Integrating language modeling components for robotic interaction *

Deryle Lonsdale
Brigham Young University
Linguistics
4039 JFSB
Provo, UT, USA, 84602
lonz@byu.edu

Rebecca Madsen
Brigham Young University
Computer Science
3361 TMCB
Provo, UT, USA 84602
rmadsen@byu.edu

Paul Benjamin
Pace University
Computer Science
1 Pace Plaza
New York, NY, USA 10038
benjamin@pace.edu

ABSTRACT

In this paper we sketch the integration of several language processing capabilities into a robotic interaction environment. Based on the Soar cognitive modeling architecture, it assures a unified treatment of several layers of language: lexical, syntactic, semantic, and discourse. We also describe how we have added speech input and output capabilities. Sample implementation scenarios are presented, along with notes about ongoing and future work. The overall purpose of this work is to instantiate a single approach to cognition, perception, real-time action, language and learning, perhaps the single most distinguishing aspect of our project.

Keywords

interaction, cognitive architecture, dialogue processing

1. INTRODUCTION

This paper introduces the natural language interaction framework used within the ADAPT (Adaptive Dynamics and Active Perception for Thought) cognitive architecture for robotics research. ADAPT integrates three principal theories: the Soar cognitive modeling architecture for cognition [26], the Robot Schemas formal model of sensorimotor activity [22], and an algebraic theory for decomposing and reformulating percepts, plans, and problem solving strategies [2]. At the present time an initial implementation of ADAPT has been achieved using a Pioneer P2 robot equipped with stereo color vision, microphone and speakers, sonars and touch sensors.

Soar[1] is an agent-based modeling theory and system that supports hierarchical goal-directed reasoning and learning within a traditional operator-based framework. It has been applied to a wide range of human modeling, interactive, and autonomous applications including robotic control. Besides being used for basic robot-based cognition, the Soar architecture also provides the foundation for natural language (NL) processing capabilities, and so we focus on this aspect of the infrastructure here; other views of the system are described elsewhere [3, 5, 4].

ADAPT is being developed by groups at three sites: Fordham University, Pace University, and Brigham Young University. A recent development in ADAPT, and one topic of this paper, is the BYU group's integration of the NL component with the robot's speech recognition and speech production software. The result is that the robot can listen to a person speak, generate an appropriate response using a discourse model, and respond to the person in spoken language. The following discussion sketches how the robotic agent hears, processes, and responds to such utterances.

First an overview of the linguistic processing modalities is presented, followed by a description of how interactive scenarios unfold. We conclude with a discussion of ongoing work and implications for the approach used in ADAPT.

2. INTERACTION COMPONENTS

Soar's approach views complex cognitive processes and behaviors as decomposable into sequences of operators that perform lower-level specific tasks. When permitted by resources, task demands and time, an agent might be able to sequence many different kinds of operators together—including language and non-language ones—thus providing a highly reactive, interruptible, interleavable real-time language capability [25]. In human/robotics communication these properties are crucial [6].

Our NL system has been previously integrated in other Soar-based task modeling situations involving such agent contexts as the NASA test director (check this citation) [24], and intelligent forces in combat situations [28].

The NL component processes each word individually and performs the following operations in order to understand the input text:

- lexical access (which retrieves morphological, syntac-

*(Produces the permission block, copyright information and page numbering). For use with ACM_PROC_ARTICLE-SP.CLS V2.6SP. Supported by ACM.
[1]See sitemaker.umich.edu/soar for more information.

Human Robot Interaction '06 Salt Lake City, UTah USA

tic, and semantic information for each word from its lexicon) [20, 29]

- syntactic model construction (linking together pieces of an X-bar parse tree) [18]
- semantic model construction (fusing together pieces of a lexical-conceptual structure) [21]
- discourse model construction (extracting global coherence from individual utterances) [10, 27, 11]

Each of these functions is performed either deliberately (by subgoaling and via the implementation of several types of lower-level operators) or recognitionally (if pre-existing operators have already been acquired via the system's machine learning capability). Three types of structure resulting from the utterance will be important for subsequent processing: the X-bar model of syntax, the LCS model of semantics, and the discourse model. The depth and breadth of the interface between these structures and the robot's incoming percepts and outgoing actions are being explored during the project.

