AN ASYNCHRONOUS VIRTUAL MEETING SYSTEM FOR BI-DIRECTIONAL SPEECH DIALOG

Takuya Nishimoto, Hidehiro Yuki, Takehiko Kawahara and Yasuhisa Niimi

Kyoto Institute of Technology
Matsugazaki, Sakyo-ku, Kyoto, 6068585 Japan
nishi@dj.kit.ac.jp, http://www-vox.dj.kit.ac.jp/

Abstract

Voice-mail and video-mail systems for the Internet are becoming very popular, because the speech and video compression technology is greatly improved in these days. Asynchronous voice-mail system, however, does not seem attractive because it's not designed for discussions. We propose a network-based voice conference system which enables asynchronous and bi-directional discussions.

The system displays the voice messages as the threaded written words. The user can manipulate voice messages just as if they are text messages. If the participant wants to quote or annotate to a message, the user has only to play the sound and barge into the message while it is playing. Such user interface increases the usefulness of the voice-mail system.

We performed a preliminary experiment to evaluate the proposed conversation method. The voice messages are transcribed manually, because our system is not integrated with the speech recognition at present. Using the system, asynchronous voice conversations were realized efficiently.

AVM is an application platform for the many kinds of speech processing, such as spontaneous speech recognition and automatic dialog tagging. The applications of the system include the entertainment systems, such as interactive radio dramas, or asynchronous multi-user game using speech dialog.

INTRODUCTION

E-mail system, which is widely used in these days, gives us the capability of reading or writing a message, making excerpts of it, and annotating to it. In this manner, various discussions or conversations are enabled using the written words.

In comparison with the text messages, voice messages are easy to input for people who don't have much skill of using a keyboard. Voice messages also contain the para-linguistic information, which enable the better discussions between the distant places. Such voice message systems are already used commonly; an automatic answering telephone is the most popular one. Voice-mail systems are also used at some offices. It is, however, very difficult to quote or annotate previous voice messages, so they are not suitable for discussions.

We designed and implemented a system called AVM (Asynchronous Virtual Meeting). It produces the effect equal to quoting and annotating other messages among speakers who participate the discussion asynchronously. The system play-backs the voice message and records the response simultaneously. It also uses the temporal information on when the voice was spoken.

INVESTIGATING THE MESSAGE SYSTEMS

In this work, we intend to make a voice messaging system, which has the merits of E-mail that uses the written words. It must preserve the both advantages of E-mail and spontaneous speech.

The followings are the merits of E-mail system that we regard as important.

- E-mail is an *asynchronous communication system.* It means the sender and the receiver of a message do not use it at the same time. The users of E-mail can make a message or reply at any time.

  Messages are accumulated in the storages, so they can be displayed at any time. It is also easy to make the copy or excerpt of the message.

- The message is *displayed as the written words.* Because we can browse the written words easily, the messages are easy to deal with.

The communication with written words, however, has the following weak points compared to the voice conversation such as telephone.

- The written words cannot convey para-linguistic features, such as emotional information.

- The mood of the person who read the message is not known until he writes the reply. At the synchronous communication such as speech dialog, however, the speaker can see the mood of the other while he is speaking.

There are some applications which are designed for sending voice mails. They have the ability to

compress the speech files to transfer with the limited bandwidth. Most of them, however, are designed based on the text-based mailer, so the written words are more handy than the voice data. Such systems do not appeal the merit of using voice message.

ASYNCHRONOUS VIRTUAL MEETING

Generally the voice mail systems separate the recording procedure and the play-back procedure. In the human-to-human conversation, however, we can barge in on the other. This manner of conversation can be simulated using the audio input and output simultaneously. Some speech dialog systems have the capability of the nodding with speech output and graphical display, which aim to show the internal status of the speech recognizer and the dialog manager[1].

The full duplex voice I/O also makes the human-to-human voice messaging system more attractive. From this point of view, we designed a system called AVM. It can play-back and record voice message simultaneously, while the other systems separate these processes (Figure 1).

