# Archivus: A multimodal system for multimedia meeting browsing and retrieval

**Marita Ailomaa, Miroslav Melichar, Martin Rajman**
Artificial Intelligence Laboratory
École Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland
marita.ailomaa@epfl.ch

**Agnes Lisowska, Susan Armstrong**
ISSCO/TIM/ETI
University of Geneva
CH-1211 Geneva, Switzerland
agnes.lisowska@issco.unige.ch

## Abstract

This paper presents Archivus, a multimodal language-enabled meeting browsing and retrieval system. The prototype is in an early stage of development, and we are currently exploring the role of natural language for interacting in this relatively unfamiliar and complex domain. We briefly describe the design and implementation status of the system, and then focus on how this system is used to elicit useful data for supporting hypotheses about multimodal interaction in the domain of meeting retrieval and for developing NLP modules for this specific domain.

## 1 Introduction

In the past few years, there has been an increasing interest in research on developing systems for efficient recording of and access to multimedia meeting data[1]. This work often results in videos of meetings, transcripts, electronic copies of documents referenced, as well as annotations of various kinds on this data. In order to exploit this work, a user needs to have an interface that allows them to retrieve and browse the multimedia meeting data easily and efficiently.

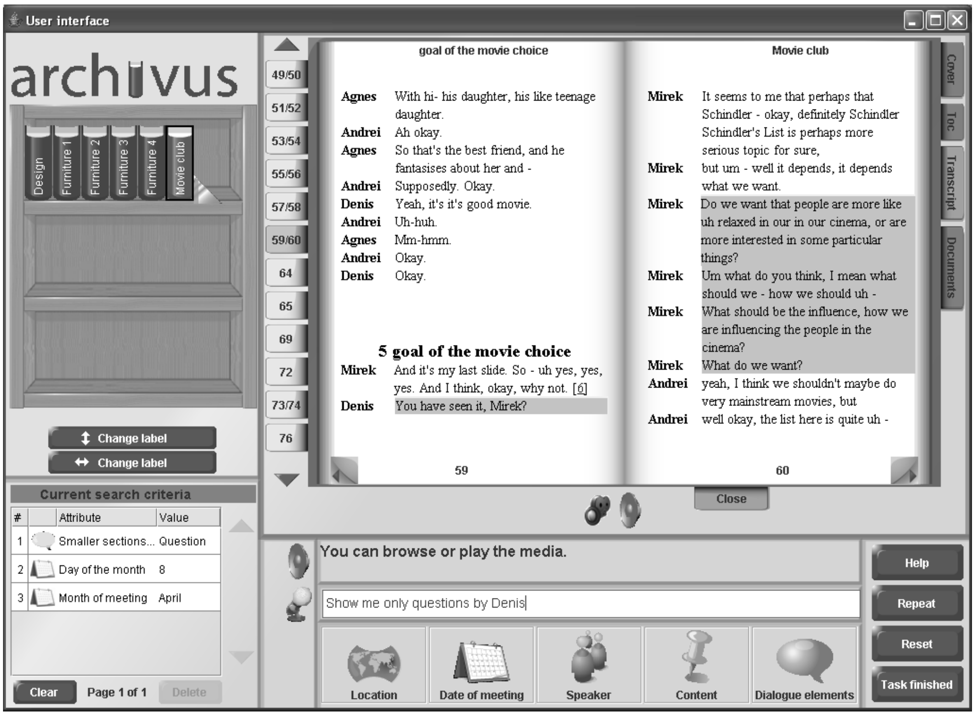
In our work we have developed a multimodal (voice, keyboard, mouse/pen) meeting browser, Archivus, whose purpose is to allow users to access multimedia meeting data in a way that is most natural to them. We believe that since this is a new domain of interaction, users can be encouraged to try out and consistently use novel input modalities such as voice, including more complex natural language, and that in particular in this domain, such multimodal interaction can help the user find information more efficiently.

