

修士論文

**会議における指示情報を用いた
議論の構造化とその応用に関する研究**

350903140 木内 啓輔

名古屋大学大学院 情報科学研究科 メディア科学専攻

2011年1月

会議における指示情報を用いた 議論の構造化とその応用に関する研究

350903140

木内 啓輔

近年、企業でのミーティングや大学研究室のゼミなど日常的な活動として会議に参加する機会が多い。それらの会議の内容を後で閲覧できるように、会議の様子を議事録として記録することも多い。そしてその議事録をより効率的に閲覧できるように、会議内容を構造化して記録することも多く行われている。

筆者の所属する研究室では、会議から映像・音声情報やテキスト情報、メタデータを獲得し、それらを統合して構造化し、再利用可能なコンテンツを作成し利用する技術であるディスカッションマイニングについて研究を行ってきた。我々は会議で行われた発言に対し発言のタイプとして、新しい話題を切り出すことを意味する「導入」と、直前の(あるいはいくつか前の)発言を受けて行った発言であることを意味する「継続」という二つのタイプを付与した。そして発言タイプに基づいて議論を話題単位にセグメンテーションすることで議論の構造化を行っている。

しかし、一般に議論が長くなるにつれ徐々に話題が逸れていってしまい、その議論の要点を把握しにくいという問題があった。そこで本研究では、同じ話題について話されているような発言群に対し、それらの発言群は他の発言と比べて、より強いつながりにあることを示すリンク情報を付与することで、議論構造の補強を行った。具体的にはまず、議論内で話されている話題と依存関係があると考えられる指示行為に着目し、専用のポインティングデバイスを用いて指示行為に関する情報を取得する仕組みを実現した。そして、同じ指示対象を参照している発言群に対しリンク情報を付与することで、議論をより詳細に構造化した。

また、付与したリンク情報の妥当性を評価するため、「継続元の発言と同じ指示対象を参照した場合は、一貫して同じ話題について話していることが多く、それ以外の話題について述べることは少ない」という仮説を検証することでその妥当性を評価した。

さらに、指示の情報および付与したリンク情報を用いた議論閲覧支援の仕組みを実現した。具体的には、付与したリンク情報を考慮した活性拡散アルゴリズムを用いて議論をフィルタリングおよびランキングすることで、閲覧者が閲覧したい発言群を選別する仕組みや、指示対象の情報と記録された発言内容を同時に閲覧できる仕組みを実現した。

A Study on Structuralization of Discussion based on Pointing Information and Its Applications in Meeting

350903140

Keisuke Kiuchi

In recent years, we have many opportunities to participate in meetings at a company, seminars of a laboratory as common activities. Minutes which describe contents of the meetings have an essential role for browsing the meeting contents after meetings. And discussions held at the meetings should be recorded and structured for browsing minutes efficiently.

We have developed a system which generates meeting minutes linked with videos and audio data of discussions and metadata as hints on semantic structures of discussions. Our technology is called "Discussion Mining." We categorize statements in meetings in two types such as "Start up type" which means the statement starts discussion of new topic and "Follow up type" which means that the statement follows the previous statement continuing the discussion of the same topic. Then we segment discussions into topics based on the statement types.

However, the structuring discussions has a problem that the topic gradually diverges from the main topic as discussions continue long time. So it becomes difficult that we browse discussions quickly. The purpose of this study is to structure discussions deeply based on strength of relationships between statements. The statements which discuss the same topic as their following statements have stronger relationships between the statements and their following ones than the statements which have different topics from the topics of their following ones.

First, we developed a system to capture pointing information (objects displayed on the screen that are referred by using pointing tools such as laser pointers) that we consider that pointed objects are related to the topics. Next, we distinguish the statements which refer to the same pointed objects from the rest of the statements.

Analyzing accumulated discussion contents, we verified a hypothesis that a statement and the following statement deal with the same topic if they refer the same pointed object.

And we also developed a system to support quick browsing of discussions with pointing information. The system consists of mechanisms of filtering/ranking of statements that employs a Spreading Activation technique on the discussion structures and quick retrieval of the statements that include pointing information.

目次

第1章	はじめに	1
第2章	ディスカッションマイニングによる議論内容の構造化と その利用	7
2.1	ディスカッションレコーダによる議論内容の獲得と構造化	7
2.1.1	ディスカッションレコーダによる議論内容の獲得	7
2.1.2	ディスカッションレコーダによる議論内容の構造化	11
2.2	ディスカッションブラウザによる会議の閲覧	14
2.3	これまでの構造化の問題点	20
2.4	まとめ	22
第3章	指示情報を用いた議論の構造化	23
3.1	会議における指示行為と議論の構造化	23
3.2	ポインタリモコンによる指示情報の取得	25
3.2.1	矩形選択	26
3.2.2	下線選択	32
3.3	指示情報を用いた議論セグメントの再構造化	33
3.4	指示情報を用いた議論セグメント間の関連付け	35
3.5	まとめ	38
第4章	議論の構造化に関する実験と考察	39
4.1	分析・検証内容	39
4.2	実験方法	40
4.2.1	ディスカッションレコーダによる指示情報の記録	40
4.2.2	アンケートによる発言と話題に関する情報の取得	41
4.3	結果と考察	47
4.4	まとめ	50
第5章	指示情報を用いた議論の閲覧支援	53
5.1	指示情報に基づく議論の構造情報を用いた発言群探索	53
5.2	発言と指示対象の同時閲覧	59
5.3	まとめ	61

第6章 関連研究	63
6.1 指示情報の取得に関する研究	63
6.1.1 講義における教材中の指示対象の抽出	63
6.1.2 会議コンテンツの協調的提示のためのレーザーポインタ情報の抽出	63
6.1.3 講義講演シーン検索のためのレーザーポインタ情報の抽出	64
6.2 議論の構造化に関する研究	64
6.2.1 gIBIS	64
6.2.2 音声認識された議事録からの議論マップの自動生成	65
6.2.3 議論構造の可視化による論点の発見と理解	66
6.2.4 ミーティングにおける会話のグラフ構造化	67
第7章 まとめと今後の課題	69
7.1 まとめ	69
7.2 今後の課題	71
7.2.1 スライド外のオブジェクトに対する指示	71
7.2.2 指示対象の種別に基づいたリンクの意味付け	71
7.2.3 指示領域に基づいた発言群同士の関連付け	72
謝辞	72
参考文献	75

第1章 はじめに

近年、我々は日常的な活動として会議に携わる機会が多い。例えば、何らかの項目に関して議論を行ったうえで意思決定を行うものや、お互いのアイデアを出し合いブラッシュアップするブレインストーミングを目的としたもの、あるいは個人では解決できなかった問題に対して複数人からの意見の獲得することで問題解決をはかるものなど、我々の周りには常に多種多様の会議が存在する。そして大学の研究室などでは、週に一度プロジェクト単位でミーティングを開き実装の問題を話し合ったり、あるいは月に一度研究室全体でゼミという形で自分の研究の進捗とともに抱えている問題を発表し意見をもらうことで問題解決をはかっていたりしている。

このような背景とともに、会議の様子を記録した議事録に対しても徐々に注目が集まり始めてきている。その理由としては、高い頻度で行われている各々の会議の場だけで会議の参加者全員がその内容を忘れず、すべて理解することは非常に難しいからである。この問題に対し、本橋らは会議中にホワイトボードに書きこまれた情報を TODO（遂行すべきタスクのリスト）や会議の資料と結び付け、TODO から関連する議論内容を参照できる手法を提案している [1]。また会議の内容を忘れないように個々人がメモを取ったとしても、各々の解釈に齟齬や情報の欠落が生じていないとは限らない。この問題に対し、倉本らは手書きメモとその元となった発言を関連付ける手法を提案している [2]。

しかし単純に会議で行われた発言を記録するだけでは、どの発言が自分にとって必要なかが分からず、後で閲覧する際に多大なコストがかかる。さらには書記による記録だけでは文量の個人差や書洩らしなど、正確な情報を振り返ることができない問題もある。そのため、自分が必要とする発言だけを効率的にかつ正確に閲覧する手段が必要となる。その手段の一つとして、会議を意味のある単位に構造化することで議事録を効率的に閲覧する方法が考えられる。

また、会議を構造化すると同時に会議の映像や音声も記録され始めた。それは人手によって作成された議事録だけでは情報の欠落を防ぐことができず、正確な情報を振り返ることができないためである。趙らは音声認識技術を用いて記録された議事録を自動的に構造化する手法を提案している [3]。Vedran や Christian らは、MISTRAL と呼ばれるシステムを用いて、映像や音声とミーティングの場所やトピックといったメタデータを関連付けた構造化を行うことで、正確な情報の記録とそれらの検索・閲覧を可能にしている [4, 5]。

我々はディスカッションマイニングと呼ばれる、人間同士の知識の交換の場であるミーティングから、映像・音声情報やテキスト情報といった実世界情報を獲得し、それらを半自動的に構造化することによって再利用可能な知識を抽出する技術の研究・開発を行っている [6]. 加えて我々は、ディスカッションマイニングによって記録されたコンテンツをウェブ上で簡単に検索・閲覧できるディスカッションブラウザの研究・開発も行っている [7].

ディスカッションマイニングでは、会議の様子を正確に振り返ることができるように、カメラやマイクによって映像や音声情報、および発言を要約したテキストに加え、効率的な検索・閲覧を実現するために、会議に関する様々な補足情報をメタデータとして記録している. 例えばミーティングの開始・終了時間や、ミーティングに使用されたパワーポイントのスライド構造情報、さらにアニメーションの切り替わり時間なども記録している. また、会議中に行われた発言に関するメタデータを記録することは、閲覧の際に重要な手掛かりになるとともに、議論の構造化を行う上で重要な情報になる可能性が高いと考えられる. そのため、我々は発言に関するメタデータとして、発言のタイムコード (発言の開始時間と終了時間)、発言者の情報、発言に対する評価情報 (賛成・反対)、発表資料と発言との対応関係などのデータを取得している.

さらに上記のメタデータに加え、発言間の依存関係をミーティング中に半自動的に記録することで、議論の構造化を行っている. 一般に、会議後に構造情報を人手で付与することはコストが高く、機械によって自動的に付与することは難しい. そのため、できるだけミーティング中に構造化を行い、かつミーティングの参加者に負担がかからないことが望ましい. また、構造化するにあたって、1つの議論に含まれる話題は1つであることが望ましい. それは後で議論を検索・閲覧する際、1つの議論内で複数の話題が話されていた場合、自分が欲している話題についての発言がどの議論に含まれているか、どこから見れば内容を理解できるのかが分かりにくくなるためである.

