A Context Sensitive Natural Language Modality for an Intelligent Room

Michael Coen, Luke Weisman, Kavita Thomas, and Marion Groh

MIT Artificial Intelligence Lab
545 Technology Square
Cambridge, MA 02139
{mhcoen, luke, marion}@ai.mit.edu, kavita@cs.cmu.edu

Abstract. This paper describes the design and implementation of a natural language interface to a highly interactive space known as the Intelligent Room. We introduce a data structure called a *recognition forest*, which simplifies incorporation of non-linguistic contextual information about the human-level events going on in the Intelligent Room into its speech understanding system. This aim of using context has been to allow multiple applications—all of which support a relatively high degree of natural syntactic variability—to run simultaneously in the Intelligent Room.

1 Introduction

This paper describes the design and implementation of the natural language interface to the MIT Artificial Intelligence Lab's Intelligent Room Project [3,5]. The Intelligent Room explores natural interaction with embedded computational systems. It has a host of computer vision and speech understanding systems that connect it to ordinary, human-level events occurring within it.

In this paper, we are concerned with the overall design and implementation of the Intelligent Room's support for natural language interactions. The main feature of this is the *recognition forest*, a linguistic data structure that simplifies incorporation of contextual information into the room's speech understanding system and allows the room's multiple applications to independently access its speech modality. Our motivation for using context is: to help manage the combinatorial explosion in processing time that followed the incorporation of natural syntactic variability into our speech recognition system; to allow for the incorporation of non-linguistic information into linguistic contextual model; to disambiguate diectic references; and to provide spoken language input to coexisting, independent applications.

In this paper, we present the recognition forest as a useful tool for creating spoken language interfaces to intelligent, interactive spaces. We will also discuss how it

---

This material is based upon work supported by the Advanced Research Projects Agency of the Department of Defense under contract number F30602—94—C—0204, monitored through Rome Laboratory.

contributed towards satisfying the goals that shaped the design of the Intelligent Room's natural language interface, including:

1. To support unimodal speech interactions, i.e. interactions that do not tie the user to a keyboard, mouse or display.
2. To leverage off very strong notions of context inherent in room applications to allow for better speech understanding.
3. To allow multiple speech-enabled room applications to coexist without a heavyweight central controller.
4. To provide for dynamic sets of recognized utterances.
5. To employ only very shallow linguistic knowledge and representations.

The decision to minimize the amount of linguistic knowledge contained in our system was made to facilitate the room's infrastructure and application development by a wide range of people, particularly computer science undergraduates who have no formal exposure to computational linguistics. Our system needed to be accessible by all researchers in the project, regardless of their background. We believer that many of the issues discussed in this paper will remain useful when applied to systems with more sophisticated linguistic representations. Given the increasingly widespread interest in highly interactive, computational environments [7], many other designers and implementers will be faced will similar challenges, and we hope our approach will be generally useful for other systems.

Over the past three decades there have been significant research efforts devoted to the development of natural language interfaces. We divide these into three distinct classes based on the modality chosen for natural language interaction. The first consists of text-only dialog systems, such as SHRDLU [12], Lunar [13], and the multitude of database query systems, such as START [8]. In these systems typewritten text is used for input and output. With the advent of improved speech recognition and synthesis, efforts were made to integrate these technologies into natural language dialog systems—our second category—such as those in [8,11,14]. However, with these the user is still expected to interact with the system at a terminal where the display of queries and recognition results are perused and then potentially disambiguated by the user. Many of these systems also make use of "push-to-speak button," in order to allow the user to signal the speech recognizer that the next utterance is to be treated as input to the system. The final class of systems is those that operate only in the speech modality, such as SUNDIAL [1], and Jupiter [3], two telephone communication systems for answering domain specific queries.

