Talking the Talk *is* Like Walking the Walk: A Computational Model of Verbal Aspect

Srini Narayanan
SNARAYAN@CS.BERKELEY.EDU
International Computer Science Institute and University of California, Berkeley
1947 Center Street Suite 560, Berkeley CA 94704 USA

Abstract

I describe an implemented computational model of verbal aspect that supports the proposition that the semantics of aspect is grounded in sensory-motor primitives. In this theory, aspectual expressions refer to *schematized* processes that recur in sensory-motor control (such as goal, periodicity, iteration, final state, duration, and parameters such as force and effort). This active model of aspect grounded in sensory-motor primitives is able to model cross-linguistic variation in aspectual expressions while avoiding some paradoxes and problems in model-theoretic and other traditional accounts.

Introduction

The study of aspect pertains to the study of linguistic devices that enable a speaker to direct the hearer's attention to the internal temporal character of a situation.[1] Many languages have grammatical aspectual modifiers such as English *progressive* construction (X -ing) which focuses on the ongoing nature of an underlying process while allowing for inferences that the process has started and that it has not yet completed. Similarly, the use of the *perfect* construction (has X -ed) allows a speaker to direct the attention of the hearer to the *consequences* of the described situation. Languages also have a variety of other means to express aspect including aspectual verbs like *start*, *end*, *cease*, *continue*, and *stop* and related grammatical forms.

The frequency with which languages refer to events and the universality of such expressions has made aspect an object of study since Aristotle. Aspect is a particularly interesting phenomenon because in all languages studied, the *natural or inherent* verb semantics combine with and modify the interpretations and entailments of the grammatical marker system. This makes a compositional account of the semantics of aspect difficult giving rise to many paradoxes and problems (Dowty 1979).

This paper demonstrates that a compositional semantics of Aspect can be constructed if we take the embodiment of action in a neural system seriously. In this context, this work is part of ongoing research in the L_0 project exploring issues in Embodied Language (Feldman *et al.* 1996; Bailey *et al.* 1997) being pursued at UC Berkeley and ICSI.

[1]Aspects differ from Tenses in that whereas tenses deal with the temporal relations between situations (such as past, present and future), aspects enable focus on the compositional attributes of a situation.

Basic Result

This paper describes a computer simulation that provides evidence to support the following proposition.

Proposition 1 *Aspectual expressions are linguistic devices referring to* **schematized** *processes that recur in sensory-motor control (such as inception, interruption termination, iteration, enabling, completion, force, and effort).*

We describe a computational model of such schematized perceptuo-motor processes called X-schemas. The model is inspired by knowledge of what is known about the cortical representation of high-level motor control and satisfies general computational constraints on modeling neural activity. In the implemented model, the semantics of individual verbs are *active* and primarily involve salient features of behavior control. Inherent aspect naturally falls out of the representation of verbal semantics.

I propose a second active structure called the **controller**, that captures a control generalization over many individual X-schemas. The controller is itself an X-schema and models important regularities that are relevant in the evolution of processes (enabling, inception, in-process, completion, suspension, resumption, etc.).

The semantics of Aspect arise from the bi-directional interaction of the generalized controller with the specific underlying X-schema for the verb in question. This active model of aspect grounded in sensory-motor primitives is able to model cross-linguistic variation in aspectual expressions while resolving paradoxes and problems in model-theoretic and other traditional accounts. Current work is extending the framework to include metaphoric expressions of aspect.

I begin by briefly outlining motor control primitives relevant to the modeling of Verbal Aspect through an example which will serve to use to illustrate the basic ideas.

Relevant Features of Perceptuo-motor control

Consider what is required for a high-level controller that monitors and controls the *walking* behavior. The controller has to be active until the destination is reached. This includes monitoring and controlling both *concurrent* and *sequential* execution of sub-actions. Some of these sub-actions may be primitive (corresponding to motor synergies in biological control). Others may be further refined (shown as shaded nodes in Figure 1). Figure 1 shows the first level refinement of the walk behavior. Subsequent refinements would include the constituent synergies that make up a step (the various *swing* and *stance* phases and their temporal arrangement).

