# Toward Human-Style Learning in Robots

**Sergei Nirenburg and Peter Wood**
Departments of Cognitive Science and Computer Science
Rensselaer Polytechnic Institute

**Abstract**

This paper presents a system that uses results of deep language understanding to guide a robot's learning of complex activities. Specifically, it demonstrates how natural language communication facilitates grouping primitive actions that the robot is assumed to be able to recognize and perform into named sequences representing complex events. As a result, the robot learns hierarchical transition networks, thus circumventing the notorious knowledge acquisition bottleneck that has posed a major obstacle to AI systems of all kinds.

## Setting the Stage

It is broadly recognized that progress in social robotics is predicated on improving robots' ability to communicate with humans. But there are different levels of communication. While robots have been able to react to vocal commands and other communications for quite some time, this ability has not so far been predicated on a sufficient level of understanding of the meaning of the utterances (or dialog turns) that the robots obtain as inputs from the human collaborator. Indeed, human-robot collaboration has been largely studied using observation-based methods of acquisition of skills (e.g., learning from demonstration).

Not that the robotics community has willfully disregarded the promise of language-endowed robots. The decision to bypass true language processing in human-robotic interaction has been forced on the community. The reason for this is that extracting, representing and manipulating the meaning of natural language texts – true language understanding – is a very challenging task. A major component of this task's complexity is that true language understanding extends well beyond language as such. If one's aim is to make robots understand language with a proficiency approaching that of humans, then one must also make the robot understand extralinguistic context: the way the world (or at least the application domain) is organized, including what role(s) the robot and the human collaborator(s) can play, and the current situation (or discourse context), including the current goals and plans of the human collaborator.

Once such "deep" natural understanding is implemented, even with a knowledge substrate narrower than that possessed by an average human, a number of new possibilities become available. One such potential new capability is to use the robot's newly acquired ability to understand language to help it learn new concepts and new lexical items that can then help it improve its understanding of the world and become more human-like as a collaborator. This kind of learning, reliant as it is on overt semantics and knowledge, will be radically different from any current machine learning paradigm. This style of learning mirrors a major mode of learning in humans. Indeed, since an early age all of us have spent a lot of our time learning through language: by being instructed in school and elsewhere and by reading. This paper presents our first step toward endowing robots with this kind of learning capability.

## Learning How to Build a Chair

Furniture assembly is a well-known application task in robotics. It has been widely used as a sample domain to demonstrate human-robot collaboration on a joint activity (e.g., Knepper et al. 2014). The system described by Yale robotics group led by Brian Scassellati (Roncone et al. 2017) supplies the robot with high-level models of tasks, represented in the hierarchical transition network (HTN) formalism, and then uses limited-bandwidth communication between the human and the robot "to convert high-level hierarchical models into low-level task planners capable of being executed by the robot." Our work is a collaborative effort with the Yale group and is directed toward broadening the bandwidth of human-robot communication by endowing the robot with the capability of understanding and, hence, learning in a way approaching human learning.

Roncone et al. assume that the initial high-level HTN representation is available to the robot and proceed to demonstrate how a narrow-bandwidth communication channel with a human collaborator can help the robot learn the low level POMDP-based representation that overtly addresses the problem of allocating tasks to the human and to the robot when they collaboratively build a chair. In this paper, we address an earlier step in this process: we demonstrate how, instead of assuming that a robot starts out with an HTN, a human can help a language-endowed robot to **learn** the HTN for building a chair.

Figure 1 shows a sample input sequence juxtaposing Roncone et al's non-linguistic communication style with the natural-language-enhanced version we are developing. The non-linguistic commands, shown without quotation marks, indicate primitive actions that the robot is ordered to carry out along with the objects to which they apply. The speech utterances, shown in quotation marks, describe the sequence in natural language. Note that the robot is completely trusting the human: the sequence may, in fact, not correctly describe the process of building a chair.

