S44-61
108

Types of Verbal Interaction With Instructable Robots

C. Crangle and P. Suppes
Stanford University
Stanford, CA 94305

S. Michalowski
Veterans Administration Medical Center
Palo Alto, CA 94304

1. Introduction

An instructable robot is one that accepts instruction in some natural language such as English and uses that instruction to extend its basic repertoire of actions. Such robots are quite different in conception from autonomously intelligent robots, which provide the impetus for much of the research on inference and planning in artificial intelligence. ~~This paper~~ examines the significant problem areas ~~we see~~ in the design of robots that learn from verbal instruction. Examples are drawn primarily from our earlier work on instructable robots (~~[1]-[4]~~) and recent work on the Robotic Aid for the physically disabled ~~[5]~~.

~~We start our enquiry in Section 2 with a discussion of~~ natural-language understanding by machines. ~~In Section 3, we examine~~ the possibilities and limits of verbal instruction. ~~We also discuss~~ the core problem of verbal instruction, namely, how to achieve specific concrete action in the robot in response to commands that express general intentions. ~~The final section of the paper, Section 4, examines~~ two major challenges to instructability: achieving appropriate real-time behavior in the robot, and extending the robot's language capabilities.

2. Interpreting commands in context

Our work on the interpretation of natural-language commands rests on the assumption that many English commands can be precisely interpreted only in the actual situation in which they are issued [1]. Some examples are straightforward, *Go to the chair*, for instance. When there is more than one chair in the surroundings, which chair is being referred to? If only one chair is within the robot's field of vision, however, that chair may in many circumstances be taken as the correct referent of *the chair*. Another straightforward example, this time at both the syntactic and the lexical level, is the command *Move the cup to the right of the spoon*. This command is ambiguous in that *to the right of the spoon* may indicate which cup is to be moved or where some cup is to be moved to. Furthermore, *right of* may be interpreted relative to the speaker, the robot, or the spoon itself (taking it to face away from its handle). The topic of the previous discourse can help disambiguate the command, as can the actual arrangement of cups and spoons. If earlier commands have clearly established the robot's point of view as pre-eminent, that can suggest an interpretation for *right of*.

A third example, discussed in more detail, reveals the ways in which a robot must exploit the context in which a command is given to interpret that command. Of particular importance is the perceptual situation, by which we mean those aspects of the physical environment accessible to the robot through its sensory apparatus. Our example shows how the perceptual situation contributes to the precise interpretation of the word *next*.

The intuitive idea behind the semantics of ***next*** can best be understood if we talk about ***the next x***, where x may, for example, be ***chair***, ***table***, or ***wooden chair***. When we say ***the next x***, we are referring to the first x, by some ordering relation, relative to some present reference entity. Three things have to be fixed by the context for the interpretation of this word: the ordering relation, the class of x-type entities from which one will be selected, and some encompassing class of entities which are ordered by the relation. This encompassing class must be specified because it makes perfect sense to talk about the next x even when the present reference entity is not itself an x. A clear example is given by the robot emulator of Maas and Suppes which accepts instruction in elementary mathematics ([2]-[4]). In the usual contexts of use for *next*, the robot has been, and is expected to continue, scanning down a column. Thus for most uses of *next* in the arithmetic instruction context, the ordering relation required by *next* is given by the relation **vertically below**, a strict partial ordering on the perceptual objects (the digits and blank spaces of an arithmetic exercise) such that each perceptual object that has a successor has a unique immediate-successor by this relation and similarly for predecessor. Suppose the robot is focused on the blank space at the top of the tens column of an arithmetic exercise. That blank space plays the role of the reference entity for the interpretation of *next* in the phrase *the next number*. For that blank space to function as the reference entity for *next number*, both the digits (numbers) and the blank spaces must stand in the relation **vertically below**.

