GRAPHICAL USER INTERFACES FOR MOBILE ROBOTS

Sushila A. Shenoy

Lakshmi N. Viswanadha

Arvin Agah

ITTC-FY2003-TR-27640-01

November 2002

Information and Telecommunication Technology Center

The University of Kansas, Lawrence, Kansas 66045

ABSTRACT

Graphical user interfaces (GUIs) for monitoring and controlling mobile robots are presented in this report. These interfaces are used to display the robot sensory information and other related data to the user, while allowing the user to control the robot. The main features of the interfaces are described, a number of examples from related work are presented, and a set of guidelines for the design of interfaces are proposed. The guidelines cover sensor views in the GUIs and aspects of user control of the robot using the GUIs.

KEYWORDS

Graphical user interfaces, mobile robots, robot GUI.

1. INTRODUCTION

A graphical user interface (GUI) can be used as the basic monitor and control interface for robots. Information flows in both the directions of the interface. Usually the information to the remote robot (control signals) comprises of much less data compared to the information that flows from the robot (monitoring information). So the GUI has more components and features dedicated to display of the in flowing data. The control communication and data transfer can be done through wireless Ethernet devices. The GUI is a rich environment for real-time two-way communication between the remote unit and the user.

There are three components that form the basis of the design approach:

- Visibility of system status
- Matching system and real world
- User control and freedom

This approach requires that a lot of information be displayed on the user interface. But this is not practical, since with this amount of data on the screen the user is sure to get confused. So at any given time only a few details about the remote unit are to be supplied to the user. But the details should include the critical information regarding the status for the control of robots.

A user can interact with the robot through one of the three types of graphical user interfaces: observer, commander, and a super user. The observer can only observe the system. The commander can send commands to the robots. The super user has the ability to control and modify not only the robot but also the environment of the system [9].

2. INTERFACE COMPONENTS AND FEATURES

2.1 Navigation

The navigation component can be comprised of compass and GPS. Compass is one of the most essential components for navigation, given the extreme working conditions and the lack of sense of direction. Figure 1 is an example of the compass component in the user interface. The window can be made more interesting and fanciful by including the directions more precisely for effective control of the robot. However, this would require more data to be transferred to the GUI for the Compass.

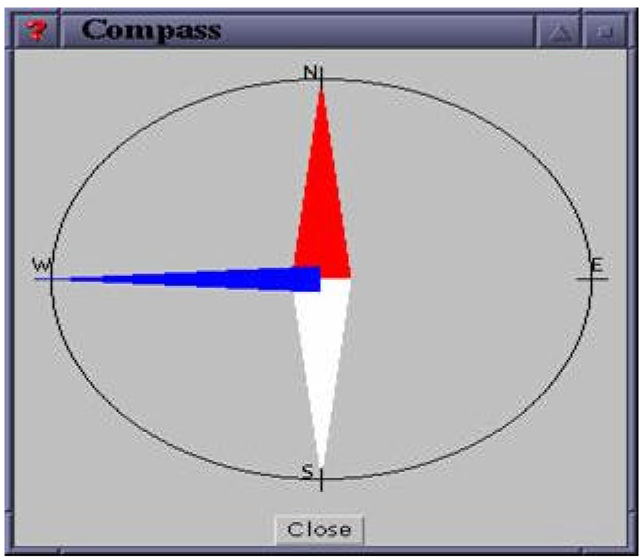

Figure 1. A compass as part of the interface.

GPS provides the actual location of the robots. The data from the GPS can be presented in x-y plane with suitable reference (an aerial view of the robot). The point of command is taken as reference [3]. This data in the user interface provides the User with all the relevant information regarding the positioning of the robot, absolute as well as relative to a reference point. The GPS location is converted into the x-y coordinates on the screen [14]. The navigation also contains the compass tool [3, 17, 27], which gives the directions.

2.2 Trajectory Generation

The GPS provides the current location of the robots in x-y mode, i.e., the exact location is determined. Therefore, going by the previous path taken by the robots, the trajectory can be traced by use of some interpolation methods and other algorithms. The trace varies almost instantaneously with the change in direction, speed and terrain. The future path is given by clicking on the x-y screen or by drawing the path directly on the screen. The path of the leader usually determines the path of the followers in a multi robot system [14].

Also the path taken by the robots is traced on the screen for future reference. The traces are given in different colors to avoid confusion.

2.3 Terrain View

The terrain is calculated as inclined upwards or sloping down or plain, by analyzing the data obtained from the different apparatus mounted on the robot. The robot can try to correct itself (autonomous mode), or the user can adjust the paths by examining the terrain (manual mode).

A gyroscope indicator to give the tilt of the robots is one of the sensors provided with the robot. The gyroscope is a horizontal situation indicator and provides the total tilt of the vehicle at any given instance. It is similar to the one used in aircraft to maintain a horizontal position in all cases. The output of the gyro is included in the GUI for decision making in control of the robots.

2.4 Camera Images

The camera view can provide the image from the camera placed on the robot. Since there can be more than one camera, organizing the camera views is an important task. The GUI should not display all the camera views at a time, but should also not hide important details from the user.

At any given time, not more than a couple of camera views are displayed on to the user interface. The user can choose from a given list of cameras. The video image is then drawn onto a window.

2.5 Robot commands

The commands for the functioning of the robot are given in a drop down menu or in a text field specified for the purpose. The command is typed in the text field and the command is then added to the queue of commands. The commands can be added in this way to the list and deleted directly from the list. For doing the modifications to the command queue the user should be authorized for the job, otherwise the command queue cannot be accessed [3, 7]. The command window has all the commands queued up in the order in which they are given. The list can be preloaded from a file to save the time spent in issuing them. Also, every new list can be saved in temporary memory and resumed after the robot starts again.

2.6 Alarms and Critical Commands

The robot is mounted with a generator for supply of power to all the mounted devices. The generator should always be up and working, even in extreme conditions. So the interface has an energy level description of the robots. The power levels and energy levels are displayed on the screen at all times. If they fall below a predetermined level, the alarms sound giving an indication of the situation [14].

Critical navigation meta-commands such as “STOP”, “RESET” and “CLEAR” are made available as an omnipresent toolbar [3, 26]. They have the highest priority in all cases and supersede the commands in the command queue.

In addition to the above features, a joystick interface to control the robot movement is provided. This feature of the interface provides the user with the complete control of the remote agent, to drive it back to the base in cases of emergency.

2.7 Multiple Views

Multiple views can be used in order to make the interface more effective [17].

2.7.1 Sensor view

The sensor view's purpose is to show the sensor data received from the robot's on-board sensors. These are all passive and readable by anyone at anytime [17]. Buttons are provided to choose the sensor associated with the data being viewed. Same window may be used for the different sensor views, or the views may be compared by having the views drawn on different windows.

2.7.2 Control view

The control view provides various means of controlling the robot, such as rotating it and making it move forwards and backwards [17].

2.7.3 Camera view

The different camera views are provided through tab buttons [17], one each for each of the cameras, mount on the robots. The views are more accurately read by zooming in the view and scrolling through the view screen. Also buttons are provided to change the direction of the cameras [17, 27].

2.7.4 Zoom In and Zoom Out

The zoom of the camera view is provided through the zoom buttons. The zoom of a particular area in the camera view can also be provided. The zoom of the image can be done by a specifying a value in the zoom field [27, 26] or by directly selecting a predetermined zoom percent from a drop down list.

2.7.5 Speed Control

Speed control allows the setting of the desired speed through the interface. The speed can be varied through spin buttons or by directly giving the value in the field specified for the purpose.

