From: AAAI Technical Report WS-03-01. Compilation copyright © 2003, AAAI (www.aaai.org). All rights reserved.

Usability and Portability Lessons Learned from the 2003 AAAI Robot Rescue Competition

Douglas A. Few, David J. Bruemmer, & Julie L. Marble

Human, Robot and Remote Systems
Idaho National Engineering
and Environmental Laboratory
Idaho Falls, ID
{fewda, bruedj, marbjl} @inel.gov

Abstract

In order to realize the broad use of robotic systems in hazardous environments, shortcomings in robot interfaces, control system configurability, and overall usability must be addressed. A concerted effort was made to build a foundation of well-engineered communication, perception and autonomous behavior that is robust to changing, unstructured environments and which can be reused across different robot geometries and sensors. During the 2003Robot Rescue competition held in Acapulco as part of American Association for Artificial Intelligence (AAAI) Fifteenth Innovative Applications of Artificial Intelligence Conference, the INEEL demonstrated a high level of success in the areas of human-robot interaction, dynamic sensor configuration, and code portability. This paper will focus on lessons learned with respect to human-robot interface usability and robotic control architecture dynamic configuration and portability.

Keywords: Mixed-Initiative Control, Dynamic Autonomy, Human-Robot Interaction, Robots

Introduction

The lack of human-centered robot interfaces, the rigidity of sensor configuration, and the platform-specific nature of research robot development environments are a few factors preventing robotic solutions from reaching functional utility in real word environments. Often the difficult engineering challenges of implementing adroit reactive behavior, reliable communication, trustworthy autonomy, and usable interfaces are overlooked in favor of far-reaching research aims. The result is that many robotic systems never reach a level of functional utility necessary even to evaluate the efficacy of the basic research, much less result in a system that can be used in a critical, real-world environment.

The Idaho National Engineering and Environmental Laboratory (INEEL) has focused on the need to increase human- centered design and usability through an emphasis on consistency, simplicity, and low bandwidth communication. A human-centered approach requires that robot interface, behaviors and perceptions be designed such that the robot's particular characteristics are transparent to the user. To support this aim, the INEEL has developed a control system, which uses a level of middleware abstraction to support robust perception and autonomous behavior for a wide variety of robotic systems. The abstractions allow for the easy addition of new robot systems as well as providing a method for developing behaviors on one platform that transfer with no source code changes to all other platforms, despite differences in size, bounding shape, or sensor configuration.

These abstractions insure not only that code can be ported from one robot to another, but also provide a means for a standardized, custom communication protocol over a reliable, low-bandwidth communication architecture. The information sent to and from the interface is not dependent on a particular sensor configuration or robot geometry, allowing novice users with no knowledge of robot size, capabilities and sensors to accomplish complex tasks. In order to support different levels of operator trust and skill, the interface is designed with several distinct modes of operator intervention that complement scalable levels of robot autonomy. The system also provides continuous sensor analysis and allows for dynamic sensor reconfiguration – a capability that proved very useful in the competition when sensors actually failed during operation.

A testament to the strength of the INEEL architecture is that the robot behavior demonstrated in the competition was developed on a platform of different make, size and sensor configuration than the one deployed in the Robot Rescue competition. Upon reaching Acapulco, the behaviors developed in Idaho where transferred to the intended robot with no source code compilation required. In fact, no code changes were necessary before or during the competition.

Control Architecture Overview

The robot used in the Robot Rescue Competition was an ATRVJr with an architecture for intelligent behavior, control, and communication developed by the INEEL. The robot is equipped with a variety of range sensors (i.e., Sonar, Laser Scanner, IR), an inertial sensor, a compass, wheel encoders, computer vision, thermal imaging, and tilt sensors. Digitally compressed video and thermal images were streamed to the user interface directly whereas all other perceptual information is analyzed and abstracted by the robot before being filtered out as needed by the interface. In particular, laser, sonar and inertial data is used to build a map of the area as the robot moves through the environment (shown in the upper right quadrant of Figure 2). The robot continuously evaluates the validity of all of its sensors and provides information on the state of each sensor to the user, together with alarms when necessary. In fact, the robot can tell the user when particular aspects of a sensor are not functioning correctly or are providing suspect data. The robot abstracts information about the environment at many levels including terse textual descriptions of local surroundings and the choices (depending on the level of autonomy) that face the human user.

