Building a Multimodal Human–Robot Interface

Dennis Perzanowski, Alan C. Schultz, William Adams, Elaine Marsh, and Magda Bugajska,
Naval Research Laboratory

No one claims that people must interact with machines in the same way that they interact with other humans. Certainly, people do not carry on conversations with their toasters in the morning, unless they have a serious problem.

However, the situation becomes a bit more complex when we begin to build and interact with machines or robots that either look like humans or have human functionalities and capabilities. Then, people well might interact with their humanlike machines in ways that mimic human–human communication.

For example, if a robot has a face, a human might interact with it similarly to how humans interact with other creatures with faces. Specifically, a human might talk to it, gesture to it, smile at it, and so on. If a human interacts with a computer or a machine that understands spoken commands, the human might converse with the machine, expecting it to have competence in spoken language.

In our research on a multimodal interface to mobile robots, we have assumed a model of communication and interaction that, in a sense, mimics how people communicate. Our interface therefore incorporates both natural language understanding and gesture recognition as communication modes. We limited the interface to these two modes to simplify integrating them in the interface and to make our research more tractable.

We believe that with an integrated system, the user is less concerned with how to communicate (which interactive mode to employ for a task) and is therefore free to concentrate on the tasks and goals at hand. Because we integrate all our system's components, users can choose any combination of our interface's modalities. The onus is on our interface to integrate the input, process it, and produce the desired results.

Requirements

As developers of speech recognition and natural-language-understanding systems no doubt know, humans expect a fairly sophisticated level of recognition, understanding, and interaction. Speech systems that limit the human to simple utterances, prescribed formulaic utter-

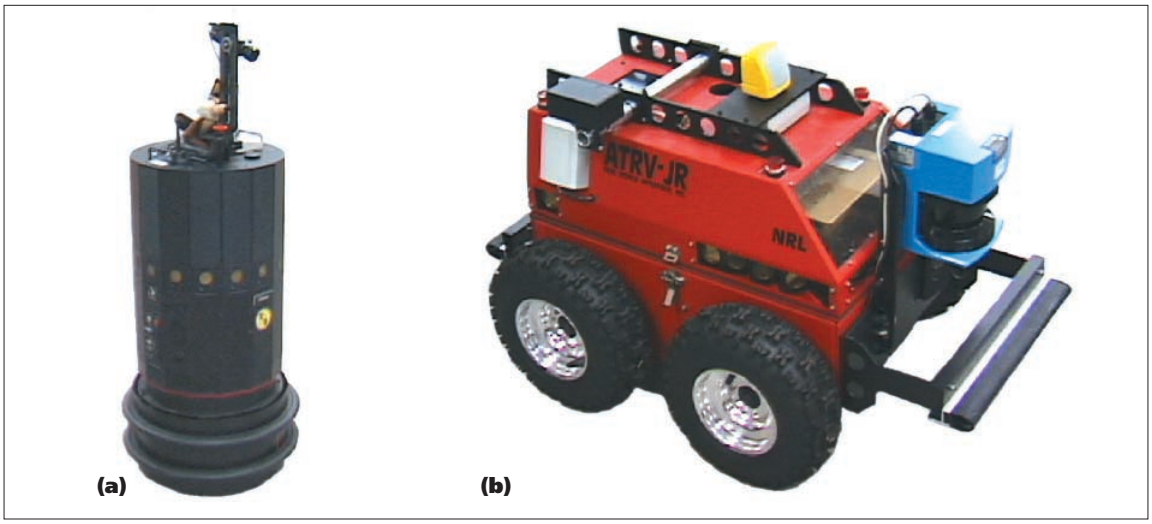


Figure 1. We implemented our interface on a team of (a) Nomad 200 and (b) Real World Interface ATRV-Jr. robots.

ances, or both do not sit well with most users. Humans want human–computer interfaces that can handle functionalities at least as complex as their needs require, given their particular application domain.

The interface should be able to handle potential problems of human speech such as sentence fragments, false starts, and interruptions. (We ignore here the obvious problems of speech recognition due to extraneous noise, mumbling, and so on.) It should also know what the referents are to the pronouns being used and be able to carry on more complex discourse functions, such as knowing a dialog's topic and focus.[1] In short, it should facilitate normal, natural communication.