The perceptual situation will not always have to yield the semantically important information for *next*. These may be set explicitly by the verbal command. Consider, for instance, the command *Choose the next person in order of height* where the ordering relation is given by the phrase *in order of height*. In the absence of such explicit directions, however, the perceptual situation imposes its own choice of ordering relation. For instance, suppose the robot is in a room containing ten chairs arranged in a row. That very arrangement of objects will tend to establish an ordering relation for sentences in which the adjective *next* qualifies the noun *chair*. If the robot were positioned alongside the second chair, facing down the row towards the third chair, and if there had been no prior discourse, the command *Go to the next chair* would probably be interpreted as a command to move to the third chair. It is clear that the appropriate ordering relation must not only be available perceptually (or by some other means such as memory), it must also be established as a focus of attention. If the robot has no ability to adduce an ordering relation from the perceptual situation, the first time the adjective *next* is used to refer to objects of a certain type, the robot must query the user for help in fixing an ordering that is known to it, which should subsequently be used as the default unless explicit instruction changes it.

Sometimes two of the three contextual factors required by *next* are set explicitly by the command. Consider the room containing only the row of chairs again, with the agent at the second chair in the row. Suppose the agent were being instructed to clean the wooden chairs by applying a furniture polish, and the row included two cane chairs, one of which was in the third position and the other in the eighth. The command *Clean the next wooden chair* would then direct the agent to the fourth chair in the row, the first wooden chair relative to the present chair. In this case, the adjective *wooden* specifies the class of wooden objects, of which one must be selected, and the noun *chair* specifies the encompassing class of chairs, both wooden and cane.

There are many different ways a command may specify the contextual factors required by *next*. Consider the command *Go to the next chair to the left*. Here *to the left* specifies the ordering relation, a relation, call it **L**, which could be defined informally as follows: for all **a** and **b**, **aLb** if and only if **a** is positioned to the left of **b** and within the compass of an arc of 30 degrees radiating horizontally from **b**. Consider, however, the command *Go left to the next chair*. Here *left* does not make a contribution to the interpretation of *next*; it serves rather as an adverb directly qualifying the verb, acting as an extra constraint on where to go. There are many other examples like this. In the command *Search for the next file in alphabetic order*, the ordering relation behind the use of *next* is given explicitly by the phrase *in alphabetic order*. In the command *Search from A to Z for the next file*, on the other hand, that same ordering relation defines a direction in which to search, but leaves open the question of what ordering lies behind the use of *next*.

Contextual information is also required to fix the interpretation of intensive adjectives, such as *large*, and comparatives and superlatives, such as *larger* and *largest*. The adjective *large*, for instance, may be thought of as a procedure that uses an underlying ordering relation of size to determine if an object, the one said to be large, stands in the appropriate size relation to some criterion object. This criterion object is also given by the context. What counts as a large book in the context of a shelf of dictionaries is not the same as in the context of a shelf of poetry volumes. Perhaps the most striking example of the role of the criterion object is given by the phrases *large elephant* and *large ant*. While the ordering relations for *large elephant* and *large ant* will both use some measure such as mass or girth, the criterion objects will be quite different.

Our emphasis on the role of context leads inevitably to another emphasis, namely the essential role that interaction must play in the interpretation of natural-language commands -- interaction between the robot and the user and between the robot and the perceptual situation. The next section examines verbal instruction in more detail, at the same time identifying its place within a spectrum of interactions between robots and humans.

3. A place for verbal instruction

Two types of verbal interaction with robots may be identified, one in which learning occurs as a result of the interaction and the other without learning. When there is no learning, the robot responds to each verbal command or enquiry as it is given, never using its experience to extend its basic repertoire of actions. In our work we refer to such a robot as a commandable robot. The mobile base of the Robotic Aid (see the companion paper in this volumn, [5]) is commandable in that it obeys a range of motion commands expressed in English, commands such as *Whenever you are within three feet of the ramp, stop*. A commandable robot may be given detailed step-by-step instructions to open the door of a microwave oven, insert a plate of food, close the door, set the timer, and switch the oven on. Yet the next time the user wants the robot to prepare a meal, the same or a similar set of detailed instructions has to be issued. There are obvious advantages if the user could give that behavior a name, such as *Prepare the meal*, and use that name later to invoke the behavior. In this way, the robot would have learned from its verbal interaction with the user.

