Modeling Embodied Lexical Development

David Bailey, Jerome Feldman, Srini Narayanan and George Lakoff
{dbailey, jfeldman, snarayan, lakoff}@icsi.berkeley.edu
International Computer Science Institute and University of California, Berkeley
1947 Center Street Suite 600, Berkeley CA 94704 USA

Abstract

This paper presents an implemented computational model of lexical development for the case of action verbs. A simulated agent is trained by an informant giving labels to the agent's actions (here hand motions) and the system learns to both label and carry out similar actions. Computationally, the system employs a novel form of active representation and is explicitly intended to be neurally plausible. The learning methodology is a version of Bayesian model merging (Omohundro, 1992). The verb learning model is placed in the broader context of the L_0 project on embodied natural language and its acquisition.

Introduction

One of the central questions of cognitive science is how concepts and language are embodied. This involves seeking answers to the following kinds of questions about the brain: How can a neural system represent concepts? Learn concepts? Organize concepts lexically? Learn lexical items?

One way of making progress at present on these ultimate questions is to ask them of structured connectionist systems, rather than the physical neural systems of the brain, in the hope that the connectionist results will give us computational insight to the how the brain captures concepts and language. The L_0 group at ICSI and UC Berkeley has been pursuing this research strategy for almost a decade (Feldman et al., 1996). This paper is an overview of our recent results and challenges.

From our perspective, modeling lexical development is an ideal task for studying embodied cognition. We can isolate linguistically and conceptually simple situations and construct and test detailed models. Our first major effort was the dissertation work of Regier (1996) which used a simplified but realistic connectionist model of the visual system plus a conventional back-propagation learning scheme in a demonstration of how some lexical items describing spatial relations might develop in different languages. Since languages differ radically in how spatial relations are conceptualized, there was no obvious set of primitive features that could be built into the program. The key to Regier's success came directly from embodiment: all people have the same visual system and visual concepts must all arise from its capabilities. By building in a simple but realistic visual system model, Regier was able to have his program learn spatial terms from labeled example movies for a wide range of languages, using simple back-propagation techniques.

Lexical development is an active field with its own methods and results. One critical empirical finding is that the child's first words label not only things, but also relationships, actions and internal states (Tomasello, 1995). Only by addressing the broader lexical acquisition problem can one appreciate the role of embodiment. In this paper, we will focus on the learning of verbs describing simple hand motions such as *push* and *yank*. Bailey's dissertation work includes a computational model that can learn to produce verb labels for actions and also carry out actions specified by verbs that it has learned.[1] This entails a methodological problem; to model a child learning to label his own actions, the program must be able to act. Bailey solves this by incorporating the *Jack* (Badler et al., 1993) human simulation system as part of his model.

A shortcoming of the standard view of lexical acquisition is that it provides no account of how a child learns to *make use* of the concepts she learns and the words that label them. This same weakness appears as a technical consequence of using back-propagation in Regier's work and in PDP models: even when the network learns perfectly how to classify a domain, it has no mechanism for inference or action. In our new work, there is a requirement that learning algorithms produce usable representations. Figure 1 outlines what this entails for Bailey's verb learning system. The conventional concept learning task is depicted by the upward arrows. After training, when the agent executes a motor action (bottom) it tries to relate the features of the action and the world state (middle layer) to a characterization of word meanings (top layer) to produce an appropriate label. But we impose an additional constraint on the process: what is learned as a verb meaning must also function as a command interface for the agent as depicted by the downward arrows of Figure 1. For example, if it learns that *shove* labels pushing actions with high acceleration, it must be able to carry out a shoving action when asked. This is done by interpreting a verbal command as choosing a motor action and its parameters in accordance with

[1] The project is reminiscent of Winograd's (1973) SHRDLU, although it differs in its focus on learning, its restriction to verbs, and its attention to finer semantic distinctions. Siskind (1992) has considered action verb learning but does not leverage internal state (e.g. intentions, motor commands), relying instead on visual features alone.

