One key issue when implementing mixed-initiative interaction is the consideration of the various robot information venues such as raw sensor data and status updates. There are also many manners in which the information can be presented (graphic images, and text dialogues). Information must be gathered from distributed sources, filtered for relevant mission stage content, and presented in a form suitable to the user preferences and needs.

In order to address this issue, an adaptive UI architecture was proposed. The design is based on previous work related to the development of an agent-based robot control architecture [12] that provides a framework for developing agents capable of communicating in a distributed software system. The basis of this design is provided in Figure 1.

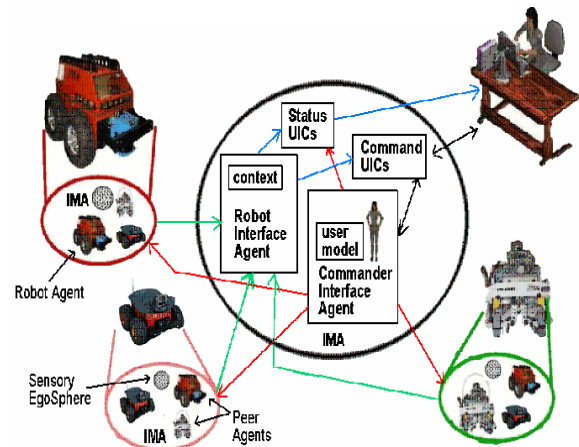


Figure 1. Adaptive UI design concept

The adaptive UI concept consists of the Robot Interface Agent, the Commander Interface Agent, and the User Interface Components. The Robot Interface Agent provides the human with necessary information regarding the robot state and environmental events. The Commander Interface Agent maintains a user model describing the user's preferences and profiles while also forwarding user commands to the appropriate robot agents. The User Interface Components (UICs) present specific information to the user, allow the user to issue commands, etc., as described further in Section 3.1.

### 3.1 The Adaptive UI Architecture

The adaptive UI Architecture is implemented as a distributed set of processing agents, each of which operates independently and performs distinct functions. The system has five primary components: the Commander Interface Agent, the Robot Interface Agent, the Command UIC, the User Interface Manager, the Status UICs, and the database. All the system agents are integrated to provide a multi-agent based HRI system.

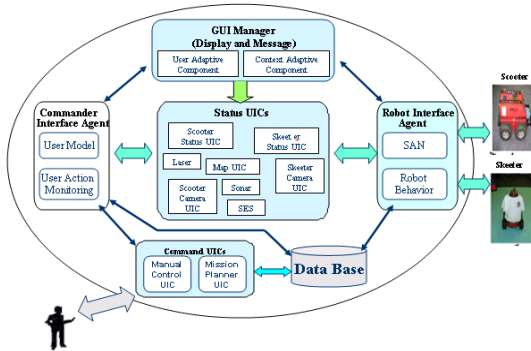


Figure 2. The Adaptive UI Architecture

The Commander Interface Agent provides the user model containing the stereotypical user profiles and adapts the interface to a specific user. The Robot Interface Agent is a robot module that controls robot operation as well as providing collaboration between the low-level agents and the high-level agents. The Command UIC offers the channel through which the human can manually control the operation or provide various high-level mission commands to the robot. This agent decomposes the user command into primitive behaviors and generates the operation plan for the situation. The Status UICs mediate human-robot interaction and enable the human to converse with and assist the robot. The associated displays also provide the ability to monitor the robot as well as receive messages sent by the robot. Finally, the UICs transform and communicate the human input to the robot. The User Interface Manager (UIM) performs necessary interface updates and manages the display of all interface components according to the operation stage. The database stores the user profiles, mission plans, primitive behaviors, and robot specifications.

#### 3.1.1 The Status UICs

The Status UICs enable the human to view mission-relevant data regarding the robot's current status and mission. The user interface components display varying degrees of detail and information based upon the user preferences. Examples of these interface components are the Map UIC, the Camera UIC, the Sonar UIC, and the Laser UIC.

The Map UIC presents a 2D topological map representation of the robot's environment as well as an indication of the robot's position. The user is able to control the robot via the task menu in order to specify tasks such as move-to-point or follow a path. The Map UIC should automatically adjust its map parameters. For example, when a robot detects a target, the portion of the map containing the target is automatically zoomed in to provide more detailed information, as in the map UIC shown in Figure 3.



