

A Multimodal Speech Interface for Accessing Web Pages

Michel Génèreux, Alexandra Klein and Harald Trost

Austrian Society for Cybernetic Studies, Schottengasse 3, A-1010 Vienna, Austria
{michel,alexandra,harald}@ai.univie.ac.at

Abstract

We present an interface for multimodal access to Web pages for German newspapers which integrates spoken and written input, as well as point and click operations and discuss the motivations behind it. As with many systems being developed recently, the speech modality is the main focus of our research. Our system shows new ways of integrating speech and language with classical access methods, and investigates the respective shortcomings and advantages of different combinations. The innovation lies specifically in two areas: the possibility for the user to refer to the *content* of pages, and the real *integration* of semantic content from different modalities. This paper also presents partial results of the project, as well as a fairly detailed analysis of the system's components.

1. Introduction

Multimodal systems offer many advantages over unimodal interfaces: enhanced error avoidance and correction, accomodation of various situations, users, and tasks as well as user preference (Cohen & Oviatt, 1995). With current state of the art in language technology reaching a point where spoken language may be used effectively as input in communication, real multimodal systems can be built. A prominent example for this need is demonstrated by access to the World Wide Web (WWW), which is of growing importance in everyday life.

The system described in this paper provides a solution for this need. It is the first prototype of an evolving system concerned with multimodal access to Web pages. It currently runs on Windows NT 4.0. The system aims at the integration of different modalities for web browsing, but also at providing a fairly comprehensive natural language understanding module which is responsible for the analysis of complex requests by the user. While the functionality of our Graphical User Interface (GUI) part is standard, we provide a flexible natural language understanding module as natural language queries can concern any combination of the following:

- what some particular piece of information refers to (i.e., content)
- how this information is presented and connected (i.e., structure)

Accordingly, we expect a multimodal interface to handle spoken browser commands (e.g., *back*) as well as content queries (e.g., *more about Clinton*) and combinations of the two (e.g.,