Furthermore, multimodal interfaces should be fairly transparent. People normally don't have to make more than a few simple adjustments to communicate with each other—for example, adjusting to another person's dialect or speech patterns. Using the interface should be as simple.

Figure 2. A researcher interacts with Coyote, one of the mobile robots, using natural language and gestures.

Our multimodal interface

We have implemented our multimodal interface on a team of Nomad 200 and RWI ATRV-Jr. robots (see Figure 1). The robots understand speech, hand gestures,[2] and input from a handheld Palm Pilot or other Personal Digital Assistant.[3]

Types of input

The human user communicates verbally with all the robots through a wireless headset. IBM's speech-to-text system ViaVoice initially processes the speech, and our natural-language-understanding system Nautilus robustly parses the text string.[4] Nautilus then translates the parsed string into a semantic representation, which the interface eventually maps to a command.

People gesture while they speak. Some gestures are meaning-bearing, some are superfluous, and others are redundant. Some gestures indicate the speaker's emotional or intentional state. Our multimodal interface deals with only meaning-bearing hand and arm gestures that disambiguate locative elements referred to during human–robot interaction. For example, when someone says "Go over there," the utterance is meaningless unless that person gestures to the physical location. Our interface interprets *natural gestures*, those made with the arm or hands (see Figure 2), and *mechanical gestures*, those made by pointing and clicking on a PDA touch screen.

For our interface, gestures designate either distances, indicated by holding the hands apart, or directions, indicated by tracing a line in the air. (This restriction is due to the limited vision system we currently employ. We are expanding our vision capabilities by changing to binocular vision.) To detect these natural gestures, our robots use a laser rangefinder that emits a horizontal plane of light 30 inches above the floor. Mounted on the rangefinder's side is a camera with a filter tuned to the laser wavelength. Because the laser and camera mount are at a right angle and the camera is tilted a fixed amount, the robot can easily triangulate the distance to a laser-illuminated point. With this sensor, the robot can track the user's hands and interpret their motion as vectors or measured distances. (David Kortenkamp, Eric Huber, and R. Peter Bonasso have developed an alternative means of mapping arm and hand gestures.[5]) The interface incorporates the gestural information into the semantic representation.

The PDA dynamically presents an adaptive map of a particular robot's environment, which comes directly from the robot through a mapping and localization module (see Figure 3).[6] Through the PDA, users can directly give a limited set of commands to the robots. They can tap on menu buttons

Figure 3. Users can enter commands through a Palm Pilot, which presents a map of the robot's environment.

