Interactive Robot Task Training through Dialog and Demonstration

Paul E. Rybski, Kevin Yoon, Jeremy Stolarz, Manuela M. Veloso
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, 15213
{prybski,kmy,jstolarz,mmv}@cs.cmu.edu

ABSTRACT

Effective human/robot interfaces which mimic how humans interact with one another could ultimately lead to robots being accepted in a wider domain of applications. We present a framework for interactive task training of a mobile robot where the robot learns how to do various tasks while observing a human. In addition to observation, the robot listens to the human's speech and interprets the speech as behaviors that are required to be executed. This is especially important where individual steps of a given task may have contingencies that have to be dealt with depending on the situation. Finally, the context of the location where the task takes place and the people present factor heavily into the robot's interpretation of how to execute the task. In this paper, we describe the task training framework, describe how environmental context and communicative dialog with the human help the robot learn the task, and illustrate the utility of this approach with several experimental case studies.

Categories and Subject Descriptors: H.4 [Information Systems Applications]: Miscellaneous

General Terms: Algorithms, Experimentation

Keywords: Learning by demonstration, Human-robot interaction

1. INTRODUCTION

The field of human-robot interaction (HRI) is developing rapidly as robots become more capable of operating with people in natural human environments. For robots to be accepted in the home and in workspaces, people will need to be able to interact with them in a natural and familiar fashion. Robotic sensing, cognitive, and actuating capabilities will need to achieve a certain level of complexity such that humans can treat them more as teammates or partners in order for the research community to reach this goal. Such enabling capabilities include the ability to recognize the presence and activities of nearby people, possess a spatial and semantic notion of the shared environment, and understand (a subset of) natural human languages. By allowing robots to behave and interact more socially with and around people, we believe that they will more readily be accepted by non-technical individuals as part of their daily lives and routines.

One of the most important methods that humans have for communicating with each other is spoken language [14]. We believe that a crucial challenge for the field of human-robot interaction is to enable a robot to be taught a task (or a set of tasks) by a human in the same fashion that the human might teach another person. In this paper, a task is a temporally ordered set of operations. Tasks can have multiple outcomes depending on the state of the environment and the objects or people within it. We are interested in exploring algorithms that would allow a human to show a robot how to accomplish a task in a natural fashion and to have the robot interact in an intelligent dialog with the human in order to verify the correctness of the task.

In this paper, we describe a set of algorithms for a mobile robot that will allow it to learn a complex task from a human through a combination of spoken commands, observation and imitation of a human performing that task, and engaging in simple spoken language dialog to verify and append to the learned task sequence. Our approach to task learning allows the robot to iteratively learn different outcomes of subtasks by querying the human for more information. By engaging the robot as an active partner in the task training process, we believe that more complex tasks can be trained.

2. RELATED WORK

Robot task learning through interaction with humans is an area that has been studied in a number of different areas of research. In [19], a human teaches a robot arm through multiple demonstrations how to maneuver through complex configurations using guarded moves. In [7], a robot arm is taught to build blocks in a controlled environment where a human shows the required configuration of blocks to the robot and the robot replicates them. In [5], a robot arm attempts to generalize its motion from multiple examples of being teleoperated through a series of actions. Additionally the robot asks the user for its intentions of each step so that it can generalize by knowing, for example, that a block had to be moved just because it was in the way of another, not because it was an essential part of the task. In general, the

HRI'07, March 8-11, 2007, Arlington, Virginia, USA.

literature on human/robot task training where the robot is a manipulation device focus primarily on recognition and plan building where the generated plans are robust to objects being in different initial conditions. In contrast, when training mobile robots through demonstration, the focus is more on the mobility of the platform and understanding where the human is in relation to other objects and features in the environment. In [16], Hidden Markov Models (HMMs) are used to learn and classify detected human actions as gestures. The robot performs a basic block distribution task after following the human teacher to learn the context of where the blocks should be sorted. The paradigm of learning from observation has also been used very successfully to train humanoids on specific tasks [4] as well as action generation [1]. Our interests are primarily with mobile platforms that will interact directly with humans in their environments. Techniques for learning control policies via user input have been explored in [20] as well.

Hierarchical task decomposition is a powerful representation for segmenting large tasks into more reasonable subtasks. In [6], a behavior framework is defined for teams of AIBO robots. Behaviors are used to control both individual robots and the entire team, with the team behaviors having additional constraints that facilitate multi-robot control. In [17], a framework for a human teaching a robot from basic sequences, to tasks, to behaviors is defined where the teacher can construct these in a hierarchical fashion. Our approach also makes use of a hierarchy of tasks in order for more complex tasks to be constructed out of simpler building blocks.