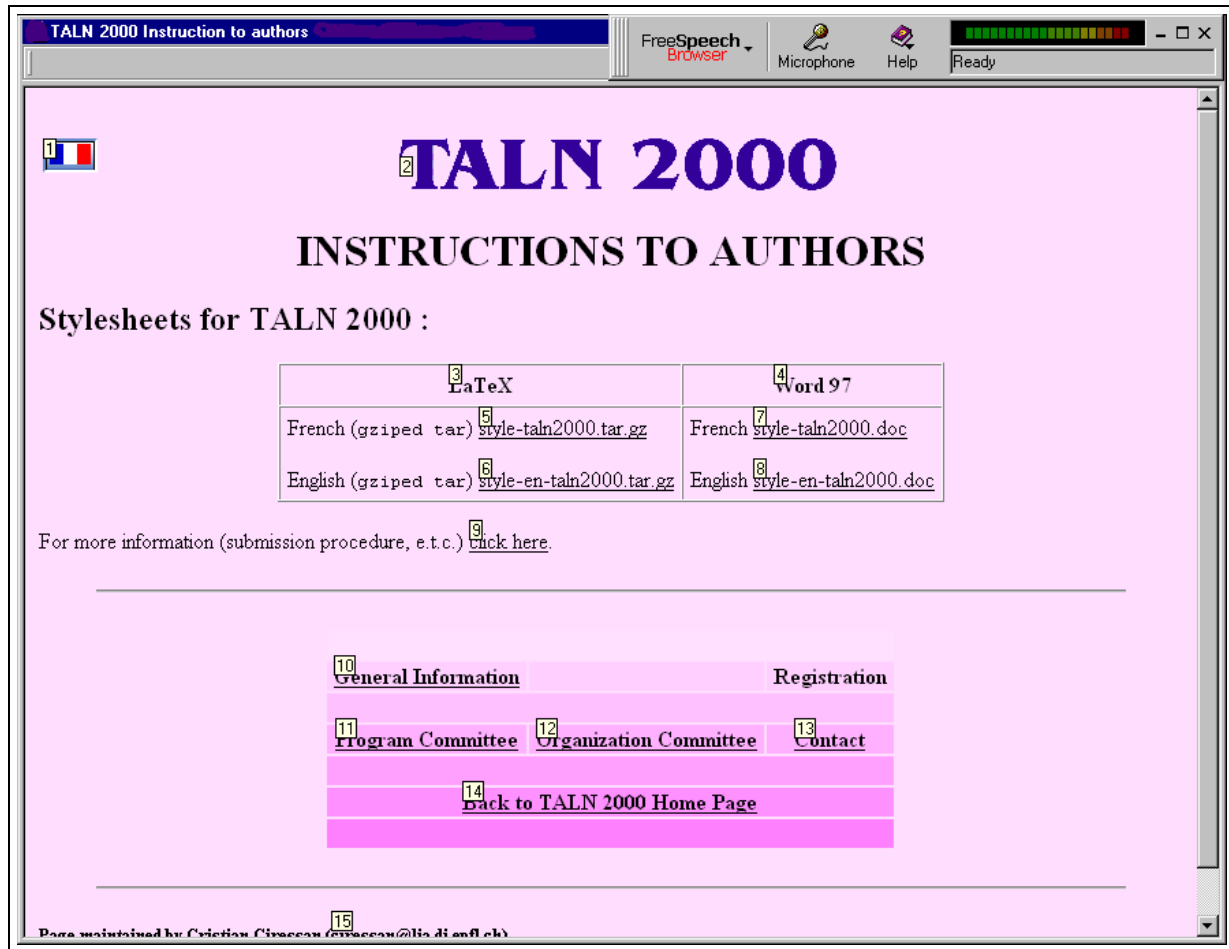


Figure 1: Philips Speech Browser

back to the page about Clinton). Natural language queries are also to be interpreted in the appropriate contexts which are derived from the current communicative situation. The language processing module will have to be able to distinguish whether the queries refer to content, structure or document hierarchy and to provide the appropriate system states, e.g., display text segments, images or entire pages.

This concept differs radically from conventional speech-driven command-and-control systems which offer only the same functionality as GUIs. Such an approach seems to offer little advantage; an example of such system is provided by the Philips Speech Browser system (Philips, 1999). Here are some command examples supported by this system, for the English language:

- (1) Click Forward
- (2) Page up
- (3) Close window
- (4) Go to <NUMBER>
- (5) Go to <LINK>

(6) Help me

As we can see on figure 1, hypertext links are given a number. For example, to activate a specific link, the user can say either (4) *Go to ten* or (5) *Go to General Information*; these are examples of *structure* related commands. We have implemented only a subset of these commands in our system, because we tend to believe, at this point where pre-design experiments have not yet been completed, that for a certain number of these commands, the user might preferably use point and click operations rather than spoken or written text.

Due to the limitations of current speech recognition technology, the vocabulary (and consequently the domain) needs to be limited. A generic speech-driven web browser thus seems ahead of the current technological possibilities. As this research project is more concerned with the possible interaction types which are feasible and usable in integrating speech understanding for browsing and search in interaction with the web, it seems legitimate to choose a more restricted application. The system provides access to German-speaking newspapers that are available on-line. Currently, many newspapers in German offer free on-line access to (most of) their daily news via the WWW. Additional material is offered by a number of news agencies and news providers. This domain is rich enough to provide an adequate testbed and should at the same time be accessible with a limited vocabulary.

2. Areas of research

Empirical evidence concerning multimodal interaction with web documents

First and foremost, it is crucial to characterize precisely what kind of interaction takes place between a user (here a web surfer) and a computer, especially when spoken input is possible. Empirically founded pre-design studies are being carried out to provide important insights about the suitable role of speech in a multimodal system for accessing web pages. While technical improvements in speech recognition have led to a number of applications for speech access to information sources there is a lack of research on the proper use of this modality. However, (Oviatt & VanGent, 1996) reports that:

- Speaking more slowly (55%) and
- Shifting input modality (50%)

were users' most effective means of resolving errors, while some experiments tend to show, among other things, that¹:

- An effective speech interface is more likely to result if the speech interface is designed from scratch based on natural dialog studies
- One way to create naturalness in a human-computer dialog is to maintain conversational context
- Users are more likely to accept a system that makes relatively frequent errors if the system provides users with frequent, but unobtrusive, feedback that moves the conversation forward

¹Taken directly from (Yankelovich, 1997 to appear).

