論　文

非同期型音声会議システムAVMの設計と評価

西本　卓也[†]　　幸　　英浩[†]　　川原　毅彦[†]　　荒木　雅弘[†]
新美　康永[†]

Design and Evaluation of the Asynchronous Voice Meeting System AVM

Takuya NISHIMOTO[†], Hidehiro YUKI[†], Takehiko KAWAHARA[†], Masahiro ARAKI[†], and Yasuhisa NIIMI[†]

あらまし　非同期・蓄積型の会議を音声によって実現できれば，使いやすくモバイル環境に適しているため，幅広いユーザによる利用が期待できる．しかしそのためには，漸次性を持つ話し言葉を自然に入力できるようにすると同時に，音声情報の相互参照や引用などを容易にする必要がある．本研究では，オーバーラップ発話を利用して非同期的な音声会議を効率的に実現するためのインタフェースを新たに考案し，クライアント・サーバ型の会議システムAVMの作成・評価を行った．特にクライアント側では文字表示によって発言の視覚化を行った．評価実験においては，文字による電子掲示板と提案システムで同じ課題を与えて議論をさせた．その結果，提案システムを用いることで従来の電子掲示板と比較してシステムののべ利用回数が削減され，総発言文字数が48%になり，主観評価の結果でも高い評価を得るなど，提案システムの有効性を確認することができた．

キーワード　音声メッセージ，非同期・蓄積型メディア，双方向通信，オーバーラップ発話，相槌

1. まえがき

時間を同じくしなくても，複数の人間が相互に発言し，コミュニケーションを行う手段として，電子メールや電子掲示板などの非同期・蓄積型メディアがある．この種のメディアには，時間的拘束がない，メッセージが常に保存される，時間をかけて返答を作成できる，複数の相手への同報が可能である，などの利点がある．これらの利点は多様な生活習慣を持つ多数のメンバーがコミュニティを形成することを支援する．また，蓄積された情報を検索することによって新たな仲間を探すことも容易になる．インターネットの普及は，このようなコミュニティ形成機能によってもたらされたとさえ言える．

一方で，コンピュータの高性能化やマルチメディア符号化技術の進歩によって，ディジタル化された音声や画像による通信が可能となっている．特にインターネットにおける音声メディアの応用を表1のように，リアルタイム型と非同期・蓄積型，片方向通信と双方向通信，という観点で分類すると，コミュニティ形成機能を実現するためには，非同期・蓄積型の双方向通信が必要とされる．しかしこれまでに，リアルタイム型の片方向通信（インターネットラジオなど），リアルタイム型の双方向通信（音声チャットやインターネット電話など），非同期・蓄積型の片方向通信（音楽配信など），といった応用は提案されているが，非同期・蓄積型の双方向通信に関する応用はほとんど進んでいない．

表1　インターネットにおける音声メディアの応用
Table 1　Applications of audio media on the internet.

	Realtime	Asynchronous
1way	Internet Radio	Radio Archive Music Archive
2way	Voice Chat Internet Phone	(not available)

非同期・蓄積型の音声メディアを対話的に用いるための提案としてはHyperAudio [1] などがある．また，音声や動画などの配置や同期，リンク情報を表すインターネット標準規格としてSMIL (注1) が広く利用されている．しかしこれらはコンテンツの選択を対話的にしているが，発言そのものの双方向化を実現するもの

† 京都工芸繊維大学，京都府
Faculty of Engineering and Design, Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto, 606–8585 Japan

(注1)：Synchronized Multimedia, http://www.w3.org/AudioVideo/

ではない．

コミュニティ形成機能を音声によって実現できれば，使いやすくモバイル環境に適したサービスが実現できる．幅広いユーザによる利用が期待できるため，高齢者や視覚障害者の社会参加の支援や，社会のディジタル格差（インターネットの接続手段や利用技術の有無による格差）の解消などに貢献できる．また，肉声を使うことにより本人の発言であることを確認しやすくなるため，発言内容の信頼性も高まる．このため，文字メッセージと比較してセキュリティ面でも有利となる．

そこで本研究では，非同期的な音声会議を効率的に実現するためのインタフェースを新たに提案し，クライアント・サーバ型の会議システム AVM の作成・評価を行う．特に音声メッセージの蓄積方法，録音および再生方法，文字による発言の視覚化などに関して詳細に検討し，評価実験によってその有効性を検討する．