The architecture controlling the robot supports four levels of autonomy:

Teleoperation: The user has full, continuous control of the robot at a low level. The robot takes no initiative except to stop once it recognizes that communications have failed. It does indicate the detection of obstacles in its path to the user, but will not prevent collision.

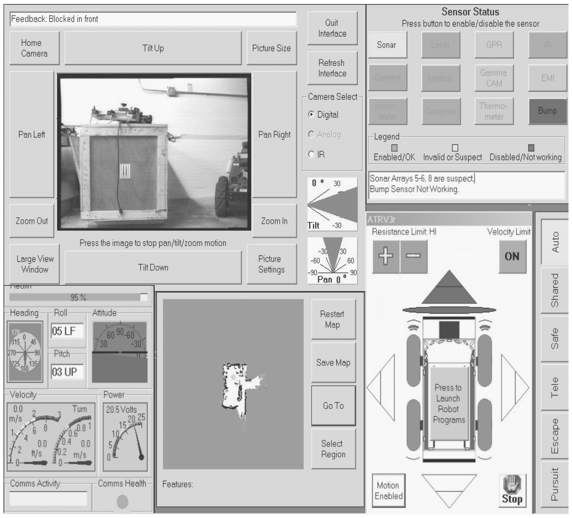
Safe Mode: User directs movements of robot, but the robot takes initiative and has the authority to protect itself based on its proprioception and self-status evaluation; for example, it will stop before it collides with an obstacle, which it detects via multiple sensors. The robot will refuse to undertake a task if it cannot safely accomplish it.

Shared Control: The robot takes the initiative to choose its own path in response to general direction input from the operator. Although the robot handles the low level navigation and obstacle avoidance, the user supplies intermittent input, often at the robot's request, to guide the robot in general directions.

Full Autonomy: The robot performs global path planning to select its own routes, acquiring no operator input except high-level tasking such as "follow that target" (specified using a bounding box within the visual display) or "search this area" (specified by drawing an area within the map interface module). If the operator uses the "pursuit" button, the robot will autonomously follow whatever object the user identifies within the visual image.

For each level of autonomy, perceptual data is fused into a specialized interface window that represents the immediate local surroundings. The INEEL found both in experiments with novice and experienced users that this window was particularly useful for cluttered environments. Indeed, this window was a great asset within the USAR environment, providing indications of resistance to motion when the robot was stuck in netting or pushing against some object. Immediate obstacles that inhibit motion are shown as red ovals to the side or to the front or back of the robot schematic in the lower right quadrant of Figure 2. Resistance to motion is shown with arcs emanating from the wheels, shown for the rear wheels on the schematic.

As the operator touches the visual image on the display, the robot's camera aligns that part of the image to center. The operator can also manipulate the camera by selecting the tilt and pan buttons located around the video display. The robot relays synthesized, high-level information (such as suggestions and requests for help) to the user in a text form using the feedback textbox above the image window.



Feedback: Blocked in front

Sonar Arrays 5-6, 8 are suspect.
Bump Sensor Not Working.

Resistance Limit: HI

Figure 2. Current human-robot interface.

Also note that the robot provides textual reports on environmental features at the bottom of the map window and reports on communications status at the bottom of the robot status window. The robot status window (lower left of Figure 2) provides a variety of information including pitch and roll, power, heading, speed and a fusion of this information into a single measurement of "health." The user may control the robot by using keyboard commands, joystick, mouse or touch screen modes. The status of each sensor type is shown in the top right hand section of the interface. The user can choose to toggle the robot's utilization of each sensor, which proves valuable if sensors become unreliable or if they prove inappropriate for a portion of the environment or task.

Lessons Learned

The challenges of Urban Search and Rescue are sufficient to warrant specialized, directed research.