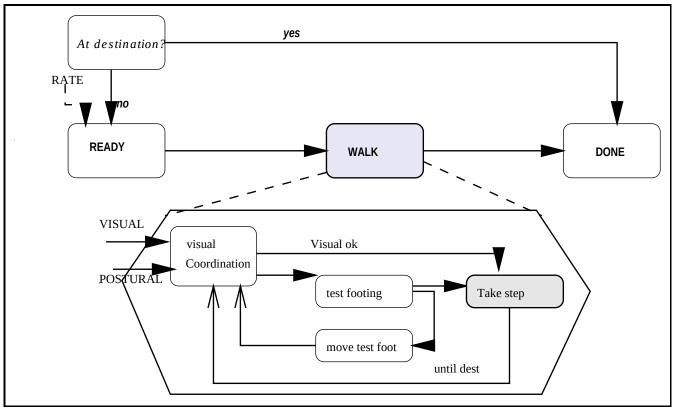

Figure 1: A simplified walk schema

The actual execution trajectory is conditional on the results of perceptual tests and/or world inputs. For example the basic cycle of *ready* $\Rightarrow$ *test footing* $\Rightarrow$ *take step* of the Walk X-schema may be interrupted if the result of *test footing* fails. The basic cycle repeats (at a rate specified by the rate input parameter) until the agent receives information (asynchronously from a perceptual process) of being at the destination. At this point the walking process is completed. Thus an X-schema has to be capable of event-based interruption and should be capable of monitoring and controlling the state of resources (such as energy levels).

In summary, to monitor and control the execution of motor programs and their effects in a dynamic environment, X-schemas need to control both **periodic** and **aperiodic** actions with real **durations** (to monitor, timeout, initiate error-recovery procedures) and to be able to check and monitor **conditions** and **resource** usage. These include monitoring resource consumption (energy level), as well as respecting mutual exclusion constraints (can't hold two blocks at the same time)), and *enabling* and *disabling* conditions (can't walk if the ground is slippery) They should be capable of **goal** based enabling and should be able to monitor and remain active until achievement of the goal. Together, I will refer to these **abstracted** primitives (duration, periodicity, resources, goals and conditions) as the **process primitives**.

The Schema Controller

The **controller** is a control generalization over different X-schemas. Thus whether our simulated robot controller controls the execution of a *push* or *walk* schema, it monitors the *enabling*, *inception*, or *ongoing* and *termination* of the motor program. It should also be capable of coordinating sensory and motor inputs to a state of *readiness* as well as be able to know about the successful *completion* of a program.

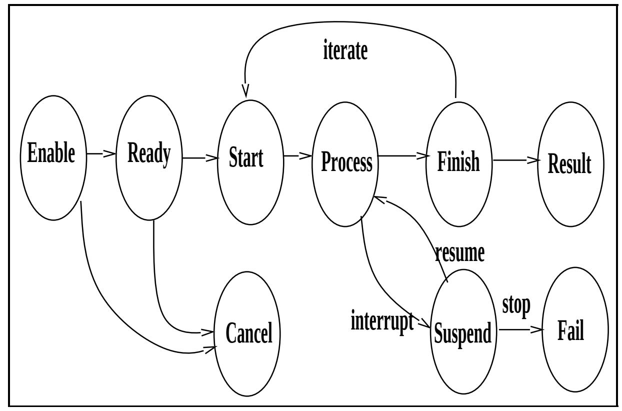

Figure 2: The Schema controller is a generic structure that captures relevant features of behavior control.

Crucially, the generic controller is itself an active structure or X-schema. The controller sends signals to individual motor schemas and may transition based on signals from the underlying schema. Thus nodes in the controller bind to process states in the underlying behavior. Directed arcs constrain behavior execution trajectories. The controller is **stateful** and the control graph encodes possible process evolution trajectories. Thus if the ongoing node is *active*, the controller can conclude that the activity in question has already started and that the next interesting transition is to the termination.

Links To Verb Semantics

We hypothesize that such sensory-motor controllers may be directly coded in our neural circuitry and be available to other cognitive processes such as language interpretation, and more relevantly may ground the semantic and grammatical structure of the well known linguistic notions mentioned above.

When composed with the **process primitives**, the **controller** forms the basis for grounding verbal aspect. The process primitives characterize the *inherent* semantics of the verb. Any individual verb may have some or all of these parameters set to specific values.[2]

The important thing to note is that both the controller and the process primitives that characterize the underlying verb are X-schemas. In this way, the semantics of the verb is *grounded* in the execution of the action itself. Figure 3 shows the same schema as the one in Figure 1 but now redrawn to include the controller abstraction.