The use of speech as a mechanism for task training is powerful because of the amount of information that can be transferred. A domain in which a robot navigates in a miniature town is defined in [8]. In this work, a mapping of phrases to simple primitives and the combination of these primitives when more complex phrases or instructions are given is defined. The complex 'behaviors' can then be referred to in subsequent instructions for the robot. If the robot is incapable of following the instructions, it asks for clarification. The use of spoken dialog for disambiguation of spoken utterances and to request additional information is important as it facilitates the understanding between human and robot. Dialog with a human has been very successfully employed in [18] for understanding human perspectives and resolving linguistic ambiguities. Another effort which uses a grounded situational model for learning percepts from human speech is described in [10]. We believe very strongly that the mobile robots should engage the human in dialog in order to gain a better understanding of their world and a large part of our effort focuses on how this can be done.

Our work is most closely related to the human/robot task training research of [11, 12]. In this body of work, a robot observes a human doing a task and learns to associate specific behaviors with the actions that the human is performing. The human speaks to the robot to tell it when to pay attention and when to perform some simple tasks such as picking up a block. Multiple training sessions on the same task are generalized to form a single representation. Finally, if the robot is having problems with a task, it attempts to communicate this by getting the human's attention and trying to perform the difficult task while he or she is watching. Our research also falls in the category of plan learning [21, 22], whereby we are interested in methods by which the domain-specific plans can be created through the generalization of observed data. Our focus is on learning plans interactively.

Our approach to the problem of a robot learning a task through human observation is to combine prior knowledge and contextual information, including environment features and the presence of nearby humans, with the use of spoken language dialog. We believe that mobile robots that interact with and learn from humans will be required to make heavy use of both. The use of question asking and dialog eliminates some need for repeated viewing of a task because contingencies can be told to the robot rather than requiring the robot experience them.

3. USING SPEECH TO BUILD BEHAVIOR NETWORKS

Our goal is to investigate methods by which a human can teach a task to a robot in a natural fashion such as how humans might interact with each other. When considering how humans teach tasks to each other, we note that a combination of both demonstration and verbal instructions can be used. If the teacher and the learner both have shared knowledge of all of the concepts that are referenced, then the teacher only needs to verbally describe the task sequence. However, as is more often the case, a combination of both demonstration and verbal descriptions must be used in order to successfully teach a human a new task. Our behavior learning architecture supports a combination of these methodologies. In order to derive a mapping from human actions (and speech) to robot actions, the robot requires both a set of behaviors that will allow it to perform the tasks required of it as well as behaviors necessary for doing the learning. Additionally, the robot must be able to differentiate between those language utterances which specify specific behaviors that should be executed by the robot, and those utterances which refer specifically to the structure of the task to execute. These structures of our architecture are described in the following sections.

3.1 Behaviors

The fundamental building block of the robot's control system is the behavior. Simply speaking, behaviors are functions which map a set of inputs, including sensor information as well as derived state information, to a set of actions. Internally, every behavior is defined as a finite state machine with an explicit start state and potentially multiple termination states, depending on whether the behavior was successful in achieving its goals. Individual behaviors are the fundamental building blocks for more complex tasks. Behaviors are responsible for relatively simple control operations such as tracking a person and navigating between waypoints. On termination, a behavior will report whether it was successful or whether it had failed. One such failure condition includes not being able to reach the goal in a timely fashion (timeout).

A list of relevant behaviors used by our robots is as follows:

- **Goto(x,y)/Goto(name)** Allows the robot to navigate safely from its current position to either a location specified in its global coordinates or else to a named location.

- **Say(s)/Tell(s,p)/Ask(s,p)** Generates speech from the robot's speech synthesizer. The first form simply