It is important to note that the language capabilities just mentioned and further explained below are all assured within one framework, not the concatenation of several specialized modules. All of these language capabilities are thus highly integrated with each other. Furthermore, the underlying Soar theory is also used in more general aspects of robotic cognition, so a tighter integration of cognition and perception with the language capabilities is also possible in this framework. While this nexus has yet to be fully explored, we show here some of the preliminary results.

Though other robotic systems have also achieved varying degrees integration with NL components (e.g. [15, 30, 23]), our approach is fundamentally different. Existing architectures consist of separate components that have been connected: for example a planner, a reactive component, a vision component, and perhaps several different NL components for parsing, generation, and dialogue processing. These components usually derive from different assumptions and methodologies, and the connections between components only assure mutual availability of results. Deeper integration is evasive: how the architecture sees bears no relation to how it solves problems or how it uses language. Our belief is that we are likely to achieve sophisticated robot performance in unstructured environments by building a unified architecture in which problem solving, perception, real-time action, use of NL and learning are based on a single representational structure and a single approach to problem solving. Our approach, then, is to build a unified cognitive robot architecture. The language component described in this paper serves as an example of ADAPT's overall functioning.

2.1 Utterance comprehension

The agent processes incoming sentences in a completely incremental fashion. Words enter the system one at a time, and their presentation rate can be controlled. They are placed into a decay-prone buffer from which they will disappear if not attended to in a timely fashion. A lexical access operator fires when a word is selected for attention; this operator retrieves all phonological, morphological, syntactic, semantic, and lexical features relevant for that word. Relevant properties are supplied from system-internal resources and external repositories (e.g. WordNet [9]).

Once a word is attended to, it can be integrated into the continually unfolding syntactic parse tree, or utterance model, describing the structure of the utterance. The standard X-bar representation formalism drives operator-based construction of the syntactic model.

As the syntactic model is being built up, a corresponding semantic model is also incrementally formulated. It expresses concepts and their relationships according to the Lexical Conceptual Semantics (LCS) formalism [13]. Semantic construction operators assure that the associations are meaningful, appropriate, and consistent with the syntactic model.

Though beyond the scope of this paper, it should be noted that the system is capable of learning strategies for processing the incoming material and leveraging it in approximately similar circumstances in the future. All levels of processing (i.e. lexical, syntactic, and semantic) can thus be "chunked up" as experience accumulates. This helps assure highly optimized, near real-time reactivity for language processing. Another feature of the parsing system is that when momentary ambiguities arise and the system commits to one interpretation that later proves to be incorrect, the system can perform limited repair of the structure, thus recovering from temporary "garden paths" [18].

