An Extensible Language Interface for Robot Manipulation

Jonathan Connell[1], Etienne Marcheret[1], Sharath Pankanti[1], Michiharu Kudoh[2], Risa Nishiyama[2]

[1]IBM T.J. Watson Research Center, 1101 Kitchawan Rd, Yorktown Heights NY 10598, USA
{jconnell, etiennem, sharat}@us.ibm.com
[2]IBM Research - Tokyo, 5-6-52, Toyosu, Koutou-ku, Tokyo, 135-8511 Japan
{kudo, lisa}@jp.ibm.com

Abstract. This paper describes our Extensible Language Interface (ELI) for robots. The system is intended to interpret far-field speech commands in order to perform fetch-and-carry tasks, potentially for use in an eldercare context. By "extensible" we mean that the robot is able to learn new nouns and verbs by simple interaction with its user. An associated video [1] illustrates the range of phenomena handled by our implemented real-time system.

Keywords. robot, language, learning, eldercare

1 Introduction

As argued in [2] with an eye toward Vygotsky, much of intelligence is actually illusory since the bulk of what we consider knowledge or competence is transmitted culturally. No one figures out how to cook macaroni and cheese by experimentation – some other person tells you how to do it. While part of the feeling of aliveness comes from the responsiveness of a creature with a reasonably deep perception of its environment, even humans from a different society can be successfully demonized as "sub-human" if you cannot understand what they say. If robots are ever to be perceived as sentient it seems crucial that they also be able to learn in this manner and thus partake of the rich prevailing culture which underpins much of "human-ness".

Language understanding and learning also has pragmatic value. For instance, a robot that could perform simple fetch-and-carry tasks would be a boon to eldercare. However the robot must be told what to do somehow. The current generation of senior citizens is not comfortable with tablets, keyboards, styli, PDAs, or Bluetooth headsets – these are just one more thing to drop or misplace. The most human-friendly interface is direct speech using an audio pickup on the robot itself. The trick then is interpreting the spoken commands robustly. In addition, a particular home may have locations, like the "solarium", or objects, like "my favorite cup", which cannot be known *a priori* and hence cannot be preprogrammed into the robot. Thus it would be convenient if the robot could just be shown such places and objects and learn whatever models it needed automatically. In addition there may be activities such as "tidy

up the nightstand" that are specific to an individual. Again, being able to learn these things on the fly given verbal (and perhaps gestural) guidance would be a benefit. This is what we have endeavored to create: a speech guided mobile robot that can learn new nouns and new verbs based on user instruction. Fig. 1 shows a block diagram for our Extensible Language Interface (ELI) and the physical robot it controls. What we have built is essentially a service dog with more language and less slobber.

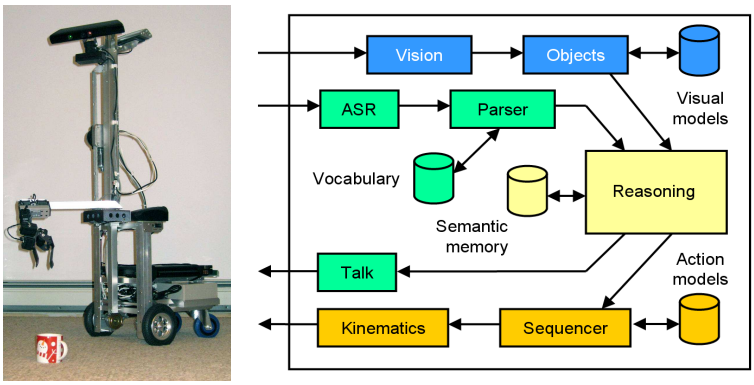

Fig. 1. Our robot can interpret spoken commands as well as learn new nouns and verbs. The experiments here were performed using the arm and camera from our large robot Eli (left) mounted on a table top in order to reduce the degrees of freedom to be controlled.