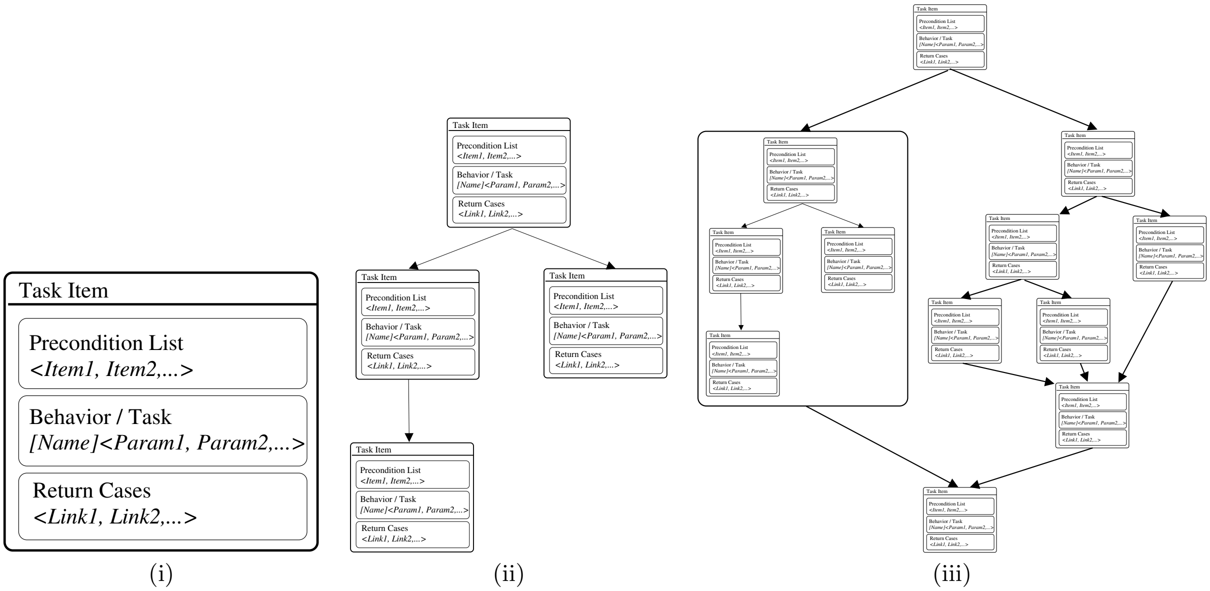


Figure 1: (i) The task item, the basic task building block. (ii) An example task consisting of four primitive task items. (iii) The completed task is invoked to create a more complex hierarchical task by linking it with simple task items.

causes the robot to speak an utterance **s**. The second requires that the robot identify and speak the utterance to a particular person **p** who is present. The third speaks the utterance to a specific person and waits for a response.

- **Follow(p)/ComeHere(p)** Causes the robot to locate person **p**, and drive towards them. The first form is a continuous following behavior while the second stops when the robot comes within a meter of the person.

In addition to the general behaviors for navigating and interacting with people, a set of specific task training behaviors has been defined:

- **LearnTask** This behavior causes the robot to listen to each utterance of the human and construct a task out of them.

- **FollowLearnTask** This behavior invokes the **Follow** behavior and drives the robot after the teacher and allows the robot to learn the task based on both the spoken utterances as well as the location of the teacher.

In order for the locations in the environment to be semantically meaningful as part of the training process, a map of the environment is provided to the robot which contains linguistic information regarding physical locations. For instance, the locations of named objects such as “couch”, “table”, and “television” can be added to the map as well as general locations of rooms such as “lab” or “living room.” This *a priori* information is used to ground locations that are either mentioned in the human's speech or are visited as the human walks about the environment.

3.2 Tasks

A task is defined as a group of behaviors connected in a graph structure, where each link in the graph represents a transition between one behavior and the next. Individual task items are wrappers for generic behaviors that allow the behaviors to be sequentially ordered. A task item, illustrated in Figure 1(i), includes three different components: (1) A precondition list that must be true before the task can execute; (2) the name of a behavior (or even another task) and the parameter list that will be passed in for execution; and (3) links to additional task items that will be executed based on the return status of the executing behavior. In general, a task item may have any number of links to additional task items. This fan-out is limited only by the expressiveness of the task and the language used to describe it. Typically, the outcomes of a particular task are either `Success` or `Failure` which is equivalent to an `If` conditional branching statement in a declarative programming language (but is not limited to just two outcomes.)

A complete task consists of a set of task items that are linked in the form of a directed acyclic graph (DAG), as shown in Figure 1(ii). The links that connect individual task items are directional and thus specify a temporal relationship between the two tasks where the first task in this temporal relationship must execute before the second is able to do so. The “root” node of the DAG is the initial starting state. There can be an arbitrary number of potential end states, which are specified as the leaf nodes with no additional nodes connected to them afterward.

Finally, tasks are named entities that can be referred to in the Behavior/Task slot of a task item. This nesting of tasks within task items can be arbitrarily deep and allows for the generation of complex tasks that reuse previously-defined tasks. An example of a hierarchical task structure is shown in 1(iii).

3.3 Training through Speech and Interaction

When executing the **LearnTask** behavior, the robot listens to the speech uttered by the teacher and uses it to build the network of task items. The **FollowLearnTask** behavior is a proper superset of the **LearnTask** behavior which is capable of building task structures from speech utterances but also uses the presence of the human and the robot's own location in the environment to learn additional behavior structures.

3.3.1 Learning from Language

The **LearnTask** behavior is invoked when the human says to the robot: `‘‘When I say <TaskLabel>’’` where `TaskLabel` is the name that will be assigned to this task. Each utterance spoken afterwards will be taken as either a specific statement of action or as a meta command which will either open a conditional link or will close one. For example, consider the statement:

```
‘‘If’’ <test-condition>
   <actionlist>
‘‘Otherwise’’
   <actionlist>
‘‘Before’’
<actionlist>
```

The `test-condition` is a boolean expression which checks the state of the environment according to the robot's sensors and internal state variables. The items called `actionlist` are (potentially empty) lists of actions that are spoken by the human. The quoted text consist of the explicit keywords that the robot will recognize as meta-commands. The `Before` keyword denotes the end of an `If` statement. Both the `Otherwise` and `Before` keywords are not necessary if their corresponding lists of actions are empty.