This prescription -- issuing a sequence of commands, baptizing the sequence, and invoking it later by name -- describes just one of many possible forms of instruction. There is also non-verbal instruction, as presently provided by the head-tracking mechanism of the Robotic Aid, for instance, which allows the user to describe a trajectory for the robot to follow. Verbal correctives, such as *Slow down!*, given while the robot is in motion are also important in communication. And non-verbal means of correction also have their place. Nonverbal methods are extensively used in the training of animals, by direct procedures of reward and punishment, and they have also been used in simple experiments with very elementary robots learning mazes. More sophisticated examples arise when the robot or system in question has a criterion for evaluating correctness of its responses, as for example in speech recognition systems where parameters must be adjusted to individual speakers. The operator does not know how to do this; the robot or system learns to adjust parameters by the correctness of its responses. It learns about the correctness of its responses by comparing its guess with the given correct answer. It does not learn how to make corrections by being given verbal instruction on the parameter adjustments that are needed. Clearly, verbal instruction is but one of several ways of producing corrective and adaptive behavior in robots.

Some important general points about verbal commands must be discussed before we examine instructability in any detail. Take the command *Pick up the cup and put it on the saucer*. This command expresses the result we would like to see. It says nothing about the process of achieving that result. Typically, ordinary language, like ordinary conscious thinking, is oriented toward results not processes. The detailed movements that are part of some action -- either one we intend to take ourselves or one we want the robot to take -- are not easily accessible to our conscious thinking and in fact for some actions quite beyond the descriptive powers of ordinary language. Two examples: we cannot verbally describe a specific trajectory to be followed in crossing a room nor can we describe the exact motion of the roll of a die from the instant of its being thrown until it comes to rest. Many actions we would want the robot to perform are for us a matter of automatic, that is, unreflective response -- flicking a switch, picking up a cup, using a screwdriver -- and are in fact actions that are seldom acquired by us through explicit verbal instruction. Other activities we would require of a robot are more amenable to verbal description -- manipulating a toggle switch, navigating with reference to objects in the environment, for example. Many tasks are ideally suited to explicit verbal instruction. The elementary mathematics emulator mentioned earlier is especially designed for primarily verbal instruction, but other kinds of robots dealing with physical equipment also engage in tasks suited to explicit verbal instruction. A good example is the activity of assembling and disassembling a piece of equipment. Not every motion involved in the assembly or disassembly is described but what is described explicitly in words is the sequence in which disassembly and assembly should take place. Also well suited to verbal instruction is the transfer of information about objects. Here the user helps the robot learn to recognize objects by directing its sensors to specific parts of the object, naming those parts, and letting the robot use autonomous procedures to determine their shape and location. The user can also provide information that is not accessible to the robot's sensors, such as what the object is used for.

There is a further complexity to actions and their verbal descriptions that we must face. In requesting action, from a robot or a human, in terms of a result description such as *Bring me the book on the table,* we seldom have in mind a detailed algorithm for executing the command. The particular path taken by the agent satisfying the command is not part of the meaning of the expressed intention. On the other hand, if the agent knocks over a chair in fetching the book, in ordinary circumstances we regard the movement of the agent as satisfying only partially or rather poorly the request made. Similarly, if when asked to pick up a cup the agent spills its contents, we do not consider the request to have been fully satisfied. Expressed intentions carry with them a bundle of ceteris paribus conditions that impose a variety of constraints on the specific procedures actually executed. These ceteris paribus conditions are not given concretely or in advance but depend on the particular context in which an action is carried out.

The semantics of a command such as *Pick up the cup* thus apparently has conflicting demands to meet. In the first place, this intention, expressed in terms of a result, must for its satisfaction be interpreted to produce a specific action-process. That is, a specific procedure must be executed. (We do not of course necessarily mean a simple sequential procedure; a highly parallel complex collection of processes may be involved. The point is that out of the many distinct actions that could take place to pick up the cup, one specific one is taken in a given situation.) We cannot specify in advance a particular set of motions for the specific action-process. Such specifications are not part of the meaning of the command and they would too narrowly delimit the contexts in which the command could be given. At the same time, however, there are the many ceteris paribus conditions we expect to be met in the satisfaction of the command.