2.7.6 Control Mode

The control of the robot has three modes. The mode can be selected from a drop down menu. The modes are manual, semi automatic, and automatic. Another control feature is specified for the type of robot in control in cases of multiple robots, determined by whether it is the leader or the follower [14, 8]. In case of a leader, more control data is to be sent as it controls the whole group of the remote units. As for a follower, it simply follows the leader. Therefore the control data to be sent is comparatively less in such a case.

2.8 Menu Driven

The control of the robot can be done through a menu-driven commanding window. The procedures are provided in the window in advance and the user picks the action from the list, as shown in Figure 2.

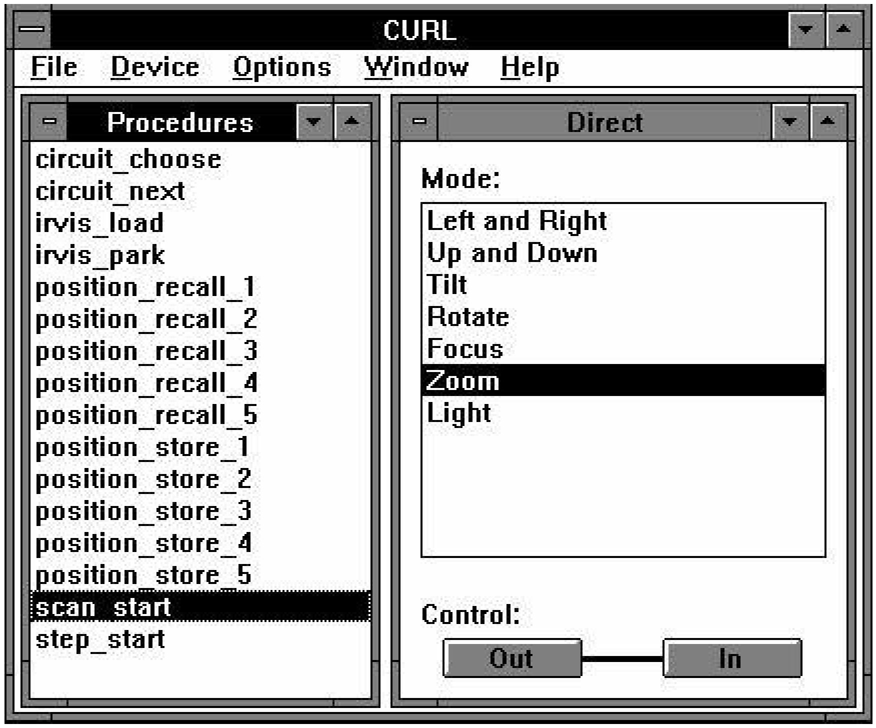

Figure 2. A menu-driven interface.

3. EXAMPLES OF INTERFACES

3.1 A Modular and Dynamic Interface

A modular and dynamic interface incorporates a number of functionalities into modules that can be added to the interface [19]. The robot sensor panel (Figure 3) provides the user with the data from the sensors in an organized fashion. The window gives the details of the sensor, along with a display of its data. The type of sensor is to be selected from a list.

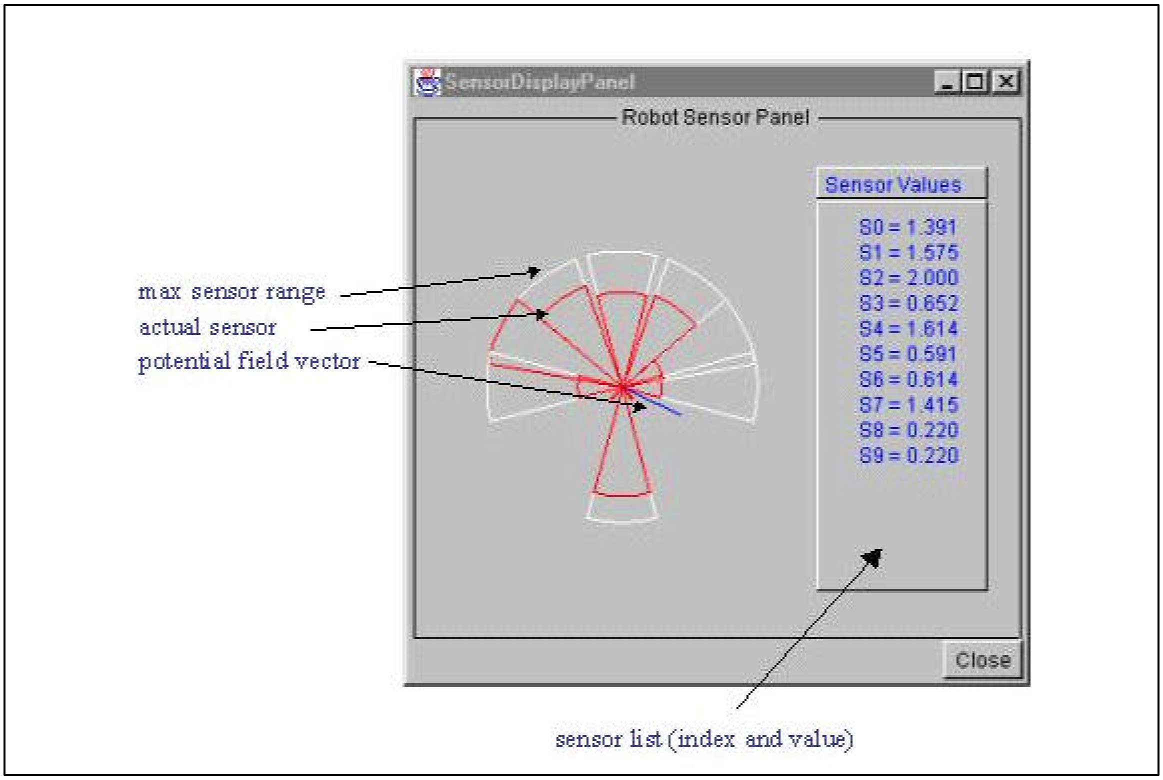

Figure 3. Robot sensor panel [19].

The group panel (Figure 4) gives the user the details of the remote robots (sometimes referred to as agents) when they are working as a group. This window has every detail of the group. This is useful in communicating with the agents when they are more than one in number. The panel shows the name of the group and some user buttons. They can start a state machine, trigger sensors, start a process or event, etc.

The joystick panel (Figure 5) allows the user to directly move the robot or activate its sensors. There are two different joystick panels, one for Millibots (performing polar commands) and one for conventional robots (velocity differential).

The robot state panel (Figure 6) reports the current state of the robot and user control functions for the individual robot, in a multi-robot system. The state gives the name of the robot, the robot type, the current state, current power levels, and some user control features like the sweep of the sensors. The robot can be placed in the protected mode in this window (short sensor reading stops the robot).

The sensor configuration panel deals with everything about the sensors. The user determines the type of sensor whose data it to be analyzed. The sensor is added to the main body of the robot at the desired location. This window was developed for a simulation. In real situations, the sensor is fixed to the robot beforehand. All the user has to do is make it active by selecting it. The user can control the sensor orientation through this window.

