

Challenges to Decoding the Intention Behind Natural Instruction

Raquel Torres Peralta, Tasneem Kaochar, Ian R. Fasel, Clayton T. Morrison, Thomas J. Walsh, Paul R. Cohen

Abstract— Currently, most systems for human-robot teaching allow only one mode of teacher-student interaction (e.g., teaching by demonstration or feedback), and teaching episodes have to be carefully set-up by an expert. To understand how we might integrate multiple, interleaved forms of human instruction into a robot learner, we performed a behavioral study in which 44 untrained humans were allowed to freely mix interaction modes to teach a simulated robot (secretly controlled by a human) a complex task. Analysis of transcripts showed that human teachers often give instructions that are nontrivial to interpret and not easily translated into a form useable by machine learning algorithms. In particular, humans often use implicit instructions, fail to clearly indicate the boundaries of procedures, and tightly interleave testing, feedback, and new instruction. In this paper, we detail these teaching patterns and discuss the challenges they pose to automatic teaching interpretation as well as the machine-learning algorithms that must ultimately process these instructions. We highlight the challenges by demonstrating the difficulties of an initial automatic teacher interpretation system.

I. INTRODUCTION

Human-instructable computing seeks to develop intelligent devices that can be taught by natural human instruction. By “natural,” we mean those patterns of communication that humans use every day while teaching each other. For example, while teaching another person, humans may provide (1) explicit *definitions* and *examples* of concepts, rules and conditions, (2) *describe* and provide *demonstrations* of procedures, and (3) provide various kinds of *feedback* based on student behavior. We refer to these patterns of instruction as *natural instruction methods*.

Translating these forms of instruction into inputs for machine learning algorithms (e.g., example-label pairs, execution traces, rewards for reinforcement learning) is nontrivial because human teaching tends to be opportunistically presented and therefore not well-structured for a direct conversion into a machine-understandable format. Human teachers freely mix modes of interaction without explicitly stating that they are doing so, and often instruction is implicit. For instance, demonstrating a procedure that involves an object may simultaneously illustrate a trace for the procedure *and* provide an example of the object of interest. Thus, most prior work on human-robot teaching has avoided the complications of learning from interleaved instruction methods by focusing on a single form of human instruction at a time.

We have built an exploratory teaching interface that can accommodate different types of human teachers, regardless of whether they use a single mode or multiple modes of natural

instruction. Thus, our teaching interface particularly targets *non-expert* teachers, that is, teachers without prior knowledge about how to structure their teaching for a machine. We conducted behavioral studies to observe *how* non-expert users teach using our interface, that is, we take as “natural” what the participants actually do, and part of our goal is to identify the interface use patterns they produce.

Our primary focus is on understanding what it would take to automatically map interleaved natural human instructions (such as the teacher instruction transcripts generated from our behavioral studies) to state-of-the-art machine learning algorithms that expect human input, such as concept [1], procedure [2], or reinforcement [3] learners, each of which may be a component of a full end-to-end human-instructable machine. To do so, we performed a behavioral experiment similar to that of [4], in which human teachers provide instructions to a simulated robot student that is secretly controlled by a human. Teachers interacted with the student through a GUI that provides the ability to freely mix natural instruction methods. A careful analysis of the transcripts reveals that while many of these teaching patterns are straightforward to map into machine learning targets, humans frequently use a number of other teaching patterns that are difficult to interpret; we characterize these as *implicit* teaching methods. We also found that we could group teachers according to three different *organizational styles* based on the frequency and manner of interleaving teaching methods. We conclude by illustrating the challenges involved in interpreting teacher instructions and varying teaching styles by describing an initial automatic teacher instruction recognition system.

II. PRIOR WORK

Many prior efforts have described robots that learn from human teachers [5], [6]. However, most of these works used only one mode of teacher-student interaction (e.g. teaching by demonstration). Roughly speaking, the methods that have been explored include teaching by *demonstration*, teaching *concepts by example*, and teaching through *reinforcement*.

In teaching by demonstration, a teacher has the same control interface to a dynamical system as the student does, and is able to provide *traces* of proper behavior. Prior work in learning from demonstration includes using traces in a batch setting to bootstrap robot behaviors [7], [8] and in apprenticeship learning over the agent’s lifetime [9].