Our approach represents a significant departure from those described above. The choice of the ability to operate in a voice-only modality separates it from the text-only and mixed voice and screen systems. This departure requires an emphasis on careful planning and structuring of dialog to take advantage of speech while also overcoming some of the difficulties inherent in the speech modality. Our approach is different from that of the other voice-only language projects in that: we are not tackling applications that monopolize the user's attention or require excessive word knowledge; we allow people interacting with the Intelligent Room to readily change application contexts; we place more emphasis on speech recognition in noisy, multiuser environments where people are primarily talking to one another; we seek to plan interactions with minimum intrusiveness; and we provide an interface to multiple applications simultaneously. Our system is not intended to be task or domain specific; it is used in the same way as a keyboard and mouse—a general input device simultaneously used by many applications. It is a tool, not an end in itself.

In the next section we briefly motivate and describe the Intelligent Room as the platform and motivation for our work. Next, we present a user's perspective of the Intelligent Room. In section 4 we outline the room's computational architecture and linguistic systems. Then we give several representative linguistic interactions illustrating the concepts we have described. Finally, we discuss limitations inherent in our approach and possible remedies.

2 The Intelligent Room

We now proceed to briefly describe the Intelligent Room as the platform for our research. More in depth discussion of the room can be found in [6] and details of its multimodal resolution are contained in [5].

The Intelligent Room is a research platform for exploring the design of intelligent environments [7]. The Intelligent Room was created to experiment with different forms of natural, multimodal human-computer interaction (HCI) during what is traditionally considered non-computational activity. It is equipped with numerous computer vision, speech and gesture recognition systems that connect it to what its inhabitants are doing and saying. The motivation for researching intelligent environments is to bring computation into the real, physical world. The goal is to allow computers to participate in human-level activities that have never previously involved computation and to allow people to interact with computational systems the way they would with other people: via gesture, voice, movement, and context.

The Intelligent Room is a space populated by computer controlled devices; these include overhead LCD projectors and displays, audio/visual multiplexers, VCRs, drapes, blinds, stereos, steerable video cameras, etc. The video cameras are used by the room's computer vision systems. These vision systems are detailed in [6], but of relevance here are the following: a person tracking system that can locate people in real-time as they move about the room; gesture recognition of both finger and laser pointing on either of the room's projected LCD displays; and a system that specifically notices when people sit down on particular pieces of furniture.

Other research in intelligent environments [2,9,10] has focused more on development of computer vision and other sensing technologies at the expense of linguistic interactions. We believe, however, that language is fundamental to having meaningful and complex interactions with these sophisticated, interactive spaces. In particular, we are interested in speech understanding systems that function more like a language modality than a voice simulation of a keyboard or mouse. Although the Intelligent Room's current linguistic systems require a great deal of development before they near this goal, we believe our approach is an extensible first approximation.

3 User Interactions

People in the Intelligent Room wear wireless lapel microphones that transmit to the speech understanding system described below. By default, the room ignores the spoken utterances of its inhabitants, which are generally directed to other people within the room. This state is known as "the room being asleep."[1] To obtain the room's attention, a user stops speaking for a moment and then says the word "Computer." The room immediately responds with an audible, quiet chirp from an overhead speaker to indicate it is paying attention. The user then has a two second window in which to begin speaking to the room. If the room is unable to recognize any utterances starting within that period, it silently goes back to sleep until explicitly addressed again. However, if what the user says is recognized, the room responds with an audible click and then under most circumstances it returns to sleep. This hands- and eyes-free style of interaction coupled with audio feedback allows a user to ignore the room's computational presence until she explicitly needs to communicate with it. There is no need to do anything other than preface spoken utterances with the cue *Computer* to enable verbal interaction. Thus, a user can interact with the room easily, regardless of her proximity to a keyboard or monitor. Additionally, explicitly cueing the room minimizes the likelihood that extraneous speech or noise will incorrectly trigger a recognition event. Importantly, it also allows detection of non-recognition events, i.e. times when the room is not able to understanding something the user is explicitly trying to convey. In the event the room erroneously wakes up due to an incorrect recognition event, it will either go back to sleep automatically when the two second window expires or the user can explicitly tell it to "go to sleep" upon hearing the wake-up chirp.