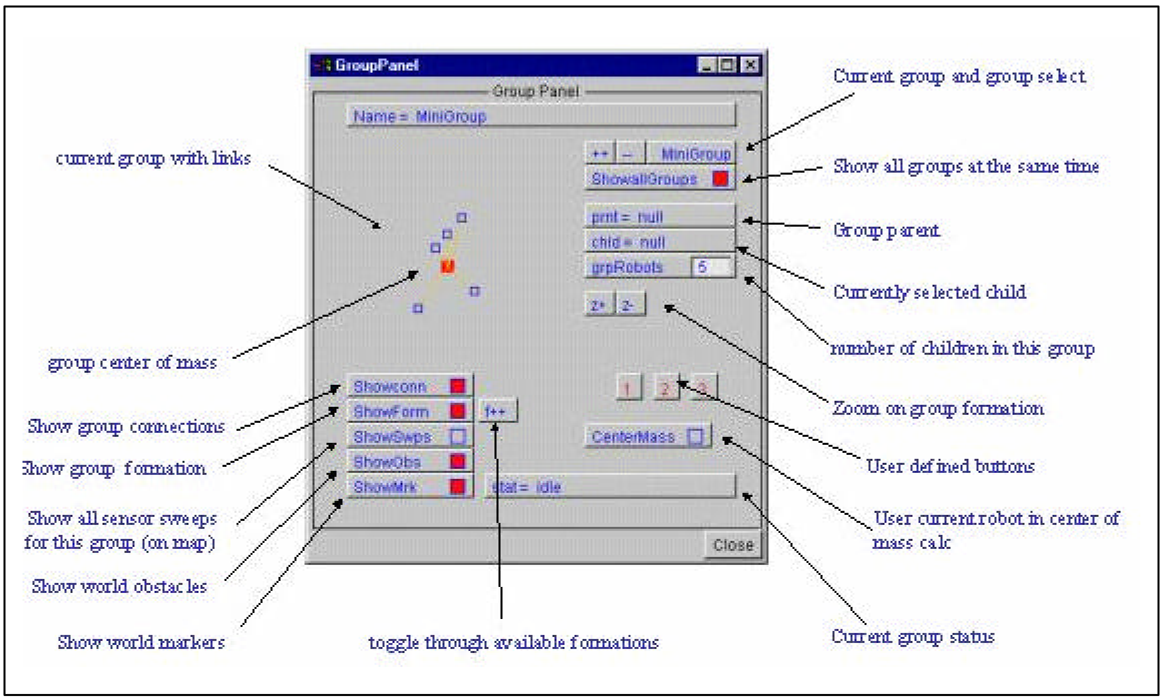

Figure 4. The group panel [19].

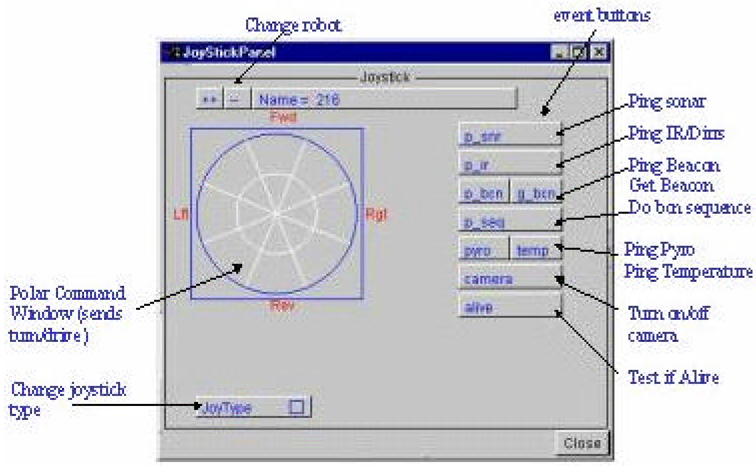

Figure 5. The joystick panel [19].

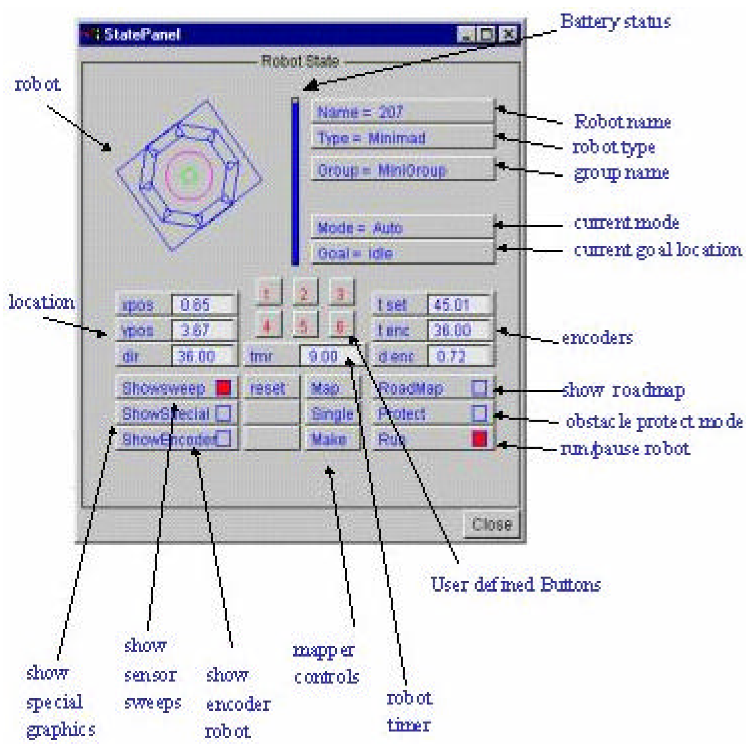

Figure 6. Robot state panel [19].

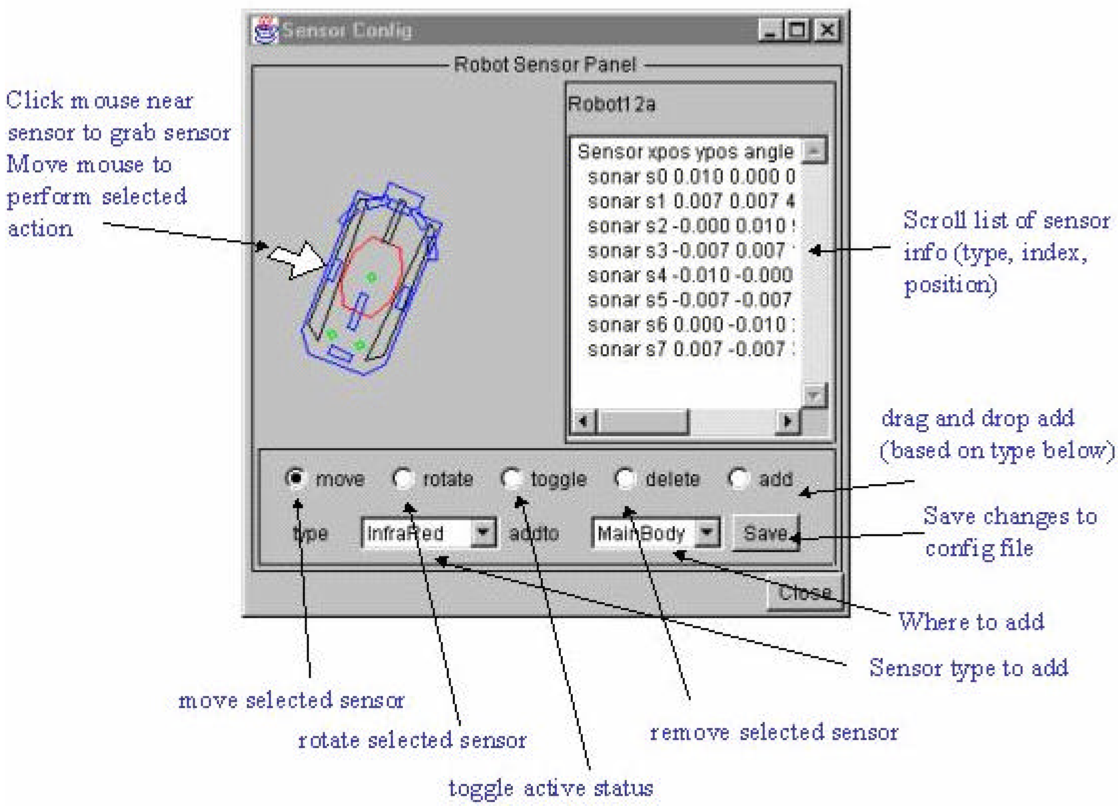

Figure 7. The sensor configuration panel [19].