Note that there is no limit to the number of tasks that can be spoken in the `If` or `Otherwise` blocks. Nested `If` statements are also supported, as shown in the example below and in Figure 2.

```
>> When I say deliver message
>> If Person1 is present
>>   Give message to Person1
>> Otherwise
>>   If Person2 is present
>>     Give message to Person2
>>   Otherwise
>>     Report message delivery failure
>>   Before
>> Before
>> Goto home
```

Note the need for the two `Before` statements. Each one is necessary so that the robot knows how to attach the next task item in the graph. If the overall task becomes very complex with multiple levels of nested `If` statements, this can rapidly become difficult for the human to describe. Section 3.3.3 describes a mechanism for avoiding this by iteratively constructing the task.

Every **Goto** command that is learned for a task – whether explicitly commanded or inferred as is the case with the behavior **FollowLearnTask** – imposes a locational precondition (that can be overridden by other **Goto** commands) on subsequent behaviors and tasks within that task. The assumption is that the actions within a task can and should only be performed when the robot is where it was when it learned them.

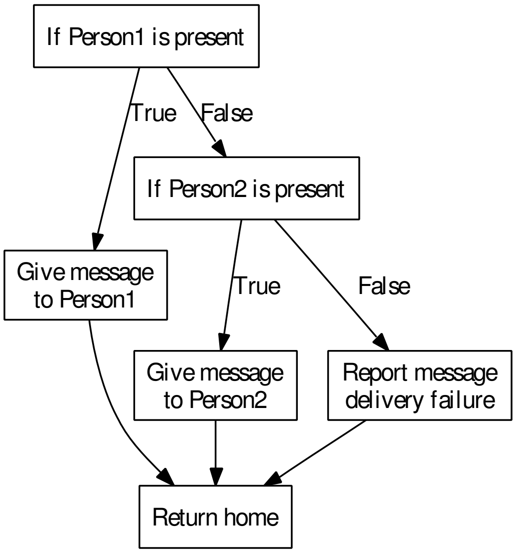


Figure 2: Example task list sequence illustrating nested `If` links.

3.3.2 Learning by Demonstration

The **FollowLearnTask** is invoked when the human says to the robot: **"Here is what to do when I say <TaskLabel>"** As stated previously, the **FollowLearnTask** has the capabilities of the **LearnTask** behavior for generating tasks from purely linguistic models. However, this behavior adds the capability to follow the human and generate task items from non-linguistic modes of communication. When invoked, this behavior will drive the robot up to the teacher and wait until the teacher says something or moves to a new location. Changes in location are compared against the robot's internal map of the environment. If the robot is near a labeled object in its map, its position will be registered as linguistic components such as "by the chair" or "in the kitchen", otherwise the position will register as an explicit x, y location.

3.3.3 Iterative Task Learning through Dialog

Once the human has taught the robot how to do the task, several verification steps are taken. First, the human can ask whether the robot understands the task after it has completed. The robot will verbally describe each task in the form that the human has described it. Additionally, because it is challenging for a person to describe an arbitrarily complex task with many conditional statements, particularly when those conditional statements are deeply nested, the human has the option of only specifying the true conditional part of any particular conditional statement. That is, only a single traversal through the tree is initially needed. When the robot is asked to verify the task, it will traverse the tree and look for any conditional statements that do not have an otherwise condition associated with them. In this case, the robot will notify the human and inquire whether this was expected. If the human intended more tasks to be added at this point, then task training is resumed and the

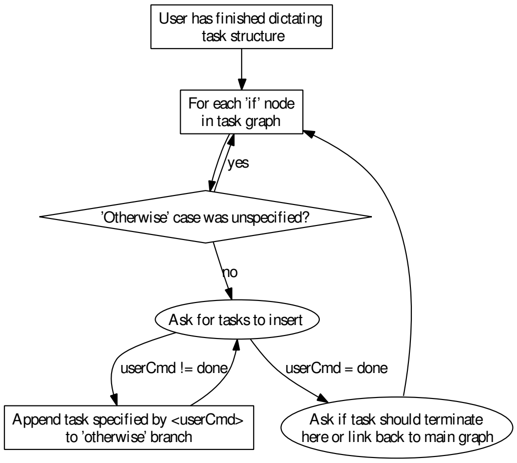

User has finished dictating task structure
For each 'if' node in task graph
yes
'Otherwise' case was unspecified?
no
Ask for tasks to insert
userCmd != done
userCmd = done
Append task specified by <userCmd> to 'otherwise' branch
Ask if task should terminate here or link back to main graph

Figure 3: Verification dialog that checks for unspecified 'otherwise' cases.

human can either speak a new set of commands, or teach the robot with a set of commands and actions. This process is repeated for every `If`-node in the behavior network that does not have an explicit `Otherwise` case. It allows the human to only have to demonstrate a single instance of a task to the robot, leaving the robot to keep track of when to ask for more details when necessary. The algorithm for this iterative process is shown in Figure 3 and an example is described in Section 5.

