Conversational Robots: Building Blocks for Grounding Word Meaning

Deb Roy
MIT Media Lab
dkroy@media.mit.edu

Kai-Yuh Hsiao
MIT Media Lab
eepness@mit.edu

Nikolaos Mavridis
MIT Media Lab
nmav@media.mit.edu

Abstract

How can we build robots that engage in fluid spoken conversations with people, moving beyond canned responses to words and towards actually understanding? As a step towards addressing this question, we introduce a robotic architecture that provides a basis for grounding word meanings. The architecture provides perceptual, procedural, and affordance representations for grounding words. A perceptually-coupled on-line simulator enables sensory-motor representations that can shift points of view. Held together, we show that this architecture provides a rich set of data structures and procedures that provide the foundations for grounding the meaning of certain classes of words.

1 Introduction

Language enables people to talk about the world, past, present, and future, real and imagined. For a robot to do the same, it must ground language in its world as mediated by its perceptual, motor, and cognitive capacities. Many words that refer to entities in the world can be grounded through sensory-motor associations. For instance, the meaning of *ball* includes perceptual associations that encode how balls look and predictive models of how balls behave. The representation of *touch* must include procedural associations that encode how to perform the action, and perceptual encodings to recognize the action in others. In this view, words serve as labels for perceptual or action concepts. When a word is uttered, the underlying concept is communicated since the speaker and listener maintain similar associations. This basic approach underlies most work to date in building machines that ground language (Bailey, 1997; Narayanan, 1997; Regier and Carlson, 2001; Roy and Pentland, 2002; Siskind, 2001; Lammens, 1994; Steels, 2001).

Not all words, however, can be grounded in terms of perceptual and procedural representations, even when used in concrete situations. In fact, in even the simplest conversations about everyday objects, events, and relations, we run into problems. Consider a person and a robot sitting across a table from each other, engaged in coordinated activity involving manipulation of objects. After some interaction, the person says to the robot:

Touch the heavy blue thing that was on my left.

To understand and act on this command in context, consider the range of knowledge representations that the robot must bind words of this utterance to. *Touch* can be grounded in a visually-guided motor program that enables the robot to move towards and touch objects. This is an example of a procedural association which also critically depends on perception to guide the action. *Heavy* specifies a property of objects which involves *affordances* that intertwine procedural representations with perceptual expectations (Gibson, 1979). *Blue* and *left* specify visual properties. *Thing* must be grounded in terms of both perception and affordances (one can see an object, and expect to reach out and touch it). *Was* triggers a reference to the past. *My* triggers a shift of perspective in space.

We have developed an architecture in which a physical robot is coupled with a physical simulator to provide the basis for grounding each of these classes of lexical semantics[1]. This workshop paper provides an abbreviated

[1]We acknowledge that the words in this example, like most words, have numerous additional connotations that are not captured by the representations that we have suggested. For example, words such as *touch*, *heavy* and *blue* can be used metaphorically to refer to emotional actions and states. *Things* are not always physical perceivable objects, *my* usually indicates possession, and so forth. Barwise and Perry use the phrase "efficiency of language" to highlight the situation-dependent reusability of words and utterances (Barwise and Perry, 1983). However, for

version of a forthcoming paper (Roy et al., forthcoming 2003).

The robot, called Ripley, is driven by compliant actuators and is able to manipulate small objects. Ripley has cameras, touch, and various other sensors on its "head". Force sensors in each actuated joint combined with position sensors provide the robot with a sense of proprioception. Ripley's visual and proprioceptive systems drive a physical simulator that keeps a constructed version of the world (that includes Ripley's own physical body) in synchronization with Ripley's noisy perceptual input. An object permanence module determines when to instantiate and destroy objects in the model based on perceptual evidence. Once instantiated, perception can continue to influence the properties of an object in the model, but knowledge of physical world dynamics is built into the simulator and counteracts 'unreasonable' percepts.