When developing a language interface for an interactive system in a new domain, the Wizard of Oz (WOz) methodology (Dahlbäck et al., 1993; Salber and Coutaz, 1993) is a very useful tool. The user interacts with what they believe to be a fully automated system, when in fact another person, a 'wizard' is simulating the missing or incomplete NLP modules, typically the speech recognition, natural language understanding and dialogue management modules. The recorded experiments provide valuable information for implementing or fine-tuning these parts of the system.

However, the methodology is usually applied to unimodal (voice-only or keyboard-only) systems, where the elicitation of language data is not a problem since this is effectively the only type of data resulting from the experiment. In our case, we are developing a complex multimodal system. We found that when the Wizard of Oz methodology is extended to multimodal systems, the number of variables that have to be considered and controlled for in the experiment increases substantially. For instance, if it is the case that within a single interface any task that can be performed using natural language can also be performed with other modalities, for example a mouse, the user may prefer to use the other – more familiar – modality for a sizeable portion of the experiment. In order to gather a useful amount of natural language data, greater care has to be taken to design the system in a way that encourages language use. But, if the goal of the experiment is also to study what modalities users find more useful in some situa-

[1] The IM2 project *http://www.im2.ch*, the AMI project *www.amiproject.org*, The Meeting Room Project at Carnegie Mellon University, *http://www.is.cs.cmu.edu/mie*, and rich transcription of natural and impromptu meetings at ICSI, Berkeley, *http://www.icsi.berkeley.edu/Speech/EARS/rt.html*

*Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 49–52,
Sydney, July 2006. ©2006 Association for Computational Linguistics



goal of the movie choice

**Agnes** With hi- his daughter, his like teenage daughter.
**Andrei** Ah okay.
**Agnes** So that's the best friend, and he fantasises about her and -
**Andrei** Supposedly. Okay.
**Denis** Yeah, it's it's good movie.
**Andrei** Uh-huh.
**Agnes** Mm-hmm.
**Andrei** Okay.
**Denis** Okay.

**5 goal of the movie choice**

**Mirek** And it's my last slide. So - uh yes, yes, yes. And I think, okay, why not. [6]
**Denis** You have seen it, Mirek?

59

Movie club

**Mirek** It seems to me that perhaps that Schindler - okay, definitely Schindler Schindler's List is perhaps more serious topic for sure,
**Mirek** but um - well it depends, it depends what we want.
**Mirek** Do we want that people are more like uh relaxed in our in our cinema, or are more interested in some particular things?
**Mirek** Um what do you think, I mean what should we - how we should uh -
**Mirek** What should be the influence, how we are influencing the people in the cinema?
**Mirek** What do we want?
**Andrei** yeah, I think we shouldn't maybe do very mainstream movies, but
**Andrei** well okay, the list here is quite uh -

60

You can browse or play the media.

Show me only questions by Denis

Current search criteria

| # | Attribute | Value |
|---|---|---|
| 1 | Smaller sections... | Question |
| 2 | Day of the month | 8 |
| 3 | Month of meeting | April |

Figure 1: The Archivus Interface

tions compared to others, language use must be encouraged without being forced, and finding this balance can be very hard to achieve in practice.

## 2 Design and implementation

The Archivus system has been designed to satisfy realistic user needs based on a user requirement analysis (Lisowska, 2003), where subjects were asked to formulate queries that would enable them to find out "what happened at a meeting". The design of the user interface is based on the metaphor of a person interacting in an archive or library (Lisowska et al., 2004).

Furthermore, Archivus is flexibly multimodal, meaning that users can interact unimodally choosing one of the available modalities exclusively, or multimodally, using any combination of the modalities. In order to encourage natural language interaction, the system gives textual and vocal feedback to the user. The Archivus Interface is shown in Figure 1. A detailed description of all of the components can be found in Lisowska et al. (2004).

Archivus was implemented within a software framework for designing multimodal applications with mixed-initiative dialogue models (Cenek et al., 2005). Systems designed within this framework handle interaction with the user through a multimodal dialogue manager. The dialogue manager receives user input from all modalities (speech, typing and pointing) and provides multimodal responses in the form of graphical, textual and vocal feedback.

