

効率的な非同期ディスカッションを可能にする 会議録システムの検討

A Study on a Minutes System to Realize Efficient Asynchronous Discussions

平島 大志郎 田中 充 勅使河原 可海
Daishiro HIRASHIMA Michiru TANAKA Yoshimi TESHIGAWARA

創価大学大学院工学研究科
Graduate School of Engineering, Soka University

要旨：一般に研究会のような形態の会議では、プレゼンテーションの発表の後に質疑応答が行われる。この質疑応答の内容は、会議後に研究室等でディスカッションし、発表者の研究を進める上で非常に有益な情報だと考えられる。そのため従来は、ペンでメモをしたり PC を用いてテキストで記録することが多く見受けられてきた。

しかしながら、質疑応答内容の記録には非常に労力がかかる上に、白熱した早いやり取りの質疑応答は正確に記録することが難しい。我々はこれまで複数の参加者で協調して質疑応答内容を記録し、会議後に議論のトピックとして提供できる協調型会議録システム検討を行ってきた。本稿では検討してきたシステムの開発と、非同期環境でのディスカッションを可能とする会議システムの検討について述べる。

Abstract: Generally, at conferences and workshops, the Q&A sessions are held after presentation. These Q&A topics are so useful to make presenters' researches improve. Therefore, researchers or participants usually record their Q&A topics with a pens or PCs. However, recording these topics needs much labor, and it is difficult to record them accurately in quick turn dialogues. Therefore, we have developed a collaborative meeting minutes system for enabling to record Q&A topics among participants and to present them after conferences and workshops. In this paper, we describe a collaborative meeting minutes system and discuss its validity to realize efficient asynchronous discussions.

1. はじめに

一般に研究会の発表のように PC を用いてプレゼンテーションを行う形の会議では、プレゼンテーションの発表の後に質疑応答が行われる。発表に関して参加者と発表者によって交わされたこれらの質疑応答内容には、発表の後に再検討を行う事項や組織内の共同研究者間で議論を行う事項が含まれており、これらの内容は発表者の研究を進める上で非常に有益な情報である。そのため、従来では発表者自身が質疑応答内容をペンでリアルタイムに筆記したり、発表者と同じ所属の参加者が PC を用いてテキストでメモを取ることが多く見受けられてきた。

しかしながら、発表と質疑応答での緊張の中、発表者がリアルタイムに質疑応答内容を詳細に記録することは非常に困難であり、記録できるのはキーワード程度にとどまる。また、リアルタイムに参加者の一人が PC を用いて質疑応答の会話内容をテキストで記録することは、記録者の質疑応答内容の理解力やタイピング力に大きく依存し、十分な記録を行うことは非常に難しい。

そこで今まで同期対面型会議の中でもプレゼンテーションと質疑応答を行う形の会議を対象に、参加者間で協調して質疑応答内容等のデー

タを記録し、会議後に効果的に提示できる協調型マルチメディア会議録システムの開発と評価を行ってきた^[1]。このシステムは、複数人で協調して発言内容の記録を可能にし、これまでに述べたような問題点を解決できる。

また、同期対面型会議の中で取得された質疑応答の内容は、会議終了後に会議参加者や共同研究者間で議論するのに有益な事項が含まれている。そこで、会議終了後の非同期な議論を支援するためのシステムとしての拡張を考える。

2. 研究の目的

本研究では、組織内で特定の研究を行っている数人から十数人の個人で構成されるグループを対象とする。研究発表での議論を、学会発表後にグループ内で議論するためには、そのときの質疑応答内容を記録する必要がある。そのため、グループ員の研究発表の質疑応答内容を可能な限り効率的にテキストで協調記録可能な Web アプリケーションの検討・開発について述べる。

また、このようにして同期型会議で記録された質疑応答内容を、発表後のグループでの議論にスムーズに繋げる際の検討を行い、この問題を解決するシステムについても検討する。