When the room is awake, it is listening for a specific set of utterances contained in its recognition forest, a data structure described in the next section. Under appropriate circumstances, users can also freely and continuously dictate expressions to the room unconstrained by any grammar or rule set. This capability is used, for example, during information retrieval queries (such as a web search) for which it is unreasonable to expect that the room's grammars already contain the sought after phrase. In these interactions, the room repeats the final utterance back to the user to verify correct recognition. The lag time between user speech and room verification is extremely small, and this mode of interaction has proved to be quite useful provided input is short in length. We note here that the room is responsible for switching between the constrained and diction speech modes; users do not explicitly change this state.

The room can also remain awake listening for utterances. Someone intending on a prolonged series of verbal interactions can simply tell the room to "stay awake." The room continues to provide an audible click after recognized statements, but these statements no longer need to be preceded by the spoken *Computer* cue. In addition, the room can wake itself up if it expects an utterance from the user for some reason. For example, when the room asks the user a question, it will stay awake for several

---

[1] The room's vision systems continue to respond to users even when it is not listening for their verbal input.

seconds waiting for an answer. If that period ends without a response being given, the room provides an audio timeout signal to indicate that it is going back to sleep.

4 Speech Understanding

In this section we present the Intelligent Room's speech understanding system. We begin with a discussion of the interaction between the software agents that control the operation of the room and the room's set of recognition grammars.

The Intelligent Room is controlled by a modular system of approximately 100 distinct, intercommunicating software agents that run on several networked workstations. These agents' primary task is to connect various components of the room (e.g., vision and speech recognition systems) to each other as well as to internal and external stores of information (e.g., a person locator or an information retrieval system). Essentially, these agents are intelligent *computational glue* for interconnecting all of the room's components and moving information among them.

The Intelligent Room listens for continuous speech utterances contained in a forest (or set) of multiple grammars, which we call the recognition forest. Each grammar in the recognition forest is created by one of the room's software agents (see Figure 1), which receives notification when any utterance contained in one of its grammars is heard. Agents do not necessarily know nor need to know of any other grammars or agents. (We note for clarification that a single agent is allowed and actually encouraged to create multiple grammars.) The notification message for a recognized utterance contains a parse tree, which the agent can manipulate to determine its content. In the recognition forest, a grammar is called "active" if the room is currently listening for it and "inactive" otherwise. Active grammars are rank ordered in terms of their expected likelihood of being heard.

Fundamental to our design is that all of the grammars must be constrained to highly specific contexts among which some component of the Intelligent Room is capable of distinguishing. Instead of keeping a single enormous recognition grammar active, the room collectively keeps subsets of small grammars active in parallel, given what it currently expects to hear. The key assumption here is that certain types of

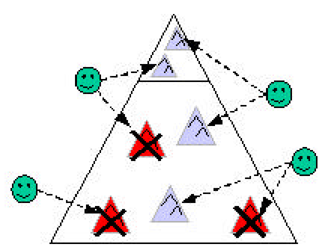

Fig. 1. Software agents creating the forest of context free recognition grammars. Each small triangle represents a grammar, and each face represents a software agent. Active grammars are lightly colored. Inactive grammars are crossed out. The uppermost, demarcated region contains universal grammars

utterances are only likely to be said under particular circumstances. These may be related to where someone is spatially, the history of her previous interactions, how she is gesturing, what devices in the room are doing, etc. At the simplest level, this can range from the implausibility of someone saying "stop the video," when none is playing, to more complex dependencies, such as the meaninglessness of asking "What's the weather there?" if no geographic entity has somehow been brought to the room's attention.

The context-dependency of these grammars is not contained within the linguistic formalism itself, which allows us to use an extremely simple representation. Rather, the room's software agents are responsible for setting and modifying the activation states of grammars they create based on whatever information the room's other software agents can provide about current goings on in and state of the room. (See Figure 2.) For example, if the room starts showing a video clip, the agent that controls the showing of videos activates the grammars that involve VCR operation. When the clip stops, these grammars are in turn deactivated. More interesting cues can involve the location of someone inside the room. For example, the fact that someone has moved near an interactive displayed map is sufficient reason for the room to pay increased attention for spoken utterances involving geographic information. However, until that cue is received, it seems quite reasonable not only for the room to ignore such requests but to not recognize them at all especially given the error rate of current speech recognition technology. We note there may be cases where this is inappropriate; for example, the room might alternatively need to recognize out-of-context utterances in order to provide guidance to a user. We are investigating techniques for dealing with this, such as having the room iteratively broaden the set of active grammars and proactively offer assistance in case the user's speech is not being recognized.

