# Grounding Language for Interactive Task Learning

**Peter Lindes, Aaron Mininger, James R. Kirk, and John E. Laird**

Computer Science and Engineering

University of Michigan, Ann Arbor, MI 48109-2121

{plindes, mininger, jrkirk, laird}@umich.edu

## Abstract

This paper describes how language is grounded by a comprehension system called Lucia within a robotic agent called Rosie that can manipulate objects and navigate indoors. The whole system is built within the Soar cognitive architecture and uses Embodied Construction Grammar (ECG) as a formalism for describing linguistic knowledge. Grounding is performed using knowledge from the grammar itself, from the linguistic context, from the agent's perception, and from an ontology of long-term knowledge about object categories and properties and actions the agent can perform. The paper also describes a benchmark corpus of 200 sentences in this domain, along with test versions of the world model and ontology, and gold-standard meanings for each of the sentences. The benchmark is contained in the supplemental materials.

## 1 Introduction

This paper considers language grounding within the context of Interactive Task Learning (ITL; Laird et al., 2017), where the goal is to teach an intelligent agent new tasks and extend existing tasks through natural language instruction by a human teacher. This kind of instruction has been done with an agent called Rosie (Mohan and Laird, 2014). Rosie has been interfaced to a tabletop robot arm and a mobile robot that can navigate and perform tasks in an indoor environment. We discuss the techniques and strategies used to ground the natural language input to the knowledge the agent has about the world and its own capabilities.

Rosie can perform simple manipulation tasks like *'Pick up the green sphere.'* or *'Put that in the pantry.'*, simple navigation tasks like *'Go to the kitchen.'* or *'Follow the right wall.'* and more complex tasks like *'Fetch a soda.'* or *'Deliver the package to the main office.'* It can also understand descriptions of objects and answer simple questions about its world.

In the work described here, the language comprehension is performed by a system called Lucia. Lucia runs as a part of Rosie, and is under continuing development. Previous work (Lindes and Laird, 2016; 2017a; 2017b) described some aspects of Lucia, but here we describe in some detail how Lucia does language grounding within Rosie. We also provide a benchmark that may be useful for comparing language grounding systems for robots.

### 1.1 Research Context

Our research is embedded in a cognitive modelling approach (Laird et al., 2012). This affects our goals and methods in three ways. First, we attempt to implement all aspects of Rosie's intelligence, including language comprehension, task planning, dialog interaction, etc., within a single agent built on an architecture designed to model general principles of human cognition. Specifically, we use the Soar cognitive architecture (Laird, 2012).

Second, we wish to apply a theory of linguistic or grammatical knowledge, based on cognitive linguistics research, that combines syntactic and semantic knowledge in a single integrated grammar. Lucia uses Embodied Construction Grammar (ECG; Feldman et al., 2009; Bergen and Chang, 2013) as that theory. This theory has the potential to scale to cover much variation in human language use, as well as relating to the complexities of human conceptual models of the world.

Third, with Lucia we seek to build a language comprehension process that conforms to psycholinguistic research on incremental human processing (Christiansen and Chater, 2016) and draws directly on all the contextual knowledge the agent has. This approach may have an advantage in meeting human expectations of how natural language will be understood by the agent.

This cognitive modeling approach differs from many other approaches to language comprehension and language grounding for robots found in the literature. For example, a number of research-

ers have built systems to satisfy the need for grounded language in a robot, including Steels and Hild (2012), Tellex et al. (2014), and Eppe et al. (2016). However, none of those systems conforms to the three aspects of a cognitive modeling approach outlined above. Our work is beginning to explore whether the cognitive approach can provide more robust language understanding.

Lucia could be characterized as a semantic parser using textual and visual context since it takes natural language utterances in, uses knowledge of the surrounding text and the objects seen by the agent, and produces semantic representations as its output. However, it differs in many respects from many other "semantic parsers."