Language is grounded in terms of associations with elements of this perceptually driven world model, as well as direct groundings in terms of sensory and motor representations. Although the world model directly reflects reality, the state of the model is the result of an interpretation process that compiles perceptual input into a stable registration of the environment. As opposed to direct perception, the world model affords the ability to assume arbitrary points of view through the use of synthetic vision which operates within the physical model, enabling a limited form of "out of body experience". This ability is essential to successfully differentiate the semantics of *my left* versus *your left*. Non-linguistic cues such as the visual location of the communication partners can be integrated with linguistic input to context-appropriate perspective shifts. Shifts of perspective in time and space may be thought of as *semantic modulation functions*. Although the meaning of "left" in one sense remains constant across usages, the words "my" and "your" modulate the meaning by swapping frames of reference. We suspect that successful use of language requires constant modulations of meanings of this and related kinds.

We describe the robot and simulator, and mechanisms for real-time coupling. We then discuss mechanisms within this architecture designed for the purposes of grounding the semantics of situated, natural spoken conversation. Although no language understanding system has yet been constructed, we conclude by sketching how the semantics of each of the words and the whole utterance discussed above can be grounded in the data structures and processes provided by this architecture. This work represents steps towards our long term goal of developing robots and other machines that use language in human-like ways by leveraging deep, grounded representations of meaning that "hook" into the world through machine perception, action, and higher layers of cognitive processes. The work has theoretical implications on how language is represented and processed by machine, and also has practical applications where natural human-robot interaction is needed such as deep-sea robot control, remote handling of hazardous materials by robots, and astronaut-robot communication in space.

the utterance and context that we have described, the groundings listed above play essential roles. It may be argued that other senses of words are often metaphoric extensions of these embodied representations (Lakoff and Johnson, 1980).

2 Background

Although robots, speech recognizers, and speech synthesizers can easily be connected in shallow ways, the results are limited to canned behavior. The proper integration of language in a robot highlights deep theoretical issues that touch on virtually all aspects of artificial intelligence (and cognitive science) including perception, action, memory, and planning. Along with other researchers, we use the term *grounding* to refer to problem of anchoring the meaning of words and utterances in terms of non-linguistic representations that the language user comes to know through some combination of evolutionary and lifetime learning.

A natural approach is to connect words to perceptual classifiers so that the appearance of an object, event, or relation in the environment can instantiate a corresponding word in the robot. This basic idea has been applied in many speech-controlled robots over the years (Brown et al., 1992; McGuire et al., 2002; Crangle and Suppes, 1994).

Detailed models have been suggested for sensory-motor representations underlying color (Lammens, 1994), spatial relations (Regier, 1996; Regier and Carlson, 2001). Models for grounding verbs include grounding verb meanings in the perception of actions (Siskind, 2001), and grounding in terms of motor control programs (Bailey, 1997; Narayanan, 1997). Object shape is clearly important when connection language to the world, but remains a challenging problem in computational models of language grounding. Landau and Jackendoff provide a detailed analysis of additional visual shape features that play a role in language (Landau and Jackendoff, 1993).

In natural conversation, people speak and gesture to coordinate *joint actions* (Clark, 1996). Speakers and listeners use various aspects of their physical environment to encode and decode utterance meanings. Communication partners are aware of each other's gestures and foci of attention and integrate these source of information into the conversational process. Motivated by these factors, recent work on social robots have explored mechanisms that provide visual awareness of human partners' gaze and other facial cues relevant for interaction (Breazeal, 2003; Scassellati, 2002).

3 Ripley: An Interactive Robot

Ripley was designed specifically for the purposes of exploring questions of grounded language, and interactive language acquisition. The robot has a range of motions that enables him to move objects around on a tabletop placed in front of him. Ripley can also look up and make "eye contact" with a human partner. Three primary considerations drove the design of the robot: (1) We are interested in the effects of changes of visual perspective and their effects on language and conversation, (2) Sensorymotor grounding of verbs. (3) Human-directed training of motion. For example, to teach Ripley the meaning of "touch", we use "show-and-tell" training in which exemplars of the word (in this case, motor actions) can be presented by a human trainer in tandem with verbal descriptions of the action.

