

Title	ChaTEL : マルチスレッド対話を容易にする音声コミュニケーションシステム
Author(s)	小倉, 加奈代; 西本, 一志
Citation	情報処理学会論文誌, 47(1): 98-111
Issue Date	2006-01-15
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/4043
Rights	<p>社団法人 情報処理学会, 小倉加奈代/西本一志, 情報処理学会論文誌 = Transactions of Information Processing Society of Japan, 47(1), 2006, 98-111. ここに掲載した著作物の利用に関する注意: 本著作物の著作権は(社)情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。 The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright (C) Information Processing Society of Japan.</p>
Description	

ChaTEL: マルチスレッド対話を容易にする 音声コミュニケーションシステム

小倉 加奈代[†] 西本 一 志[†]

日常の対話では、対話参加者全員が単一の話者を一定時間共有し、その話題に関連がある発話を行う必要がある。しかも、基本的に話し手は1人だけで他は聴き手となることが求められるため、今交わされている話題とは直接に関係ないが、何か興味深い話題やすばらしいアイデアを思いついたとしても、それを思いついたときにすぐ話すことは許されない。その結果、せっかく思いついた内容そのものを忘れてしまうことがしばしば起こる。このように、音声対話は非常に効率が悪い側面を有している。そこで本論文では、リアルタイム・マルチスレッド対話を容易に実現可能とする音声コミュニケーションシステムを提案する。試作システム ChaTEL を用いて被験者実験を行ったところ、本システムで採用した相手指定/先行発言対応機能により、これらの機能を持たない単純な音声コミュニケーションシステムよりも1秒単位でのスレッド数、分岐数、最大同時存在スレッド数、複数スレッド同時存在の継続時間がいずれも多くなることを確認し、本システムにより音声でのマルチスレッド対話が可能となることが明らかとなった。

ChaTEL: A Voice Communication System That Facilitates Multi-threaded Communication

KANAYO OGURA[†] and KAZUSHI NISHIMOTO[†]

We usually have to share single topic-thread in an everyday conversation for a while and to talk about what is related to the current thread. Moreover, we have to keep a rule of turn-taking: there can always be only one speaker and the others should be listeners. Therefore, it is impossible for people who do not have "turn" to talk about what they want to talk even if they think of interesting topics or good ideas. As a result, regrettably, they often forget the topics or ideas. Thus, such a daily conversation has an inefficient aspect. Hence, in this study, we develop a novel communication system to achieve "multi-thread voice communication", named "ChaTEL". We show results of experiments with subjects for evaluating our system. Based on the results, we confirmed that, with using ChaTEL, a number of threads, a number of adjacent pairs, maximum number of concurrent threads and the duration of multi-threaded conversations are more or longer than those of a simple voice communication system. Consequently, we can conclude that ChaTEL facilitates the multi-threaded conversations.

1. はじめに

本論文では、音声によるリアルタイム・マルチスレッド対話を容易に実現可能とする音声コミュニケーションシステム "ChaTEL" を提案する。これまでに実施したテキストチャットにおけるマルチスレッド対話に関する分析研究から得た知見に基づき、我々はマルチスレッド対話を支援する機能を ChaTEL に組み込んだ。そこで、この支援機能を持たない単純な音声コミュニケーションシステムと ChaTEL とを比較することにより、ChaTEL を用いた場合に音声対話において

も対話がマルチスレッド化しやすく、しかも長時間マルチスレッド状態を維持できることを示す。

なお、ここでいう「マルチスレッド対話」とは、1つの対話空間で複数の話題に関する対話スレッドが同時に進行しており、かつ1人の話者が同時に複数の対話スレッドに参加しているような対話のことである。立食パーティーなどでは、1つの対話空間で複数の話題に関する対話スレッドが同時に進行しているケースが見られるが、通常1人の話者は1つの対話スレッドにしか参加していない。すなわちこれは対話空間の単なる分割にすぎず、各スレッドは互いに独立で無関係であるため、本論文ではこのような状態をマルチスレッド対話とは見なさない。

また本論文でいう「リアルタイム・コミュニケーション

[†] 北陸先端科学技術大学院大学

Japan Advanced Institute of Science and Technology

ン」とは、話し手による発言の送信 → 聴き手による発言の受信 → 内容理解 → 返答生成 → 返答送信、という対話遂行のための一連の処理において、ある処理と次の処理との間の時間が無視できる程度の範囲で各処理が接続して実行されるコミュニケーションのことである。対面や電話での音声対話では、一般に発言の送信から内容理解までのプロセスが同時並行的に実施され、話し手による発言の送信が完了すると、接続して返答生成と返答送信が同時並行的に実行される。テキストチャットでは、個々の処理が並行的にではなく逐次的に行われるが、一般に各処理はやはり接続して実行される。したがって、これらのコミュニケーションはいずれもリアルタイムなものであると見なされる[☆]。一方、電子メールのやりとりは、相手によるメールの送信と読み手によるメールの受信の間に無視できない長い「間」が一般的に存在するため、リアルタイムなコミュニケーションとは見なさない。

なお、3人以上でなされる対話の場合、ある人の発言に対して必ずしも全員が応答する必要はなく、上記のプロセスが内容理解の段階までで停止する人もいる。またマルチスレッド対話状況の場合には、直前の発言に対してではなく、時間的に大きく遡った先行発言に対して応答することもある。これらの場合についても、対話全体として一連の処理が接続して実行されていると見なされる場合（つまり、現在なされている対話内容に関して、対話参加者全員がいずれの処理もいっさい行っていないと見なしうる状況が生じていない場合）、やはりリアルタイムなコミュニケーションが行われていると見なす。

音声は、我々にとって最も馴染み深く使いやすい意思疎通手段である。しかも、テキストによるコミュニケーションでは欠落する、イントネーションなどの豊富な非言語情報も伝達することができる。しかしながら、これらの音声対話の特長を考慮しても、現在の音声による対話は必ずしも効率が良いとはいえない。その原因は、音声対話が持つ「同期性の制約」にあると考えられる。対面での対話や会議、あるいは電話や遠隔会議などの音声による対話では、対話参加者全員が単一的话题を一定時間共有し、その話題に関連がある発話を行う必要がある。しかも、基本的に話し手は1人だけで他は全員聴き手となることが求められ、話者交替規則のもとで話し手は順番に交替しなければな

らない。これが同期性の制約である。このため、今交わされている話題とは直接に関係ないが、何か興味深い話題やすばらしいアイデアを思いついたとしても、それを思いついたときにすぐ話すことは許されない。その結果、せっかく思いついた内容そのものを忘れてしまうことがしばしば起こる。このように、音声対話は実は非常に効率が悪い側面を有している。

同期性の制約は、人の音声識別能力の限界と短期記憶容量の少なさに起因する。互いに内容が異なる複数の発言を同時に聞いても、それらを区別して聞き分けることは難しく、しかもそのすべてを記憶して、個々に適切な応答を行うことは一般に不可能である。したがって、この制約を解消するには、すべての発言を区別して記憶し、どの発言に対してもいつでも容易にアクセスすることを可能とする手段が必要となる。

テキストチャットは、このような手段をすでに実現しているリアルタイムコミュニケーションメディアである。テキストチャットでは、すべての発言は個々に時系列的に発言履歴上に記録される。また、個々の発言には発言者のログイン名が付与される。そのため、どの参加者のいつの発言についても、容易かつ正確に随時読み出すことができる。この結果、特にテキストチャットに慣れた利用者間では「マルチスレッド対話」が日常的かつ意図的に行われていることが示されている¹⁾。しかしながら、音声を用いたコミュニケーションシステムで、リアルタイム性を持ち、しかもマルチスレッド対話を容易に行えるものはまだ現れていない。一般に従来の音声コミュニケーションシステムでは、マルチスレッド対話は想定外であり、むしろ「好ましくない対話状況」として抑制される傾向すらある。

我々は発想を180度転換し、音声対話における同期性の制約を解消することにより、音声対話のマルチスレッド化を目指している。音声リアルタイム・マルチスレッド対話は、多人数での雑談的対話にももちろん有用である。しかしより実用的には、思いついたアイデアを漏れなく対話の場に乘せ、それについて議論することが可能となるため、新たなアイデアをできる限り多く生み出すことが要求される「ブレインストーミング」などの発散的思考²⁾を行う対話で特に有効であり、アイデアの生産性が向上することが期待される。また近年、携帯電話やインスタント・メッセージング・サービスの急速な普及にともない、コミュニケーションのモバイル化・ユビキタス化が進行し、「いつでも、どこでも」コミュニケーションをとることが可能となった。しかしながら、特に音声によるコミュニケーションそのもののスタイルは、先述のとおり旧

[☆] 内容理解や返答生成に時間を要して長い無発話時間が生じていても、この間は内容理解あるいは返答生成の処理が実行されており、いずれかの処理と処理との間で時間を要しているわけではないので、リアルタイム性は失われない。

態依然で非効率的である。モバイル/ユビキタス・メディアによるコミュニケーションの効率化をさらに推し進めるためにも、音声対話のマルチスレッド化は不可欠であると考えられる。また、特に屋外や移動中の使用状況では、音声による入力とは文字入力に比べて圧倒的に簡単かつ便利で使いやすい。以上のように、音声対話のマルチスレッド化は、今後のITによる知識創造社会において、非常に重要な意義を持つと思われる。

そこで本論文では、容易にリアルタイム・マルチスレッド対話を行うことを可能とする音声コミュニケーションシステムであるChaTELを提案する。ChaTELは、音声のみを用いるコミュニケーションシステムであるが、一般的なテキストチャットシステムと同様に人間の短期記憶を補助する発言履歴を持つ。このため、利用者は随時任意の発言を聴取し、それに対して即座に応答を作成・送信することができるので、同期性の制約が解消される。ただし、音声は聴取しない限り内容を把握できないため、発言履歴を用意するだけでは、どの発話とどの発話が同じスレッドに属するかなどの対話スレッドの状況を把握することが依然として難しく、マルチスレッド化が起こりにくいことが懸念される。そこで筆者ら自身によるテキストチャットの分析から得られた知見に基づき、発言間の関連性の把握を容易とし、対話のマルチスレッド化を支援するために、各発言に対して「対話相手指定情報」と「先行関連発言情報」を付与する機能を追加した。実際に開発したシステムと、発言履歴のみを持ち、発言間の関連性を指定する機能を持たないシステム（BaseLineシステム）との比較実験を実施し、得られた対話データを分析することによって、音声対話のマルチスレッド化における本システムの有効性と問題点を検討し、さらに今後の展望について考察する。