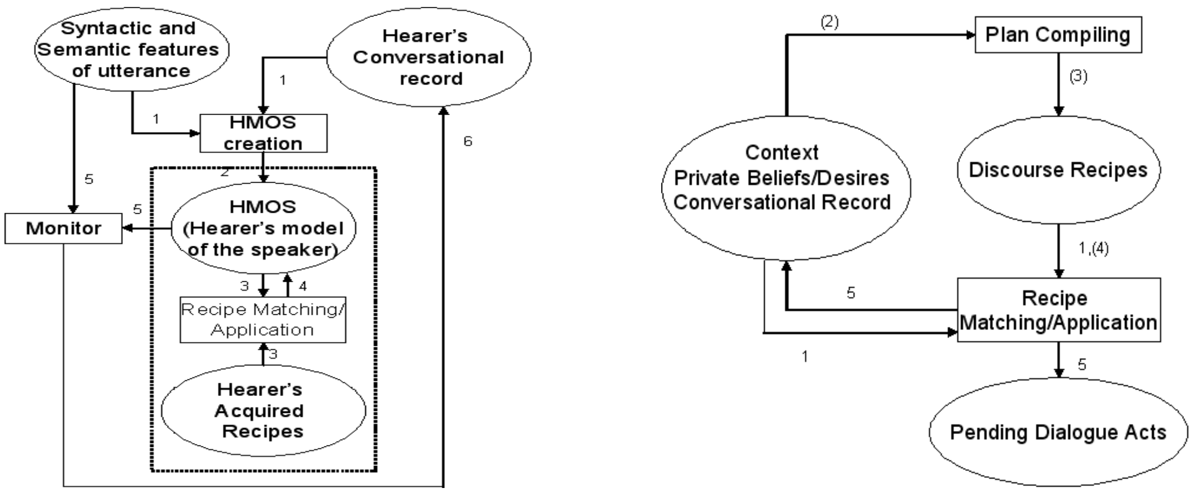
2.2 Utterance generation

As outlined above, utterance comprehension involves processing an input stream of words through lexical, syntactic, and semantic levels of processing. On the other hand, utterance generation constitutes the reverse of this process: the agent formulates a sentence from semantic content using the same type of linguistic structures. Once a semantic LCS model has been selected or formulated, the agent traverses the network, incrementally selecting nodes to process. Processing each concept involves converting it to a lexical form (usually a word) and then adding the word to an ongoing X-bar syntactic model. Crucially, the same syntactic construction operators can be used for generation as for comprehension, so bootstrapping is possible across these modalities [19].

One possible strategy—a conservative one—for generating the output utterance sentence is to traverse the syntactic model when it is completed, collecting the lexical content from all the leaf nodes and linearizing them into a sentence. This is the default strategy for generation, assuring that the model is grammatically correct and complete before the agent "says" its content. Of course other, more incremental, strategies are also possible; the risk is that, if structures must be reformulated, dysfluencies will arise in the output utterance as repairs are carried out.

2.3 Utterance dialogue processing

Once the agent has comprehended an utterance, dialogue processing must occur in order to generate a response. Dialogue processing is the step between comprehending an utterance, and formulating a response to that utterance. Different approaches to managing this step—processing dia-



(a) (b)

Figure 1: Processing for dialogue plan recognition 1(a) and plan generation 1(b), based on [11].

logue in computer/human interactions—have been explored, each revealing advantages and disadvantages [27].

Finite-state machines (FSM) seem most popular when the domain and scope of a dialogue is limited and well-defined. Template-based models introduce more flexibility in a computer-human conversation, but are still suited for small-scope, task-oriented problems [17, 32]. Belief-desire-intention (BDI) architectures [14, 12] and dialogue planning systems [1] were introduced to help explain why the computer system might react in a certain way; these models are generally intended for larger-scope dialogues. The ADAPT system uses a method of dialogue planning, termed discourse "recipes" [11], in order to take advantage of semantic information, context, and learning.

The dialogue management technique ADAPT leverages is a plan-based approach, managing models of discourse referents and participants. It maintains a model of given information (a common ground), and new information. The system uses a model of conversational strategies, or plans, as well as speech or dialogue acts. Using the same operator-based approach to learning as the rest of the system, it can learn dialogue recipes from previously compiled plans. ADAPT also takes the same general approach in motion planning, constructing a 3D world model of its environment, and then exploring alternatives within that model to select an action, and chunking the result.

To recognize a dialogue plan, the dialogue component combines the syntactic and semantic features of the utterance (from comprehension) with the agent's conversational record to create the hearer's model of the speaker (HMOS). The hearer in this case is the robot; the speaker is the human. Figure 1(a) illustrates this discourse comprehension process. The language agent takes the HMOS and attempts to match it against possible dialogue moves (dialogue acts) and dialogue plans that the human may be intending to accomplish with the interaction. This creates the language agent's model of the discourse context (including the human user) and updates its conversational record.