3.2 Helpmate GUI

HelpMate [21] is designed to make use of user-centered interface as a means of designing a GUI for display and control of mobile robots including their sensors. This is accomplished through a user interface agent called the Commander Agent. Figure 8 illustrates the agent-based Human-Robot Interface (HRI.

The User interface has a camera view with scroll bars, sonar to image representation , a tracking system and fields to specify the maximum rotation, repelling force, maximum speed and other data. The sensor data can be viewed by checking the show sensor check box provided in the main window. The robot is represented with all the sensors in the main window.

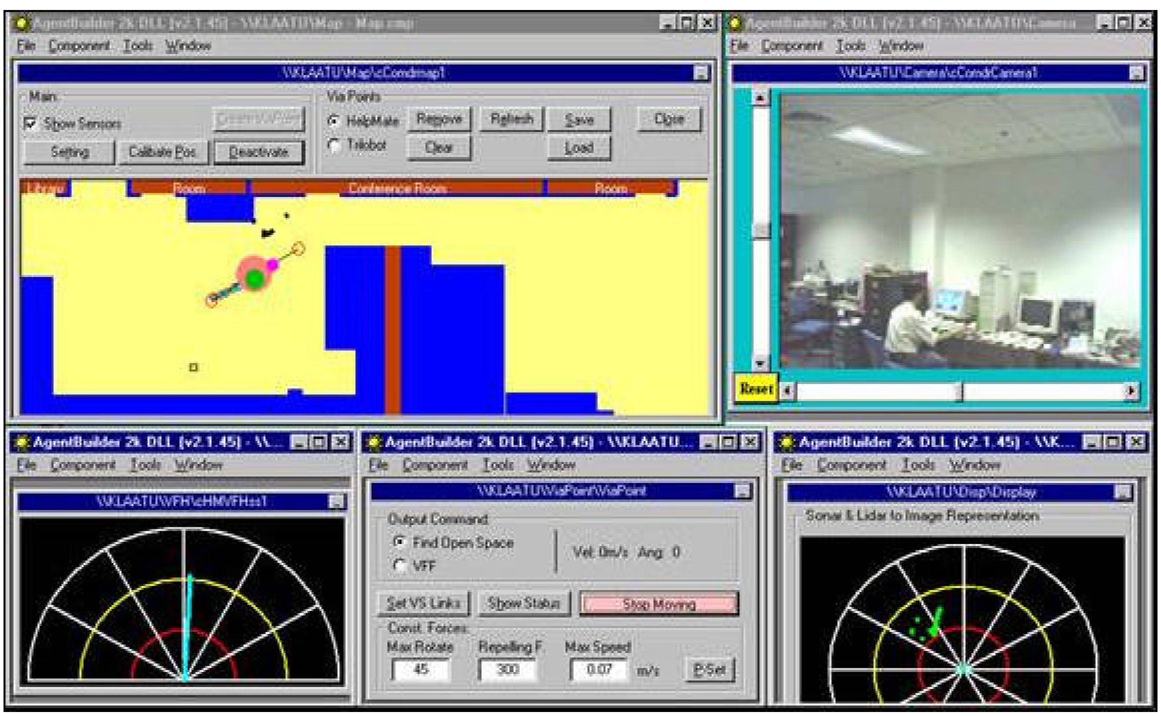

Figure 8. A HelpMate GUI showing a map, image and navigation control [21].

3.3 Spiral Track Autonomous GUI

The Spiral Track Autonomous Robot (STAR) [28] is a maneuverable multi-terrain mobile vehicle that uses computer technology and two Archimedes screws, in contact with the local environment, to intelligently negotiate a hostile environment. The GUI of the STAR system provides different information about the system to the human operator. The information is displayed clearly to make the interface user-friendly. Figure 9 shows the picture of the user interface of the STAR system.

The list of the information provided by the user interface includes: mode of operation, autonomous or tele-operation, state of star, mine detection (alarms if a mine is detected), obstacle detection, total of eight alarms, alarms corresponding to which sensor detected the obstacle, visible trace on a graph showing STAR's path, current location and orientation of STAR, location in X and Y, orientation in degree, motor battery power, alarms when low, electronics battery power, and alarms when low.

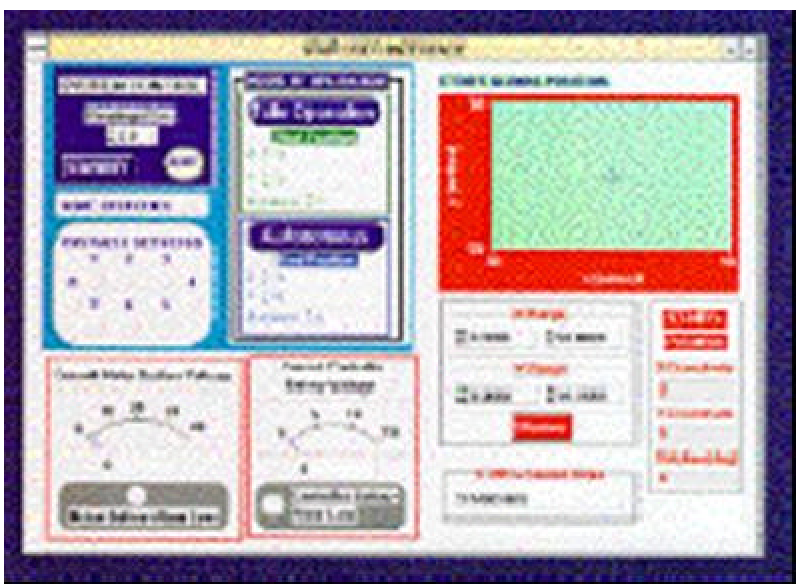

Figure 9. Close-up of the GUI used to operate the STAR vehicle [28].

3.4 GUI for Self organizing Autonomous Incremental Learner

The manual head tab of the SAIL user interface [22] includes the views of the left and right camera mount on the robot (Figure 10). The cameras can be moved using the calibrated scroll provided along the image. Fields to adjust the frame rate and the tilt of the cameras are provided by the interface. The autonomous mode interface would look the same with the variation that all the values are determined automatically, with little user intervention.