For example, Zettlemoyer and colleagues (Zettlemoyer and Collins, 2007; Artzi and Zettlemoyer, 2013) have built systems for learning mappings from utterances to logical forms. Berant et al. (2013) learn a system to map questions to answers in relation to a large database. Wang et al. (2016) report on a system for learning the language needed to instruct a computer to perform certain tasks.

All these systems have produced impressive results. Our results with Lucia are complementary in two important ways. The meaning representations they produce are not just logical forms but are connected to the perception and action knowledge of a fully embodied agent. Lucia also satisfies the cognitive modelling constraints we outlined above, which none of these other systems attempt to do. The richness of this variety of different approaches should help advance future research. Our contribution is to show something of what is possible using a cognitive model embodied in a robotic agent.

## 1.2 Theoretical Background

Rosie is built within the Soar cognitive architecture. The Soar architecture has a working memory with information about its current perception of the world as well as its internal goals and state. A procedural memory contains production rules that represent knowledge of how to perform internal and external actions. A long-term semantic memory holds knowledge of the categories of objects the agent knows about, what properties these objects can have, and the actions the agent knows how to perform. Dynamic operation in Soar consists of a series of decision cycles, where in each cycle a single operator is selected and applied, and that operator influences which production rules fire to make changes in working memory or initiate external actions during that cycle.

In order to learn new tasks from instruction, Rosie must have a natural language understanding capability. That capability must be able to produce a meaning structure for each input utterance that is *grounded* to the agent's perception, action capabilities, and general world knowledge. By grounded we mean that the resulting meaning structure refers directly to the agent's internal representation of objects, of the actions it knows how to perform, and of any other relevant knowledge the agent has, such as spatial relations between objects.

Several approaches to language comprehension in Soar have been used previously (Lehman et al., 1991; Mohan et al., 2012; Mohan and Laird, 2014; Kirk et al. 2016). More recently a language comprehension system in Soar called Lucia (Lindes and Laird, 2016; 2017a) has been developed. In addition to the general cognitive abilities inherent in Soar, Lucia uses a cognitive theory of language called Embodied Construction Grammar (ECG; Feldman et al., 2009; Bergen and Chang, 2013).

The ECG grammar formalism (Bryant, 2008) defines a grammar in terms of two kinds of items: *constructions* and *schemas*. A construction is a pairing of form and meaning. Some constructions match individual lexical items and others match one or more constituents in a recursive hierarchy. Each construction describes its meaning in terms of schemas.

A schema can be thought of as a feature bundle. It defines a data structure that has a type name and one or more *roles*, or slots, to hold information. A construction defines what schema is to be evoked when it is recognized and how to fill the roles of that schema. Roles can be filled with constants provided in a construction or from the meaning structures of the constituents of a construction, gradually building a complex hierarchical meaning structure as each sentence is comprehended.