Aspect Modifiers or other grammatical devices are like knobs which when set activate the corresponding **controller** node, sanctioning which inferences can be made

[2] These parameters inspired by sensory motor primitives generalize Talmy's aspectual primitives (Talmy 1985) and can easily derive the Vendler-Taylor-Dowty classification *VDT* (Dowty 1979).

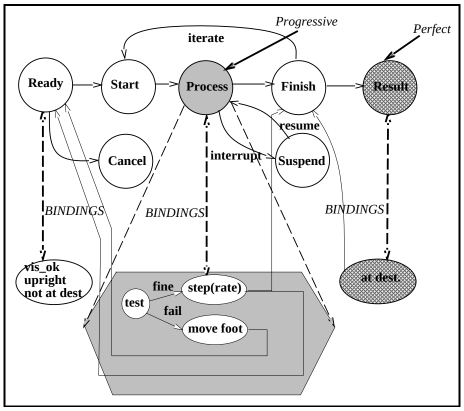

Figure 3: Grammatical devices activate states in the **controller** Schema which bind to process states or dynamic sub-processes generating interpretations and allowing for inferences.

---

by the hearer, given the same underlying schema (verb form). Languages may differ in which knob settings they allow, and hence may vary which aspects and how much bandwidth they allow the speaker.

In Figure 3, we see how the English *Perfect* construction essentially activates the result state of the underlying verb (X-schema). In this case, what is relevant are the characteristics of the process that bind to the result stage of the controller. In the case of *Jack has walked to the store*, this implies that the speaker directs the hearer's attention to the fact that Jack is at the store, probably a little tired. In contrast, *Jack is walking to the store* activates the *process* node of the controller thus calling the hearer's attention to facts such as Jack is not at the store, but actively walking towards it, expending energy, etc.

Thus, meaning arises from the **dynamic binding** of a specific activation state of the controller X-schema to a specific activation state of the verb X-schema. The structure of the controller and the relevant set of features that characterize the verb jointly control the compositional possibilities.

Computational Model

The computational model of X-schemas is based on an extension to Petri nets (Murata 1989).

Definition 1 . X-schema

An **X-schema** consists of *places*(P) and *Transitions* ($\mathcal{T}$) connected by weighted directed arcs $\mathcal{A}$ ($\mathcal{A} \in (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$). Each arc $a_{ij} \in \mathcal{A}$ has weight w_{ij} ($w_{ij} \in \mathcal{N}$). Input Arcs $*\mathcal{T}$ ($*\mathcal{T} \in (\mathcal{P} \times \mathcal{T})$) connect Input Places to Transitions. Output Arcs $\mathcal{T}*$ ($\mathcal{T}* \in (\mathcal{T} \times \mathcal{P})$) connect Transitions to Output Places. Places are typed as *enable* places $\mathcal{E}$, *inhibitory* places $\mathcal{I}$, or *resource* places $\mathcal{R}$.

The underlying model of an X-schema is a weighted, bipartite graph that consists of **Places** and **Transitions**. *Input Arcs* connect Places to Transitions, *Output Arcs* connect Transitions to Places. The bi-partite nature of the X-schema naturally captures the well known **state/event** distinction that pervades linguistic analysis (Langacker 1987). In X-schemas, both states and events are distributed over the entire net. A specific state of the schema corresponds to a **marking**. Formally a marking is a function that assigns either 0 or a positive integer to each place. The state of the X-schema is thus described by an M *-vector*, where the i_{th} element of the vector is $M(i)$, denoting the number of tokens in the i_{th} Place. Clearly, the M -vector ranges over the number of Places in the net.

Definition 2 . Execution Semantics

X-schemas have a well specified real-time execution semantics where the **next state** function is specified by the **firing rule**. In order to simulate the dynamic behavior of a system, a Marking of the X-schema is changed according to the following **firing rule**.[3]

1. A transition t is said to be enabled if **no** *inhibitory* input place $i \in \mathcal{I}$ of t is **marked and** if each enable place $e \in \mathcal{E}$ of t is **marked** and all other input places $p \in \mathcal{R}$ have at least w_{pt} tokens, where w_{pt} is the weight of the arc from p to t.
2. An instantaneous transition *fires* as soon as it is enabled. A timed transition fires after a fixed delay or at an exponentially distributed rate.
3. The firing of an enabled transition t, removes w_{pt} tokens from each non-inhibitory, non-enabled input place p and places w_{tp} tokens in each output place of t.