そこで我々は発言を、新しい話題を切り出すことを意味する「導入」と、直前の発言を受けて発言を行うことを意味する「継続」という二つのタイプに分類している. このように発言のタイプを単純な二つに限定することで、参加者はほとんど構造化に対して思考を割くことなく議論に集中できる. また我々はポイントリモコンと呼ばれる専用のデバイスを用いて、構造化に必要な発言のタイプと発言の開始・終了時間を会議中に簡単に入力できるようにしている. このようにして取得した発言のタイプおよび発言の開始・終了時間を用いて、一つの話題についての議論を、一つの導入発言をルートとし継続発言が連なる木構造として表現している. 我々はこの一つの木構造からなる議論を議論セグメントと呼んでいる. そしてこの議論セグメントによる議論の構造をミーティング中にも可視化することで、現在の議論の流れをリアルタイムに把握できるようにしている [8]. これにより、現在の議論の構造を把握したうえで発言が行えるため、より誤りの少ない

構造化が期待できる。さらに発言中における、発言タイプや、継続元発言などの変更を可能にすることで、たとえ発言開始時に発言タイプや継続元が間違っているとしても訂正することができ、発言者の意図をより正確に反映した構造化が可能となる。

しかし、従来のディスカッションマイニングにおける議論セグメントの構造化だけでは不十分な点が存在した。それは1つの議論セグメント内に複数の話題を含むことである。我々は過去の実験から議論セグメントの妥当性は評価しているが[8]、その実験から得られた知見と、普段のミーティングでの経験から、必ずしも1つの議論セグメント内で常に同じ話題について話されているとは限らないことが分かってきた。具体的な事例としては、議論が長くなるにつれより話題がより抽象的ものに遷移いたり、あるいは議論についていけなくなったため、同じ議論セグメント内で質問を行ったことで、新たな視点からの意見がなされたりと様々である。

このような問題が起こってしまう原因には以下の2つがあると考えられる。1つは行われている議論をより内容のあるものにするために、敢えて新たな話題を追加することがある点である。我々が議論を行う際、常に同じ話題について話していればよい意見が出るとは限らない。時には自分の体験をもとに新たな視点を導入したり、あるいはそれまでの議論によって気付いた問題点を指摘したりすることで様々な意見が出され、結果としてより発表者のためになる議論が行われることがある。そのため最終的に記録される議論セグメントとしては途中で話題が遷移しているものが出てきてしまう。もう1つは、現在使用している発言のタイプだけでは議論を柔軟な視点で議論を分割することができない点である。先の事例に挙げたように、同じ文脈で話されている議論内で話題が遷移したとき、それを取得して記録し、議論セグメントをさらに構造化する方法が無かった。

議論セグメントをさらに構造化することは、議論を理解するために過不足ない量の発言群を抽出することに役に立つと考えられる。我々はディスカッションブラウザによってミーティングでの議論を正確に見直すことはできる。しかし、一つのミーティングで行われる発言の数は非常に多く、「導入」「継続」による構造化だけでは1つの議論セグメントに含まれる発言が多くなってしまう。そのため、発言に対するメタデータが存在していても、閲覧者が自身にとって有用な発言を選定し閲覧することはコストがかかる。また、たとえ有用な発言を見つけられたとしても、その発言を理解するために必要な、過不足ない発言群を選定することは困難である。そのため「導入」「継続」による構造化に加え、議論セグメント内で話されている話題に着目した更なる構造化が必要となる。

そこで本研究ではミーティング中に行われる指示行為に着目した。我々はミーティング中、様々な指示行為を行っている。例えば「～の発言を受けてだが」といったように発言内容そのものを指示したり、「スライドのこの部分についてだが」といったようにスライド内のオブジェクト(スライドの内容)を指示したりする。特に後者のスライド内のオブジェクトに対する指示は、明確な指示対象がスライド

上に存在しているため、スライドのオブジェクトに対する指示を伴った発言は、そのオブジェクトに関する発言であると考えられる。そのため、同じオブジェクトを指示して行われた発言同士は、同じ話題について話されている可能性が高いと考えられる。ゆえに、オブジェクトに対する指示の情報を取得し記録することで、現在使用している議論セグメントをさらに話題ごとに構造化できるのではないかと考えた。

そこで我々は、ポインタリモコンを用いてスライド内オブジェクトの柔軟な指示を可能にし、それらの情報を取得し記録する仕組みを実現した。我々はこれまでにポインタリモコンによってスライド内のオブジェクトを指示する仕組みを提案してきた [9]。しかし、従来の方法では複数のスライドに跨るオブジェクトを同時に指示しながら比較したり、図の一部を指示して言及したりすることができない、といった問題があった。そこで本研究では、従来の機能をさらに拡張することで上記の問題を解決した。具体的にはスライド内のオブジェクトを、スライドショーの操作とは独立して移動や拡大が可能なオブジェクトとして切り出すようにした。これにより、異なるスライドに存在する複数のオブジェクトを同時に指示して比較する、といったことが可能となる。本研究では、スライド内のオブジェクトを1つの独立したオブジェクトとして切り出すことを独立化、そして独立化されたオブジェクトのことを独立エレメントと呼ぶ。そして上記の独立化の機能に加え、指示者が任意に指示対象を新たに定義して独立化させられる機能を実現した。これにより、図の一部などパワーポイントによって定義されていない領域でも指示することが可能となる。

また、このような機能を提供するとともに指示に関する様々なメタデータも記録している。ポインタリモコンによる指示の特徴は、指示対象の情報を正確に取得できることと、指示している状態を常在化できることである。従来のレーザーポインタや指示棒による指示行為から自動で指示対象を推定する研究は丸谷らによって既になされている [10]。しかし、やはり認識には誤りが存在し、指示対象を正確に取得することが困難であった。それに対しポインタリモコンによる指示では、パワーポイントによって生成されたオブジェクトの情報をを用いて指示対象を定義し、それを指示するようにしている。そのため、指示者の意図を正確に反映した指示が可能となる。このようにして正確に取得できるメタデータには、オブジェクトの座標情報だけでなく、指示者の Id や、指示対象のスライド内におけるオブジェクトの Id、といったものがある。加えて、ポインタリモコンによる指示では常にどこが今指示されているのかが表示されるため、指示の開始・終了時間を取得できる。従来のレーザーポインタなどを用いた指示行為は一時的なものが多く、指示行為は終わっているが、発言の内容としてはまだその指示対象について言及していることが多い。そのため、発言のどこまでが指示対象について言及しているかを取得することが困難であった。それに対しポインタリモコンでは指示の状態が常在化し、指示者が指示の状態を解除するまで指示行為を維持できる。これ

により明確な指示の開始・終了時間を取得することができる。

そして、このように取得したメタデータを用いて議論の構造化を行う。我々はまず同じ指示対象を指示して行われている発言は、同じ話題についての議論である可能性が高いと考えた。言い換えれば、指示対象が変わった、あるいは指示が終了したときには何らかの話題の遷移が行われていると考えることができる。そこで一つの議論セグメント内において、同じ指示対象を指示して行われている発言同士は、同じ話題について話している可能性が高いと判断し、それぞれの発言間に新たなリンク情報を付与する。こうすることで、1つの議論セグメント内でも同じ話題について話されているような、強いつながりを持つ発言群を判別することができる。本研究では、既存の議論構造に対して新たな意味を持つリンク情報を付与することで、議論セグメントをより詳細に構造化する。もちろん、指示が行われていない部分でも同じ話題について話されている部分は存在するだろう。しかし、それらの発言は何について話されているかが可視化されておらず、それゆえに話題の対象を定義することが困難である。そのため、正確な話題の対象を抽出し、より詳細に構造化することは困難であると判断し、本研究では構造化の対象としないことにする。

また我々は、異なる議論セグメント間であっても、同じ指示対象を持つ発言同士に新たなリンク情報を付与する。これは同じ対象について異なる文脈で話された議論同士を比較できるようにするためである。例えば、あるシステムの図を指示して、一方はシステムの機能拡張や機能改善の議論が、もう一方はそのシステムの目的や研究における位置づけに関する議論が行われたとする。これらの議論を比較することで、例えば研究の目的に、拡張する機能が適切に対応しているか、システムの目的に適したよりよい改善はないか、といったことを確認することができる。このように複数の議論を比較することは、新たな発見や理解を生み出す可能性がある。そのため我々は、比較する議論を選別するために異なる議論セグメント間であっても同じ指示対象を持つ発言群にはリンク情報を付与する。ただし、ここで付与するリンク情報は、同じ話題についての議論である、ということとは言えないため上記のリンク情報とは区別する。

そして本研究では、同じ指示対象を指示している発言は一貫して同じ話題について議論されている可能性が高い、という仮説を確かめるために、実際のミーティングのデータを用いた検証実験を行った。この検証実験により、上記の仮説の妥当性を評価することで、指示情報による構造化の妥当性を評価する。

そして我々はこのように取得した新たな議論の構造に加え、すでに記録されている発言に関するメタデータを用いてディスカッションブラウザの閲覧を支援する仕組みを提案する。具体的には、ミーティングで行われている議論をインタラクティブにフィルタリングし、ランキングする仕組みを実現した。従来のディスカッションブラウザでは前述の通り、1つの議論セグメントが長くなり、閲覧者が重要な発言やそれを理解するために過不足ない発言群を見つけることが困難だっ

た．そこで我々は新たに加えられた議論の構造をもとに，活性拡散アルゴリズムを用いて議論をフィルタリングしランキングすることで，発言がなされた文脈を理解するのに十分な発言群だけを抽出する．

以下に本論文の章構成について述べる．まず第2章で，ディスカッションマイニングによる議論内容の構造化と，記録した会議の様子を検索・閲覧を行うディスカッションブラウザについて述べる．次に第3章で，ポインタリモコンによる指示情報の取得と記録そしてそれらの情報を用いた議論の構造化について述べる．次に第4章で，第3章で行った構造化の妥当性を検証した実験について述べる．次に第5章で，取得した構造情報を用いた議論閲覧支援について述べる．そして第6章で，議論の構造化や指示に関する関連研究について，第7章で，まとめと今後の課題について述べる．

第2章 ディスカッションマイニング による議論内容の構造化と その利用

本章では、本研究において前提となっている、人間同士の知識交換の場であるミーティングにおける活動から、映像・音声情報やテキスト情報、メタデータなどの実世界情報を獲得し、それらを統合して構造化し、会議コンテンツと呼ばれる再利用可能なコンテンツを作成し利用する技術である、ディスカッションマイニングについて述べる。まず、実際の会議から半自動的に会議の様子をコンテンツとして記録するシステムであるディスカッションレコーダについて述べる。次に、獲得したコンテンツを検索・閲覧するためのシステムであるディスカッションブラウザについて述べる。最後に、これまでに行われていた議論の構造化の問題点について述べる。

2.1 ディスカッションレコーダによる議論内容の獲得と構造化

本節では初めに、実際に会議から半自動的に議論内容を獲得して構造化し記録するシステムであるディスカッションレコーダについて述べる。次にディスカッションレコーダによって取得された議論内容の構造化について述べる。

2.1.1 ディスカッションレコーダによる議論内容の獲得

ディスカッションレコーダは意思決定を目的とするのではなく、発表を主体とした議論を行う会議を対象としている。対象とする会議はモデレータとなる発表者、その発表を聴き意見を述べる参加者、そして会議で行われた発言を記録する書記がいる。会議のモデレータである発表者は、発表資料としてMicrosoft PowerPointによって作成されたスライドをプロジェクタで投影して発表を行うことを前提としている。