1. "You will build a chair."
2. "First, you need to get a screwdriver."
3. Action: Get; Object: screwdriver
4. "Then, you are going to build the legs."
5. "First, you will build a front leg."
6. Action: Get; Object: dowel
7. Action: Get; Object: leg
8. Action: Tighten; Object: screw
9. Action: Get; Object: dowel
10. Action: Get; Object: leg
11. Action: Get; Object: top-bracket
12. "You built a back leg."
13. "Then you will build another front leg."
14. Action: Get; Object: dowel
15. Action: Get; Object: leg
16. Action: Tighten; Object: screw
17. "You will build another back leg."
18. Action: Get; Object: dowel
19. Action: Get; Object: leg
20. Action: Get; Object: top-bracket
21. "You are going to build the back."
22. Action: Get; Object: board
23. Action: Get; Object: bracket
24. Action: Tighten; Object: screw
25. Action: Get; Object: board
26. Action: Get; Object: bracket
27. Action: Tighten; Object: screw
28. "You built the seat."
29. Action: Get; Object: back
30. Action: Get; Object: dowel
31. Action: Get; Object: chair-bracket
32. "You attached the back to the seat."

***Figure 1****. A sample input sequence combining non-linguistic commands with natural language utterances.*

At the start of the process, the robot is expected to know a set of primitive actions (e.g., fetching) and primitive objects (e.g., a dowel). The "visible" part of the process consists of the human giving the robot a sequence of inputs (Figure 1). The result of the process will be the robot knowing a set of complex actions – both their names and their semantics – represented as an HTN. The inputs from the human are of two types: a) commands to execute a (known) action affecting certain objects, whose types are also already known to the robot; and b) natural language utterances introducing (naming and partially describing) actions hitherto unknown to the robot.

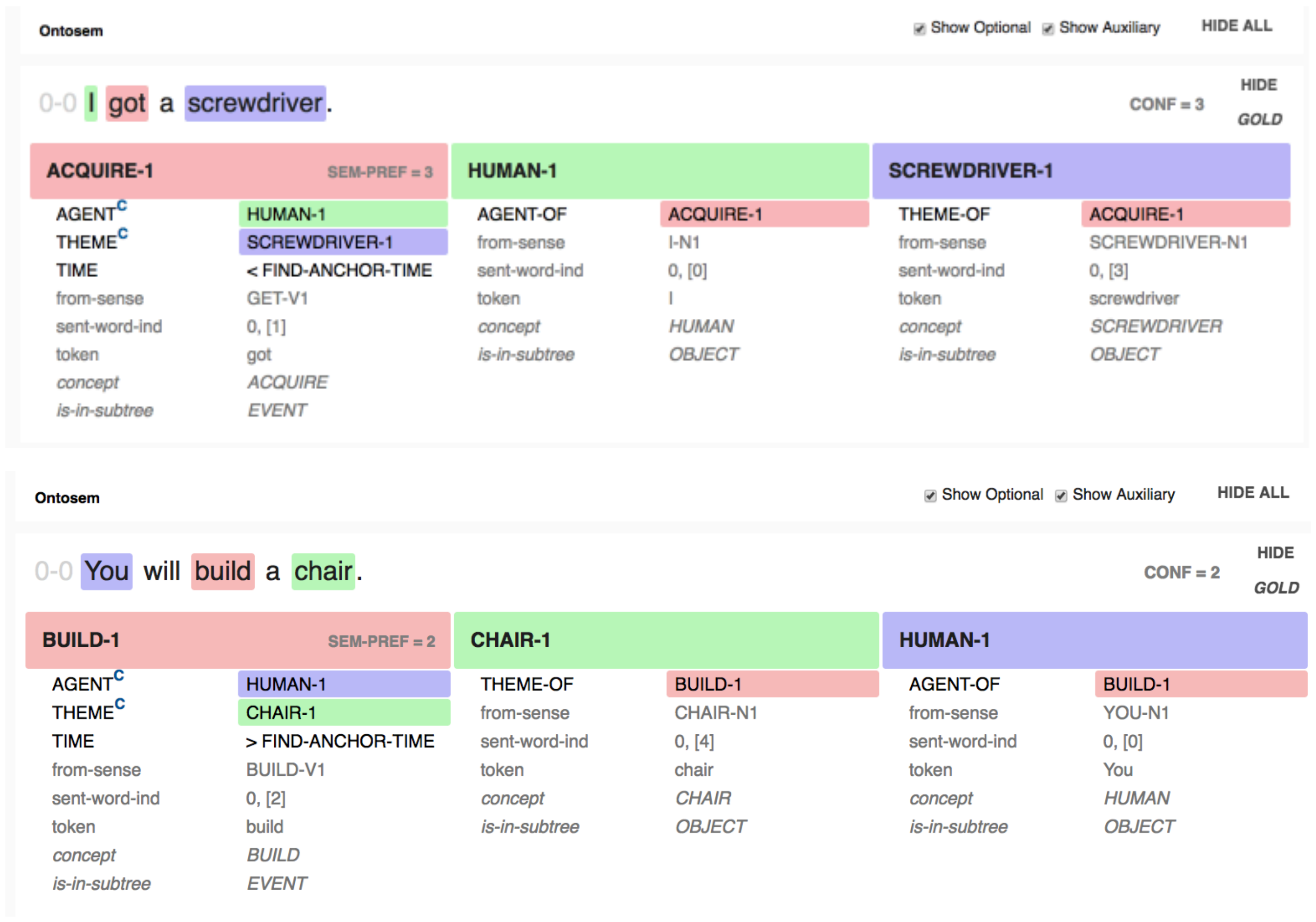
The decision to use both types of inputs stems exclusively from the desire for smooth integration with the system of Roncone et al., where non-linguistic commands to execute an action are directly connected with the robot's effectors, with no need for representing and manipulating their semantics. Once that system is augmented with the capabilities described in this paper, the robot will no longer require the non-linguistic commands and, instead, will be able to learn by reasoning over the results of processing natural language inputs from the human, without having to actually carry out the actions directly specified in the commands. A formal representation of the semantics of the actions and objects in the input utterances is a prerequisite for this capability that our approach provides.