As an example of Lucia's comprehension, consider the sentence *'Pick up the green sphere.'* Figure 1 shows the data structures Lucia builds to comprehend this sentence. The blue rectangles represent the constructions that were recognized, the green ovals are the meaning schemas that were
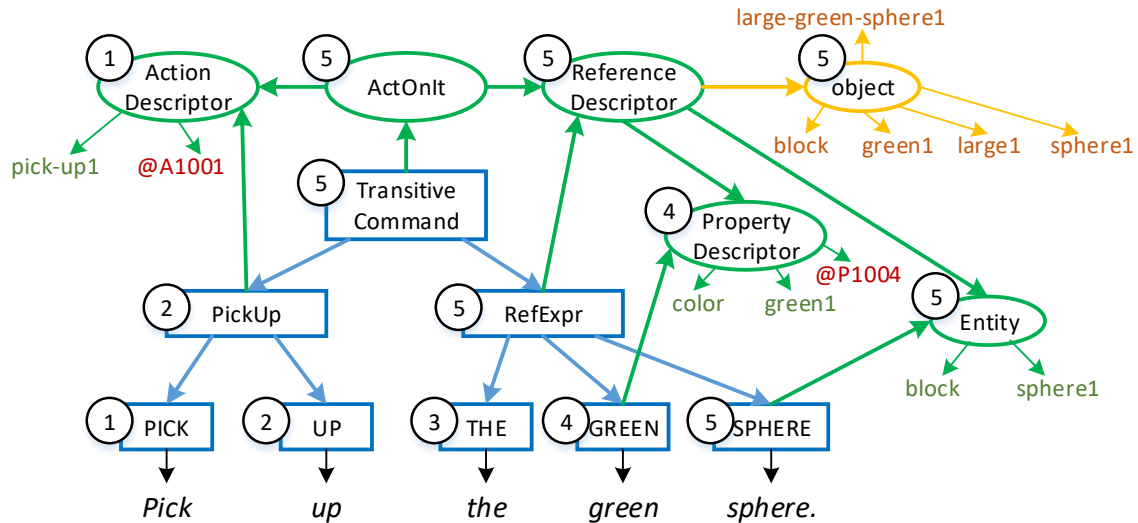
Figure 1: Comprehension of a simple sentence. (Adapted from Lindes and Laird, 2016.)

```
construction TransitiveCommand          schema Action
    subcase of Imperative                   roles
    constructional                              action
        constituents                            location
            verb: ActionVerb
            object: RefExpr                 schema ActOnIt
    meaning: ActOnIt                            subcase of Action
        constraints                            roles
            self.m.action <--> verb.m                object
            self.m.object <--> object.m
```
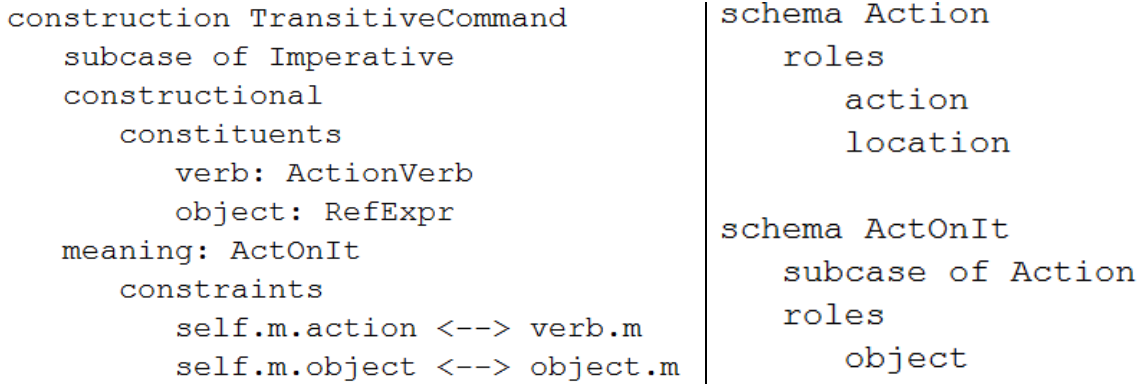
Figure 2: ECG example. (Adapted from Lindes and Laird, 2016.)

instantiated, and the orange and red items represent information derived from grounding.

In the example in Figure 1, there are five word cycles, one for each word in the input sentence. The circled numbers in the figure indicate during which word cycle each construction or schema was instantiated or each grounding link formed.

Figure 2 shows the ECG form of the construction for TransitiveCommand, which produces the blue rectangle with the same name in the figure, along with the schemas used to build its meaning. The construction specifies the types of constituents needed to trigger the instantiation of this construction, the type of the schema to be evoked, and constraints for mapping the meanings of its constituents to the slots in the meaning structure.

Lucia stores all its linguistic knowledge in Soar's procedural memory, thus avoiding the overhead of retrieving this knowledge from semantic memory as the earlier systems generally do.

## 1.3 Overview

This paper concentrates on the methods used for grounding language in the Lucia system. Supplemental materials describe a benchmark to enable evaluating other systems against the same sentence corpus we have used for testing and a set of gold-standard meanings for those sentences. We have found a lack of published material to compare systems for language grounding in robots, and our intention is that this benchmark can be one attempt to fill this gap.