These results, combined with our own pre-design experiments, should provide answers to a number of interesting issues relating to what kind of deictic references users would want to make to combinations of graphical and textual material on the web: would they want to do more sequential actions, point to identify, then issue a command to be applied to the selection, or do both at once².

Generally, the contribution of speech to the contrastive functionality³ of the multimodal interface is being examined. We expect that speech will play an important role in the immediate access (triggered by the user's individually formulated request) to information (deeply) embedded in hierarchical hypertext structures to complement point-and-click operations which rely on the surface document structure.

Integrating and coordinating different modalities

Point and click operations are well known, but they should be integrated into a multimodal interface. In general, the project yields insights concerning appropriate methods for analyzing spoken utterances in the context of a multimodal environment.

Coordination of different modalities is a challenging matter. First, information should circulate freely among modalities. For example, a speaker should have access (or should be able to refer) to the content (or result) of a previous point and click operation. Second, multimodal parsing is required when more than one modality is used to express a single request. For example, the user might want more information (by formulating a spoken request) on a highlighted text (point and click operation). Unification-based multimodal parsing (Johnston, 1998) is an interesting framework (see further).

Multimodal parsing

In multimodal parsing, the system must build a single request from the meaning of different modalities, and put some constraints on those modalities. For example, the user could make a request about a fragment of highlighted text. Based on works by (Johnston, 1998) and ongoing work by the first author, this is done by means of unification of feature structures along with general constraints on modalities. To sketch this approach briefly, it should first be pointed out that multimodality would also require multi-dimensional parsing and that each event is represented by a feature structure. Then, the combination of feature structures is performed by an integration rule schema, from which the system can draw the resulting action to perform. To give a flavor of how this can be achieved in our web-browsing system, let's look at a very simple example in which a user asks for more information on a highlighted news text. Figure 2 shows the basic integration rule schema that gives the resulting feature structure describing the action to be carried out by the system. Constraints must also be applied on feature structures:

- The time of the speech [6] must practically overlap the time of the highlighting [9]
- The resulting modality [2] is based on the combination of modalities [5] and [8]
- As shown, predicates of both modalities must unify
- The content [1] is a list of all newly instantiated arguments from [4] when unifying with [7]

²Michael Johnston, personal communication.

³The use of different modalities as a means to resolve understanding errors.

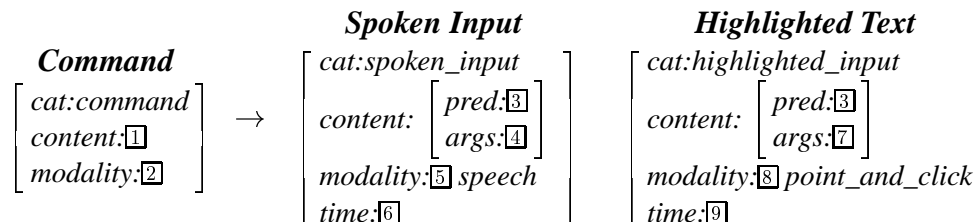


Figure 2: Schema combining modalities

It follows from those constraints that a request of the form *Wer gibt Clinton etwas ?* (Who gives Clinton something ?) on a highlighted text such as *Monica gibt Clinton ein Buch* (Monica gives Clinton a book) will result in a list of arguments of the form $\boxed{1}$ [Monica, Buch]. In accord with the predicate being examined, the system then may choose to make a search on one or more arguments, using conjunction or disjunction, or simply provides available information on the topic(s). This approach is similar with those used in natural language interface for database queries. To be fully integrated in multimodal web-browsing, it would need to be able to handle general requests such as, for example, *Geben Sie mir mehr Informationen über diesen Text* (Give me more information on this text), as well as requests on graphical (e.g. images) components of web pages.