Teaching concepts by example follows a standard supervised-learning protocol where the teacher is responsible for providing labeled examples. However, the teacher may

The University of Arizona, Department of Computer Science, Tucson, AZ 85721-0077
{rtorres,tkaochar,clayton,ianfasel,twalsh,cohen}@cs.arizona.edu

also provide explanations or hints as to *why* a specific example was classified a certain way. An example of a system that uses both inputs in an inductive logic programming framework is WILL [1]. Other systems for more traditional classification problems, such as training support vector machines [10] and email classification [11] have used this paradigm, with the latter focussing explicitly on natural ways that humans provide reasons for categorization decisions. Similar techniques can be helpful in learning conditional dynamics (e.g. you cannot walk through walls) [12].

In teaching through reinforcement, the teacher is able to give a feedback signal indicating a degree of happiness (or unhappiness) with the agent’s behavior, either at specific timesteps or when an episode has ended. This form of feedback has been shown to be moderately successful in complex tasks such as a simulated cooking scenario [13] and tactical battles in a real-time strategy game [14]. However, recent work [3], [15] indicates that many “natural” ways of incorporating numeric or ordinal feedback from humans into a reinforcement learning problem can be perilous, as humans often provide incompatible feedback or do not follow “standard” definitions of reward and value. As such, this channel is usually best suited for fine-grained refinements.

In addition to incorporating and analyzing the usage of each of these modes of interaction, we also consider the organizational style of the teacher (how structured their lessons are) in an attempt to provide better context for interpreting instructions. This form of analysis has not been widely studied in the literature, though other works [4] have noticed that many teachers often overlap lessons and commands.

III. EXPERIMENTS

Our goal is to better understand how non-expert human teachers combine natural instruction methods to teach a robot when the teaching interface provides multiple instruction modes that can be freely intermixed. We used an experiment protocol in which novice human participants taught a series of inter-dependent concepts and tasks to a simulated electronic student in a Wizard of Oz (WoZ) paradigm, specifically we asked 44 students from the University of Arizona to teach a task to an “electronic student” that was secretly controlled by a human who simulated robot learning. The teaching interface was designed so that we did not depend on parsing and interpreting natural language input but instead used standard GUI elements to enforce simplified input for potential use with machine learning algorithms.

Wizard of OZ Experimental Setup: Each teaching session consisted of a participant taking on the role of the *Teacher* while a researcher played the *Student*. The Teacher was led to believe that he/she was interacting with an electronic student. The Teacher was first trained (by a second *Experimenter*) on the use of the interface and then presented with a teaching task that outlined the knowledge the Student should attain by the end of the teaching session. The teaching sessions lasted from 25 to 35 minutes. The same two researchers took on

the role of the Student and the Experimenter, respectively, across all experiments.

The task took place in an Intelligence, Surveillance and Reconnaissance (ISR) domain in which the Student controls a simulated unmanned aerial vehicle (UAV) and is taught to carry out missions. The simulated environment includes a terrain map with objects that can be scanned using two sensors: a *high-resolution camera* (providing detailed object information, such as whether a boat has a cargo hold), and a *radiation sensor* (which detects the radiation level of an object in range). The Student can only perceive the world through the sensors and the Teacher must teach the Student how to use the UAV sensors in the appropriate circumstances.

Teacher Interface: Our interface provides the Teacher three tools to teach the Student: the Instruction Command Interface (ICI), which sends commands to the Student; a Timeline Display that shows a list of all prior Teacher instructions; and a Map Display that includes the UAV sensor state and range, and the UAV flight path (Fig. 1). Action commands (from the ICI) allow the Teacher to control the UAV with controls such as *track object*, *fly to location*, *descend*, etc. Four modes of instruction are supported:

- (1) *Teaching by demonstration:* The Teacher can label a sequence of actions as an example of a procedure. This can be done either by labeling the beginning and end of the procedure while providing a demonstration, or after the fact by selecting already executed actions from the Timeline Display. The Teacher can provide multiple examples of a single procedure and each instance can be labeled as a “good” or “bad” example. We refer to the ICI tools available for this functionality as the *procedure demonstration facility*.
- (2) *Teaching concepts by examples:* The Teacher can define concepts (such as “cargo boat”) by selecting an object on the map interface and giving it a label. Instances of labeling can be designated as a positive or negative example of correct label attribution. We refer to this ICI tool as the *object labeling facility*.
- (3) *Teaching by reinforcement:* The Teacher can give feedback to the Student, in the form of 1-3 “happy faces” or 1-3 “frowny faces”, issued at any time, and can also send a “You achieved goal <goal name>” message.
- (4) *Testing the Student:* The Teacher can use simple commands that ask the Student to provide a label for an object or execute a previously defined procedure. In this way, the Teacher can test whether the Student has learned the labels or procedures correctly.