Of course this is not the first home robot or the first mobile manipulator. There is the impressive PR2 from Willow Garage [3] which can do things like fold towels (but slowly, and for $400K). HERB, developed at CMU [4] is also intended to perform household tasks, but currently requires environmental modifications for its vision system. Then there is El-E from Georgia Tech [5] that was specifically created to retrieve objects for disabled persons. However, none of these robots are designed around a speech interface – to change their actions you either completely change their programs or you configure options in a GUI. Other robot such as Carl [6] and Cosero [7] can take speech input, but require a handheld or headset mike. Furthermore, in general these robots are not intended to learn in the field from user interaction. Instead they have various preprogrammed competencies, object models, and environmental maps which are developed offline.

Other work has addressed language-based learning. Much of this, however, has started at a very low level. Steels [8] looks at the emergence of a private language between cooperating agents while Roy [9] attempts to directly associate acoustic fragments with visual fragments. What we believe is more useful is to stick with a human language and just attempt to find suitable bindings for a few unknown words. This is akin to the approach taken in HAM [10] for learning place names. Similarly, procedure learning is often attempted through trial-and-error experimentation [11] or using the impoverished feedback of reinforcement learning [12]. Yet explicit macro definitions or verbal scripting [13] is often faster and more effective in practice.

2 Multimodal Instructional Dialog

The goal for our system is for the user to describe a task, through a combination of speech and gesture (multi-modal), and then have the system successfully accomplish this task. If it is unsure about some aspect of the task, it should ask clarifying questions (dialog). In addition, we want the system to be able to learn about new objects and new procedures to enable a "verbal programming" facility (instruction). All these capabilities are described below and demonstrated in an associated video [1].

2.1 Robotic Substrate

Since our example tasks all concern objects on a table, it is important for the robot to detect objects. To do this it looks for "obvious" objects, as shown in Fig. 2. It starts by color enhancing the scene from its camera, then builds a model in terms of HSI bands that pass the bulk of the pixels (i.e. the table). The "holes" in this mask are then potential objects. A similar method is used with the depth camera on the large robot. However instead of modeling the table in terms of a dominant color, it is modeled as a 3D plane. Again, deviations from this model are potential objects. Once an image segmentation has been performed, the color(s), shape, size, and relative positions of the objects can be computed.

Fig. 2. The system uses a simplified object finding routine. The steps of this process are to take the input image (left) and enhance its color, find a uniform description for the majority of the area (middle), then identify isolated, non-table regions (right).

To actually grab an object, the 2D image coordinates must be turned into a 3D position for the arm. To do this we compute a homography based on 4 calibrated points that maps 2D image locations to 2D locations on the table surface. We then select an image point likely to be near the middle of the bottom of the object and apply this transform in order to find its x and y. A fixed z position of 1.5 inches above the table is specified to complete the grasp point. Next we solve for the inverse kinematics of the arm, then plot a linear endpoint trajectory from the current position to a "via" point in front of the object such that the gripper is aimed toward the object at this location. A second short trajectory then leads from the via point to the grasp point to ensure a reasonable approach direction for the gripper.

Another important basic capability is understanding human gesture. Here (Fig. 3) we use background subtraction to find the user's moving hand. We track the most extreme point of the difference region (left) and, once it stops moving, generate a

"click" on the image. Given the previously detected objects, we can map this to the most likely one (middle). Similar processing allows the robot to detect when a human hand has entered an object "transfer zone" (right). In such a case the robot either opens or closes its gripper. Once again, this same algorithm for hand detection can be applied even more easily and robustly to depth data.

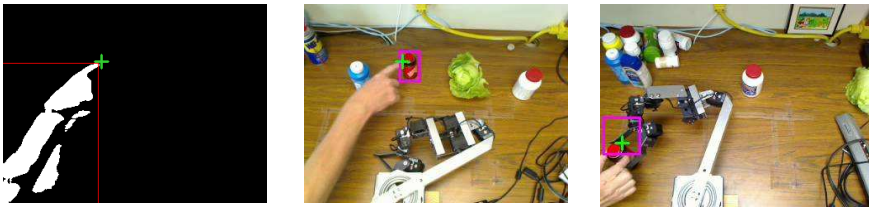

Fig. 3. Gesture recognition is implemented by using background subtraction to track the user's hand. The most extremal portion of this mask (left) selects one of the objects previously identified (middle). User motion detection for object handoff (right) works in a similar manner.

