

マルチスレッド指向音声対話システム ChaTEL を用いた 非対面対話の発言・聴取行動分析

小倉 加奈代† 西本 一志‡

{k-ogura, knishi}@jaist.ac.jp

† 北陸先端科学技術大学院大学知識科学研究科

‡ 北陸先端科学技術大学院大学知識科学教育研究センター

日常の音声対話は、対話参加者全員が単一の話題を一定時間共有し、話者交替しながら対話を進めなければならない、効率が良いとは言い難い。そこで我々は、話したい時に話したいことを話したい相手と話せる、「リアルタイム・マルチスレッド音声対話」を可能とするコミュニケーションシステム ChaTEL を提案した。ChaTEL の特徴は、マルチスレッド対話を支援するための相手指定／対応先行発言指定機能を有する点であり、これらの機能を持たないシステムとの比較により、ChaTEL を用いた場合にマルチスレッド化が顕著かつ継続しやすいことをすでに示した。しかしながら、これらの機能が実際の対話の中でどのように使用され、どのようにマルチスレッド化に寄与しているのかは明らかではなかった。そこで本論文では、発言行動分析と聴取行動分析を行い、ChaTEL を用いた場合になぜマルチスレッド化が生じやすいかを検討する。

Analyses of Speaking and Listening Behaviors in Non Face-to-face Multi-threaded Communications using ChaTEL

Kanayo Ogura † Kazushi Nishimoto ‡

† Japan Advanced Institute Science and Technology, School of Knowledge Science

‡ Japan Advanced Institute Science and Technology, Center for Knowledge Science

Everyday voice conversations require people to obey the turn-taking rule and to keep a single topic thread; therefore, it is not an effective way to communicate. Hence, we proposed “ChaTEL,” which is a voice communication system to facilitate real-time multi-threaded voice communications. ChaTEL has two functions to support the multi-threaded communications: A function to indicate who he/she talks to and a function to indicate which utterance he/she responds to. Comparing ChaTEL with a baseline system that does not have these functions, we have already shown that the multi-threaded conversations occur more frequently than that with the baseline system. However, it has not been clear yet how the functions were utilized and contributed to the multithreaded-conversations. This paper discusses why ChaTEL can facilitate the multi-threaded conversations based on analyses of users’ speaking behaviors and listening behaviors.

1. はじめに

従来の CMC (Computer-Mediated Communication) システムは、電子掲示板、チャットなどのテキストベースのコミュニケーションを扱うものが中心であった。しかし、近年のインターネットの普及により、IP 電話や Skype などの音声コミュニケーションシステムも手軽に使用できる状況になってきた。さらに、携帯電話においても、PTT (Push-To-Talk) 技術を基盤としたサービスが開始されるなど、新しい音声コミュニケーションシステムが急速に広まりつつある。しかし、これ

らは基本的に伝送網をインターネットに置き換えたものに過ぎず、そこでの対話スタイルは、従来の対面音声対話におけるスタイルと差異は無い。

従来の音声対話には、2つの制約が存在する。第1の制約は、基本的に話者は常に1人だけであり、他はすべて聞き手とならねばならず、現在の話し手が話し終えて発話権を手放せる状態にあると認識可能になるまでは、他者が発言開始することはできないという制約である(話者交替の制約)。第2の制約は、ある話題に関する対話が進行しているとき(このような、あるひとつの話題に関する一連の発言列のことを「スレッド」とい

う), その話題に関する発言のやり取りが一段落したと認識できる状態(すなわち, スレッドが収束した状態)になるまでは, 対話参加者全員が協力してそのスレッドを維持しなければならないという制約である(単一スレッドの制約). これらの制約のために, 通常の音声対話では時間あたりの発言密度・話題密度がともに非常に薄く, 結果として情報の交換効率が低いものとなっている.

このように, 従来の音声対話は, 話し手が話したい内容がある際に, もれなく伝達するという点で, 効率がよいとは言いがたい. しかしながら, 我々が極めて慣れ親しんだコミュニケーション形態であるためか, これらの問題を解決し, 効率的な音声コミュニケーションを実現しようとする取り組みは, 著者らの知る限りでは存在しない. そこで我々は, 音声による新しいコミュニケーションスタイルを模索する試みとして, 「マルチスレッド対話」を可能とする音声コミュニケーションシステム“ChaTEL”を試作した[1]. 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するような対話のことである¹. つまり, マルチスレッド対話では上述の2つの制約が無く, 誰もが話したい相手と話したいことを話したいときに話すことが可能となる. この結果, 発言と話題の密度が大幅に濃くなり, 情報交換効率が向上する. これは会議などの効率化に寄与することに加え, 新たな娯楽的音声コミュニケーションスタイルを産み出す可能性もあると思われる.