以下、2章では、関連研究について述べ、3章では、マルチスレッド対話を実現するための要件について検討する。4章では、3章で示した要件に基づき開発したマルチスレッド指向ボイスチャットシステムChaTELのシステム概要、構成について述べ、5章では、ChaTELを用いた評価実験の概要と実験の分析結果を示す。6章では、実験結果に基づき、音声対話のマルチスレッド化の可能性について議論するとともに、提案システムのユーザインタフェースについて検討する。7章では、まとめと今後の展望について述べる。

2. 関連研究

計算機とインターネットを利用した音声によるコミュニケーションシステムで、すでに実用化されてい

るものには、ボイスメモやボイスメール、ボイスチャット、IP電話などがある。

ボイスメモは、思いついたアイデアや各種情報をボイスレコーダなどを利用して声で記録しておき、あとで個人的に利用するシステムであり、本来はコミュニケーションを意図したものではない。留守番電話はボイスメモを他者への情報伝達に応用した例と見ることができるが、この機能だけを用いて双方向的なコミュニケーションを行うことは、不可能ではないが現実的ではなく、一般には単一の伝言を一方向的に伝えるためだけに用いられる。ボイスメールは、文字ベースの電子メールに補完的な情報として音声データを添付して送るものであり、音声データのみを継続的にやりとりしてコミュニケーションすることは一般に想定されておらず、やはり留守番電話の場合と同様に音声によるコミュニケーションは一方向的な使い方が主となる。またボイスメモやボイスメールによるリアルタイムコミュニケーションは一般には実施できない。なお、ごく最近、MSNメッセンジャー7.5³⁾に「ボイスメモ」という機能が追加された。これはテキストベースのインスタントメッセンジャーシステムに、短い音声データを録音して相手に送信する機能を加えたものであり、従来のボイスメモとは異なり、音声によるコミュニケーションが可能である。ただしMSNメッセンジャーの場合は、あくまでテキストが主として用いられ、音声のみによるコミュニケーションは想定されておらず、ボイスメモは補助的な情報送信用として提供されていると思われる。

代表的な既存ボイスチャットシステムとしては、MSNメッセンジャー³⁾とYahooメッセンジャー⁴⁾のボイスチャット機能がある。MSNメッセンジャーのボイスチャット機能は、従来の電話と同様の1対1での対話のみに対応している。Yahooメッセンジャーのボイスチャット機能は、多人数でのチャットでも使用可能だが、ある瞬間に話せるのは1人に制限されており、その間他者は音声での発言はできない半二重通信形態になっており、話者交替規則が前提となっている。また、いずれについても音声での発言に関する履歴はいつさい残らない。したがって、任意の発言を随時聞き、それに答えるようなことはできない。VoIPを用いた安価な音声通信システムであるIP電話やSkype⁵⁾などは、まさに電話の機能そのものを計算機上に移植したものである。結局、これらのシステムを用いたコミュニケーションは、リアルタイムであると同時に同期性の制約を持つものとなるため、マルチスレッド対話を行うことはほぼ不可能である。

一方、研究レベルでは音声による複数スレッドの同時対話を実現しようとする試みは、わずかに存在する。Bercら⁶⁾は、遠隔ビデオ会議中に、並行して特定メンバーどうしだけで“ささやく”というサイド・コミュニケーション機能を付与した Pssst というシステムを開発している。参加者全員による議論と同時に“ささやき”による議論も起こっているという意味でマルチスレッド状態と見なせるが、“ささやき”に参加していない人にとってはマルチスレッド状態になっていない。また、“ささやき”に参加している人にとっては、おそらく“ささやき”だけがその時点での参加対話スレッドとなっていると思われる。結局これは、ごく短期的な対話空間の単純分割と見る方が正しいと思われる。

西本らの VoiceCafe⁷⁾では、発言間の応答関係に基づく木構造情報でスレッドを管理する。発言は音声で入力可能であり、入力された音声発言は音声認識システムを用いて文字化される。その際、最新の発言は、その発言入力の直前に発言者が聴取した発言に対する応答であるという前提に基づき、直前に聴取した発言に対応するノードの子ノードとして木構造情報に追加される。つまりこのシステムは、スレッド型の電子掲示板に似たインタフェースを持つ非同期型音声会議システムである。このシステムでは、複数のスレッドが同時に存在することは可能であるが、マルチスレッド対話の実現が目的ではなく、音声によって入力された発言を、自動的に適切なスレッドの適切な位置に追加することの実現が主目的であると思われる。

Aokiら⁸⁾は、各話者の発話区間と無音区間の相互関係に基づき、空間的位置関係にかかわらず、誰が同一フロアに属する話者かを自動判定し、同一フロアに属する話者の発話音声により明確に聞こえるように、自動的に音響調整する機能を付加した同時的会話環境である“The Mad Hatter’s Cocktail Party”を開発した。しかし、1つのフロア内での対話は基本的に単一スレッドであり、しかもある話者が複数フロアに属することは原理的に無理である。したがって、これも対話空間の単純な分割であり、マルチスレッド対話の実現されているわけではない。

以上のように、今のところ音声によるマルチスレッド対話の実現を主たる目的とする事例は、筆者らの知る限り見当たらない。また、文献7)と8)は、いずれも音声による話者交替規則に基づく自然な対話を可能とするために、余分な操作をなくすべく発言間の対応づけなどを自動化しているが、反面、意図しない関連づけの発生をある程度避けられない。一方本研究は、話者交替規則に基づく自然な音声対話の実現が目的で

はなく、あくまでもマルチスレッド対話を容易に実現できる音声対話システムの構築を目的としている。そこで、本研究では、ユーザの意図を的確に反映することができる、手動によるシンプルな発言間の関連性指定機能を採用する。

3. マルチスレッド対話を実現するための要件

3.1 発言履歴の存在

マルチスレッド対話を行うためには、我々は、一度に複数の発言を聞き分け、記憶し、理解することが必要とされる。しかし、これは人間の認知能力の範囲内では困難である。したがって、通常の音声対話でマルチスレッド対話が行われることはめったにない。一方テキストチャットでは、誰が、いつ、何を発言したかという発言にかかわるデータが保存され、それが発言履歴として表示され、参加者は自由に履歴を閲覧できる。これが人間の認知能力を補う役割を果たしているために同期性の制約が解消される。この結果、テキストチャットではマルチスレッド対話が可能となっていると思われる。つまり音声によるリアルタイム・マルチスレッド対話の実現にも、発言履歴が不可欠であると考えられる。

3.2 発言相手指定と先行発言指定

前述のとおり、発言履歴を提供することにより、任意の発言を随時参照可能となるため、マルチスレッド対話を行うことができるようになる。しかし、スレッド数が多くなったり発話対の間隔が離れたりすると、どの発話がどの発話を受けてなされたものかという発言相互の関係を把握することが難しくなるため、マルチスレッド状況を維持することが実際には困難になる。この状況は、発言履歴を見ただけでは発話内容を把握できない音声対話の場合に、より顕著となる。

このため、テキストチャットにおいては、各発言の文末に、どの話題あるいはどの発話への応答かを特定する表現や、誰に向けての発言か、誰に返答してほしいかを明示する相手指定表現をユーザ自身が自発的に付加することで、この問題を解決している。小倉ら⁹⁾は、この点について、チャット対話収録で得たデータを分析している。分析対象としたデータは、2人対話が3対話311発言分、3人対話が5対話559発言分、計870発言分の発言履歴データであり、すべてチャット経験者を被験者としたデータである。その結果、発言履歴の参照しにくさがある中で、話の流れを追い、できる限りスムーズに対話を進めるために貢献していると考えられる。以下の3種類の発言間関連性指定表現が存在することを明らかにした：

- (1) 誰に向けた発言であるかを明記する表現（固有
名詞を含む）
例) まだまだ今年はこれからですよ。 > Bさん
- (2) どの話題に関連した発言をしているかを明記す
る表現
例) 私も大好きですよ。 > チーズケーキ
- (3) どの発言に対して発言をしているかを明記する
表現（この場合はコピー&ペーストを行っている
と推測される場合）
例) 私も大好きですよ > チーズケーキ > 奇遇
ですねえ。私も負けなくらいにマニアです。

これら3つの表現が対話中に出現した割合を図1に示す。さらに、発言間の意味的つながりの判定作業を行ったうえで、3つの表現を発言間のインターバル（発言間距離）ごとに分類した出現頻度の結果を図2に示す。これらの図中、「>人」は上記の(1)の表現に、「>単語」は(2)に、「>コピペ」は(3)に、それぞれ対応する。図1の結果から、3つの表現をあわせると、分析対象の約1/4にのぼる発言でいずれかの表現を用いていることが分かる。また、図2の結果から、関連する発言間の距離が2以上の場合、すなわち