8 第2章 ディスカッションマイニングによる議論内容の構造化と その利用

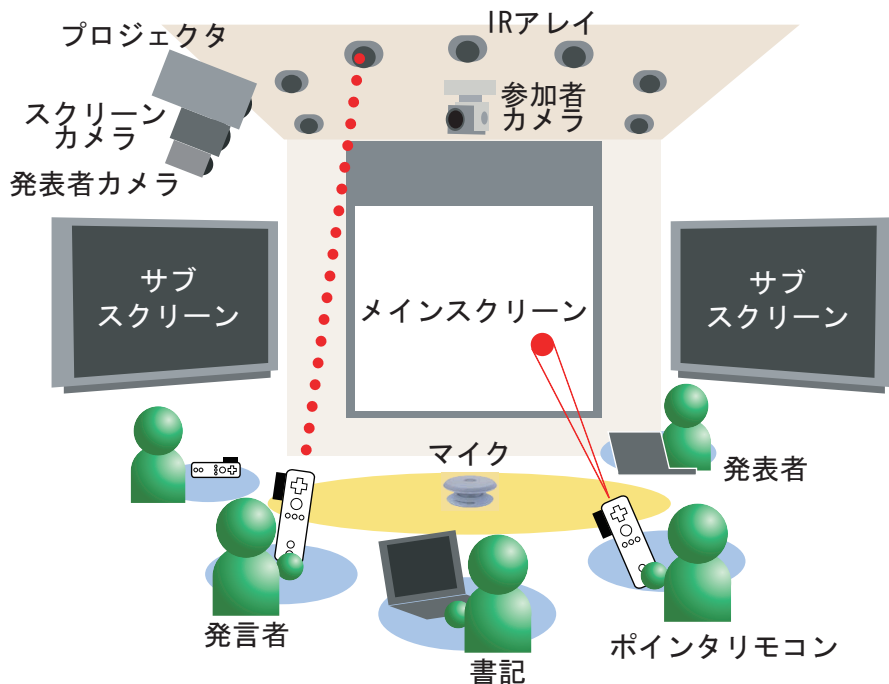


図 2.1: ディスカッションルーム

会議を行うミーティングルームは図 2.1 のような空間を想定している．まずこのミーティングルームには音声を記録するためのマイクが設置されている．加えて，ミーティングルーム内の参加者の様子を記録するためにパンチルトカメラ (参加者カメラ) が 1 台，スライドを投影するメインスクリーンを記録するために固定カメラ (スクリーンカメラ) が 1 台，そして発表者の様子を記録するために固定カメラ (発表者カメラ) が 1 台設置されている．これらの設備により会議中の音声や映像を正確に記録している．また，このディスカッションルームには，会議の参加者や会議の経過時間，あるいは後述する議論構造といった，会議を円滑に進行するための補助的な情報を提示するためのサブスクリーンが設置されている．このような設備により会議参加者は自身の行っている会議の状態を把握しながら議論を行うことができる．またディスカッションレコーダでは，カメラやマイク，サブスクリーンといった実際の会議の現場にある設備の他に，会議コンテンツの作成を行うためのサーバが用意されている．

このような設備の中で発表者は図 2.2 に示されるブラウザベースの専用ツールを用いてプレゼンテーションの操作を行う．まず，会議開始時に専用ツールから発表に用いるスライドファイルをサーバにアップロードする．アップロード時には発表者の氏名や所属するプロジェクト名，さらに発表のカテゴリ (サーベイレポートや研究進捗報告など) を選択する．アップロードが完了すると，会議が開始され開始時刻がサーバに送信される．発表者は専用ツール，あるいは後述するポインタリモコンを用いてスライド操作を行う．そして同時にスライドの切り替えタイ

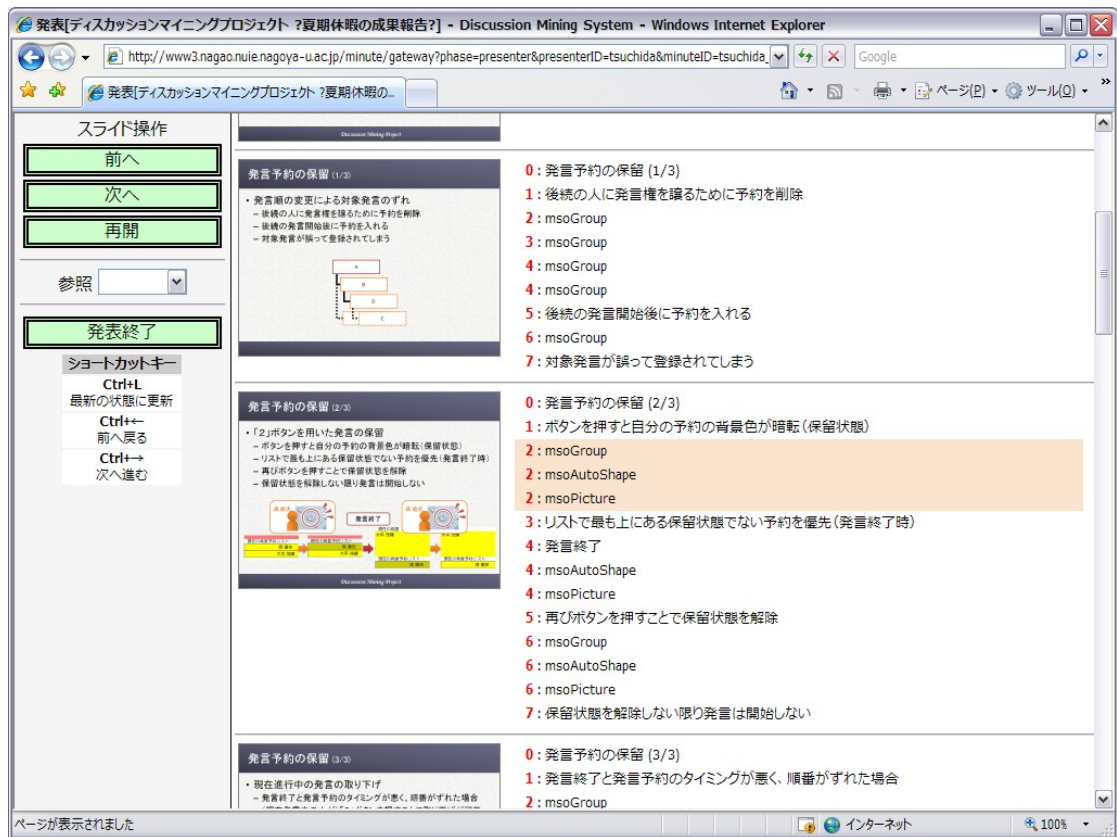


図 2.2: 発表者用インタフェース

ミングやスライドアニメーションの表示タイミングといった情報は随時サーバに送信され、記録される。また、スライド以外の資料 (例えばデモや Web の参照など) を用いてプレゼンテーションを進める場合には、資料を追加することもできる。そして会議終了時には、終了時間がサーバに送信される。

一方、会議中に行われる発言に関するメタデータを記録するために、我々は図 2.3 のようなポインタリモコンと呼ばれるデバイスを用いている。ポインタリモコンには赤外線を受信するためのデコーダが装着されている。そして受信する赤外線信号にはスクリーンからの識別子および座席の位置情報が含まれている。加えてポインタは常に Bluetooth を通じてリモコンに記憶されている参加者の ID やボタン情報を送信している。そして参加者は発言を行うとき、図 2.4 のようにポインタリモコンを上に掲げて発言を行う。このときポインタリモコンは天井の位置情報を表わす信号を赤外線 LED から受信する。そして受信した赤外線信号に含まれる位置情報をもとにカメラの向きを決定し、Bluetooth を通じて情報が送信される。そしてリモコンを掲げたときの角度 (ひねり) によって後述する発言タイプが決定する。サーバにはこれらの情報に加えて受信した時刻が、すなわち発言の開始時刻が送信・記録される。また、発言の終了時刻を記録する際にもポインタリ



図 2.3: ポインタリモコン



図 2.4: 発言時の様子

モコンを用いている。我々はこのようなデバイスを用いることで参加者にできるだけ負担をかけることなく様々なメタデータを取得している。またポインタリモコンには会議中の発表者や参加者の発言に対して、随時自分のスタンスを入力できるボタンがある。スタンスに応じたボタンを押下することでサーバに自分のスタンスの情報を送信することができる。具体的なスタンスには「賛成」と「反対」の二つがある。加えて会議終了後、自分が発言を見直すための目印として発言に対してマーキングを施す機能もある。このように発言に対する情報を付与することで、会議終了後に自分が議論を見直す際の重要な手掛かりを残すことができる。

そして会議中に行われた発言の内容は書記によって図 2.5 に示される Web ブラウザベースの専用のツールを用いて記録される。書記ツールには各発言を表わすノードが表示され、発言の開始に応じて随時追加されていく。そして書記はこのノードを選択し、テキストを入力することで発言内容を記録する。また、この書記ツールには、後述する、ポインタの指示対象のテキスト情報を書記テキスト内に自動的に挿入する機能もある。これにより本来は指示語などで代用されていた



図 2.5: 書記用インタフェース

テキストを簡単にエレメントのテキスト情報として挿入できるので、指示語が少なくなり、結果としてより議論の検索や閲覧に適した発言が記録されることが期待される。

2.1.2 ディスカッションレコーダによる議論内容の構造化

ディスカッションレコーダでは前節で述べたように発表を主体として議論を行うことを目的とした会議を対象としている。そのため2.2節で述べるようなインタフェースを用いて会議内容を閲覧する際、多くは会議で行われた議論を閲覧することが目的となる。議論閲覧の動機は、会議で重要だと思った議論を見直すことであったり、自分が参加していない会議で話された議論の内容を把握することであったりと様々である。しかしいずれの動機にせよ、その会議で行われた議論内容を理解することは必要不可欠である。ゆえに会議で行われた議論内容が理解しやすい形で記録されていることが望ましい。我々はそのための手法の一つは会議で行われた議論を構造化して記録することであると考えている。議論に含まれる発言に構造を付与し議論内容を構造化することで各発言間の関係が明らかになる。そうすることで、会議で行われた議論の文脈が明らかになり、結果として議論内容の理解を促す。

ではどのような構造化が望ましいだろうか。ディスカッションレコーダが対象としている会議にはいくつかの特徴があると考えられる。構造化を行う際にもこ

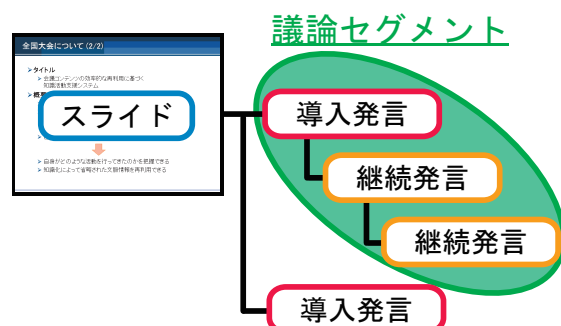


図 2.6: 議論セグメント

の特徴を考慮した構造化を行うことが望ましい。

一つ目の特徴は発言が話題ごとにある程度のまとまりを形成する点である。我々が議論を行う際、終始同じ話題について議論を行っているとは限らない。時には別の視点から新たな話題が提示されたり、同じ話題について議論を行っているつもりでも徐々に話題が推移していったりと様々である。そのため会議で行われている議論内容を理解するためには話題ごとに議論をセグメントすることが望ましい。

もう一つの特徴は個々の発言が互いに独立ではなく、複雑な依存関係を持っている点である。我々が発言を行う際、必ずしも直前に行われた発言を受けて意見を言うとは限らない。時には二つ前の発言を受けて行ったり、あるいは議題として提示された内容を受けて発言を行ったりと、様々である。そのため発言を記録する際、単純に発言を時系列順に記録するだけでは議論内容の理解が難しいため、発言の依存関係も記録することが望ましい。