So, instead of the non-linguistic commands, the human instructor will be able to issue language utterances describing those commands. The robot will understand the utterances and interpret their semantics in terms of concepts already present in its knowledge base. Those concepts will, of course, also be linked to specific actuator procedures of the robot, thus providing a semantic substrate for connecting language perception, reasoning and action. (Once the robot is equipped with additional perception modalities, such as vision, the results of their processing will also be connected with the robot's knowledge substrate.)

Even if we retain the two types of input from humans, the results of processing the commands and the utterances should be represented in a single formalism. This will facilitate a variety of robot functionalities down the line, but the immediate need is the use of a semantically uniform representation for all the elements (nodes) of the HTN resulting from the process described in this paper. This includes the HTN's terminal nodes that stand for the primitive actions (conveyed by commands) that the robot is supposed to know at the start of the process as well as its non-terminal nodes that stand for complex actions (conveyed by utterances) learned by the robot during this system's operation.

## Stage One: Input Understanding with OntoSem

The first stage of the robot's learning process takes as input a sequence of non-linguistic commands and utterances like the one in Figure 1 and outputs a sequence of meaning representations (MRs) of these commands and utterances. These MRs are obtained automatically using the OntoSem language understanding system (McShane et al. 2016), a



Ontosem

Show Optional ✓ Show Auxiliary ✓ HIDE ALL

0-0 I got a screwdriver. CONF = 3 HIDE GOLD

| ACQUIRE-1 | SEM-PREF = 3 | HUMAN-1 | | SCREWDRIVER-1 | |
|---|---|---|---|---|---|
| AGENT^C | HUMAN-1 | AGENT-OF | ACQUIRE-1 | THEME-OF | ACQUIRE-1 |
| THEME^C | SCREWDRIVER-1 | from-sense | I-N1 | from-sense | SCREWDRIVER-N1 |
| TIME | < FIND-ANCHOR-TIME | sent-word-ind | 0, [0] | sent-word-ind | 0, [3] |
| from-sense | GET-V1 | token | I | token | screwdriver |
| sent-word-ind | 0, [1] | *concept* | *HUMAN* | *concept* | *SCREWDRIVER* |
| token | got | *is-in-subtree* | *OBJECT* | *is-in-subtree* | *OBJECT* |
| *concept* | *ACQUIRE* | | | | |
| *is-in-subtree* | *EVENT* | | | | |