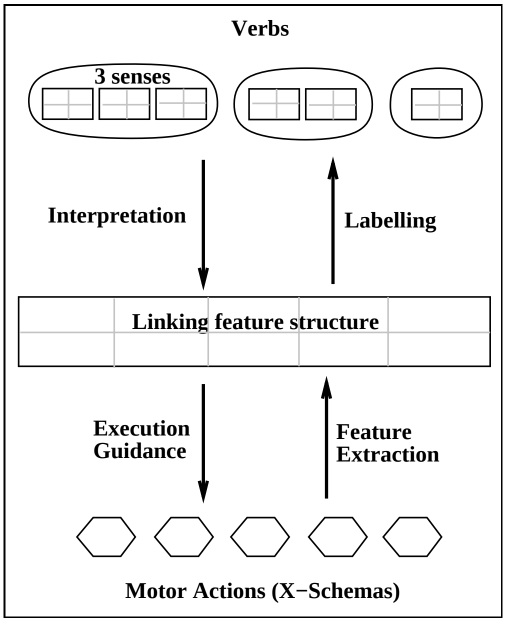

Figure 1: Architecture of Bailey's verb learning system.

the world state, and then using them to guide execution.

Levels of Representation

The basic questions of neural and cognitive development are receiving increasing attention from the connectionist perspective. It is universally assumed that genetic and environmental factors interact in elaborate ways in the acquisition of lexical terms. Although there is a good deal of theoretical and modeling work on the acquisition of syntax (including the past-tense controversy), there does not appear to be any detailed theory of lexical development like the one presented here. We hope that we have made some progress toward supplying a scientific notation for expressing mechanisms of lexical acquisition.

Any such formalism needs to bridge the gap from embodied experience to its expression as abstract symbols in language. This inherently involves multiple levels of description. Regier's work combined three levels of scientific discourse, the lowest of which (neural) was implicit. For the more complicated domains discussed here, we have added a "computational" level above the connectionist level. This is analogous to Marr's computational level and comprises a mixture of familiar notions like feature structures and a novel representation—executing schemas. The four levels of discourse are:

cognitive:	words, concepts
computational:	f-structs, x-schemas (see below)
connectionist:	structured models, learning rules
neural:	[still implicit]

The intermediate representation levels provide a much-needed scientific language for specifying proposed structures and mechanisms. We also require that intermediate representations be implementable as computational simulations to allow us to test the consequences of our hypotheses. A third requirement is that the computational mechanisms be reducible to structured connectionist models so the embodiment can be realized. There is also a fourth requirement: the representational formalisms must also support computational learning algorithms so that we can perform experiments on acquisition.

Executing Schemas

The most novel aspect of our current effort is the extensive use of *executing schemas* (*x-schemas* for short) to represent actions, so named to distinguish them from other notions of schema and to remind us that they are intended to really execute when invoked. One motivation for x-schemas comes from motor control. Simple behaviors like grasping involve coordinated movement of a range of muscles in a stereotyped yet parameterized manner called a *synergy* (Bernstein, 1967). At a higher level, motor control involves the coordination of such synergies and the accompanying perception and control. The concept of motor-schema, which pervades the literature on motor control, is a flow-chart like depiction of such activity patterns (e.g. Arbib (1987)).

We currently represent x-schemas using a formalism known as Petri nets in computer science (Murata, 1989). Their most important features are clean ways to capture concurrency and event-based asynchronous control in addition to the standard ideas of sequence, hierarchy and parameterization. A Petri net is a bipartite graph containing *places* (drawn as circles) and *transitions* (rectangles). Places hold *tokens* and represent predicates about the world state or internal state. Transitions are the active component. When all of the places pointing into a transition contain an adequate number of tokens (usually 1) the transition is enabled and may fire, removing its input tokens and depositing a new set of tokens in its output places. Generally, as a side effect a firing transition triggers an external action (e.g. a motor synergy) although this is optional. From these simple constructs, a wide variety of control structures can be built.

The bottom third of Figure 2 depicts an example x-schema for sliding an object on a tabletop. The Slide x-schema captures the fact that people shape the hand while moving the arm to an object and that large and small objects are handled differently. It includes a loop that continues motion when not yet at the goal and a separate little schema for tightening the grip if slip is detected.

