Simulation Semantics, Embodied Construction Grammar, and the Language of Events.   
  
  Jerome Feldman and Srini Narayanan, ICSI and UC Berkeley   
  
 Extended Abstract for an AAAI workshop on:   
  
Language-Action Integration for Artificial Agents: Integrating Vision, Action and Language

August, 2011; San Francisco  
  
It remains challenging to communicate with Artificial Agents about actions, events, and processes where the agents are embedded in dynamic, partially observable environments. This talk will present an overview of relevant efforts in the ICSI/UC Berkeley Neural Theory of Language (NTL) project (Feldman 2006). Now well into its third decade, the NTL project combines advanced computational methods with theories and representations based on all relevant biological and behavioral research. One foundational NTL idea is Simulation Semantics (Narayanan, 1999) and its formalization as Coordinated Probabilistic Relational Models (Barrett 2010), which have been applied in a wide range of studies. A related, but previously separate, core concept is Embodied Construction Grammar (ECG) and the related notion of best-fit analysis (Bryant, 2008).   
  
A major current undertaking is the integration both techniques in a system for language understanding, shown in Figure 1. Part of our current motivation for the integrated system is to use ontologies and situation representations and inferences that are compatible with web based semantic representation languages. The two initial task domains are interaction with artificial agents in (simulated) robotics and card games. For the card game task, the initial project goal is to build a system that will be able to understand any of the hundreds of Solitaire descriptions well enough to play the game. The robotics task involves less complex language, but a much richer real-time simulation environment (Schilling 2010).

Figure 1. Overview of a Language-Action System.

The labels in italics denote data and the roman labels denote processes. Starting from the upper right, the system is assumed to operate in a real or simulated environment, WORLD, with a task- specific API; most of the rest of the system is independent of the specific task and domain. Actions are represented by X-nets (Narayanan, 1999), a generalization of Petri nets.

Focusing on the TEXT input to the box on the lower left, the Analyzer program (Bryant, 2008) uses a grammar in ECG (Feldman 2006) and a matching ontology, currently implemented in OWL. It also uses an internal model of the discourse context to help resolve references. It produces a deep semantic representation, called the *SemSpec*, from text input in context. The Analyzer been running for several years and been used in several UCB dissertations, including those cited. The new effort involves using the *SemSpec* to control action. This requires an additional program, the Specializer, which extracts the task-relevant meaning from the analyzed input. Our effort is explicitly geared toward using language for instruction and synthesis of procedures, as in synthesizing action networks for new games of solitaire from textual descriptions of the game settings and legal moves typically found in online game resources.

For example, a typical specification in Klondike solitaire is: “If a space is created in the tableau, it may only be filled with a king.” The grammar is conventional, but the meaning is obviously specific to card games. The Specializer extracts the task relevant information for transmission (as N-Tuples) to the Problem Solver, which itself has no language capability. The Problem Solver (lower right) uses domain-specific knowledge plus the set of N-tuples describing this specific game to compile a specialized X-net for playing Klondike solitaire. Our current efforts use OWL-S (Narayanan and McIllraith 2003) for the synthesis of action nets. This compiled X-net is the Action X-net shown in the upper right of Figure 1.

But action depends not only on the input language, but also on the *Situation*, depicted in the circle on the upper left of the Figure. For card games, the situation involves at least the current state of visible and hidden cards. In a robotics domain, the situation includes the known positions of objects and the states of the robot, etc. The situation description is at a higher level than the detailed actions and is used in several ways. On the language side, situational context is often needed to understand language involving deixis or reference resolution. The situation also models the indirect effects of actions, using logical and probabilistic inference techniques. In addition, the Problem Solver (lower right) uses situational information to choose appropriate action. General perception does not play a large role in our current efforts, but would fit in as another mode of input to the Situation.

All of the individual components have been developed and tested, but only pilot systems have been assembled as of May 2011.

References

Barrett, L., 2010. *An Architecture for Structured, Concurrent, Real-time Action*. University of California, Berkeley dissertation.

Bryant, J. E., 2008. *Best-Fit Constructional Analysis*. University of California, Berkeley Dissertation.

Feldman, J., 2006. From Molecule to Metaphor. Cambridge, MA: MIT Press.

Narayanan, S., 1999. Reasoning about actions in narrative understanding. In Proc. Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99). Morgan Kaufmann Press.

Narayanan, S., and McIllraith, S.. 2003. Analysis and simulation of web services. Computer Networks 42.675–693.

Schilling, Malte, 2010 *Universally manipulable body models for cognitive control*, Bielefeld University Dissertation.

Sinha, S., 2008. *Answering Questions about Complex Events*. University of California, Berkeley dissertation.