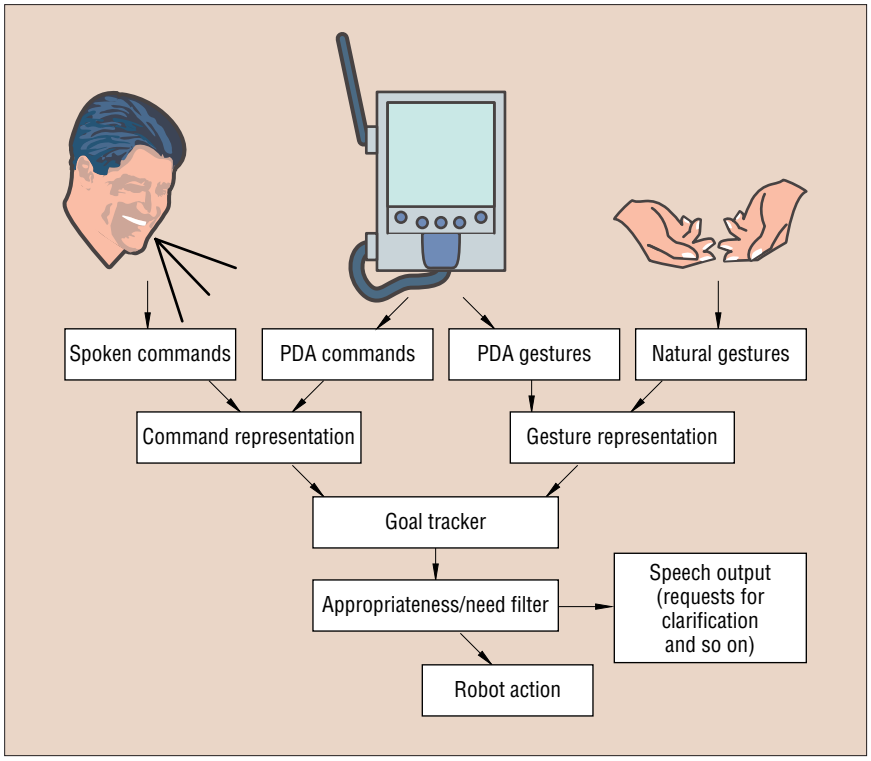


Figure 4. A schematic of the multimodal interface.

on the device's touch screen, or gesture (by tapping or dragging across a map of the environment on the PDA screen) to indicate places or areas for the robots.

To command one or a team of robots to navigate to a location, users can combine the speech, gesture, and PDA input in various ways. For example, a user could

- point at the actual location and say "Go over there,"
- click on the location on the PDA map and utter the same sentence,
- click on a command on the PDA menu and select the location on the map, or
- click on a command on the PDA menu and point at the actual location.

Users can also control the robots with a joystick, and they can command a robot to move to a particular set of *x*, *y* coordinates in 2D space by touching points on the PDA screen with a stylus. However, users seem to prefer using natural language and one of the gesturing modes. (We have not yet conducted any formal experiments measuring this; however, we are designing an experiment to test our assumptions and hypotheses.)

Processing input

Figure 4 represents the various input modes and the subsequent merging and integration of information that occurs in the *command representation* and *gesture representation* modules. The interface then integrates the command and gesture information through the *goal tracker*, which also stores additional information (we describe goal tracking in more detail later). Next, the appropriateness/need filter determines what speech output or action, if any, is necessary.

Owing to errors in speech recognition, some sentences "heard" do not get parsed. External noise can cause recognition errors. When this happens, the system tries to parse what it thought it heard. If the utterance might have been grammatical, the system asks the user for confirmation.

For example, if the robot does not recognize the utterance "Go over there" with the degree of confidence set by the speech recognition system, the robot asks the user, "Did you say, 'Go over there'?" This provides the user with some natural feedback. Humans do exactly the same thing. However, if the robot parses an utterance and determines that the utterance is nongrammatical, such as the utterance "Go to the," the system simply utters "What?" Again, we feel this is a natural interaction in this context. People don't ask each other what misunderstood grammatical sentences mean. And they usually don't repeat them as if they were grammatical, such as "Did you say, 'Go to the'?" They simply confirm that an utterance was perceived and ask for additional information through the one-word utterance "What?"

Whenever the system obtains a grammatical utterance, the appropriateness/need filter checks the resulting representation against any perceived gesture. The filter checks the appropriateness of various gestures with the perceived utterance and filters out redundant gestures. If a gesture is not needed to disambiguate the utterance further, the filter simply ignores the gesture. For example, most gestures made while users utter "Stop" are superfluous. Users uttering this want one thing only: immediate cessation of activity. Granted, arms waving frantically during the utterance might indicate the human's emotional state, but we do not consider this a disambiguating gesture, so we ignore it here.

However, if a gesture is needed to disambiguate the utterance, as for example when someone says "Go over there," the appropriateness/need filter checks to ensure that a gesture accompanies the utterance. If the filter perceives an appropriate gesture, the robot performs an action. However, if the filter does not perceive a gesture, the robot asks the user for one: "I'm sorry, you told me to go somewhere, but you didn't tell me where. What do you want me to do?" Likewise, if the filter perceives an inappropriate gesture with the utterance in question, the system informs the user: "I'm sorry, but that gesture makes no sense with that command."

The robots also use speech output to inform the user of what the various agents in the interchange are experiencing. For example, if the user tells a robot to go to a door but accidentally gestures to the wrong place, the robot responds, "There is no door over there." Based on the robots' acquired knowledge of the environment, they feed information back to the user. Participants keep each other informed of their own awareness, their current states, and how these things affect the other participants in the dialog. Rather than having robots that simply act on commands, we are trying to build robots that interact and

cooperate with humans, much as humans do in human–human communication.