本論文の構成は次の通りである．第 2 章では，音声会話とその漸次性について検討を行う．第 3 章では，双方向の会議システムに必要な機能を音声で実現する方法について述べる．第 4 章では，提案システムの設計について述べる．第 5 章では，作成した評価システムについて述べる．第 6 章では，提案システムの有効性を検討するために行った実験について述べる．第 7 章では，まとめと今後の課題について述べる．

2. 音声会話の漸次性と相槌

メールやウェブ情報の音声リーダーソフトや，音声認識機能を統合した電子メールソフトはすでに製品化されている．これらによって，コンピュータに不慣れなユーザや視覚障害者などが容易に通信を行える環境が整備されつつある．しかし，文字による通信手段に音声インタフェースを付加したシステムでは，入力された音声が含んでいた声質や感情などの非言語情報が欠落して伝わってしまう．これは，音声による豊かなコミュニケーションの可能性を切り捨てていると言える．そこで本研究では，例えば音声認識機能は発言の閲覧や検索のための補助的手段として用い，発言そのものは肉声で録音・再生することを前提とする．

また，書き言葉と話し言葉の違いにも考慮する必要がある．日常の会話や電話などで用いられる話し言葉には漸次性があり，相手の知らない情報だけを伝えたり，発話の断片を思いつくまま次々に伝えたりすることが多い．この漸次性が，話し手と聞き手で共有されている知識や情報を省略したり，統語構造が簡単で認知的な負荷の小さい表現を可能にしている [2]．メッセージシステムにおいて音声が有効に利用されるためには，このような漸次的発話を許容する設計が重要となる．

非同期・蓄積型でありながら擬似的にリアルタイムの会話を実現するためには，漸次的な会話を促すための配慮が必要である．そこで本研究では，発話にオーバーラップする他の話者の発話や相槌などの現象に注目する．

過去の対話研究では，一人の話者が話し終わる前に次の話者が話し出す，という現象は稀有であるとして例外視されてきた．しかし，我々はこれまでに，RWC プロジェクトで収録された対話について検討を行い，応答発話の過半数でオーバーラップ現象が見られ，これにより 2 話者の発話時間を合計した時間の 13%が節約されていたことを確認した [3]．また榎本ら [4] も，地図課題対話コーパス中にオーバーラップ発話が全発話中の 45%にも昇っていると報告している．川口ら [5] は，Grice の会話の含みの理論に基づいて，「相手が何を言おうとしたかを推論し理解した」ことを示すものとしてオーバーラップ現象の一部を説明している．

オーバーラップ発話の中で，特に相槌の生成に注目した研究もある．最も簡単な相槌の生成方法は，声の出されていない無音時間が一定の長さ以上続いた場合に相槌を発する，というものである [6] [7]． また，ピッチパターンに基づいた相槌生成モデル [8] も提案されている．音声対話における相槌の役割という観点からは，話者交替におよぼす影響の検討 [9] [10] や確認の役割に関する検討 [11] などがなされている．また，実時間性の高い対話制御によって相槌を生成する音声対話システム [12] が構築されている．

これらの先行研究を踏まえ，本研究では，オーバーラップ発話を例外ではなく一般的な現象としてモデル化する．また，相槌が何らかの役割を持つ，という観点からのインタフェース設計を行う．

3. 音声メッセージの相互編集機能

文字メッセージとの比較において，音声メッセージは，高い現実感や豊富なパラ言語情報を持ち，また，キーボード操作などを用いず容易に入力することができる，という利点がある．その一方で音声は，多くの情報の中から欲しい項目を流し読みして探すことや，必要な部分を引用したり加工したりする編集行為が困

難である．ここでは，音声のこのような問題点を補う方法として二つの提案を行う．

第一の提案は，音声メッセージの検索や流し読みを実現するために音声認識を用いる，というものである．システムの見かけの機能は，音声で録音されたメッセージを音声で再生する，というものに限定する一方で，音声メッセージがあたかも文字情報であるかのように扱えるようにする．これにより，音声メッセージの持つさまざまな利点と，文字メッセージの扱いやすさを両立させることを目指す．

さらに，我々はこの提案を通じて，たとえ音声認識の性能が完璧でなくても，何らかの実用的な応用が可能である，ということを実証したいと考えている．本提案では音声認識の結果を最終目的としてユーザに与えるのではなく，聞きたい音声を選ぶための手段として利用できればよい．元の音声を聞けば内容は理解できるため，誤認識があっても実用性が大きく損なわれることはない．既存の技術によって「いますぐ使える」と感じさせる応用システムを提供することによって，音声応用システムのユーザに対する啓蒙や市場開拓が進むだろう．