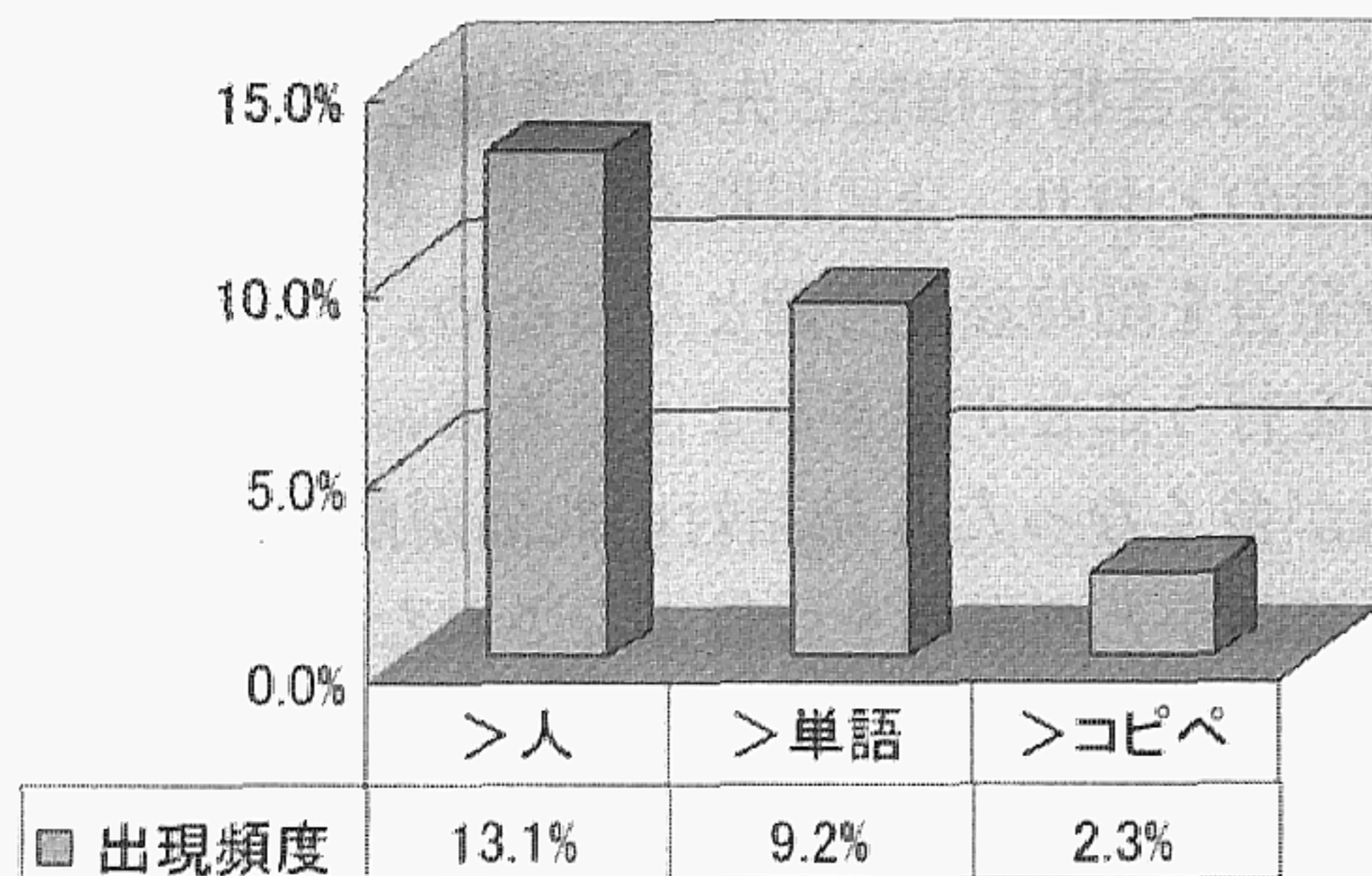


図1 対話中における各発言間関連性指定表現の出現頻度

Fig. 1 Frequencies of usage of each cue expression for specifying relations between utterances in text chat conversations.

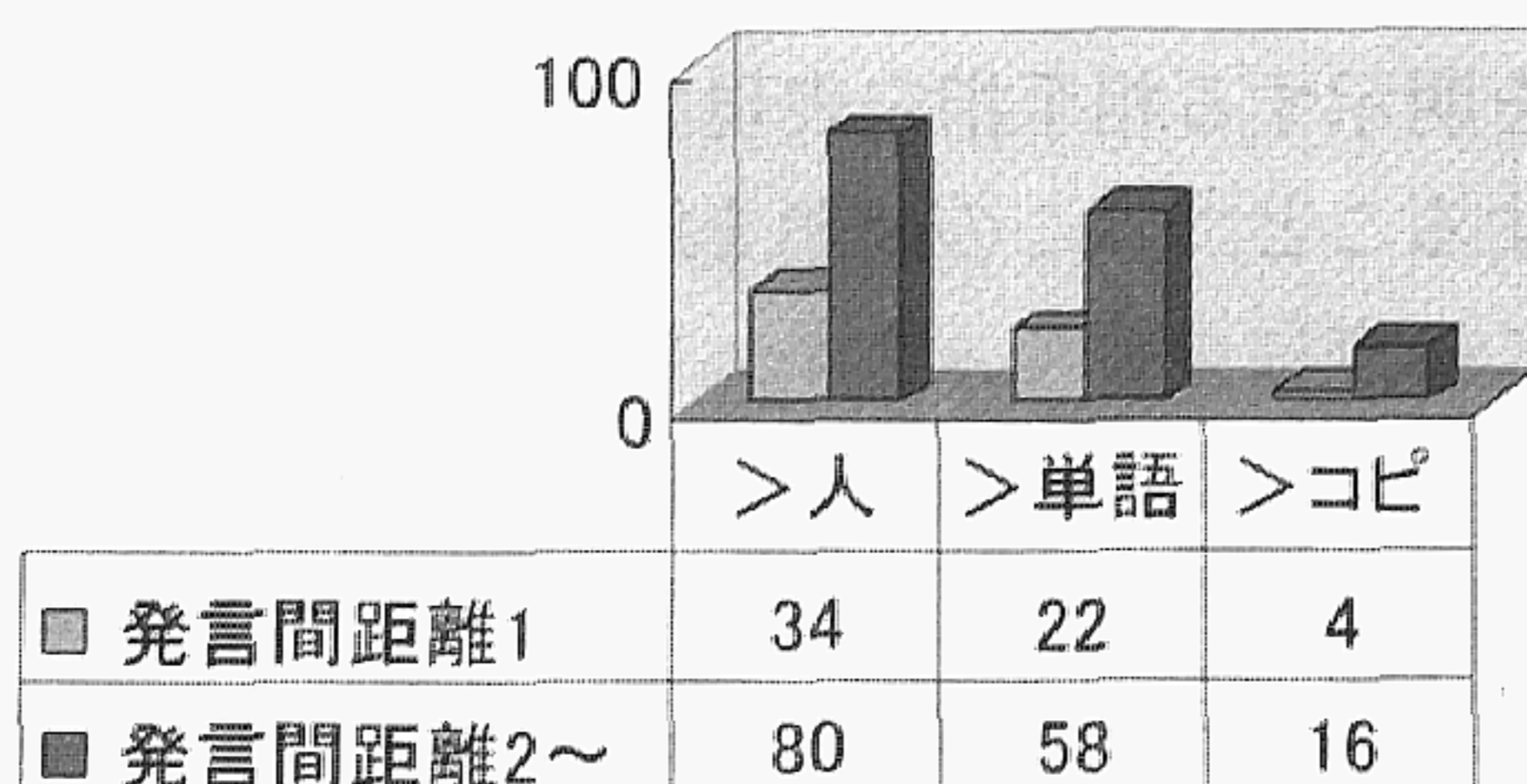


図2 個々の発言間関連性指定表現の出現頻度と発言間距離の関係（発言間距離1とは隣接発言どうしが同一話題であることを示し、発言間距離2以上は、隣接発言どうしが異なる話題であることを示す）

Fig. 2 Relationships among frequencies of each cue expression and interval between related utterances.

隣接する発言どうしが異なる話題である場合に、どの表現も出現頻度が増加していることが分かる。このことから、より複雑な状況になればなるほど、話の流れを追いやすくするために、誰に対する発言なのか、どの話題、どの発言に関する発言なのかを明記して対応していることが推測できる。

4. ChaTELのシステム構成

テキストチャットでの対話のマルチスレッド化に貢献している発言間関連指定表現に関する知見に基づき、音声によるマルチスレッド対話を実現可能とするマルチスレッド指向音声コミュニケーションシステム“ChaTEL”を構築した。ChaTELは発言履歴を有し、しかも発言間関連指定情報として「対話相手指定情報」と「先行関連発言情報」を簡便に付与する機能を有する。これらの発言間関連指定情報と、テキストチャットにおける発言間関連指定表現との対応を表1に示す。本システムは、Microsoft Visual C# .NET 2003を用いて構築された。ChaTELは、一般的なテキストチャットシステムと同様、サーバ・クライアント構成をとる。参加者は、クライアントシステムを使用する。クライアント側で録音された発言は、すべてサーバにアップロードされる。また、発言を聞く場合は、クライアントからサーバにアクセスしてファイルをダウンロードし、その発言内容を聞く。

図3に、ChaTELのユーザインタフェースを示す。ChaTELユーザは、最初にハンドル名を入力し、ログインする。ログインすると、自分を含め、すでにログインしているメンバのハンドル名が図3右部分の参加メンバー一覧に表示される。なお、いったんログインした後は、キーボードからの文字入力はいっさい必要ない。したがって、自分専用のクライアントマシンがあり、つねに一定のハンドル名を利用できる状況であれば、最初からいっさいキーボードを使用しないでChaTELを使用することができる。

発言を録音する方法は、(1)通常録音、(2)対応発言ID付与録音、(3)相手指定録音の3つがある。

通常録音の場合には、図3の録音関連ボタンの“録

表1 マルチスレッド対話の進行に貢献する表現（テキストチャットと提案システムとの対応）

Table 1 Expressions of contribution to continuing multi-threaded conversations (TextChat vs. Our Proposed System).

指定内容	テキストチャット	提案システム
誰に向けた発言か	>人	対話相手指定情報
どの話題への発言か	>単語	先行関連発言情報
どの発言への発言か	>コピペ	先行関連発言情報

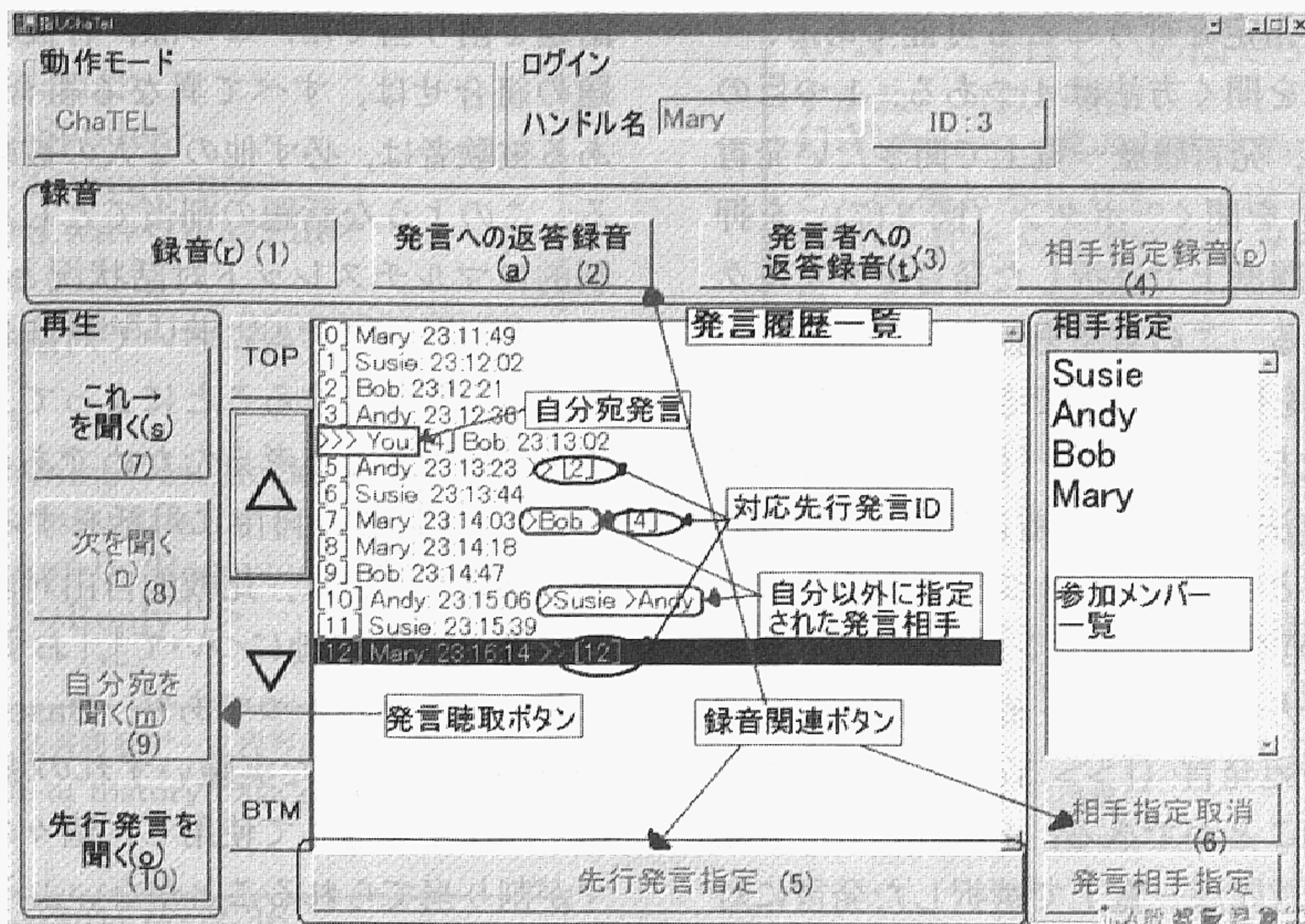


図 3 ChaTEL のユーザインタフェース

Fig. 3 User interface of ChaTEL.