The following theorem is stated here without proof.[4]

Theorem 1 . *An X-schema is formally equivalent to a* **bounded** *Generalized Stochastic Petri Net (GSPN). The reachability graph of a marked X-schema is thus isomorphic to a continuous time semi-markov process.*

The theorem is important to our effort since Petri nets are one of the most popular and well studied Computer Science formalisms for specifying, modeling, and analyzing highly distributed and concurrent systems (Murata 1989). Algorithms and analysis techniques from that literature can directly be brought in to our work. For instance, to model hierarchical action sets with variables and parameters, we extend the basic model to allow tokens to carry variable binding information (i.e. they are individuated and typed). The expressive power remains unchanged since it is well known in Petri net theory that a net with a finite set of colors can be *unfolded* into one with a single color.

[3] While the firing rule semantics allows enabled Transitions to fire in a distributed manner using local clocks, our sequential simulation adjusts step size to be able to fire multiple enabled transitions.

[4] The proof can be found in (Narayanan 1997). Intuitively, the idea of the proof should be obvious to readers familiar with Petri nets, especially since I have used Petri net terminology in the definitions. The non-trivial issues are proving the boundedness of X-schemas and converting the various place and arc types to their equivalent Petri net primitives without decreasing decision power.

Results of the Aspect Model

Aspectual modifiers such as (*-ing, has X-ed*), and other techniques (such as verbs and adverbs like *start, stop* etc.) provide an *initial marking* to the **controller** schema. The initial marking consists of placing tokens in selected place(s) preferentially selecting one or more transitions. The *inherent semantics* of a verb form interacts with the controller by enabling or disabling specific transitions. Our model allows any of the controller nodes to be decomposed further. For instance the beginning of an activity may involve a limited number of sub-processes (the starting process). The *process* transition is usually decomposed to an appropriate subnet.

Figure 4 shows the implemented computational model of that corresponds to Figure 3. In the graphical representation of X-schemas, Places are drawn as circles, and Transitions as boxes. If a Place p is marked with the integer k, we say that the "*Place* p is marked with k *tokens*." Graphically, this is illustrated by the presence of a black dot with the integer k alongside the relevant Place p.

The top half (enclosed by the broken line rectangle) corresponds to the controller abstraction. In the model, every verb has and interacts with an instance of the **controller** X-schema. The specific interactions come from the inherent semantics of the verb through its *process features*. For example, walking involves specific *enabling conditions* (such as a proper upright posture, a visual test indicating a steady ground, etc.) specific *resources* like energy, and may have a specific *goal* such as being *at the store*. These features interact with the controller preferentially enabling or disabling transitions.

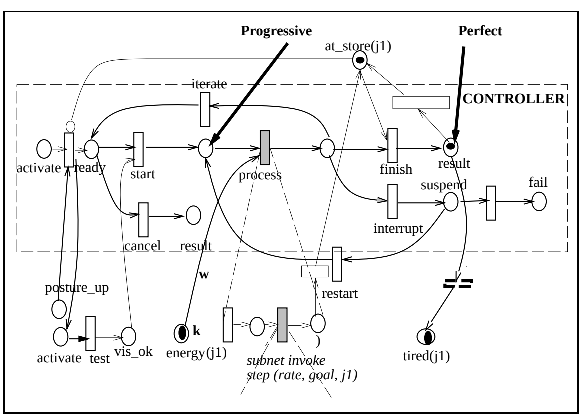

Figure 4: Interpreting the sentence *Jack has walked to the store.* The generic controller X-schema binds to the process X-schema, in the **context** of activation to the generic controller provided by the *perfect* aspect.