第二の提案は，非同期・蓄積型メディアによる双方向的な議論は発言の相互編集行為なしには不可能である，という立場から，テキスト編集とは異なる発想によって同等の機能を提供する，というものである．例えば，電子メールや電子掲示板では，メッセージを読み，その一部を引用し，それにコメントをつける，といった操作を容易に行うことができる．ここで行われている操作の機能は，(a) どのメッセージに対する返答であるかを示すこと，(b) そのメッセージ内でどの部分に注目してるかを示すこと，に大別されると考えられる．ツリー表示によるメッセージの操作は（a）を実現するためのインタフェースであり，発言に「⟩⟩」などの文字を付与して部分引用するのは（b）を実現するためのインタフェースである．これらの機能はいわば発言の相互編集機能である．この機能こそが，非同期型メディアによる双方向的な議論や雑談を可能にしている．

文字による発言を編集するもっとも簡単な手段は，カット＆ペースト機能などを含む，コンピュータのテキスト編集機能である．しかし，音声会話において同等の機能は，簡便なインタフェースでは実現できない．そこで本研究では，前章での検討を踏まえ，オーバーラップ発話とそのタイミング情報に積極的な意味を持たせる，新たな操作体系を提案する．つまり，音声メッセージの再生中に，自由なタイミングでの割り込み（barge-in）を許すこととし，その割り込みが，(a) どの発言に対して行われたか，(b) 発言のどの部分の再生中に行われたか，という情報を相互編集機能の代替として使用する．この操作体系は，我々が日常行っている自然会話から類推しやすいインタフェースであると考えられる．また，メッセージの関連付け操作をユーザに委ねているため，システム側がメッセージ内容を理解する必要がない．基本的に音声入出力以外のデバイスを必要としないため，電話やモバイル環境に適したインタフェースとなることも期待できる．

4. AVM システムの設計

我々は前章までの議論をふまえて非同期型音声会議システム AVM（Asynchronous Voice Meeting）の設計を行った．本章ではその詳細について述べる．

4.1 サーバ・クライアント構成

AVM システムはメッセージサーバとクライアントから構成される．ユーザはクライアントを用いてメッセージの録音を行い，録音されたメッセージはサーバに集中的に蓄積される．またサーバは，クライアントからの要求に応じて，複数のメッセージを一つの連続した音声ファイルに編集してクライアントに送信する．この編集は動的に実行されるものであり，サーバには常に，クライアントで録音された個々のメッセージがそのままの形で蓄積される．

4.2 利用方法の流れ

AVM システムのユーザから見た操作の流れは次のようになる．

（1） 参加したいグループを選んで，過去に発言されたメッセージの一覧を取得する．

（2） メッセージの一覧から特定のメッセージを選択して，音声を取得する操作を行う．

（3） 取得された音声を再生しながら，それにオーバーラップするように返答を発声し，録音する．

（4） 返答音声を聞き返し，録音を取り消すならば (3) に戻る．

（5） 録音された返答音声をサーバに登録する．

4.3 通信プロトコルとデータ構造

AVM システムでは XML(注2) 準拠の情報ファイルである AVML ファイルと音声ファイルとがサーバ・ク

(注2)：Extensible Markup Language, http://www.w3.org/XML/

```
<?xml version="1.0" ?>
<avml>
  <segment mesid="1" sender="nishi" playtime="0"
     mestime="0" length="1.5" indent="0">
    <text mesid="1" begin="0.0" end="1.01">
      Good morning!
    </text>
  </segment>
</avml>
```

図 1　サーバが生成する AVML 情報の例
Fig. 1　An example of AVML generated by the server.

```
<?xml version="1.0" ?>
<avml sender="canny">
  <message parent="2" reltime="0.4" length="0.3"
     overlap="1">
    <text begin="0" end="0.3">
      Yes.
    </text>
  </message>
</avml>
```

図 2　クライアントが生成する AVML 情報の例
Fig. 2　An example of AVML generated by the client.