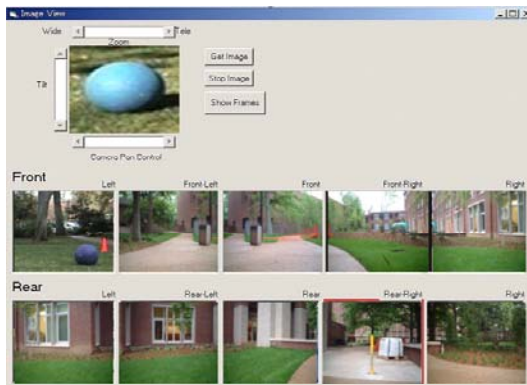
Figure 3. The HRI displaying a zoomed map.

The Camera UIC provides real-time camera images from each robot. The images are adapted according to the user preferences. For example, two robots have been assigned a task to explore a particular target. While the human is reviewing one robot's camera view, the other robot locates the target. The Camera UIC automatically changes the user's camera view to the one provided by the robot that has located the target. This automatic adjustment provides the user with what the system deems the most mission relevant information. Figure 4 shows the HRI at the start of this task including the current Camera UIC information.



**Figure 4. The HRI at the start of the task.**

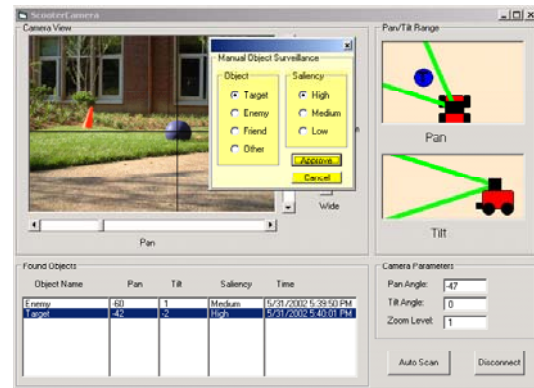
The following examples illustrate how the UICs may take the initiative. Suppose the human sends a command to a robot to locate a blue ball. The robot uses the 360° field of view camera in an attempt to locate the ball. The robot may automatically locate the target using an attention network [5] (see Figure 5), or require human assistance as shown in Figure 6.



**Figure 5. Camera UIC: Automatic Target Detection**

### 3.1.2 The Command UICs

The Command UICs provide a gateway for receiving user commands, planning missions, and generating tasks. This UIC is composed of two primary components: the manual control UIC and the mission planner UIC.



**Figure 6. Camera UIC: Manual Target Detection**

The manual control UIC allows the human to directly manipulate the robot via the interface screen when a task plan is not present (e.g. move forward). The mission planner UIC is composed of the mission planning agent and the Spreading Activation Network (SAN) [13] generator. The mission-planning agent receives a high-level user mission, decomposes the mission into primitive behaviors, and generates a sequence of task plans. After the task plans have been created, the SAN generator obtains the tasks from the mission planner. Then the SAN generator retrieves the robot specification from the database, such as the robot's primitive behavior and the conditions for each action. Finally, the generator links each primary behaviors together to create a spreading activation network based on the mission's goal and the robot's state.

The Command UICs should assist the user with mission planning tasks as well as develop the required behavior network to complete the mission.

### 3.1.3 The Robot Interface Agent

The robot interface agent manages the robot's actions and behaviors during the mission. This agent provides multiple parallel perception processes. These processes provide behavior selection and task operation based upon the current situation. This work employs a Spreading Activation Network (SAN) as a technique to provide the robot's action selection mechanism. The SAN attempts to achieve a number of goals in an unpredictable complex dynamic environment, [3, 9, 13]. The robot employs the SAN to activate the most suitable behavior in response to the current conditions while the robot continues working towards the goal.

### 3.1.4 The Adaptive User Interface Manager

The Adaptive UI Manager is responsible for managing the data displays on the user interface as well as the commands issued by the user. A user may select settings indicating their preferences for the presentation format as well as how much information should be presented. The UI Manager should automatically determine a presentation layout taking into consideration the user's preferences and knowledge of past presentations for this user during similar tasks. When the robot's environment changes and/or internal events occur, the UI Manager should change the data displays accordingly. For example, if the robot locates a target that it is searching for, the UI Manager should send commands to zoom the camera on the target while also enlarging the image of the target on the interface display.