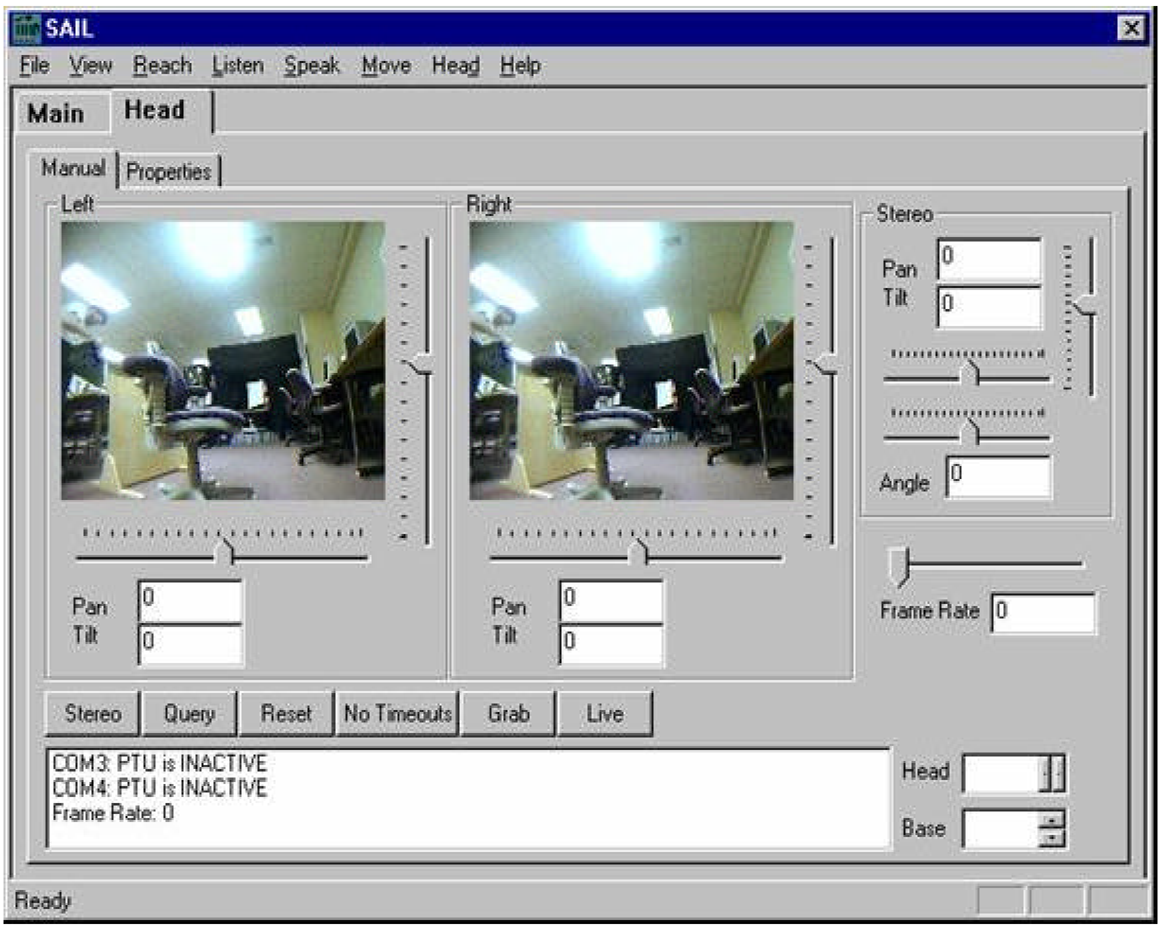

Figure 10. SAIL user interface [22].

3.5 RoboSiM GUI

RoboSiM [20] is a robot simulation and monitoring system for an industrial robot (Figure 11). The user interface includes buttons to change the orientation of the robot, scroll bars to change the direction, vary the speed, adjust the position of the robot model within the window, and other available features.

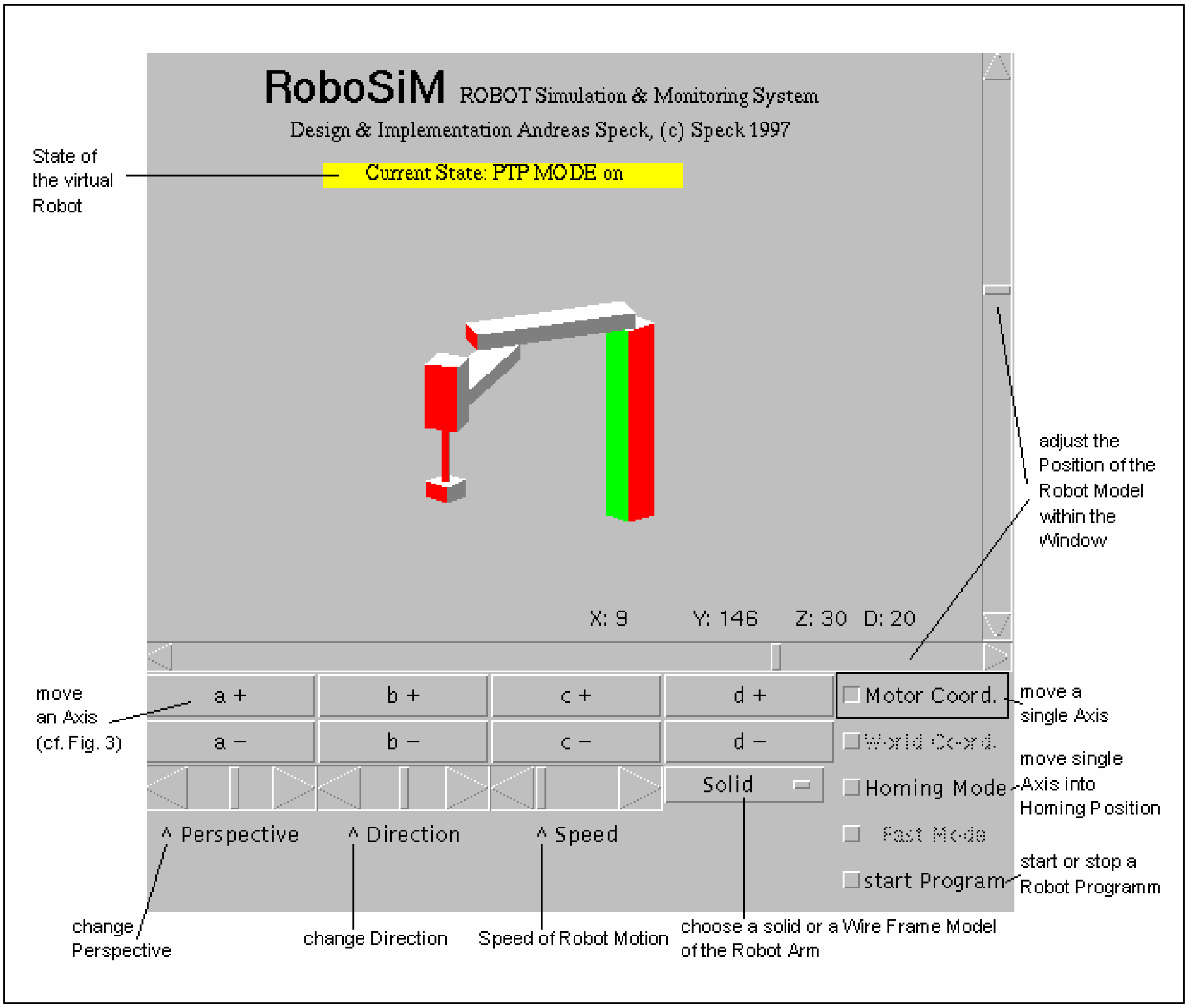

Figure 11. RoboSiM user interface [20].

3.6 Web Interface for Telescience (WITS)

The Web Interface for Telescience (WITS) GUI [29] was developed for the Mars Pathfinder to simulate and create missions for the pathfinder to execute (Figure 12). It was written in Java to be accessible to scientist in different areas around the globe. It supports four user types to connect and collaborate on creating a mission, viewers, navigators, scientists, and mission planners. Scientists specify targets and navigators enter waypoints for the robot to follow.

The GUI includes multiple views to orient the user comprised of photographs taken by the rover of the landscape. The panorama view combines pictures taken at different angles. In some cases a panorama view may not be very useful, as the landscape will may not be visually interesting. Also incorporated is an overhead view of the robot which shows the robot outlines, waypoints, and features of the landscape. The GUI lets the user control the robot by editing waypoints for the robot to follow. The user can enter coordinates or click a point on the overhead view to add a waypoint, changing the path of the robot.