ライアント間で転送される．音声ファイルは非圧縮のフォーマット（11.025KHz サンプリング，16bit 量子化，モノラル）を用いているが，効率的な転送のために将来的には圧縮を施す予定である．

（1）通信プロトコル

ファイル送受信に使用するプロトコルとしては WWW で用いられる HTTP(注3) の GET および PUT メソッドを流用した．URL の path 部分を用いてグループ名とデータを表現する．例えばグループ名 room1 の AVML 情報を指定する場合には /room1/text/avml が path として用いられる．また，メッセージ ID を指定するために path の末尾に ?index=1,2,3 といった形式の文字列が付加される．

（2）サーバが生成する AVML 情報

サーバが生成する AVML 情報は，グループ内のメッセージ一覧として単独で使用される．また，音声ファイルを編集してクライアントに送信する際には，その音声に付随する属性情報としても使用される．

1 つのファイルは複数の segment エンティティによって構成される．segment は 1 つの音声ファイルを時系列上のいくつかの音声区間に分割したものであり，segment エンティティは，その音声区間がどのメッセージのどの部分から生成されたかを示す．segment エンティティの属性には，mesid（元メッセージの ID），sender（元メッセージの発言者），mestime（segment 先頭に対応する元メッセージ上の位置），playtime（segment 先頭に対応する音声ファイル上の位置），length（segment の長さ），indent（ツリー表示時の階層の深さ），がある．また，segment エンティティはメッセージ自身の内容や他の話者の相槌などを表す text エンティティを含むことができる．サーバが生成する AVML 情報の例を図 1 に示す．

（3）クライアントが生成する AVML 情報

クライアントが生成しサーバに送信する AVML 情報は，発言された音声をサーバに登録する際に用いられるものであり，音声区間として切り出された 1 つの範囲が 1 つの message エンティティに対応する．その音声が録音されたときに再生されていたメッセージ（親メッセージ）に関する情報を音声に付与するのが目的である．message エンティティの属性には，parent（親メッセージの ID），reltime（新規メッセージ先頭に対応する親メッセージ上の位置），length（新規メッセージの長さ），overlap（新規メッセージが相槌であるか否か），がある．overlap 属性については 4.5 節で詳しく述べる．クライアントが生成する AVML 情報の例を図 2 に示す．

4.4　メッセージの録音と関連付け

AVM クライアントにおいては，既存のメッセージを再生しながら，全二重的に新規メッセージが録音される．録音された音声は始終端検出によって無音部分が除去される．また，再生中のメッセージの segment 情報に基づいて，既存メッセージとの相対的な時間関係が新規メッセージに付与される．segment 情報は，サーバで編集された音声の特定の区間が元のメッセージのどの位置に相当するかを示しているため，録音されたメッセージがサーバに登録される際には，サーバに登録されている元メッセージと，新規にサーバに登録されたメッセージとの相対的な時間関係を保存できる．サーバ上でのメッセージの関連付けデータの構造を表 2 に示す．

4.5　BISP 機能

本システムの予備的な実装による実験 [13] を行ったところ，音声メッセージを再生しながら新規メッセージを録音する際に，再生されている音声を一時停止するか否かを適切に制御する必要があることが明らかになった．つまり，既存メッセージの再生を行いながら長い発話の録音を行うと，録音中にシステムが再生し

(注3)：Hypertext Transfer Protocol HTTP/1.1, RFC2068.

表 2　メッセージの関連付けデータの構造
Table 2　Data structure of the message relations.

フィールド名	内容
mesid	メッセージ ID
length	メッセージの長さ（秒）
parent	親メッセージ ID
offset	親メッセージとの開始時間の相対位置（秒）
memberid	発言者 ID
wavefile	音声ファイル名
overlap	オーバーラップ属性
date	メッセージ登録日時

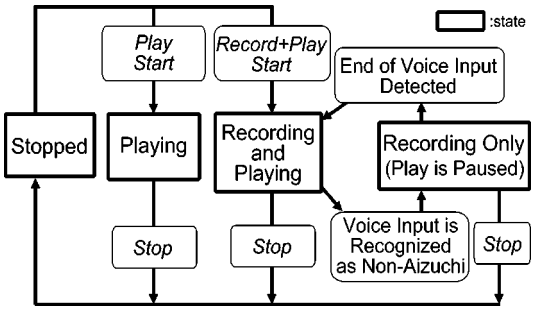


図 3　クライアントの状態遷移図
Fig. 3　State transision chart of the client.