ChaTEL を用いた被験者実験により, 実際に音声によるマルチスレッド対話を実現可能であることを, 我々はすでに示している[1]. しかしながら, 実際の対話の中で ChaTEL に付与した機能がどのように使われるのか, 各機能がどのように対話のマルチスレッド化に貢献しているのかについては, まだ明らかになっていない. そこで本稿では, ChaTEL を用いた音声対話におけるユーザの発言行動および聴取行動を分析し, マルチスレッド対話において人はどのような行動を取るのか, 何が対話のマルチスレッド化を可能としているのかについて検討する. これらは, 実用的な音声マルチスレッド対話システムを構築するために不可欠な知見であるといえる. さらに, 聴取行動の分析については, 対面対話では, どの発言をどのタイミングで聞いたかということについては分析困難であり, その点で, 人間の会話行動分析の新しい取り組みにつながるといえる.

¹ 立食パーティ会場などで見られる, 単一スレッドが複数並行している状態とは異なることに注意されたい.

以下, 2 章では, 関連する研究や実用化事例について概観する. 3 章では, 構築した音声コミュニケーションシステム ChaTEL の概要を説明する. 4 章では, ChaTEL を用いた実験概要について, 5 章では実験結果について述べる. 6 章では, 実験結果についてのまとめとその考察を行う.

2. 関連研究・事例

まず, 既存の音声コミュニケーションシステムでマルチスレッド対話に近い対話状況が生じる可能性のあるシステムについて概観する.

Pssst[2]は, 遠隔ビデオ会議中に, 並行して特定メンバー同士だけで「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システムである. しかし, 「ささやく」は特定のメンバーだけで交わされ, 他の参加者はそこでどのような対話がなされているか知ることも参加することもできない. また, 逆に「ささやく」に参加している人にとっては, 「ささやく」だけがその時点での参加スレッドとなっている.

VoiceCafe[3]は, 音声で入力された発言を音声認識システムによって文字化する. 次に, その発言は, 直前に発言者が聴取した先行発言に対する応答であるという前提に基づき, その発言を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し, 提示する. したがってこのシステムでは, 複数のスレッドが同時に存在することは可能であるが, 個々のスレッドを異なる枝として提示するため, あるスレッドに参加している状態では, 他のスレッドの対話進行状況を把握することが難しくなる.

“The Mad Hatter's Cocktail Party”[4]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, 誰が同一対話グループに属する話者かを自動判定し, 同一対話グループ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同時的音声会話環境である. しかし, ひとつの対話グループ内での対話は単一スレッドであり, ある話者が複数の対話グループに同時に属することは原理的に無理である.

以上のように, 既存の音声コミュニケーションシステムに関する研究は, いずれも複数の単一スレッドの並行状態を支援するものであり, マルチスレッド対話を実現しようとしているものは見当たらない.

次に, すでに実用化されている音声コミュニケーションサービスについて概観する. 電話やトランシーバー等での対話形態は, やはり通常の対話の形態と同一であり, マルチスレッド対話は実現できない. IP電話や

Skype なども同様である。ボイスチャットも、話者間で発話音声を蓄積せずリアルタイムにやりとりするものであり、接続の手順などが多少異なる以外、用法としては IP 電話などとほとんど差が無い。

ボイスメールは、電子メールと同様の通信形態を取る蓄積型非同期音声コミュニケーションシステムである。これを用いれば、ダウンロードした音声データは何度でも再生し、再聴取することができるし、どのメッセージに対しても随時応答することができる。この意味で対話形態は通常の対話とは異なったものとなり、マルチスレッド化の可能性を持つ。ただし、リアルタイムにコミュニケーションすることはできない。