To address the first consideration, Ripley has cameras placed on its head so that all motions of the body lead to changes of view point. This design decision leads to challenges in maintaining stable perspectives in a scene, but reflect the type of corrections that people must also constantly perform. To support acquisition of verbs, Ripley has been designed with a "mouth" that can grasp objects and enable manipulation. As a result, the most natural class of verbs that Ripley will learn involve manual actions such as touching, lifting, pushing, and giving. To address the third consideration, Ripley is actuated with compliant joints, and has "training handles". In spite of the fact that the resulting robot resembles an arm more than a torso, it nonetheless serves our purposes as a vehicle for experiments in situated, embodied, conversation. In contrast, many humanoid robots are not actually able to move their torso's to a sufficient degree to obtain significant variance in visual perspectives, and grasping is often not achieved in these robots due to additional complexities of control. This section provides a description of Ripley's hardware and low level sensory processing and motor control software layers.

3.1 Mechanical Structure and Actuation

The robot is essentially an actuated arm, but since cameras and other sensors are placed on the gripper, and the robot is able to make "eye contact", we often think of the gripper as the robot's head. The robot has seven degrees of freedom (DOF's) including a 2-DOF shoulder, 1-DOF elbow, 3-DOF wrist / neck, and 1-DOF gripper / mouth. Each DOF other than the gripper is actuated by series-elastic actuators (Pratt et al., 2002) in which all force from electric motors are transferred through torsion springs. Compression sensors are placed on each spring and used for force feedback to the low level motion controller. The use of series-elastic actuators gives Ripley the ability to precisely sense the amount of force that is being applied at each DOF, and leads to compliant motions.

3.2 Motion Control

A position-derivative control loop is used to track target points that are sequenced to transit smoothly from the starting point of a motion gesture to the end point. Natural motion trajectories are learned from human teachers through manual demonstrations.

The robot's motion is controlled in a layered fashion. The lowest level is implemented in hardware and consists of a continuous control loop between motor amplifiers and force sensors of each DOF. At the next level of control, a microcontroller implements a position-derivative (PD) control loop with a 5 millisecond cycle time. The microcontroller accepts target positions from a master controller and translates these targets into force commands via the PD control loop. The resulting force commands are sent down stream to the motor amplifier control loop. The same force commands are also sent *up stream* back to the master controller, serving as dynamic proprioceptive force information

To train motion trajectories, the robot is put in a gravity canceling motor control mode in which forces due to gravity are estimated based on the robot's joint positions and counteracted through actuation. While in this mode, a human trainer can directly move the robot through desired motion trajectories. Motion trajectories are recorded during training. During playback, motion trajectories can be interrupted and smoothly revised to follow new trajectories as determined by higher level control. We have also implemented interpolative algorithms that blend trajectories to produce new motions that beyond the training set.

3.3 Sensory System and Visual Processing

Ripley's perceptual system is based on several kinds of sensors. Two color video cameras, a three-axis tilt accelerometer (for sensing gravity), and two microphones are mounted in the head. Force sensitive resistors provide a sense of touch on the inside and outside surfaces of the gripper fingers. In the work reported here, we make use of only the visual, touch, and force sensors. The remaining sensors will be integrated in the future. The microphones have been used to achieve sound source localization and will play a role in maintaining "eye contact" with communication partners. The accelerometer will be used to help correct frames of reference of visual input.

Complementing the motor system is the robot's sensor system. One of the most important sets of sensors is the actuator set itself; as discussed, the actuators are force-controlled, which means that the control loop adjusts the force that is output by each actuator. This in turn means that the amount of force being applied at each joint is known. Additionally, each DOF is equipped with abso-

lute position sensors that are used for all levels of motion control and for maintaining the zero-gravity mode.