As with the interpretation of individual words, the key lies in the context. We want specific action to be produced in response to a command but we cannot explicitly build the details of that response into the semantics. These details are to be taken from the actual situation in which the command is given. In this way they will not have been inappropriately specified in advance and they will include those ceteris paribus conditions accessible from the context. Take the example of *Pick up the cup*. The particular motions of the joints and the gripper that will pick up a given cup in a given situation cannot be specified in advance. What can be specified are generic procedures for moving the arm which are selected and combined as required by the fact that a cup not a book is to be picked up, by the present position of the cup, by its dimensions, and by the nature of its handle if present -- in other words, by the context. That is the challenge we face devising generic procedures which can be combined to accomplish a wide range of navigation and manipulation tasks as demanded by the specific context in which a command is issued. Just as important as the procedures themselves is the control environment in which the procedures execute, for it is this environment which determines the temporal and logical connections allowed between procedures. And there is the parallel challenge of devising the rules of semantic interpretation that connect the surface structure of a command, that is, the English words in their given order, with the executing procedures.

Our design of the commandable base of the Robotic Aid and of the natural-language interpreter for it bore these considerations in mind. A range of motion commands can now be successfully interpreted and obeyed in the context of a room containing fixed items of furniture. An important next step in this work will be to interpret commands that better exploit the perceptual functioning of the robot (which is still under development) for it is through perceptual functioning that many details of the context are made known to the robot, especially in a changing environment.

There is one tempting approach to the problem of achieving specific action in response to requests that entail few specifics in their expression as natural-language commands. The approach is a familiar one in programming practice, namely specifying defaults that operate in the absence of explicit information and that are overridden by the presence of explicit information. So, for instance, a picking-up procedure would be designed, one that as a default looked for and used the handle of the object and that moved the object at a default speed, one that for most liquids and most cups would prevent spilling. While we accept that some default specifics will inevitably be built into the procedures of an instructable robot (our motion procedures for the mobile base of the Robotic Aid individually move the robot at a default speed), we have two reasons for rejecting this as a general solution to the problem of achieving specific and appropriate behavior in response to natural-language commands.

First, such an approach will make robot instruction too much like programming: everything of importance must be anticipated. Secondly, the problem of overriding defaults in real-time is non-trivial. The "solution" offered through defaults does not reduce the technical difficulty of achieving appropriate behavior in a robot. What we propose rather, and therefore acknowledge as an important part of the research effort in instructable robots, is that the robot's initial understanding of explicit verbal commands must be adjusted over time through learning. Here we have in mind forms of learning studied extensively in psychology, learning that advances by making successive discriminations and by generalizing from past experience. In learning to make discriminations or generalizations along any dimension that has a continuum of values it is essential that smoothing distributions of some sort be added to the experience gained from specific learning trials. Detailed mathematical analyses of such smoothing procedures and their application to learning data are to be found in Suppes [6]. Other forms of machine learning have been explored in artifical intelligence research and they too are relevant. We mention just a few key studies here, all to be found in Michalski, Carbonell, and Mitchell [7]: learning by experimentation (Mitchell, Utgoff and Banerji), learning from examples -- a comparative review (Dietterich and Michalski), and learning from heuristic-guided observation (Lenat).

To return now to instruction, we can see how the same set of concerns outlined for the semantics of a command such as *Pick up the cup* surround verbal instruction. Ordinary language, as we pointed out earlier, is oriented towards results not processes. Giving explicit verbal instruction that details a step-by-step process will not be easy for many actions. For some, it will in fact be inappropriate and should be forgone in favor of other forms of learning. But even for those tasks for which verbal instruction seems suitable, our instruction, to be concrete and testable, will often be aimed at the execution of a specific action whereas what we really want in the end is for the robot to take whatever specific action is appropriate in the context. To take a simple example, we may instruct the robot to pick up a cup by finding and grasping the handle and then raising the cup without disturbing its vertical orientation. But when there is no handle we want it to grasp across the rim and if the cup is empty we want the robot to tilt the cup as necessary to get it through a constricted space. The final section of this paper therefore addresses the following problem. In giving explicit concrete instruction how are we to ensure that the robot will later exploit the context in which is it operating to successfully perform the action?