## 4. The Graphical User Interface

The Graphical User Interface (GUI) is an integral part of this system. Robust interaction between a user and a robot is the key factor in successful human-robot cooperation. One means of facilitating the user's access to a wide range of robot information is to provide an interface based upon an agent-based architecture in which the agents provide specific display capabilities. The GUI permits bi-directional communication between the human supervisor at a control station and a robot in the field. This work has implemented a PC-based version of the HRI illustrated in Figure 4. The primary UI agents are:

1. The *User Command Agent* that allows the user to issue high-level commands; for example, follow path, task selection, path planning, and mission planning. This agent provides the primary human-to-robot communication capabilities.
2. The *Landmark Map Agent* presents information regarding the robot's location in the environment. The user may also drive the robot via the map agent by clicking on the map to create a series of waypoints and then click *M2V* (move-to-via point) button. While the robot moves, the map displays the perceived path; any detected objects; as well as the target location.
3. The *Camera View Agent* presents the user with the onboard camera images.

4. The *Navigation Information Agent* presents miscellaneous information such as the sensory data (sonar, laser, and compass); the robot position, heading, and velocity; as well as the target locations.

## 5. Current Status

A test bed scenario was designed to permit the verification of the proposed architecture as well as a demonstration of the proposed interface adaptivity. The system should adapt the interface based on events triggered by the robot actions, such as the detection of a target.

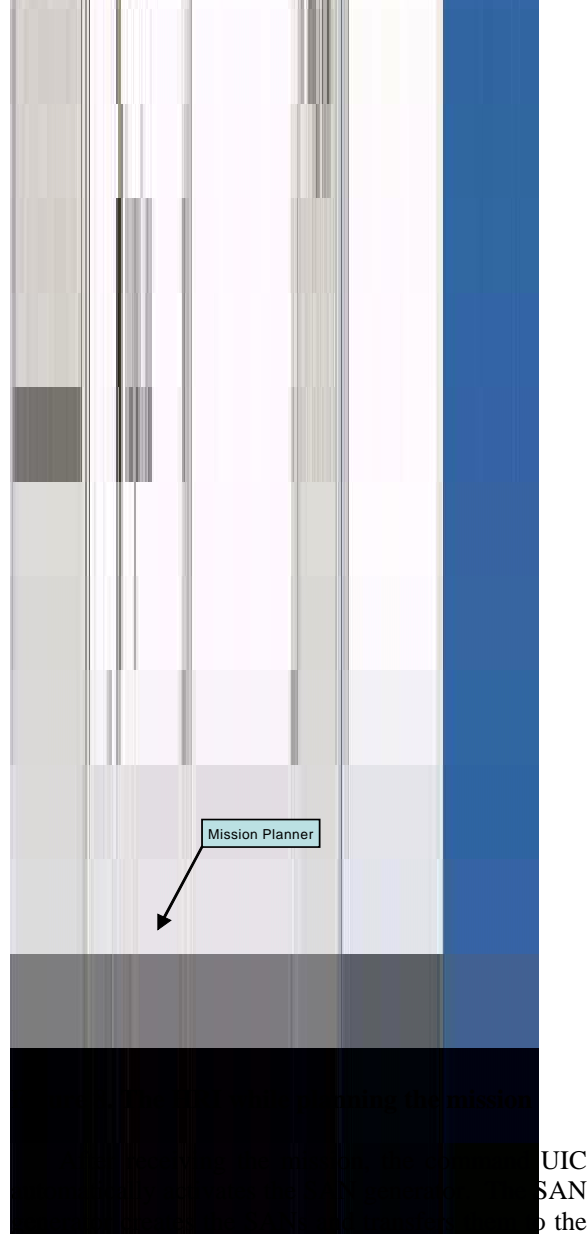
An experiment was conducted employing an ATRVjr robot in an indoor environment. During the experiment, the user issued a mission command to search for a target and then follow that target.

The mission command is provided via the Command UICs. The appropriate Command UIC translates the mission command into the task plans and generates the SANs. The mission planner then transfers the SANs to the robot interface agent. The robot interface agent is responsible for moving the robot to a given location before scanning the environment for the specified target. Once the target is found, the robot interface agent is responsible for ensuring that the robot follows the target.