てしまった既存メッセージの内容をユーザが把握することができず，再度既存メッセージを聞き返さなくてはならなくなる．しかし，録音すべき音声区間において常に既存メッセージの再生を止めてしまうと，たとえ相槌であっても必ず再生が止まってしまうため，相槌を打たない方がメッセージを聞きやすい，という事態が生じる．

また，サーバでのメッセージ編集においては，発話時間の長いメッセージ同士が重なって再生されると内容を聞き取ることが難しくなる．このため，長いメッセージに関しては，親メッセージにオーバーラップせずに挿入するような編集が望ましい．しかし相槌などは親メッセージにオーバーラップするような編集をしたほうが，再現される会話の自然性が高まる．

そこで，ある発話区間が単なる相槌（相槌メッセージ）であるか，内容的に意味を持つ発言（非相槌メッセージ）であるかをオーバラップ属性によって区別し，サーバに登録することとする．発言内容を再現する場合には，オーバラップ属性に応じて，相槌であれば親メッセージと重ね合わせ，非相槌であれば親メッセージに割り込ませる形でメッセージを編集し，仮想的な対話音声を再現する．このようなサーバ側の処理は次のアルゴリズムで実現される．

（1） メッセージ ID のリストをクライアントから受け取り，そのリスト中からルート（根）となるメッセージを検索する．

（2） ルートメッセージの子となるメッセージを検索し，非相槌メッセージのみを親メッセージの対応する位置に再帰的に挿入する（オーバーラップさせない）．このとき，対応する segment 情報も同時に作成する．

（3） 全ての非相槌メッセージの挿入を繰り返して作られた音声に対して，相槌メッセージの重ね合わせ（オーバーラップ）を行う．相槌メッセージはそれぞれ親メッセージとの相対時間によって管理されているので，segment 情報を用いて重ね合わせる場所を決定する．

また，クライアント側での新規発言の録音においては，始終端検出と同時に，発話長によって相槌であるか非相槌であるかを簡易的に検出するようにした．新規発話の発話長が短ければ相槌とみなし，再生中の音声は途切れることなく，単に録音だけが行われる．しかし，新規発話が一定の長さ（現在の実装では 1.0 秒）を超えると，新規発話の終端を検出するまで既存発話の再生を中断し，この新規発話を非相槌として保存する．この機能を BISP（Barge-in to Stop Playing）と呼ぶ．これにより，任意のタイミングで自由に発話した相槌を再現できると同時に，再生中に新たに長い発話を行っても，再生中の既存発話の内容を聞き漏らすことがなくなった．この機能を実現したクライアントの状態遷移図を図 3 に示す．

4.6　既読管理機能

1 つのグループに多くのメッセージが蓄積されていくと，ユーザがすでに聞いた発言がどれであるかを把握することが困難になる．これを解消するために，サーバ側でユーザごとに既読メッセージの ID を管理し，クライアントでの一覧表示時に既読メッセージをマークで示す機能を設けた．

5.　評価システムの構成

第 4 章で述べた設計に基づいて，以下のような評価システムを構築した．

5.1　サ　ー　バ

AVM サーバ Voxer は移植性を考慮して Perl および C 言語によって実装されており，HTTP によるク

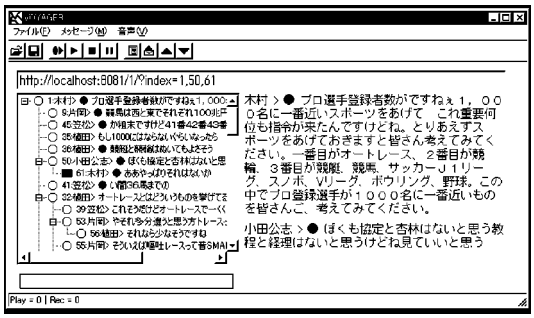


図 4　クライアントの画面表示
Fig. 4　Screen image of the client.

ライアントからの要求を処理する．音声ファイルとメッセージ間の関連付けなどの情報を蓄積するデータベース機能と，再生用音声の編集機能を備える．Linux 上でも動作するが，後述する実験では Windows 上で運用した．音声認識機能の実装も行っているが，本実験では使用していない．

5.2　クライアント

AVM クライアント Voyager は Microsoft Visual C++ を用いて実装され，全二重で音声入出力が可能な Windows98 システム上で動作する．HTTP によるメッセージの送受信機能と，BISP 機能を含む音声の録音・再生機能，音声メッセージのツリー表示機能などを備える．再生中の音声がそのまま全二重で録音されることを防ぐために，メッセージの録音と再生にはヘッドセットを用いる．Voyager の画面表示を図 4 に示す．左側はツリー表示によるメッセージの選択ウィンドウで，右側は音声認識によってつけられた text エンティティ（4.3 節参照）に基づいてメッセージの内容を表示する部分である．上部ツールバーには「再生＋録音」「再生のみ」「停止」などのボタンがあり，下部にはマイク音量が表示される．