The vision system is responsible for detecting objects in the robot's field of view. A mixture of Gaussians is used to model the background color and provides foreground/background classification. Connected regions with uniform color are extracted from the foreground regions. The three-dimensional shape of an object is represented using histograms of local geometric features, each of which represents the silhouette of the object from a different viewpoint. Three dimension shapes are represented in a view-based approach using sets of histograms. The color of regions is represented using histograms of illumination-normalized RGB values. Details of the shape and color representations can be found in (Roy et al., 1999).

To enable grounding of spatial terms such as "above" and "left", a set of spatial relations similar to (Regier, 1996) is measured between pair of objects. The first feature is the angle (relative to the horizon) of the line connecting the centers of area of an object pair. The second feature is the shortest distance between the edges of the objects. The third spatial feature measures the angle of the line which connects the two most proximal points of the objects.

The representations of shape, color, and spatial relations described above can also be generated from virtual scenes based on Ripley's mental model as described below. Thus, the visual features can serve as a means to ground words in either real time camera grounded vision or simulated synthetic vision.

3.4 Visually-Guided Reaching

Ripley can reach out and touch objects by interpolating between recorded motion trajectories. A set of sample trajectories are trained by placing objects on the tabletop, placing Ripley in a canonical position so that the table is in view, and then manually guiding the robot until it touches the object. A motion trajectory library is collected in this way, with each trajectory indexed by the position of the visual target. To reach an object in an arbitrary position, a linear interpolation between trajectories is computed.

3.5 Encoding Environmental Affordances: Object Weight and Compliance

Words such as "heavy" and "soft" refer to properties of objects that cannot be passively perceived, but require interaction with the object. Following Gibson (Gibson, 1979), we refer to such properties of objects as affordances. The word comes from considerations of what an object affords to an agent who interacts with it. For instance, a light object can be lifted with ease as opposed to a heavy object. To assess the weight of an unknown object, an agent must actually lift (or at least attempt to lift) it and gauge the level of effort required. This is precisely how Ripley perceives weight. When an object is placed in Ripley's mouth, a motor routine is initiated which tightly grasps the object and then lifts and lowers the object three times. While the motor program is running, the forces experienced in each DOF (Section 3.2) are monitored. In initial word learning experiments, Ripley is handed objects of various masses and provided word labels. A simple Bayes classifier was trained to distinguish the semantics of "very light", "light", "heavy", and "very heavy". In a similar vein, we also grounded the semantics of "hard" and "soft" in terms of grasping motor routines that monitor pressure changes at each fingertip as a function of grip displacement.

4 A Perceptually-Driven "Mental Model"

Ripley integrates real-time information from its visual and proprioceptive systems to construct an "internal replica", or mental model of its environment that best explains the history of sensory data that Ripley has observed[2]. The mental model is built upon the ODE rigid body dynamics simulator (Smith, 2003). ODE provides facilities for modeling the dynamics of three dimensional rigid objects based on Newtonian physics. As Ripley's physical environment (which includes Ripley's own body) changes, perception of these changes drive the creation, updating, and destruction of objects in the mental model. Although simulators are typically used *in place of* physical systems, we found physical simulation to be an ideal substrate for implementing Ripley's mental model (for coupled on-line simulation, see also (Cao and Shepherd, 1989; Davis, 1998; Surdu, 2000)).

The mental model mediates between perception of the objective world on one hand, and the semantics of language on the other. Although the mental model reflects the objective environment, it is biased as a result of a projection through Ripley's particular sensory complex. The following sections describe the simulator, and algorithms for real-time coupling to Ripley's visual and proprioceptive systems.