Ontosem

Show Optional ✓ Show Auxiliary ✓ HIDE ALL

0-0 You will build a chair. CONF = 2 HIDE GOLD

| BUILD-1 | SEM-PREF = 2 | CHAIR-1 | | HUMAN-1 | |
|---|---|---|---|---|---|
| AGENT^C | HUMAN-1 | THEME-OF | BUILD-1 | AGENT-OF | BUILD-1 |
| THEME^C | CHAIR-1 | from-sense | CHAIR-N1 | from-sense | YOU-N1 |
| TIME | > FIND-ANCHOR-TIME | sent-word-ind | 0, [4] | sent-word-ind | 0, [0] |
| from-sense | BUILD-V1 | token | chair | token | You |
| sent-word-ind | 0, [2] | *concept* | *CHAIR* | *concept* | *HUMAN* |
| token | build | *is-in-subtree* | *OBJECT* | *is-in-subtree* | *OBJECT* |
| *concept* | *BUILD* | | | | |
| *is-in-subtree* | *EVENT* | | | | |

*Figure 2. MRs for the non-linguistic command Action: Get; Object: screwdriver (converted to "I got a screwdriver" for purposes of MR generation) and for the natural language utterance "You will build a chair."*

component of the OntoAgent cognitive architecture (e.g., Nirenburg and McShane 2015). The latter is used to integrate all the components of our learning-endowed robot. The MRs obtained as a result of processing input sequences like the one in Figure 1 are stored in the robot's short-term memory (STM).

The operation of OntoSem is supported by extensive knowledge resources, including an ontological world model and an English semantic lexicon. The meanings of entries in the lexicon are interpreted in terms of the ontological world model. Within the OntoAgent architecture, the ontology and the lexicon are components of the robot's long-term memory (LTM). The family of interrelated metalanguages for representing MRs, the ontology, the lexicon and other OntoAgent knowledge resources is described in detail in Nirenburg and Raskin (2004).

At the beginning of each run of the learning process, we ascertain that the robot's ontology covers all the concepts – actions and objects – that are used in the commands in the inputs, such as those in Figure 1. The robot's lexicon covers all the lexical items that we expect the human collaborator to use in a training session. (Treatment of unexpected input in OntoSem is an active research direction (see, e.g., Nirenburg and McShane 2016, McShane et al. 2017). The decision to avoid the treatment of unexpected input at this time was made to simplify the experimentation with robotic learning through language. We fully expect to remove this constraint in future versions of the system.) The results of learning are made manifest through the automatic augmentation of the robot's ontology. Specifically, as a result of the learning process, the robot will:

- create new complex events,
- name them,
- determine their constituent subevents,
- determine the events of which they are subevents, and
- record this new knowledge in its ontology.

In the system of Roncone et al., the robot does not record its actions in memory. The learning robot that we are describing must remember what it has done. So, it must record the results of its processing of both kinds of input to use them in the subsequent learning stage. All MRs are generated by OntoSem: those for utterances are generated in the usual way; those for non-linguistic commands are generated by representing the commands as simplified utterances. Thus, to obtain an MR for the command *Action: Get; Object: screwdriver* we send the 'utterance' "Get screwdriver" to OntoSem.

Figure 2 illustrates the output MRs produced by OntoSem for the utterance "You will build a chair" and the command *Action: Get; Object: screwdriver*, respectively. Note that the latter is represented in the robot's STM not as a command but as the result of an action that it has already performed, which is rendered as the MR for the meaning of the statement "I got a screwdriver." An attentive reader will notice that at the time of processing the concept ACQUIRE[1] in the robot's ontology has the constraint human on the filler of its agent case role. This is because this robot 'inherited' a worldview in which this action could be carried out only by people. After learning that it can also get a screwdriver, the robot will be able to automatically augment its ontology to include robot as another legal filler for the agent case of acquire. We do not describe this process in this paper.

## Stage Two: Robotic Learning

Once Stage One processes have generated MRs for every member of the input sequence and stored them as a time-ordered sequence in the STM, the learning stage commences. The objective of this stage is to produce an HTN from the sequence of MRs at input. Effectively, the process expects the MRs corresponding to commands in the original input to form the set of terminal nodes in the resulting HTN tree. We will call these MRs MRPs (MR-Primitive). Non-terminal nodes in the resulting HTN will be created, named and placed in the tree on the basis of the MRs corresponding to utterances in the original input. These latter MRs we will call MRUs (MR-Utterance).