3. 想定する環境

本研究の想定する環境は、研究発表が行われる同期型会議の環境と、発表後に Web 上で議論を行う非同期環境の 2 種類である。

まず始めの同期型会議の環境は、発表者がプレゼンテーションと質疑応答をする際に、複数人の参加者のうち 2 名ほどが協調して質疑応答内容の記録を行う環境を想定する。

次に非同期環境に関しては、同期型会議の環境で記録された質疑応答内容等をトピックとして、発表後に Web 上で議論を行うことを考える。同期型会議の環境で収集された質疑応答内容をトピックにして、会議後に非同期環境でそれらを議論するモデルを図 1 に示す。

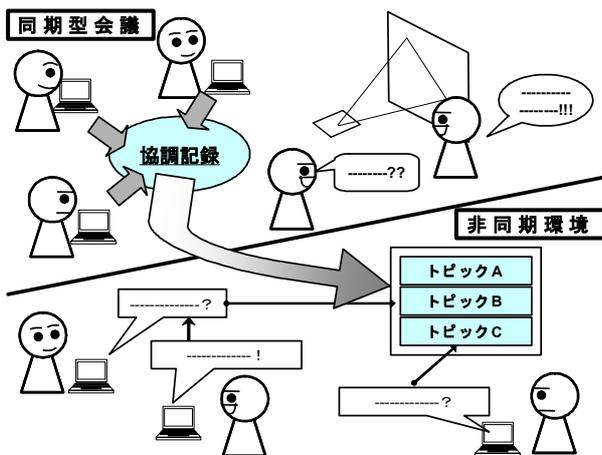


図 1. 同期型会議と非同期環境の議論のモデル

これらの協調作業は、リアルタイムな会議中においては、その場にいる会議参加者が PC と無線 LAN でのネットワーク接続環境さえあれば全員が参加できるように、Web ブラウザのみで動作する Web アプリケーションとして設計を行う。このモデルに従って、同期会議で行われた質疑応答内容を会議後に効率的に参加者で議論できるようになると、発表者が研究内容をより深く検討することが出来るようになると思われる。

4. 同期型会議の会議録システム

本研究の目的を達成する同期型会議のシステムには必要な要件が複数存在する。本章ではそれらの要件を示し、それらをクリアする協調型会議録システムについて具体的に述べる。

4.1 システムの必要な要件

① Web で動作し、複数人が協調記録・修正

本研究では同期型会議の参加者が容易にシステムに参加できるように、Web アプリケーションとして設計を行う。そのために、Web ベースの環境で、その上複数人が効率よく協調入力できるシステムである必要がある。

② 発言者を容易に記録

質疑応答内容を記録する際には、発言者が誰なのかも同時に記録する必要がある。しかしながらマウスを使う動作はキーボードから手

を離すことになってしまうので、手間がかかってしまいあまり好ましくない。また事前に発言者全員の名前を用意することは難しいので、誰が発言しても記録できる仕組みが必要となる。

③ 会議プレゼンスの記録

一つの会議には複数の発表があり、それらを区別するために、発表毎に記録を区切る必要がある。また個々の発表にも「プレゼンテーション中」や「ディスカッション中」など会議プレゼンスの変化を記録して、会議後に分りやすく切り分けて提示する必要がある。

4.2 協調型会議録システム

同期型会議において、効率よく質疑応答の内容を記録・管理することを目的としたシステムのプロトタイプを Web アプリケーションとして開発した。開発は JSP で行い、サーバは Apache + Tomcat で構築した。本節ではこの協調型会議録システムについて、前節に検討した要件とともに述べる。

4.2.1 協調記録機能

4.1 で述べた要件を満たすために、システムのプロトタイプは Web チャットのようなアプリケーションとして開発した。この動作している協調型質疑応答録システムの画面を図 2 に示す。