Feature Structures

To keep things minimal, our models use only one other computational mechanism—*feature structures* (*f-structs* for short, drawn a a row of double-boxes). F-structs are used for static knowledge representation, parameter setting, and binding. They have been chosen to be compatible with what have been called "f-structures" in the literature on unification grammars, and are similar to well-known AI slot-filler mechanisms. Our version is

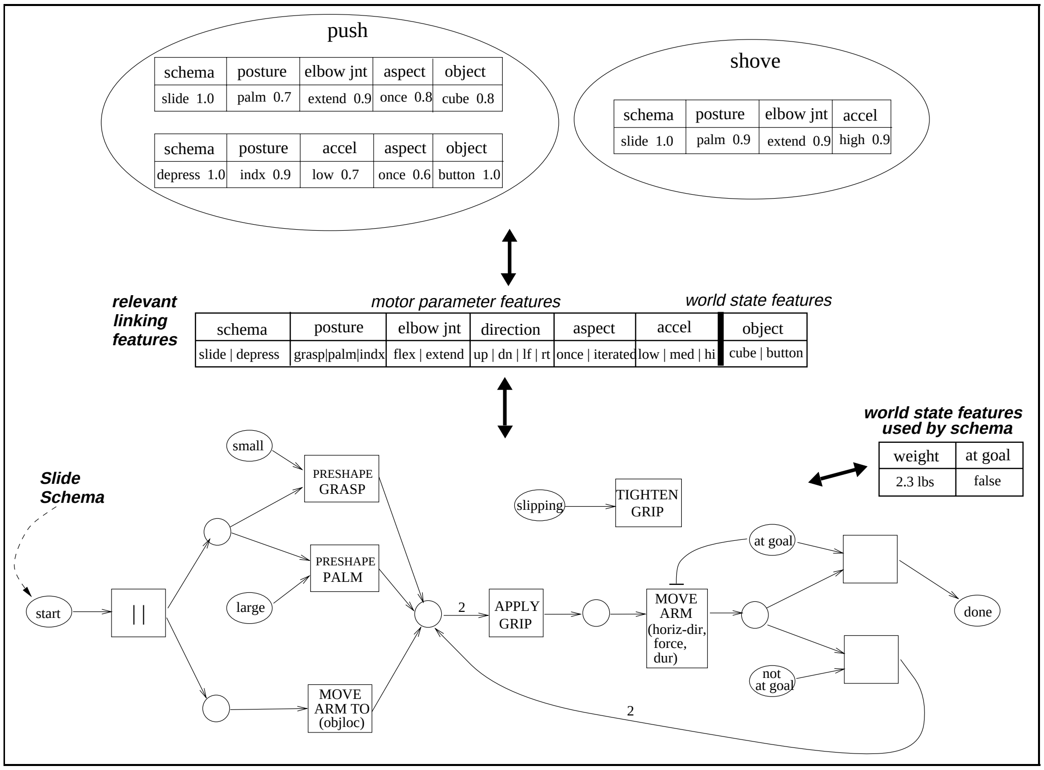
push
schema | posture | elbow jnt | aspect | object
slide 1.0 | palm 0.7 | extend 0.9 | once 0.8 | cube 0.8
schema | posture | accel | aspect | object
depress 1.0 | indx 0.9 | low 0.7 | once 0.6 | button 1.0
shove
schema | posture | elbow jnt | accel
slide 1.0 | palm 0.9 | extend 0.9 | high 0.9
relevant linking features
motor parameter features
world state features
schema | posture | elbow jnt | direction | aspect | accel | object
slide | depress | grasp|palm|indx | flex | extend | up | dn | lf | rt | once | iterated | low | med | hi | cube | button
world state features used by schema
weight | at goal
2.3 lbs | false
Slide Schema
start
small
large
PRESHAPE GRASP
PRESHAPE PALM
MOVE ARM TO (objloc)
2
APPLY GRIP
slipping
TIGHTEN GRIP
MOVE ARM (horiz-dir, force, dur)
at goal
not at goal
done
2

Figure 2: An elaboration of Figure 1 at the computational level, showing details of the Slide x-schema, the some linking features, and two verbs: *push* (with two senses) and *shove* (with one sense).

simpler in that the structures are not nested, yet more complex in that the values can be probabilistic.