Knowledge representation

Knowledge representation and knowledge management are also important issues in the analysis. Apart from linguistic knowledge, initially no further knowledge sources are present. As the domain is large, it renders any contingent representation of extralinguistic knowledge impossible. Experiments will have to show whether it is useful to include typical entities and relations which occur frequently or which are typical for certain situations. It seems desirable to represent frequently occurring concepts in a wordnet-style hierarchy in order to allow for a broader interpretation of the input query, thereby providing more freedom for the lexicalization of user queries. Our system currently memorizes a certain number of previous actions and results for further use, a feature essential for context interpretation. This relates directly to commands such as *Back to the page about Clinton*.

Controller design

As users are able to input various types of requests in spoken utterances, a powerful interaction control is necessary in order to recognize the user's intent by comparing it to what the system knows about the addressed entities and their relation to each other as well as to the data which are accessible at the specific moment in the interaction. This interaction control module has to determine the action which is required for carrying out the user's request. As a result, either new information is displayed (if a browser command was used, with or without information concerning content), the data which is currently displayed is analyzed according to the request, or more data which is not displayed at present is searched. The user then obtains the appropriate answer for the request as it was interpreted and found by the system. Generally, several types of knowledge sources are needed: domain-specific information and world knowledge as well as linguistic knowledge. The information which has been activated should be stored for further use. The user's request as well as the action taken by the system is recorded in the interaction memory.

Information Retrieval

For matching the required information as it is expressed in the query to the information as it is distributed in the texts, methods taken from information retrieval are used. Since this project is not concerned with the creation of powerful search engine, only a simple search within the current document being displayed in the bottom frame is provided at this stage, but we are aiming at searching through nested pages as well. Improving the search will involve comparing a few statistical approaches for their success (Hochberg *et al.*, 1998 to appear), as well as more linguistically oriented methods such as in (Chandrasekar & Srinivas, 1998) and (Riloff, 1995).

Multimodal generation

The area of output management in a multimodal interface is an area which has not yet received much attention. That is, how to map from content to combinations of, e.g., graphics and text or graphics and speech. A lot of interesting human factors issues of the effectiveness of different modalities in output could be investigated here. We intend to look closely at those issues in pre-design experiments and possibly adapt a generation module based on the experiment results.

3. The system

The project is still in its early phase. We have developed a prototype interface, have adapted a speech recognition component as well as a parsing module. We are moving towards a controller which would handle multimodal parsing. As described in the previous section, information retrieval and multimodal generation are still in the preliminary phase. Figure 3 shows the architecture of the system.

The interface

The core of the system is a two-frame web page. Interaction takes place in the JAVA applet which lies in the top frame, while results appear in the bottom frame. To access the data from the daily newspapers, the interface provides browsing functionality via conventional point-and-click operations plus the ability to process voice commands, supplemented by a text input/output line. Figure 4 shows a preliminary version of the interface. In the example shown, the user has uttered *Ich suche den Sport*, a sentence which was first incorrectly recognized as *Ich suche der Sport*. The controller then looked at the second alternative (the user may choose it himself from the pull-down menu), which was then successfully processed by the PROLOG parser. We show the result of a simple search in the page currently displayed in the bottom frame by highlighting the word *Sport*. It is worth mentioning here that all those actions could have carried out using alternatively voice, keyboard and/or point and click operations.

The system allows users to voice browser commands like, e.g., *zurück zur Hauptseite* 'back to main page' as they are already possible in command-and-control systems. Users will also be able to ask for information contained in the displayed page or other pages within reach in a certain environment. The requested action is then carried out by the system, and the appropriate information is displayed or highlighted.