ボイスメモも、蓄積型の非同期的音声コミュニケーションシステムであり、しかも本来の目的は「対話」ではなく、一方向的に単一のメッセージを送りつけることが主たる用法である。マイクロソフト社の MSN メッセンジャー Version 7.5 に追加されたボイスメモ機能は、テキストでのメッセージのやり取りの中に、音声によるメッセージ(ボイスメモ)を織り込むことを可能としている。このボイスメモは、随時再生し再聴取ことができ、どのボイスメモに対しても随時応答できるため、マルチスレッド化の可能性を持つ。ただし、ボイスメモは、届いた瞬間に自動的に再生される仕様となっているため、多数の参加者でボイスメモだけをやり取りする形態のコミュニケーションを行うことはほぼ不可能である。Yahoo メッセンジャーにもボイスメッセージ機能があるが、ボイスメッセージはある瞬間には一人しか送ることができない。

Push-to-talk(PTT)は、携帯電話やPDAをプラットフォームとする、トランシーバーと非常に類似した音声コミュニケーションシステムである。米国での普及が著しく、欧州でも急速に普及しつつある。さらに日本でも、2005 年秋より NTT DoCoMo[5]と au[6]がこのサービスを開始した。PTT では、相手との接続は、通信可能な相手を選んでボタンを押すだけであり、ボタンを押している間に用件を話す。これにより、音声メッセージがリアルタイムに対話相手に伝えられる。グループに対してメッセージを送ることも可能である。ただし、一人が話している間(ボタンを押している間)は、他のユーザは話すことができない。

以上のように、近年新しい音声コミュニケーションシステムが次々と実用化されつつあるが、いずれについてもマルチスレッド化などの、対話スタイルの変革をもたらすものではない。

3. ChaTEL の概要

3.1 マルチスレッド化の実現手法

マルチスレッド対話を行なうには、一度に複数の発言を聞き分け、記憶し理解する必要がある。しかし、これは人間の認知能力の範囲内では困難なため、通常の音声対話のマルチスレッド化は困難である。

しかしながら、テキストチャットでは、全ての発言が発言履歴に保存・表示され、参加者は随時任意の発言を読むことができる。つまり発言履歴が人間の認知能力を補っている。このため、個々の発言を時間的に分離して発言する必要が無いため、話者交替の制約が解消される。しかも、複数のスレッドが混在しても、それらを分離して読み、理解することができるため、単一スレッドの制約も解消される。

実際にテキストチャットではマルチスレッド対話が生じやすいことが確認されている。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とすると、マルチスレッド状況を維持することが実際には困難となる。このため、ユーザ自身が各発言に以下のような表現を自発的に付加することにより、この問題を解決していることが、著者ら自身のこれまでの研究で明らかとなっている[7]:

1. 相手指定: 誰に向けた発言かを明記する表現
例) まだまだ今年はこれからですよ。 > **Bさん**
2. 話題指定: どの話題の関連発言かを明記する表現
例) 私も大好きですよ。 > **チーズケーキ**
3. 先行発言指定: どの発言に対する発言かを明記する表現(対象発言の一部をコピー&ペーストし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。

調査の結果、テキストチャットにおいて、相手指定が全発言の 13.1%で、話題指定が 9.2%で、先行発言指定が 2.3%で使用されており、全発言の約 1/4 の発言でこれら 3 つのいずれかの表現が使用されること、特に隣接しない発言同士を関連付ける必要がある場合にこれらの表現が多用され、マルチスレッド状態が維持されていることが明らかとなった。したがって、以上のテキストチャットに関する知見から、音声対話にも発言履歴を導入し、さらに上記3つのような発言同士を関連付ける機能を追加することにより、音声対話をマルチスレッド化できると考えられる。

3.2 システム概要

前節の検討結果に基づき、音声によるマルチスレッド対話を容易に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 の

特徴は、発言履歴を有することで任意の発言を随時聴取できる点、および個々の発言を関連付ける「相手指定情報」と「対応先行発言指定情報」を付与する機能と、これら関連付け情報を用いて発言を聴取する機能を提供している点にある。

ChaTEL は、一般的なテキストチャットシステム同様、サーバ・クライアント構成をとる。ユーザは、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音声ファイルをダウンロードし、その発言内容を聞く。したがって、本システムで実現されるコミュニケーションは、厳密にはリアルタイムなものではない。しかし、発言が録音されてから、聴取者によってダウンロードされ聴取されるまでの時間差は、ボイスメールやボイスメモと比べて格段に短い。ゆえに、テキストチャット(これも厳密にはリアルタイムではない)と同等のリアルタイム性が実現されている。