When users shift to a new application context, the system lowers the relative rankings of and eventually deactivates grammars from the previous contexts according to a least-recently-used strategy. Thus, agents need not explicitly deactivate all of their grammars nor even know all appropriate circumstances for doing so.

Notions of context can also help adjust expected probabilities of utterances. Even

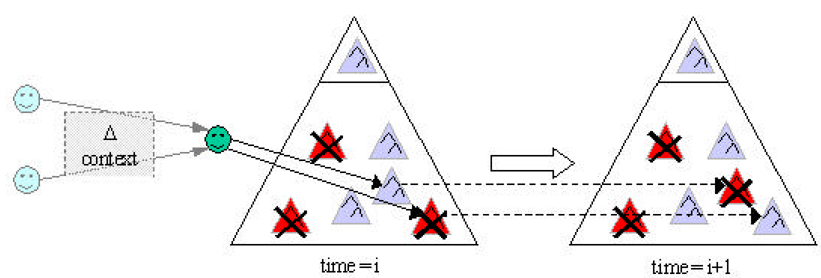

Fig. 2. A transition in the forest of grammars. An agent can activate and deactivate its grammars based on context changes in the Intelligent Room. Notification of context changes comes in the form of messages from other room agents.

in systems where all utterances are valid at all times, it is generally not the case that all utterances are equally likely at all times. For example, tracking context can help disambiguate the output of bigram-based speech recognition systems that return the probabilistically weighted N-best set of utterances for each recognition event. We used this scheme to process the results returned by the Galaxy System [14], which was the first speech recognition system used in the early days of the Intelligent Room.

We have found it useful to have several different notions of ongoing room context for determining which grammars are active at any given moment. However, no single agent defines "the context" but rather the context is a product of many loosely connected entities. In ranked order, these consist of the following:

1. Always active grammars – These are for low-level control and providing feedback to the room. These grammars allow direct manipulation of room state; we have found it essential for users to feel they control the room's physical infrastructure if they are to feel comfortable interacting with it. These grammars also allow the manual adjustment of various room parameters in the event of incorrect data from one of the room's input modalities.
2. Context shifting grammars – These are explicit cues for the room to change its context. They generally start new room activities and automatically lead to changes in the contents of the next two categories.
3. Current applications' grammars — These are application specific verbal interactions with the room given its current state. The room frequently modifies these grammars while it is running.
4. Previous applications' grammars — These are for interacting with previously run applications. These are particularly useful in case of inadvertent or incorrect context shift or if some device failed to respond appropriately and must be corrected via verbal interaction. Backgrounded tasks often have grammars here so they can be quickly recalled and resumed.

Agents can also modify the structure and content of an extant grammar. This ability is used currently only for inserting and deleting noun phrases to reflect newly obtained information. This can be gotten from: the user verbally dictating new phrases to the room; mechanical extraction from other sources, (e.g., anchor link text in web pages); or the room applying machine learning techniques to augment its vocabulary (as discussed below).

Another advantage of the distributed nature of the grammars is individual agents can monitor their small piece of the overall context in a very simplistic fashion. For example, the VCR agent can pay attention to events only relevant to knowing whether the VCR is part of the context or not, and modify its associated grammars accordingly. This avoids the problem of clearly defining the overall context, and also eliminates the need for control logic for deciding which grammars should be active when.

Furthermore, our approach allows for a natural resolution of many types of diectic reference.