The following sections discuss our approach to grounding, give some examples of language used for ITL, and describe the grounding processes in Lucia. Finally we describe the files contained in the benchmark, which are submitted as supplemental material with this paper.

## 2 Grounded Meanings

Human interaction with robots using natural language often needs language to be grounded to the agent's perceptions of the physical world and its knowledge of its own action capabilities.

Several projects have used the Rosie system to explore ITL. Mohan et al. (2012) discuss interactive methods for learning words that are grounded in the agent's physical environment and actions with a table-top robotic arm. Kirk and Laird (2016) report using interactive instruction to teach Rosie to understand and play new games. The application to games raises many issues with grounding language in hypothetical settings, but this paper does not consider this aspect. Mininger and Laird (2016) extend the table-top version of Rosie to one that can navigate in an indoor environment and comprehend language about objects that are unseen or unknown. These projects have contributed much to task learning, but their language comprehension systems are ad-hoc.

Lucia (Lindes and Laird, 2016; 2017a; 2017b) is a comprehension system built within the same Rosie agent and using the Soar architecture, but it also is built around the ECG theory of language. Its linguistic knowledge is written by hand in the ECG formalism (Bryant, 2008) and translated automatically into Soar production rules. Since all this knowledge is procedural and does not have to be retrieved dynamically from long-term memory, Lucia simulates skilled comprehension in simulated time close to human real-time performance. It uses a human-like incremental processing system, distinct from the best-fit over a whole sentence approach used by other ECG systems (Bryant, 2008). In what follows we look at how well Lucia succeeds in grounding language within the constraints imposed by ECG and incremental processing.

What does it mean to ground natural language in this context? The comprehension of each input sentence must produce a meaning structure in working memory that is sufficient for the agent to use its knowledge of perception and action to perform the internal and external actions the instructor intended. In ITL the interaction process may include requests from the agent for additional information or clarification.

In this paper we do not consider the details of how the agent's perception and action work. Rather we assume that before trying to comprehend a given input utterance, the agent already has knowledge about what its vision system currently perceives in the world. It also has knowledge in long-term memory about what actions it can perform. Knowledge of actions can be either built into the agent or learned through interaction. In either case, the perception and action concepts which the agent grounds to physical percepts or motor control programs are represented by internal symbols shared by the linguistic and robotic parts of the agent. Thus we are concerned here with grounding the natural language to these internal symbols and compositions of them.

In order to ground the meaning of a sentence, each linguistic unit involved must be grounded, including words, phrases, clauses, and complete sentences. To comprehend *'Pick up the green sphere.'* as shown in Figure 1, *pick up* must be grounded to an action the agent knows how to perform, *the green sphere* must be grounded to a specific object that the agent sees in its current environment, and these two meanings must be composed into a sentence-level meaning that can produce an actionable "message" which tells Rosie what action to take. Along the way the meanings of individual words like *green* and *sphere* must be grounded to the corresponding properties in the agent's long-term knowledge that are required to find the object in its perceived scene.

## 3 Language Used for Interactive Task Learning

In ITL the agent starts with sufficient linguistic and operational knowledge to perform some tasks, but then needs to learn new tasks and extensions to known tasks through interaction with a human instructor (Laird et al., 2012; Mohan et al., 2012; Mininger and Laird, 2016). In this section we give some examples of the language input involved in learning a few example tasks. Although the interaction also involves requests from Rosie to the instructor, we consider only language comprehension and not production here.

Assume that at first the agent knows the primitive manipulation command to *pick up* an object in its visual field and another to *put* or *put down* that object in one of its known locations. Now we can instruct it to learn the verb *move* with an interaction that includes the following sequence of instructions, interspersed with agent responses that are not shown.