図 1 に、ChaTEL のユーザインタフェースを示す。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、自分と、すでにログインしているメンバーのハンドル名が図 1 右部分の参加メンバー一覧に表示される。

発言を録音する場合、(1)通常録音、(2)対応先行発言指定録音、(3)相手指定録音の 3 つの方法がある。発言が録音されると、その発言の通し番号(発言 ID)と、発言者のハンドル名、および発言時刻が発言履歴の最下段に追加される。通常録音によって録音(「録音」ボタンを使用)された場合、履歴上に表示される情報はこれらのみとなる。対応先行発言指定録音(「発言への返答録音」ボタンを使用)は、先行するいずれかの発言に対する応答を録音する方法である。この方法で録音された発言については、発言履歴上に表示される情報の末尾に、「> [10]」のように、関連付けられた先行発言の発言IDが追加される。相手指定発言(「発

言者への返答録音」あるいは「相手指定録音」を使用)は、どの参加者に対する発言かを指定して録音する方法である。この方法で録音された発言については、発言履歴上に表示される情報の末尾に「>>Mary」のように、指定された参加者のハンドル名が追加される。

発言を聴取する場合には、(1)発言履歴一覧上で選択した発言を聞く方法(「これを聞く」ボタン)、(2)直前に聞いた発言の次の発言(発言履歴上で直下にある発言)を聞く方法(「次を聞く」ボタン)、(3)相手指定で自分が指定されている発言を聞く方法(「自分宛を聞く」ボタン)、および、(4)直前に聞いた発言に指定されている対応先行発言を聞く方法(「先行発言を聞く」ボタン)、の 4 つの方法がある。

4. 実験

次の 2 つのシステムの比較実験を行なった。

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。単純な録音と再生機能のみをもつシステム。

ChaTEL:3.2 で説明した ChaTEL を使用。

いずれのシステムも発言履歴を持つため、マルチスレッド化を実現できる可能性がある。しかし、前述のとおり、テキストチャットにおいて、マルチスレッド状態が複雑になった場合には発言間の関連付け情報が重要となっている。そこで音声の場合でも、発言間関連付けが可能な場合にマルチスレッド化がより促進されるかどうか、またその場合にどの関連付け情報がより有効に機能し、発言および聴取がどのように行われるかを、比較実験によって調査する。

被験者は、4 人の大学院生からなるグループ 4 組、計 16 人である。全員、テキストチャット利用経験はあるが、音声 CMC システムの利用経験はない。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。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 の順序で実験を行なう被験者群と、まず ChaTEL を使用した後 Baseline の順序で実験を行な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み合わせは、すべて異なる組み合わせとした。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り当てとしたのは、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うなる