The ODE simulator provides an interface for creating and destroying rigid objects with arbitrary polyhedron geometries placed within a 3D virtual world. Client programs can apply forces to objects and update their properties during simulation. ODE computes basic Newtonian updates of object positions at discrete time steps based object masses and applied forces. Objects in ODE are currently restricted to two classes. Objects in Ripley's workspace (the tabletop) are constrained to be spheres of fixed size. Ripley's body is modeled within the simula-

[2]Mental models have been proposed as a central mechanism in a broad range of cognitive capacities (Johnson-Laird, 1983).

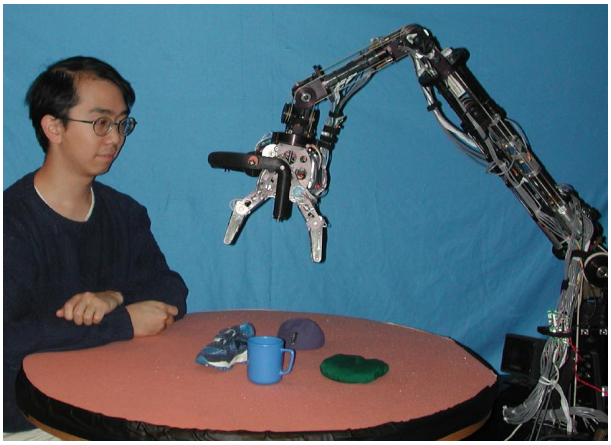

Figure 1: Ripley looks down at objects on a tabletop.

tor as a configuration of seven connected cylindrical links terminated with a rectangular head that approximate the dimensions and mass of the physical robot. We introduce the following notation in order to describe the simulator and its interaction with Ripley's perceptual systems.

4.1 Coupling Perception to the Mental Model

An approximate physical model of Ripley's body is built into the simulator. The position sensors from the 7 DOFs are used to drive a PD control loop that controls the joint forces applied to the simulated robot. As a result, motions of the actual robot are followed by dampened motions of the simulated robot.

A primary motivation for introducing the mental model is to register, stabilize, and track visually observed objects in Ripley's environment. An object permanence module, called the *Objecter*, has been developed as a bridge between raw visual analysis and the physical simulator. When a visual region is found to stably exist for a sustained period of time, an object is instantiated by the Objecter in the ODE physical simulator. It is only at this point that Ripley becomes "aware" of the object and is able to talk about it. Once objects are instantiated in the mental model, they are never destroyed. If Ripley looks away from an object such that the object moves out of view, a representation of the object persists in the mental model. Figure 1 shows an example of Ripley looking over the workspace with four objects in view. In Figure 2, the left image shows the output from Ripley's head-mounted camera, and the right image shows corresponding simulated objects which have been registered and are being tracked.

The Objecter consists of two interconnected components. The first component, the *2D-Objecter*, tracks two-dimension visual regions generated by the vision system. The 2D-Objecter also implements a hysteresis function which detects visual regions that persist over time.

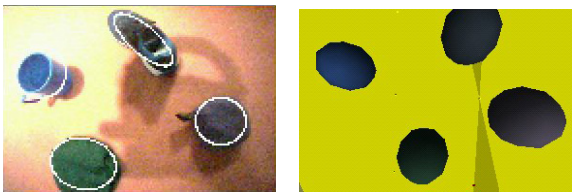

Figure 2: Visual regions and corresponding simulated objects in Ripley's mental model corresponding to the view from Figure 1

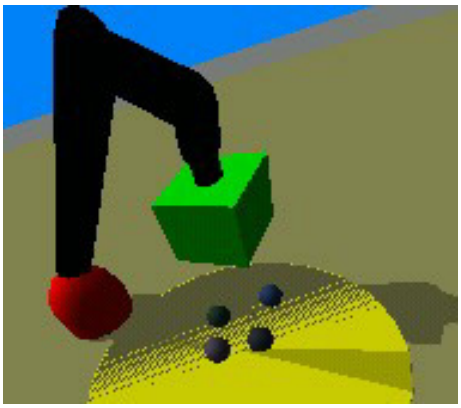

Figure 3: By positioning a synthetic camera at the position approximating the human's viewpoint, Ripley is able to "visualize" the scene from the person's point of view which includes a partial view of Ripley.