Teaching Task: In each of the Teaching sessions, there were two kinds of objects: cargo boats and fishing boats. The Teacher was asked to teach the Student how to distinguish them, to use the radiation sensor only on cargo boats, and to generate a report of the readings. Although this task is very simple, it requires teaching multiple concepts and procedures that depend on one another.

Teacher-Student Interaction Data: During the teaching session, we captured the entire state of the system and recorded all changes over time, which were used to generate

a series of Teacher-Student transcripts for human analysis. Participants were also given a post-experiment questionnaire.

IV. ANALYSIS OF TEACHING PATTERNS

The goal of our exploratory experiments is to discover if human instructions with a flexible interface such as the ICI could be easily translated into inputs for machine learning systems, and if not, identify the kinds of teaching patterns a system would need to accommodate. We found that there are indeed several teaching patterns that may be challenging for automated systems, but which we found were quite common and natural for untrained human teachers using our system.

A. Multiple uses of the procedure demonstration facility

As described above, use of the object labeling facility involves simply clicking on an object in the map and entering a text label, while procedure labeling is a more complex operation involving selecting and labeling a sequence of actions. In our post-study questionnaire, 41% of teachers identified the object labeling facility as “easy to use”, while only 16% identified the procedure demonstration facility this way. Twenty-three percent reported procedure definitions as difficult to teach. Despite the perceived difficulty, a total of 80% of the teachers used the procedure demonstration facility at some point. However an estimated 41% used the procedure demonstration facility in ways that were not originally intended in our design. In particular:

1) *Implicit Object Labeling*: Of the 41% of teachers who used the procedure demonstration facility in an unexpected way, 25% appear to have used it as a means to label objects. These teachers maneuvered the UAV up to an object and then appeared to use the name of the object as the name of a procedure. We called this general pattern *implicit labeling*.

To illustrate the challenge of identifying implicit labeling, we first show an example of *explicit labeling*. In the following transcript excerpt¹, the Teacher selects a boat from the map interface and then enters the object label:

```
12: 12:32 T: Object @lat. = 39.04, long. = -122.89
is a good example of object label 'Cargo Boat'
(Object name = Boat11)
```

This command is unambiguous and directly interpretable as labeling the object.

In contrast, in implicit labeling the teacher first provides the label as the name of a new procedure, starts the procedure, positions the UAV near an object, and then instructs the student to track the object. This is demonstrated in the following transcript excerpt; here the teacher introduces the concepts ‘Cargo Boat’ and ‘Fishing Boat’:

```
39: 43:29 T: Start good example of procedure 'Cargo Boat'
40: 43:38 T: Fly plane to object/location @ lat= 39.10,
lon=-122.82 (Object name = Boat10)
41: 44:13 T: Use camera to track object/location @ lat
39.10, lon -122.82 (Object name = Boat10)
43: 44:52 T: End example of procedure 'Cargo Boat'
(...change in location...)
53: 48:23 T: Start good example of procedure 'Fishing Boat'
```

¹Transcript excerpts are formatted as follows: Line number, timestamp, source (T indicates Teacher’s command and SR indicates Student’s response) and content.

```
54: 48:30 T: Fly plane to object/location @ lat= 39.09,
lon=-122.86 (Object name = Boat12)
55: 49:08 T: Use camera to track object/location @ lat.
39.04, lon. -122.80 (Object name = Boat12)
58: 51:58 T: End example of procedure 'Fishing Boat'
```

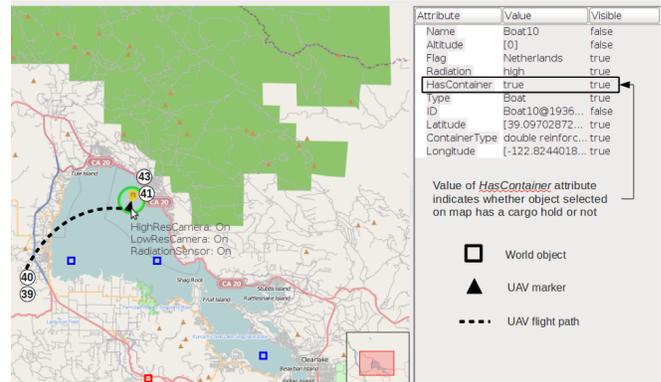


Fig. 1. Map Display: A display of the progression of teacher commands (lines 39-43) defining a good example of procedure *Cargo Boat* from the earlier transcript excerpt.