Next to the button *Suchen*, an input/output line for natural language, similar to the input line used for search engines is supplied. In our interface, however, this line displays the user's spoken request basic semantic interpretation. For transparency in interaction, users need to obtain immediate feedback whether they were understood correctly. Therefore, all utterances

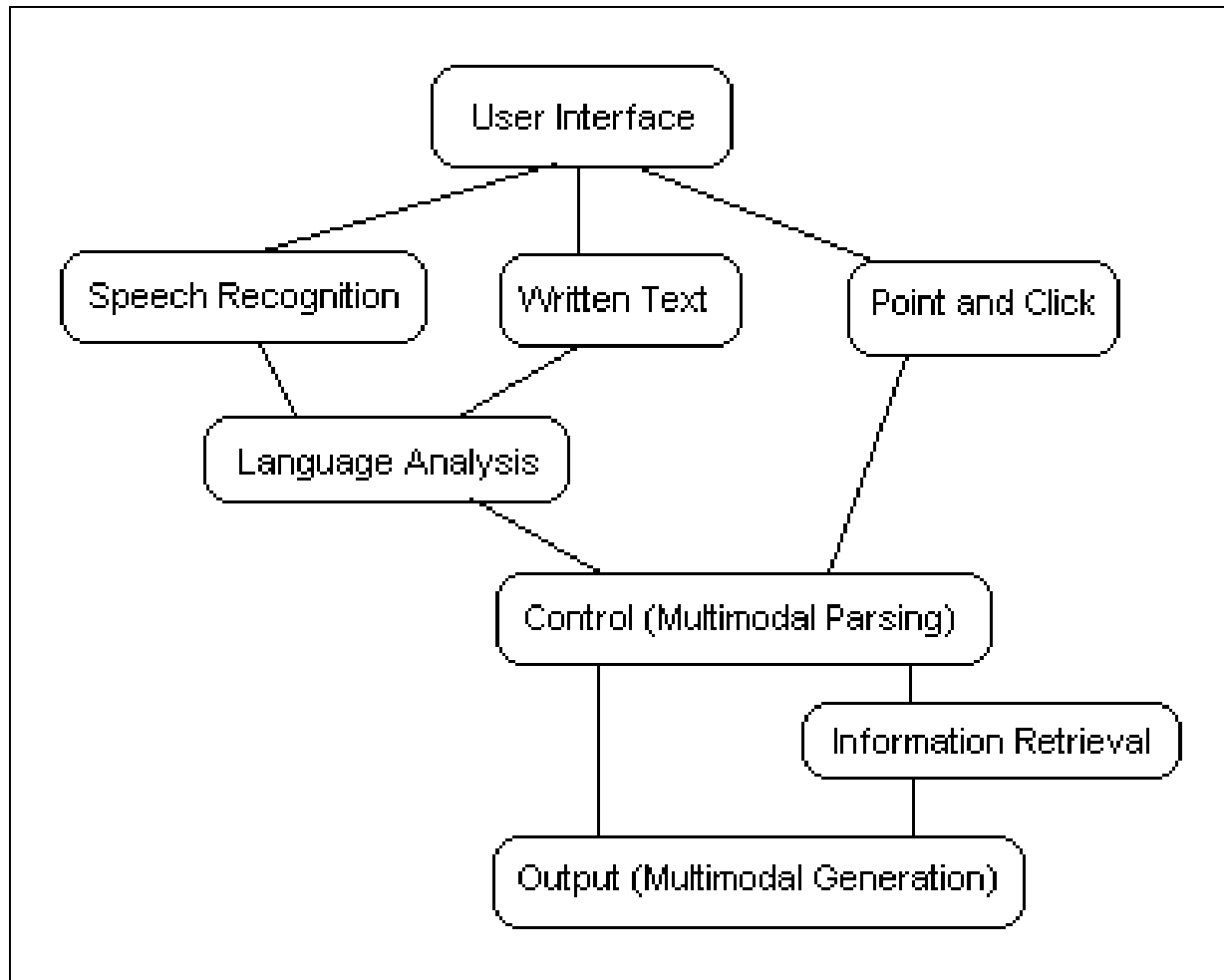


Figure 3: System architecture

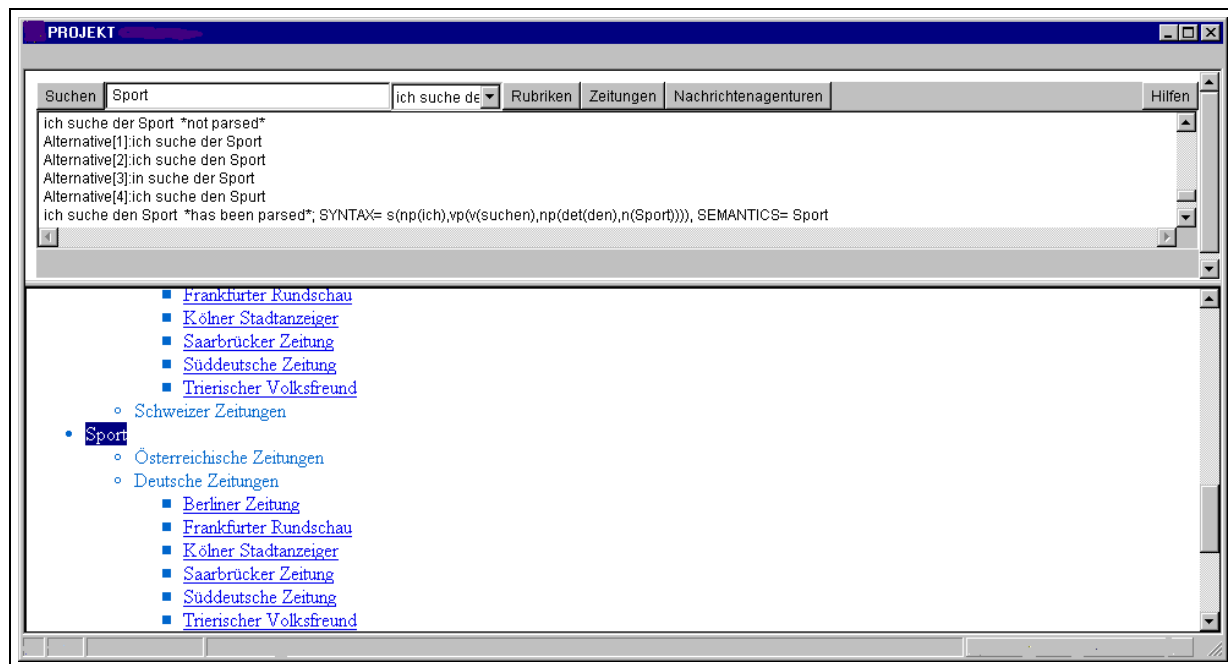


Figure 4: The interface

and their alternatives, syntactic and semantic outputs are displayed as they were recognized by the system. In order to avoid pragmatic ambiguities, the action which was associated with the user utterance will be displayed (in future versions) on the screen while the system carries out its responsive action. In case the user finds the recognition result to be incorrect or changes his/her mind, this line can be used to correct the query via the keyboard. The input/output line can also be used in its own right for written input only. In figure 4, the utterance *Ich suche den Sport* is interpreted semantically, from the DCG grammar, as a research for the single word *Sport* in the newspapers' web pages.

We are currently testing our interface using Wizard-of-Oz experiments, where system action is simulated by human 'wizards'. We are aware that careful design and realization of Wizard-of-Oz experiments is crucial, as speech recognition and understanding errors have to be simulated for obtaining usable results. Yet those experiments are a valuable indicator for usability and corpus assessment.

As the domain will be news texts, web pages in German provided by news agencies, newspapers and TV channels are used. From the large amount of available web pages concerning news from Austria, Germany and Switzerland, we restrict ourselves to the most relevant information sources. Of course, the system architecture is independent from the specific selection of knowledge sources.