(1) Sentences for teaching the verb *move*

a. `Move the red block to the left of the orange block.`

b. `The goal is that the red block is to the left of the orange block.`

c. `Pick up the red block.`

d. `Put the red block to the left of the orange block.`

e. `The task is done.`

In (1a) a command to perform an unknown task is given, and the agent asks for help. Then the instructor states the end goal of the task (1b). In some cases this may be sufficient, and the agent may be able to perform the reasoning needed to plan a sequence of actions to perform the task. In (1) we show a case where the agent asks for more help after (1b), and the instructor gives a sequence of known commands needed to complete the task (1c-e). Rosie can then remember the goal, its relation to the original *move* command, and the sequence of steps so that if given another *move* command in the future it can perform it unaided. Other examples of similar interactions are given in (2) through (4).

(2) Sentences for teaching the *discard* task

a. `Discard the green box.`

b. `The goal is that the green box is in the trash.`

c. `Pick up the green box.`

d. `Put the green box in the trash.`

e. `The task is finished.`

(3) Sentences for teaching a *deliver* task

a. `Deliver the box to the main office.`

b. `The goal is that the box is in the office.`

c. `Pick up the box.`

d. `Go to the main office.`

e. `Put down the box.`

f. `You are done.`

(4) Sentences for teaching a *fetch* task

a. `Fetch a stapler.`

b. `The goal is that the stapler is in the starting location.`

c. `Remember the current location as the starting location.`

d. `Find the stapler.`

e. `Pick up the stapler.`

f. `Go to the starting location.`

g. `Put down the stapler.`

h. `The task is over.`

These examples illustrate the kinds of sequences involved in ITL, but do not represent the full linguistic range of the system. The benchmark described below contains a corpus of 200 sentences that apply to the object manipulation and indoor navigation domains, as well as to learning complex tasks in these domains. These sentences have been designed by hand to accomplish three purposes: provide the information needed to achieve our ITL goals, say things in a way that seems natural to humans, and experiment with different linguistic forms.

The entire corpus includes declarative sentences to describe objects or relations, commands for object manipulation and indoor navigation, conditional *if/then* commands, commands with *until* clauses, sentences about goals and task progress, and questions to Rosie about its knowledge of the world. Unrestricted human interaction with the agent might well produce many additional linguistic forms we have not yet considered. As a group, the 200 sentences provide a number of comprehension challenges, including lexical, syntactic, and semantic ambiguities (Lindes and Laird, 2017b).

## 4 The Grounding Process in Lucia

This section examines how Lucia grounds words, phrases, clauses, and sentences, eventually producing a *message* to the operational part of Rosie for each sentence it comprehends. We describe the knowledge sources used to provide information for the grounding, give an overview of the comprehension process in Lucia, and describe the various grounding processes.

### 4.1 Knowledge Sources

Information for grounding the various linguistic units comes from four sources: the ECG grammar, the current state of the comprehension, the current perceived visual scene, and an ontology of classes, properties, and actions.

**The ECG grammar:** Lucia uses a grammar built by hand in the ECG language. An off-line program translates the constructions and schemas in the grammar into Soar production rules (Lindes and Laird, 2016). As the comprehension proceeds, these rules fire at appropriate times to instantiate constructions and schemas and fill the schema

roles whose fillers are defined in the grammar. Additional Soar rules not built from the grammar fill in grounding information from the other knowledge sources.

**The comprehension state**: The Lucia comprehension system works incrementally, doing as much processing as possible as each individual word comes in (Lindes and Laird, 2017a). In doing so it recognizes lexical and phrasal constructions and builds a hierarchy of their instances. Schemas are also instantiated and filled as soon as possible and attached to the constructions. At any point in time the comprehender has a stack of construction instances built so far that have not yet been incorporated as constituents in higher level constructions, and each of these has its attached meaning. Thus the rules that are trying to ground any new meaning being constructed can draw on the knowledge contained in this comprehension state as one of their information sources.

**The world model:** Rosie has a scene graph is brought into Soar's working memory by its visual perception system and thus is available to Lucia and to the part of Rosie that implements actions. Each object is identified by a unique identifier and has category, color, size, and shape properties set by the visual perception system. In addition to objects, the world model contains information about spatial relations between objects, an indicator of which object the instructor is currently pointing to, and a special object to represent the robot itself. This world model is a key source of knowledge for grounding language.