2.2 Natural Language Interpretation

For speech recognition we use an Acoustic Magic VT2 far-field array microphone. Interpretation is performed using a semantic grammar with the Microsoft ASR Engine in Windows XP, although we have also successfully used the IBM Attila engine [14].

An example semantic grammar is shown in Fig. 4. Here there are a number of rules prefixed by "=" that offer several valid expansions for each non-terminal. Elements in parenthesis are optional, whereas the asterisk denotes an unconstrained dictation of up to 5 words. In general, we assume that all expansions for "toplevel" start and end with a silence segment. To prevent spurious action when humans are talking to each other, we require the presence of an attention word (e.g. "Eli") at either the start or end of each such directive. After generating a valid parse, the resulting tree of expansions is mined to generate a simple slot-value representation for the utterance (top). To do this we take each capitalized non-terminal as a slot and assign it the value of whatever first level expansion was used. As can be seen in the example utterance, many of the surface words are simply discarded.

Using the visual object detection and characterization methods previously described, along with a more complex semantic grammar, the robot can grab objects specified by color, size, position, or gesture. It can also answer questions about objects that have been selected in this way. Fig. 5 provides a transcript of an experiment testing the robot's proficiency. One interesting aspect of this conversation is how the robot resolves pronouns through non-linguistic means. If there is only one object present, the binding for "it" is obvious. However if there are several objects, the robot will execute a dialog move to seek clarification. By contrast, if some particular object had recently been mentioned, the robot assumes that this is the proper grounding for the pronoun instead. Eli is also capable of executing a mixed mode dialog response, as when it suggests which of the two white objects the user might have wanted by pointing. Finally, the robot also knows the limits of its own abilities in terms of reach

and grasping size. That is why, when directed to grasp the green object (the head of lettuce shown in Fig. 3), it demurs.

"Eli, please grab the blue bottle now." → { CMD=hand_grab, COLOR=blue }

```
=[toplevel]
<attn> (<intro>) <request> *
* <request> (<intro>) <attn>

=[attn]
Eli
robot

=[intro]
please
first
next

=[request]
<MOVE>
<CHAT>
<QUERY> <desc>
<CMD> <desc>
<learn>

=[CMD]
<hand_indicate>
<hand_select>
<hand_grab>
<hand_give>

=[hand_grab]
grab
grasp
lift
touch
pick
pick up
select

=[desc]
<np> (<pp>)

=[np]
<PRON>
<POINT> <obj>
(<det>) (<SIZE>) (<COLOR>) <obj>
(<det>)  (<POSITION>)  (<COLOR>)
<obj>

=[det]
the
a
an

=[COLOR]
<red>
<orange>
<yellow>
<green>
<blue>
<purple>
<black>
<gray>
<white>

=[blue]
blue
dark blue
light blue

=[obj]
(<measure>) <NAME>
<REF>
object
objects
thing
things
bottle
bottles
```

Fig. 4. This is part of the grammar used for speech parsing. A full utterance is converted to a set of slots and values (top) based on the capitalized categories and their immediate children.

"Grab it." (1 object)
    <grabs object> ← *no confusion since only 1 choice for "it"*
"Grab it." (4 objects)
"I'm confused. Which of the 4 things do you mean?" ← *knows a unique target is required*
"What color is the object on the left?" (4 objects)
"It's blue." ← *understand positions & colors*
"Grab it" (4 objects)
    <grabs blue object> ← *uses "it" from previous interaction*
"Grab that object" (human points)
    <grabs object> ← *understands human gesture*
"Grab the white thing." (2 white objects)
"Do you mean this one?" <robot points> ← *uses gesture to suggest alternative*
"No, the other one."
    <grabs other object> ← *uses "other" from previous interaction*
"Grab the green thing." (head of lettuce)
"Sorry, that's too big for me." ← *sensitive to physical constraints*

Fig. 5. As this transcript of one of the video demos [1] shows, the robot can resolve pronouns based on context, understand gestures, and request clarification when needed.