4. ROBOTS: CMASSIST

Our team of robots, called CMAssist (shown in Figure 4), are based on the ER1 mobile robot platform from Evolution Robotics. Mobility is provided by a custom set of DC motors. The robots have a CAMEO [15] omnidirectional camera rig mounted on the top of their sensor mast. People are identified and tracked by color histograms [2]. The top camera is also used for landmark identification by visually recognizing objects with clusters of SIFT [9] features. A particle filter localization algorithm [3] is used to maintain an estimate of the robot's pose through its environment. The robots use a stereo camera for obstacle avoidance as well as to assist with tracking people. Computational power is provided by two Pentium-M laptops.

Understanding of human speech is done in two parts. First, IBM ViaVoice is used for the initial capture and processing of the spoken utterances. A natural language processing system called NAUTILUS [13], developed by the Naval Research Labs (NRL), is used to process the utterances and match them against an *a priori* grammar that represents what the robot can understand. These programs are run on a third computer that the robots connect to wirelessly.

5. EXPERIMENTAL VALIDATION

To validate our task-learning architecture, we trained the robot to give a tour of our lab. We labeled four circular regions[1] in the robot's map of our lab: Lab center, Door, CMAssist Robots, and Robot dogs. The tour consists of greeting a visitor at the door and leading the visitor to where the robots are located, where it asks a lab member to talk about them.

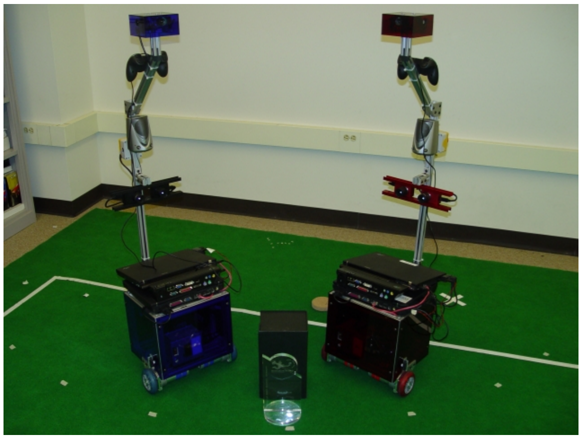


Figure 4: The robots of CMAssist 2006: Erwin and Carmela.

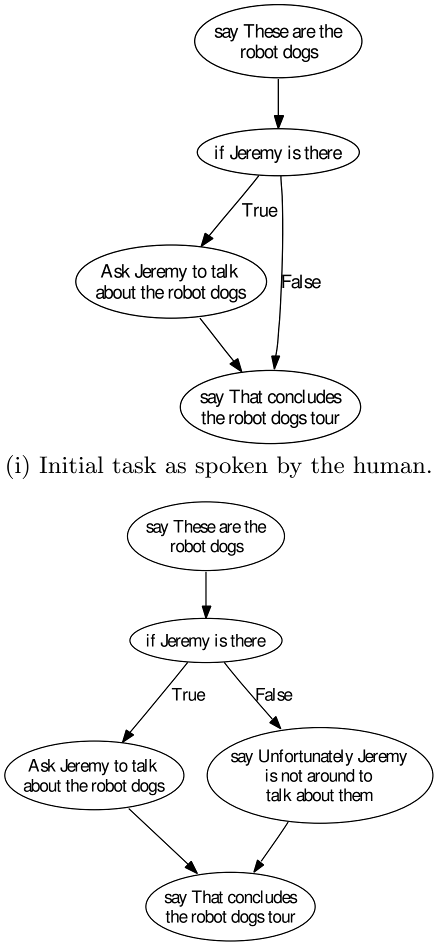
We broke down the task into the sub-tasks *Give the CMAssist tour* and *Give the robot dogs tour* which were trained using the **LearnTask** mechanism. Next, we trained the top-level task *Give the lab tour* that invokes the two learned tasks by using the **FollowLearnTask** mechanism. Finally, we executed the *Give the lab tour* task twice: once under conditions in which it would succeed and once under conditions in which the robot was rendered unable to successfully complete certain parts of the task. In so doing, we demonstrate the use of precondition-propagation during training and precondition-checking at runtime to enable the robot to gracefully carry on with the rest of a task when it is unable to complete a part of it.

5.1 Interactive Task Training

The following training dialog was used to train the *Give the robot dogs tour* task and demonstrates the system's ability to verify branches of the task tree that were not explicitly stated by the trainer (Section 3.3.3). The lines starting with "$>>$" indicate what the human says. The lines starting "**R**$>>$" indicate what the robot says.

```
   >> When I say give the robot dogs tour
   >> say These are the robot dogs
   >> if Jeremy is there
   >> Ask Jeremy to talk about the robot dogs
   >> before
   >> Say That concludes the robot dogs tour
(1)>> Is that understood?
  R>> Yes, you said if Jeremy is there: ask
      Jeremy to talk about the robot dogs.
      before say that concludes the robot dogs
      tour.  Is this correct?
   >> yes
  R>> Now checking unspecified 'otherwise'
      cases.  If Jeremy is there is FALSE,
      should I say that concludes the robot dogs
      tour?
   >> no
```

[1] Note that although the regions we defined in this experiment were circular, they could have been of arbitrary shape.


say These are the robot dogs
if Jeremy is there
True
False
Ask Jeremy to talk about the robot dogs
say That concludes the robot dogs tour
say These are the robot dogs
if Jeremy is there
True
False
Ask Jeremy to talk about the robot dogs
say Unfortunately Jeremy is not around to talk about them
say That concludes the robot dogs tour

(i) Initial task as spoken by the human.