**The ontology:** The objects and relations in the world model are in working memory and can change as Rosie proceeds through a task. We also need fixed knowledge to represent categories, property values, and actions. The source for this kind of knowledge is an ontology stored in Soar's long-term semantic memory. This knowledge, like the world model, is shared between Lucia and the operational rules in Rosie.

## 4.2 Lucia Comprehension Overview

Lucia processes a sentence word-by-word and left-to-right. A number of Soar operators are selected and applied during each word cycle. By the end of the word cycle, the comprehension state will have the lexical construction for that word, larger phrasal constructions that combine it as appropriate with items previously on the stack, and

grounded meaning schemas corresponding to these new constructions.

Each word known to the system has one or more lexical constructions in the grammar. If a word has multiple senses, each of these constructions is instantiated at first, and later processes select the correct one for the current context (Lindes and Laird, 2017b). Phrasal constructions combine constituents into higher level constructions. As each construction is instantiated, it evokes a schema to represent its meaning. Along the way, the various forms of grounding are performed.

## 4.3 Grounding Referring Expressions

We define a *referring expression* as a linguistic unit meant to describe some object that the system can know about. The general construction for a referring expression is called a `RefExpr`, and its meaning is represented by a `RefDesc`, or *reference descriptor*. These are built up as words are being processed. A `RefExpr` can consist of a simple pronoun, like *it* or *this*, a noun phrase like *the green sphere*, or a more complex expression like *the green rectangle to the left of the large green rectangle* or *a green block that is on the stove*. As the individual words in the expression are processed, the complete `RefExpr` and its `RefDesc` meaning are gradually built up.

A common noun generates a schema that represents some class of object, and sets the roles of this schema to identifiers for the category and/or shape of that class of objects. An adjective generates a schema to describe a property class and a value for that property, such as the color *green*. A determiner sets whether the expression is *definite* or *indefinite*. At this lexical level, part of the grounding is performed by instantiating the grammatical knowledge incorporated in the lexical constructions. Then an operator is selected to retrieve information about categories and properties from the ontology.

As soon as a complete noun phrase has been built, an operator takes the `RefDesc` that has been assembled and searches in the world model for one or more objects that match the description. Similarly, a pronoun is grounded by an operator that deals with pronouns. The object that is found from this grounding is then set as the referent of the `RefDesc`.

More complex referring expressions are grounded in several steps. For instance, *the green rectangle to the left of the large green rectangle*

causes several phrase-level grounding operations in addition to the lexical ones. The phrase *the green rectangle* will be grounded to the set of two green rectangles in our test world model, while the *large green rectangle* will be grounded to a single object. The preposition *to the left of* is looked up in the ontology to see what kind of relation it represents. Finally that relation is used to select from the possible green rectangles the one which satisfies the complete expression.

A complex referring expression like *a green block that is on the stove* also requires several grounding steps. First when *a green block* is grounded, a set of several objects that match is formed. Next *that* is recognized as a relative pronoun, and connected to the preceding noun phrase. The phrase *on the stove* finds the stove and a relation to it, which is then applied to the set of objects found for the noun phrase that *that* is attached to, resulting in a single object that satisfies the whole expression.

### 4.4 Grounding and Attaching Prepositional Phrases

Consider the full sentence: *'Move the green rectangle to the left of the large green rectangle to the pantry.'* Above we looked at grounding *the green rectangle to the left of the large green rectangle* as an isolated expression. Within the complete sentence, however, things get more complicated. The prepositional phrase with *to the left of* could attach to the previous noun phrase, but it could also attach to the verb as its target location. Then later on, as the incremental processing proceeds, we get to the phrase *to the pantry*. Where should this be attached?

Lucia has a strategy for resolving issues of this sort within its incremental, single-path parsing strategy context (Lindes and Laird, 2017a). When the first prepositional phrase has been assembled, two attachment sites are considered: the immediately preceding noun phrase or the previous verb. If the verb is one that requires a target location, such as *put* or *move*, the prepositional phrase will be attached to the verb. If the verb is one like *pick up* that does not require a target location, the phrase will be attached to the preceding noun phrase.