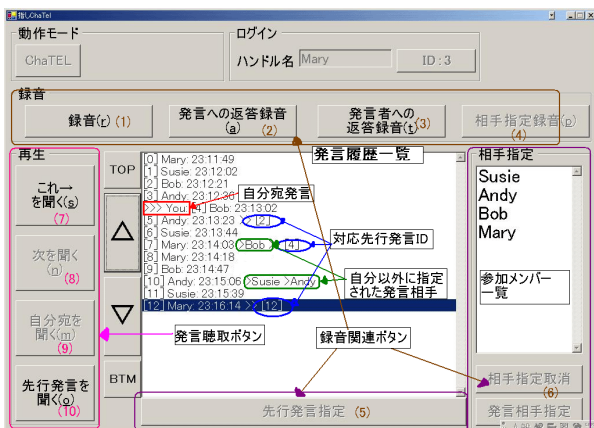


図 1: ChaTEL のユーザインタフェース

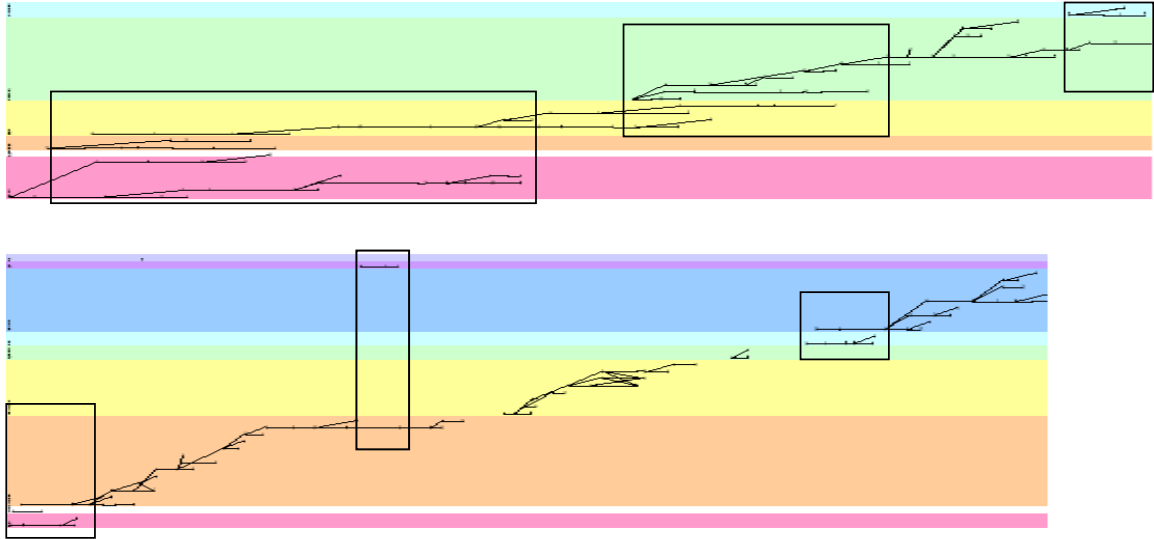


図 2：スレッド構造図（上：ChaTEL，下：Baseline）：縦軸の個々の色帯はスレッドに対応。横軸は時間（1 秒単位）に対応。四角で囲まれている部分でマルチスレッド化が起きている。

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 分自由に対話するよう教示した。なお、与えた話題は、「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、比較的自由的なテーマである。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事や、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。

5. 結果

5.1 マルチスレッド化状況の比較

図 2 に、Baseline と ChaTEL のそれぞれについてのスレッド構造図を示す。このスレッド構造図は、実験での発言履歴データと、どのボタンをどの時刻にどのユーザが使用したかという発言および聴取行動ログに基づいて実験者が作成したものである。図において、横軸は時間に対応している。横に延びる色のついた帯がそれぞれスレッドに対応している。色帯の中にある小さな黒点が個々の発言に対応し、それらを結んだ細い黒い線が意味的に隣接する発言を示している。途中、いくつか枝分かれしている部分（分岐）は、たとえば 1 つの質問に対して、複数の被験者による回答があったような場合の隣接ペアに相当する。四角で囲んだ部分は、複数のスレッドが同時進行している部分を示している。

5.1.1 時間単位の平均スレッド数及び分岐数

Baseline と ChaTEL のそれぞれについて、1 秒ごとの平均スレッド数とスレッド内の分岐数を求めた。具

表 1: Baseline 対 ChaTEL の秒単位の平均スレッド数および平均分岐数

	平均スレッド数		平均分岐数	
	Baseline	ChaTEL	Baseline	ChaTEL
平均	1.2	1.62	1.97	2.63
標準偏差	0.62	0.92	1.11	1.42

体的には、同時にいくつのスレッドがあるか、および同一スレッドに含まれる意味的に隣接する発言間を連結させた直線が同時に何本存在するかを 1 秒毎に求めることによって、1 秒単位でのスレッド数と分岐数を算出し、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。結果を表 1 に示す。平均スレッド数および平均分岐数のいずれについても、Baseline と ChaTEL の間に 1% 水準で有意差が認められた。すなわち、ChaTEL の方が複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分岐が多い複雑な構造になっていることがわかる。

5.1.2 スレッド継続時間と最大スレッド数

2 つ以上のスレッドが同時存在している部分について、同時存在している並行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。結果を図 3 に示す。図に示すとおり、最大並行スレッド数は ChaTEL では 4 であるのに対し、Baseline では 3 である。また、複数スレッド状態の継続時間についても、Baseline よりも ChaTEL のほうがトータルで 2 倍ほど長く、しかもどのスレッド数についても ChaTEL のほうが長いことがわかる。特にこの差は、並行スレッド数が増えるほど顕著で

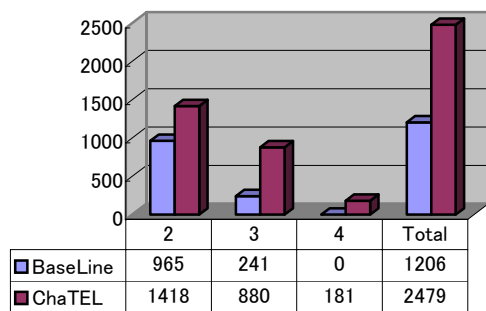


図 3: Baseline と ChaTEL における複数スレッド同時存在継続時間 (縦軸が継続時間 (秒), 横軸が並行スレッド数)