The algorithm for the learning process is illustrated in Figures 3-5. The learning process that we report at this time is implemented using two procedures – **Build-HTN** and **Resolve-Disputes**. We expect the input sequence to be ambiguous. Therefore, we expect **Build-HTN** to yield situations where the ancestry of some nodes will be disputed. A disambiguation procedure, **Resolve-Disputes**, is included in the learning process. At present, this procedure is called after the initial processing of the input string of MRs by Build-HTN. The instances of disputes that must be resolved are marked when a call to **Find-Closeable-Branch** fails.

Build-HTN (Figure 3) processes MRs in the order they appear in the input. The procedure **Add-Nod**e has three parameters: a) current-parent (CP), the node of which the newly created node will become a child,[2] b) the type of the node – terminal or nonterminal; and c) the node's name. The procedure **Name-Node** (Node) concatenates the EVENT of the MR with the THEME of the MR. Next, if there is a secondary theme, in the MR, it concatenates "-AND-" THEME2. If the INSTRUMENT case role in the MR is filled, it is concatenated and preceded with "-WITH-."

```
Build-HTN
For each MR in an input sequence:
Read the MR
IF  MR = MRP
 THEN IF CP has no nonterminal children
    THEN Add-Node(CP, terminal, Name-Node(MRP_i))
       ;routinely incorporate this MRP, note call to Name-Node
    ELSE CP ← Add-Node(CP, nonterminal, dummy)
        ;we need a dummy nonterminal but we can't yet name it
        Add-Node(CP, terminal, Name-Node(MRP_i))
         ;and we add the new terminal node as the child
         ;of the newly created nonterminal
  ELSE   ;MR is an MRU!
          IF MRU_i.time ≥ t_0  ;t_0 is time of speech
             ;we call such MRUs non-past MRUs, NPMRUs
       THEN
           Adjust-Branch(MRU_i, MRU_i-1)
             ;find correct CP to attach
           Add-Node(CP, non-terminal, Name-Node(MRU_i))
           CP ← (MRU_i)
             ;the root from which to grow the next branch
       ELSE ;the case when MRU is a past MRU, PMRU
             ;Either a previously-opened branch will be closed,
             ;or a dispute will be recorded.
         Find-Closeable-Branch
         IF Find-Closeable-Branch returns failure
         THEN CP ← CP.parent
                 Add-Node(CP, non-terminal, Name-Node(MRU_i))
                 Disputes += (MRU_i, MRU_i-1)
                   ;MRU_i-1 is the previous child of CP)
```

*Figure 3. The Build-HTN procedure.*

The procedure **Adjust-Branch** (Figure 4) assesses boundaries between consecutive (*current* and *previous*) nodes in the tree and adjusts the current parent so that the new branches are appropriately positioned.

```
Adjust-Branch(current, previous)
IF current is MRU and previous is MRP
  THEN CP ← CP.parent
IF current is NPMRU
  THEN candidate ← CP.parent
  WHILE current.theme not PART-OF candidate.theme
     candidate ← candidate.parent
  IF candidate != root
     THEN CP ← candidate
```

*Figure 4. The Adjust-Branch procedure.*

When an NPMRU follows an MRP in the input, this means that a new branch of the tree must be added. By default,

[1] We follow the OntoAgent notational convention of using SMALL CAPS to designate concepts in the agent's ontology. In the current OntoAgent ontology, ACQUIRE does not presuppose inalienable possession.

[2] This parameter is optional. The root node is created unattached.

this means that the current parent should be moved one level up, so that the root of the new branch is a sibling of the root of the branch that was completed when the current NPMRU was input. However, conceptually, the new branch can belong to a higher node in the tree. To determine what node that should be, the Adjust-Branch procedure uses ontological knowledge to determine whether the filler of the THEME case role in the MR in question is a component of (technically, a member of the set that forms the filler of) the HAS-AS-PART property of the filler of the case. Thus, since the robot's ontology contains information that legs can be parts of chairs but not of screwdrivers, node 4 is attached to node 1 and not node 2 in Frame B of the example below.