In the verb learning model a special f-struct called the *linking f-struct* (center of Figure 2) plays an important role as the sole interface between language and action. It maintains bidirectional connections to the x-schemas: an x-schema receives bindings from f-structs and produces additional bindings during its execution. In this way, actions may be translated to and from semantic features. More generally, we will want to claim that the requirements of parameterizing x-schemas are the principal determiner of which semantic features get encoded in a language. The features are chosen with the intention of allowing the model to learn the relevant verbs from any language. One critical linking feature is the name of the x-schema generating the action. Others include motor parameters such as force, elbow joint motion, and hand posture. Some world state features are also relevant such as object shape.

Verb Representation

Each sense of a verb is represented in the model by an f-struct whose values for each feature are probability distributions. Features are presumed independent and the representation is conjunctive or gestalt-like in nature. With this scheme it is necessary to allow multiple senses for each verb, since capturing the meaning in a single sense would in some cases require overly broad distributions which obscure details needed to carry out the corresponding action.

The word sense representation is compatible with Rosch's (1977) prototype theory of categorization and stands in contrast to the necessary and sufficient conditions structure found in logical formalisms. It has a central case which yields the highest response but also gives a graded response when some feature values differ from the central case.

The top third of Figure 2 shows several word senses for the verbs *push* and *shove*. The upper left ellipse gives f-structs for two senses of *push*. The top sense is a hand motion that invokes the Slide x-schema. It also codes somewhat for open-palm hand posture and codes more strongly for elbow joint extension. The bottom sense corresponds to depressing a button with one's finger and invokes a different x-schema not illustrated here. The ellipse on the upper right shows that *shove* also codes for the Slide x-schema but specifies high acceleration.

In execution mode, a verbal command is interpreted by choosing the sense which best matches the current world state. This sense in turn is used to set the linking f-struct, thus determining which x-schema is to execute and with what parameters. For example, *shove* specifies both a Slide x-schema and high acceleration, but the actual amount of force required depends (at least) on the weight of the object involved and that is not specified in the utterance. Our model has a additional f-struct—the world-state f-struct—that encodes things like weights and positions of objects.

Learning

Recall that the central question of our work is lexical development: how children learn to label what they experience. For the verb learning task we assume that the child has already acquired various x-schemas for the actions of one hand manipulating an object on a table. We further assume that the agent hears an informant labeling actions that the agent is performing. As in Regier's work, we avoid some hard but, we feel, separable problems by assuming that the informant supplies just the verb. The learning mechanism assumes that the informant will usually provide at least partially appropriate labels. The problem faced by the model (and the child) is to learn how the verbs relate to its actions and goals.

Several principles from the literature are incorporated into the learning model. One such principle is children's tendency to learn action verbs corresponding to their own actions before extending the meanings to others' actions (Huttenlocher et al., 1983)—thus the focus on motor schemas. Another principle is that learning occurs without explicit negative evidence. Third, the principle of fast mapping (Carey, 1978) notes that children often learn to use a word sensibly after as little as one observation of its use.

Our solution relies on the intimate relation between x-schemas and the passive linking f-struct. Since x-schema executions can be translated into f-structs and vice versa, the action-labeling task is reduced to manageable proportions. The system needs only to correlate informant labeling with sets of feature bindings. This involves determining (1) the correct number of senses for each verb; (2) the relevant features in each sense; and (3) the probability distributions on each included feature. There are many ways of doing this in the computational learning literature. One way that won't suffice is the standard back-propagation used by Regier and the PDP modelers, since we require that the learning technique produce invertible solutions.

Our current experiments use a version of *Bayesian model-merging* (Omohundro, 1992). As applied to the verb learning task, model merging proceeds by first assuming that each execution example is a separate word sense (or model) and then merging senses (models) when this provides a "better" description of the training set. "Better" is formalized in a Bayesian framework: We aim to maximize the posterior probability $P(L \mid T)$ where where L is the lexicon—or collection of word senses—and T is the training set. This is accomplished by applying Bayes' Law to yield $P(L)P(T \mid L)$ and maximizing this product. The first term, $P(L)$, is a prior which assigns higher probability to preferred (usually more compact) lexicons; in our case we use a prior which is an exponentially decreasing function of the total number of word senses. The second term, $P(T \mid L)$, is the likelihood and assigns higher probability to lexicons which would be more likely to generate the training data.