これらの特徴を踏まえたうえで我々は、どの発言を受けて行った発言かという依存関係に基づいて発言を分類し、さらに発言間にリンク情報を付与することで、議論内容の構造化を行っている。具体的には、発言を「導入」と「継続」二つのタイプに分類し、各発言間にどの発言を受けて行われたかを表わすリンク情報を付与することで議論内容の構造化を行っている。こうすることで各発言がどのような経緯を経て行われたか、各議論がどのように進行していったか、といった文脈が明らかになり議論の理解が容易になる。

我々は各発言タイプを以下のように定義している。

- 導入：新たな話題の切り出しを行った発言のタイプ
- 継続：直前の（あるいはいくつか前の）発言を受けて行った発言のタイプ

このように発言のタイプを二つに限定することで、会議中、議論内容の構造化に対して思考を割くことなく、議論に集中することができる。そして継続発言と、その継続発言を誘発する元となった発言との間にリンク情報を付与して記録する。これを繰り返していくことによって図 2.6 のように、一つの導入発言をルートノー



図 2.7: 議論セグメントビュー

ドとし、継続発言がそれに連なるような木構造を形成する。そして会議が進み新たな導入発言が行われていくことで、先のような木構造をなす発言の集合が複数生成される。我々はこの一つの発言集合を議論セグメントと呼んでいる。そして一つの会議コンテンツ内に複数の議論セグメントが生成されることで、会議で行われた議論を話題単位で構造化して閲覧することが可能となる。

加えてディスカッションレコーダでは、より参加者の意図を反映した構造化を行うためにいくつかの機能を設けている。一つは発言の予約機能である。これは議論中にクロストークが起きないように参加者の発言の順番を制御するための仕組みである。例えば、ある参加者が発言を開始したとき、誰かの発言中だったとする。このとき発言中に開始しようとした発言は発言予約リストに追加され、直前の発言が終了すると自動的に発言権が移譲されるようになっている。こうすることでクロストークを防ぎ、かつ適切な順番で発言を順次行うことができる。

もう一つは継続発言の継続元を修正する機能である。これは会議中に、ポインタリモコンの操作によって自身の発言の継続元を変更することができる機能である。例えば実際の議論の場では上記の発言予約機能によって一つの発言が行われている最中に複数の発言が予約リストに追加されることがある。そして順次発言が行われていくが、すべての予約発言が最初に予約した時の発言に対する継続であるとは限らない。それは予約中に他の参加者の発言を聴くことによって自分の発言の内容が変わり、それに伴い継続すべき発言が変わる場合があるからである。このとき参加者は、発言開始時に自分が継続すべき発言へ継続元を修正することによってより参加者の意図を反映した構造化ができる。

最後の一つは発言の保留機能である。これは自分の発言の予約を一時的に取り

下げることができる機能である。上記の発言予約機能によって、予約した順に発言が行える。しかし実際の議論では必ずしもその順番で行うことが適切とは限らない。時には自分の発言権を譲ることでその場の議論がよりスムーズに行えたりする。そのため参加者が自ら自分の保留することによって、それを可能にしている。

そして自身の発言の状態や現在の議論セグメントの状態を会議中に知ることができるよう、会議中の様々な情報を表示するサブスクリーン上で、図2.7のように現在の議論の構造を可視化している。こうすることで単に自分の発言の状態を確認するだけでなく、議論構造を意識した発言が行えるようになる。

2.2 ディスカッションブラウザによる会議の閲覧

前節のディスカッションレコーダによって、発言内容や議論の構造といった議論に関する様々なメタデータ、そして議論時の音声や映像といった情報を取得できる。そして取得したメタデータを用いて効率的な議論内容の閲覧を実現することによって、議論内容のより深い理解に繋がり、知識活動の活性化を促すことが期待できる。

議論内容の正確な把握には映像・音声を含むマルチメディア情報を用いた閲覧が有効であると考えられる。そのため会議コンテンツの閲覧にもマルチメディア情報を用いる。ただし、マルチメディア情報には膨大な量の情報が含まれているため、閲覧目的に合わせて閲覧方法を提供する必要がある。しかし、会議コンテンツの閲覧目的は重要な発言の再確認や、自分が参加できなかった議論の内容把握など数多く存在する。そこで様々な閲覧目的を包括できるような仕組みを実現するために我々はディスカッションブラウザと呼ばれる、会議コンテンツを効率的に閲覧するためのシステムの研究・開発を行っている。

ディスカッションブラウザで会議コンテンツを閲覧するときは、まずシステムのトップページに表示される会議コンテンツの一覧を見て、その中から閲覧したい会議コンテンツを選択する。システムは要求された会議コンテンツをデータベースから取得し、閲覧インタフェースを提示する。

ディスカッションブラウザの閲覧インタフェースの全体を図2.8に示す。このディスカッションブラウザは以下に示すコンポーネントから構成されている。これらのコンポーネントがそれぞれ相互に連携しながら動作することによって議論内容の効率的な閲覧を実現している。以下でそれぞれのコンポーネントの詳細について述べる。

- ビデオビュー
- スライド一覧ビュー
- 層状シークバー

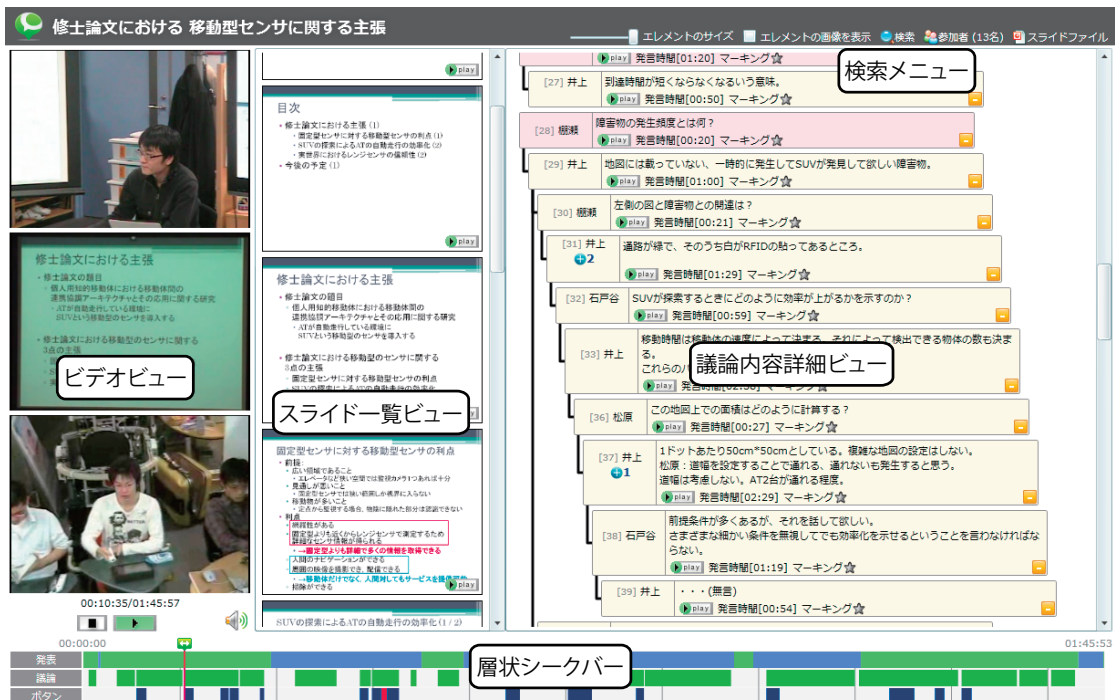


図 2.8: ディスカッションブラウザ

- 議論内容詳細ビュー
- 検索メニュー

ビデオビュー

議論中の様子を閲覧するためのビデオには、参加者ビデオ、発表者ビデオ、スクリーンビデオの3種類がある。ビデオビューではこれら3つのビデオの同時閲覧を実現している。このビデオビューは後述する層状シークバーや議論内容詳細ビュー上の操作と連動しており、発言単位やスライド操作単位でのスキップ再生を行うことができる。またビデオビューにおけるビデオの再生時間に応じて層状シークバーのスライダーや議論内容詳細ビューの表示が変化するため、議論全体の中でどの時間を閲覧しているかを確認できる。

スライド一覧ビュー

スライド一覧ビューでは会議に使用されたスライドサムネイルが表示される。スライドサムネイルはスライドの表示された順番にソートされ、また会議中に同じスライドが複数回表示された場合は、スライドサムネイルも複数枚表示される。そしてサムネイルの再生ボタンを押すことで、そのスライドが表示された時間から

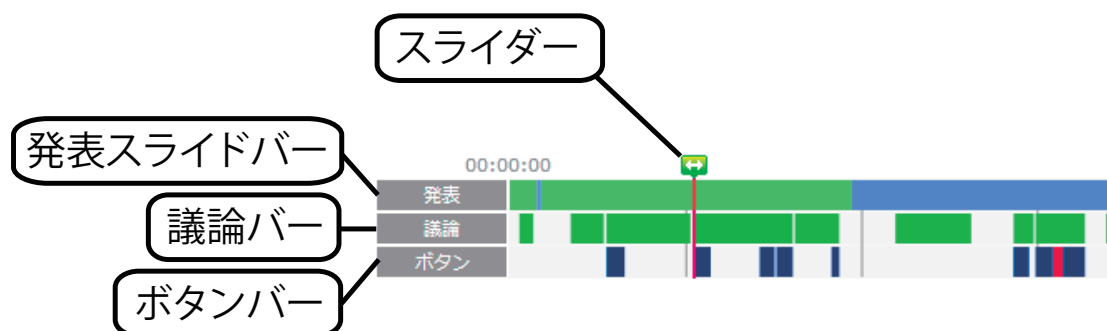


図 2.9: 層状シークバー

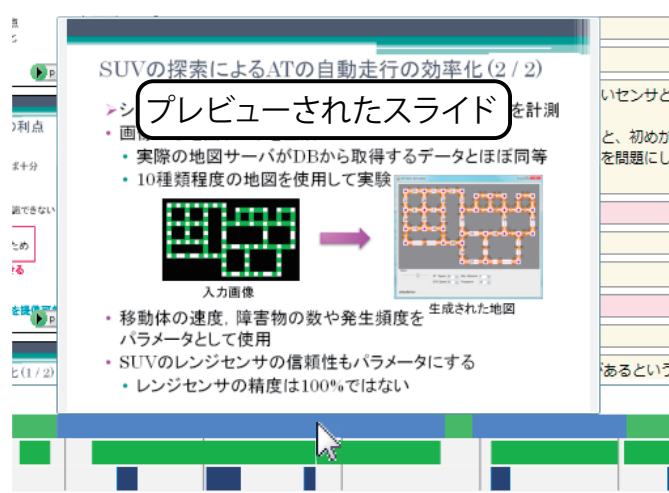


図 2.10: 発表スライドのプレビュー

ビデオを再生することができる。こうすることで発表者がスライドの説明を行っている部分にも容易にアクセスすることができる。また、スライドサムネイル自体をクリックすることで拡大されたスライドのイメージが表示され、スライドの詳細を閲覧することができる。

層状シークバー

ディスカッションレコーダで取得したメタデータは、層状シークバーと呼ばれる、図 2.9 に示すコンポーネント内のタイムライン上に表現される。この層状シークバーはスライダーと複数のバーによって構成される。