2.3 Visual Object Naming

While colors, sizes, positions, and pointing can be used to draw attention to specific objects, in some cases it is more convenient to give objects names. One can then simply say "Give me the WD-40" and have the robot figure out which object this is. Of course to do this, the robot must know that "WD-40" is a valid object. It must also know what the object looks like in order to find it. To teach the robot new nouns like this, we use a simple speech pattern: "NP is called X". Here the NP is any valid noun phrase in the grammar, such as "The big bottle" or "That thing" (with pointing). The X is then either drawn from a list of likely (but unknown) object words, or is an unconstrained dictation item.

When the user names an object, the first thing that happens is that a visual model of the object is built. This consists of a coarse size and shape description, plus a histogram of semantic color features (e.g. 50% blue, 30% yellow, 20% red). For our small universe of objects on a table, this is sufficient to find similar objects. If the same name is taught multiple times, the system will learn multiple models for the object. This nearest-neighbor classifier adds robustness since the appearance of objects often varies from side to side, or from different vantage points. Note, that although an object can be described verbally with enough specificity to select it from among other items, when the robot actually experiences an object it can build a much richer model.

"Eli, what is the object on the left?"	
"I don't know."	← *no existing visual model matches object*
"Eli, that is aspirin."	
<new word added to grammar>	← *word acquired via dictation*
<new visual model for object>	
"Okay. This is aspirin." <points>	
"Eli, this object is Advil." (human points)	
<new visual model for object>	← *word already known*
"Okay. That is Advil."	
"Eli, how many Advil do you see?"	
"I see two."	← *uses existing visual model to find item(s)*
"Eli, give me the Tylenol."	
<gets bottle>	← *uses existing visual model to find item(s)*
"Here you go"	
<waits for user hand motion> <releases>	
<waits for user hand motion> <regrabs bottle>	
"Thanks."	
<replaces bottle>	
"Eli, where is the aspirin?"	
"Here." <points>	← *uses existing visual model to find item(s)*

Fig. 6. As this transcript of one of the video demos [1] shows, the robot can be taught new nouns by simply showing it objects. The new visual model can then be used in various ways.

The second step in learning is to add the declared name to the <NAMES> category in the grammar. This is kept distinct from generic nouns like "object" because items in the <NAMES> class usually have one or more visual models associated with them. An interesting problem we have run into is that the dictation results are not always reliable. For instance, when the user says "aspirin" the system sometimes hears "of-

fering". For a speech-only system this is fine since a name is just a random acoustic label. If the robot hears "Pick up the offering" it will perform the correct action. In fact, humans managed to exist for thousands of years with just such cues, having no written language or fixed orthography. However when trying to look up properties of an object elsewhere (as in the next section), the correct term "aspirin" yields much more relevant information.

Fig. 6 gives the transcript from an experiment in which the robot's learning of new nouns was tested. As can be seen, objects can be indicated either verbally or by pointing. The robot can then use its learned models to find things, count them, and name them when requested.

2.4 Semantic Web Access

Many useful functions can be performed by an eldercare robot with just the perceptual and manipulation capabilities already described. However, we can also provide smarter guidance about proposed actions using external data. At our Tokyo lab we built a remote consultation agent called Brainy Robot And Intelligence Networked System (BRAINS) that has access to richer semantic information, largely based on the names (types) of objects. Every time the robot interprets a local utterance, it forms a potential action plan and transmits this (via TCP/IP socket) to BRAINS for vetting. A sample of the communication is shown in Fig. 7. The robot generates semantic network triples describing the proposed action, then BRAINS can either accept or veto the action, or counter-propose some other action.

"Now hand me some aspirin"

robot: act-7 --instance-of--> give
robot: act-7 --status--> proposed
robot: act-7 --target--> obj-3
robot: obj-3 --status--> visible
robot: obj-3 --instance-of--> aspirin
robot: *over*
  BRAINS: act-7 --status--> **vetoed**
  BRAINS: act-8 --instance-of--> say
  BRAINS: act-8 --status--> allowed
  BRAINS: act-8 --message--> "**But that will hurt your stomach.**"
  BRAINS: *over*
robot: act-8 --status--> completed
robot: *over*
  BRAINS: *over*

Fig. 7. The robot communicates with the BRAINS system using semantic network triples.