A direct interpretation of these actions as defining a procedure appears to miss that the teacher is labeling the boats. There are two main sources of evidence for this: (1) the teacher gave a procedure name that only consisted of a noun phrase (e.g., ‘Cargo Boat’) rather than a verb phrase (as was typical of other procedure definitions); and (2) in each of these cases, the teacher instructed the student to move the UAV very close to the object (e.g., directly above the boat) and also asked them to track the object (Figure 1). This situation presents a challenge for an automated interface to infer that the teacher is indeed labeling.

2) *Implicit procedure definition*: In addition to the unexpected use of procedure demonstration for implicit object labeling, we also observed many cases (for all but 2 teachers) in which procedures appeared to be taught but the procedure facility was not used. In these cases, we observed teachers directing the Student to perform sequences of steps repeatedly, such as using the sensors to gather information and then generating a report. It appears that the teachers assumed the Student was “paying attention” to these sequences and would be able to infer the boundaries between the implicit procedure definitions.

3) *Ill-defined procedure boundaries*: We also found that human teachers frequently gave procedure definitions with imprecise boundaries. For example, in the following transcript excerpt the teacher provides action commands *fly to*, *camera on* and *camera track* on lines 50, 63, and 66 respectively, well outside the explicitly defined boundaries of procedure *Fly to cargo boat*, although these appear to be key parts of the intended procedure:

```
50: 29:542 T: Fly plane to lat = 39.04, lon = -122.80
(...UAV heading towards destination...)
63: 29:603 T: Turn on high resolution camera
66: 29:609 T: Use camera to track object @ lat = 39.04,
lon = -122.80 (Object name = Boat12)
(...UAV nearing destination...)
72: 29:807 T: Start bad example of proc. 'Fly to cargo boat'
73: 29:815 T: Unpause the plane
77: 29:821 T: Pause the plane
```

(...UAV reached destination...)
 79: 29:831 T: End example of procedure 'Fly to cargo boat'

At a first glance, the Teacher’s instructions in lines 73 and 77 appear to be meaningless. However, the name of the procedure (“Fly to cargo boat”), in addition to the command at line 50, helps to clarify the situation: the UAV is directed to fly toward a “fishing boat” and sensors are turned on. The Teacher then lets the UAV continue until it is just above the object, at which point the Teacher pauses the plane to teach that the object directly below is *not* a cargo boat (the procedure is a bad example, indicated in line 72).

This example illustrates that we cannot always rely on the human to provide correct lesson boundaries to the Student. In the example above, for instance, the Teacher omitted several commands, including “Fly to”, from the procedure definition. Moreover, it may be that the procedure demonstration facility in this example is being used for an *implicit object labeling*.

Overall, these varied uses of the procedure demonstration facility reveal that the mapping from natural human teaching to machine learning inputs requires some sophistication. An automated mapping facility would need to identify likely procedure boundaries, even in cases where the teacher has provided some boundaries, and then deliver them to the appropriate learning mechanism.

B. Frequent use of testing followed by feedback

An important facet of teacher-student interaction is the testing of student performance. The majority of the teachers (87%) made use of the ICI testing tools at some point during their teaching sessions. We also found there were important differences between teaching and testing over time. To see this, we split each teaching session into 3 equal time phases and analyzed how the usage of teaching and testing tools differed (Figure 2). We found that teaching tools were mostly used in the first phase (82%, 11%, 7%) while the use of the testing tool was more evenly distributed (41%, 32%, 14%) across teachers.

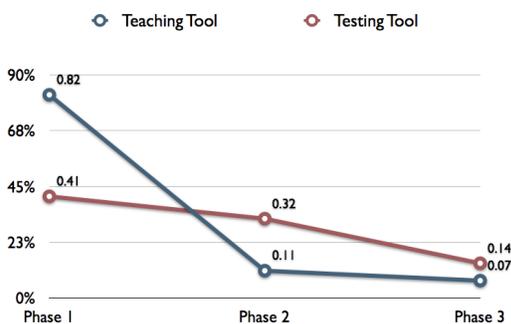


Fig. 2. Percentage of Teachers that used the Teaching and Testing tools at least once in each phase

Another class of patterns we observed pertains to the frequency of testing paired with teaching. Half of our teachers preferred to test the Student right after each lesson, probably to ensure that each concept has been learned before providing a new one. Eleven students preferred to loosely interleave

teaching and testing, introducing several new concepts to the Student before going back to test any of them. Finally, 6 of the teachers decided to test the Student only at the end of the teaching session, while 5 others never tested the student.