The occurrence of a PMRU in the input string signifies that a branch in the nascent tree must be closed. The procedure **Find-Closeable-Branch** (Figure 5) finds at what node in the tree this branch opened and adjusts the CP accordingly in anticipation of a new branch opening at the next input.

```
Find-Closeable-Branch(MRU)
candidate ← CP
WHILE not (candidate is dummy or candidate has same name as MRU)
  candidate ← candidate.parent
IF candidate = root
  THEN return failure
IF candidate.name = dummy
  Name-Node(candidate, MRU)
CP ← candidate.parent
```

*Figure 5. The Find-Closeable-Branch procedure.*

If the procedure fails, then a) a new node is created as the child of the current CP (that is, a sibling of the parent of the previous nonterminal node), and b) a dispute is recorded for future resolution.

Disputes arise when the input string of MRs contains a substring of consecutive MRPs with an NPMRU, $N_p$, preceding them and a PMRU, $N_f$, following them (cf. the substring between the $13^{th}$ and the $20^{th}$ element in the sample input of Figure 1). This means that the input is ambiguous: some of the disputed terminal nodes may be children of the nonterminal preceding them and some others, of the terminal following them.

**Resolve-Disputes** attempts to resolve this ambiguity. The heuristic it uses is contextual (not ontological, as in the case of **Adjust-Branch**): it tries to find in the tree a nonterminal N, with undisputed children whose name is the same as that of $N_p$ or $N_f$. If found, the children of N are compared with the list of disputed terminals and if the appropriate match is found, the dispute is resolved by assigning the matched subsequence to $N_p$ if it matches the beginning of the disputed sequence and to $N_f$ if it matches the end of it, with the remainder of the sequence being assigned to the $N_f$ and $N_p$, respectively. If this type of match cannot be found, the dispute remains unresolved. This means that the robot's learning will be incomplete in this case. The robot will still record the incomplete result in its LTM and will expect to resolve this issue in future processing (a major property of the kind of learning we are implementing is that it is life-long). An operation of Resolve-Disputes is illustrated in Frames H and I below.

## An Example

The frames A-H in Figure 6 show the nascent HTN for the input sequence in Figure 1 at various points in the robot's learning process. Numbers in the nodes correspond to the elements of the input sequence, the highest number in the frame being the element last processed. To save space, sequences of MRP s are bunched in one node. The nodes highlighted in bold are CPs at that point in the process. Sets of disputed nodes are marked with dotted lines. MRPs are in lowercase, MRUs, in all-caps. Thus, Frame A shows the nascent HTN after the first three elements of the input sequence are processed. Frame B captures the state after **Adjust-Branch** is called, while Frame D illustrates the state of the HTN before a call to **Adjust-Branch**. Frame C illustrates the first, and Frame E, the second time a dispute is marked. Frame F illustrates the situation when an unnamed grouping of primitive actions is introduced, and Frame G illustrates how, after the PMRU "You have attached the back to the seat," the corresponding nonterminal node is named. Finally, Frame H illustrates the operation of **Resolve-Disputes**, made possible by the comparison of Input 13 with a subset of the disputed sequence 6-11 triggered by the fact that inputs 5 and 13 both announce building a front leg.