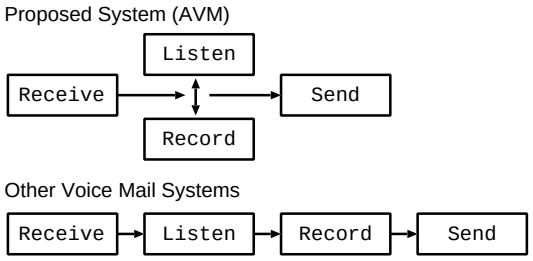

Figure 1: The method for recording messages in proposed system compared to the other voice mail systems.

This system utilizes both the voice itself and the temporal information which the voice was spoken. An example of playing and recording messages in AVM is shown in Figure 2. At first, the utterance of speaker A is recorded as 1A1. In the second session requested by B, 1A is play-backed as 2A. Speaker B speaks three parts of utterances in reply to A. Each part of speech is separated by the certain length of silences and labeled as 'interrupting' or 'overlapping.' 1B1 is an interrupting part, and 1B2 and 1B3 are overlapping parts in this case. In the second session, therefore, the utterance 2A was paused by 2B1 and divided into 2A1' and 2A1". The message 2B, which consists of 2B1, 2B2 and 2B3, is stored to the server after the second session.

In our goal, whether a part is interrupting part or not would be recognized in real time. At this moment, however, all parts of speech are regarded as 'interrupting' or 'overlapping' according to the application's option menu.

Each part of speech has the relative time to it's *parent* part. In Figure 2, the parent part of 2B1 is 1A1, and the relative time to the parent is shown as $t(2B1)$. In case of 2B2 and 2B3, the parent is also 1A1, and the relative time is shown as $t(2B2)$ and $t(2B3)$ respectively.

In AVM, the time scale in each meeting space changes dynamically. When the user selects a set of messages, the message management server decides the layout of the messages, then it generates the sound and its markup data on demand. The relative time of a part and its parent is used in this procedure. Other attributes are also used, such as the 'interrupting' or 'overlapping' label, and the words contained in the message itself (transcribed manually or by using speech recognition). In Figure 3, there are four segments which correspond to the former part of 1A1, 2B2, the latter part of 1A1, and 2B3, respectively. The first and the third segments are not equal to 2A1' and 2A1", because the message server found the more preferable point to divide 2A, which may be the nearest word-boundary of $t(2B1)$.

The dynamic conversation composing is one of the most important features of the system. Several functions such as speech recognition, automatic dialog tagging, language processing including modality conversion, and speech synthesis must be used to generate natural conversations.

VISUALIZING VOICE MESSAGES

Another important feature of the system is the visualization of the voice messages. We designed an user interface, which displays the voice messages as if they are the written words. In the on-line bulletin board system (BBS), usually the written messages are threaded and viewed as trees. When we write messages on BBS or in the E-mail, some conventions are used to show the quoted messages, such as '$\gg$' prefix.

We designed a user-agent software of our voice messaging system with such a look-and-feel. Figure 4 is the screen image of the software. The window consists of two panes. The left pane is tree-view, which the user can select the messages. When the user selects a node in the tree, voice messages are merged into a file, which includes the selected message itself and the parent or ancestors in the tree. The right pane is text-view, on which the user can read the messages as the written words. Each paragraph corresponds to the segment generated by the server (Figure 3). While the sound is playing, each line in the text-view is highlighted with synchronizing to the voice.

Figure 5 shows the system organization and the usage of the proposed system. The whole system consists of the user-agent and the message management server. Each server manages some meeting rooms and all the original utterances are stored in the server.