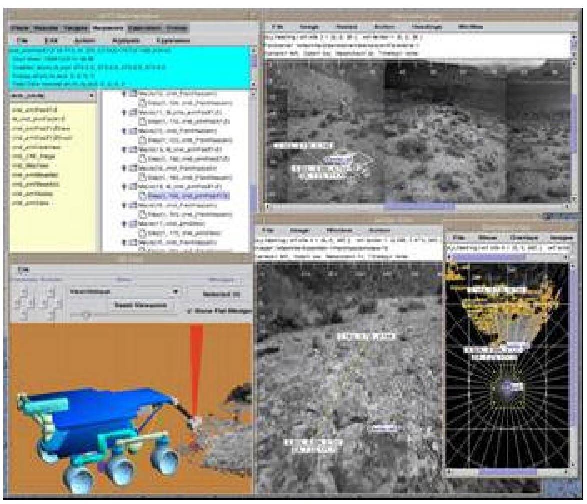

Figure 12. The WITS GUI [29].

3.7 Columbia Robotics Lab Mobile Control Interface/AvenueUI

The Columbia Robotics Lab GUI [3] is made up of several different components running in separate windows (Figure 13). Like WITS, it uses waypoints to control motion but unlike WITS, it is in real time. A map is used to visualize obstacles and place waypoints. The Robot's status is shown in one window and its path is shown on the map.

AvenueUI is a live 3-D interface that has the same functionality as the overhead view but allows the user to visualize the robot's position better. Avenue UI is not as user-friendly as some of the other GUIs because it is meant to adapt to different robots. It is designed with the programmer in mind more than the user. This GUI as well as the WITS GUI is meant primarily for planning and "programming" a robot rather than monitoring or driving it in real time. As this is the case, neither GUI provides much in the way of sensor feedback.

The navigation (NavServer) [3] component of the control system communicates with the low-level sensor/actuator control classes provided by the RWI architecture. The List and Map views are visible along with the Meta-command toolbar and the status panel. The dialog is an expanded view of a single target. The NavServer and Mobility interface hierarchy is to the left. The sonar can be viewed as raw data, a plotted two-dimensional range view, or on the integrated map canvas along with the visualization of the robot's current position and projected path. The human operator can simultanously view the world as seen by the robot and from a detached aerial view.

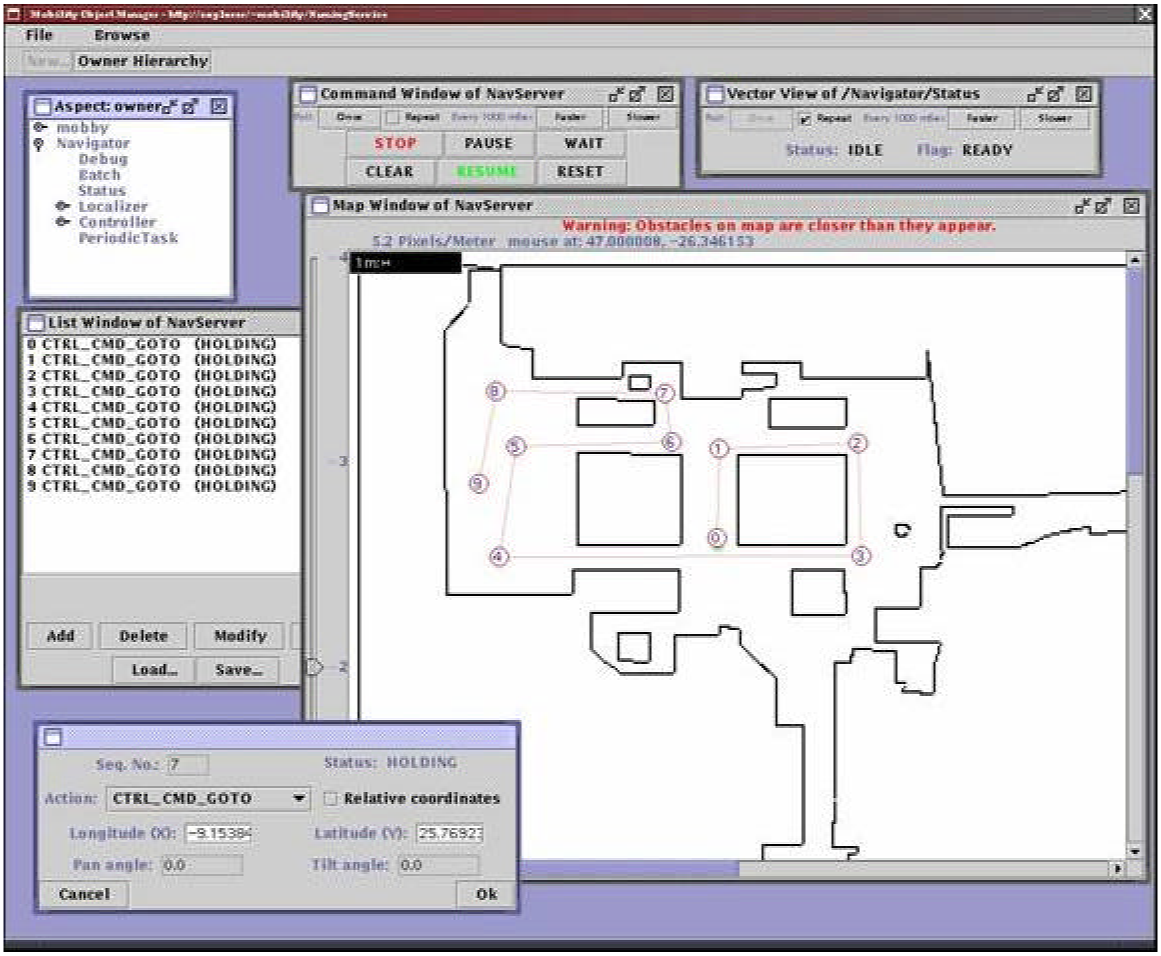

Figure 13. The AvenueUI GUI [3].

3.8 CCI Telerobotics Operator Interface

The Centre for Computational Intelligence (CCI) Telerobotics Operator Interface [30] includes support for a joystick to drive a remote robot (Figure 14). In the center of the GUI are the two most important elements for navigation, the camera view and the overhead view. The camera view includes the distance between the robot and obstacles directly ahead and to either side based on range sensors. The overhead view also displays this data, with the ability to zoom in and out to see objects further away. The system has an option to include a map, if available, in the overhead view as well.

The GUI can switch between the robot navigating itself based on some predetermined method (e.g., collision avoidance) and being driven by the user. This GUI is also meant to be easily customizable based on the robot's abilities and sensors.

Figure 14. The CCI GUI [30].

3.9 Big Signal

Big Signal [31] is the outreach part of the Nomad Antarctic Meteorite Mission (Figure 15). Its Robot Console is a Java applet that allows students to view the scientific data gathered by the robot. The Robot Console includes a panoramic photograph, and an overhead map of where meteorites were found. It also includes the readout of weather information.