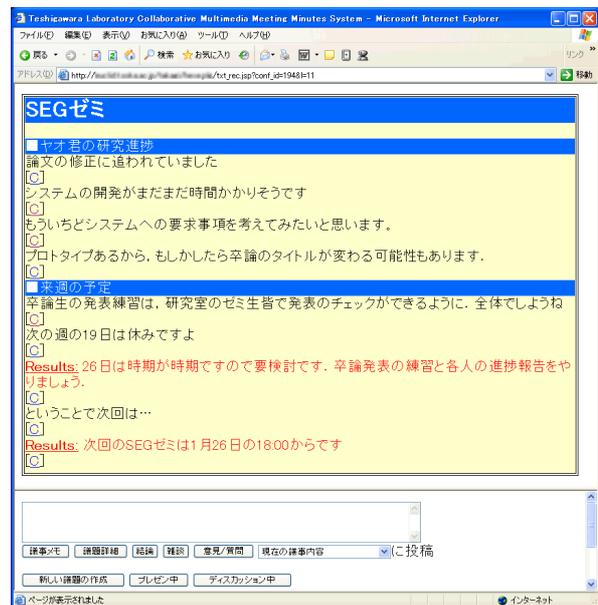


図 2. 協調型質疑応答録システム

画面はフレームで別れており、下のフレームのテキストボックスに質疑応答の内容を打ち込み、ボタンを押すと投稿が完了される。上の画面は、現在のところ 5 秒間隔で更新され、投稿された内容が順に表示されるようになっている。一度投稿した内容を修正する際には、該当するコメントの下の[C]のリンクをクリックするとコメント修正用の新しいウインドウが別に表示され、誤字や内容の修正が可能である。質疑応答記録者の PC でこの Web アプリケーションを使いながらお互いに内容を投稿していくことで、

同期型会議の中でリアルタイムに同じ画面を共有して質疑応答の記録が可能になる。これらにより、4.1 で述べた要件①は満たすことが出来ると考える。

4.2.2 発言者の効率的な記録

リアルタイムな発言を PC にテキストで記録することについて事前に調査を行った結果、発言の内容を聞き、内容を把握した上で適度に要約し記録するという工程にかなりの負荷がかかることが分っている。そのため、その上キーボードから手を離してマウスで操作し発言者の名前を記録するような動作をすることは極めて難しいと判断した。そこで、発言者の名前の記録は、質疑応答内容の後に簡単な識別子をつけて記録することにした。具体的に記録方法を表 1 に示す。

表 1. 発言者の記録方法の例

協調入力についてはどのような運営形態をとるのでしょうか？（だいしろう）
↓
平島：協調入力についてはどのような運営形態をとるのでしょうか？

質疑応答内容の後に記録された識別子は“（”によって区切られ、簡単に入力できる簡略化した名前等で質疑応答記録者が記録する。システムは入力された内容から識別子を抽出し、すでに登録されている人物であれば発言者の認識を行い、表 1 の下のように質疑応答内容を変換する。システムに登録されていない識別子であった場合には変換を行わずにそのまま表示を行うが、後から識別子の登録を行うことによって、同様に質疑応答内容の発言者をシステムが認識することができる。これらにより 4.1 で述べた要件②は満たすことが出来ると考える。

4.2.3 会議プレゼンスの記録

個々の発表の区切りや「プレゼンテーション中」や「ディスカッション中」など会議プレゼンスの区切りは、前項の発言者の記録に比べて入力する時間的余裕がある。そのため、会議プレゼンスの記録はボタンを押して記録することにした。発表者が変わるときには図 2 の下の「新しい発表の作成」を押すと、新しいウィンドウが現れて発表の情報を入力できる。また「プレゼンテーション中」や「ディスカッション中」の切り替えも同じくボタンを押して容易に切り替えることができる。会議プレゼンスを切り替えると、その時点で画面上には青い線で区切られて表示される。これらにより 4.1 で述べた要件③は満たすことが出来ると考える。

5. 会議録システムの使用実験

今回、本システムの協調記録の有効性を検証するために、10 分程度のプレゼンテーションの後ディスカッション形式で個人の研究の議論を行うゼミで使用実験を行った。実験は 3 度行い、