In AVM, the spoken messages may be transcribed

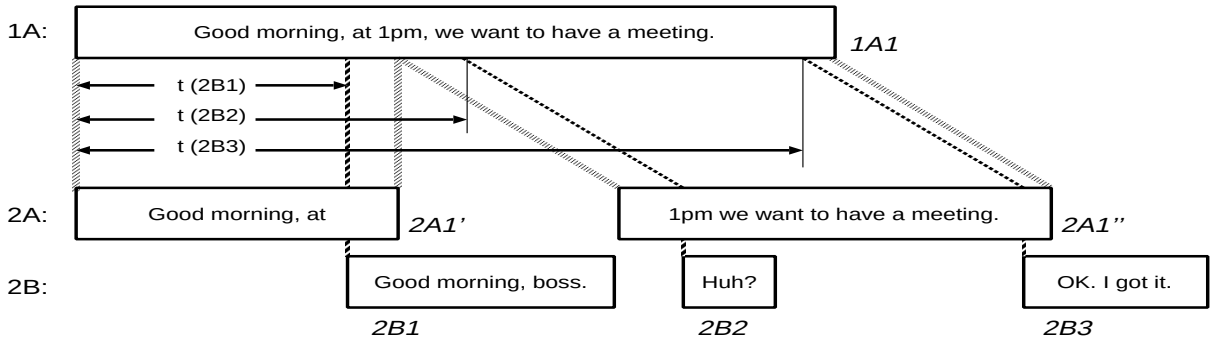

Figure 2: An example of playing and recording messages in AVM.

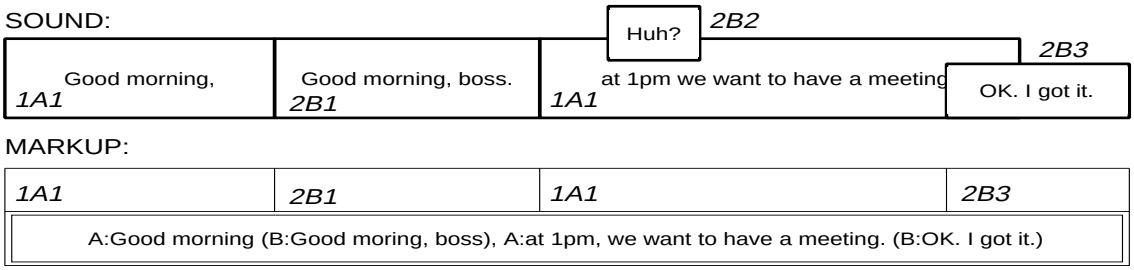

Figure 3: An example of merged message by the AVM server.

Figure 4: The user-agent software (VOYAGER). Left pane is the tree-view and right pane is the text-view.

in two ways. One of the possibilities is that the user-agent has the capability of speech recognition. When the utterance is recorded, the voice is transcribed and transmitted to the server. Another possibility is that the server has the speech recognizer. When the server received a speech data from the user-agent, it is transcribed and stored to the database.

The important point is that the both ways could be supported in our system. If the user-agent software has very limited speech recognition ability, the server can compensate it. If the recognition performance of user-agent is improved, it can contribute to the better usability of the user-agent software, and the server can use the information received from the user-agent. So we can put it to practical use now, and improve the speech processing for the future while the users make conversations using the present system.

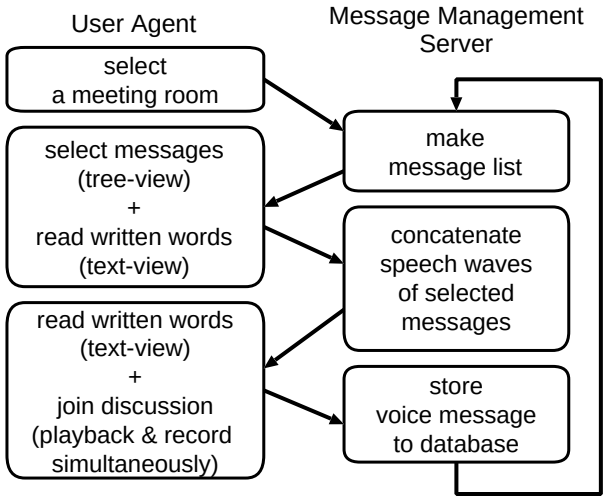

Figure 5: The usage of proposed system.