4. The challenge of instructability

To examine the problem posed at the end of the last section, we take for discussion a simple example. Suppose we want to teach a mobile base equipped with sensors to circumvent an obstacle it has encountered. Specifically, we want the robot to "bounce" its way around the obstacle by retreating from it, moving to one side, and then advancing in the original direction of travel. We represent two such cases in the figure below, indicating the mobile base by a triangle with the front of the robot shaded in. This recoil action is to be repeated each time the sensors detect the obstacle, until the robot has moved beyond it. Suppose we, as the operator, start by teaching the robot the following basic recoil behavior. We assume that when the robot encounters the obstacle, it is facing in the direction of its travel and that when the robot moves left in response to a command to go left, it retains its forward-looking orientation. We issue the following set of commands:

Stop moving

Go back twelve inches
Go left twelve inches
Carry on as before

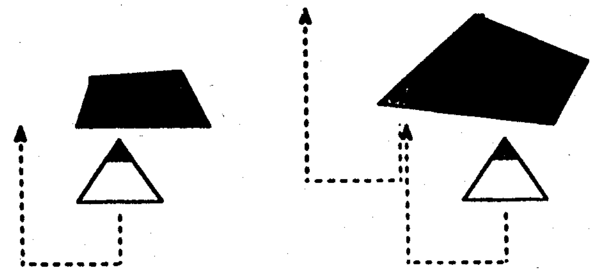

Figure 1

These commands are assembled off-line and given the label *Recoil* after which the action described by them is tested for appropriate real-time behavior. That is, the commands are interactively interpreted and obeyed in a particular situation. Only through interaction will the intended meaning of *back* and *left* be established. That interaction must establish whether *back* is relative to the direction of travel (a possible interpretation only if the robot possessed path-following behavior, not now present in the Robotic Aid) or whether it is relative to the direction the robot is facing. If the robot were moving backwards as it approached and hit the obstacle (contrary to our stated assumption that the robot is facing in the direction of its travel), this second interpretation should not be considered. But to eliminate it would require a highly sophisticated understanding of the intention of the instruction -- that it was to avoid the object not push it, for instance. In the absence of such understanding by the robot, a short dialogue with the user must establish one of the two interpretations. Interaction is also required for *left*: is it left relative to the operator or the robot? Again, a brief dialogue with the user establishes the desired interpretation.

For the two cases depicted in the figure above, the recoil routine would produce satisfactory results with one call to recoil for the obstacle on the left, two for the obstacle on the right. If in such early test situations the recoil sequence proves satisfactory, the operator can embed the *recoil* command in the more general command *Whenever the bumpers are hit, recoil.* At this point, the operator would have made certain assumptions about the physical environment in which the robot will be obeying these commands -- for instance, that the obstacle is not shaped as in the figure below. In such a case, the robot would hit the object again during its leftwards motion and when another call to the recoil routine were issued, the robot would be unable to go back except by scraping along the edge of the object, prompting repeated calls to recoil that, as they were successively executed, would steer the robot far to its left, significantly off course.

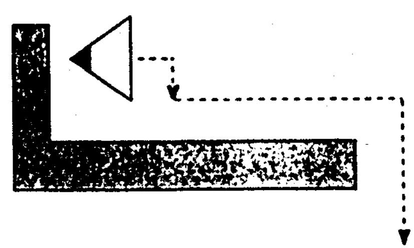

Figure 2

The operator has also assumed that the robot is not acting under the constraints of other general commands. Suppose, for instance, the following command had been issued earlier: *Whenever you are within one foot of the chair go right one foot.* And suppose the chair is immediately to the left of the obstacle in such a way that as the robot took its leftwards step during the recoil action it came within on foot of the chair. The robot would never complete the leftwards motion and so never finish the recoil action and resume its original motion. Under such circumstances the operator should be able to interrogate the robot about its behavior. In

answer to the operator's enquiry, the robot should indicate (verbally or graphically) that it is reacting to the earlier command. Note that such interaction between robot and operator requires a degree of "self-understanding" by the robot.

At this point we can see that the successful execution of the learned recoil routine depends on two factors. First, there must be a congruence between the robot's and the operator's perception of the physical environment. Although the perceptual situation does not have to be perfectly comprehended by either the operator or the robot -- it is not necessary to know exactly how many obstacles are present and where they are nor their precise dimensions --, the operator's judgement about the absence of irregular shapes does have to be consistent with the robot's perception of the objects through its sensors. Secondly, the operator must have an accurate understanding of the robot's functioning. The operator must know, for instance, that the robot "remembers" what it is doing when told to stop moving and recalls that action when told to resume.