When the second prepositional phrase has been processed, however, we have a problem. The verb already has a target location attached, and the noun phrase before that has been hidden under the construction that makes that attachment. Here Lucia uses a strategy called *local repair*. A *snip* operation disconnects the first prepositional phrase from the verb, and then it is reattached to *the green rectangle,* and that complete expression is regrounded. Now the phrase *to the pantry* can be attached to the verb as its target location. This local repair operation is described more fully by Lindes and Laird (2017b).

### 4.5 Grounding Full Sentences

The ECG grammar provides constructions that combine the lexical items for verbs with the referring expressions that form the verb's arguments to form complete sentences. These are often called argument structure constructions. Verbs describing actions are grounded by looking up their identifiers in the ontology to connect to the actions the agent knows how to perform. This lookup provides the referring link to the agent's knowledge of how to perform the given action.

Once the comprehension process has recognized a complete sentence as a single construction, an *interpretation* process is performed. This process converts the top-level meaning structure produced by the language comprehension system into a grounded, actionable *message* for Rosie to act on. Every message has a type field, plus other arguments depending on its type. Here we give brief descriptions of these messages; more detail can be found in the supplemental materials.

In the example in Figure 1, the `ActOnIt` schema is interpreted to form a message of type `command` with arguments `pick-up1` and `large-green-sphere1`.

Declarative sentences produce `object-description` messages. This message type has an object argument to indicate the object being described and a property argument showing the property to be assigned to the object. All action commands produce `command` messages. Each `command` message has an action argument, most have an object argument, and others have varying arguments.

The command *'If you see the soda then pick it up.'* illustrates a conditional command which produces a "conditional" message an action for the *then* clause and a condition for the *if* clause. The condition must be met first, and the action will be performed when the condition is met. A command can also have an *until* clause describing a condi-

tion to terminate the action, as in *'Explore until you see a soda.'*

Lucia and Rosie understand several types of questions, as shown in (5).

(5)   Some questions that Rosie can understand
a.   `Is the large sphere green?`
b.   `Is the small orange triangle behind the green sphere?`
c.   `What is inside the pantry?`
d.   `Where is the red triangle?`
e.   `What color is the large sphere?`
f.   `What shape is this?`

Such questions produce various forms of question messages, with arguments to define the objects, properties, or relations being asked about.

## 5   Benchmark

We have assembled various data items discussed in this paper into a package to be submitted as supplementary material. We hope this package, which we are calling *The University of Michigan Robot Language Benchmark #1*, will be useful to other researchers as a benchmark against which to evaluate their systems for robot language grounding as we are using it to evaluate and continue to develop Lucia.

The supplementary materials are contained in a file called UMRLB-1_v0.1.zip containing the files listed in Table 1. The "-1" indicates that we expect there to be others in the future, and the "_v0.1" indicates the specific version. The files containing data structures are in the industry standard JSON format to make them easily machine-readable across many systems.

## Acknowledgments

| File Name | Description |
|---|---|
| UMRLB-1 .pdf | A document describing in detail the files in the benchmark and their meanings. |
| Sentences.txt | The corpus of 200 sentences, grouped by their linguistic types. |
| World.json | A definition of a particular snap-shot of the world perceived by a robot that can be used to ground linguistic expressions. |
| Ontology.json | An ontology defining properties of perceived objects and robot actions. |
| GoldStandard .json | A file giving the gold-standard meaning for each sentence in the corpus, along with other metadata. |

Table 1: Files included in the Benchmark

## References

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions. *Transactions of the Association for Computational Linguistics, 1, 49-62.*

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. *EMNLP 2013: Conference on Empirical Methods in Natural Language Processing.*

Benjamin Bergen and Nancy Chang. 2013. Embodied construction grammar. In Thomas Hoffman and Graeme Trousdale, editors, *The Oxford Handbook of Construction Grammar*, Oxford University Press, Oxford, chapter 10, pages 168–190.