Interactions also come from the grammatical devices, which typically supply an initial activation (marking) to the controller. For instance, consider the situation faced by our interpreter upon hearing the utterance *Jack has walked into the store.* Figure 4 shows this situation graphically.[5]

An example is shown in Figure 4. The *Perfect* perspective that results from the *has walked to the store* results in a specific activation to the *result* stage of the controller X-schema resulting in a specific *binding* to the walk situation. Here the hearer is sanctioned the inference that Jack is at the store and possibly tired. More importantly, using the **perfect** aspect the speaker sets the **context** that inferences relevant for future discourse are the world conditions and agent state as a consequence of having walked to the store.

Imperfective Paradox

The Imperfective Paradox (Dowty 1979) comes from trying to separate verbs of accomplishment from verbs of activity. One diagnostic test comes from the different entailments of the two verb classes when used in the Progressive Aspect.

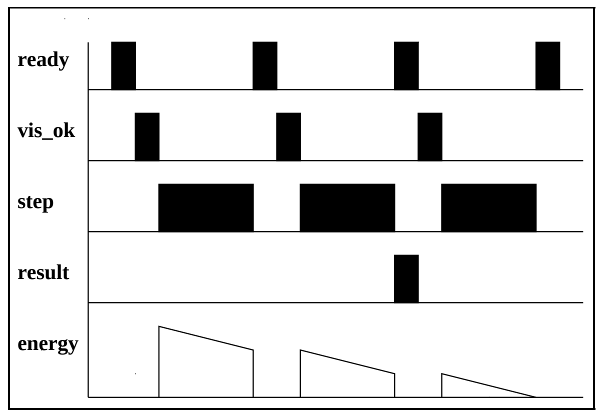

Figure 5: Interpretation of *walking* as a trace of a portion of the **Marking** vector

Consider the difference between *Jack was walking* and *Jack was walking to the store.* The first sentence sanctions the inference that *Jack walked*, whereas the second does not sanction the inference that *Jack walked to the store.* This creates what is referred to as the imperfective paradox, and model theoretic accounts are forced to invent new unanalyzed primitives such as the inertial world primitive **Inr** (Dowty 1979) (set of predictable world futures) to establish truth conditions that satisfy this test.

Figure 6 and Figure 5 graphically depict the relevant portion of the Marking vector for the situations described by the two sentences. In our model, the difference comes from the constraint that in one case the *result* is obtained only if the *goal* (reaching the store) obtains.[6] In the case of *walking*, no such constraint exists and the result obtains after every two steps (taken from

[5] For exposition purposes, the *test-foot* branch is not shown.

[6] Goal attainment can be asynchronously asserted by a scheduled or unscheduled perceptual process (or can be time based (I saw Jack walking to the store yesterday)), and the X-schema reacts appropriately.

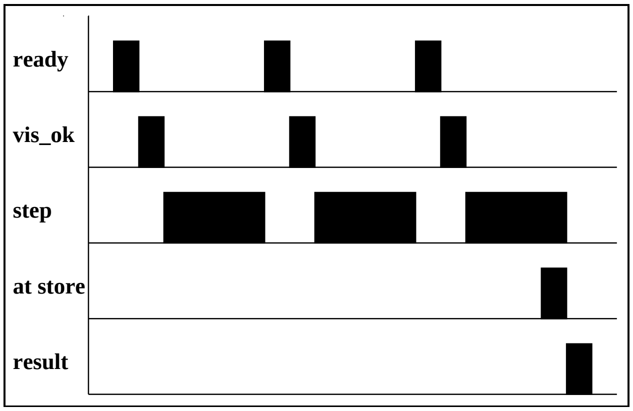

Figure 6: Interpretation of *walking to the store* as a trace of the relevant portion of the **Marking** vector

Dowty's definition (Dowty 1979)). Technically the resolution of the paradox relies on whether the **reachability** graph of the active X-schema contains the result as a submarking, a question that can be answered by activation propagation over the X-schema. In general the active, action-based representation including the ability of our model to *dynamically* model the effects of resources and voluntary or involuntary suspensions seems to be essential to capture the issues involved.

