A Natural Language Web-based Dialogue System with a Talking Face

Anton Nijholt, Mathieu van den Berk, and Arjan van Hessen

University of Twente, Centre of Telematics and Information Technology (CTIT)
PO BOX 217,
7500 AE Enschede, the Netherlands
{anijholt, berk, hessen}@cs.utwente.nl

Abstract. In this paper we discuss our research on interactions in a virtual theatre that has been built using VRML and therefore can be accessed through Web pages. In the virtual environment we employ two agents. Presently, our WWW-based virtual theatre allows navigation input through keyboard and mouse. In development is a navigation agent that allows speech input. We also have an information agent which allows a natural language dialogue with the system where the input is keyboard-driven and the output is both screen and (speech) synthesizer based. The system's spoken dialogue contribution is presented by visual speech; that is, a simple 'talking face' on the screen mouths the systems questions and responses.

1 Introduction

In this paper we present our research on developing an environment in which users can display different behaviors and have goals that emerge during the interaction with this environment. Users who, for example, decide they want to spend an evening outside their home and, while having certain preferences, cannot say in advance where exactly they want to go, whether they first want to have a diner, whether they want to go to a movie, theatre, or to opera, when they want to go, etc. During the interaction, both goals, possibilities and the way they influence each other become clear. One way to support such users is to give them different interaction modalities and access to multimedia information.

2 The Virtual Environment

A virtual theatre has been built according to the design drawings made by the architects of a local theatre. Part of the building has been realized by converting AutoCAD drawings to VRML97. Video recordings and photographs have been used to add 'textures' to walls, floors, etc. Sensor nodes in the virtual environment activate animations (opening doors) or start events (entering a dialogue mode, playing music, moving spotlights, etc.). Visitors can explore the environment of the building, enter the theatre and walk around, visit the hall, admire

the paintings on the walls, go to the balconies and, take a seat in order to get a view of the stage from that particular location. Information about today's performances is available on a blackboard that is automatically updated using information from the database with performances. In addition, as may be expected, visitors may go to the information desk in the theatre, see previews and start a dialogue with an information and transaction agent called 'Karin'. The first version of Karin looked like other standard avatars available on World Wide Web. The second version, available in a prototype of the system, makes use of a 3D talking face.

3 The Navigation Agent

The WWW-based virtual theatre we are developing allows navigation input through keyboard and mouse. Such input allows the user to move and to rotate, to jump from one location to an other, to interact with objects and to trigger them. In addition, a navigation agent has been developed that is prepared to allow the user to explore the environment and to interact with objects in this environment by means of speech commands. A smooth integration of the pointing devices and speech in a virtual environment requires means to resolve deictic references that occur in the interaction. The navigation agent should be able to reason about the geometry of the virtual world in which it moves. The current version of the navigational agent is not really conversational. Straightforward typed commands (to mimic future speech commands) make it possible for the user to explore the virtual environment. Navigation also requires that names have to be associated with the different parts of the building, the objects and the agents, which can be found inside of it. Clearly, users may use different words to designate them, including implicit references that have to be resolved in a reasoning process.

Speech Recognition on local machines turns out to be pretty good but speech recognition on the World Wide Web results in various problems. Every user should have a speech recognition engine that can recognize their commands and send this information to the server system. Good speech recognition systems are very expensive and bad systems result in bad recognized commands. Another solution would be to have the speech recognition on the server side but the way to record and the robust transporting of the audio for web-applications is in a too early stage of development.

4 The Information/Transaction Agent

Karin, the information/transaction agent, allows a natural language dialogue with the system about performances, artists, dates, prices, etc. Karin wants to give information and to sell tickets. Karin is fed from a database that contains all the information about performances in our local theatre. Developing skills for Karin, in this particular environment, is one of the aims of our research project.

This research fits in a context of much more general 'intelligent' (web-based) information and transaction services.

Our current version of the dialogue system of which Karin is the face is called THIS v1.0 (Theatre Information System). The approach used can be summarized as rewrite and understand. User utterances are simplified using a great number of rewrite rules. The resulting simple sentences are parsed. The output can be interpreted as a request of a certain type. System response actions are coded as procedures that need certain arguments. Missing arguments are subsequently asked for. The system is modular, where each 'module' corresponds to a topic in the task domain. There are also modules for each step in the understanding process: the rewriter, the recognizer and the dialogue manager. The rewrite step can be broken down into a number of consecutive steps that each deal with particular types of information, such as names, dates and titles. The dialogue manager initiates the first system utterance and goes on to call the rewriter and recognizer process on the user's response. Also, it provides an interface with the database management system (DBMS). Queries to the database are represented using a standard query language like SQL. Results of queries are represented as bindings to variables, which are stored in the global data-structure, called context. The arguments for the action are dug out by the dedicated parser, associated with the category. All arguments that are not to be found in the utterance are asked for explicitly. More information about this approach can be found in [1].