The dialogue manager contains only linguistic knowledge and interaction algorithms. Domain knowledge is stored in an SQL database and is accessed by the dialogue manager based on the constraints expressed by the user during interaction.

The above software framework provides support for remote simulation or supervision of some of the application functionalities. This feature makes any application developed under this methodology well suited for WOz experiments. In the case of Archivus, pilot experiments strongly suggested the use of two wizards – one supervising the user's input (Input Wizard) and the other controlling the natural language output of the system (Output Wizard). Both wizards see the user's input, but their actions are sequential, with the Output Wizard being constrained by the actions of the Input Wizard.

The role of the Input Wizard is to assure that the user's input (in any modality combination) is correctly conveyed to the system in the form of sets of semantic pairs. A semantic pair (SP) is a qualified piece of information that the dialogue system is able to understand. For example, a system could understand semantic pairs such as `date:Monday` or `list:next`. A user's utterance *"What questions did this guy ask in the meeting yesterday?"* combined with pointing on the screen at a person called "Raymond" could translate to `dialogact:Question`, `speaker:Raymond`, `day:Monday`.

In the current version of Archivus, user clicks are translated into semantic pairs automatically by the system. Where written queries are concerned, the wizard sometimes needs to correct automatically generated pairs due to the currently low performance of our natural language understanding module. Finally since the speech recognition engine has not been implemented yet, the user's speech is fully processed by a wizard. The Input Wizard also assures that the fusion of pairs coming from different modalities is done correctly.

The role of the Output Wizard is to monitor, and if necessary change the default prompts that are generated by the system. Changes are made for example to smooth the dialogue flow, i.e. to better explain the dialogue situation to the user or to make the response more conversational. The wizard can select a prompt from a predefined list, or type a new one during interaction.

All wizards' actions are logged and afterwards used to help automate the correct behavior of the system and to increase the overall performance.

## 3 Collecting natural language data

In order to obtain a sufficient amount of language data from the WOz experiments, several means have been used to determine what encourages users to speak to the system. These include giving users different types of documentation before the experiment – lists of possible voice commands, a user manual, and step-by-step tutorials. We found that the best solution was to give users a tutorial in which they worked through an example using voice alone or in combination with other modalities, explaining in each step the consequences of the user's actions on the system. The drawback of this approach is that the user may be biased by the examples and continue to interact according to the interaction patterns that are provided, rather than developing their own patterns. These influences need to be considered both in the data analysis, and in how the tutorials are written and structured.

The actual experiment consists of two parts in which the user gets a mixed set of short-answer and true-false questions to solve using the system. First they are only allowed to use a subset of the available modalities, e.g. voice and pen, and then the full set of modalities. By giving the users different subsets in the first part, we can compare if the enforcement of certain modalities has an impact on how they choose to use language when all modalities are available.

On the backend, the wizards can also to some extent have an active role in encouraging language use. The Input Wizard is rather constrained in terms of what semantic pairs he can produce, because he is committed to selecting from a set of pairs that are extracted from the meeting data. For example if "Monday" is not a meeting date in the database, the input is interpreted as having "no match", which generates the system prompt *"I don't understand"*. Here, the Output Wizard can intervene by replacing that prompt by one that more precisely specifies the nature of the problem.

The Output Wizard can also decide to replace default prompts in situations when they are too general in a given context. For instance, when the user is browsing different sections of a meeting book (cover page, table of contents, transcript and referenced documents) the default prompt gives general advice on how to access the different parts of the book, but can be changed to suggest a specific section instead.