Compositional Issues

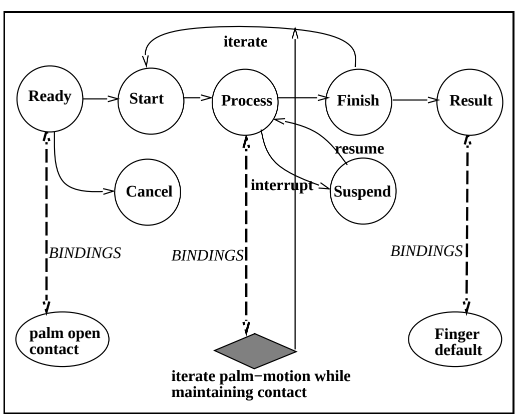

Figure 7: Interaction of Lexical Semantics with Grammatical Aspect

Often, the inherent semantics can make generating the meaning of aspectual expressions non-compositional. For example in the cases *He is rubbing ointment* or *He is coughing*, the normal reading is *inherently* iterative. (Talmy 1985) points out that this has to do with the inherently iterative nature of certain activities. Figure 7 shows how the inherently iterative nature of certain activities can preferentially enable the iterate transition of the **controller**, generating the desired interpretation.

Another example of a problem with the VDT classification comes from trying to classify verbs like *die*. Traditional accounts are forced to abandon compositionality (Comrie 1976) and invent a new class of situations, which once *started* cannot be *prevented*. In our model (assuming Comrie's reading) this is simply done through preferential inhibition of the iterate and interrupt transitions of the **controller** shown in Figure 8.

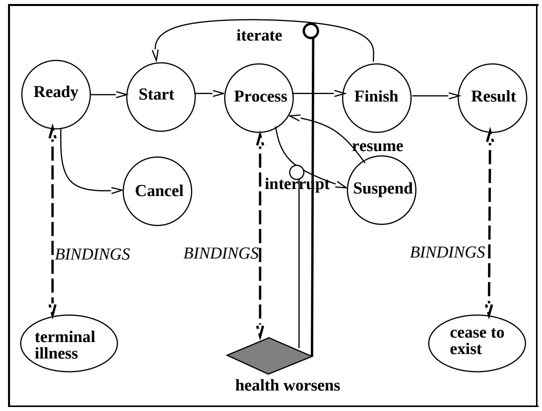

Figure 8: Inherent Semantics may preferentially constrain possible evolutions

Perfectivizing Operators

Figure 9 shows one possible abstraction from the **controller** where the process is not monitored, only starts and finishes are. In this case, through a well defined net transformation (Murata 1989), we get a a simplified controller that (see Figure 9) corresponds to the the **perfective** perspective present in many languages (Langacker 1987). Note that a perfective allows iteration but not interruption since there is no internal structure.

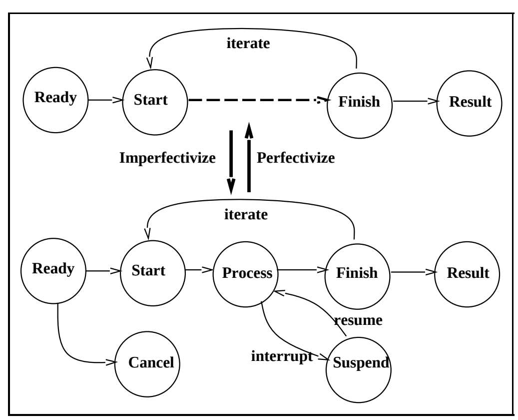

Figure 9: Perfectiving and Imperfectivizing operators

Other Ways To Express Aspect

While most languages have grammatically obligated structures for expressing aspect, there are many optional devices used as well. In the model proposed here, these correspond to activating specific states (trajectories) of the controller. Figure 10 shows some examples of this phenomenon. One interesting result of our design is that many lexical items code for specific types of interrupts in the controller (for example *stumble* codes interrupting a walk schema through a bumpy ground). Secondly, we note that metaphors from different source domains *map onto* the controller (ex. *set out* (journey metaphor) and *enter* (container) map to the **start** state). These provide independent evidence that the controller abstraction seems to capture the inherent temporal structure of events. An X-schema based model that can interpret metaphoric expressions about events is described in (Narayanan 1997).