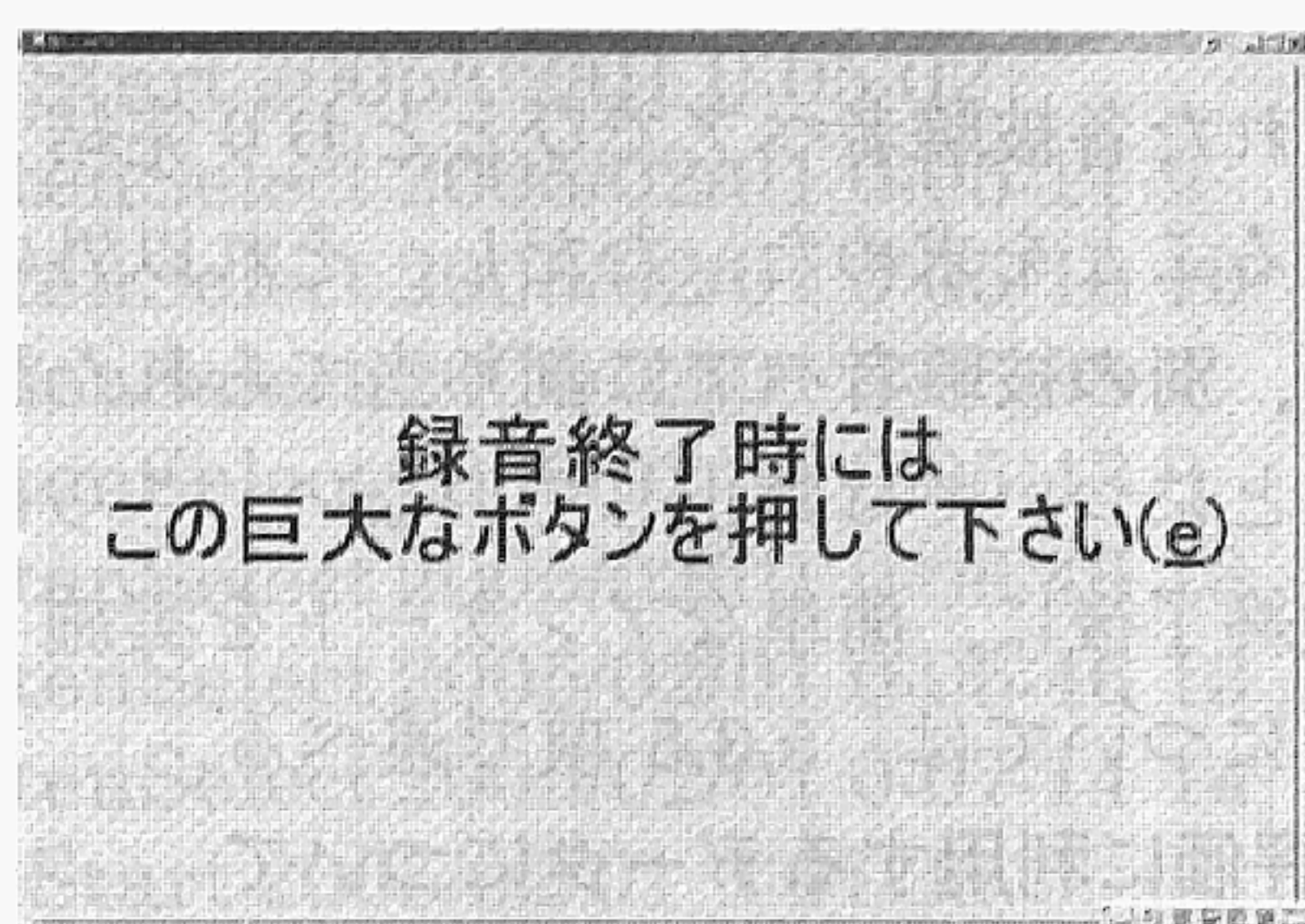


図 4 録音開始後に表示される録音終了ボタン

Fig. 4 A button to stop recording.

音”ボタン (図 3(1)) を押す。すると、図 4 の録音終了ボタンが、図 3 のユーザインタフェース画面をすべて覆い隠す形で画面上に表示される。話したい内容を発言して発言完了時にこのボタンを押すことで発言が完了する。録音終了ボタンをこのようなデザインとした理由は、予備実験において録音を終了しないまま次の操作を行うケースが若干見られたので、確実に録音終了操作を行わせるようにするためである。録音を終了すると、図 4 のボタンが消えて図 3 のユーザインタフェースが再表示され、図 3 中央部分の発言履歴一覧の最後に、新しい発言の発言 ID (サーバが発言データを受信した順に付与する通し番号)、発言者のハンドル名、およびサーバが発言を受信した時刻が発言情報として表示される。

先行発言と関連づけられた発言を行う方法は 2 つある。1 つ目の方法は、先に述べた通常録音を完了させた直後に、発言履歴一覧上で対応づけたい先行発言を

選択し、そのうえで図 3 下部の“先行発言指定”ボタン (図 3(5)) を押す方法である。もう 1 つの方法は、発言履歴一覧上で対応づけたい先行発言をまず選択した後に、“発言への返答録音”ボタン (図 3(2)) を押す方法である。これらの方法で発言を行うと、発言履歴上の発言情報の末尾に「>> [2]」のように対応づけられた先行発言の ID が付与される。

対話相手を指定した発言を行う方法は 3 つある。1 つ目の方法は、通常録音で発言を完了した直後に、参加メンバー一覧から指定したい相手を選択したうえで“発言相手指定”ボタン (図 3(6)) を押す方法である。たとえば、通常録音完了後に、参加メンバー一覧から“Bob”を選択したとする。そうすると、“発言相手指定”ボタン (図 3(6)) が「> Bob」という表示のボタンに変わり、そのボタンを押すことで“Bob”への発言が完了する。2 つ目の方法は、1 つ目の方法の手順とは逆で、先に参加メンバー一覧上で相手選択を行った後に、“相手指定録音”ボタン (図 3(4)) を押す方法である。3 つ目の方法は、発言履歴一覧上で指定したい相手の発言を選択し、その状態で“発言者への返答録音”ボタン (図 3(3)) を押す方法である。これらの方法で発言を行うと、発言履歴上の発言情報の末尾に「> Bob」のように指定された相手のハンドル名が付与される。なお、自分が他のメンバから相手指定を受けた場合には、行頭に「>>> You:」と表示される。

また、以上のそれぞれの方法を組み合わせる「> Bob >> [4]」のように相手指定と対応発言 ID の両方を付与することも可能である。さらに、「> Susie > Andy」

のように複数の相手指定を行うことも可能である。

発言履歴上の発言を聞く方法は4つある。1つ目の方法（通常再生）は、発言履歴一覧上で聞きたい発言を選択し、“これ → を聞く”ボタン（図3(7)）を押すか、あるいは発言履歴上で選択した発言をダブルクリックする方法である。この方法を使うと、選択した発言IDの発言内容を聞くことができる。2つ目の方法（次再生）は、“次を聞く”ボタン（図3(8)）を押す方法である。この方法を使うと直前に聞いた発言IDの次の発言を聞くことができる。3つ目の方法（自分宛再生）は、“自分宛を聞く”ボタン（図3(9)）を押す方法である。この方法を使うと、直前に聞いた発言よりも後にある自分宛発言（「>> You:」が付与されている発言）を聞くことができる。4つ目の方法（先行発言再生）は、発言履歴一覧上で選択した発言に対応先行発言IDが付与されている場合に“先行発言を聞く”ボタン（図3(10)）を押す方法である。この方法を使うと指定されている対応先行発言を聞くことができる。

5. システム評価

5.1 評価実験の概要

4人の大学院生からなる被験者群4組、計16人に対し、以下2つのシステムを用いた実験を行った。

BaseLine ChaTELが提供する機能のうち、発言履歴と、「これを聞く」、「次を聞く」および「録音」ボタンのみを使用可能とし、他の機能のボタンをすべて非表示としたもの。したがって対話相手指定や先行関連発言指定に関する機能は使用できない。

ChaTEL 前章で説明したChaTELそのもの。

被験者は全員、何らかの形でテキストチャットを利用した経験はあるが、計算機上での音声コミュニケーションシステムを利用した経験はない。実験は非対面状況で実施され、被験者は全員、離れた個室で実験システムを利用した。なお、実験時の環境は、ログイン時にログイン名を入力する際にキーボード、マウス操作を行うのみで、実際の対話中にはこれらの操作を行う必要はない。また発言の録音・再生操作のためのボタン操作は、小型のタッチパネル付きモニターを使用して行っている。これは将来的にPDAや携帯電話などの小型化を想定しているためである。また、システムの慣れによる影響を抑えるため、まずBaseLineを使用した後ChaTELの順序で実験を行う被験者群と、まずChaTELを使用した後にBaseLineの順序で実験を行う被験者群とに分けて実験を実施した。