スライダーはビデオの再生時間に応じてタイムライン上を移動し、スライダーをドラックすることでその時間からビデオを再生することができる。これにより議論中の任意の位置にアクセスすることができる。

層状シークバーではディスカッションレコーダによって取得されたメタデータごとにタイムライン上にバーを作成することによって様々な視点から議論全体を

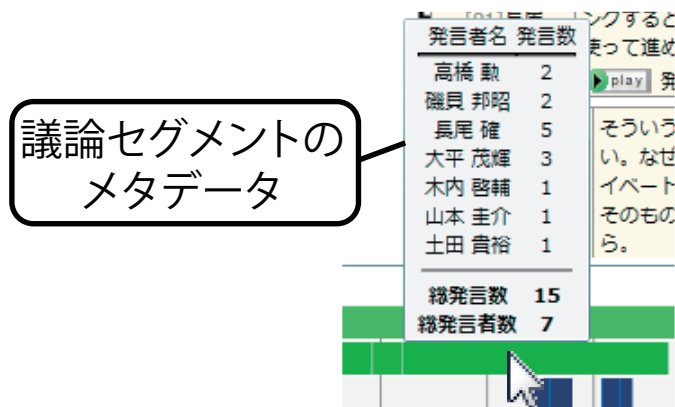


図 2.11: 議論セグメントのメタデータ表示

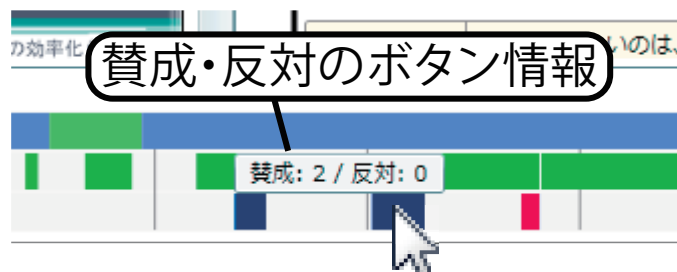


図 2.12: ボタン情報の表示

俯瞰することができる。例えば発表スライドバーは発表者のスライド操作を表現している。色の切り替わり目がスライドの切り替えを表しており、それぞれの箇所にマウスカursorを合わせることでそのときに表示されていたスライドを確認できる(図 2.10)。また、それぞれの箇所をクリックすることによって、該当するスライドが表示された時間からビデオを再生することができる。

会議中に行われた議論の情報は議論バーに表示される。議論バーは議論セグメントを単位とするアイテムから構成されており、それぞれの議論における単位時間当たりの発言数によって表示される色が異なる。単位時間当たりの発言数が多ければ色が濃くなり、逆に少なければ薄くなる。またそれぞれのアイテムにマウスカursorを合わせることで図 2.11 のようにその議論セグメント内で発言した発言者名や各発言者の発言回数などが表示される。そしてアイテムをクリックすることで、該当する議論セグメントの導入発言からビデオを再生することができる。これらの情報によって行われていた議論を大まかに把握することができる。

会議中に入力されたボタン情報はボタンバーに表示される。賛成と反対のボタン情報が存在する部分はそれぞれ青色と赤色に表示され、マウスカursorを合わせることで図 2.12 のようにボタン情報が表示される。この情報によってそこで行われていた議論が重要なものかどうか、見る必要があるかどうかといった判断を行うことができる。そして青色に表示されている部分ををクリックすることで、ボ

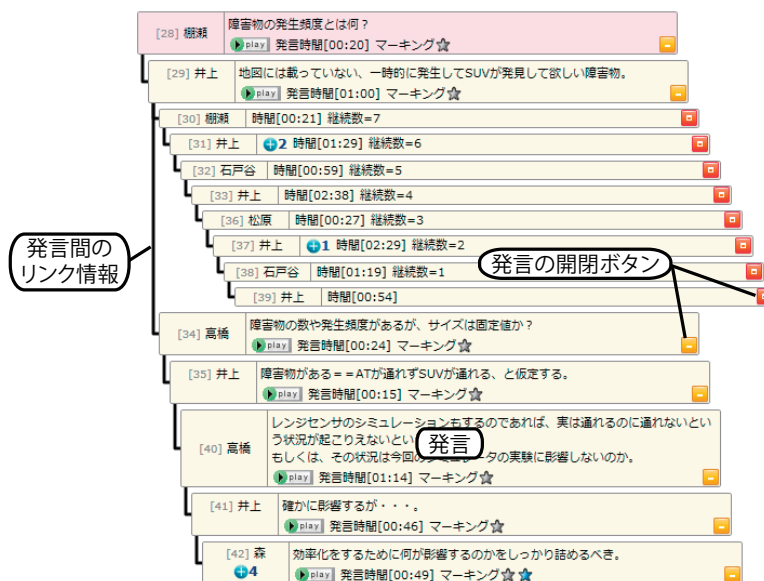


図 2.13: 議論内容詳細ビュー

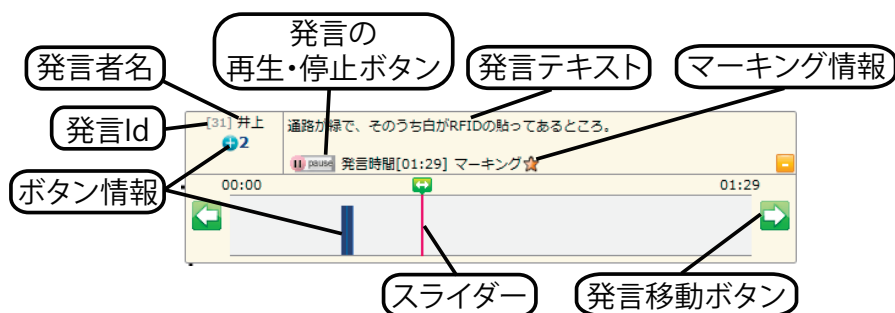


図 2.14: 発言のノード

タン情報が付与された発言からビデオを再生することができる。

議論内容詳細ビュー

層状シークバーによって様々な視点から議論全体の俯瞰を行った後は議論内容詳細ビューによってその詳細を閲覧することができる。議論内容詳細ビューでは図 2.13 のように取得した議論構造を保持したまま、各発言をノードとした木構造を成すように表示される。また、各発言のノードごとに発言の開閉ボタンが付与されている。このボタンをクリックすることで発言ノードの全表示・縮退表示の切り替えをすることができる。これにより閲覧者が見たい発言のみを選択して表示することができる。そのため発言間の関係を容易に把握することができ、単に発言がリスト状に表示されるだけでは理解が困難な議論の文脈の理解を促すことができる。

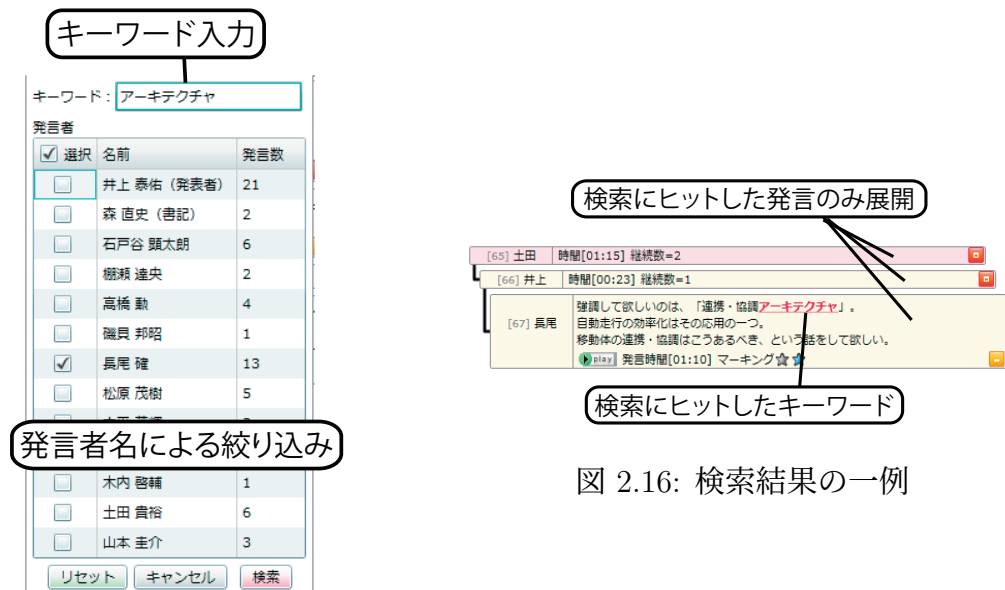


図 2.16: 検索結果の一例

図 2.15: 検索インターフェース

そして各発言のノードには図 2.14 のように書記が入力した発言テキストをはじめとして発言の ID, 発言者名, 発言時間, その発言に対して入力されたボタン情報などが含まれる。ユーザはこれらの情報をもとにビデオの閲覧を行う。また、ビデオの再生と同時に、層状シークバーのスライダーも自動的に移動するので、現在閲覧している箇所を容易に把握できるとともに、発言の部分に対しても、容易にアクセスすることができる。また、動画再生と同期してスライダーが移動するため、現在再生している発言の位置を知ることができる。そして再生されている発言中に押された、賛成や反対、マーキングといったボタン情報が表示される。これにより発言内容をより詳細に把握することができる。また、次の発言が存在する場合は、再生中の発言が終了すると、あるいは発言移動ボタンで次の発言に移動すると同時に自動的に次の発言が再生される。この機能により連続した発言の閲覧が可能となる。

検索メニュー

検索メニューは曖昧な手掛かりをもとに閲覧者とスライドや発言を結び付ける役割を果たす。同時に、閲覧者が検索結果に対してインタラクティブな操作を行うことで、閲覧目的を明確にし、より具体的な閲覧対象へと閲覧者を導く手助けを行う。

閲覧者は図 2.15 のようなインターフェースによるキーワードによる検索が行える。キーワードによる検索では、単純なキーワードマッチングだけでなく、発言者名による絞り込みも可能である。そして検索にヒットした発言以外は縮退表示になり、

かつ検索キーワードは図 2.16 のようにハイライトされる．これによりキーワードを用いて、閲覧したい発言を素早く探し出すことができる．

2.3 これまでの構造化の問題点

本章ではここまでに会議の様々なメタデータを記録するディスカッションレコーダと、会議の内容を効率的に閲覧するためのディスカッションブラウザについて述べた．しかし記録されたメタデータが正しくなければディスカッションブラウザで効率的な閲覧はできない．それに対し筆者の所属する研究室では過去に、システムによって取得しているメタデータの評価実験を行った [8]．本節では当時行われた実験と、そこから導き出されたこれまでの構造化の問題点について述べる．

ディスカッションレコーダでは第 2.1 節に述べたように「導入」と「継続」といった発言のタイプ分けや発言の予約機能によって議論構造を生成している．しかし実際の議論の場では、自分がこれから行う発言のタイプが導入なのか継続なのかを迷ったり、予約を入れるタイミングを逸して意図したリンク情報が付与されなかったりと、生成される議論構造には誤りが混在する．あるいは会議中は正しいと思っていてもディスカッションブラウザで自分の発言を客観視したときに、タイプや継続元が適切ではなかったと判断する場合がある．そのため、議論構造の妥当性の評価を行った．