The second component, the *3D-Objecter*, takes as input persistent visual regions from the 2D-Objecter, which are brought into correspondence with a full three dimensional physical model which is held by ODE. The 3D-Objecter performs projective geometry calculations to approximate the position of objects in 3D based on 2D region locations combined with the position of the source video camera (i.e., the position of Ripley's head). Each time Ripley moves (and thus changes his vantage point), the hysteresis functions in the 2D-Objecter are reset, and after some delay, persistent regions are detected and sent to the 3D-Objecter. No updates to the mental model are performed while Ripley is in motion. The key problem in both the 2D- and 3D-Objecter is to maintain correspondence across time so that objects are tracked and persist in spite of perceptual gaps. Details of the Objecter will be described in (Roy et al., forthcoming 2003).

4.2 Synthetic Vision and Imagined Changes of Perspective

The ODE simulator is integrated with the OpenGL 3D graphics environment. Within OpenGL, a 3D environment may be rendered from an arbitrary viewpoint by positioning and orienting a synthetic camera and render-

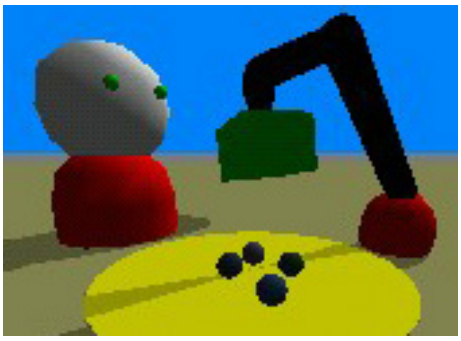

Figure 4: Using virtual shifts of perspective, arbitrary vantage points may be taken. The (fixed) location of the human partner is indicated by the figure on the left.

ing the scene from the camera's perspective. We take advantage of this OpenGL functionality to implement shifts in perspective without physically moving Ripley's pose. For example, to view the workspace from the human partner's point of view, the synthetic camera is simply moved to the approximate position of the person's head (which is currently a fixed location). Continuing our example, Figures 3 and 4 show examples of two synthetic views of the situation from Figures 1 and 2. The visual analysis features described in Section 3.3 can be applied to the images generated by synthetic vision.

4.3 Event-Based Memory

A simple form of run length encoding is used to compactly represent mental model histories. Each time an object changes a properties more than a set threshold using a distance measure that combines color, size, and location disparities, an event is detected in the mental model. Thus stretches of time in which nothing in the environment changes are represented by a single frame of data and a duration parameter with a relatively large value. When an object changes properties, such as its position, an event is recorded that only retains the begin and end point of the trajectory but discards the actual path followed during the motion. As a result, references to the past are discretized in time along event boundaries. There are many limitations to this approach to memory, but as we shall see, it may nonetheless be useful in grounding past tense references in natural language.

5 Putting the Pieces Together

We began by asking how a robot might ground the meaning of the utterance, "Touch the heavy blue thing that was on my left". We are now able to sketch an answer to this question. Ripley's perceptual system, and motor control system, and mental model each contribute elements for grounding the meaning of this utterance. In this section, we informally show how the various components of the architecture provide a basis for language grounding.

The semantic grounding of each word in our example utterance is presented using algorithmic descriptions reminiscent of the procedural semantics developed by Winograd (Winograd, 1973) and Miller & Johnson (Miller and Johnson-Laird, 1976). To begin with a simple case, the word "blue" is a property that may be defined as:

property Blue(x) {
  c ← GetColorModel(x)
  return $f_{blue}(c)$
}

The function returns a scalar value that indicates how strongly the color of object x matches the expected color model encoded in f_{blue}. The color model would be encoded using the color histograms and histogram comparison methods described in Section 3.3. The function $GetColorModel()$ would retrieve the color model of x from memory, and if not found, call on motor procedures to look at x and construct a model.