実験では、4つの話題を用意し、各被験者に2つの

話題を割り当てた。この際、各被験者に割り当てた話題の組合せは、すべて異なる組合せとした。つまり、ある被験者は、必ず他の2人の被験者と話すことになる。このような話題の割当てとしたのは、最初から強制的にマルチスレッド対話状況となるように設定し、それぞれのシステムにおいて対話状況がどのように移行するかを調査することによって、本システムの有効性を評価できると考えたためである。これらの話題について、約20分自由に対話をするよう教示した。なお、与えた話題は、比較的対話に近い「行ってみたい場所」、「出身地について」、「おすすめの食べ物」、「今欲しいもの」の4つであり、BaseLineを用いた実験、ChaTELを用いた実験いずれの場合にもこれら4つのテーマを共通して使用し、個々の被験者に同じテーマが割り当てられることがないように割振りを行った。これは、与えるテーマによって対話構造そのものが異なる可能性があるため、BaseLineとChaTELを比較する際に、できる限り余分な変数を作らないための配慮である。また、これらの話題については、話題を割り当てられた各被験者ペアでひととおり完結するまで話を続けることを求めた。ただし、それ以外の話をすることや、別の被験者ペアに割り当てられた話題に参加することについては禁止していない。さらに、すべての実験終了後に、簡単なアンケートを実施した。アンケートについては、5.3.5項で述べる。

5.2 評価に利用するデータについて

5.2.1 取得データ

5.1節で説明した本システムの評価実験では、以下の4種類のデータを取得している。

- (1) 各発言ごとの音声ファイル
- (2) 発言履歴
- (3) メンバリストファイル
- (4) 行動履歴

(1)は、個々のユーザが発言の際に録音を行ったデータであり、1発言につき1ファイルの構成となっている。(2)は、個々の発言イベントについての情報をXML形式で記録したものである。図5に発言履歴データの例を示す。記録内容は、発言ID (<Serial>), 発言者ID (<Sender>), 発言者内での通し発言番号 (<Sequence>), 対応する音声ファイル名 (<Wav-File>), 発言受信時刻 (<DateTime>), 相手指定された参加者のID (<Receiver>), 指定された先行発言ID (<ReplyTo>) などである。(3)は、発言者IDについての情報を記録したものである。記録内容は、発言者ID, ログイン名, 発言総数, 発言再生総数である。(4)は、個々のユーザが本システム上で行ったすべ


```

<Table1>
  <Serial>3</Serial>
  <Sender>0</Sender>
  <Sequence>3</Sequence>
  <WavFile>0-3.wav</WavFile>
  <DateTime>
    2005-01-31T14:51:47.0500000+09:00
  </DateTime>
  <Receiver />
  <IsChecked>>false</IsChecked>
  <IsListened>>false</IsListened>
  <ReceiverNum>0</ReceiverNum>
  <ReplyTo>-1</ReplyTo>
</Table1>

```

図 5 発言履歴データ内容の例

Fig. 5 A sample of history data of utterances.

```

Invoked 2005/02/01 0:02:36
BaseLine 2005/02/01 0:02:42
LoggedIn 2005/02/01 0:03:06kanayon
ListenSelect 2005/02/01 0:03:45 6
ListenSelect 2005/02/01 0:04:55 9
RecStart 2005/02/01 0:05:02
ListenSelect 2005/02/01 0:05:11 9
ListenSelect 2005/02/01 0:05:16 10
ListenSelect 2005/02/01 0:05:38 7
ListenSelect 2005/02/01 0:05:43 8
ListenSelect 2005/02/01 0:06:46 13
ListenSelect 2005/02/01 0:06:46 13
ListenSelect 2005/02/01 0:07:14 14
RecStart 2005/02/01 0:07:35

```

図 6 行動履歴内容の例

Fig. 6 A sample of log data of a user's actions.

```

40: N:今一番ほしいのはやっぱり記憶力で
すかね. (以下略)
41: Y:えっと, 僕のおすすめの食べ物はえー
と, 大根おろしです. (以下略)
42: Y:それたぶんですね, メモるといいです
ね.
43: H:出身地の自慢っていうほどでもない
んですけど, 実は私, 奈良県の生駒市って
いうところの出身地です. (以下略)
44: M:えーと, 俺の出身地イタリアはくい
モンがうまいですよ. ほんま, 一番自慢で
きるのはそこだと思います.
45: H:なんか余計なものを覚えていそうで
それはそれで嫌な気もするんですが.

```

図 7 ChaTEL を用いた実験の対話例 (左から発言 ID, 発言者のハンドル名, 発言内容)

Fig. 7 A sample of utterances with using ChaTEL.

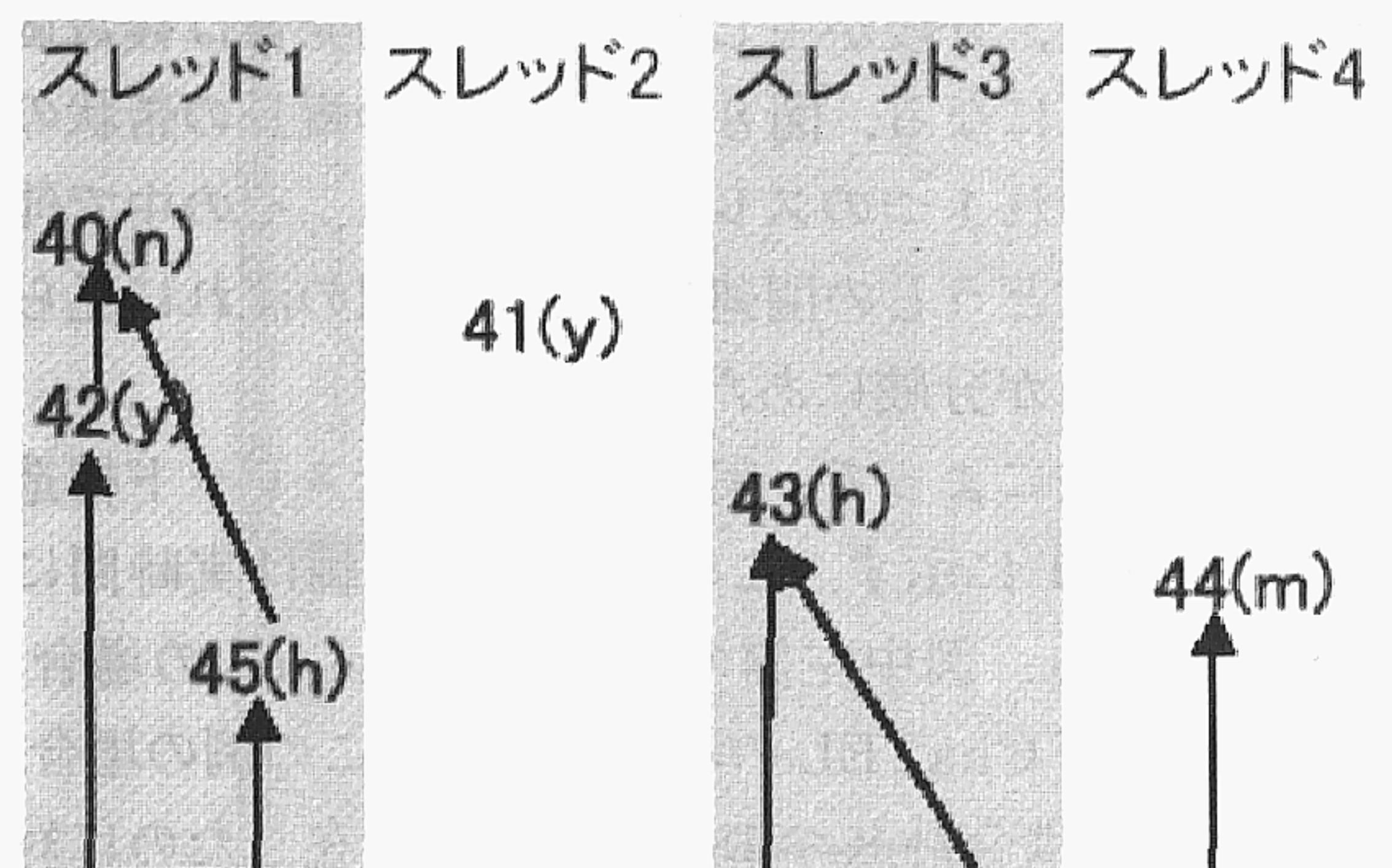


図 8 図 7 の対話例に基づくスレッド構成図 (簡略版) (図中の番号 (英字) は発言番号 (発言者) を示す)

Fig. 8 A sample of thread structure based on the sample utterances in Fig. 7.

ての操作履歴を記録したものである。図 6 に例を示す。記録内容は、本システムを起動した時刻 (Invoked)、選択したシステムの種類 (この例では BaseLine)、ログイン時刻とハンドル名 (LoggedIn)、使用された再生ボタンの種類 (この例では、「これ → を聞く」を示す ListenSelect)、録音開始時刻 (RecStart) などである。

5.2.2 データ処理方法

本システムがマルチスレッド対話を容易にするかを検証するためには、発言内容、スレッド構造に立ち入る必要がある。テキストチャットに関する研究^{9)~11)}で、複数の話題が同時に進行する際には、個々の発言が必ずしも意味的に隣接しないことが示されており、ボイスチャットでも同様のことが予想される。そこで、今回は、5.2.1 項で説明した音声ファイルから人手により発言内容を書きおこし、さらに、音声ファイルと発言履歴データ、行動履歴データの対応をとり、対話

開始から対話完了までのすべての発言行動を可視化できるようにデータの処理を行った。さらに、個々の発言がどの先行発言と直接に意味的なつながりを持つかを判定する作業を行った。発言者によって先行発言 ID が付与されている場合には、付与されている ID の発言を直接つながりのある発言と見なし、そうでない場合には、発言内容からどの発言とつながりを持つかを判定した。さらに、個々の発言間のつながりを判定した後、個々の発言がどのようにスレッドを構成しているかを発言完了時刻との対応づけをあわせて判定し、スレッド構成図を作成した。

スレッド構成図の作り方について、図 7 の実験中に取得した音声データ、発言履歴データをもとに人手によって書き起こした対話例と、この対話例から得られるスレッド構成図 (図 8) をもとに、さらに詳細に説明する。なお、実際のスレッド構成図では、縦軸は実時間 (秒単位) に対応しているが、図 8 は、説明の