Testing and Feedback. One of the most frequent patterns we observed was the ‘teach-test-feedback’ loop. The following is an example transcript excerpt of this pattern:

```
35: 24 T: Start good example of procedure 'cargo boat'
...
159: 31:38 T: End example of procedure 'cargo boat'
160: 31:48 T: Perform procedure 'cargo boat' near
        lat. 39.10, long. -122.82
...
164: 32:48 SR:Radiation sensor reading: high
165: 32:56 T: You achieved goal 'find cargo boat'
166: 32:59 T: 1 happy face
```

Here, the Teacher immediately follows the demonstration of a new procedure concept (lines 35-159, details omitted for space) by prompting the Student to perform the newly-defined procedure in a new scenario location. Between lines 160-164, the Student flies to the desired location and starts performing the action specified by the Teacher as part of procedure cargo boat, including taking the radiation sensor reading of the boat object in the new location (line 164). Finally, the Teacher responds to the Student’s test performance by providing positive feedback (lines 165-166).

We found that most of the teachers who tested the student (85%) provided some feedback to either express their satisfaction/frustration, or used the goal labeling facility to indicate a goal has been reached. Those who used feedback did so consistently throughout the session. Fifty-eight percent of teachers used feedback exclusively after testing. The teachers who did not use feedback in conjunction with testing tended to leave their approval/disapproval implicit, instead either choosing to move on to the next step in the lesson, or, if the student made a mistake, providing a new example of the same concept. A robust automated interface should keep track of lesson topic advancement as possible implicit feedback about performance.

C. Frequent use of consistent naming conventions

It is not surprising, but still worth mentioning, that human teachers consistently used meaningful naming conventions when providing labels for object or procedures. Objects were generally given names derived from the vocabulary of the task domain, such as ‘cargo boat’ or ‘fishing boat’, and procedure names included verb phrases, such as ‘fly to boat’, ‘track boat’, or ‘scan boat’. Even very simple natural language processing may provide additional means for extracting teacher intent. For example, identifying verb phrases versus noun phrases could help identify when the procedure definition facility might be used for object labeling.

D. Different “types” of teachers

Finally, our transcript analysis also revealed three distinct styles of teaching based on the *organization* of lessons: (1) *Structured* teachers (16% of participants) were consistent and methodical in the execution of their instruction commands. They consistently used the interface’s object labeling facility

to teach object concepts and the procedure demonstration facility to define procedures (no implicit object labeling or implicit procedure definitions). These teachers always tested concepts from previous lessons and did not attempt to test Student’s comprehension of concepts not yet taught. Furthermore, all procedures defined by *structured* teachers were well-formed, (no missing commands). (2) *Semi-Structured* teachers (50%) began with a less structured teaching style but became progressively more structured as the teaching session continued. They made use of the GUI features almost as intended, sometimes with early exploration of usage. (3) *Free style* teachers (34%) were the most difficult to follow, mainly because these teachers made use of GUI features in novel ways, such as using procedures for implicit labeling and sometimes using testing tools before teaching, perhaps because of high expectations of the Student’s initial capabilities.

The distribution of styles shows that semi-structured and free-form teaching styles were far more common than the structured approach. This may be a function of teacher experience and training. However, to the extent that we want to accommodate non-expert users, an interface will need to handle potentially unstructured teaching.

V. AUTOMATICALLY DECODING TEACHER INTENTIONS

In the previous section, we identified a variety of patterns which are likely to present considerable difficulty to a system in “decoding” a teacher’s intended meaning for an instruction. In this section, we describe a baseline system that attempts to convert as much of a transcript as possible into segmented teacher intentions. This includes identifying which instructions appear to be about “setting up a lesson” (e.g., maneuvering the UAV to a particular location) as opposed to actual object or procedure lessons, and when a procedure is really an *implicit definition*. This initial approach is intentionally simple, in order to draw attention to areas where simple techniques are likely to fail.