However, technology deployed by INEEL during the Robot Rescue competition was developed under a Department of Energy (DOE) Laboratory Directed Research and Development (LDRD) program aimed at developing a usable system for remote characterization of hazardous environments. The research team at the INEEL recognizes the AAAI Robot Rescue competition as an excellent venue for evaluating their system performance in an unknown environment.

For the INEEL Team, lessons learned at AAAI come in three forms: proof, discovery and feedback. Lessons learned through proof are either positive or negative in nature and cover lessons learned via verification (such as whether guarded motion proved beneficial in the Urban Search and Rescue (USAR) environment). Lessons learned through discovery cover unpredicted items such as system bugs or insight to needed robot behavior modifications. Perhaps the most beneficial lessons we learned from Robot Rescue competition, came from critical feedback provided by expert system users found at the AAAI Robot Rescue competition.

Lessons Learned via Proof

Recently the INEEL team incorporated several major systemic changes to the robot control architecture. The first was to completely abstract all data and function calls with respect to robot specific geometry, sensor suite, and development environment from the robot control architecture. Doing so required funneling all robot sensor data into standard constructs. The constructs contain robot sensor information in a form generic to ground vehicles enabling the easy addition of future platforms into the INEEL architecture. Additionally all evidence of proprietary robot development environments (i.e., iRobot's 'Mobility,' ActivMedia's ARIA) have been removed from the behavioral content of the control architecture. The combination of these efforts resulted in a system capable of being transferred from one robot to another with out the need of porting or compiling the robot control architecture. The added benefit of this effort is the ability to develop and modify behaviors in complete abstraction allowing for behavior modification and development that applies the each platform in the INEEL control architecture as well as robots owned by other institutions. Recently, the INEEL has ported the "universal" architecture to unmanned systems owned and operated by the Army, Navy, and DOE as well as robots used at other research institutions.

The other major systemic change in our control architecture and user interface that proved successful was the way we handled our sensor validation. Each sensor has its data run through a feasibility check to verify the quality of sensor data. Additionally, the algorithm recognizes if there is a failure of an affiliated sensor process (e.g., compass server, or laser server). The sensor status portion of the user interface reflects the state of each sensor through color indications; green is 'okay', yellow is 'invalid or suspect data,' and red is 'sensor failure.' In addition to reflecting the sensor state, the user interface allows the ability to add and remove sensors from the control architecture. If a sensor is selected which is not available on the robot, the user is notified.

These capabilities proved immensely useful during the Robot Rescue competition. During one run in the NIST test arena the robot local environment section of the user interface (lower right corner or Figure 2) indicated a blockage in front. No blockage could be seen by the incoming video queues. On the sensor status window, the sonar status had suddenly become yellow and a message appeared indicating suspect or invalid data for sonar sensors 1-5. Simply clicking on the sonar button removed the sonar from the data fusion algorithm and the blockage disappeared. The sonar problem was later attributed to a problem with the robot's electronic architecture and had it not been for the dynamic sensor configurability of the INEEL system, the rescue effort would have been aborted mid-mission.

Lessons learned via Discovery

Lessons learned via discovery came in two forms for the INEEL team: system bug discovery and the discovery of a much-needed feature. The system bug was result of a keystroke handling error within the graphical user interface. In a rushed effort to implement keystroke commands available in the text mode to the graphical user interface, proper systematic testing was not completed prior to arriving in Acapulco. Once in Acapulco it became readily apparent there was an error with the graphical user interface. It turned out that we implemented the keystroke function with respect to robot action correctly but neglected to deactivate the default keyboard functions for the active window in the Visual Studio application. The result was that the arrow keys not only provided robot directional control, they also highlighted buttons in the active window. Although distracting, the fact that the arrow keys rotated the button selection would not have posed a severe problem had it not been for the fact that a spacebar keystroke (used to stop the robot) followed by an up arrow (used as a forward command) had the same effect as pressing <Enter> and would depress the active button. Until the complete nature of the problem was understood operators experienced bizarre user interface actions that ranged from inadvertently panning or tilting the camera to shutting down robot processes. Once the bug was understood, users performed mission tasks utilizing a joystick. The keyboard mapping issue was immediately fixed upon return to Idaho.