ため、実時間との対応はつけずに発言順序のみを考慮した簡略版となっている。本論文では、「スレッド」と「分岐」に着目して分析を行う。ここで「スレッド」とは、ある新たな話題の開始となる「始端の発言」（図7の対話例における発言番号40, 41, 43, 44が始端発言に相当する）に始まり、その話題に関する最後の発言となる「終端の発言」に至る、すべての発言群のことをいう。一方「分岐」とは、先行発言（質問、勧誘、提案などの何らの働きかけのある発言）に対して、複数の後続発言（返答、受諾などの働きかけに対する応答にあたる発言）に相当する発言が起こる場合にできる枝分かれのことである。図7の対話例でいうと、発言番号42, 45に相当する発言である。この2つの発言はそれぞれ発言番号40に対する後続発言となるため、分岐と見なしている。分岐は1つのスレッド内で発生するので、分岐が生じた場合、終端発言は複数個存在することになる。図8では、灰色の縦長の帯状の領域がそれぞれ1つのスレッドに対応し、その中で伸びている発言どうしの関連性を示す直線の流れに見られる枝分かれが分岐にあたる。

以上の方法によって、実際に作成したスレッド構造図の例を図9に示す。この図では、縦軸は実時間に相当している。図中左側の(a)がBaseLineの場合、右側の(b)がChaTELの場合である。この例の場合、BaseLineでは合計8つのスレッドが存在したのに対し、ChaTELでは6つのスレッドが存在した。図中、黒枠で囲われている箇所が、複数のスレッドが同時存在している箇所である。

5.3 対話のマルチスレッド化に対する有効性の検証

5.3.1 全般的発言状況とChaTELにおける録音手法

BaseLineおよびChaTELを用いた実験すべての総対話時間（秒）、総発言数と、1秒あたりの発言数を表2に示す。表2より、システムの差異による発言数などの違いは見られなかった。また、ChaTELを用いた場合、4章で説明したように、録音方法は、1) 通常録音、2) 対応発言ID付与録音、3) 相手指定録音の3通り存在する。これら3種類の録音方法が、それぞれの程度の頻度で使用されたかを表3に示す。表3より、ChaTELの全発言数337のうち86%にあたる290発言が本システムで新たに提供した録音方法によるものであることが分かる。この結果は、本システムが提供する発言間の関連づけ機能を被験者が積極的に利用していたことを示している。

5.3.2 全般的発言再生状況

BaseLineおよびChaTELを用いた実験すべての再

表2 実験での総対話時間、総発言数、および1秒あたりの発言数
Table 2 Total conversation time, total number of utterances and average number utterances per second.

	BaseLine	ChaTEL
総対話時間	5,528 秒	5,689 秒
総発言数	339	337
1秒あたりの発言数	0.06	0.06

表3 ChaTEL使用時の3種類の録音方法の使用頻度
Table 3 Frequencies of the three recording ways when ChaTEL is used.

通常録音	47
対応発言ID付与録音	265
相手人指定録音	25
合計	337

表4 BaseLineおよびChaTEL使用時の4種類の再生方法の使用頻度
Table 4 Frequencies of the four listening ways when BaseLine and ChaTEL are used.

	BaseLine	ChaTEL
通常再生	595	1,002
次再生	801	737
自分宛再生	—	1
先行発言再生	—	16
総再生数	1,396	1,756

生状況を表4に示す。4章で説明したように、再生方法は、通常再生、次再生、自分宛再生、および先行発言再生の4種類がある。ただしBaseLineについては、自分宛再生と先行発言再生機能は提供していない。表4より、BaseLineよりもChaTELのほうが再生数が増加していることが分かる。さらに、個々の再生方法の頻度について検討すると、ChaTELでのみ提供されている自分宛再生と先行発言再生はほとんど使われていない。また、BaseLineでは次再生の方が多用されているのに対し、ChaTELでは通常再生の方が多用されている。

5.3.3 時間単位での平均スレッド数および分岐数

作成したスレッド構造図をもとに、BaseLineと本システムで、1秒あたりの平均スレッド数と平均分岐数を求めた。すなわち、同時にいくつのスレッドがあるか、および発言間を連結させた直線が同時に何本存在するかを1秒ごとに求め、すべてのデータについてのスレッド数総和、分岐数総和をデータ分析対象となった総時間数（秒）で割ることで、平均スレッド数と平均分岐数を求めた。結果を図10に示す。

図10に示すとおり、平均スレッド数、平均分岐数どちらについても、BaseLineよりもChaTELの方が有意に値が大きい。この結果から、ChaTELの方が複

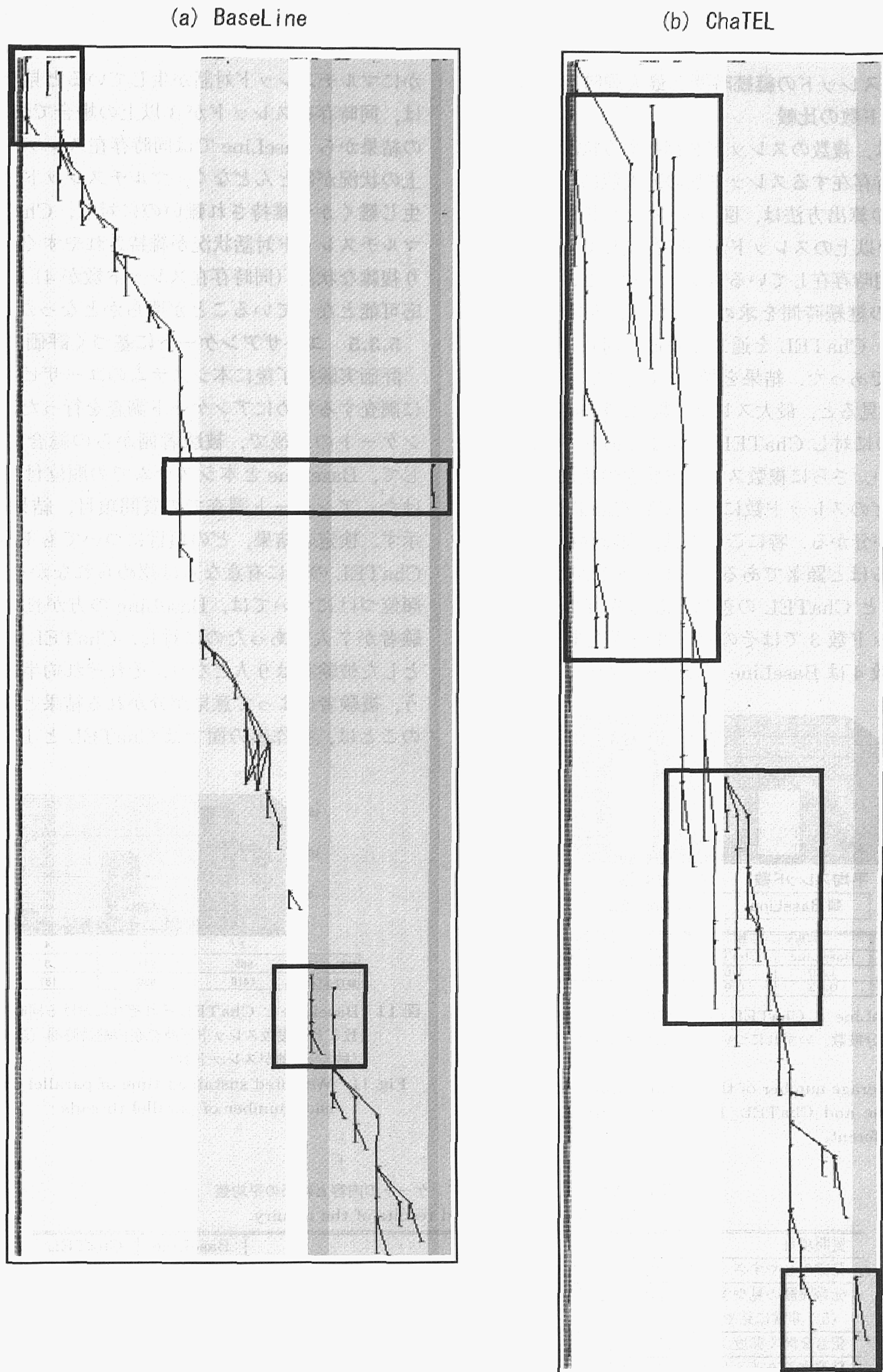


図9 スレッド構造の例 (図中枠内は、実際にマルチスレッド起こっている箇所を示す)

Fig. 9 A sample of structure of threads.

数スレッドが同時に存在する状況が多く、しかも個々のスレッドに関しても枝分かれが多い複雑な構造の対話になっていることが分かる。

5.3.4 スレッドの継続時間と最大同時存在スレッド数の比較

本項では、複数のスレッドが存在する状況の継続時間と、同時存在するスレッド数の最大値について比較する。この算出方法は、図9のスレッド構造図をもとに、2つ以上のスレッドが同時存在している部分について、同時存在しているスレッド数ごとに分けて秒単位でその継続時間を求めた。なお、今回の実験でBaseLine, ChaTEL を通じての最大同時存在スレッド数は4であった。結果を図11に示す。

図11を見ると、最大スレッド数は、BaseLineでは3であるのに対しChaTELでは4であり、ChaTELの方が多。さらに複数スレッド状態の継続時間についても、どのスレッド数についてもChaTELの方が長いことが分かる。特にこの差は、同時存在スレッド数が増えるほど顕著である。スレッド数が2の場合、BaseLineとChaTELの差は1.5倍程度の差であるが、スレッド数3ではその差が3倍程度まで広がり、スレッド数4はBaseLineでは存在しない。