1 人で質疑応答内容全てを記録する方式と本システムを使って 2 人で協調して記録する方式を、システムのログと質疑応答記録者へのアンケートにより比較検討した。対象としたゼミの参加者は 9 名（教員 2 名 + 修士学生 7 名）である。各実験のディスカッション時間と質疑応答の記録個数を表 2 に示す。

表 2. 実験結果

	実験 A	実験 B	実験 C
ディスカッション時間	14 分 44 秒	5 分 00 秒	14 分 12 秒
1 人	18 個	7 個	5 個
2 人（協調）	36 個	8 個	18 個

実験結果により、1 人よりも 2 人で協調記録した場合の方が多くの質疑応答内容を記録できていることがわかる。また 1 人で記録を行う場合と 2 人で記録を行う場合の違いを質疑応答記録者の 4 名にアンケートを取得した結果、1 人の場合は、ディスカッションが行われている時間中、常に集中することが必要になり非常に疲れるという意見や、記録できない場面やミスが頻発するという意見が得られた。それと比較して 2 人協調して記録した場合は、1 人に比べて余裕があるため聞き流しやミスが少なくなるという意見や、記録を取る際に非常に楽になり、もしミスして記録した場合があつたとしても互いに修正し合えて、はるかに余裕を持って記録できるという意見が見られた。他にも、2 人で記録した場合のみに、質疑応答記録者自身が発表者との質疑応答に参加する場面が観察できた。これは、発言内容をもう一方の記録者が記録できることにより可能になったことであると考えられる。また 4.1 の要件②と要件③で述べた発言者の記録と会議プレゼンスの変更の機能も質疑応答の記録と共に使用されて有効に機能した。

以上のことにより、本研究の提案する議論のモデルの中で、Web ベースの協調型会議録システムが有効に動作し、同期型会議の中で生成されるトピックを効率的に収集記録できたと考える。これらのトピックは非同期ディスカッションを支援するシステムで用いることが出来ると考えられ、検討項目に洩れのない効率的な非同期ディスカッションを行えるようになると考えられる。

6. 非同期環境のディスカッションを支援するシステムの検討

会議後、質疑応答内容についてディスカッションするには、記録された質疑応答内容をトピックとしてグループ内で共有することが必要である。また、共有された質疑応答内容から Web 上での議論を行えるようにする仕組みが必要である。本研究では、近年組織内で導入されている blog を用いて Web 上で議論することを考える。blog は従来のグループウェアの掲示板に比べて、ユーザが気軽に情報発信することができ

る。それは「公式」の場所で自分の意見を投稿することに比べて、blogの自分のエントリーに投稿するほうが安心感があるからである、とされている。本来blogは個人や組織で容易に情報発信をすることが主な目的で使われているが、本研究のように同期型会議で行われた内容を議論のトピックとして、非同期環境のblogへ議論をリンクすることが可能になれば、blogの利点を活かしつつ、より柔軟で自由な非同期の議論が可能になると考える。

同期型会議でのシステムである協調型会議録システムとblogを関連付けるために、本研究ではトラックバックを用いる。トラックバックはblogシステムで使われてきている技術で、他のblogのエントリーやWebページに存在するトラックバックURLに対してトラックバックPingを送ることにより、そのページのリンクが生成できる。トラックバックは一般には記事の引用やリンクを張る際に使用されるが、これを本研究では同期型会議で生成されたトピックなどにトラックバックURLを設置し、blogからトラックバックPingを送ることを可能にする。トラックバックを用いて同期型会議で生成されたトピックとblogを関連付けた例を図3に示す。