(ii) After querying the human on the implicit connection of the otherwise case.

Figure 5: The process of iteratively constructing a task through dialog between the human and the robot.

```
  R>> What shall I do if Jeremy is there is FALSE?
      Say 'done' to end.
(2)>> Say Unfortunately Jeremy is not around to
      talk about them
   >> done
  R>> After I say Unfortunately Jeremy is not
      around to talk about them shall I say That
      concludes the robot dogs tour?  Say yes to
      confirm or say no to exit the task at this
      point.
   >> yes
  R>> Task training is complete.  Thank you.
```

Figure 5(i) shows the generated task at (1). The *FALSE* condition still links to the task specified after `Before`. The robot, however, reminded the user that an `Otherwise` case was not given and prompted him/her to insert tasks during this time. After (2) was inserted, the user had the option of making the task exit or link back into the task sequence and chose the latter. Figure 5(ii) shows the final task structure. The *Give the CMAssist tour* task was trained in a similar manner.

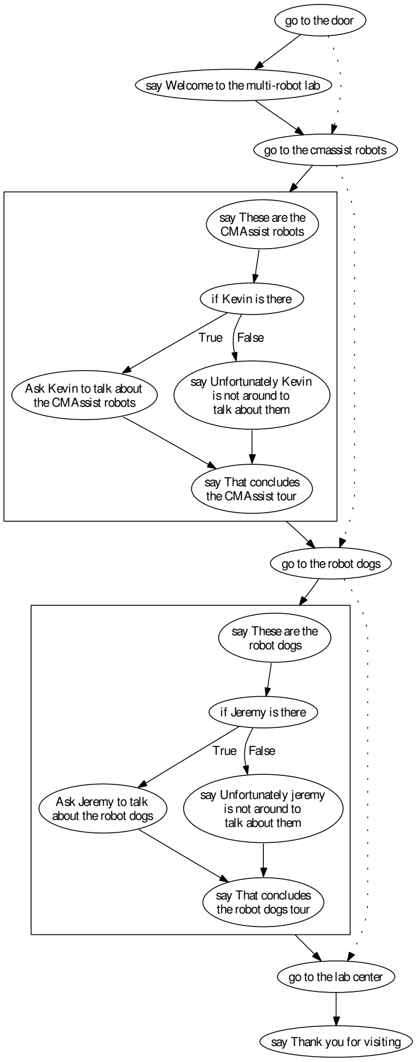
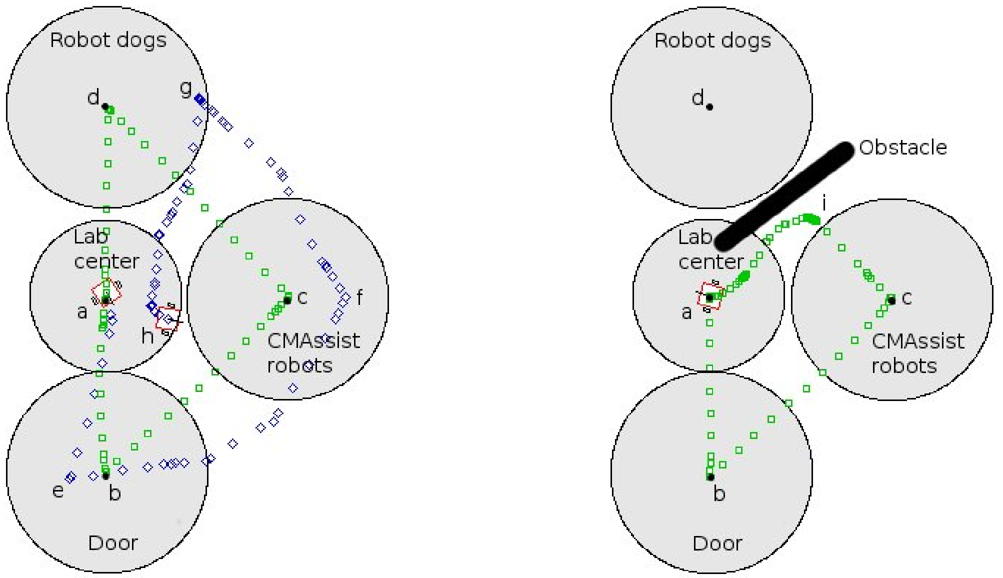

go to the door
say Welcome to the multi-robot lab
go to the cmassist robots
say These are the CMAssist robots
if Kevin is there
True
False
Ask Kevin to talk about the CMAssist robots
say Unfortunately Kevin is not around to talk about them
say That concludes the CMAssist tour
go to the robot dogs
say These are the robot dogs
if Jeremy is there
True
False
Ask Jeremy to talk about the robot dogs
say Unfortunately jeremy is not around to talk about them
say That concludes the robot dogs tour
go to the lab center
say Thank you for visiting

Figure 6: The hierarchical task of giving the lab tour with pre-defined tasks expanded to show their individual task. The dotted arrows show the implicit preconditions enforced by the GoTo behaviors.