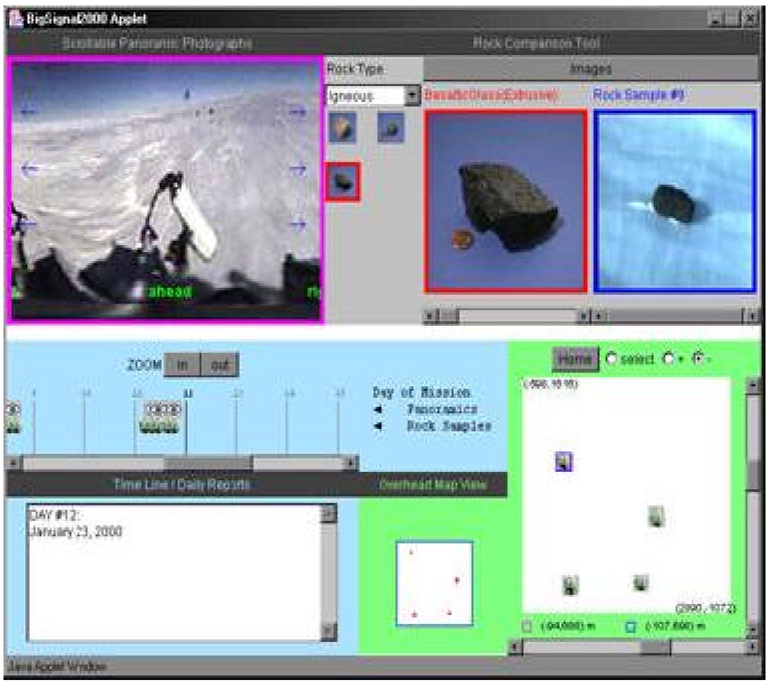

Figure 15. The Big Signal GUI [31].

3.10 Stanford University Aerospace Robotics Laboratory (ARL) GUI

Stanford University Aerospace Robotics Lab based their GUI [32] on popular real-time strategy games in which the player must control multiple units at once (Figure 16). In order to control the robot, the operator first selects a robot to use, then selects the object to be acted on, and then selects an action from a context-dependant menu. The interface uses OpenGL to display the environment in three dimensions, as true-to-life as possible.

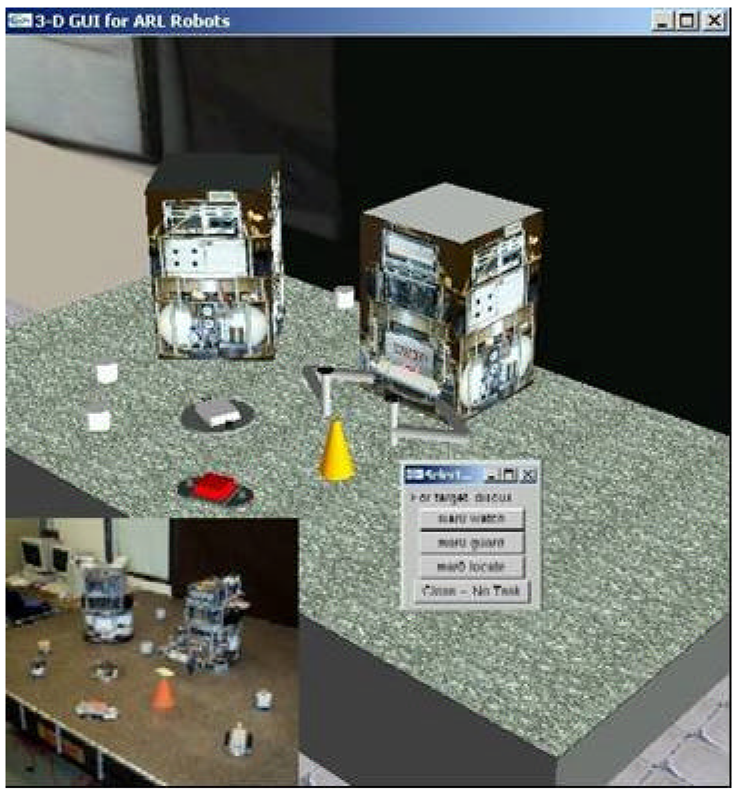

Figure 16. The Stanford ARL GUI [32].

4. COMPARISON OF THE INTERFACES

One common element of the interfaces is the overhead view. The most useful of these are those that incorporate multiple sensors into one display and also display waypoints. Also, those that are scalable allow the most flexibility. Some sort of forward camera view is also usually present, probably due to the fact that a forward view from the robot's location orients the user as if they were actually in the robot, instead of above the robot looking down, which is not as natural, although the overhead view may offer more information overall.

Differences found in different GUIs correspond both to differences in robot function and ability as well as differences based on the intended end user of the GUI. For example, the ARL GUI is intended to be used by a person not necessarily very familiar with the robots or their abilities. This person is probably not interested in lower-level functions of the robot, i.e., they do not care about how fast the robot travels or what direction they are traveling in as long as it is doing what they instructed it to do. The CCI Telerobotics Operator Interface seeks to orient the user as if they were in the location of the robot in order to ensure that the user may easily control the robot in a more specific manner.

The main similarity for all the GUIs is some sort of view, virtual or camera view, of either the robot itself or of its surroundings. For example, the ARL GUI gives a view of the robots from any angle, even those not possible in real life since it is a virtual representation of the robots and their surroundings. The Big Signal GUI does not give a picture of the robot, but does provide panoramic views of the landscape around the robot. This makes sense since the purpose of the Big Signal GUI is to share scientific information and not is centered on the robot itself.

5. DESIGN GUIDELINES FOR INTERFACES

Based on the GUIs which other research groups have developed for their robots, a set of guidelines have been developed that are specific to a mobile robot operating remotely. First, guidelines for sensor feedback, important for supervising the robot as well as driving it, are described. Second, guidelines concerning control of the robot are presented. Finally possible layouts for the information that needs to be presented are discussed.

5.1 Sensor Views

Sensor views should be designed to present the user with as much data as possible in an intuitive format. Graphical representations of sensor data are always preferred to text. In addition, colors should be used to alert users to sensor readings outside of their normal, safe range.

5.1.1 Overhead View

An overhead view can integrate pre-existing data such as maps or satellite images with sensor data from sensors such as the laser range finder to form a real-time map of the robot and terrain. An example is shown in Figure 17.

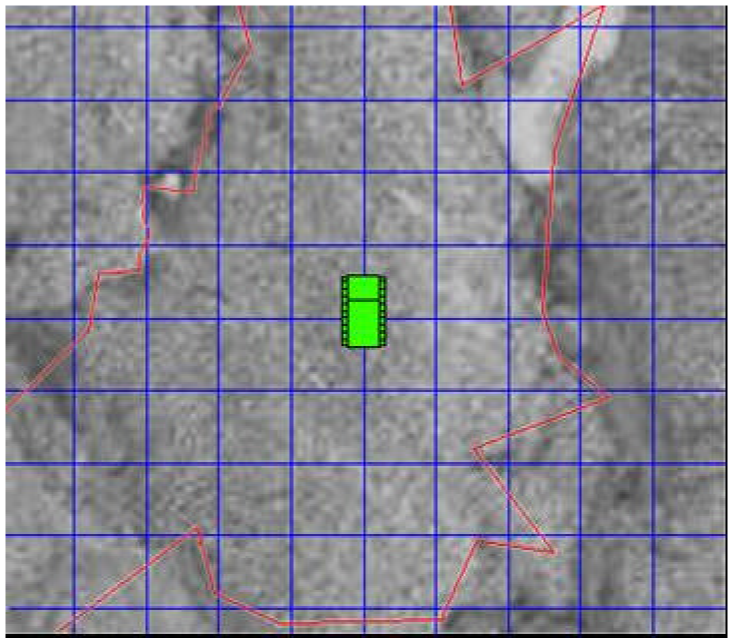

Figure 17. Example of an overhead view.