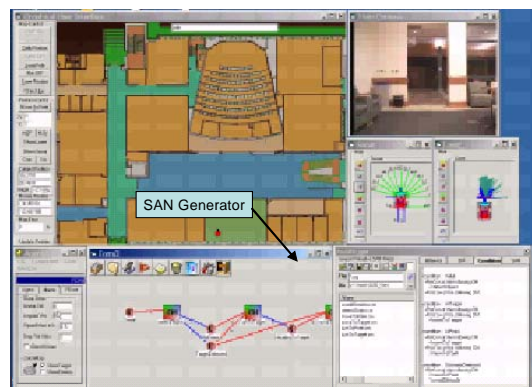
During the task execution, the robot generates events that reflect the current operational state. Such events include the target detection, and obstacle detection. The robot may also create events that trigger the UICs to request information from the user. For example, if the robot detects multiple possible targets, the robot may request the user identify the proper target. The generation of such a request triggers an event that initiates interface adaptation based upon the information required by the user to determine a response.

Figures 7 – 12 show the human-robot interface during the experiment. The initial interface, as presented in Figure 7, is composed of the command UIC, the mission planner, the SAN generator, the Map UIC, the Camera UIC, the Sonar UIC, and the Laser UIC. Figure 8 shows the interface after the user has provided the command to the mission planner and the mission planner agent generates the appropriate mission plan.



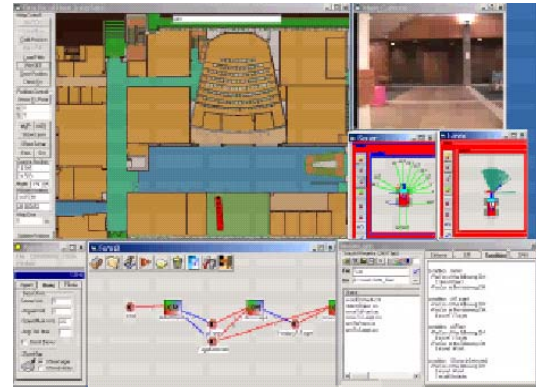


robot interface agent. Figure 9 shows the HRI and the SAN generator.



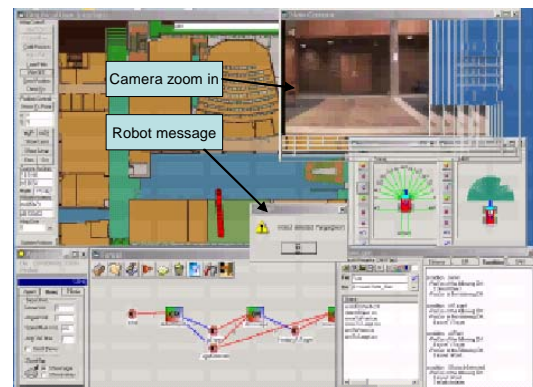
**Figure 9. The HRI showing the generated SAN.**

Figure 10 shows the interface as the robot drives to a specific location. During the operation, the robot detects an obstacle. This event triggers an adaptation of the interface that causes the background of the sonar and laser displays to flash. This adaptation is intended to draw the user's attention to the detected obstacle.



**Figure 10. The HRI after the sensor UIC Adaptations.**

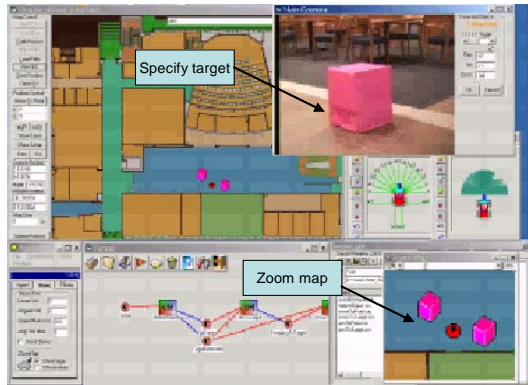
The robot activates the scan behavior when it reaches the specified location. The purpose of the scan behavior is to attempt to identify a possible target. Figure 11 shows the message presented to the user that a target has been detected and demonstrates the zooming of the camera onto the target.



**Figure 11. The HRI while notifying the user.**

In this experiment, the robot detected two similar targets and was unable to autonomously determine which one to follow. The system adapts the interface by zooming in the map to provide a better view of the detailed target location. At the same time, the system

requests that the user specify which target the robot should follow. Figure 12 shows the interface containing the zoomed in map and camera view. After the user indicates which target to follow, the system activates the follow target behavior.



**Figure 12. The HRI displaying the zoomed map and camera image.**

This experiment demonstrated the adaptive UI architecture's ability to automatically modify the HRI based upon the current situation. The experiment also demonstrated event-triggered adaptation between the system's agents. Finally, the experiment demonstrated basic bi-directional human-robot interaction.