Finally, *Give the lab tour* was trained using **FollowLearnTask**. The resulting task structure is shown in Figure 6. The training sequence is shown below with the actions of the human teacher denoted in brackets ([ ]). The letters beside the teacher's actions correspond to the marked locations in Figure 7(i), which shows the path of the robot during training.

```
(a)>> When I say give the lab tour
   [Walk to the door]
(e)>> say Welcome to the Multi-robot lab
   [Walk to the CMAssist robots]
(f)>> give the CMAssist tour
   [Walk to the robot dogs]
(g)>> give the robot dogs tour
   [Walk to the center of the lab]
(h)>> say Thank you for visiting
   >> Thank you
```



(i) Robot paths while following a human teacher during training (◇) and while executing the tour task (□).

(ii) The path of the robot when an obstacle prevented it from reaching the robot dogs.

Figure 7: A top-down view of the process of training and executing the *Give the lab tour* task. The grey circles denote pre-labeled regions where sub-tasks are learned and executed in this top-down view of the lab. An example of failure recovery during execution is shown on the right.

5.2 Robust Task Execution

For the first execution trial, the scene was set such that the *Give the lab tour* task would execute successfully. Kevin was near the CMAssist robots, Jeremy was near the robot dogs, and the paths between locations were not obstructed. The robot successfully carried out the task as shown in the following execution transcript. As before, the letters correspond to the marked locations in Figure 7(i) (Robot actions are shown in <>.)

```
(a)  >> Give the lab tour
(b) R>> Welcome...
(c) R>> These are the CMAssist robots. Kevin, could
        you talk about the CMAssist robots?
    <Wait for signal of completion from Kevin>
    R>> That concludes the CMAssist tour
(d) R>> These are the robot dogs. Jeremy, could you
        talk about the robot dogs?
    <Wait for signal of completion from Jeremy>
    R>> That concludes the robot dogs tour
(a) R>> Thank you...
```

In the second execution trial (Figure 7(ii)), we modified the environment so that the task would not successfully complete. This time, Kevin was not present in the room and an obstacle obstructed the path between the CMAssist robots and the robot dogs. The execution transcript follows:

```
(a)  >> Give the lab tour
(b) R>> Welcome...
(c) R>> These are the CMAssist robots.
    R>> Unfortunately Kevin is not around to talk
        about them
    R>> This concludes the CMAssist tour
(i) <Robot is blocked by obstacle. Attempt to reach
     robot dogs fails due to timeout.>
(a) R>> Thank you...
```

The robot greeted the guest at the door (b) and traveled to the CMAssist robots where it then initiated the *Give the CMAssist tour* task (c). However, Kevin was not present and the task graph was traversed appropriately (by stating that Kevin was not available). Then, blocked by an obstacle at (i), the robot could not reach the robot dogs location, resulting in the **Goto** behavior eventually timing out. The next item in the task was to *Give the robot dogs tour* but because this task was preconditioned on the robot's being in the robot dogs region, it was skipped entirely. The robot then returned to position (a) where it ended the tour normally.

6. SUMMARY AND FUTURE WORK

In this work, we have described an algorithm for training robots which involves the combination of spoken language understanding, dialog, and physical demonstration. Our approach allows a human to interact with a robot in a more natural and familiar fashion in order to train it on a new task. Our task representation allows tasks to be constructed from smaller subtasks in order to create complex hierarchical structures. Once an initial task is demonstrated to the robot, the robot can verify the task with the human and permit the human to add additional conditional cases, thus increasing robustness. Finally, the execution of the task itself is made robust to failure by checking for necessary preconditions in the environment before executing the associated task item as well as by timing out on a task if the robot is not making progress. Disambiguation of commands and words is handled by the NLP module. If a spoken utterance does not fit the known grammar, vocabulary, or even the response expected by the behavior, the robot notifies the user that it does not understand and requests that the command be repeated.