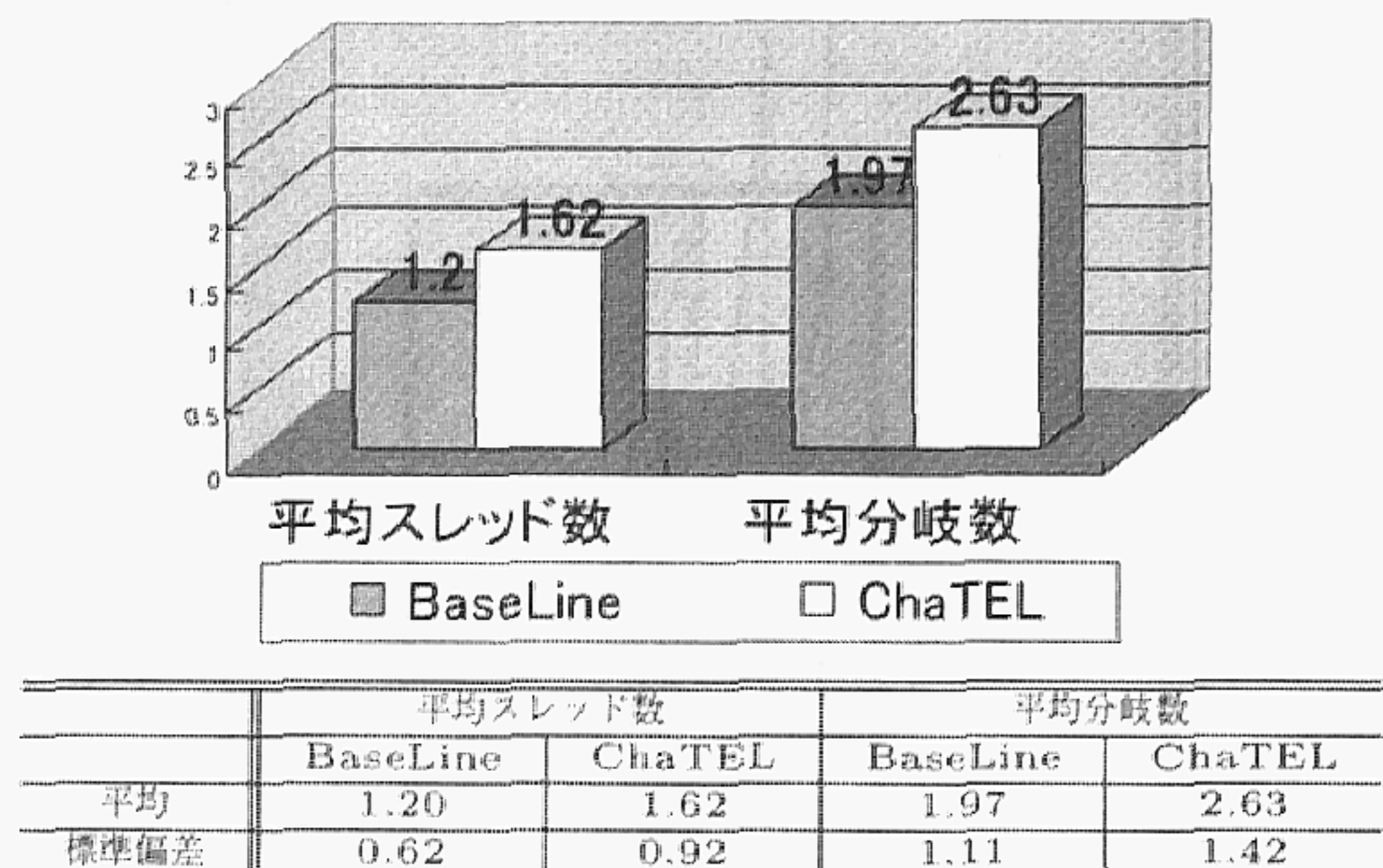


図10 BaseLineとChaTELの秒単位の平均スレッド数および平均分岐数。いずれについても0.1%水準で有意差が認められた

Fig. 10 Average number of threads and branches for BaseLine and ChaTEL. Both results are significantly different.

話者数が4人なので、同時存在スレッド数が2の場合は、必ずしもマルチスレッド対話状況ではなく、単なるグループの分割と見なしうる。したがって、明らかにマルチスレッド対話が生じていると見なしうるのは、同時存在スレッド数が3以上の場合である。図11の結果からBaseLineでは同時存在スレッド数が3以上の状況がほとんどなく、マルチスレッド対話状況は生じ難くかつ維持され難いのに対し、ChaTELではマルチスレッド対話状況が維持されやすく、しかもより複雑な状況(同時存在スレッド数が4)にまでも対応可能となっていることが明らかとなった。

5.3.5 ユーザアンケートに基づく評価

評価実験終了後に本システムのユーザビリティを主に調査するためにアンケート調査を行った。また、アンケートの最後で、被験者側からの総合的な評価として、BaseLineと本システムでの順位付け項目を設けた。アンケート調査での質問項目、結果を表5に示す。検定の結果、どの項目についてもBaseLineとChaTELの間に有意な差は認められなかった。また、順位づけについては、BaseLineの方が良いとした被験者が7人であったのに対し、ChaTELの方が良いとした被験者は9人となり、それぞれ約半数ずつという、被験者によって意見が分かれる結果となった。このことは、操作感の面ではChaTELとBaseLineに

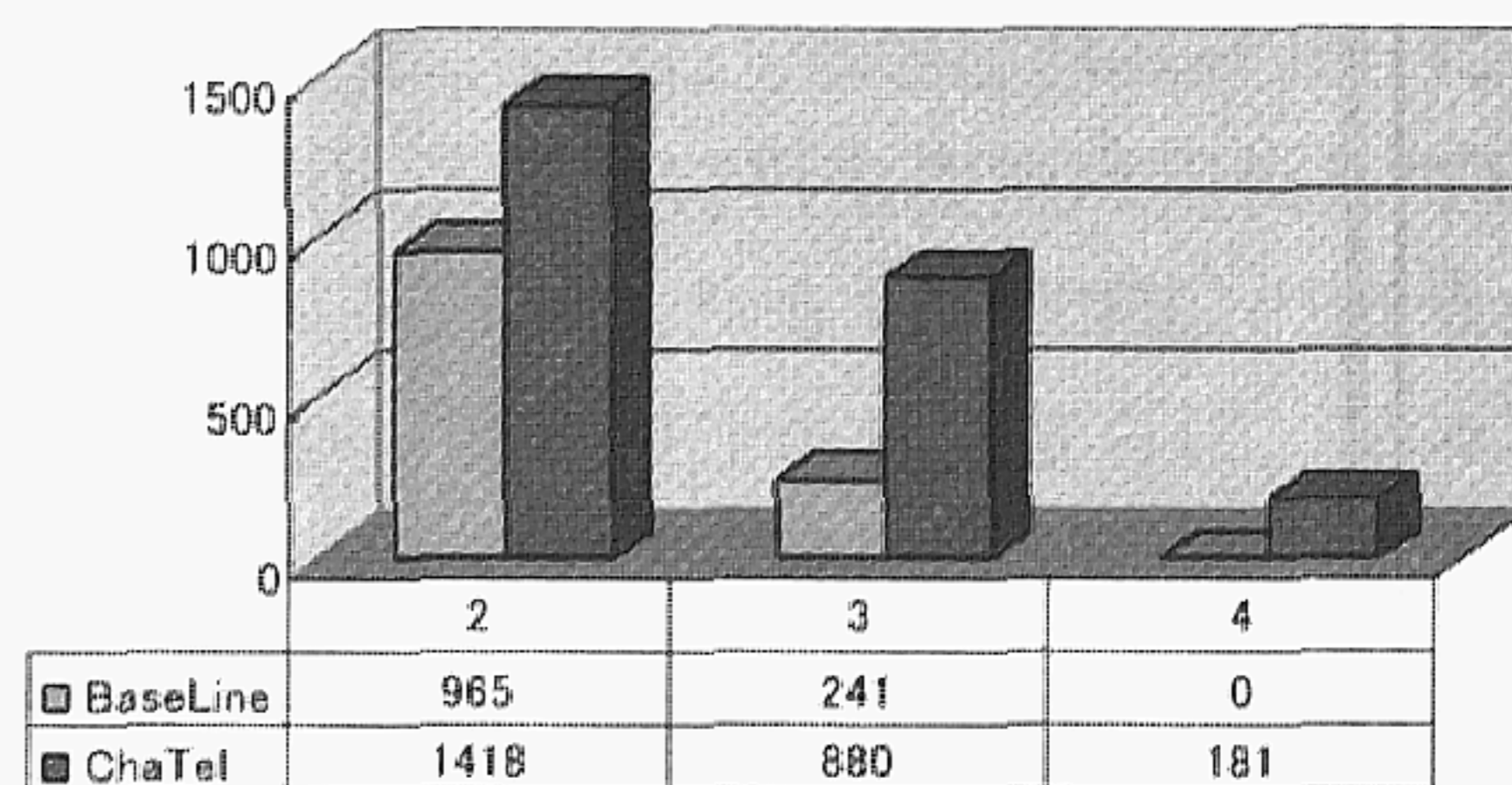


図11 BaseLineとChaTELそれぞれにおける同時存在スレッド数ごとの複数スレッド同時存在の継続時間(縦軸が計測時間(秒)、横軸がスレッド数)

Fig. 11 Weighted sustained time of parallel threads for each number of parallel threads.

表5 実験終了後に実施したアンケートの内容と結果の平均値
Table 5 Questions and results of the inquiry.

質問項目	BaseLine	ChaTEL
発言のしやすさ(操作性の点で)(5:非常に簡単)	3.4	3.1
発言履歴の見やすさ(発言の再生,録音の参考になるか否かという点で)(5:非常に見やすい)	3.3	3.3
発言を聞く頻度(5:ほぼ全部聞いた)	5.0	4.9
発言するタイミングのとりやすさ(5:非常にとりやすい)	2.7	3.2
最初に与えられた話題以外の話をする事があったか(5:高い頻度であった)	3.5	3.3
システムそのものの使いやすさ(5:非常に使いやすい)	3.6	3.3
システムそのもののおもしろさ(5:非常に面白い)	3.5	3.7

大きな違いはないことを示している。BaseLine に比べて ChaTEL は多くの機能を有し、その分ユーザインタフェースの構造や操作方法は複雑になっている。このため、ユーザビリティの面でははより低い評価となることを予想していたが、実際には BaseLine と比べて遜色ない評価を得た。つまり、ChaTEL で追加された機能によって使用感が悪化することはなかったといえる。

6. 議論：本システムのマルチスレッド化への貢献と課題

5章で示したように、本論文で提案した「対話相手指定情報」と「先行関連発言情報」を付与する機能を持つ音声コミュニケーションシステム“ChaTEL”は、これらの機能を持たない単純な音声コミュニケーションシステムよりも、対話のマルチスレッド化を生じさせやすく、しかも、マルチスレッド状態を継続させやすいということが明らかになった。また、個々のスレッド内での分岐の発生が多いこともあわせて示された。分岐の発生は、対話状況が話者交替規則から外れた状況になっている結果と考えられる。したがって、この結果も、筆者らが ChaTEL で目指した効果が得られていることを示すものであるということができよう。以上の結果から、本論文で提案した手法によって、音声によるリアルタイム・マルチスレッド対話を実現可能であることが示された。

しかしながら、5.3.5 項に示すとおり、被験者に対するアンケートでは、いずれの項目についても ChaTEL と BaseLine の間に有意差が認められておらず、主観的には ChaTEL がとりたてて優れているとは感じられていない。この原因は2つ考えられる。

第1は、被験者の音声マルチスレッド対話への適応に関する問題である。テキストチャットの場合でも、慣れた利用者は積極的に対話のマルチスレッド化を行うが、逆に初心者の場合はなかなかマルチスレッド対話を実行しない(できない)といわれている^{1),12)}。これは、はじめに述べたように、対話は話者交替規則に沿って行うべきであるという「常識」が身につけているため、そもそも対話を積極的にマルチスレッド化させようという発想が出にくく、自然発生的にマルチスレッド状態が生じても、その状態を回避して通常の話者交替規則にのっとった「単一スレッド対話」に戻そうという意識が働くことによると思われる。今回の実験でもおそらく同様の状態にあったと考えられる。「音声によるマルチスレッド対話」を経験したことがある者は皆無であり、今回の被験者はほぼ「初心者」に近