The other lesson learned through discovery was the need for a sliding velocity control in shared control mode.

Although it is quite impressive to see the robot traverse ground rapidly while mapping an area, in a search and rescue environment the system user needs to have greater control of the speed, especially in the higher levels of autonomy. At the time of the Robot Rescue competition, the shared control mode set the robots velocity by determining the maximum safe velocity based on the robot's local environment. The resultant speed was typically too great for the human operator to adequately process the incoming video stream or conduct a search using the pan, tilt, or zoom features of the camera system with out stopping the vehicle. It was learned that the cognitive burden of driving the robot, stopping, conducting a visual survey and continuing could greatly be reduced by allowing the user to limit the bounds of the velocity in shared or autonomous control mode. Since the AAAI, a velocity throttle feature has been implemented and now applies across autonomy modes. When the user is directly controlling the robot, the joystick's upper and lower bounds are dependent on the throttle. When the robot is driving itself, the throttle likewise constrains the range of speeds available to the robot. Using the throttle, the user can now insure that the robot explores autonomously at a speed appropriate for video surveillance.

Lessons Learned via Feedback

The INEEL team views feedback from experts as essential to the further refinement of the robot control architecture. The following is the feedback provided by the expert systems users.

- Implement a visual depth indication into the video imagery.
- There is a preference for an analog rather than discrete indication of obstacles in the robot local environment.
- Allow users to customize the GUI.
- The 'Robot Parameters' section can be reduced to health, coms and power. All other parameters are either duplicated elsewhere or nonessential.

Currently the INEEL team is forming strategies for addressing visual depth queues in the video display as well as evaluating the possibilities of implementing an analog versus discrete obstacle indicator without varying from the current 9600 baud rate communication scheme between the robot and user interface. We agree that the reduction of Robot Parameters could significantly reduce the visual noise in the display (e.g., Pashler, 1998; Yantis, 1998). Also, we recognize that there may be significant differences in operator preference and skill; however, it is important not to create a system that enables the user to configure the graphical interface in a manner contrary to the principals and practices of human centered design (e.g., Newman & Lamming, 1995; Marble & Proctor, 1998). In fact, although some users claimed that the interface was too "busy," we have ascertained that the suggestion of creating an interface where information must be accessed through separate panes would in fact provide a devastating cognitive load for the operator and inhibit the presentation of critical information. Our experiments with expert and non-expert users have shown us that users do not always understand their own needs, nor is there always an correlation between their performance and their stated preferences. Further subject testing will be performed both at the INEEL and also at NIST to gain more empirical data on how the interface can be improved.

Summary

The AAAI Robot Rescue competition, in Acapulco, Mexico proved to be a great venue for the evaluation of the INEEL robotic control architecture. The nature of the NIST search and rescue course and interaction with the community of exhibitors and competitors provides a unique situation where researchers can prove, discover, and learn methods essential for the growth of their respective research while simultaneously moving the field of robotics towards useful applications and difficult environments.

Reference

Marble, J. L., Bruemmer, D. J., & Few, D. A. (in press). Lessons learned from usability tests with a collaborative cognitive workspace for human-robot teams. In *Proceedings of the IEEE Conference on Systems, Man, & Cybernetics*, Washington, DC, October 4-8, 2003.

Marble, J. & Proctor, R. W. (2000). Emergent features, instruction and graphic enhancement in display design. In *Proceedings of the XIV triennial Congress of the International Ergonomics Association and 44th Annual Meeting of the Human Factors and Ergonomics Society*, San Diego, CA.

Newman, W., & Lamming, M. (1995). *Interactive System Design.* Addison-Wesley: Reading, MA.

Pashler, H. (1998a). *The Psychology of Attention.* MIT Press: Cambridge, MA.

Yantis, S. (1998). Control of visual attention. In H. Pashler (Ed.), *Attention.* Psychology Press: East Sussex, UK, pp. 223-256.