John Edward Bryant. 2008. *Best-fit Constructional Analysis*. Ph.D. thesis, Berkeley, CA, USA.

Morten H. Christiansen and Nick Chater. 2016. The Now-or-Never bottleneck: A fundamental constraint on language. *Behavior and Brain Science.* doi:10.1017/S0140525X1500031X.

Manfred Eppe, Sean Trott, Vivek Raghuram, Jerome Feldman, and Adam Janin. 2016. Application independent and integration-friendly natural language understanding. In Christoph Benzmüller, Geoff Sutcliffe, and Raul Rojas, editors, GCAI 2016. *2nd Global Conference on Artificial Intelligence*. EasyChair, volume 41 of EPiC Series in Computing, pages 340–352.

Jerome Feldman, Ellen Dodge, and John Bryant. 2009. Embodied construction grammar. In Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, Oxford University Press, Oxford https://doi.org/10.1093/oxfordhb/9780199544004.013.0006.

James Kirk, Aaron Mininger, and John Laird. 2016. Learning task goals interactively with visual demonstrations. *Biologically Inspired Cognitive Architectures* 18:1–8.

James R Kirk and John E Laird. 2016. Learning general and efficient representations of novel games through interactive instruction. *Adv. Cogn. Syst 4*.

John Laird, Keegan Kinkade, Shiwali Mohan, and Joseph Xu. 2012. Cognitive robotics using the soar cognitive architecture. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence, Cognitive Robotics*. https://www.aaai.org/ocs/index.php/WS/AAAIW12/paper/view/5221.

John E. Laird, K. Gluck, John Anderson, Ken Forbus, O. Jenkins, Christian Lebiere, D. D. Salvucci, Matthias Scheutz, A. Thomaz, G. Trafton, R. E. Wray, Shiwali Mohan, and James R. Kirk. 2017. Interactive Task Learning, *IEEE Intelligent Systems*, In press.

John E Laird. 2012. *The Soar cognitive architecture*. MIT Press.

Fain Lehman, Richard L Lewis, and Allen Newell. 1991. Integrating knowledge sources in language comprehension. In Proceedings of the Thirteenth *Annual Conference of the Cognitive Science Society*. pages 461–466.

Peter Lindes and John E Laird. 2016. Toward Integrating Cognitive Linguistics and Cognitive Language Processing. In *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM)*.

Peter Lindes and John Laird. 2017a. Cognitive Modeling Approaches to Language Comprehension Using Construction Grammar. In 2017 AAAI Spring Symposium Series, Computational Construction Grammar and Natural Language Understanding. https://aaai.org/ocs/index.php/SSS/SSS17/paper/view/15285.

Peter Lindes and John E Laird. 2017b. Ambiguity Resolution in a Cognitive Model of Language Comprehension. To be published in *Proceedings of the 15th International Conference on Cognitive Modeling (ICCM 2017)*. In press.

Aaron Mininger and John E Laird. 2016. Interactively Learning Strategies for Handling References to Unseen or Unknown Objects. *Adv. Cogn. Syst 4*.

Shiwali Mohan and John E Laird. 2014. Learning Goal-Oriented Hierarchical Tasks from Situated interactive instruction. In *AAAI*. pages 387–394.

Shiwali Mohan, Aaron Mininger, James Kirk, and John Laird. 2012. Acquiring Grounded Representations of Words with Situated Interactive Instruction. *Advances in Cognitive Systems* 2:113–130.

Luc Steels and Manfred Hild, editors. 2012. Language Grounding in Robots. Springer, New York. https://doi.org/10.1007/978-1-4614-3064-3.

Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. 2014. Asking for help using inverse semantics. In *Proceedings of Robotics: Science and Systems*. Berkeley, USA. https://doi.org/10.15607/RSS.2014.X.024.

Sida I. Wang, Percy Liang, and Christopher D. Manning. 2016. Learning Language Games Through Interaction. arXiv preprint arXiv:1606.02447.

Luke S. Zettlemoyer and Michael Collins. 2007. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 768-687.