## 6. Conclusions

This paper has presented an agent based adaptive user interface architecture, the current architecture implementation, as well as a demonstration of the interface adaptivity. The architectural design as well as the adaptive capabilities were demonstrated using a real robot executing an actual task.

The future work includes completion of the architecture implementation. The Adaptive User Interface Manager and the Commander Interface Agent are not yet implemented. Work is currently under way to implement the Adaptive User Interface Manager. The future work also includes the addition of new Status UICs in order to demonstrate the extensibility of the architecture.

Work has already begun to port the architecture to a PDA based adaptive user interface for the mobile robot domain. As the Adaptive User Interface Manager, and the Commander Interface are

completed, they will also be incorporated into the PDA based adaptive user interface.

## Acknowledgements

This work has been partially funded through a DARPA sponsored grant (Grant Contract DASG60-01-1-0001).

## References

- [1] Allen, J.F. "Mixed-initiative Interaction", in the Proceedings of *IEEE Intelligent Systems*, Marti A. Hearst (ed.), Vol. 14, No. 5, p.p. 14-16, 1999.
- [2] Ardissono, L., A. Coy, G. Petrone, and M. Segnan, "Adaptive User Interfaces for On-line Shopping", *2000 AAAI Spring Symposium*, Technical Report SS-00-01, p.p. 13-18, 2000.
- [3] Bagchi, S., G. Biswas and K. Kawamura, "Task Planning under Uncertainty using a Spreading Activation Network", *IEEE Transactions On Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 30, No. 6, p.p. 639-650, November 2000.
- [4] Bennett, K. B., J.D. Cress, L. J. Hettinger, D. Stauber, and M.W. Hass, "A theoretical Analysis and Preliminary Investigation of Dynamically Adaptive Interfaces", *The International Journal of Aviation Psychology*, p.p. 169-195, Vol. 11, 2001.
- [5] Cave, K.R., "The Feature Gate Model of Visual Selection", *Psychological Research*, Vol. 62, p.p. 182-194, 1999.
- [6] Fijakiewicz, P., et al, "Cheshire: An intelligent adaptive user interface", in *Advanced Displays and Interactive Displays Consortium: Second Annual Fedlab Symposium*, p.p. 15-19, February 1998.
- [7] Fong, T., C. Thorpe, and C. Baur, "Advanced Interfaces for Vehicle Teleoperation: Collaborative control, sensor fusion displays, and remote driving tools", *Autonomous Robots*, Vol. 11, No. 1, p.p. 77-85, 2001.
- [8] Fong, T., "Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation", Technical Report CMU-RI-TR-01-34, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, November 2001.
- [9] Gaines, D. M., Wilkes, M., K. Kusumalukool, S. Thongchai, K. Kawamura and J. White, "SAN-RL: Combining Spreading Activation Networks with Reinforcement Learning to learn configurable behaviors", In the *Proceedings of the International Society of Optical Engineering Conference*, Oct 28-20, 2001.

- [10] Kawamura, K., R.A. Peters II, C. Johnson, P. Nilas, and S. Thongchai, "Supervisory Control of Mobile Robots using Sensory EgoSphere", *Proceedings of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Banff, Alberta, p.p. 531-537, 2001.
- [11] I.S. Lin, F. Wallner, and R. Dillmann. Interactive control and environment modeling for a mobile robot based on multisensor perceptions. *Robotics and Autonomous Systems*, Vol. 18, No. 3, p.p. 301—310, August 1996.
- [12] Pack, R.T., D.M. Wilkes, and K. Kawamura, "A Software Architecture for Integrated Service Robot Development", In the *Proceedings of the 1997 IEEE International Conference On Systems, Man, and Cybernetics*, Orlando, pp. 3774-3779, September 1997.
- [13] Maes. P. How to Do the Right Thing. *Connection Science*, Vo. 1, No. 3, p.p. 291-323, 1989.
- [14] Murphy, R., and E. Rogers, Cooperative Assistance for Remote Robot Supervision, *Presence, special issue on Starkfest*, Vo. 5, No. 2, p.p. 224-240, Spring 1996.
- [15] Sheridan, T. B., *Telerobotics, Automation, and Human Supervisory Control*, MIT Press, Cambridge, MA, 1992.
- [16] Terrien, G., T. Fong, C. Thorpe, and C. Baur, Remote Driving with a multisensor user interface, in the *Proceedings of SAE 30<sup>th</sup> International Conference on Environmental Systems*, Toulouse, France, 2000.