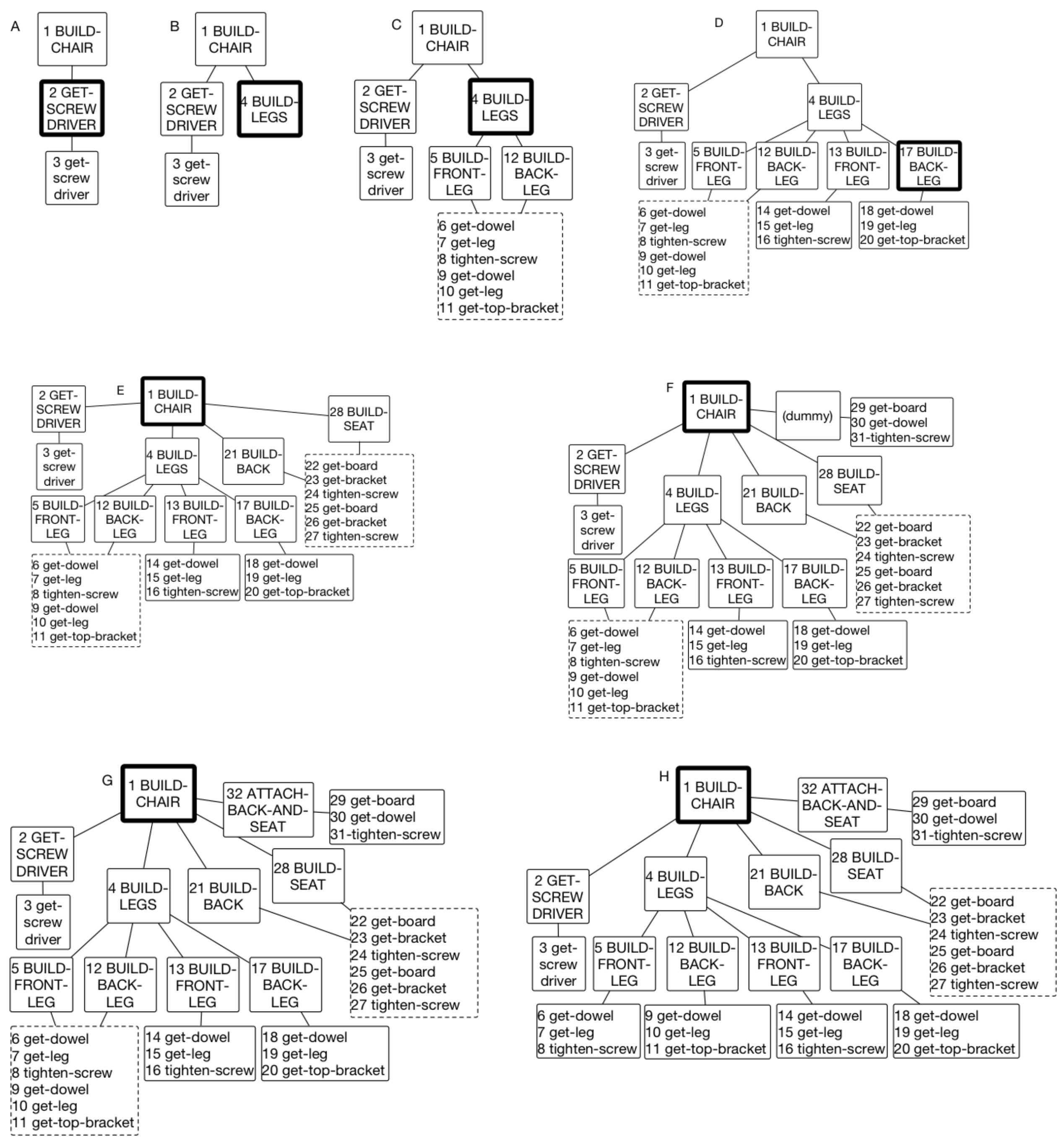


*Figure 6. The nascent HTN for the input sequence in Figure 1 at various points in the robot's learning process.*

## Discussion and Future Work

The mechanism described in this paper is a step toward integrating true language understanding, reasoning and robotic action in a comprehensive cognitive robotic architecture. Experiments with such integration have been conducted before, notably by Matthias Scheutz and his coworkers (e.g., Scheutz et al. 2013, 2017). Our group's theory and methodology shares many high-level objectives and assumptions with Scheutz's group. We cannot discuss the differences in detail due to space constraints. Suffice it to say that our group extensively models the knowledge internalized in the robot's memory, which allows deeper integration of language processing and general reasoning, including understanding what was not overtly mentioned in the incoming utterances.

We view the work presented in this paper as the beginning of a sustained research initiative, with a large number of issues to be tackled in the future. This is another way of saying that the current implementation of the system operates under a number of constraints and assumptions.

Thus, as currently implemented, the robot does not make sufficient use of its memory. The system presumes that the complex events the robot learns are new to the it. A more realistic situation would be if the robot already had in its ontology the concept corresponding to the complex event described in an input sequence. The newly learned information would then help it hone the concept. An important side effect of such learning can be made manifest if the overall set-up allowed for communication with more than one other member of the robot's team (either human or a robot like itself). This would introduce the potential for attributing any discrepancies in the specification of newly learned concepts to specific sources, which can provide invaluable heuristics for managing the robot's trust in the other team members on specific tasks.