"Touch" can be grounded in the perceptually-guided motor procedure described in Section 3.4. This reaching gesture terminates successfully when the touch sensors are activated and the visual system reports that the target x remains in view:

procedure Touch(x) {
  repeat
    Reach-towards(x)
  until touch sensor(s) activated
  if x in view **then**
    return success
  else
    return failure
  end if
}

Along similar lines, it is also useful to define a weigh procedure (which has been implemented as described in Section 3.5):

procedure Weigh(x) {
  Grasp(x)
  resistance ← 0
  while Lift(x) **do**
    resistance ← resistance + joint forces
  end while
  return resistance
}

$Weigh()$ monitors the forces on the robot's joints as it lifts x. The accumulated forces are returned by the function. This weighing procedure provides the basis for

grounding "heavy":

property Heavy(x) {
  w ← GetWeight(x)
  return $f_{heavy}(w)$
}

Similar to $GetColorModel()$, $GetWeight()$ would first check if the weight of x is already known, and if not, then it would optionally call $Weigh()$ to determine the weight.

To define "the", "my", and "was", it is useful to introduce a data structure that encodes contextual factors that are salient during language use:

structure Context {
  Point-of-view
  Working memory
}

The point of view encodes the assumed perspective for interpreting spatial language. The contents of working memory would include, by default, objects currently in the workspace and thus instantiated in Ripley's mental model of the workspace. However, past tense markers such as "was" can serve as triggers for loading salient elements of Ripley's event-based memory into the working model. To highlight its effect on the context data structure, $Was()$ is defined as a *context-shift* function:

context-shift Was(context) {
  Working memory ← Salient events from mental model history
}

"Was" triggers a request from memory (Section 4.3) for objects which are added to working memory, making them accessible to other processes. The determiner "the" indicates the selection of a single referent from working memory:

determiner The(context) {
  Select most salient element from working memory
}

In the example, the semantics of "my" can be grounded in the synthetic visual perspective shift operation described in Section 4.2:

context-shift My(context) {
  context.point-of-view ← GetPointOfView(speaker)
}

Where $GetPointOfView(speaker)$ obtains the spatial position and orientation of the speaker's visual input.

"Left" is also grounded in a visual property model which computes a geometric spatial function (Section 3.3) relative to the assumed point of view:

property Left(x, context) {
  trajector ← GetPosition(x)
  return $f_{left}(trajector, context.point-of-view)$
}

$GetPosition()$, like $GetColorModel()$ would use the least effortful means for obtaining the position of x. The function f_{left} evaluates how well the position of x fits a spatial model relative to the point of view determined from $context$.

"Thing" can be grounded as:

object Thing(x) {
  if (IsTouchable(x) and IsViewable(x)) return true;
  else return false
}

This grounding makes explicit use of two affordances of a thing, that it be touchable and viewable. Touchability would be grounded using $Touch()$ and viewability based on whether x has appeared in the mental model (which is constructed based on visual perception).

The final step in interpreting the utterance is to compose the semantics of the individual words in order to derive the semantics of the whole utterance. We address the problem of grounded semantic composition in detail elsewhere (Gorniak and Roy, protectforthcoming 2003). For current purposes, we assume that a syntactic parser is able to parse the utterance and translate it into a nested set of function calls:

Touch(The(Left(My(Heavy(Blue(Thing(Was(context))))))))

The innermost argument, $context$, includes the assumed point of view and contents of working memory. Each nested function modifies the contents of $context$ by either shifting points of view, loading new contents into working memory, or sorting / highlighting contents of working memory. The $Touch()$ procedure finally acts on the specified argument.

This concludes our sketch of how we envision the implemented robotic architecture would be used to ground the semantics of the sample sentence. Clearly many important details have been left out of the discussion. Our intent here is to convey only an overall gist of how language would be coupled to Ripley. Our current work is focused on the realization of this approach using spoken language input.