In order to aid in the navigation, the overhead view should show an outline of the robot as well as relevant features of the landscape. For example, a map and a satellite image, if available. Robot outline scale should match that of map or satellite image. Known waypoints should also be illustrated in the overhead view.

Overhead view should include a visual representation of sensors such as the laser range finder data that is scaled to the map or satellite image. This will also aid in navigation around obstacles that may not be visible on a map or satellite image.

Finally, the user should be able to zoom in overhead view for a close up view of the robot and any obstacles that may be nearby or zoom out to see the general direction of the robot's next waypoint.

5.1.2 Camera View

The camera view is also important for navigation since obstacles may be visible to the camera that are not on the map or satellite image and are too high or low to be seen by sensors such as the laser range finder. The GUI should include pan and tilt controls for the camera.

5.1.3 Other sensors

Other sensors pertaining to the health of the robot such as internal temperature, roll, pitch, and yaw should also be displayed.

5.1.4 Sensor representations

A graphical representation of sensor data is preferred so more information is available at a glance. For instance, instead of showing the angle readout from a tilt sensor, a graphic illustration of the tilt angle can be displayed.

Sensor illustrations should use color to draw the users' attention to sensor readings outside their safe range. For example, an internal temperature illustration might be green while it is at a normal temperature, change to yellow as the temperature approaches an unsafe level, and turn red when the temperature is dangerously high or low.

5.2 Control Guidelines

Although mobile robots can operate autonomously, it is necessary to have a back-up plan in case of emergencies. In such a case it will be useful not to have to go after the robot in possible hazardous conditions, and the robot should be controlled remotely through the interface.

5.2.1 Stop

The user should always have the ability to stop the robot immediately at any time. In addition, the stop button should be clearly marked and visible.

5.2.2 Low-level control

In case the robot cannot navigate its way around or out of an obstacle, the user should be able to control the robot directly using a joystick or keyboard arrow keys. The user should be able to switch between low-level and high-level control.

5.2.3 High-level control

The user should be able to change the robot's path by adding waypoints and editing an existing waypoint path. Waypoints should be illustrated on the overhead view and user should be able to either click on overhead view or manually enter coordinates.

5.2.4 Robot Function

The user should be able to choose between robot functions specific to the missions. For instance, a mobile robot that is carrying radars should be able to switch between different radar modes.

5.3 Layout

Since not all interface components may fit on one screen, or at least not at a reasonable size, there are several ways in which the information can be presented that are feasible. One approach is to present each component in a separate window, so that the user could position them as they needed and switch between windows easily. The other method is to use different tabs for different sets of information. For example, one tab could have the overhead view and some other sensors, and another tab could have the camera view and other sensors.

ACKNOWLEDGEMENTS

This work was sponsored in part by a grant from the National Science Foundation, entitled: "ITR/SI+AP: Mobile sensor web for polar ice sheet measurements."

References

[1] Backes, P.G., Tso, K.S., Norris, J.S., Tharp, G.K., Slostad, J.T., Bonitz, R.G., and Ali, K.S. (2000). Internet-based operations for the mars polar lander mission. In *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, CA, May 2000.

[2] Backes, P.G., Tso, K.S., and Tharp, G.K. (1999). The web interface for telescience, *Presence*, Vol. 8, No. 5, October 1999, 531-539.

[3] Gold, E. (2002). *Mobile robot control interface*. http://www.cs.columbia.edu/robotics/projects/avenue/.

[4] Graves, A.R. (2002). User interface issues in teleoperation. Centre for Computational Intelligence, De Montfort University, Leicester, UK. http://www.cms.dmu.ac.uk/~arg/tmmi/wp3.html.

[5] Graves, A.R. (2002). The Back Seat Driver – A user interface agent for teleoperation. Centre for Computational Intelligence, De Montfort University, Leicester, UK. http://www.cms.dmu.ac.uk/~arg/tmmi/wp5.html.

[6] Jones, H. and Snyder, M. (2002). *Supervisory control of multiple robots based on a real-time strategy game interaction paradigm*. Aerospace Robotics Laboratory, Stanford University, Stanford, CA.

[7] Keates, S., Clarkson, J., and Robinson, P. (2002). *Designing a usable interface for an interactive robot*. http://rehab-www.eng.cam.ac.uk/papers/lsk12/icorr99/.

[8] Keselman, Y. (2002). *Robot graphical user interface manual*, Department of Computer Science, Rutgers University.

[9] Kevin, John, Wesley, Christiaan, Pradeep. (2002). *RAVE, a real and virtual environment for multiple mobile robot systems*, Institute for Complex Engineered Systems, Carnegie Mellon University.

[10] Makatchev, M. (2000). *Human robot interaction - a comprehensive studies report*, City University of Hong Kong.

[11] Lopez, J.F. (2002). *MarS Car project*. http://www-scf.usc.edu/~jflopez/robotics/paper.html.

[12] Miles, E.S. (1997). *A Real-Time Human Perception Interface for Task-Level Control of a Robot in Unfamiliar Environments*. PhD thesis, Stanford University, Stanford, CA.

[13] Miles, E.S. and Cannon, R.H. (1996). An interactive operator interface for task-level direction of a robot in uncertain environments. In *Proceedings of the Second ASCE Conference on Robotics for Challenging Environments*, Albuquerque, NM, 50-56.

[14] Perez, M.L. (2002) *A cost-effective multi-terrain autonomous vehicle for hostile environments*. http://www.llnl.gov/automation-robotics/2_star.html.

[15] Rastogi, A. (2002). *Design of an interface for teleoperation in unstructured environments using augmented reality displays*.

[16] WWW. (2002). *A supervisory interface for Unimation PUMA 762 Robot*. http://www.mb.hs-wismar.de/Mitarbeiter/Pawletta/00Uwe/robot/supervisor.html.

[17] WWW. (2002). *A www interface for robot control-User's guide*. http://www.nada.kth.se/projects/prup98/wwwrobot/docs/user/.

[18] WWW. (2002). *Modular video imagery exploitation workstation*. http://www.ai.sri.com/~oaa/mviews/index.html.

[19] WWW. (2002). *Multi-robot simulator for use in multibot project*. http://www.contrib.andrew.cmu.edu/~rjg/simulator/simulator_page1.html.

[20] WWW. (2002). *Robot simulation and monitoring in Java1.1 and Java 3D*. http://www-pu.informatik.uni-tuebingen.de/users/speck/robosim/.

[21] WWW. (2002). *The HelpMate mobile robot*. http://shogun.vuse.vanderbilt.edu/CIS/IRL/html/helpmate_mobile_robot.html.

[22] WWW. (2002). http://www.cse.msu.edu/~weng/research/altonlei/robot.htm.

[23] WWW. (2002). http://vered.rose.utoronto.ca/people/anu_dir/thesis/thesis.html.

[24] WWW. (2002). http://www.cs.nps.navy.mil/research/auv/papers/pubs1.html.

[25] WWW. (2002). http://www.cs.rutgers.edu/~keselman/Links/links_all.html.

[26] WWW. (2002). http://www.cs.technion.ac.il/Labs/Isl/Project/Projects_done/nomad_spring98/GUI.

[27] WWW. (2002). http://robotics.jpl.nasa/tasks/tmr/ocu.html.

[28] WWW. (2002). http://www.contrib.andrew.cmu.edu/~rig/simulator/.

[29] WWW. (2002). http://wits.jpl.nasa.gov/.

[30] WWW. (2002). http://www.cse.dmu.ac.uk/~arg/tmmi/interface.html.

[31] WWW. (2002). http://www.bigsignal.net/2000/index2.html.

[32] WWW. (2002). http://arl.stanford.edu/projects/interaction/.