A. Human Transcript Annotation

In the first phase of our approach we identified a set of labels that can be applied to each line of a transcript to indicate how the teacher intended each instruction to be used by a learning system. For instance, given the teacher command sequence:

```
Direct the UAV to a location
'Start procedure X'
Perform a few actions...
'End procedure X'
```

the lines before the procedure should be labeled as “Setting Up”, and the lines between the procedure boundaries should be labeled as belonging to a single procedure.

Two human annotators (two of the authors) labeled each of the 44 transcripts. The left column of Figure 3 contains the set of labels used. Because of the exploratory nature of this study, the labelers conferred and iterated multiple times over all the transcripts in order to refine the label interpretations and create a consistent set of labels. The labelers generally

agreed on most of the initial interpretations and only a few iterations were needed to reach consensus. Most transcripts were analyzed several times by both labelers. We decided a traditional “null hypothesis significance testing” paradigm was not appropriate at this stage of the research, as it would lead to an overly narrow analysis. We therefore did not perform a statistical analysis of inter-coder reliability.

B. Automatic Transcript Annotation

Once the transcripts were annotated, we constructed a simple script that attempted to use the heuristics the human annotators used while performing the annotation. The center column of Figure 3 provides a quick description of these heuristics. Processing is done in a single-pass, and some rules involve operations on multiple lines simultaneously. We did not use any sophisticated statistical techniques for inference in structured data.

All 44 transcripts were annotated using the script and the resulting labels were compared with the human labels. We found that for the most part, transcripts could be automatically annotated based on the small set of rules and heuristics. The right column of Figure 3 shows which commands could be automatically translated to agree with the human labels: instructions labeled with a checkmark were annotated with 100% accuracy. Instructions labeled with an 'X' or the check with a star were not labeled accurately: in these cases, the script could not handle implicit object and procedure definitions, and “Setting Up” commands were identified correctly only when the script correctly identify the teaching that followed.

| Label | Labeling Heuristic | Script Successful |
|-------------------------------|--|-------------------|
| Explicit Object Label | Teaching command defining good or bad object example using the <i>object labeling construct</i> | √ |
| Implicit Object Label | Teaching command defining good or bad object example using the <i>procedure construct</i> | × |
| Explicit Procedure Definition | Teaching command defining a procedure using the <i>procedure construct</i> | √ |
| Implicit Procedure Definition | Repetition of similar set of action commands (camera track, take radiation reading or picture, generate report) in different scenario locations | × |
| Test | Testing command (ask Student to perform a previously taught procedure or give the label of a world object) | √ |
| Feedback | Command evaluating Student's performance (1-3 <i>happy</i> or <i>frowny</i> faces) | √ |
| Goal Specification | Command labeling a goal that Student has achieved | √ |
| Setting Up for Teaching | All setup commands (change in location, speed, altitude or sensor settings) preceding a teaching command (object labeling or procedure definition) | √* |
| Setting Up for Testing | All setup commands preceding a testing command | √ |

Fig. 3. Automatic Transcript Annotation (* means only when teaching command has been correctly identified)

C. Teacher-Style Identification

The teacher type (*structured*, *semi-structured*, or *free-style*) determines the way participants used the system and hence, the overall characteristics of the transcript. Using the heuristics specified in Figure 3, the script was able

to correctly identify all of the teachers who followed the *structured* teaching style. However, classifying teachers into *semi-structured* vs. *freestyle* remains an open problem.

VI. DISCUSSION AND CONCLUSIONS

The teaching interface we built for this study is an exploratory system, designed to accommodate interleaving of multiple modes of instruction during teaching. Our goal was to use this system as a testbed to observe the types of teaching behavior that are elicited when given the opportunity to utilize multiple instruction modes. To the best of our knowledge, our system is one of the first of its kind but there is much room for improvement. While we have attempted to simplify the interface and reduce the number of teaching tools available, the complexity of the system can initially overwhelm a non-expert user. Although our teacher participants were given a 30 minute tutorial on the system usage, the complexity can be distracting and may have had an effect on the productivity and structure of teaching. One of the teaching tools, the procedure definition facility, was particularly difficult for our participants to use and elicited unexpected behavior such as implicit object labeling. However, we do believe that the interesting teaching behaviors we have observed cannot all be ascribed to interface flaws; rather they highlight issues that should be considered in building a successful human instructable electronic student.