Using the context—private beliefs, private desires, the agent's model of the human's goals, and the updated conversational record—the language agent attempts to generate a dialogue plan, as illustrated in 1(b), to respond to the human. Based on the context, the language agent tries to determine if there is a discourse recipe, or a previously learned plan, that matches the current context. If there is not a match already, the agent compiles a new dialogue plan to generate a response. This new plan can then be learned through a compilation procedure to create a discourse recipe for future use. The discourse recipe is what the language agent uses to create dialogue acts, which will then go through the utterance generation process to send a response to the user.

2.4 Speech integration

A more recent development in the system is to extend its capabilities to allow for speech-based interaction, rather than requiring incoming words to have a textual form. This allows the human interlocutor to escape mouse-based and keyboard input, using instead the more natural spoken modality for interacting with the robot.

For speech input we have developed an implementation of the SPHINX [16, 8] automatic speech recognition (ASR) system (version 4)[2]. It is designed to reside on the robot's Linux platform.

Though the system is capable of recognizing speaker-independent, domain-independent speech using a general language model, we have chosen to develop and encode our own context-free

[2]See cmusphinx.sourceforge.net for more information.

grammar of possible domain-specific utterances. This helps improve recognition accuracy.

When the human speaks into a high-quality microphone, the robot "hears" the signal and processes it through an onboard Sphinx server system. The output is a word-based textual transcription of the utterance, which is then passed word-by-word over a socket to the language agent (which for now is off-board). These words then enter the system and undergo comprehension processing as outlined above.

The system also includes a text-to-speech (TTS) synthesis system built on the Festival toolkit[3]. A client/server architecture was developed so that the robot could run a Festival server onboard, connected with the system's speakers. As utterances are generated by the language agent as explained above, they are sent—again over sockets—to the robot's TTS server along with a few parameters to specify voice gender, intonation, etc. The utterance is then output over the onboard speaker.

As described above, the current implementation uses a textual, word-based representation of speech input and output for communication between agents. This is a temporary simplification; the ASR engine is capable of delivering a speech lattice representing several recognition hypotheses, and we are in the process of allowing it to be directly represented in the language agent's hierarchically structured input memory. This will eventually allow a closer modeling of speech perception, which is not crucial at this stage of the research.

Another temporary simplification mentioned above is that the speech ASR and TTS components run on the robot, whereas the language agent is run on an offboard controller. Though in theory the language agent can also run onboard, for the moment we are not doing so, instead running it off-board along with some other robot controller functions.

3. SCENARIOS

To illustrate the modularity and generality of the processing mechanisms outlined above, we have implemented two kinds of interactive scenarios. The first is a typical speech-based HCI setting, and the other where two computer agents interact through language.

3.1 Human-robot interaction

The HCI scenario, though not yet integrated directly with the robot, presents an interaction between a human and the language agent.

In this scenario, the human is able to use the microphone as described previously to send an utterance to the robot, and the robot sends the response to speakers so the human can hear it.

A partial trace of processing for a typical interaction is given in Figure 2. It shows operator firings while the robot is hearing the user say "My name is Rebecca. Over". In this situation we have a two-utterance turn, and the trace picks up in decision cycle 203 where operator A230 performs lexical