6. 実　　験

6.1　実験方法

議題として「次にあげるスポーツの中で，プロ登録者数が 1000 人に近いスポーツを挙げて下さい．オートレース，競艇，競輪，騎手（中央競馬会），サッカー J1 リーグ，スノーボード，V リーグ，ボウリング，野球.」というクイズを与えて，AVM システムまたは電子掲示板（BBS）を用いて議論をさせた．

このような設問においては，すべての項目について深い知識を持っている参加者はいないが，参加者がそれぞれの知識を相互補完的に提供し合うことが可能であり，双方向的な議論によって正解に近づくことが期待できる．議論の結果が正解に近いかどうかによって活発な議論が行えたかどうかを判断できると同時に，議論が収束するまでのシステム利用回数や発言回数などを定量的に評価することもできる．このような観点から本実験を計画した．

被験者たちが真剣に議論することを促すために，1000 人により近いスポーツ名を回答したチームには報酬を多く支払うことを予告した．

AVM システムとしてはクライアントとサーバを 1 台の Windows98 搭載 PC で実行させた．BBS 用のソフトウェアとしては WWW サーバ上で動作し，発言をツリー構造で管理できる WebForum(注4) を用いた．

被験者は理工系の大学研究室に所属する 20 代男性の学生（音声の研究に従事する学生を含む）10 名であり，全員がキーボード操作に熟練している．この 10 名が 5 名ずつ 2 チームに分かれて，AVM と BBS を各 1 チームが用いて実験を行なった。各チームから議長を 1 名ずつ選出し，まず議長からチームのメンバーに対して，各システムを用いてクイズの問題を通知させた．以後は乱数により 1 名ずつユーザを選び，各システムを交替で使用させた．

AVM の音声認識機能に関しては，実用的な音声認識性能を保ちつつ，ある程度の誤認識を含んだテキストを得るために，ユーザが使い終わるたびに，新たに発言された音声をオペレータが ViaVoice98（日本 IBM 社製の音声認識ソフト）に対して再度読み上げて，認識結果を AVM サーバに登録することとした．

議論によってチーム全員による結論が得られたら，議長がオペレータに口頭で回答することとした．最後に各被験者に対してアンケートを行った．

6.2　実験結果

議論の結果，どちらのチームも正解または正解に近い回答が得られた（勝者は AVM チームであった）．発言の定量的分析を表 3 に，メッセージあたりの文字数の比較を図 5 に示す．AVM におけるメッセージの文字数は，実験終了後に音声を再度手作業で書き起こしたテキストを用いて求めた．また，BBS の発言においては引用部分を除いて文字数を求めた．

ViaVoice98 によるメッセージの認識性能は，単語認識率 85.2%，認識精度 82.5%であった．

(注4)：http://www.kent-web.com

表 3　AVM および BBS におけるメッセージの分析結果
Table 3　Analysis of the messages on AVM and BBS.

	AVM	BBS
システムののべ使用回数	20	24
メッセージ数	71	24
総発言文字数（引用を除く）	2303	5657
メッセージの平均文字数	32.4	235.7
総発言時間（秒）	501.3	–
平均発言時間（秒）	7.1	–
オーバーラップ発話数	12	–
非オーバーラップ発話数	59	–

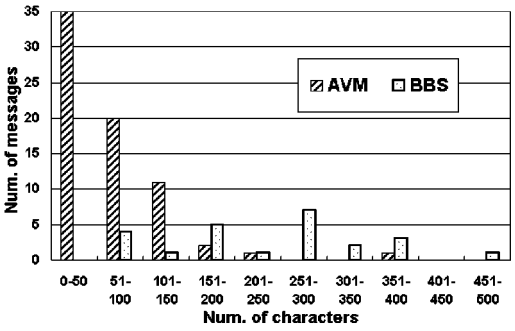


図 5　メッセージの文字数の比較
Fig. 5　Length of the messages on AVM and BBS.

表 4　アンケート結果の平均
Table 4　Result of questionnaire.