It is easy to produce other examples where both these factors are critical to instruction. For example, suppose the user issues the following sequence of commands for changing the arrangement of furniture in a room, giving the sequence the name ***Rearrange the furniture***

Move towards the table while avoiding the chair
Go to the other side of the table
Go west two feet and north six inches then face the table and move it east two feet
Now go to the back of the chair, without hitting the chair
Face the chair and move it forward until it is within one foot of the desk

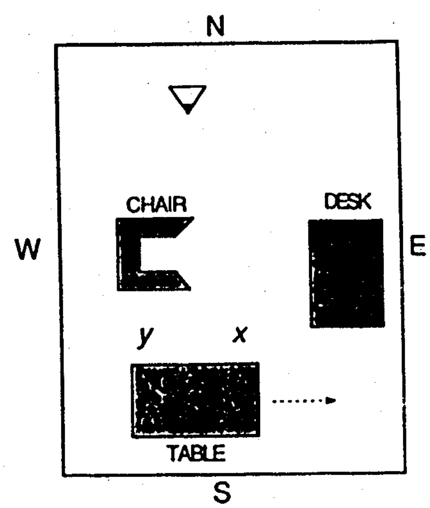

Figure 3

The purpose of these commands is to move the table, as shown by the arrow, towards the center of the room and to shift the chair up to the desk. But if the robot's position were slightly more west than depicted, and if the robot were the Robotic Aid, it would approach the table via the west side of the chair, not the east as the commands presuppose. Consequently, the robot's position before obeying *Go to the other side of the table* would be somewhere around y, not x as required for the successful execution of the rest of the sequence of commands. The problem may seem to lie in the non-determinism of the robot's behavior. If when we said *Go to the table* we knew exactly where the robot would stop, we would have no trouble choosing the right commands. Or so the argument would go. But, as we have discussed, natural-language instruction demands flexibility in the interpretation of commands, as shown by our normal and customary use of English. We freely use any of the following commands and each time impose no restrictions, other than those explicily given, on how they should be obeyed: *Go to the table; Go to the table, avoiding the chair; Go to the table by skirting around the back of the chair; Go to the table, performing pirouettes along the way.* The interpretation of a natural-language command should ideally introduce no restrictions that are not explicitly given.

These two examples, as simple as they are, suggest that an operator will seldom put together the right set of commands the first time around. The operator can try to anticipate the execution environment by including appropriate conditional commands in the instruction, commands such as *Recoil whenever your bumpers are hit* or *Go forward until you see the line*, commands that exploit the robot's sensory capabilities. That will not be enough, however. What is needed is for the robot to accept real-time adjustments to its behavior as it obeys commands. If, for instance, the robot encounters an irregularly shaped obstacle during the execution of the recoil routine, one that prevents it from completing the recoil action, the operator should be able to adjust the robot's motion by giving a corrective command such as *Move right a little!*. Furthermore, in the end, we will want more than real-time response to corrections. We will want the robot to incorporate adjustments into its learned routines. Such a capability is beyond our present endeavors but we acknowledge its importance and inevitability in our program. It would be intolerable if the operator were to be responsible for the repeated correction of an action. For instance, if initial instruction to the robot caused it to push too hard on a button so that on its first test run the user intervened by saying *Pull back!*, the user would not want to monitor each pressing of a button to give the same corrective command.

The discussion in Section 2 of this paper anticipated several of the remarks in this section about the real-time testing of verbal command sequences. In Section 2 we saw how many ordinary English words can be precisely interpreted only in the actual situation in which they are used and only in interaction with the person using the word. In light of that semantic fact and of the remarks above, it is clear that the off-line assembly of command sequences is less useful than the more complex form of instruction we call real-time trial-run instruction. In this form of instruction the robot immediately obeys each command as it is given, interpreting the command in context and in interaction with the operator who, in monitoring the robot's performance, is adjusting it as necessary by interjecting other commands to achieve the desired result. During this time, the robot assembles the interpreted commands, together with the real-time adjustments made by the operator, to produce a new routine that it adds to its repertoire of actions.