Underlying Speech Technology and Computational Complexity

For processing spoken utterances, we use IBM's ViaVoice speech recognition system. ViaVoice is a commercially available system primarily used for continuous speech dictation. This, with its relatively low word accuracy for single word and short utterances, would have been an intolerable speech interface to the room. However, ViaVoice also supports explicit construction of continuous speech, context-free recognition grammars, which allow for much higher degrees of recognition accuracy. Via its Java interface, it also provides control over low-level aspects of its behavior to external applications, which makes it ideal for incorporating into other systems.

We wanted the Intelligent Room's recognition grammars to be reasonably unconstrained. In particular, we wanted to allow people to interact with the room without memorizing scripts of recognized utterances or lists of permissible syntactic constructions. However, we found that as our grammars became increasingly large, speech recognition accuracy fell correspondingly. As room grammars started to support more than a few thousand individual utterances, recognition accuracy dropped below acceptable levels.

There is a clear tradeoff between making the room's recognition grammars sufficiently large so that people can express themselves somewhat freely versus making the grammars small enough so that the system runs with high accuracy and in real-time. Thus, we decided to make use of the natural context specificity in room applications so that agents could dynamically activate different subsets of grammars depending on the context of the activity within the Intelligent Room.

5 Sample Interactions

The Intelligent Room supports a variety of narrow application domains, all of which can run simultaneously. The collection of all these domains in turn gives an extremely broad and flexible 'domain' encompassing a wide range of tasks. The distributed, independent control of the recognition forest allows for this—there is no need for a centralized controller. The room's applications range from simple voice control over physical devices to more complex multimodal scenarios involving position, gesture tracking and spoken dialog. We first outline these domains and then present two applications in more detail:

1. Manual control over devices – These include the Intelligent Room's lights, blinds, drapes, VCRs, video displays, stereo components, etc.
2. Manual interaction with modal subsystems – We have found it extremely useful to have direct verbal interactions with the room's modalities. These can be used to gather information about what the room is observing, to modify internal representations of its state, or to correct a perceptual error. It is also of enormous benefit to be able to verbally interact with the room's vision systems while developing or debugging them, because it is generally impossible to manually interact with them at a workstation while remaining in the camera's foveal areas.
3. Information access – There are many types of these interactions, including web browsing, weather reporting, accessing an online video collection, querying Haystack (a personal

information manager), and the information retrieval system described below. The room also functions as a spoken language front-end to START, a natural language query database [9].

4. Presentation manager – This allows the Intelligent Room to assist in multimedia presentations and is a demonstration of the room's information management capabilities. A lab tour guide agent that uses this application for presenting a broad overview of our laboratory's research to visitors has been previously described in [5].
5. Command post – This application provides the means to test full integration of all our modal subsystems and to experiment with different techniques for performing multimodal reconciliation. It is a mock command center for planning hurricane disaster relief.

We now present two primarily linguistic interactions with the Intelligent Room. They are annotated with the changes made to the recognition forest to reflect the course of ongoing interactions. The first of these is the above-mentioned command post. It makes use of two interactive projected displays that respond to finger pointing gestures. The second interaction is a primarily unimodal dialog that allows users to interactively refine a document retrieval query.

In the following interaction the user is attempting to plan disaster relief for a hurricane in the Virgin Islands.

Command Post:

User: *"Computer, stay awake."*
[The room will now listen for utterances without requiring they be prefaced by the word Computer.]
[The person's approach of projected displays causes the room to pay attention to statements involving them. This is illustrated in Figure 3.]
User: *"Show me the Virgin Islands."*
Room: "I'm showing the map on the display next to you."
[Room shows map on video display closest to the user.]
[Room activates grammars associated with the map.]
[User now points with his finger at St. Thomas.]
[Room adds nouns (such as city names) relating to St. Thomas to active grammars.]
[Room now considers St. Thomas the default geographical entity.]
User: *"Zoom in."*
[Room zooms in to St. Thomas.]
User: *"How far away is Hurricane Marilyn?"*
Room: "The distance between Hurricane Marilyn and the city of Charlotte Amalie located in St. Thomas is 145 miles."
User: *"Where's the nearest disaster field office?"*
[Room highlights them on the map.]
Room: "The St. Thomas DFO is located one mile outside of Charlotte Amalie. There is a new weather forecast available. Do you want to see it?"
[Room activates a grammar containing possible expected responses for this question.]
User: "Yes, show me the satellite image."
[Room displays image on adjacent projected display.]
[Additional weather grammars are now activated.]