当時の実験にはデータセットとして 2007 年度以降に作成された会議コンテンツにおける議論セグメントの中から、発言数が多い上位 18 個の議論セグメントを用いた．なお、18 個の議論セグメントに含まれる全発言数は 199 発言（うち継続発言は 181 発言）であり、議論セグメントごとの発言数の平均は 11.1 発言、議論セグメント内の発言者数の平均は 4.6 名であった．そして各項目に対して正解データを作成し、比較することで妥当性の評価を行った．

具体的な評価項目の中に継続発言の妥当性を検証するものがある．我々は経験的な問題として、議論が長くなるにつれ、話題が次第にずれていくということがしばしば発生することが分かっている．そのため、発言数の多い議論セグメントに含まれる個々の発言内容を吟味していくと、その議論セグメントの親である導入発言の内容や意図とは関連の弱い継続発言が含まれると考えられる．そのため、議論セグメントの中で新たな話題を提示している継続発言の数を調べることで、継続発言の妥当性を評価した．具体的には、まず各議論セグメントの導入発言者に、導入発言以降の発言に対して導入発言の内容や意図と関連が強いかわい、という判断をしてもらった．そして、その回答を正解データとし、導入発言と関連の強い発言の数を調べることで、継続発言の妥当性を評価した．

そしてディスカッションレコーダによって生成された議論構造と正解データの比較を行った結果を表 2.1 に示す．この結果から、ほぼすべての議論セグメントに新たな話題を提示している継続発言が存在する．つまり複数の話題が存在するこ

表 2.1: システムが生成した議論構造と正解データとの比較結果

ID	発言者数	総発言数	導入発言と関連が強いとされた発言数
1	6	14	1
2	5	13	1
3	5	13	1
4	4	10	2
5	4	13	2
6	4	8	1
7	4	7	1
8	4	6	1
9	5	17	1
10	4	11	1
11	4	11	1
12	3	5	0
13	5	16	6
14	5	13	1
15	4	14	1
16	7	11	2
17	5	9	1
18	5	8	0

とが分かる．新たな話題を提示する継続発言として最も多かったのが、直前までの議論から派生した話題を提示する発言であった．例えば、直前までは「意味単位を分割できるか」という議論を行っていたが、「逆に意味単位を統合できるか」と別の視点から話題を切り出す発言がこれに該当する．「意味単位」という大きな視点からとらえると一貫した議論を行っているが、より詳細な視点からとらえると別の話題について議論を行っている、という特徴がある．

以上のような実験結果と考察から、「1つの議論セグメント内でも話題が推移している」ことがわかる．加えてこの話題の推移は議論が長くなればなるほど起こりやすくなる．なぜなら議論が長くなるにつれ自身の発言を議論セグメントに対して巨視的に捉えることが難しくなり、結果として近視眼的に自身の発言の継続元を決めているからである．つまり、従来の構造化では長い議論セグメントを話題単位で詳しく分割することができず、長い議論セグメントを閲覧するためにはそれだけの時間を必要としてしまう、という問題があったと言える．

そのため、より効率的に議論内容を閲覧するためには、より細かいセグメント

情報を、現在使用している発言タイプとは別の新たな手法と組み合わせて取得することが必要となる。本研究では、発言中に行われる指示行為から取得できる指示情報を用いて新たな構造化を行う。この指示情報を用いて現在の議論セグメントに新たな構造情報を付加することで、より効率的な閲覧が可能な議論構造を生成する。

2.4 まとめ

本章では、人間同士の知識交換の場であるミーティングにおける活動から、映像・音声情報やテキスト情報、メタデータといった情報を獲得し、それらを統合して構造化し、会議コンテンツと呼ばれる再利用可能なコンテンツを作成し利用する技術である、ディスカッションマイニングについて述べた。まず、実際の会議から半自動的に会議の様子をコンテンツとして記録するシステムであるディスカッションレコーダについて述べた。次に、ディスカッションマイニングによって作成された会議コンテンツを検索・閲覧するためのシステムである、ディスカッションブラウザについて述べた。最後に、「1つの議論セグメント内で話題が推移している場合に、それを適切に構造化できない」という現在の構造化における問題点について述べた。

次章では現在の構造化における問題点への解決策として取り入れた手法である、発言中に行われる指示行為から取得できる指示情報を利用した構造化について述べる。

第3章 指示情報を用いた議論の構造化

前章ではこれまでのディスカッションマイニングで行われていた議論の構造化とその問題点について述べた。本章ではまず、会議中に行われる指示行為とその指示行為から取得できる情報を用いた議論の構造化について述べる。次に、ポインタリモコンによって取得している指示情報について述べる。最後に取得した指示情報を用いた議論セグメントの再構造化と議論セグメント間の関連付けについて述べる。

3.1 会議における指示行為と議論の構造化

我々は会議の中で発言時に様々な指示行為を行っている。例えば、直前の発言を受けて「さっきの内容についてだが～」といったように言葉によって対象を指示するものや、指示棒やレーザーポインタでスライド内の図を指しながら対象を指示するものなど様々である。

そこで本研究ではまず指示行為を「指示動作を伴う指示行為であるかどうか」という観点によって2種類に分類する。具体的には、前者の例で挙げたような発言の内容や発言そのものなどといった、指示対象が可視化されておらず指示棒やレーザーポインタなどで指示することが困難な指示行為を、「指示動作を伴わない指示行為」とする。一方で、後者の例のようなスライドの図や文章、あるいは会議中に描かれた図といったように、指示対象が可視化され指示棒やレーザーポインタなどで直接指示が可能なものを、指示動作を伴う指示行為とする。

そして本研究では、指示動作を伴う指示行為に着目する。具体的にはスライド内のオブジェクト、スライド内の文章およびその一部、そして次節で述べるポインタリモコンによって描かれた図に対する指示行為に着目する。

本研究でこの指示行為に着目する理由は二つある。一つは指示対象が可視化されているオブジェクトに限定することで、個々人の解釈の齟齬を減らすことができると思うからである。桑畑は発言の論旨を会議参加者全員で齟齬なく共有することの重要性とその困難さについて述べている [11]。特に言葉の定義といったような人やシチュエーションによって様々な解釈が可能なものに関しては誤解を生

みやすいと述べている。そのため、指示対象が可視化されておらず、誤解を生む可能性が高い指示行為に関しては、本研究の対象とせず、指示対象が可視化され、個々人の解釈の齟齬が少ないと考えられる指示行為を対象とする。

もう一つの理由は、我々の開発した仕組みを用いることで、詳細な指示対象の情報を取得することが可能であるとともに、指示行為そのものに関するメタデータの取得も可能だからである。従来の指示棒やレーザーポインタによる指示行為から、スライドのどのオブジェクトを指示しているのか、そこに含まれているテキストは何か、といったような詳細な指示情報を取得することは困難であった[10]。一方で我々は、前章で述べたディスカッションレコーダによって、会議で使用されたスライドに含まれるオブジェクトのIdや座標、インデント構造やテキスト情報など、指示対象に関する様々な情報を自動で取得している。これらの情報を用いることで指示対象に関する詳細なメタデータを自動的に取得することができる。またディスカッションマイニングにおける会議では参加者は必ずポインタリモコンを所持することが前提となっている。このポインタリモコンを発言の開始・終了をシステムに伝達するデバイスとしてだけではなく、スライド内のオブジェクトを指示するためのポインティングデバイスとしても用いることで、誰が、いつ指示を行ったか、といった指示行為そのものに関するメタデータも容易に取得することが可能である。このように指示対象と指示行為に関する様々なメタデータが取得できるため、正確な情報に基づいた応用が可能となる。

我々はその応用例の一つとして指示情報を用いた議論の構造化を行う。ここでの構造化とは、議論を話題単位でまとめることを意味している。

我々はこれまでに導入・継続という発言のタイプ分けによって会議での議論を話題ごとに分割し、一つの議論セグメントは一つの話題についての議論であるとしてきた。しかしその議論セグメント内にも細かく見ると話題の推移が行われており、複数の話題が含まれることは前章で述べた通りである。このような話題の推移は議論セグメントが長くなればなるほど、議論全体の把握が困難になるため起こりやすくなる。その一方で長い議論セグメントであるほど十分に議論された結果が発言として記録されているため重要な発言を含むことが多い。しかし議論セグメントが長く続く場合、話題が徐々に推移していることがあるため、どのようにしてその重要な発言がなされたのか、なぜその発言が重要なのか、といったことを理解するには長い議論セグメントを導入発言から順に見ていくしかない。なぜならその議論で行われていた発言の意味を理解するためにはその発言に関係する部分の文脈を理解する必要があるが、話題が徐々に推移しているため、どの発言は見なくてもいいのか、どの発言から見れば目的の発言の意味を理解できるかがわからないためである。そのため効率的に議論を閲覧するためには、重要な発言の意味を理解するために必要な議論はどこからか、言い換えるなら重要な発言と同じ話題で話されている議論はどこからかを知る必要がある。

そこで本研究では、同じ指示対象を指示して行われている発言同士は、同じ話

題について話している可能性が高いと判断し、同じ指示対象を指示して行われているそれぞれの発言間に新たなリンク情報を付与する。こうすることで1つの議論セグメント内でも同じ話題について話されているような、強いつながりを持つ発言群を選別することができる。このように本研究では、既存の議論構造に対して新たな意味を持つリンク情報を付与することで議論セグメントのより詳細な構造化を行う。

より詳細な構造化を行うことにより、例えばディスカッションブラウザで重要な議論を閲覧する際、議論セグメントすべてを見ることなく同じ話題について話されている発言だけを見ることが可能となる。そのため短時間で重要な議論の内容を把握することができるようになる。また、もちろん指示が行われていない部分でも同じ話題について話されている部分は存在するだろう。しかしそれらの発言は何について話されているかが可視化されておらず、それゆえに話題の対象を定義することが困難である。そのため正確な話題の対象を抽出し、議論の構造化を行うことは困難であると判断し本研究の対象外とする。

加えて異なる議論セグメント間であっても、同じ指示対象を持つ発言同士の間にもリンク情報を付与する。例えばあるシステムの図を指示して、一方の議論セグメントではシステムの機能拡張や機能改善の議論が、もう一方の議論セグメントではそのシステムの目的や研究における位置づけに関する議論が行われたとする。これらの議論セグメントを比較することで例えば研究の位置づけや目的に拡張する機能が合致しているか、システムの目的に合ったよりよい改善はないか、といったことを確認することができる。このように複数の議論を比較することは、新たな発見や理解を生み出す可能性がある。そのため我々は比較する議論を選別するために異なる議論セグメント間であっても同じ指示対象を持つ発言同士にはリンク情報を付与する。こうすることで同じ対象について異なる文脈で話された議論同士を比較できるようになる。これにより、例えば議論をディスカッションブラウザで閲覧する際、単一の議論セグメントを見るだけでなく、関連している異なる議論セグメントを閲覧者に提示することで複数の議論の内容を比較でき、新たな発見や理解を促すことができるだろう。

3.2 ポインタリモコンによる指示情報の取得

ポインタリモコンにはスライド内のオブジェクトの選択機能として以下の2つが存在する。

- 矩形選択 スクリーン上でマウ斯卡ーソルのドラッグ操作と同等の操作を行うことで矩形領域が表示され、その矩形領域に接触したスライド内のオブジェクトをすべて選択する機能。