Fig. 8 shows the transcript of an experiment with BRAINS in the loop. In one case, it consults a database for the user and discovers an aspirin intolerance and thus vetoes dispensing it. Tylenol (paracetamol) does not raise such concerns, hence BRAINS allows this action to be performed. However we also maintain a personal history (LifeLog) for the user and record when Tylenol was given. Thus, when in the last

interaction the user again requests Tylenol (perhaps because of memory loss or simply impatience), BRAINS vetoes the action because sufficient time has not elapsed between doses. The other interaction demonstrated here makes use of a taxonomy built for IBM's Jeopardy! project [15]. The user requests a medication (Rolaids) which is not only unknown, but not present on the table. Yet by using the taxonomy and information about the scene, BRAINS can suggest a similar item that is present.

"Eli, this object is aspirin." (human points)	
<new word added to grammar>	← *word acquired via dictation*
<new visual model for object>	
"Okay. That is aspirin"	
"Eli, the object on the right is called Tums."	
<new visual model for object>	← *word already known*
"Okay. This <points> is Tums."	
"Eli, give me some aspirin."	
<check against personal database>	← *uses existing visual model to find item(s)*
"But that will hurt your stomach."	
"Eli, give me some Tylenol instead."	
<gets bottle>	← *uses existing visual model to find item(s)*
"Here you go"	
<waits for user hand motion> <releases>	
<waits for user hand motion> <regrabs bottle>	
"Thanks."	
<replaces bottle> <records dose in lifelog>	
"Eli, give me some Rolaids."	
"I don't know what Rolaids looks like."	← *no visual model for item*
<ontology used to find available alternative(s)>	
"Do you want another antacid, Tums?"	
"Eli, just give me some Tylenol."	
<lifelog consulted for last dose>	← *uses existing visual model to find item(s)*
"You just had Tylenol."	

Fig. 8. As this transcript of one of the video demos [1] shows, the backed system can look up personal information, reason about substitutions, and monitor events over time.

2.5 Verbal Procedure Learning

Not only can Eli learn new nouns, he can also learn verbs. Fig. 9 shows the transcript from an experimental run where the robot is being taught to poke things. The user teaches the action as a series of steps, like a verbal scripting language, as opposed to imparting some declarative specification of a desired result state. The steps themselves are indexical (as needed) so that, when they are composed, the whole sequence is also indexical. In other words, since the "point" action requires a focus object, the resulting "poke" action does also. As the later part of the transcript indicates, once an action has been learned it can be directly applied to other objects in the scene.

Fig. 10 shows the part of the grammar associated with the verb acquisition process. Learning is initiated either by the user requesting an unknown action, or by explicitly saying "I'm going to teach you how to X". If a word is specified for X, it is added to the grammar and becomes the label for the new action. Once the learning mode is entered, the robot records each successive action request made by the user. Learning

is terminated by a phrase such as “That’s how you do it”. At this point the sequence of parameterized actions is recorded and associated with the X term (possibly from the termination phrase) to give a new action primitive. This “macro” sequence is now invoked when the label X is used as a verb. And, since the user can call for it directly, it can also be included as a step in some other more complicated learned procedure.

“Eli, poke the thing in the middle.”
<new action sequence opened for input> ← *no existing action sequence to link*
“I don’t know how to poke something.”
“Eli, point at it.”
<points> ← *resolves pronoun from previous selection*
“Eli, extend your hand.”
<advances> ← *low level incremental move*
“Eli, retract your hand.”
<retreats> ← *low level incremental move*
“Eli, that is how you poke something.”
<links action sequence to word> ← *recognizes closing of action block*
“Okay. Now I know how to now poke something.”
“Eli, poke the red object.”
<pokes> ← *retrieves action sequence for verb and executes*
“Eli, poke the object on the left.”
<pokes> ← *retrieves action sequence for verb and executes*
“Eli, poke the Tylenol.”
<pokes> ← *retrieves action sequence for verb and executes*

Fig. 9. As this transcript of one of the video demos [1] shows, the robot can be taught a new verb by simply walking it through the appropriate steps.