The following interaction is an example of the user dynamically switching contexts while in the midst of interacting with the Intelligent Room. The room has an information retrieval system that helps users refine their queries by asking them questions that maximally disambiguate it. These questions are determined by searching a document tree obtained from the Alta Vista web search engine for phrases with the highest entropy levels.

Information retrieval:

User: *"Computer, I need information."*
[Because user didn't specify a topic, room will ask for it to be dictated.]
Room: "About what? Please tell me."
User: *"Software agents."*
Room: "Is software agents correct?"
User: *"Yes."*
[Room will now remember this phrase in the context of information retrieval to allow recognition of it in the future. Other agents might eavesdrop and also add this word for their own purposes, in an attempt to anticipate the user's eventual needs.]
Room: "Thank you. Please wait a moment…Are you interested in HCI, Research & Indexes, or General AI?"
[Room activates a grammar of expected responses based on these topics.]
User: *"I'm interested in HCI."*
[User can respond with continuous speech even though the phrase "H C I" was not in any room grammar before this interaction, since it was just added.]
[Above constructed response grammar is deactivated.]
Room: "Okay, are any of these documents of interest? I will put them on the left display."
[Room displays document list on wall.]
[Room activates a response grammar that incorporates the titles of displayed documents.]
User: *"Please move them to the display near the window."*
[Room does so]

The above examples illustrate how the room manages the recognition forest via data from both its perception subsystems and from expectations of what the user is likely to say in a given situation. This allows the room to approximate natural linguistic interactions with the user. Of course, it is still somewhat stilted, and the user cannot make complete non-sequiturs. Overall, however, the main problem we have is convincing the user to push the limits and speak naturally, rather than linguistically downsizing and second-guessing the room's capabilities, but this fault is prevalent in most, if not all, current speech understanding systems.

6 Achieving Design Goals

We review the design goals presented in the first section, considering not only how well they were achieved, but possible remedies for where our approach was unsuccessful.

A language modality

A key aspect of the Intelligent Room is for an occupant to have full access to all of the room's computational power regardless of where in the room she is. She should not need to type at a particular keyboard nor interact with a particular display to interact with the room, and in fact, the room does not have a keyboard or mouse within it.

Therefore, we decided that in many circumstances speech interactions had to be unimodal. It could not be the case that the room would need to display candidates for recognized utterances, thereby allowing the user to select among them or to disambiguate didactic references. This is contrasted with the approach of [11,14] where users provide critical feedback during the recognition process via a graphical user interface. Our approach of having the room ask the user when it is unsure of something can be somewhat intrusive, but it is certainly no more so than a graphical interface.

Context sensitivity

We wanted to make use of context in terms of both the room and user's states to be able to both resolve diectic references and control sets of possible utterances and who should receive those utterances. As in most speech understanding efforts, we wanted to support some measure of natural syntactic variability on the part of a person interacting with room. Our intent was to leverage off the well-defined notions of context inherent in the Intelligent Room's application domains to keep the total active grammar size small at any given time. This can be enormously frustrating if the room inappropriately deactivates a grammar to which the user would still like to refer. We are currently exploring techniques for dealing with this, such as reprocessing the spoken audio signal under an iteratively broadened set of grammars.

Non-static recognition sets

We sought to avoid limiting the room to a static set of recognition grammars. It would not have been reasonable to suppose that we could determine everything in advance users would want to say, and it would have made routine tasks like information retrieval difficult, if not impossible.

The ability of agents to change grammars on the fly has proved to be extremely useful, in applications such ranging from web browsing, where link anchor text is captured, to information retrieval, which typically involves an iterative query refinement process. The ability of the room to incorporate user-dictated noun-phrases into its recognition grammars is one of the capabilities that most impresses new users.