Speech Recognition

Requirements are continuous speech recognition and speaker-independence. An extensible vocabulary is also important. Users may want to use abbreviations, so it should be easy to integrate those as well, if they are common. We have chosen to use IBM ViaVoice (IBM, 1999) as the speech recognition system. This system has the big advantage of being coupled with a well-documented interface called JSAPI, for Java Speech Application Programmer Interface. A complete training with ViaVoice lasts approximately 45 minutes. The reader can judge by himself the precision of the speech recognition by looking at the following text, which is the result, in dictation mode, as perceived by the speech recognition component, of a fragment of german text ⁴:

Sprechen sie deutlich, aber ganz natürlich, wären sie dir weis werden. Wären sie ihr persönliches Sprechmuster erstellen oder verbessern, können Sie die Sachsen in diktieren müssen dies aber nicht. Wenn sie später mit ViaVoice arbeiten, müssen Sie jedoch eilig Satzzeichen diktieren.

Figure 5 shows how the JAVA applet handles the PROLOG grammar, creates a recognizer, a dictation grammar and a synthesizer. JSAPI offers a *rule grammar* as well as a *dictation grammar*. A rule grammar recognizes specific linguistic constructions, and uses the rigid Java Speech Grammar Format (Sun_Microsystems, 1998b) to process those patterns. In (Sun_Microsystems, 1998a), dictation grammars are defined as follows:

Dictation grammars come closest to the ultimate goal of a speech recognition system that takes natural spoken input and transcribes it as text. They are used for free text entry in applications.

Dictation grammars basically use a statistical grammar for proposing recognition patterns (see alternatives 1,3 and 4 in figure 4 for spurious readings), so utterances need be post-processed

⁴The user was NOT a native speaker of German. Mismatches are underlined.

```

public class multi_modal extends Applet {
    ...
    public void init() {
        ...
        // load PROLOG interpreter and the grammar
        sp = new SICStus(null,null);
        sp.load("Grammar.pl");

        // create a recognizer matching default locale, add audio listener
        recognizer = Central.createRecognizer(new EngineModeDesc(Locale.GERMAN));
        recognizer.allocate();
        recognizer.getAudioManager().addAudioListener(audioListener);
        recognizer.addEngineListener(engineListener);

        // create dictation grammar
        dictationGrammar = recognizer.getDictationGrammar(null);
        dictationGrammar.addResultListener(dictationListener);

        // commit new grammars, start recognizer
        recognizer.commitChanges();
        recognizer.requestFocus();
        recognizer.resume();

        // create a synthesizer, speak a greeting
        synthesizer = Central.createSynthesizer(new SynthesizerModeDesc(Locale.GERMAN));
        ...
        synthesizer.speak("Ich höre Ihnen zu.");
        ...
    }
}

```

Figure 5: Parsing, speech recognition and synthesizer within the applet

by our parsing module. Because of the rigidity of the rule grammar and the unnecessary work involved when switching the recognizer from one mode to the other, we have chosen to use exclusively a dictation grammar, and it is its output that is processed by the language analysis component.

Language Analysis

In order to understand users' queries, natural language input has to be analyzed morphologically and syntactically. It has to be taken into account that utterances may be incomplete or ungrammatical (according to the standards of written language), therefore it is not useful to employ a rigid grammar. As we can see in figure 4, the speech recognition system may also introduce errors. Parsing is currently performed by a DCG grammar.

In the syntactic analysis, it is useful to employ a simple rule-based parsing mechanism in order to determine whether the utterance or parts of it relate to browsing, structural, or actual content information. This parser may use simple syntactic patterns to extract the structure of the utterance with respect to the different command modes. Methods taken from shallow parsing are particularly useful as they are able to cope with speech effectively. Verbal cues such as particles,

hesitations etc. in communication situations are important for determining the requested type of action. Parsing of voice requests is better when offering feedback to the user in case of failure.

The syntactic analysis reveals the type of action which is assigned to the utterance meaning. Possible instances of actions as they are encountered may refer to the status of the browser or the information content, or combinations of both. The relation between browser status and displayed content defines an interaction state which is used by interaction control in order to enforce coherence of interaction. In order to represent the sequences of actions in a homogeneous form, user's requests as well as changes carried out by the system to fulfill the requests are treated similarly in the interaction protocol.