One important point about these newly learned routines should be mentioned. To allow the robot to fully exploit information in its environment, routines acquired by the robot through instruction should be more than simple "in-line macros;" they should be parameterized subroutines. Such routines will rely on learning through discrimination and generalization to set the right parameter values. For each parameter, a smoothing distribution of a given form with a given variance will be assumed, that is, built in to the robot. For mathematical simplicity, parameter changes may be restricted to adaptation of the mean of the distribution (for examples, see Suppes [6]). In many cases, however, we expect that it will be essential during learning to modify the variance as well as the mean of the distribution to enable the robot to adapt fully to the environment.

We introduce one more example at this point to emphasize the role of interaction in instructable robots. Interaction between the robot and its environment was essential to the robot emulator that was taught elementary mathematics. Here is a brief description of one interactive encounter to illustrate the kind of solution offered by interaction. Consider the following commands which appear in sequence as part of the instruction on how to add multi-digit figures.

> *Look at the next spot down until you see a number or a bar*
> *If it is a number then add it to the total so far and remember the sum*
> *Continue looking down, looking for a number or a bar, adding and remembering until you see a bar*

The third command makes reference to the preceding two steps, forming a loop. It is not clear whether this loop, when compiled as a program, should be top-tested or bottom-tested or tested in the middle. That is, it is not clear where the test for the presence of a bar should be placed. In fact, for well-formed arithmetic problems, a bar can never appear during execution of the top or bottom of the loop; it can appear only during the execution of the middle of the loop. The operator lets the robot emulator discover this fact by running the program on test data, that is, on a particular addition exercise. Tentatively, the emulator places the exit condition after each step of the loop. The loop is then run in trial mode, showing the operator what takes place. If an exit condition, when encountered, is not satisfied, control is simply passed to the next step of the loop. If, however, the exit condition is satisfied, the operator is asked if this is the right time and place to stop repeating the sequence of steps gathered into the loop. If the operator answers *Yes*, the exit condition is installed at that location and all remaining trial locations are eliminated. If the operator answers *No* (it is not the time and place to stop repeating), the exit condition is removed from that location since it cannot be consistently satisfied and chosen as the right exit point on a later pass. If all possible exit places are rejected in this way (unlikely since in most loops the exit condition will not be satisfied at all places), a fatal error is signalled. A successful interactive session will locate the exit condition in the right place. The robot's ability to be instructed thus lies in its capacity to resolve ambiguities (such as exactly when to stop repeating a set of steps) through attempting to follow a given instruction and interacting with the operator and with test data in its execution environment.

To summarize briefly, we have identified several forms of interaction that contribute to the operation of verbally instructable robots. First the robot must interact with its perceptual environment in interpreting individual words, individual commands, and sequences of commands to resolve the inevitable ambiguities that characterize ordinary language. Secondly, the robot must interact with the operator to resolve those ambiguities when its perceptual abilites are limited or when the operator's intent must be determined. Thirdly, the robot must interact with its perceptual environment to meet those ceteris paribus conditions that accompany natural-language commands. Fourthly, there must be interaction between the robot and the operator to ensure that they share a common understanding of the perceptual environment and of the robot's behavior. Lastly, the robot must accept and acknowledge real-time adjustments to its behavior.

We end this paper with one further major challenge in the design of instructable robots. The problem lies in naming new routines. The suggestion made above, without comment, was to name a routine such as the one for moving furniture *Rearrange the furniture*. That raises the problem, however, of how to integrate this new use of language into the robot's lexicon and grammar. The robot can easily be made to respond to the phrase *Rearrange the furniture* as an unanalyzed semantic whole. But if the robot is to respond naturally to the following commands where *rearrange the furniture* appears embedded in a compound command and the verb *rearrange* occurs in the past tense and in the declarative mood, the robot's lexicon will have to have appropriate entries for *rearrange* and *furniture*.

Rearrange the furniture without bumping the cat
Switch off the lights after you have rearranged the furniture

The extent of the problem can be seen when we ask what is an appropriate entry for *rearrange*? First, the category of the verb has to be correctly assigned so that the verb's occurrence in a range of sentences can be recognized. In addition, various grammatical features have to be identified and correctly assigned. Consider the verbs *turn* and *face*, for instance, which at first glance seem to require parallel syntactic treatment that could be achieved simply by assigning the words to the same category:

Turn towards the wall	*Turn away from the wall*
Face towards the wall	*Face away from the wall*

But, of course, *turn* may be used in ways for which there is no natural parallel for *face*, and similarly for *face*, as suggested by the examples *Turn to the wall* and *Turn clockwise until you are facing the wall*.