At this time, for our learner to succeed, the human trainer must be rather precise in his or her communications. This requirement is clearly unrealistic in the real world. The current system is already attuned to some cases of imprecisions and paraphrasing in human communications. The procedures **Adjust-Branch** and **Resolve-Disputes** have been introduced specifically for the purpose of dealing with them. However, many more heuristics must be brought to bear in order to allow humans to communicate with the robot in a way approaching that in which they communicate with other humans. This means modeling the human ability to balance vagueness and ambiguity against length and complexity of natural language utterances directed at a particular recipient: the higher the expectations that the recipient understands the language and the context, the more lapidary and imprecise (and shorter) the message can be. This view ultimately ascends to the principle of least effort. Its application to language communication is discussed in Piantadosi et al. (2012).

We will independently continue to enhance OntoSem, the natural language component of our system. In particular, we will concentrate on eliminating brittleness in the face of a variety of types of "unexpected" input and enhancing its ability to resolve a variety of referential expressions (including, notably, elliptical ones) to elements of the robot's ontology and its episodic memory that straddles its STM and LTM. But in parallel to this work, we will also work on expanding the set of heuristics for treating the various failures in the learning process that are due to substandard input from the human. We will look for additional sources of such heuristics in the robot's ontology and the STM component of its episodic memory. We will also investigate using other components of the robot's LTM, such as its long-term episodic memory of actual events and results of its reasoning.

At present the language communication is initiated exclusively by the human. This is not realistic. We will introduce mixed-initiative communication, initially by allowing the robot to backchannel (e.g., saying "OK, got it") and to ask clarification questions. Special attention will be given to heuristics for deciding when (and when not) to ask such questions as well as how to formulate the questions so as to increase the chances of getting the most appropriate answer from the human.

The learning process should not stop with learning what the human trainer wants the robot to learn. We saw above that the robot will have the opportunity to hone its ontological and lexical knowledge by reasoning over the meanings of human's utterances, as illustrated by the above example of modifying the filler of AGENT of ACQUIRE. In an advanced cognitive robot, this kind of learning must be always "on," irrespective of whether any specific communication with others is related to learning.

In the current implementation algorithm, the entire input sequence is interpreted and represented in MRs before the learning process starts. In future versions, we will interleave the interpretation and learning processes, making them more realistic.

The algorithm described in this paper covers conceptual learning on the basis of a mixed input sequence of robotic actions and language utterances. A similar process can improve the execution side of the robot's functioning. It is worth investigating whether broadening the communication channel between the human and the robot that is used in Roncone et al. (2017) for converting the high-level HTN into an execution-ready POMDPs will improve that process.

**Acknowledgements**. This work was supported in part by Grant N00014-17-1-221 from the U.S. Office of Naval Research. Any opinions or findings expressed in this material are those of the author and do not necessarily reflect the views of the Office of Naval Research.

## References

Knepper, R.A., T. Layton, J. Romanishin, and D. Rus, "IkeaBot: An autonomous multi-robot coordinated furniture assembly system," in *IEEE International Conference on Robotics and Automation*, 2013.

McShane, M., Nirenburg, S. and Beale, S. 2016. Language understanding with Ontological Semantics. *Advances in Cognitive Systems* 4:35-55

McShane, M., K. Blissett, and I. Nirenburg. 2017 Treating Unexpected Input in Incremental Semantic Analysis. Proceedings of The Fifth Annual Conference on Advances in Cognitive Systems

Nirenburg, S. and McShane, M. 2015. The interplay of language processing, reasoning and decision-making in cognitive computing. Proceedings of the 20th International Conference on Applications of Natural Language to Information Systems (NLDB 2015). Passau, Germany.

Nirenburg, S. and McShane, M. 2016. Slashing metaphor with Occam's Razor. Proceedings of The Fourth Annual Conference on Advances in Cognitive Systems

Piantadosi, S. T., Tily, H. & Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition* 122, 280–291.

Roncone, A., O. Mangin and B. Scassellati. 2017. Transparent Role Assignment and Task Allocation in Human Robot Collaboration. Proceedings of International Conference on Robotics and Automation. Singapore.

Scheutz, M. J. Harris and P. Schmermerhorn. 2013. Systematic Integration of Cognitive and Robotic Architectures. Advances in Cognitive Systems 2:277-296.

Scheutz, M., E. Krause, B. Oosterveld, T. Frasca, and R. Platt. 2017. Spoken Instruction-Based One-Shot Object and Action Learning in a Cognitive Robotic Architecture. Proceedings of AAMAS '17.