Control

As the coordination of global interaction between user, interface, knowledge sources, web pages and analysis components requires complex organization, a crucial task is the development of a controller which already:

- handles requests by voice, by written input or by mouse click,
- arranges storage of activated information (Johnston *et al.*, 1997),

is partially responsible for task integration (Grasso & Finin, 1997) such as multimodal parsing, and will ultimately:

- coordinate consultation of knowledge bases,
- be in charge of output selection.

Results provided by the various knowledge sources in natural language understanding must be integrated into a single request for action. They must also be integrated into inputs in other modes.

4. Conclusion and future work

We have implemented a basic system for accessing web pages using different modalities. We have made few assumptions about the behavior of users with a multimodal interface, but instead we have based our design on ongoing experimental testing with regards to users reactions. Two new areas of research are being investigated; how spoken input can be fully integrated into web-browsing and how multimodal parsing can synchronize multimodal integration. These, along with knowledge representation and controller design, will fully provide the user with multimodal *content* access to web-pages. At this point, basic information retrieval techniques are being used while the role of multimodal generation is being more clearly defined for multimodal environment.

Our main concern while building this first version of the interface was to lay the basis for every aspects of a multimodal browser; we have achieved this goal. The system integrates point/click operations with written/spoken input, provides natural language understanding, gives access to content as well as the structure of web pages and offers feedback to the user. Although most components of the system remain to be investigated and implemented in more depth in the long run, we have shown that integrating speech in a multimodal interface is *feasible*.

In the short terms, Wizard of Oz experiments should give us important insights on users' behaviors and lead us to revise our interface to better suit their needs. Because it offers interesting possibilities for integrating modalities, the multimodal parsing methods presented is a long term project. But the results obtained so far lead us to believe that multimodal interface systems shall also prove to be *usable*.

References

- CHANDRASEKAR R. & SRINIVAS B. (1998). Glean: Using syntactic information in document filtering. *Submitted for publication to Information Processing and Management*.
- COHEN P. R. & OVIATT S. L. (1995). The Role of Voice Input for Human-Machine Communication. *Proceedings of the National Academy of Sciences*, **92**(22), 9921–9927.
- GRASSO M. A. & FININ T. (1997). Task Integration in Multimodal Speech Recognition Environments. *Crossroads*, **3**(3), 12–22.
- HOCHBERG J., SCOVEL C., THOMAS T. & HALL S. (1998, to appear). Bayesian stratified sampling to access corpus utility. In *Proceedings of WVLC-6*.
- IBM (1999). *ViaVoice Benutzerhandbuch*.
- JOHNSTON M. (1998). Unification-based multimodal parsing. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, volume I, p. 624–630, Montréal, Québec, Canada: Université de Montréal.
- JOHNSTON M., COHEN P. R., MCGEE D., OVIATT S. L., PITTMAN J. A. & SMITH I. (1997). Unification-based Multimodal Integration. In *Proceedings 35th Annual Meeting of the ACL*, Madrid: ACL Press.
- OVIATT S. L. & VANGENT R. (1996). Error Resolution during Multimodal Human-Computer Interaction. In *Proceedings of the ICSLP*, volume 1, p. 204–207: Philadelphia.
- PHILIPS (1999). *FreeSpeech 2000 Benutzerhandbuch*.
- RILOFF E. (1995). Little words can make a big difference for text classification. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 130–136.
- SUN_MICROSYSTEMS (1998a). *Java Speech API Programmer's Guide*. <http://java.sun.com/products/java-media/speech/>.
- SUN_MICROSYSTEMS (1998b). *Java Speech Grammar Format Specification*. <http://java.sun.com/products/java-media/speech/forDevelopers/JSGF/index.html/>.
- YANKELOVICH N. (1997 to appear). Using natural dialogs as the basis for speech interface design. In *Automated Spoken Dialogue Systems*. MIT Press.