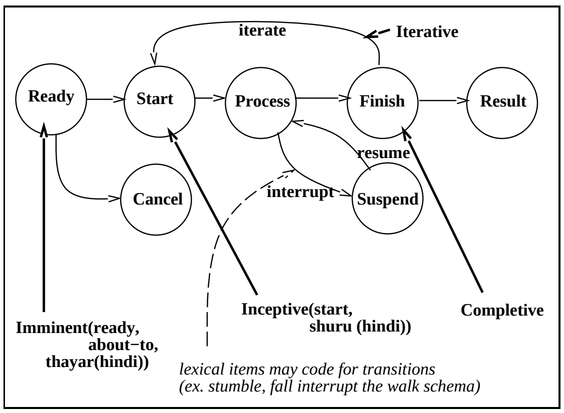

Figure 10: Other ways of Expressing aspect

Conclusion

The main focus of this paper was to provide evidence for the proposition that the semantics of verbal aspect is grounded in primitives of sensory- motor control. To this end, we outlined a novel computational representation and simulation, inspired by well known perceptuo-motor process features, that captures interesting distinctions made by aspectual expressions while avoiding some paradoxes in standard accounts. Recent work has shown that the representation can deal with particle constructions (such as *eat up, back off*) and temporal adverbials.

The active, dynamic, highly-responsive nature of X-schemas enables them to model real-time, defeasible inferences. This novel feature of our model distinguishes it from previous attempts to model Aspect (Moens 1988; Scheler 1996), and allows for a natural solution to the attendant issues of context-sensitivity and inference. We note that (Steedman 1995) proposes the use of dynamic logic to represent the semantics of Tense and Aspect. In this context, we are exploring the connection between X-schemas and a dynamic version of situation calculus. Other related work shows an equivalence between the multiplicative fragment of linear logic (Girard 1987) and X-schemas. Furthermore, there are similarities between X-schemas and connectionist models of general purpose reasoning (Grannes *et al.* 1997). We hope that such links may provide some answers regarding the interaction between deliberative and reactive cognitive processes.

In the L_0 project, we hypothesize that "much" of what is grammaticalized in a language is grounded in patterns generated by our sensory and motor systems as we interact in the world. We conjecture that perceptual and motor control generalizations similar to and interacting with the **controller** can model tense, conditionals, and modals. Preliminary work in this regard has been promising but much remains to be done.

Acknowledgements

Jerry Feldman and George Lakoff made significant contributions. Thanks also to David Bailey, Dean Grannes, Lokendra Shastri and other members of the L_0 group.

References

Bailey, D. R., J. A. Feldman, S. Narayanan, G. Lakoff (1997). Modeling Embodied Lexical Development *Proc. of the Annual Cognitive Science Conference, 1997 (to appear).*

Comrie, B. (1976). *Aspect*, Cambridge University Press.

Dowty, D. (1979). *Word Meaning and Montague Grammar.* Synthese Language Library, Reidell, New York.

Feldman, J. A., G. Lakoff, D. R. Bailey, S. Narayanan, T. Regier, A. Stolcke (1996). L_0 —the first five years of an automated language acquisition project. *AI Review* v. 10, Number 1. Special issue on Integration of Natural Language and Vision Processing.

Girard, J. (1987). Linear Logic, *Theoretical Computer Science* (50), 1-102, 1987.

Grannes, D., L. Shastri, S. Narayanan, J. Feldman (1997). A Connectionist Model of Schemas and Reactive Plans. *Proc. of the Annual Cognitive Science Conference, 1997 (to appear).*

Langacker, R. (1987). *Foundations of Cognitive Grammar I: Theoretical Prerequisites.* Stanford University Press, Stanford.

Moens, M., M. Steedman (1988). Temporal Ontology and Temporal Reference. In *Proc. ACL–88, V4, Number 2, June 1988, pp. 15-29.*

Murata, T. (1989). Petri Nets: Properties, Analysis, and Applications. In *Proc. IEEE–89, V77, Number 4, April 1989, pp. 541-576.*

Narayanan, S. (1997). *Embodiment in Language Understanding: Perceptuo-motor Representations in Metaphoric Reasoning about Event Descriptions.* Ph.D. Thesis, Department of Computer Science, University of California, Berkeley (in preparation), 1997.

Scheler, G. (1996). Learning The Semantics Of Aspect. In *New Methods in Langauge Processing*, Univ. College London Press, 1996.

Steedman, M. (1995). Dynamic Semantics for Tense and Aspect. Proc. IJCAI, pp. 1292-98 1995.

Talmy, L. (1985). Lexicalization Patterns, Semantic Structures in Lexical Form, in *Language Typology And Symbolic Description*, v3: Cambridge University Press, 1985.