Working together

The processes on the various robots communicate with each other through TCP/IP. Each robot has been programmed to respond only to commands addressed directly to it or to communal commands. For example, when a user addresses one of the robots, Coyote, with the utterance, "Coyote, go to the door," only Coyote responds. Other robots, such as Roadrunner, process the utterance but will not act, because the command was not directed to them. However, if the user utters "Robots, go to the door," all robots will respond.

We decided to have all the robots process all the commands because this approach seemed natural. When several people converse, hopefully all the individuals are processing all the utterances, even though they might not be directly addressed or involved at the moment. So, people can intelligently involve themselves in later stages of the conversation, having processed earlier information. Likewise, in a future version of our interface, various robots will interact with each other, much as individuals do when involved in a group conversation. Even though they might not be immediately involved in the interchange, they will have information from earlier parts of the dialog and will be able to join in intelligently later. Currently, our robots do not directly interact with each other, except to avoid collisions, of course. All interactions are between a human and a robot or group of robots. In future versions, we hope to incorporate robot–robot interactions.

An integrated system

By letting the user choose from a variety of input modes, we hope to incorporate ease and naturalness of interaction. However, this requires a system that integrates the various components to produce intelligent results. Such integrated components should share knowledge about each other and about the actions that are occurring, thereby reducing redundancy. Toward that end, we are implementing the *3-T Architecture*,[7] which integrates the interface with the various robotic modules that control navigation, vision, and the like.

This shared knowledge should produce a more intelligent system capable of more sophisticated interaction. Levels of independence, interdependence, autonomy, and cooperation should increase. The robot no longer has to be the passive recipient of commands and queries and purveyor of information. Instead, the system can infer what it needs to do and act accordingly, referring back to the human user when necessary. Humans or robots can initiate goals and motivations. We therefore are trying to build a system that provides *adjustable* autonomy; we call this a *mixed-initiative system*.

```
((imper    (:verb gesture-go
            (:agent (:system you))
            (:to-loc (:object door))
            (:goal (:gesture-goal there))) 0)
```

Figure 5. The natural-language-understanding system Nautilus translates the command "Go to the door over there" into this imper structure. The numeral 0 indicates the action is uncompleted.

Tracking goals

To obtain a mixed-initiative system, we use information from the dialog—that is, we track the interaction's goals. This involves tracking *context predicates* and implementing a *planning component*.

Context predicates

Our interface incorporates the natural language and gestural inputs into a list of context predicates. These constructs are the input's verbal predicates and arguments.[8] For example, in an imperative sentence (a command) such as "Go to the door over there," the verb *go* is the predicate and *door* and *over there* are the arguments.

When Nautilus processes this utterance, it translates it to a regularized form. Because this utterance is an imperative sentence, it regularizes to an imper structure (see Figure 5). These structures contain verbs, and depending on the verb's semantic class, certain arguments are either required or optional. In our domain, go belongs to the semantic class of gesture- go verbs. This class might or might not exhibit the arguments agent, to-loc, and goal. (When we say a verb might or might not exhibit a particular argument structure, we simply mean that the argument might or might not be present in the input signal—spoken utterance or PDA click. If it is not present, we can still reconstruct the argument structure because of the main verb's semantic structure.) These arguments, furthermore, take objects that themselves belong to certain semantic classes. In our example, door belongs to the semantic class of objects in our domain, and the adverb there belongs to the semantic class of gesture-goals.

When the robotic system receives this translation, it notes what action it must take (and checks if a gesture is needed). If the information is complete, the robotic system translates this expression into a command that produces an action, which causes a robot to move.

Context predicates also hold information about the goal's status. A placeholder in the representation contains information about whether or not the action has been completed. (In Figure 5, 0 means the action is uncompleted; 1 would indicate a completed action.) Once the action is completed, the system updates the information's placeholder. A system searching for uncompleted actions disregards completed actions. When the discourse changes focus, the system further updates the list of context predicates to remove redundancies and outdated information. In addition, the context predicate contains information such as which robot is being addressed, to assist in cooperative acts among the robots.

Employing context predicates in this manner facilitates mixed initiative because the various robot agents have access to the information and can act or not act on what needs doing. However, determining which robot acts in such a situation requires a planning component.