## 4 Analysis of elicited language data

The data collected with Archivus through WOz experiments provide useful information in several ways. One aspect is to see the complexity of the language used by users – for instance whether they use more keywords, multi-word expressions or full-sentence queries. This is important for choosing the appropriate level of language processing, for instance for the syntactic analysis. Another aspect is to see the types of actions performed using language. On one hand, users can manipulate elements in the graphical interface by expressing commands that are semantically equivalent with pointing, e.g. *"next page"*. On the other hand, they can freely formulate queries relating to the information they are looking for, e.g. *"Did they decide to put a sofa in the lounge?"*. Commands are interface specific rather than domain specific. From the graphical interface the user can easily predict what they can say and how the system will

| Part 1 condition | Pointing | Language |
|---|---|---|
| Experiment set 1 | | |
| voice only | 91% | 9% |
| voice+keyboard | 88% | 12% |
| keyboard+pointing | 66% | 34% |
| voice+keyb.+pointing | 79% | 21% |
| Experiment set 2 | | |
| voice only | 68% | 32% |
| voice+pointing | 62% | 38% |
| keyboard+pointing | 39% | 61% |
| pointing | 76% | 24% |

Table 1: Use of each modality in part 2.

respond. Queries depend on the domain and the data, and are more problematic for the user because they cannot immediately see what types of queries they can ask and what the coverage of the data is. But, using queries can be very useful, because it allows the user to express themselves in their own terms. An important goal of the data analysis is to determine if the language interface enables the user to interact more successfully than if they are limited to pointing only. In addition, the way in which the users use language in these two dimensions has important implications for the dialogue strategy and for the implementation of the language processing modules, for instance the speech recognition engine. A speech recognizer can be very accurate when trained on a small, fixed set of commands whereas it may perform poorly when faced with a wide variety of language queries.

Thus far, we have performed 3 sets of pilot WOz experiments with 40 participants. The primary aim was to improve and finetune the system and the WOz environment as a preparation for the data-collection experiments that we plan to do in the future. In these experiments we compared how frequently users used voice and keyboard in relation to pointing as we progressively changed features in the system and the experimental setup to encourage language use. The results between the first and the third set of experiments can be seen in table 1, grouped by the subset of modalities that the users had in the first part of the experiment.

From the table we can see that changes made between the different iterations of the system achieved their goal – by the third experiment set we were managing to elicit larger amounts of natural language data. Moreover, we noticed that the modality conditions that are available to the user in the first part play a role in the amount of use of language modalities in the second part.

## 5 Conclusions and future work

We believe that the work presented here (both the system and the WOz environment and experimental protocol) has now reached a stable stage that allows for the elicitation of sufficient amounts of natural language and interaction data. The next step will be to run a large-scale data collection. The results from this collection should provide enough information to allow us to develop and integrate fairly robust natural language processing into the system. Ideally, some of the components used in the software framework will be made publicly available at the end of the project.

## References

Pavel Cenek, Miroslav Melichar, and Martin Rajman. 2005. A Framework for Rapid Multimodal Application Design. In Václav Matoušek, Pavel Mautner, and Tomáš Pavelka, editors, *Proceedings of the 8th International Conference on Text, Speech and Dialogue (TSD 2005)*, volume 3658 of *Lecture Notes in Computer Science*, pages 393–403, Karlovy Vary, Czech Republic, September 12-15. Springer.

Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz Studies – Why and How. In Dianne Murray Wayne D. Gray, William Hefley, editor, *International Workshop on Intelligent User Interfaces 1993*, pages 193–200. ACM Press.

Agnes Lisowska, Martin Rajman, and Trung H. Bui. 2004. ARCHIVUS: A System for Accessing the Content of Recorded Multimodal Meetings. In *In Procedings of the JOINT AMI/PASCAL/IM2/M4 Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Bourlard H. & Bengio S., eds. (2004), LNCS, Springer-Verlag, Berlin.*, Martigny, Switzerland, June.

Agnes Lisowska. 2003. Multimodal interface design for the multimodal meeting domain: Preliminary indications from a query analysis study. Project report IM2.MDM-11, University of Geneva, Geneva, Switzerland, November.

Daniel Salber and Joëlle Coutaz. 1993. Applying the wizard of oz technique to the study of multimodal systems. In *EWHCI '93: Selected papers from the Third International Conference on Human-Computer Interaction*, pages 219–230, London, UK. Springer-Verlag.