=[learn]
<NEW-ACT> do something
<NEW-ACT> <ACT-0>
<NEW-ACT> <ACT-1> <arg>
<FINISH> do it
<FINISH> <ACT-0>
<FINISH> <ACT-1> <arg>

=[NEW-ACT]
<teach> <demo> you how to

=[teach]
I'm going to
I am going to
let me

=[demo]
show
tell
teach

=[FINISH]
that's how you
that is how you

=[vp]
do it

=[arg]
something
an object
<desc>

=[ACT-0]
wave

=[ACT-1]
poke
nudge

“poke”

point	1.0
out	1.0
out	-1.0

Fig. 10. Here is a fragment of the grammar (left) the robot uses to learn how to “poke” something (upper right). The result is a parameterized sequence of actions (lower right).

3 Conclusion

We have described how Eli, our speech-based robot manipulator, selects and moves objects around on a table. We explained how the language parsing works, how objects are found, and how human gestures are detected. The robot is also able to answer questions about the scene in front of it and resolve ambiguities in any commands it receives. In addition it can be taught the names of objects and use these labels to access information in remote databases. Finally, it is also possible to "program" the robot by teaching it new named action sequences. The operation of the system and these components was illustrated via transcripts from a series of video experiments [1] with the actual robot. Although our language interpreter is built with conventional technologies, consider a Turing machine by analogy. At its heart there is an FSM which, in itself, is not so interesting. Yet having something like this allows the creature to manipulate the "tape" of culture and thus greatly expand its capabilities.

References

1. J. Connell, "Eli Arm Demos" (video), http://www.johuco.com/eli_arm_demos.wmv, 2012.
2. J. Connell, "Fusing Animals and Humans", *Proc. AGI-08,* pp. 389-393, 2008.
3. S. Chitta, E. Jones, M. Ciocarlie, and K. Hsiao, "Perception, Planning, and Execution for Mobile Manipulation in Unstructured Environments", *IEEE Robotics and Automation Magazine*, 19(2), pp. 58-71, June 2012.
4. S. Srinivasa et al., "HERB: A Home Exploring Robotic Butler", *Autonomous Robots*, 28(1), pp. 5-20, 2010.
5. Y. Choi et al., "Hand It Over or Set It Down: A User Study of Object Delivery with an Assistive Mobile Manipulator", *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, pp. 736-743, 2009.
6. L. Seabra Lopes and A. Teixeira, "Human-Robot Interaction through Spoken Language Dialogue", *IEEE Int. Conf. on Intelligent Robots and Systems (IROS),* pp. 528-534, 2000.
7. J. Stuckler, D. Holz, and S. Behnke, "Demonstrating Everyday Manipulation Skills in RoboCup@Home", *IEEE Robotics and Automation Magazine*, pp. 34-42, June 2012.
8. L. Steels, "The origins of syntax in visually grounded robotic agents", *Artificial Intelligence*, 103, pp. 133-156, 1998.
9. D. Roy, "Grounded Spoken Language Acquisition: Experiments in Word Learning"*, IEEE Trans. on Multimedia*, 5(2), pp. 197-209, 2003.
10. J. Peltason et al., "Mixed-Initiative in Human Augmented Mapping", *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 2146-2153, 2009.
11. J. Siskind, "Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic", *J. of Artificial Intelligence Research*, 15, pp. 31-90, 2001.
12. C. Breazeal et al., "Using perspective taking to learn from ambiguous demonstrations", *Robotics and Autonomous Systems*, 54, pp. 385-393, 2006.
13. J. Connell, "Beer on the Brain", *Proc. of AAAI Spring Symposium – My Dinner with R2D2: Natural Dialogues with Practical Robotics Devices,* pp. 25-26, March 2000.
14. H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila Speech Recognition Toolkit", *IEEE Spoken Language Technology Workshop (SLT)*, pp. 97-102, 2010.
15. J. Murdock et al., "Typing candidate answers using type coercion", *IBM J. of Res. and Dev.*, 56(3/4), pp. 7:1-13, 2012.