い状態であったと考えるべきである。このため、音声対話を積極的にマルチスレッド化させようとする意識は被験者にはなかったであろう。つまり、現時点では「音声によるマルチスレッド対話が容易」であることがメリットとして十分に認識されていないと考えられる。しかしながら、これは結局単なる「慣れ」の問題にすぎない。テキストチャットにおいて利用者が自発的にマルチスレッド対話を行うようになったという事実を考慮すれば、ChaTEL のようなコミュニケーションシステムが普及することによって、音声対話においてもやがてマルチスレッド対話を行うことは当たり前のことになり、その段階においてマルチスレッド対話のメリットが理解されるであろう。

第2は、ChaTEL のユーザインタフェースの複雑さの問題である。4章で本システムの操作方法について説明したように、現在のインタフェースデザインでは、発言録音および発言再生のどちらの機能についても複数の多様な操作手段が提供されている。評価実験の際には、すべての可能な操作方法を説明し、実験前に本システムに慣れるための時間を与えることで、ある程度被験者が操作に慣れることができるようにしている。しかし、それでも被験者アンケートでの感想記述項目中に「表示がごちゃごちゃしている」、「BaseLine のほうがボタンが少なくて使いやすかった」などの記述が見られた。前述のとおり、BaseLine との比較では有意差は見られない結果となっているが、操作の煩雑さ、インタフェースの複雑さが、マルチスレッド対話が容易というメリットによる評価の向上を妨げている可能性も考えられる。この問題については、利用者の行動履歴のさらに詳細な分析に基づき、使いやすいボタン配置などを今後検討したい。

また、発言時には対話相手指定や先行発言指定機能が多用されるにもかかわらず、聴取時にはこれらの関連づけを利用した聴取手段はほとんど使用されないという非対称性も興味深い現象である。これは、単にユーザインタフェースデザインの問題かもしれないが、一方で人の対話行動特性に根ざした現象である可能性もある。これについては、聴取行動も含めて、対話行動全体のより詳細な分析を実施して、今後明らかにしていきたい。

本論文では、ChaTEL によってマルチスレッド対話を実現できることを示したが、話者交替規則に従う通常のシングルスレッド対話よりもマルチスレッド対話の方が良いのか、スレッド数がいくつまでであれば対応可能か、スレッド数は多いほど良いのか、などについてはまだ解明できていない。現段階でこれらの

問いに明確に答えるための材料はそろっていないが、ここで注意すべきことは、ChaTELは対話のマルチスレッド化を強制するシステムではないので、被験者はChaTELを用いてBaseLineと同じ形態の対話を行うことも可能であったということである。つまり、音声でのリアルタイム・マルチスレッド対話が人間の能力的に無理なものであったとすれば、あるいは話者交替規則に従ったシングルスレッド対話が人間にとって最も好ましい音声対話形態であったとすれば、たとえChaTELを用いたとしてもBaseLineと同程度のスレッド数とマルチスレッド維持時間の結果とならずである。しかし実際には、ChaTELを用いた場合に、聴取回数が増加して負荷が高まっているように思われるにもかかわらず、それによって発話数が減少することはなく、しかも明らかに対話のマルチスレッド化が生じている。このことは、音声によるリアルタイム対話において、たとえ聴取頻度が少なくとも、必ずしもシングルスレッド対話が最適な対話形態ではなく、トータルとしてマルチスレッド対話の方が望ましい形態である可能性を示唆している。したがって、この程度の同時対応スレッド数の範囲ならば、過負荷による知識生産性の低下は生じないものと思われる。

ただし、人間の認知能力の限界を考慮すれば、直感としてスレッド数が多いほど良いということはないと推測される。評価実験において、4人の被験者での最大スレッド数が4つということから考えて、同時にこなすスレッド数の限界、最適数があることが推測される。この最適なスレッド数、スレッド数の限界についても、さらなる実験・分析を進めることで明らかにしたい。

最後に、マルチスレッド対話が有効に活用されうる場面について検討する。本システムは、1章で述べたように、ブレインストーミングなどのできる限り数多くのアイデアを産出するタイプの議論や、雑談のような思いついたら即発話を行うタイプの対話に適していると思われる。実際、評価実験では、雑談的な比較的自由な対話においてマルチスレッド対話が生じることが示された。またブレインストーミングでは、口数の多い人ばかりが発言することの問題が従来より指摘されている。その対策としてブレインライティング¹³⁾などの手法が考案されているが、マルチスレッド対話によっていつでも誰でも発言できるようになれば、口数の少ない人の発言も漏れなく引き出すことが可能となる。一方、特定の話題について収束的な議論を行うような場合には、マルチスレッド対話は議論の収束の妨げとなる可能性があり、また発言の再生に時間がか

かるために、議論を長引かせる可能性がある。反面、長時間の議論では「過去に何を話していたか自体を忘れかねない」という問題があるが、本システムによって過去の任意の発言を随時参照可能となれば、この問題を回避することができると考えられる。

7. おわりに

本論文では、筆者ら自身によるこれまでのテキストチャットに関する分析から得られた知見に基づき、各発言に対して「対話相手指定情報」と「先行関連発話情報」を付与する機能を提供することにより、リアルタイム・マルチスレッド音声対話を可能とする音声コミュニケーションシステムChaTELを提案した。音声対話をマルチスレッド化し、これをモバイル/ユビキタス環境と融合することによって、様々な場面で知識生産効率の大幅な向上が期待できる。対話相手指定と先行関連発話指定機能を除いたBaseLineシステムとChaTELとの比較実験を実施し、得られた対話データをもとに音声対話のマルチスレッド化における有効性について分析を行った。その結果、1秒あたりの平均スレッド数、平均分岐数ともに、ChaTELの方がBaseLineよりも多くなることが示された。また、複数のスレッドが同時存在する場合の継続時間と最大スレッド数についても、ChaTELを用いた場合、BaseLineよりも、より多くのスレッドがより長く同時存在することが示された。以上の結果から、本論文で提案した手法によって、音声対話をマルチスレッド化することが可能であることが示された。

今後は、本論文で考案したマルチスレッド化支援手法以外の手法の考案と実装を進めるとともに、ユーザの行動履歴のさらなる分析に基づき、ユーザインタフェースの改良を進め、より使いやすい音声コミュニケーションシステムを実現したい。さらに、発言履歴を視覚的に提示せず、すべて聴覚のみで対応可能な、高いモバイル性を持つシステムの実現についても検討したい。

謝辞 本研究の一部は、文部科学省知的クラスター創成事業石川ハイテク・センシング・クラスターにおける「アウェアホーム実現のためのアウェア技術の開発研究」プロジェクトの一環として行われたものである。多忙な中、実験に協力してくださった被験者の皆さんに御礼申し上げます。

参考文献

- 1) Ogura, K. and Nishimoto, K.: Is a Face-to-Face Conversation Model Applicable to

- Chat Conversations?, *Proc. 18th PRICAI2004 Workshop on "Language Sense on Computer"*, pp.26-31 (2004).
- 2) 折原良平: 発散的思考支援ツールの研究開発動向, *人工知能学会誌*, Vol.8, No.5, pp.560-567 (1993).
- 3) MSN メッセンジャー.
<http://messenger.msn.co.jp/>
- 4) Yahoo メッセンジャー.
<http://messenger.yahoo.co.jp/>
- 5) Skype. <http://www.skype.com/intl/ja/>
- 6) Berc, L., Gajewska, H. and Manasse, M.: Pssst: Side Conversations in the Argo Telecollaboration System, *Proc. 8th ACM UIST*, pp.155-156 (1995).
- 7) 西本卓也, 北脇裕康, 高木治夫: 非同期型音声会議システム VoiceCafe, *情報技術レターズ (FIT2003 講演論文集)*, LK-005, pp.273-274 (2003).
- 8) Aoki, P.M., Romaine, M., Szymanski, M.H., Thornton, J.D., Wilson, D. and Woodruff, A.: The Mad Hatter's Cocktail Party: A Social Mobile Audio Space Supporting Multiple Conversations, *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems*, pp.425-432 (2003).
- 9) 小倉加奈代, 石崎雅人: チャット対話の話題推移に関する特徴分析, *人工知能学会研究会資料: 言語・音声理解と対話処理研究会*, SIG-SLUD-A202-3, pp.13-19 (2002).
- 10) 細馬宏道: チャットは何を前提としているか—チャットの時間的構造と音声会話の構造, *bit 別冊: 身体性とコンピュータ*, pp.338-349, 共立出版 (2002).
- 11) 水上悦雄, 右田正夫: チャットの会話の秩序—インターバル解析による会話構造の研究, *認知科学*, Vol.9, pp.77-88 (2002).
- 12) 細馬宏通: 相互行為とメディア—チャットという「会話」はどのような時空間構造を持つか, *相互行為の社会心理学*, pp.179-197, 北樹出版 (2002).

- 13) 高橋誠: 創造技法—主要 88 技法—ブレインライティング (BW) 法, *創造力事典*, pp.294-296, 日科技連出版社 (2002).

(平成 17 年 6 月 1 日受付)

(平成 17 年 11 月 1 日採録)



小倉加奈代

1999 年東北学院大学教養学部言語科学専攻卒業。2001 年北陸先端科学技術大学院大学知識科学研究科博士前期課程修了。現在, 同大学院博士後期課程在学中。CMC (特に

テキストチャット) における対話行動プロセスと対話構造に興味がある。人工知能学会, 認知科学会各会員。



西本 一志 (正会員)

1987 年京都大学大学院工学研究科機械工学専攻博士前期課程修了。同年松下電器産業 (株) 入社。1992 年 (株) ATR 通信システム研究所

出向。1995 年 (株) ATR 知能映像通信研究所客員研究員。1999 年より北陸先端科学技術大学院大学助教授。2000~2003 年科学技術振興事業団さきがけ研究 21「情報と知」領域研究員兼任。2001 年より (株) ATR メディア情報科学研究所非常勤客員研究員兼任。1996 年度人工知能学会研究奨励賞, 1997 年度 DiCoMo シンポジウムベストプレゼンテーション賞, 1999 年度情報処理学会坂井記念特別賞, 1999 年度人工知能学会論文賞, インタラクシオン 2004 ベストインタラクティブ発表賞, ACM Multimedia 2004 Best Paper Award 各受賞。IEEE computer society, ACM, 人工知能学会, ヒューマンインタフェース学会各会員。博士 (工学)。