As stated, the model-merging algorithm collects all training data before performing its series of merges. However it is easy to convert the algorithm to work *online*—that is, to develop and refine the lexicon as training examples arrive. In this case the algorithm performs the merge phase every time k new examples have accumulated. One can set $k = 1$ but since model-merging is a greedy algorithm performance can often be improved by setting k to around 10.

A simplified version of the algorithm for handling each new training example follows:

```
incorporate example(f-struct $f$, verb $v$, lexicon $L$)
  create new sense for $f$ and add it to $L$
  when ($k$ such senses have been created) loop:
    $s_1$, $s_2$ ← best candidate merge
            [most similar pair of senses of $v$ in $L$]
    old_post ← compute $L$'s posterior
    replace $s_1$ and $s_2$ in $L$ with merge($s_1$, $s_2$)
            [which sums counts on all feature values]
    new_post ← compute $L$'s posterior
    if (new_post < old_post) then terminate loop
  endloop
end
```

Note that the merging step inevitably leads to a more compact lexicon, but lowers the likelihood of the training set. We stop when the combined effect is detrimental. This algorithm has two key advantages. First the prior function $P(L)$ is an explicit, tunable mechanism for striking a balance between generating too many, very specific senses of a word, and generating too few, excessively general senses. Second, the online version produces sensible results emerge after just one training example, unlike back-propagation style algorithms.

Figure 3 gives a graphical overview of an idealized learning run by the system with $k = 1$. Shown in the left column are linking f-structs summarizing four successive example executions that have been labeled *push* by the teacher. After the first training example, the system assumes that the word *push* labels just that example, involving a SLIDE with the elbow joint undergoing extension, a palmar posture and an acceleration level of about 6 out of 10. A second use of the same word to label a similar action results in the system broadening the range of accelerations it believes can be denoted as a *push*. The third example is quite different and might arise from pushing a doorbell or keypad. The MDL–like evaluation function in the model merging algorithm prefers a second word sense to widening all the slots of its existing model. It has now learned approximately the two different senses of *push* shown in the upper left ellipse in Figure 2. The final example most closely matches the first sense and merges with it. Because this example used a different posture, the posture slot is broadened to allow both possibilities with probabilities approximating the frequencies observed. Also, because the acceleration in this example was so low, the algorithm concludes that acceleration isn't criterial for this verb.

Of course, this is all just a cartoon version of a complex system's operation, but it should convey the flavor