[3]See www.cstr.ed.ac.uk/projects/festival/ for more information.

```
 203: O: A230 (access word: 'rebecca' spkr: user)
 219:       O: C576 link(is.v--comp-->rebecca.n)

 235:          O: O392 try-operator(fuse(is.v-stative--internal-->rebecca.n-person)
 242:             O: C650 (check check-syntax)

*** Soar hears you say: over
 243:             O: C648 (check adjacency)

 246:       O: C647 fuse(is.v-stative--internal-->rebecca.n-person)

*** Soar recognized discourse content: long-self-identifier ***

 255: O: O406 (comp-d-move)
*** Soar comprehended discourse move
 move:         self-introduction
 speaker:      Y1
 addressee:    S14
 content type: speaker-identifier

 257: O: A326 (access word: 'over' spkr: user)
 275:       O: O432 (empty-op)
 279: O: O441 (u-constructor152)

 285:    O: O446 (s-constructor154)
 288:          O: O454 try-operator(link(--context-->over.j-pertainy)
 289:       O: O455 link(--context-->over.j-pertainy)

 311:    O: O485 (comp-d-move)
*** Soar comprehended discourse move
 move:         end-turn
 speaker:      Y1
 addressee:    S14
 content type: end-turn
 313: O: O488 (comp-release-turn)
 314: O: O490 (learn-discourse)
 316:    O: O491 (d-plan-constructor157)
 318:       O: O493 (acquire-turn)
 319:       O: O492 (achieve-d-goal)
 320:       O: O495 (release-turn)
 322:    O: O496 (return-d-plan)
 323: O: O491 (d-plan-constructor157)
 324: O: O500 (lcs-choose)
 325: O: O510 (select-semobj)
*** Soar selected concept S39
```

Figure 2: Trace of discourse processing (comprehension and generation) after understanding a sentence and before generating a response.

access on the word "Rebecca"; prior operators have already processed the previous words; we have omitted in the figure operators not consequential to the discussion at hand. In cycle 219 operator C576 links the noun "Rebecca" into the syntactic model in the complement position of the verb "is", and then in cycle 235 operator O392 initiates planning for semantic model construction by fusing the corresponding person concept into the internal argument slot of the stative verb interpretation of the verb "be".

Between cycles 242 (where operator C650 checks syntax to license the proposed semantic attachment) and 243 (where operator C648 checks linear ordering of the words), the robot perceives the incoming word "Over". Note that this unnatural signal for the user's dialogue move is a temporary simplification to avoid endpointing in the ASR system and is inherited from earlier versions of the system which were based on cockpit communications, where the overt turn terminator was mandatory. In cycle 246 operator C647 performs the planned successful semantic model attachment.

In cycle 255 operator O406 shows that the system realizes that a discourse move has just been completed. It recognizes the move as a self-introduction speech act on the part of the human (Y1), and that the system (S14) was the intended recipient. The move's content type is to identify the speaker.

In cycle 257 operator A326 attends to the word "over" and performs little syntax and semantics since the word is essentially a standalone item. Crucially, though, cycle 311 involves another comprehended discourse move, and "Over" is intepreted as a turn-ending signal from the human to the system. In cycle 313 operator O488 recognizes that the turn has been terminated and releases initiative from the speaker. This concludes the discourse comprehension phase.

Next, the system initiates learning a discourse plan to respond. It first plans to acquire a turn, then formulates a plan to achieve its discourse goal, and then plans to release the turn. In cycle 322 the discourse plan for execution is returned, which is then executed in the next cycle.

At this point the discourse generation plan has been completed, and the system begins to generate a response (in this case its own self-identifying utterance directed at the human). Cycle 324 invoves choosing part of the LCS to begin constructing the target sentence's syntactic model. The last cycle shown selects that object for lexical and syntactic generation as explained in section 2.2. Further cycles not shown here perform the remaining complex stages of generating and saying the utterance in question.

Figure 3(a) outlines the complete communication cycle detailed above. A video demonstrating this process has also been created, showing roundtrip communication between a human user and the language agent on a separate computer.

3.2 Agent-agent interaction

The system is also capable of supporting agent-agent interactions, though we have not yet actually instantiated this type of dialogue on robots—just as software agents. In this type of scenario an agent formulates and activates a plan to communicate some information to another agent. The appropriate type of discourse act is chosen, the prelinguistic concepts are selected, and the semantic LCS is constructed. It is then mapped to a syntactic structure, and a sentence is generated from the leaf nodes. The words are sent to another agent where they enter the input buffer[4].