ある。さらに、複数のスレッドが同時存在しているときに、各スレッドに何人の被験者が同時参加しているかを発言ログに基づき調査した。結果を表 2 に示す。ChaTEL の場合、4 人全員が 1 つのスレッドに参加しているケースが 8 もあり、確実にマルチスレッド状態が生じていることが示された。

以上から、Baseline ではマルチスレッド対話状況が生じにくかつ維持され難いのにに対し、ChaTEL ではマルチスレッド対話状況が生じやすかつ維持されやすく、しかもより複雑な状況でも対応可能であることが明らかとなった。

5.2 発言・聴取行動

5.2.1 発言・聴取方法の使用頻度と割合

表 3 に、Baseline と ChaTEL を用いた場合の総対話時間と総発言数を示す。この結果に示すとおり、いずれのシステムを用いても発言数に差は無い。

表 4 に、ChaTEL を用いた場合の各発言録音方法の使用頻度と割合を示す。表に示すとおり、被験者は 8 割近い割合で対応先行発言指定録音を使用している。このことは、発言の際に、自分の発言がどのスレッドに関するものであるかを強く意識し、それを明示しようとする傾向があることを示している。一方、相手指定録音はわずか 7.4% しか使用されていない。つまり ChaTEL においては、「誰宛の発言であるか」よりも、「どの発言への応答であるか」の方が重視されている。これは、テキストチャットにおいて、相手指定が 13.1% であったのに対し、対応先行発言指定がわずか 2.3% でしか使用されていなかった結果と対照的である。

表 5 に、Baseline と ChaTEL のそれぞれにおいて使用された各聴取方法の頻度と割合を示す。Baseline においては、選択聴取よりも次聴取が多いのにに対し、ChaTEL では次聴取よりも選択聴取が多い。また、ChaTEL のみに提供されている、自分宛聴取機

表 2: 複数スレッドが同時存在する際の 1 スレッドへの同時参加人数別のスレッド数

参加人数	Baseline	ChaTEL
2 人参加	11	6
3 人参加	8	8
4 人参加	0	8

表 3: Baseline と ChaTEL のそれぞれを用いた実験における総対話時間と総発言数

	Baseline	ChaTEL
総対話時間	5,528 秒	5,689 秒
総発言数	339	337

表 4: ChaTEL を用いた場合の各発言録音方法の使用頻度と割合

	頻度	割合
通常録音	47	14.0%
対応先行発言指定録音	265	78.6%
相手指定録音	25	7.4%
合計	337	100%

表 5: Baseline と ChaTEL のそれぞれにおいて使用された各聴取方法の頻度と割合

	Baseline		ChaTEL	
	頻度	割合	頻度	割合
選択聴取	595	42.6%	1,002	57.0%
次聴取	801	57.4%	737	42.0%
自分宛聴取	—	—	1	0.1%
対応先行発言聴取	—	—	16	0.9%
合計	1,396	100%	1,756	100%

能と対応先行発言聴取機能の使用頻度は、両方合わせてわずかに 1% であり、ほとんど使用されていない。

5.2.2 聴取行動と発言履歴の関係

前述のとおり、ChaTEL では発言全体の 9 割近くに対応先行発言と相手が指定されているため、発言聴取順序もこの情報に依存することが予想される。一方 Baseline では、発言を聴取するまで内容も他発言との関連性もわからないため、発言履歴に表示されている順番どおりに聴取を行うこと、さらにその結果として、発言履歴上の最新の発言からあまり離れることなく再生を行うことが予想される。

そこで、各被験者の聴取行動のログデータに基づき、聴取された発言が、その直前に聴取された発言と発言履歴上でどれだけ離れていたか (聴取間隔)、および、

聴取された発言がその時点での最新発言からどれだけ遅れていたか(遅延距離)についての分析を行った。

聴取間隔については、直前に聴取した発言の次の発言を聴取する(「次聴取」だけでなく、「選択聴取」でこれを行った場合も含む)を標準的な行動パターンと考え、以下のようなポイントを各聴取行動に付与し、そのポイントの絶対値を距離とした。