The planning component

Collaboration entails team members adjusting their autonomy through cooperation. Recent planning research indicates that collaborative work between multiple agents requires a planning component.[9] We implement this component through goal tracking. That is, our interface uses the list of context predicates to plan future actions. Such planning integrates knowledge of all the necessary actions, the completed actions, and the uncompleted actions.

For example, assume the user tells the robot to explore a particular area of a room, but the robot is interrupted while performing this task. After the interruption, the robot will be able to complete the task

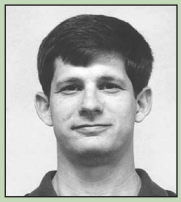
because it has a list of the goals it still must attain. Likewise, with a team of robots, the system might tell another robot to continue where the first left off.

Or, in a more interactive scenario, the planning components of the various robots can determine, by knowing the context predicates and the current situation, which robots will benefit most by completing a goal. The human doesn't have to remember what a robot was doing before the interruption or even which robot to direct to an uncompleted goal.

In this scenario, a robot that completes the goal earns "points," while one that does not complete its goal might lose points. Human-directed interruptions do not affect the score; that is, a robot that tries to achieve a goal but is interrupted to do something else will not lose points. Uncompleted goals are, in a sense, fair game; any robot can attempt to complete them.

Dennis Perzanowski is a computational research linguist in the Intelligent Multimodal Multimedia Group at the Navy Center for Applied Research in Artificial Intelligence at the Naval Research Laboratory in Washington, D.C. His technical interests are in human–robot interfaces, speech and natural language understanding, and language acquisition. He received his MA and PhD in linguistics from New York University. He is a member of the AAAI and the Association for Computational Linguistics. Contact him at the Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC 20375-5337; dennisp@aic.nrl.navy.mil; www.aic.nrl.navy.mil:80/~dennisp.

Alan C. Schultz is the head of the Intelligent Systems Section of the Navy Center for Applied Research in Artificial Intelligence. His research interests are in genetic algorithms, robotics, machine learning, adjustable autonomy, adaptive systems, and human–robot interfaces. He received his BA in communications from the American University and his MS in computer science from George Mason University. He is a member of the ACM, IEEE, IEEE Computer Society, and AAAI. Contact him at the Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC 20375-5337; schultz@aic.nrl.navy.mil; www.aic.nrl.navy.mil:80/~schultz.



William Adams works in the Intelligent Systems Section of the Navy Center for Applied Research in Artificial Intelligence. He received his BS from Virginia Polytechnic Institute and State University and his MS from Carnegie Mellon University. Contact him at the Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC 20375-5337; adams@aic.nrl.navy.mil; www.aic.nrl.navy.mil:80/~adams.

Elaine Marsh is a supervisory computational research linguist in the Intelligent Multimodal Multimedia Group at the Navy Center for Applied Research in Artificial Intelligence. Contact her at the Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC 20375-5337; marsh@aic.nrl.navy.mil; www.aic.nrl.navy.mil:80/~marsh.

Magda Bugajska is a computer scientist at the Intelligent Systems Section of the Navy Center for Applied Research in Artificial Intelligence. Contact her at the Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, DC 20375-5337; magda@aic.nrl.navy.mil; www.aic.nrl.navy.mil:80/~magda.

However, the planning component needs to take into account other factors, such as a robot's distance from a physical goal. So, if two robots are aware of an uncompleted goal—for example, obtaining an object on the opposite side of a doorway, the robot physically closer to the goal will earn points. The farther robot will lose points if it tries to achieve the goal, because the other robot is closer to that goal.

Taking into account dynamic factors, such as a changing environment[10] and a constantly changing dialog,[11] each robot assesses its role in completing an action. Unless the user directly orders a robot to complete an action, it must determine whether it will be rewarded or penalized. Achieving a goal becomes a function of one or more factors: a direct command or the immediate needs of the interaction (elements of the dialog), the changing situation, and who benefits most by completing an action.

By generating plans from goals and prioritizing them—almost on the fly—the robotic system can achieve the coordination only obtainable by systems that internally adjust and cooperate with systems that themselves are adapting to their changing roles and environment.