The second agent comprehends the input utterance by accessing each word, building the syntactic model, and building a corresponding semantic model. A discourse recipe is inferred from the content, and the second agent formulates a plan to respond. In one scenario we have demonstrated, for example, the agent is driven by a goal to "better" its counterpart, responding to an utterance by displaying one-upmanship using relevant conversational strategies. For example, if it is told by the other agent "I have a nice car." the agent will respond with "I have a better car." The second agent assembles a discourse plan for its response, constructs an LCS from the individual concepts selected, and generates its own output syntactic model. The leaf nodes are converted to a sentence, which it then sends back to the first agent over a socket.

The first agent then processes the other's utterance, plans a response, generates an utterance, sends it to the other agent, and the cycle continues (see Figure 3(b)).

4. FUTURE WORK AND CONCLUSIONS

The framework is in place; now we need to add more basic components to the store of ingredients needed to generate the discourse recipes. At the present time the interactions are rather simplistic. We are working on introducing more flexibility into the communication process by allowing a greater number of multi-utterance dialogue turns. Another area being extended is the library of dialogue plans. As evidenced earlier, most of the low-level plans existing in the system were inherited from work done for other domains, so some of the existing plans are not applicable to the robotics domain. There are block-world dialogues and route-based instruction corpora [7] elsewhere which would likely be more applicable; we should be able to leverage external work to build up our repository.

For example, a useful type of dialogue plan we would like to add to the system is clarifications. Task-based ambiguities could be resolved by employing a dialogue plan to clarify an instruction. For instance, if the robot were given the command to "pick up the block from the table", but two blocks were on the table, the robot could use the clarification plans to inform the interlocutor that two blocks are on the table. This would enable the robot to explicitly address the task-based ambiguity via natural language, and ask further questions to determine which of the two blocks it should pick up.

One of the benefits of our framework is that perception and language can be expressed in the same operators, making it easier to translate visual cues to language cues, and vice versa. Color then holds more meaning in an utterance, into-

[4]For now the communication is text-based across sockets; we have not yet implemented—but could—a scenario where the first agent "speaks" to the second one via speech synthesis, which would thus require the agent-addressee to perform speech recognition on the incoming utterance.

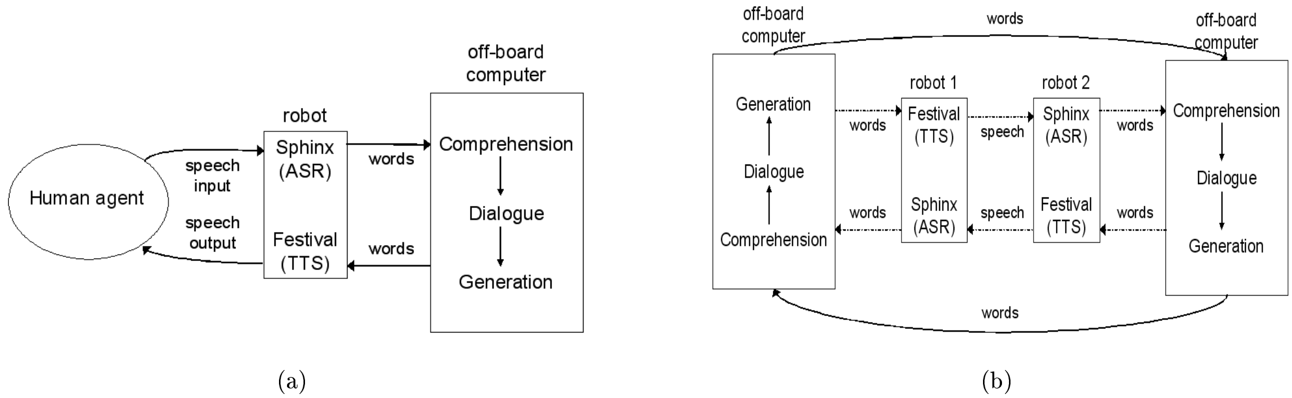


Figure 3: Processing during interaction in the HCI scenario 3(a), and during the two-agent scenario 3(b) with dotted arrows showing future functionality.

nation in speech puts emphasis on things in the real world instead of just on words in isolation, deixis presumes that information on which object is "the object" is already in the common ground, etc. As we integrate the language system with the robot itself, we have the opportunity to more tightly associate percepts with language to leverage this potential advantage.