Presently the input to Karin is keyboard-driven natural language and the output is both screen and speech based. In development is an utterance generation module. Based on the most recent user utterance, on the context and on the database, the system has to decide on a response action, consisting of database manipulation and dialogue acts.

5 Speech Generation through Templates

The utterance generation by the information agent uses a list of utterance templates. Templates contain gaps to be filled with information items: attribute-value pairs labeled with syntactic and lexical features. Templates are selected on the basis of five parameters: utterance type, the body of the template and possible empty lists of information items that are to be marked as given, wanted and new. The utterance type and body determine the word-order and the main intonation contour. The presence and number of information items in the given, wanted and new slots, as well as special features affect the actual wording and intonation of the utterance.

For pronouncing the utterance templates we use the Fluent Dutch Text-to-Speech system [2]. Fluent Dutch runs on top of the MBROLA diphone synthesizer [3]. It uses a Dutch voice, developed at the Utrecht institute of linguistics (OTS). Fluent Dutch operates at three levels: the grapheme level, the phoneme level and a low-level representation of phones where the length and pitch of sounds is represented. For many words, the phonetic description is taken from

lexical resources of Van Dale dictionaries. Other prosodic information is derived by heuristic rules. It is possible to manipulate prosody by adding punctuation at the grapheme level, by adding prosodic annotations at the phoneme level or by directly manipulating the phone level. More details of the utterance generation module can be found in [4].

6 Facing the Information Agent

The visual part of the information agent is presented as a talking face. It has become clear from several studies that people engage in social behavior toward machines. It is also well known that users respond differently to different 'computer personalities'. It is possible to influence the user's willingness to continue working even if the system's performance is not perfect. They can be made to enjoy the interaction, they can be made to perform better, etc., all depending on the way the interface and the interaction strategy has been designed. It also makes a difference to interact with a talking face display or with a text display. Finally, the facial appearance and the expression of the face matters. From all these observations (see [5] for details) we conclude that introducing a talking face can help to make interactions more natural and shortcomings of the technology more acceptable to users.

We developed a virtual face in a 3D-design environment. The face consists of various three-dimensional coordinates and is connected through faces. These faces are shaded to visualize a three-dimensional virtual face. The 3D data is converted to VRML-data that can be used for real-time viewing of the virtual face. A picture of a real human face can be mapped onto the virtual face. We are researching various kinds of faces to determine which can be best used for this application. Some are rather realistic and some are more in a cartoon-style. This face is the interface between the users of the virtual theatre and the theatre information system. A dialogue window is shown when users approach the information-desk while they are navigating in the virtual theatre. The face is capable of visualizing the speech synchronously to the speech output. This involves lip-movements according to a couple of visemes. The face also visualizes facial expressions according to user's input or the system's output. Figure 1 represents the architecture of the visual speech system.

The last element in the chain (the VRML-browser) is also the first element. We use Cosmo Player, which is a plug-in for an HTML-Browser, for viewing VRML-files. These files are specifications of a three dimensional virtual environment. The whole virtual theatre is a collection of VRML files, which can be viewed by the browser. As mentioned earlier, the user will see a virtual face when the information desk is approached. A dialogue window also pops up at this time. This is called the JAVA Schisma applet. In this window, the user can formulate questions or give answers to the system's questions. The user types the questions on a keyboard in Dutch sentences. The answers to the questions are to be determined on the server side: the Schisma server. Answers or respond-