The two main bottlenecks we've encountered thus far are speech and vision recognition. To our knowledge, no commercial, off-the-shelf speech recognition system is robust enough for noisy office environments or on the battlefield, where gunfire and machine and equipment noises can obscure communication. Given these drawbacks, we are enhancing our gesture recognition component. In a preliminary study, we observed individuals in such noisy environments and noticed that gestures become larger to compensate for lack of audible understanding. Individuals might even use a set of predefined symbolic gestures for the group, such as Marines using symbolic hand gestures to communicate with each other during a battle maneuver. So, our gesture recognition component will not only process natural gestures but also incorporate symbolic gestures that might or might not accompany speech.

We are also working on changing our impoverished vision system to a binocular vision system so that the robots can perceive a wider range of gestures. We hope to

incorporate object recognition into this component.

Our work on incorporating a PDA into the interface is fairly robust; however, we would like to add GPS (Global Positioning System) technology. This would let our mobile robots traverse a wider area, knowing where they are, and communicate to a user their location, the locations of objects of interest, and the locations of other participants. Finally, as work in wearable computers progresses, we hope to adapt our PDA so that the user can wear it. This would involve making its screen light and flexible enough so that users can wear it, unencumbered by a handheld object, and making it touch sensitive so that a stylus is unnecessary.

Besides these hardware improvements, we plan to expand the dialog-based planning component. As individuals communicate their actions with each other and interact in various ways, the component will update or alter plans on the basis of information obtained in the dialog.

With these improvements, we hope to build a more robust and habitable multimodal interface. ■

Acknowledgments

The Naval Research Laboratory and the Office of Naval Research partly funded this research.

References

1. B. Grosz and C. Sidner, "Attention, Intentions, and the Structure of Discourse," *Computational Linguistics*, vol. 12, no. 3, Sept. 1986, pp. 175–204.
2. D. Perzanowski, A.C. Schultz, and W. Adams, "Integrating Natural Language and Gesture in a Robotics Domain," *Proc. IEEE Int'l Symp. Intelligent Control*, IEEE Press, Piscataway, N.J., 1998, pp. 247–252.
3. D. Perzanowski et al., "Towards Seamless Integration in a Multimodal Interface," *Proc. 2000 Workshop Interactive Robotics and Entertainment*, AAAI Press, Menlo Park, Calif., 2000, pp. 3–9.
4. K. Wauchope, *Eucalyptus: Integrating Natural Language Input with a Graphical User Interface*, tech. report NRL/FR/5510-94-9711, Naval Research Laboratory, Washington, D.C., 1994.
5. D. Kortenkamp, E. Huber, and R.P. Bonasso, "Recognizing and Interpreting Gestures on a Mobile Robot," *Proc. 13th Nat'l AAAI Conf. Artificial Intelligence*, AAAI Press, Menlo Park, Calif., 1996, pp. 915–921.
6. A. Schultz, W. Adams, and B. Yamauchi, "Integrating Exploration, Localization, Navigation and Planning with a Common Representation," *Autonomous Robots*, vol. 6, no. 3, June 1999, pp. 293–308.
7. E. Gat, "Three-Layer Architectures," *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, D. Kortenkamp, R. Peter Bonasso, and R. Murphy, eds., AAAI Press, Menlo Park, Calif., 1998, pp. 195–210.
8. D. Perzanowski et al., "Goal Tracking in a Natural Language Interface: Towards Achieving Adjustable Autonomy," *Proc. 1999 IEEE Int'l Symp. Computational Intelligence in Robotics and Automation*, IEEE Press, Piscataway, N.J., 1999, pp. 144–149.
9. B. Grosz, L. Hunsberger, and S. Kraus, "Planning and Acting Together," *AI Magazine*, vol. 20, no. 4, Winter 1999, pp. 23–34.
10. M. Pollack and J.F. Horty, "There's More to Life Than Making Plans," *AI Magazine*, vol. 20, no. 4, Winter 1999, pp. 71–83.
11. M. Pollack and C. McCarthy, "Towards Focused Plan Monitoring: A Technique and an Application to Mobile Robots," *Proc. 1999 IEEE Int'l Symp. Computational Intelligence in Robotics and Automation*, IEEE Press, Piscataway, N.J., 1999, pp. 144–149.