質問内容	AVM	BBS
(a) 納得できる答えを得られた	4.4	3.6
(b) 自分の意見を十分に言えた	4.0	4.4
(c) 活発な議論が行えた	3.8	3.4
(d) 議論の流れに不自然さがなかった	4.0	4.0
(e) 実際に集合して交わす議論と雰囲気が近い	2.2	2.8
(f) 反論・同意といった意見を出しやすい	4.4	3.8
(g) メッセージを入力しやすい	3.4	4.2
(h) メッセージ内容の把握がしやすい	3.4	4.0
(i) 会議全体の流れを把握しやすい	3.4	3.8
(j) ユーザインタフェースがわかりやすい	3.8	4.0
(k) 音声再生よりも文字表示をよく利用した	4.0	–
(l) 誤認識を含むテキストの利用価値があった	4.6	–

各チームのアンケート結果（1〜5 の 5 段階評価，1 が「まったくそう思わない」，5 が「とてもそう思う」に対応）の平均値を表 4 に示す．ただし項目（k）(l) は AVM のみで行った質問である．

6.3 検　　討

ここでは，「AVM によって話し言葉的な音声会話を非同期的に実現できた（仮説 1)」「AVM において話し言葉の漸次性が生かされた（仮説 2)」「BISP が有効に機能し，発言しやすさに貢献した（仮説 3)」「誤認識を含んだ認識結果が有効に活用できた（仮説 4)」「AVM は BBS の代替として充分に機能した（仮説 5)」という仮説を挙げ，実験結果がこれらを支持したかという観点から検討を行う．

まず仮説 1 について検討する．AVM では，例えば「サッカー言うたら今ー，20 チームぐらい．で 1000 人やとしたら 1 チーム 50 人．そんなにおらんでしょう.」といった口語的でくだけた表現が多用された．これに対して BBS では，「クイズの僕なりの解釈ですが，みんなの言われているように野球は 1,2 軍あわせると 1000 ÷ 12=84 人以上いるような気がします．しかし，1 球団で選手登録人数の上限が決められていますので……」といった書き言葉による表現が多用された．被験者はすでに互いに親しい間柄であるため，グループによる差異は考えにくい．従ってこの実験結果は仮説を支持している．

仮説 2 については，これが成立していれば，第 2 章で述べたように，情報が省略されたり，簡潔で断片的な表現が多用されたはずである．AVM によって総発言文字数が 48%になったこと（表 3）や，各メッセージの文字数が少ない値に集中したこと（図 5)，「あー近いといえば近いですよねー」といった文脈に依存した発言が多くみられたことなどが，この仮説の正しさを裏付けている．

仮説 3 については，発言の 17%を占めるオーバーラップ発話を聴取したところ，発話区間の検出に失敗して断片化されたものが多かった．これは表 4 の項目 (g) の評価の低さとも関連しており，オーバーラップ発話の処理が不充分だったことを示す．しかし，表 4 の項目（a）〜（d）の評価は高く，本実験の用途での発言しやすさは実現されていたと言える．

仮説 4 については，表 4 の項目（k）(l) の評価が高得点であることと，アンケートの自由記述における「大まかな内容をテキストで確認し，詳しい内容は音声を再生し確認した」「(文字情報があれば）いろんな箇所の声を聞くより手間が少なくなる」などの回答によって支持されたと考えられる．

仮説 5 については，表 4 で BBS と比較して AVM が著しく劣っている項目がなかったこと，議論の結果どちらも適切な答えが得られたこと，などにより支持されたと考えられる．ただし表 4 の項目（e）の評価が BBS と同様に低かったことから，対面での議論の代替として AVM を位置付けることは難しい．

以上より，本実験によって仮説 1,2,4,5 は全面的に，仮説 3 は部分的に支持されたと考えられる．

7. むすび

非同期・蓄積型で双方向通信が可能な音声会議システム AVM を提案し，その設計と評価について述べた．電子メールや電子掲示板などが持つ非同期型通信の利点を損なわずに，表現力が高く発言しやすい，という音声メッセージの利点を生かせたことが確認できた．

今後の予定としては，サーバに実装中の音声認識について，特に話し言葉での性能向上を目指す．また，既読発言の再生を省略したり，メッセージが単語の途中で分割されることを防ぐなど，より会話しやすくするための改良が必要である．また，キーボード操作に不慣れなユーザなど，多様なユーザによる大規模な運用実験を行う必要がある．さらに，電話回線や携帯情報端末による AVM の実現方法についても検討し，本システムが幅広いユーザに利用されるようにしていきたい．

文　献

[1] M. J. Hirayama, T. Sugahara, Z. Peng, J. Yamazaki, Interactive listening to structured speech content on the Internet, Proceedings of ICSLP'98, pp.1627–1630, Dec. 1998.