IMPLEMENTATION

The AVM file transfer protocol is based on the HTTP[2] which is used for World Wide Web. The transcriptions of the voice messages and and miscellaneous information, such as its parent message ID and the relative time, are represented in the AVML markup language, which is based on XML[3].

The AVM server VOXER is implemented with Perl and C, which runs on UNIX or Win32 systems. The user-agent software VOYAGER (Figure 4) is written in C++ and runs on the Microsoft Windows system with a full-duplex audio I/O.

EXPERIMENT

To evaluate the usability for the conversations, preliminary experiments were carried out using the system. Our system is not integrated with the speech recognition at present, so the voice messages were transcribed manually. In the experiments, one person used the system at a time. After each session we transcribed the messages, then the next person used the system. All the subjects are the university students who speak Japanese in our research group.

Overlapping Merge Mode

In the first experiment, all the utterances are regarded as 'overlapping.' The subjects are eight men. First we prepared a lecture-style speech, and subjects were asked to listen to it and say any questions or comments. Because the subjects are not familiar with the topic, the subjects rarely barged in on it.

We also asked them to do casual conversations. Now they were familiar to the topic, so they made responses easily. In case the overlap is small, the speech merged by the server seemed very natural. When several long utterances are overlapped, however, they found difficulty to understand what the speakers are saying.

Interrupting Merge Mode

In the second experiment, all the utterances are regarded as 'interrupting.' During the 4 days of experiment, 14 subjects talked about the events that all the subjects commonly experienced. We encouraged them to say as many messages as they could. As the result, about 100 parts of speech are recorded. The subjects said that the operations is very intuitive and easy to learn.

We observed the subject's behavior in using the system. Most of the subjects read the written messages first. When they decided to make a reply to a message, they selected the message and listened to it. The users could make messages without listening to the all voices. This fact shows the efficiency of this method.

It also turned out that the messages that are already heard should be easily took off to make the system more usable.

CONCLUSION

In this paper, we proposed a method for asynchronous virtual meeting called AVM. This system provides the easy manner to quote the voice messages and increases the usefulness of the voice-mail system. As the result of our preliminary evaluations, conversations were realized efficiently with the system.

AVM is an application platform for the many kinds of speech processing. We are integrating the spontaneous speech recognition and the dialog tagging functions into our AVM server.

The integration of the AVM and the E-mail or the BBS enables the seamless communication of written words and spoken words. Text-to-speech technology is expected to be used in such a system.

Many streaming media systems on the Internet are developed and used in these days. The HyperAudio[4] system has a radio-like user interface, and the user can listen to the audio contents interactively using the 'jump' button. The telephone also can be used as the client system. The HyperAudio with the feature of submitting voices from the listeners will be a system resembles to the AVM. In comparison to such systems, AVM is a metaphor of reading and writing text messages. The data structure of AVM is designed for the bi-directional conversations. It is also extensive for making excerpts of many kind of continuous media, such as video-messages. Some systems, including HyperAudio and RealPlayer (by RealNetworks), use SMIL[5] for describing the structure of the streaming contents. The role of the data which the AVM server generates is similar to SMIL, so an extension to the server enables the SMIL browsers to listen to the conversations on AVM.

Because the AVM server makes the audio contents on demand, many applications can be implemented into the server. Our future works include the realization of an multi-user voice game system using AVM.

References

[1] Jun-ichi Hirasawa, Noboru Miyazaki, Mikio Nakano, Takeshi Kawabata, Implementation of Coordinative Nodding Behavior on Spoken Dialog Systems, Proceeding of ICSLP'98 pp.2347-2350, 1998.

[2] Hypertext Transfer Protocol HTTP/1.1, RFC2068.

[3] Extensible Markup Language (XML), http: //www. w3. org/ XML/

[4] Makoto J.Hirayama, Taro Sugahara, Zhiyong Peng, Junichi Yamazaki, Interactive Listening to Structured Speech Content on the Internet, Proceeding of ICSLP'98 pp.1627-1630, 1998.

[5] Synchronized Multimedia, http: //www. w3. org/ AudioVideo/