The most challenging problem lies with the semantic entry for a new verb. This problem affects the very choice of instruction tasks, particularly what tasks should be taught first. For instance, we could choose for initial instruction those activities that correspond to single verbs and label the learned routines by those verbs. The problem of embedding this new word in the robot's lexicon and grammar does not become any easier however. Tense and mood remain semantically significant. Furthermore, one verb may be used to express many different intentions, which will give rise to many different interpretations. Consider the following three commands.

Move towards the table
Move three feet forward
Continue going towards the door until you have moved forward six feet

Each of these commands expresses a distinct intention and consequently, despite the fact that the verb *move* occurs in each command, distinct procedural interpretations are produced by the mobile base of the Robotic

Aid. The first command uses the **RegionSeeking** procedure, the second the **Piloting** procedure, and the third the test procedure **DistanceCovered?**. The partially specified interpretations of these commands are as follows. Full details of the interpretation process can be found in [5].

Move towards the table
(Sequence (RegionSeeking <the region around the table> Towards))
Move three feet forward
(Do (Piloting Shift Forward) (DistanceCovered <three feet> Forward))
Continue going towards the door until you have moved forward six feet
(Do <going towards the door> (DistanceCovered? <six feet> Forward))

No one semantic entry for *move* suffices for these three uses of *move*. If this were the new word being taught to the robot, that semantic complexity would also have to be acquired.

While we have not yet attempted any verb acquisition in our work on instructable robots, we once again recognize the role that interaction will play in it and report here on related work by Haas and Hendrix (also to be found in [7]). The goal of their work was to create a computer system that could hold a conversation with a user in English about a subject of interest to the user and subsequently retrieve and display the information conveyed in the conversation. Whenever a new word was presented to the system, a special procedure was called that temporarily assumed control of the dialogue and prompted the user for relevant information. The system would try to find out if a verb was transitive or if it took an indirect object, for instance, by introducing sample sentences and asking the user to complete them if they displayed acceptable uses of the verb. The system would also ask directly for the -ed and -en forms of the verb, showing the user the example of *went* and *gone* for *go*. Interaction with the user was thus exploited to obtain important syntactic and semantic information about a new word.

5. Conclusion

Through our initial efforts with two robot systems -- one in simulation, the other implemented in hardware -- we have identified several ways in which interaction (between the robot and its perceptual environment and between the robot and the operator) is essential to instructable robots. Such robots need interaction to interpret ordinary English commands in context, to determine the intentions of the operator when a command or sequence of commands is ambiguous, and to ensure that the robot and the operator share a common view of the environment and of the robot's functioning. At the same time, our work has shown that explicit verbal instruction must be accompanied by other forms of communication and learning if the robot is to function successfully in its environment.

6. References

[1] Suppes, P. and Crangle, C. (in press) Context-fixing semantics for the language of action. In Taylor, C., Dancy, J., and Moravcsik, J. (Eds.), **Language and Value**. Stanford, Calif.: Stanford University Press.

[2] Maas, R.E. and Suppes, P. (1983) Natural-language interface for an instructable robot. Technical Report No. 306, Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, Calif.

[3] Maas, R.E. and Suppes, P. (1984) A note on discourse with an instructable robot. *Theoretical Linguistics* 11: 5-20.

[4] Maas, R.E. and Suppes, P. (1985) Natural-language interface for an instructable robot. *International Journal of Man-Machine Studies* 22: 215-240.

[5] Michalowski, S., Crangle, C., and Liang, L. (this volume) A natural-language interface to a mobile robot.

[6] Suppes, P. (1964) Some current developments in models of learning for a continuum of responses. (Discrete Adaptive Processes Symposium, American Institute of Electrical Engineers, June 1962.) *The Institute of Electrical and Electronics Engineers Transactions on Applications and Industry* 83: 297-305.

[7] Michalski, R.S., Carbonell, J.C., and Mitchell, T.M. (1983) **Machine learning: An artificial intelligence approach**. Palo Alto, Calif.: Tioga Publishing Company.