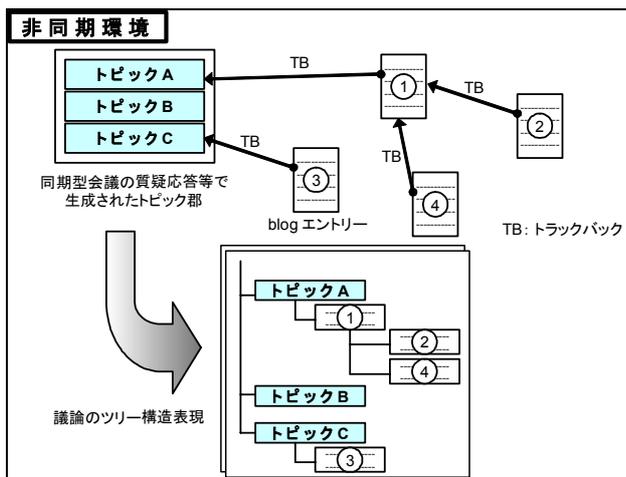


図 3. 同期型会議のトピックとblogの関連

このようにblogとトラックバックを用いて議論を構築していくと、同期型会議で話題になったどの部分から派生した議論なのかが明確化出来て、また投稿内容もhtml等を用いて自由に記述できるという利点の反面、集中管理されていないために議論の流れが見えにくくなるという欠点がある。この欠点を解消するためには、図3のようにシステムがトラックバックの送信を認識し議論のオーバービューをツリー構造等で行う。

これらのことにより、同期型会議で議論された質疑応答内容をトピックとして提示し、トラックバックによって議論の関連を明示化でき、非同期環境での議論が可能になると考えられる。

7. 関連研究について

会議でテキストを用いてリアルタイムに議事録を作っていくシステムとしてLiteMinutes^[2]とEgitool^[3], EnhancedChat^[4]が存在する。LiteMinutesは会議単位の議事録のみを作成するため、本研究のように特定の議事やトピック単位で管理が出来ない。また複数人による協調入力も検討されていない。Egitoolは会議の消極的参加者に着目し、議事録を記録させることにより会議に貢献させることを目的にしており、本研究の議論のモデルとは異なる。EnhancedChatは学会発表でリアルタイムにチャットで議論の促進とログの可視化を目的として、チャットのログを2次元的に配置して音声と関連付けるシステムであり、質疑応答の記録を協調して取ることを考えていない。いずれのシステムも会議後に非同期の議論をすることを考慮してなく、本システムのように同期型会議の内容を効率的に記録して、非同期環境での議論につなげることは、個人の研究を促進させるだけでなく、研究室での情報共有の観点から見ても非常に有効に働くと考える。

8. まとめと今後の課題

本稿では、同期型会議での質疑応答内容を議論トピックとして会議後にディスカッションを行うまでのモデルを提示し協調型会議録システムについて検討した。使用実験により、同期型会議においてシステムは有効に動作し、より多くの質疑応答内容を記録することが可能であった。また、非同期環境でのblogを用いた議論手法についても考察した。

今後は非同期環境でのシステムを検討・開発し、同期型会議から非同期環境での議論までに有効に機能するか、経営的視点で定量的・定性的に評価を行っていく。

9. 参考文献

- [1] 平島大志郎, 田中充, 勅使河原可海: 協調型テキスト議事録システムの有効性の検討, 情報処理学会第55回グループウェアとネットワークサービス研究会(GN)研究報告2005-GN-55, pp.81-86, 2005.3
- [2] Patrick Chiu, John Boreczky, Andreas Girgensohn, Don Kimber: LiteMinutes: An Internet-Based System for Multimedia Meeting Minutes, Proceedings of the tenth international conference on World Wide Web, pp.140-149, April 2001
- [3] 江木啓訓, 石橋啓一郎, 重野寛, 村井 純, 岡田 謙一: 協同記録作成を基にした対面議論への参加支援環境の構築, 情報処理学会論文誌, Vol45, No.01, pp.202-211
- [4] 中西泰人, 倉持正之, 松川昌平: Enhanced-Chat: 音声と関連付け可能な2次元チャット, 情報処理学会論文誌, Vol43, No.12, pp.3550-3558