Our analysis has shown that teachers often give instructions that require considerable interpretation and are not amenable to straightforward, direct identification. Humans make extensive use of implicit instructions, often fail to clearly indicate the beginnings and ends of lessons, appear to rely on meaningful linguistic names for objects and procedure labels, and frequently interleave testing and feedback, followed by new examples (and often do not give direct feedback after tests).

We found that a relatively simple set of rules can accommodate important aspects of human instruction but is only successful at completely tracking teacher intention if the teacher follows a structured teaching style. Such a style makes teacher intent tracking “easy” because there is (1) a near 1-1 mapping between teacher instruction and intention, (2) most information relevant to intent is in local time context (based on very recent Student actions) and (3) no environmental context disambiguation is necessary. This style of teaching, however, was the least common among our teacher participants; in most cases, one or more of the three criteria for easy interpretation were violated. A successful instructable robot will need to deal with the more subtle forms of teaching we found frequently in our transcripts.

A critical next step is to improve recognition of implicit instructions used in semi-structured and freestyle teaching. This in turn will allow us to better detect the different organizational styles of humans. More specifically, the au-

tomatic identification of implicit object labeling and implicit procedures as well as teacher-specified procedure boundaries needs to be improved and tested. These behaviors make the teacher intent tracking problem challenging and in order to address them, an instructable robot student system would need to include some nontrivial mechanisms to: (1) track environmental changes in the teaching session, (2) identify what context (local or global) is relevant to teacher’s instruction (restricted language comprehension capability may be necessary), and (3) handle interleaving teacher instructions. Not only will this help robots learn from natural instructions, it will also be useful for helping the robot actively guide the interaction through robot-human feedback.

VII. ACKNOWLEDGMENTS

This work was supported by DARPA under contract 27001328 and ONR “Science of Autonomy” contract N00014-09-1-0658.

REFERENCES

- [1] S. Natarajan, G. Kunapuli, R. Maclin, D. Page, C. O’Reilly, T. Walker, and J. Shavlik, “Learning from human teachers: Issues and challenges for ILP in bootstrap learning,” in *AAMAS Workshop on Agents Learning Interactively from Human Teachers*, 2010.
- [2] E. Winner and M. M. Veloso, “DISTILL: Towards learning domain-specific planners by example,” in *Proceedings of ICML’03*, 2003.
- [3] W. B. Knox and P. Stone, “Combining manual feedback with subsequent MDP reward signals for reinforcement learning,” in *AAMAS*, 2010.
- [4] E. Kim, D. Leyzberg, K. Tsui, and B. Scassellati, “How people talk when teaching a robot,” in *Human Robot Interaction*, 2009, pp. 23–30.
- [5] S. Schaal, A. Ijspeert, and A. Billard, “Computational approaches to motor learning by imitation,” *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1431, p. 537, 2003.
- [6] D. Matsui, T. Minato, K. MacDorman, and H. Ishiguro, “Generating natural motion in an android by mapping human motion,” in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3301–3308.
- [7] W. D. Smart and L. P. Kaelbling, “Effective reinforcement learning for mobile robots,” in *ICRA*, 2002.
- [8] B. Argall, S. Chernova, M. M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [9] T. J. Walsh, K. Subramanian, M. L. Littman, and C. Diuk, “Generalizing apprenticeship learning across hypothesis classes,” in *ICML*, 2010.
- [10] S. Chernova and M. Veloso, “Interactive policy learning through confidence-based autonomy,” *Journal of Artificial Intelligence Research*, vol. 34, no. 1, pp. 1–25, 2009.
- [11] S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. Dietterich, E. Sullivan, and J. Herlocker, “Interacting meaningfully with machine learning systems: Three experiments,” *International Journal of Human-Computer Studies*, vol. 67, no. 8, pp. 639–662, 2009.
- [12] A. L. Thomaz and M. Cakmak, “Learning about objects with human teachers,” in *HRI*, 2009.
- [13] A. L. Thomaz, G. Hoffman, and C. Breazeal, “Reinforcement learning with human teachers: Understanding how people want to teach robots,” in *ROMAN*, 2006.
- [14] K. Judah, S. Roy, A. Fern, and T. G. Dietterich, “Reinforcement learning via practice and critique advice,” in *AAAI*, 2010.
- [15] A. L. Thomaz and C. Breazeal, “Teachable robots: Understanding human teaching behavior to build more effective robot learners,” *Artificial Intelligence*, vol. 172, no. 6-7, pp. 716–737, 2008.