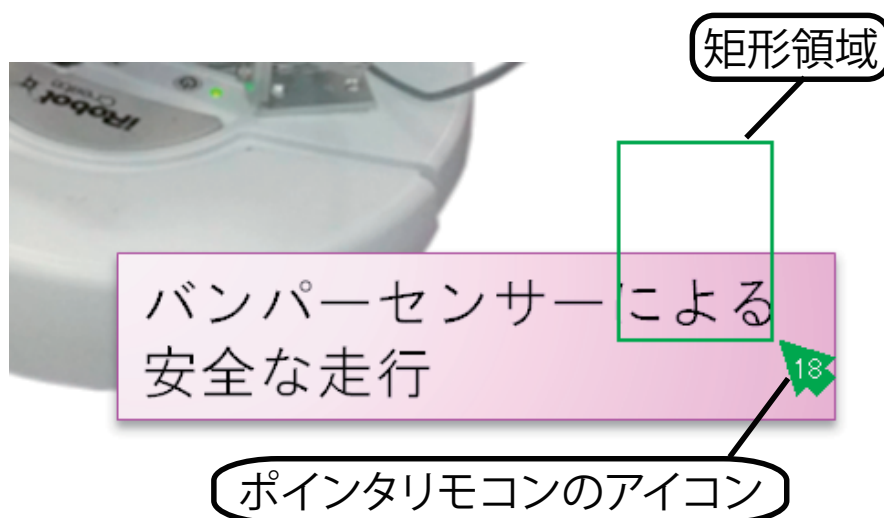


図 3.1: 矩形選択による指示

- 下線選択 スクリーン上の任意のテキストに対して、下線を引くことで指示をするための機能

本研究ではスライド内で指示したい対象に応じてこれらの機能を使い分けることで、スライド内のオブジェクトや、ストローク機能によって描かれた図、あるいはテキストの一部といった様々な指示対象への指示を可能にするとともに、指示対象の様々なメタデータを取得している。以下では上記した2つの機能についてそれぞれ詳細に述べる。

3.2.1 矩形選択

矩形選択機能は、図 3.1 に示すようにスクリーン上でマウスカーソル操作と同等の操作を行うことで矩形領域が表示され、その領域に接触するスライド内のオブジェクトを選択する機能である。スライド内のオブジェクト情報は、Microsoft の COM 技術 (Microsoft が提唱する分散コンポーネント技術) によって取得・解析した結果を利用しており、具体的なデータとしてはオブジェクトの Id やバウンディングボックス、オブジェクトに含まれるテキストやスライドショーでの表示タイミングといったものがある。また、この機能によって指示が可能なオブジェクトは以下の通りである。

- 箇条書きのテキスト
- グループ化された図形

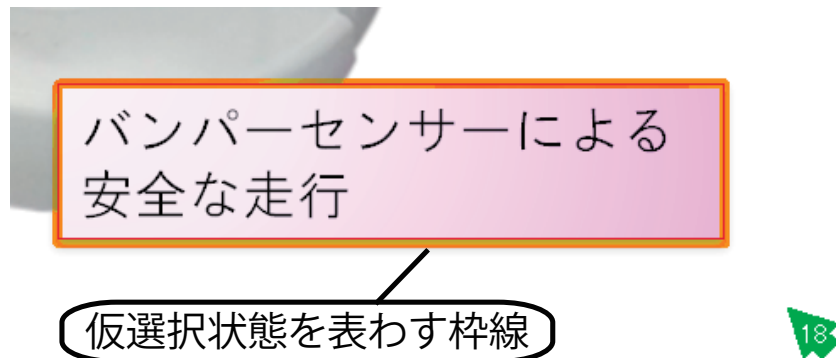


図 3.2: オブジェクトの仮選択

- JPEG や BMP などのイメージ
- オートシェイプ (Microsoft Office アプリケーションで利用できる, 多角形や楕円といった図形)
- 表・グラフ

そして, 矩形選択機能には上記のようなオブジェクトに対する正確な指示を可能にするために以下の5つの機能を有している.

- 仮選択 オブジェクトへの指示を一時的に保留しておく機能
- 可視化と常在化 今どのオブジェクトが誰によって指示されているかを常に表示する機能
- 独立化 指示対象をスライド操作に依存しないオブジェクトとして切り出し, オブジェクトの移動や拡大・縮小を可能にする機能
- トリミング スライド内の任意の領域を独立化させる機能
- ストローク機能と図形化 ポインタの軌跡を描画する機能とその軌跡を独立化する機能

以下ではこれら4つの機能について詳細に述べる.

仮選択機能

我々は矩形選択によって選択を行う際, 指示者の操作ミスにより意図していないオブジェクトが選択される場合がある. このような操作ミスによる情報はノイ

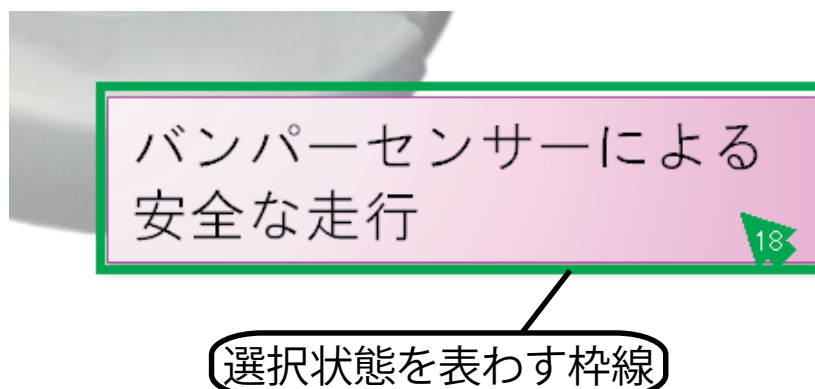


図 3.3: 指示状態の可視化と常在化

ズデータになってしまうため、正確な指示情報を取得するためにはノイズデータの作成を抑制する必要がある。そこで矩形選択機能では、矩形領域によって選択されたオブジェクトを仮選択と呼ばれる、指示の確定を一時的に保留状態にする機能を有している。仮選択の状態にあるオブジェクトは図3.2のようにオレンジ色の枠線で囲まれる。この状態のオブジェクトは指示されたとは判断されず、指示者は簡単な操作で仮選択の解除を行うことができる。また、指示者は仮選択されていないオブジェクトを矩形領域によって選択することで、新たに仮選択のオブジェクトを追加することもできる。このように指示を確定する前に本当に指示したいオブジェクトが選択されているかどうかを確認できるようにすることで、意図しない指示を未然に防ぐことができる。

可視化と常在化

指示棒やレーザポインタによって指示を行う際、指示行為自体は終了していたとしても、発言としてはその指示対象について言及している、というケースが多く存在する。そのため指示対象について言及している時間を正確に取得することが困難であったり、そもそも指示された対象がどれだったか忘れてしまい、指示対象を誤解してしまったりする可能性がある。そこで矩形選択機能では、指示されたオブジェクトが指示されている状態であることがわかるように指示の状態を可視化し、さらに指示行為が終了するまでその状態を保持している。具体的には指示されている状態にあるオブジェクトは図3.3のように色のついた枠線に囲まれた状態になる。この枠線の色は各ポインタリモコンが持つ固有の色と同じものである。そのため枠線の色によって自分が指示したオブジェクトであるかを容易に知ることができる。そして指示の状態はオブジェクトを指示している人が指示の状態を解除するまで続く。こうすることで、指示の開始・終了時間を正確に取得することができるだけでなく、誰が指示を行ったか、という情報も取得できる。加えて指示対象を可視化し常在化することで単純に指示の開始・終了時間や指示者



図 3.4: 独立化したオブジェクト

の情報を取得できるようになるだけでなく、指示対象を誤解して発言を行う、といったことも防ぐことができる。

独立化

矩形選択機能によって単にスライドに表示されているオブジェクトが選択できるだけでは柔軟な指示ができない。具体的には、例えば異なるスライドに存在するオブジェクトを同時に指示し比較する、といった行為ができない。そこで矩形選択機能では、指示されたオブジェクトをスライド操作に依存せず、移動や拡大が可能なオブジェクトとして切り出すことを可能にした(図3.4)。本研究では、このスライド内のオブジェクトを切り出すことを独立化、そして独立化されたオブジェクトのことを独立エレメントと呼ぶ。独立エレメントはスライド操作に依存していないため、複数のスライドからオブジェクトを独立化させ、二つを並べて指示する、スライドの一部を拡大して見やすくするといったことが可能となる。

我々は独立エレメントの情報を記録する際、矩形選択によって取得できる情報に加えて様々な情報を記録している。具体的には以下のようなものがある。

- 独立化させたオブジェクトが持つ情報（オブジェクトのIdやタイプ、バウンディングボックス、テキスト情報）
- 独立化させたオブジェクトの画像
- 独立エレメントの移動や拡大・縮小といった操作情報

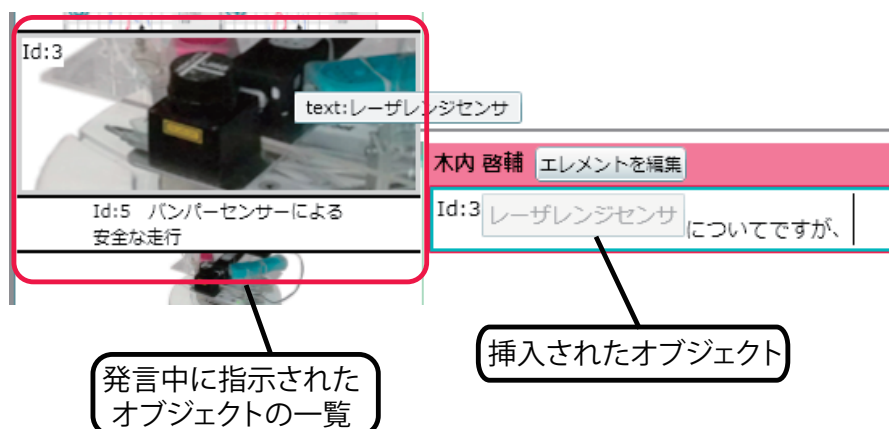


図 3.5: 書記インタフェースでのオブジェクトの挿入

- 独立エレメントに関連付けられたテキスト情報 (独立化させたオブジェクトに含まれるテキスト情報を含む)

上記の情報は基本的には独立化を行った際に、あるいは独立エレメントに対して操作を行った際に自動で取得される。ただし、イメージを独立化させた場合など、テキスト情報を持たないオブジェクトを独立化させた場合は、独立エレメントに関連付くテキスト情報が存在しない。そのため、例えば後で独立エレメントを閲覧する際、その独立エレメントが何を意味しているかが分からなかったり、あるいは独立エレメントをキーワードで検索する際、検索対象に含まれなかったりする問題が生じる。そこで我々は、前章で述べた書記インタフェース上で独立エレメントに対するテキスト情報の付与を可能にした。書記インタフェースでは発言が行われ、オブジェクトが独立化された場合、図 3.5 のように発言ノードの左側に指示されたオブジェクトが一覧表示される。書記はこの一覧から発言中に指示されたオブジェクトを選択することで、オブジェクトを書記テキスト中に直接埋め込むことができる。その際、インデント単位のテキストなどオブジェクト自体がテキスト情報を持つ場合は自動的にテキストが補完され、逆にイメージや後述するトリミングされた独立エレメントなどのテキスト情報を持たないものは書記がオブジェクトへのキャプションとしてテキスト情報を入力することができる。こうすることで独立エレメントに対してテキスト情報を関連付けるだけでなく、書記テキストとオブジェクトを関連付けることができる。