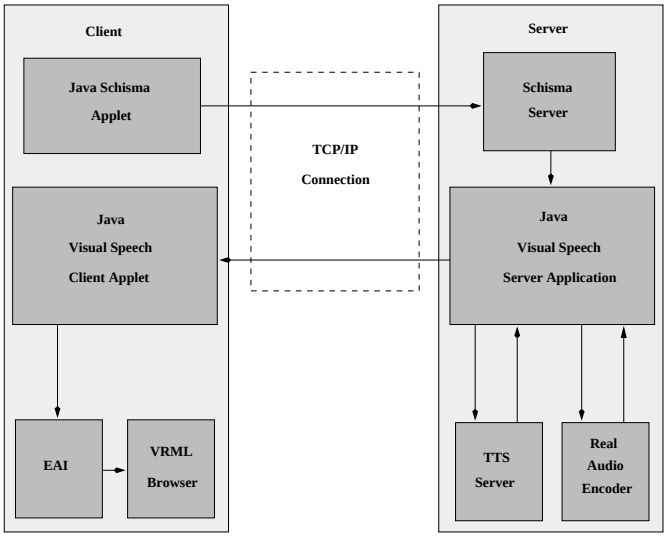

Fig. 1. Architecture of the visual speech system. It's a client-server architecture for use as a www-site with Java client applets connected to a Java server application

ing questions are passed to the JAVA Visual Speech Server Application on the server side.

This application filters the textual output of the dialogue system in parts that are to be shown in a table or a dialogue window and parts that have to be converted to speech. The parts that are to be shown in the dialogue window or a table, like lengthy descriptions of particular shows or lists of plays are send to the Schisma Client Applet where they are showed on the screen. The parts of the Schisma output that are to be spoken by the virtual face are converted to speech with the Text-to-Speech Server. The input is the raw text and the output is the audio file of this spoken text and information about the phonemes in the text and their duration.

For example, the Dutch word for "speech generation" is "spraakgeneratie". This word contains the following phonemes: S p r *a k x e n @ r a t s I. When the resulting audio file is played, each phoneme has it's own duration. This information is gathered from the TTS-server:

```
s 79 p 71 r 38 a 106 50 127 k 53 x 90 e 113 20 102 n 60 @ 38 r 53
a 101 t 23  s 113 I 119 20 75
```

The characters are the phonemes and the first number after the characters are durations of the corresponding phonemes in milliseconds. If more numbers follow then the first number is a percentage of the whole duration in which the pitch of the voice changes to the following number. So the first 'a' is spoken for 106 milliseconds and on 50% of this 106 milliseconds the pitch changes to 127 Hz. The audio file, which the TTS-server produced, will be compressed to a Real-

Audio file for a fast transfer-rate. The previously described information from TTS-server will be sent to the JAVA Visual Speech Client Applet together with the converted audio file. The Visual Speech Client Applet uses the phoneme information to map the phonemes onto different mouth states or visemes. All the phonemes are categorized in five visemes.

When the audio file is loaded on the client side, the mouth states and their durations are passed to the External Authoring Interface (EAI). This is an interface between JAVA and the VRML browser. This interface triggers animations in the virtual environment. It starts the sound playback and all the corresponding animations. Only the mouth states are specified in the VRML-file. The animation is done by interpolating between mouth states in the given amount of time. This results in smooth lip-movements.

7 Future Research: speech recognition

The use of speech technology in information systems will continue to increase. Most currently installed information systems that work with speech, are telephone-based systems where callers can get information by speaking aloud some short commands. Also real dialogue systems wherein people can say normal phrases become more and more common, but one of the problems in this kind of systems is the limitation of the context. As long as the context is narrow they perform well, but wide contexts are causing problems. Moreover, there are technical problems concerning the use of speech over the Internet. As pointed out above, Real Audio assures that speech is transmitted, although the quality can be low. Vice-versa is causing problems: most speech recognition systems work only when the speech satisfies certain standards (8 or 16 bit, 8 kHz sample frequency etc.); with Real Audio this can not be guaranteed. A possible solution is to do recognition on the client side, but this is only possible if there are low-cost, good working recognizers available for a wide range of platforms.

References

1. Lie, D., J. Hulstijn, A. Nijholt, R. op den Akker. A Transformational Approach to NL Understanding in Dialogue Systems. Proceedings NLP and Industrial Applications, Moncton, New Brunswick, August 1998, to appear.
2. Dirksen, A. and Menert, L. (1997). Fluent Dutch text-to-speech. Technical manual, Fluency Speech Technology/OTS Utrecht.
3. Dutoit, T. (1997). High-quality text-to-speech synthesis: An overview. Electrical and electronics engineering, 17 (1), 25–36.
4. Hulstijn, J, and A. van Hessen. Utterance Generation for Transaction Dialogues. Proceedings 5th International Conf. Spoken Language Processing, Sydney, Australia, 1998, to appear.
5. Friedman, B. (ed.). Human Values and the Design of Computer Technology. CSLI Publications, Cambridge University Press, 1997.