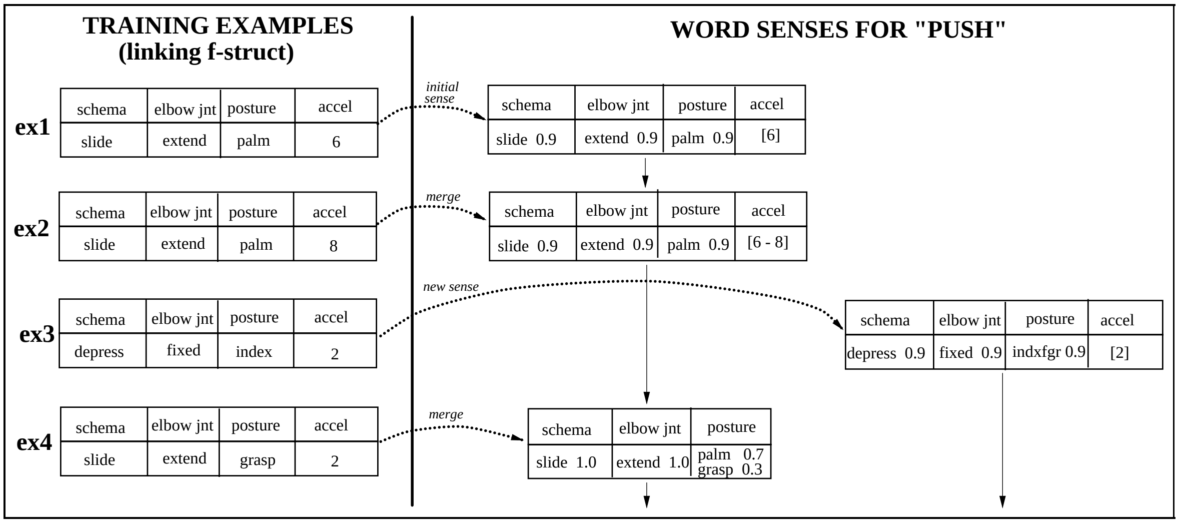

Figure 3: Illustration of model merging for learning the verb *push*.

of the mechanisms. Full training results cannot be reported in this space; instead, we briefly review a training run on a simplified version of the model. Fifty random executions of SLIDE were generated and labeled as *push*, *pull* and *slide* (for sideways motions). Half were chosen as a training set. The merging algorithm collapsed the 12 instances of *push* into a single sense, and did likewise for the 9 instances of *pull*. However the 4 instances of *slide* were collapsed down to two senses; the use of Gaussians for the direction feature's probability distribution prevented merging of the leftward- and rightward-motion instances since the direction feature would become overgeneralized. The resulting lexicon correctly recognized 22 out of 25 similarly generated test cases. All 3 errors involved mislabeling *slide*s as *push*es when the direction was between the prototypical values for the two verbs; *push* was chosen due to its greater frequency in the training set. All three verbs were executed correctly in several randomly generated initial world configurations involving varied hand postures, hand positions and object positions.

When training with a set of 5 x-schemas like SLIDE, *push* converged on two senses, one for SLIDEs with extending elbow and another for DEPRESSing with the index finger. *Pull* convgerged on a single sense encoding SLIDEs with flexing elbow and medium force. *yank* was similar but encoded high force and additionally its examples involved the "grasp" posture often enough to be included in the word sense.

The model's algorithms involve a number of tunable parameters. The most sensitive of these has been the parameter for deciding when a word sense's probabilistic feature value is "peaked" enough to justify using the feature during schema execution.

Connectionist Implementation

We have described the model at the computational level but as stated earlier we are committed to connectionist reducibility for important parts of the model, namely x-schemas and the model-merging algorithm.

Since Petri nets involve only local control, x-schemas could be straightforwardly modeled using connectionist units to represent places and transitions—if it weren't for the complication of passing parameters. A solution to this problem has been developed (Grannes et al., 1997) using a temporal synchrony approach to binding. *Focal clusters*, which in the SHRUTI reasoning system represent assertions and queries of predicates plus their arguments, are extended to trigger and coordinate primitive synergies plus their parameters.

A connectionist account of the model-merging algorithm has been sketched in the framework of *recruitment learning* (Feldman, 1982). Binder nodes represent the conjunction of lexical item, motor parameters and world state features which make up a word sense. These compete to explain new training examples, and when a close match is found, the winning binding node is slightly modified to account for the features of the training example (effectively merging the new example into the existing sense). However if no clear winner is found, a new binder unit is recruited to represent the new training example (effectively creating a new sense). This account applies only to the online version of model-merging.

Discussion

Bailey's model of lexical development of action verbs offers several advantages. The bidirectional mapping between word senses and x-schemas allows the model to not only recognize but also carry out the verbs it has learned. This bidirectionality is accomplished by the intermediary linking f-struct which confers two benefits: (1) Hardwiring relevant features (as opposed to allowing hidden units to evolve novel features) facilitates translation from verbs back to a set of features. (2) Using a restricted, parameterized formalism for x-schemas facilitates translating linking features to and from an executing schema (bridging the declarative-procedural gap).

The use of conjunctive probabilistic feature structures to represent word senses is compatible with the notion of gestalt perception and prototype theory. The model also offers an account of pragmatics: verb representation is simplified by the fact that many context dependencies will be handled during schema execution. The learning algorithm operates without negative evidence and exhibits fast mapping. It also provides an explanation for children's ability to learn typological patterns such as verbs' tendency to encode manner (English) *vs.* path (Korean), although this feature is not discussed here.