[2] 岡田美智男, 口ごもるコンピュータ, 共立出版, 東京, 1995.

[3] 西本卓也, 新美康永, 非同期音声メッセージシステムの設計, 信学技報, MVE97-98, pp.39–46, Feb. 1998.

[4] 榎本美香, 土屋俊, オーバーラップ発話の評定方法とその基礎統計～日本語地図課題対話を通して～, 情処研報, 99-SLP-29-25, pp.145–150. Dec. 1999.

[5] 川口由紀子, 土屋俊, ターン交替規則の破綻例の会話の含みによる説明の試み～日本語地図課題対話を通して～, 情処研報, 99-SLP-29-26, pp.151–156, Dec. 1999.

[6] 西宏之，五味和洋, 小島順治, 音声対話における確率的発声終了検出法, 信学論 D, Vol.J70-D, No.11, Nov. 1987.

[7] 向後千春，山西潤一, あいづち留守番電話の試作, 日本認知科学会第 8 回大会発表論文集, pp.72–73, Jul. 1991.

[8] N. Ward, Using prosodic clues to decide when to produce back-channel utterances, Proceedings of ICSLP'96, pp.1728–1731, Oct. 1996.

[9] 菊池英明, 杉田洋介, 白井克彦, 自由会話における時間的制約の影響の分析, 人工知能学会研究会資料, SIG-SLUD-9702-5, pp.31–36, Oct. 1997.

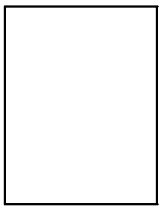
[10] 堀内靖雄, 小磯花絵, 土屋俊, 市川薫, 自発的音声対話における話者交替の制御に関する発話末の統語的・韻律的特徴, 情処研報, 96-SLP-10-9, pp.45–50, Feb. 1996.

[11] 岡登洋平, 加藤佳司, 山本幹雄, 板橋秀一, 相槌を打つ音声対話システムの評価, 人工知能学会研究会資料, SIG-SLUD-9804-2, pp.7–12, Feb. 1999.

[12] J. Hirasawa, N. Miyazaki, M. Nakano, T. Kawabata, Implementation of coordinative nodding behavior on spoken dialog systems, Proceedings of ICSLP'98, pp.2347–2350, Dec. 1998.

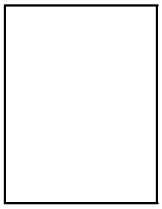
[13] T. Nishimoto, H. Yuki, T. Kawahara and Y. Niimi, An asynchronous virtual meeting system for bi-directional speech dialog, Proceedings of Eurospeech'99, pp.2471–2474, Sep. 1999.

（平成 x 年 xx 月 xx 日受付）



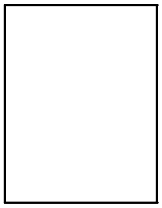
西本　卓也　（正員）

1993 早稲田大学理工学部卒．1995 同大大学院理工学研究科修士課程了．1996 京都工芸繊維大学工芸学部助手．音声対話システムの研究に従事．日本音響学会，情報処理学会，人工知能学会，ヒューマンインタフェース学会会員．



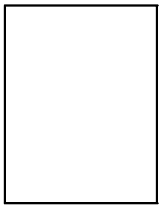
幸　　英浩

1998 京都工芸繊維大学工芸学部卒．現在，同大大学院工芸科学研究科修士課程．音声対話システムの研究に従事．



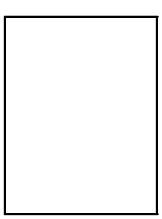
川原　毅彦

1999 京都工芸繊維大学工芸学部卒．現在，同大大学院工芸科学研究科修士課程．音声対話システムの研究に従事．



荒木　雅弘　（正員）

1988 京都大学工学部卒．1990 同大学院修士課程修了．1993 同博士課程単位取得退学．同年京都大学工学部助手．1997 京都大学総合情報メディアセンター講師．1999 京都工芸繊維大学工芸学部助教授．音声言語理解の研究に従事．博士 (工学)．



新美　康永　（正員）

1962 京都大学工学部卒．1964 同大大学院修士課程了．同年同大工学部助手．1970 京都工芸繊維大学工芸学部助教授．1988 同教授．音声認識，自然言語処理などの研究に従事．工学博士（京都大学）．情報処理学会，日本音響学会，人工知能学会，ESCA 等会員．