On a larger scale, adding agents should be easy. By having individual agents control their own grammars and activation states, agents can indeed be added quickly and without worry as to their disrupting other agents' interfaces.

Simplicity

Finally, we were also interested in employing very shallow linguistic knowledge during implementation to minimize the knowledge engineering problem. Given that new room applications are being created on a regular basis, it is not possible to build carefully handcrafted linguistic models of expected input. Speech orientated agents have proliferated markedly since our system came up, to a point of fault. Speech is such a natural and easy modality that the temptation to solve all problems with it has distracted us from really striving for a multi-modal system that pays attention to harder to discern inputs, such as gesture.

Future work on the system includes incorporating a machine learning mechanism into the recognition forest so that it can learn the probabilities of individual grammars be used in particular application contexts. We are also interested in learning the transition probabilities among the grammars, to better predict activation states without requiring explicit action be taken by the room's software agents.

8 References

[1] Andry, F., Fraser, N.M., McGlashan, S., Thornton, S., and Youd, J. Making DATR Work for Speech: Lexicon Compilation in SUNDIAL. Computational Linguistics, 18(3), pp. 245-267. 1992.

[2] Bobick, A.; Intille, S.; Davis, J.; Baird, F.; Pinhanez, C.; Campbell, L.; Ivanov, Y.; Schütte, A.; and Wilson, A. Design Decisions for Interactive Environments: Evaluating the KidsRoom. Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments. AAAI TR SS-98-02. 1998.

[3] Chung, G., and Seneff, S. Improvements in Speech Understanding Accuracy Through the Integration of Hierarchical Linguistic, Prosodic and Phonological Constraints in the Jupiter Domain, Proc. ICSLP '98, November 1998.

[4] Coen, M. A Prototype Intelligent Environment. In Streitz, N., et al. (Eds.), Cooperative Buildings - Integrating Information, Organization, and Architecture. Proceedings of the First International Workshop on Cooperative Buildings (CoBuild'98). Lecture Notes in Computer Science. Springer: Heidelberg. 1998.

[5] Coen, M. Building Brains for Rooms: Designing Distributed Software Agents. In Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence. (IAAI97). Providence, R.I. 1997.

[6] Coen, M. Design Principles for Intelligent Environments. In Proceedings of The Fifteenth National Conference on Artificial Intelligence. (AAAI98). Madison, Wisconsin. 1998.

[7] Coen, M. (ed.) Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments. AAAI TR SS-98-02. 1998.

[8] Federico, M. and Vernesoni, F. "A speech understanding architecture for an information query system". Proceedings of EUROSPEECH 95, Madrid, Spain, 1995.

[9] Katz, B.. Using English for Indexing and Retrieving. In Artificial Intelligence at MIT: Expanding Frontiers. Winston, P.; and Shellard, S. (editors). MIT Press, Cambridge, MA. Volume 1. 1990.

[10] Mozer, M. The Neural Network House: An Environment that Adapts to its Inhabitants. Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments. AAAI TR SS-98-02. 1998.

[11] Stock, O., Carenini, G., Cecconi, F., Franconi, E., Lavelli, Magnini, B., Pianesi, F., Ponzi, M., Samek-Lodovici, V., and Strapparava, C.. ALFRESCO: Enjoying the Combination of Natural Language Processing and Hypermedia for Information Exploration. In Maybury, M. (ed.), Intelligent Multimedia Interfaces, The MIT Press, pp. 197-224.

[12] Winograd, T. Five lectures on artificial intelligence. Technical Report, Stanford-CS, Number AI Memo 246, Stanford University, 1974.

[13] Woods, W., Kaplan, R., and Nash-Weber, B. The Lunar Sciences Natural Language Information System: Final Report. Technical Report, Bolt Beranek and Newman, Number 2378, June 1972.

[14] Zue, V. Human Computer Interactions Using Language Based Technology. IEEE International Symposium on Speech, Image Processing & Neural Networks. Hong Kong. 1994