Yet, the model has several shortcomings. It offers no account of the radial category structure connecting word senses (Lakoff, 1987). Nor does it explain how x-schemas might interact with image schemas such as Regier's to model phrases like *push through*. An account of x-schema learning and how it relates to lexical learning would also be desirable.

Abstract Concepts and Words

Of course, the whole program of directly embodied meaning only applies to a limited range of concrete concepts and lexical items. Cognitive linguists in general and Lakoff (1987) in particular believe that abstract concepts ultimately derive their meaning from mappings to the directly embodied ones and have studied these mappings for many years. Part of our current effort is to apply the computational techniques described above to model how such mappings might occur.

The key computational mechanisms, x-schemas and binding f-structs, also are at the core of our abstract concept story. Narayanan (1997) describes an implemented program that demonstrates through simulation how an x-schema structure which is a *controller* abstraction over multiple motor actions is able to offer some answers to well known problems in modeling the semantics of verbal aspect. Another project involves a model which interprets certain abstract concepts using, inter alia, metaphorical mappings to concrete source domains. We imagine our model to be reading news stories about international economics and politics and trying to understand them. We test understanding by examining bindings in various f-structs after processing the story. We are currently validating our model by modeling examples from a database of about 30 simple (2–3 sentence) newspaper stories. A critical test is the system's ability to understand stories that appear after the validation is complete. This work is summarized in Narayanan (1996) and will be described more fully in his 1997 dissertation.

References

Arbib, M. A., Iberall, T., and Lyons, D. (1987). Schemas that integrate vision and touch for hand control. In Arbib, M. A. and Hanson, A. R., editor, *Vision, Brain and Cooperative Computation*. MIT Press, Cambridge, MA.

Badler, N. I., Phillips, C. B., and Webber, B. L. (1993). *Simulating Humans*. Oxford University Press, New York.

Bernstein, N. A. (1967). *The Co-ordination and Regulation of Movement*. Pergamon Press, New York.

Carey, S. E. (1978). The child as word learner. In Halle, M., Bresnan, J., and Miller, G. A., editor, *Linguistic Theory and Psychological Reality*. MIT Press, Cambridge, MA.

Feldman, J. A. (1982). Dynamic connections in neural networks. *Biological Cybernetics*, 46:27–39.

Feldman, J. A., Lakoff, G., Bailey, D. R., Narayanan, S., Regier, T., and Stolcke, A. (1996). L_0—the first five years of an automated language acquisition project. *AI Review*, 10:103–129. Special issue on Integration of Natural Language and Vision Processing.

Grannes, D. J., Shastri, L., Narayanan, S., and Feldman, J. A. (1997). A connectionist encoding of schemas and reactive plans. Poster presented at 19th Cognitive Science Society Conference.

Huttenlocher, J., Smiley, P., and Charney, R. (1983). Emergence of action categories in the child: Evidence from verb meanings. *Psychological Review*, 90(1):72–93.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. University of Chicago Press.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of IEEE*, 77(4):541–580.

Narayanan, S. (1996). Embodied language understanding: Modeling the semantics of causal narratives. In *AAAI Fall Symposium on Embodied Cognition and Action*, pages 87–91. AAAI Press TR FS-96-02.

Narayanan, S. (1997). Talking the talk is like walking the walk: A computational model of verbal aspect. Proceedings of the 19th Cognitive Science Society Conference.

Omohundro, S. (1992). Best-first model merging for dynamic learning and recognition. Technical Report TR-92-004, International Computer Science Institute, Berkeley, CA.

Regier, T. (1996). *The Human Semantic Potential*. MIT Press.

Rosch, E. (1977). Human categorization. In Warren, N., editors, *Advances in Cross-Cultural Psychology*, volume 1. Academic Press, New York.

Siskind, J. M. (1992). *Naive Physics, Event Perception, Lexical Semantics and Language Acquisition*. PhD thesis, Massachusetts Institute of Technology.

Tomasello, M., editors (1995). *Beyond Names for Things: Young Children's Acquisition of Verbs*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Winograd, T. (1973). A procedural model of language understanding. In *Computer Models of Thought and Language*. W. H. Freeman, New York.