- 1) Next: 次の発言を聴取. 0 point
- 2) Repeat: 同一発言の繰り返し聴取. -1 point
- 3) Back: $n(n>0)$ 発言前を聴取. $-n-1$ point
- 4) Jump: $n(n>1)$ 発言先を聴取. $n-1$ point

一方、遅延距離については、聴取された発言の発言IDと、その時点での最新発言の発言IDとの差を距離の値とした。表6に、BaselineとChaTELのそれぞれについての、聴取間隔と遅延距離の平均値を示す。また表7に、Next以外の聴取行動によって、聴取ログ全体がいくつのブロック(1つのブロック内はNext聴取行動のみによって、発言履歴に沿って時系列的に聴取されている)に分割されたか(ブロック数)、各ブロックの平均長(なお、N-1回目の聴取とN回目の聴取の後にいずれもNext以外の聴取行動が行われた場合、N回目の聴取はブロック長1のブロックとみなす)を示す。

Baselineについて、表6より聴取間隔が0に近く、また表7よりブロック長がChaTELより3倍ほど長いことから、Baselineでは発言履歴上の時系列順に聴取行動を取る傾向が強いことがわかる。逆にChaTELの場合は、発言履歴に沿わない聴取行動を取る傾向が見られる。また、遅延距離についても、BaselineのほうがChaTELよりも距離が小さいことから、Baselineでは常に発言履歴上の最新発言を追う形で聴取行動を行う傾向が強いことがわかる。

さらに、BaselineとChaTELのそれぞれについて、4種の聴取行動それぞれがどの程度行われていたかを、やはり聴取行動ログに基づいて調査した。表8に、各聴取行動の平均頻度を示す。Back聴取行動については、ChaTELではBaselineに比べて有意に4倍近く多く行われていることがわかった。

5.2.3 聴取行動とスレッド構造の関係

ChaTELの場合は、8割ほどの発言が対応する先行発言と明示的に対応付けられているため、スレッド構造に沿った聴取行動が行われることが予想される。これに対し、Baselineの場合は発言間の関連付けが発言履歴上では不明であるため、スレッド構造に従った聴取行動が行われ難いと思われる。表9に、同じスレッドに属する発言を連続して聞く場合の連続長(スレ

表6: 聴取間隔と遅延距離の平均 (***)は1%水準で有意差あり)

	Baseline	ChaTEL	
聴取間隔	0.49	1.63	***
遅延距離	1.01	2.53	***

表7: 発言履歴に沿ったブロック数、ブロック長 (***)は1%水準で有意差あり)

	Baseline	ChaTEL	
ブロック数	20.8	43.1	***
ブロック長	9.5	3.3	***

表8: 4つの聴取行動の平均頻度 (***)は1%水準で有意差あり)

	Baseline	ChaTEL	
Next	69.1	74.1	
Repeat	4.6	6.7	
Back	5.3	19.6	***
Jump	9.7	16.1	

表9: スレッド構造に従ったブロック数、ブロック長 (***)は1%水準で、**は5%水準で有意差あり)

	Baseline	ChaTEL	
スレッドブロック数	32.3	51.5	***
スレッドブロック長	3.4	2.3	**

ッドブロック長)とスレッドブロックの数(スレッドブロック数)を示す。表9から、予測に反してBaselineの方がスレッドブロック長が長いことがわかった。しかし、これはBaselineの方が単一スレッド状態が多いため、Next聴取によって同一スレッドを連続聴取する可能性が高く、また同じ理由によりBackやJumpなどを行っても、やはり同一スレッドに偶然行き当たる可能性が高いためということが考えられる。

そこで、複数のスレッドが並行している箇所のみを切り出し、それらの箇所においてBackとJumpで同一スレッドを連続聴取している割合を調べた。この結果、Baselineでは20.1%であったのに対し、ChaTELでは41.2%であり、両者には2%水準での有意差が認められた。このことから、マルチスレッド状態になっている際、ChaTELの方がBaselineよりスレッド構造に沿った聴取を行っていることがわかった。ただし、その割合はたかだか4割程度であるため、ChaTELを用いた場合でも完全にスレッド構造に従うわけではなく、比較的高い頻度で異なるスレッドに移動することも示された。