Evaluation of dialogue systems is an issue of current research in the literature [31]. Several viable techniques have been proposed and discussed, but without a standard in evaluation, it is still difficult to measure improvements and to compare systems.

Despite the work in progress, we have achieved interesting results already. We have integrated the language component within the same cognitive model that serves the other robotic functions; lexical, syntactic, semantic, and discourse processing are within the same operator-based framework as the rest of ADAPT. The speech integration we demonstrate is versatile, able to support both human-robot interaction as well as agent-agent interaction.

5. ACKNOWLEDGMENTS

We would like to thank Robbie Haertel, Eric Roundy, and Merrill Hutchison for valuable programming assistance in this work, and the BYU NL-Soar research group for its collaboration.

6. ADDITIONAL AUTHORS

Additional authors: Damian Lyons (Fordham University, email: `dlyons@fordham.edu`).

7. REFERENCES

[1] L. Ardissono, G. Boella, and R. Damiano. A plan-based model of misunderstandings in cooperative dialogue. *International Journal of Human-Computer Studies*, 48:649–679, 1998.

[2] D. P. Benjamin. On the emergence of intelligent global behaviors from simple local actions. *Journal of Systems Science special issue: Emergent Properties of Complex Systems*, 31(7):861–872, 2000.

[3] D. P. Benjamin, D. Lonsdale, and D. Lyons. Integrating perception, language and problem solving in a cognitive agent for a mobile robot. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1308–1309, 2004.

[4] P. Benjamin, D. Lonsdale, and D. Lyons. ADAPT: A cognitive architecture for robotics. In *Proceedings of the Sixth International Conference on Conceptual Modeling*, pages 337–338. Lawrence Erlbaum Associates, 2004.

[5] P. Benjamin, D. Lonsdale, and D. Lyons. Designing a robot cognitive architecture with concurrency and active perception. In *Fall Symposium: The Intersection of Cognitive Science and Robotice: From interfaces to intelligence*. American Association for Artificial Intelligence, 2004.

[6] N. Blaylock, J. Allen, and G. Ferguson. Synchronization in an asynchronous agent-based architecture for dialogue systems. In *Proceedings of the 3rd SIGdial workshop on discourse and dialog*, Philadelphia, 2002. Association for Computational Linguistics.

[7] G. Bugmann, E. Klein, S. Lauria, J. Bos, and K. T. Corpus-based robotics: A route instruction example. In *Proceedings of IAS-8*, pages 96–103, Amsterdam, 2004. 10-13 March 2004.

[8] L. Chase, R. Rosenfeld, A. Hauptmann, M. Ravishankar, E. Thayer, P. Placeway, R. Weide, and C. Lu. Improvements in language, lexical, and phonetic modeling in Sphinx-II. In *Proceedings of the Spoken Language Systems Technology Workshop.* Morgan Kaufmann Publishers, 1995.

[9] C. Fellbaum. *WordNet: An electronic lexical database.* MIT Press, Cambridge, MA, 1998.

[10] N. Green and J. F. Lehman. Compiling knowledge for dialogue generation and interpretation. Technical Report CMU-CS-96-175, School of Computer Science, Carnegie Mellon University, 1996.

[11] N. Green and J. F. Lehman. An integrated discourse recipe-based model for task-oriented dialogue. *Discourse Processes*, 33(2), 2002.