The system described in this paper has successfully demonstrated the feasibility of our proposed task training algorithms. We are encouraged by these results and are actively exploring mechanisms for improving the expressiveness and robustness of the algorithms. One such area for improvement stems from the fact that CMAssist's learned behavior structure is represented as a DAG. One shortcoming with this representation is that learned plans cannot have cycles, such as repeating a sequence of operations 'n' times. In future work, we plan on evaluating alternate plan representations such as domain-specific planners with loops [21, 22] or grounded situational models [10]. Another area of active improvement includes addressing the robot's task learning capabilities. These are only as expressive as its ability to understand the physical world around it. To address this, we are actively working to improve the robot's perceptual modules to include recognition of additional objects as well as the ability to recognize a much larger set of activities performed by people. Another area that we are exploring is how to increase the functional capabilities of our robot. Because our robots do not have manipulators, the kinds of tasks that the robots can learn are solely based on navigation to different locations. CMAssist can then mainly handle tasks which involve traveling to a specific location and querying/interacting with people found there. Tasks that are more complex, such as those which might involve manipulation, can be addressed through additional robotic hardware (we are currently examining how to add manipulators to our robots) as well as through the addition of more complex behavior modules which are designed to perform such tasks. The CMAssist perceptual module would need to be enhanced to handle recognition of human manipulation actions as well as visual recognition of the objects that are to be manipulated.

7. ACKNOWLEDGMENTS

We would like to thank the Naval Research Labs for developing the NAUTILUS natural language processing system and for helping us understand how to use it.

This research was supported by the National Business Center (NBC) of the Department of the Interior (DOI) under a subcontract from SRI International. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, by the NBC, DOI, SRI or the US Government.

8. REFERENCES

[1] D. Bentivegna, C. Atkeson, and G. Cheng. Learning from observation and practice at the action generation level. In *IEEE International Conference on Humanoid Robots*, Karlsruhe and Munich, Germany, September/October 2003.

[2] J. Bruce and M. Veloso. Fast and accurate vision-based pattern detection and identification. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taiwan, May 2003, to appear.

[3] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pages 1322–1328, 1999.

[4] M. Ehrenmass, R. Zöllner, O. Rogalla, S. Vacek, and R. Dillmann. Observation in programming by demonstration: Training and execution environment. In *IEEE International Conference on Humanoid Robots*, Karlsruhe and Munich, Germany, September/October 2003.

[5] H. Friedrich and R. Dillmann. Robot programming based on a single demonstration and user intentions. In *3rd European Workshop on Learning Robots at ECML'95.*, 1995.

[6] G. A. Kaminka, Y. Elmaliach, I. Frenkel, R. Glick, M. Kalech, and T. Shpigelman. Towards a comprehensive framework for teamwork in behavior-based robots. In *Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS-8)*, 2004.

[7] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *Transactions on Robotics and Automation*, 10:799–822, 1994.

[8] S. Lauria, G. Bugmann, T. Kyriacou, and E. Klein. Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38(3–4):171–181, 2002.

[9] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1150–7, 1999.

[10] N. Mavridis and D. Roy. Grounded situation models for robots: Where words and percepts meet. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[11] M. Nicolescu and M. Matarić. Linking perception and action in a control architecture for human-robot domains. In *Proceedings of the Thirty-Sixth Hawaii International Conference on System Sciences (HICSS-36)*, Hawaii, USA, January 2003.

[12] M. Nicolescu and M. Matarić. Natural methods for robot task learning: Instructive demonstration, generalization and practice. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, July 2003.

[13] D. Perzanowski, A. Schultz, W. Adams, E. Marsh, and M. Bugajska. Building a multimodal human-robot interface. *IEEE Intelligent Systems*, 16(1):16–21, January/February 2001.

[14] D. B. Roe and J. G. Wilpon. *Voice Communication Between Humans and Machines.* National Academy Press, Washington, DC, 1994.

[15] P. E. Rybski, F. de la Torre, R. Patil, C. Vallespi, M. M. Veloso, and B. Browning. Cameo: The camera assisted meeting event observer. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, April 2004.

[16] P. E. Rybski and R. M. Voyles. High-level task training of a mobile robot through human gesture recognition and imitation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 664 – 669, May 1999.

[17] J. Saunders, C. L. Nehaniv, and K. Dautenhahn. Teaching robots by moulding behavior and scaffolding the environment. In *Human-Robot Interaction*, Salt Lake City, Utah, March 2006.

[18] D. Sofge, J. G. Trafton, N. Cassimatis, D. Perzanowski, M. Bugajska, W. Adams, and A. C. Schultz. Human-robot collaboration and cognition with an autonomous mobile robot. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, and B. Kröse, editors, *In Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS-8)*, pages 80–87. IOS Press, March 2004.

[19] R. M. Voyles, J. D. Morrow, and P. K. Khosla. Towards gesture-based programming: Shape from motion primoridal learning of sensorimotor primitives. *Robotics and Autonomous Systems*, 22:361–375, November 1997.

[20] Y. Wang, M. Huber, V. Papudesi, and D. Cook. User-guided reinforcement learning of robot assistive tasks for an intelligent environment. In *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, 2003. IEEE.

[21] E. Winner and M. Veloso. Analyzing plans with conditional effects. In *Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems*, Toulouse, France, April 2002.

[22] E. Winner and M. Veloso. DISTILL: Towards learning domain-specific planners by example. In *Proceedings of ICML'03*, Washington, DC, August 2003.