References

D. Bailey. 1997. *When push comes to shove: A computational model of the role of motor control in the acquisition of action verbs*. Ph.D. thesis, Computer science division, EECS Department, University of California at Berkeley.

Jon Barwise and John Perry. 1983. *Situations and Attitudes*. MIT-Bradford.

Cynthia Breazeal. 2003. Towards sociable robots. *Robotics and Autonomous Systems*, 42(3-4).

Michael K. Brown, Bruce M. Buntschuh, and Jay G. Wilpon. 1992. SAM: A perceptive spoken language understanding robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 22 . IEEE Transactions 22:1390–1402.

F. Cao and B. Shepherd. 1989. Mimic: a robot planning environment integrating real and simulated worlds. In *IEEE International Symposium on Intelligent Control*, page 459464.

Herbert Clark. 1996. *Using Language*. Cambridge University Press.

C. Crangle and P. Suppes. 1994. *Language and Learning for Robots*. CSLI Publications, Stanford, CA.

W. J. Davis. 1998. On-line simulation: Need and evolving research requirements. In J. Banks, editor, *Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice*. Wiley.

James J. Gibson. 1979. *The Ecological Approach to Visual Perception*. Erlbaum.

Peter Gorniak and Deb Roy. forthcoming, 2003. Grounded semantic composition for visual scenes.

P.N. Johnson-Laird. 1983. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge University Press.

George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.

Johan M. Lammens. 1994. *A computational model of color perception and color naming*. Ph.D. thesis, State University of New York.

Barbara Landau and Ray Jackendoff. 1993. "What" and "where" in spatial language and spatial cognition. *Behavioral and Brain Sciences*, 16:217–265.

P. McGuire, J. Fritsch, J.J. Steil, F. Roethling, G.A. Fink, S. Wachsmuth, G. Sagerer, and H. Ritter. 2002. Multimodal human-machine communication for instructing robot grasping tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

George Miller and Philip Johnson-Laird. 1976. *Language and Perception*. Harvard University Press.

Srinivas Narayanan. 1997. *KARMA: Knowledge-based active representations for metaphor and aspect*. Ph.D. thesis, University of California Berkeley.

J. Pratt, B. Krupp B, and C. Morse. 2002. Series elastic actuators for high fidelity force control. *Industrial Robot*, 29(3):234–241.

T. Regier and L. Carlson. 2001. Grounding spatial language in perception: An empirical and computational investigation. *Journal of Experimental Psychology*, 130(2):273–298.

Terry Regier. 1996. *The human semantic potential*. MIT Press, Cambridge, MA.

Deb Roy and Alex Pentland. 2002. Learning words from sights and sounds: A computational model. *Cognitive Science*, 26(1):113–146.

Deb Roy, Bernt Schiele, and Alex Pentland. 1999. Learning audio-visual associations from sensory input. In *Proceedings of the International Conference of Computer Vision Workshop on the Integration of Speech and Image Understanding*, Corfu, Greece.

Deb Roy, Kai-Yuh Hsiao, and Nick Mavridis. forthcoming, 2003. Coupling robot perception and on-line simulation: Towards grounding conversational semantics.

Brian Scassellati. 2002. Theory of mind for a humanoid robot. *Autonomous Robots*, 12:13–24.

Jeffrey Siskind. 2001. Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic. *Journal of Artificial Intelligence Research*, 15:31–90.

R Smith. 2003. ODE: Open dynamics engine.

Luc Steels. 2001. Language games for autonomous robots. *IEEE Intelligent Systems*, 16(5):16–22.

John R. Surdu. 2000. *Connecting simulation to the mission operational environment*. Ph.D. thesis, Texas A&M.

T. Winograd, 1973. *A Process model of Language Understanding*, pages 152–186. Freeman.