[12] J. Harland and M. Winikoff. Agents via mixed-mode computation in linear logic: A proposal. In *Proceedings of the ICLP'01 Workshop on Computational Logic in Multi-Agent Systems (CLIMA-01)*, Paphos, 2001.

[13] R. Jackendoff. *Semantic Structures*. MIT Press, Cambridge, MA, 1990.

[14] S. Larsson, P. Ljunglöf, R. Cooper, E. Engdahl, and S. Ericsson. GoDiS: an accomodating dialogue system. In *Proceedings of ANLP/NAACL-2000 Workshop on Conversational Systems*, Seattle, 2000.

[15] S. Lauria, G. Bugmann, T. Kyriacou, and E. Klein. Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38:171–181, 2002.

[16] K. F. Lee. *Automatic Speech Recognition: The development of the SPHINX system*. Kluwer Academic Publishers, Boston, MA, 1989.

[17] E. Levin, S. Narayanan, R. Pieraccini, K. Biatov, E. Bocchieri, G. Di Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, M. Rahim, P. Ruscitti, and M. Walker. The ATT-Darpa communicator mixed-initiative spoken dialog system, 2000. Retrieved from citeseer.ist.psu.edu/576288.html.

[18] R. Lewis. *An Architecturally-based Theory of Human Sentence Comprehension*. PhD thesis, Carnegie Mellon University, 1993.

[19] D. Lonsdale. Leveraging analysis operators in incremental generation. In *Analysis for Generation: Proceedings of a Workshop at the First International Natural Language Generation Conference*, pages 9–13. Association for Computational Linguistics, 2000.

[20] D. Lonsdale. An operator-based integration of comprehension and production. In *LACUS Forum XXVII*, pages 123–132. Linguistic Association of Canada and the United States, 2001.

[21] D. Lonsdale and C. A. Rytting. An operator-based account of semantic processing. In *The Acquisition and Representation of Word Meaning*, pages 84–92. European Summer School for Logic, Language, and Information, 2001.

[22] D. Lyons and M. A. Arbib. A formal model of computation for sensory-based robotics. *IEEE Transactions on Robotics*, 5(3), 1989.

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

[24] G. Nelson, J. F. Lehman, and B. E. John. Experiences in interruptible language processing. In *Proceedings of the 1994 AAAI Spring Symposium of Active NLP*, 1994.

[25] G. Nelson, J. F. Lehman, and B. E. John. Integrating cognitive capabilities in a real-time task. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, Atlanta, GA, 1994.

[26] A. Newell. *Unified Theories of Cognition*. Harvard University Press, 1990.

[27] R. D. Rees. Investigating dialogue managers: building and comparing FSA models to BDI architectures, and the advantages to modeling human cognition in dialogue. BYU Department of Physics: Honors Thesis, 2002.

[28] R. Rubinoff and J. F. Lehman. Natural language processing in an IFOR pilot. In *Collected Papers of the Soar/IFOR Project: Spring 1994*. CMU, 1994.

[29] C. A. Rytting and D. Lonsdale. Integrating WordNet with NL-Soar. In *WordNet and other lexical resources: Applications, extensions, and customizations*, pages 162–164. North American Association for Computational Linguistics, 2001.

[30] J. Schmid, T. Kollar, E. Meisner, V. Sweetser, D. Feil-Seifer, C. Brown, B. Atwood, J. Turner, D. Calrese, S. Cragg, H. Chaudhary, and M. Isman. Mabel: Building a robot designed for human interaction. In *Proceedings of the AAAI Workshop on Robotics: AAAI Technical Report WS-02-18*, pages 24–31. AAAI Press, 2002.

[31] M. Walker, R. Passonneau, and J. Boland. Quantitative and qualitative evaluation of Darpa communicator spoken dialogue systems. In *Proceedings of the Association for Computational Linguistics*, 2001.

[32] W. Xu and A. Rudnicky. Task-based dialog management using an agenda. In *ANLP/NAACL 2000 Workshop on Conversational Systems*, pages 42–47, 2000.