6. まとめと議論

本稿では、音声による新しいコミュニケーションスタイルを模索する試みとして試作した「マルチスレッド対話」を可能とする音声コミュニケーションシステム“ChaTEL”を用い、マルチスレッド対話を促進するための機能がどのように使われるのかを、ユーザの発言行動および聴取行動を分析し、検討した。その結果、発言時は、ChaTEL を用いた場合「対応先行発言指定情報」の付与機能を多用しながらマルチスレッド対話を行っていることが明らかとなった。一方、聴取行動は、Baseline の場合、発言履歴に沿って時系列に聴取行動を行うのに対し、ChaTEL の場合は、遡って過去の発言を聴く頻度が高い。しかしその際、ChaTEL が提供する自分宛聴取や対応先行発言聴取機能はほとんど使用されず、また同一スレッドに属する発言のみをたどって聴取するわけではないことも明らかとなった。以上から、ChaTEL では、同一スレッドをひとまとまりとして聴取し発言するのではなく、複数のスレッドを渡り歩きつつ、自分の聞きたい発言を逐次聴取しながら必要に応じて応答発言を行っていることが伺える。

Baselineでも過去に遡った発言に対する応答をすることは可能だが、それがどの先行発言に対応しているのかを聴取者が把握することが難しくなることが容易に推察できる。このため、発言者はそのような応答をすることを心理的に抑制され、関連性が把握しやすい最新発言近傍に対する応答のみをすると考えられる。一方ChaTELの場合は、発言間の関連付けを先行発言対応機能等によって明示的に示すことができるため、「発言番号をたどれば関連性を理解してもらえる」という安心感が生じる。この結果、発言者は遠い過去の発言にも気軽に応答可能となり、最新発言近傍以外の箇所にも自由に跳躍するようになると考えられる。

また、ChaTELの場合、録音時には相手指定録音の使用頻度は他の録音方法よりも低く、聴取時には、自分宛聴取、対応先行発言聴取の利用頻度が非常に低いというように、録音・聴取方法のうち、利用頻度の高いものとそうでないものの差が大きいことがわかった。相手指定録音の利用頻度が低いことについては、先に述べた「発言番号をたどれば関連性を理解してもらえる」安心感から先行発言対応録音がかなり高い頻度で使われていることが大きく影響していると推測できる。さらにその他の理由として、特定の参加者にどうしても応答してほしい場合や、新しいスレッドが生じた直後である場合には、相手指定録音を利用し、そう

でない場合には、他の録音方法を利用するといった、発言の種類やスレッドの状況により利用する録音方法が異なる可能性が考えられる。また、聴取方法の自分宛聴取、対応先行発言聴取の利用頻度が非常に低いことについては、ChaTELでの選択聴取の利用頻度がBaselineよりも増えており、発言を順番にきく次発言聴取が減っていることから、選択聴取によって、自分宛聴取、対応先行発言聴取を手動で実行している可能性があると推測できる。被験者のインタビューから「機能が複雑」「ボタンが多すぎる」という指摘があり、機能の簡略化の要求が高い。そこで録音方法については、それぞれの方法とそれに対応する発言内容の詳細な分析を進めることで、発言状況にあった録音方法を提供し、機能の煩雑さを軽減するためのシステム改良を検討したい。また、機能の複雑さに対し、被験者インタビューで「なれると使いやすくなると思う」「コツがいる」というような意見もあり、さらに本システムの運用事例を増やすことで、インタフェースデザインの改良を行い、より実用的なシステムの構築を進めたい。

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。

参考文献

- [1] 小倉加奈代,西本一志: 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,情報処理学会論文誌「ユビキタス社会におけるコラボレーションサービス」特集号,Vol.47,No.1,2006.1.(印刷中)
- [2] L. Berc, H. Gajewska and M. Manasse: Pssst: Side Conversations in the Argo Telecollaboration System, Proc. of the 8th ACM UIST, pp.155-156, 1995.
- [3] 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, FIT2003 講演論文集, LK-005, pp.273-274, 2003.
- [4] P.M. Aoki, M. Romaine, M.H. Szymanski, J.D. Thornton, D. Wilson, and A. Woodruff: 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.
- [5] http://www.nttdocomo.co.jp/info/news_release/page/20051019.html
- [6] <http://japan.cnet.com/news/tech/story/0,2000047674,20089462,00.htm>
- [7] 小倉加奈代,石崎雅人: チャット対話における話題推移に関する特徴分析,人工知能学会研究会資料,SIG-SLUD-A202-03,pp.13-19,2002.11.