このように様々な情報を記録することで、例えば独立化させたオブジェクトの画像や操作情報を記録しておくことによりディスカッションブラウザなどで独立エレメントを表示する際に画像として表示したり、実際にどのように操作されたかを再現したり、あるいはキーワードによる検索を可能にしたりすることができる。

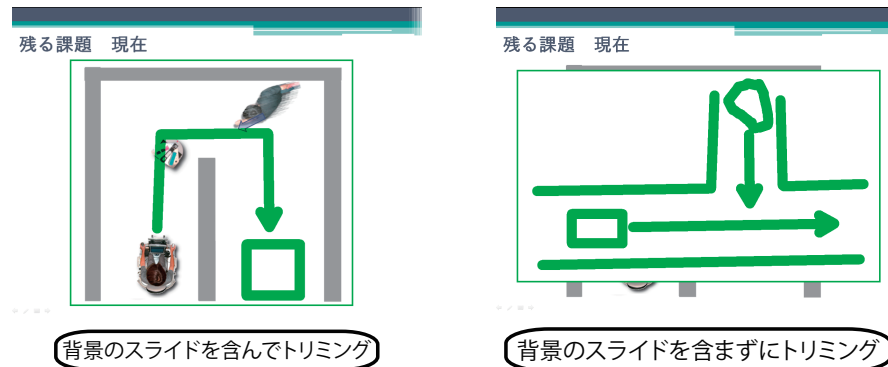


図 3.6: ストローク（緑色の線）の図形化

トリミング

矩形選択機能ではオブジェクト情報を取得するために、Microsoft の COM 技術によって取得・解析した結果を利用していることは先に述べた通りである。しかし、この解析の結果によって定義されたオブジェクトの単位が必ずしも指示したい対象そのものであるとは限らない。例えば、グラフの一部や、イメージの一部、あるいはグループ化されていない複数のオブジェクトを一つのオブジェクトとして指示したい場合など、柔軟な指示を行えない場合がある。そこで矩形選択機能ではトリミング機能と呼ばれる、指示者が自身が本当に指示したい領域を指定して新たなオブジェクトとして定義し独立化させる機能を設けている。これによりイメージの一部や複数のオブジェクトを独立化し、新たな独立エレメントとすることで柔軟な指示を可能にしている。しかし一方で、トリミングによる独立化は、解析によって取得し定義されたオブジェクトに依存せず行えてしまうゆえに問題もある。それは本来取得できるはずであったオブジェクトの Id やテキスト情報といったメタデータが失われてしまうことである。そこでこのトリミング機能で独立化を行った場合は、トリミングされた領域の包含関係を考慮して、どのオブジェクト内をトリミングしたか、そしてトリミングした領域内にどのオブジェクトが存在していたか、を記録している。これによりどのオブジェクトの一部を指示したのか、あるいはどのオブジェクトをまとめて指示したかったのか、という情報が記録できる。

ストローク機能と図形化

ストローク機能は、ポインタリモコンを用いてスライド上にポインタの軌跡を表示することで、図や手書きの文字を描画することができる機能である。パワーポイントの機能によって、スライドの操作者がマウスを操作して線を書くことは可能であるが、会議の場ではスライドの操作者である発表者だけが線を描きたいという要求があるとは限らない。例えば、会議の参加者が図を描くことで自分が

伝えたいことをより具体的に話したり、二つのオブジェクト間に矢印や直線を引くことで二つのオブジェクトの関係を示したりと用途は様々である。そこで参加者が持つポインタリモコンにスライド上にストロークを描く機能を設けることで、必要に応じてメインスクリーンに図を描けるようにした。

しかし一方で参加者がポインタリモコンを使って自由に描いたストロークを記録することは困難である。なぜならどのストロークが一つの図として描かれたものであるかを自動的に判断することが困難だからである。この問題を解決するために、ストローク機能によって描かれた図形に対するトリミングを可能にした。こうすることでトリミングされた領域が一つのオブジェクトとして新たに定義される。これにより、一連のストロークが一つの図形として描かれたものであると判断することができる。

また、ストロークを図形化する際に背景にあるスライド（の一部）を含めるかどうかの選択を可能にした。ストローク機能を使って描かれる図形には、スライド内のオブジェクトに対して線や図形を追加するため、背景に意味を持つものと、ストロークによって新たな図を描くため、背景に意味を持たないものの2種類が存在する。後者の場合、背景の情報とともに図形化させることは、かえってストロークによって描かれた図を見にくくする可能性がある。そのためストロークを図形化する際には、図 3.6 のように背景のスライドをそのままトリミングする方法と、背景を白くしてストロークだけをトリミングする二つの方法を用意している。これにより状況に応じて適宜見やすい状態でストロークを図形化することができる。

以上のような機能を有する矩形選択機能を用いることで、指示対象となったオブジェクトの Id やバウンディングボックス、あるいはどのオブジェクトの一部であったかといった様々な指示対象に関するメタデータ、そして誰が、いつ指示を開始しいつ指示を終了したか、といった指示に関するメタデータを柔軟に、かつ正確に記録することができる。

3.2.2 下線選択

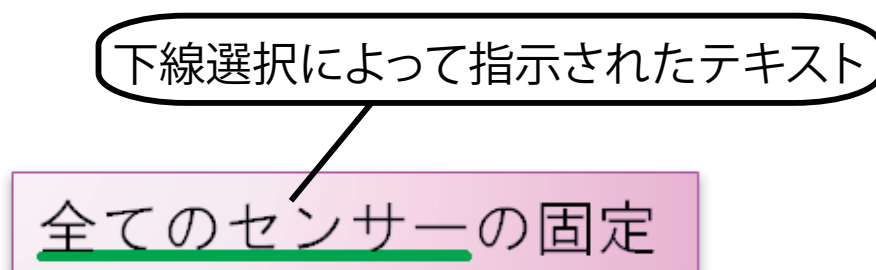


図 3.7: 下線選択による指示

下線選択機能は図 3.7 のようにスクリーン上の任意のテキストに対して下線を引くことで、テキストの一部を指示する機能である。インデント単位のテキストに対する指示は矩形選択機能によって実現されている。しかし文中に存在する単語について言及する場合のように、テキストの一部を指示したいケースが存在する。そこで下線選択機能では、参加者がテキスト内の指示したい部分をポインタリモコンのカーソルでなぞることで、図 3.7 のような下線を引くことができる。

ところで、レーザポインタなどでも文字列を正確になぞることは手が震えるなどして難しい。そのため下線選択では、指示する方法には単に指示したい部分をなぞるだけでなく、指示したいテキストの先頭を指定することで下線を引く方法もある。さらに一度下線が引かれた後でもポインタリモコンの操作によって下線部の範囲を変更することができる。こうすることで指示者はより簡単な操作で任意のテキストを正確に指示できる。

この下線選択ではスライドが表示されているスクリーンに対して OCR (Optical Character Recognition) を行った結果を利用している。ただし OCR による認識結果は誤りを含んでいるため、スライドテキストとの DP マッチング (動的計画法による文字列類似度計算) の結果を指示対象のテキストとしている。また、スクリーン上のスライドの表示が変更されたときに、自動的にスクリーンに対して OCR を実行し、スライド内の文字の座標を適宜抽出している。そのため、OCR により取得できるテキストであれば、それが図の中のテキストであったとしても、下線選択による選択が可能である。また、この下線選択機能による指示においても矩形選択機能と同様に、より誤りの少ない情報を取得するために仮選択機能を実現することで、正確な指示の開始・終了時間を取得し、さらに今指示されている箇所を常に確認できるようにするために指示状態の可視化と常在化を行っている。

3.3 指示情報を用いた議論セグメントの再構造化

我々は、3.2 節で述べたように、取得した指示情報をもとに議論セグメントを再構造化する。ここでの再構造化とは、同じオブジェクトを指示した発言間に新たなリンク情報を付与することである。例えば、図 3.8 において発言 Id が 3~6 の発言で同じオブジェクトに対して指示が行われたとする。このとき同じオブジェクトを指示している Id3~6 の発言それぞれに対して完全グラフをなすようなリンク情報を付与する。こうすることで、同じ指示対象を指示している（つまり、同じ話題について話されている可能性の高い）発言同士を関連付けている。これにより、3.1 節で述べたように、議論セグメント内で同じ話題で話されているような強いつながりを持つ発言群を選別できるため、結果として効率的な議論の閲覧を実現することができる。

ところで先のようなリンク情報を付与するためには Id3~6 の発言者全員が意識して同じオブジェクトを指示していることが前提となる。しかし、実際の会議の

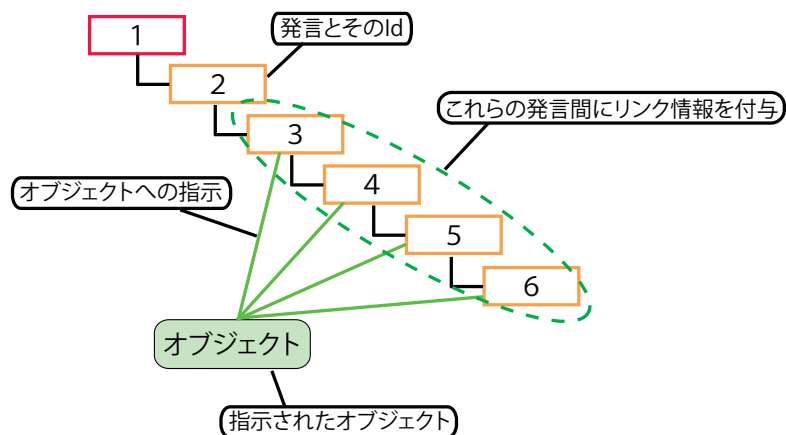


図 3.8: オブジェクトを介したリンク情報の付与

場で発言者が常に自分の発言は現在指示されているオブジェクトと関係があるかどうかを意識することは難しい。なぜなら、たとえ指示対象が可視化され常に確認できる状態であったとしても、発言を行う際は話すことに集中してしまうからである。もちろん導入発言などでオブジェクトを指示しながら質問をするといったように意識的に指示を行うケースも存在するだろう。しかし、議論が進むにつれ話すことに注力してしまい、自分が行っている発言が本当にそのとき指示されていたオブジェクトと関係があるかどうかという判断を忘れてしまう。

そこで図 3.9 のように、次の発言が開始すると同時に「現在の指示対象は発言に関係するかどうか」を Yes/No で問うようなダイアログを表示するようにした。発言者はポインタリモコンを操作し、自分の発言内容と関係があると判断し、指示対象を継続して指示する場合は Yes を、自分の発言内容とは関係がないと判断し、指示対象を参照しない場合は No を選択する。この機能により発言者に対して発言を開始する際、常にオブジェクトへの指示を継続するかどうか、という判断を促すことができる。また、発言が開始されてから Yes, No のいずれも選択せず一定時間経過すると、ダイアログがスライドの閲覧の邪魔にならないようスクリーンの右下に移動する。これにより、たとえ発言を開始した段階では話す内容が十分に整理されておらず、すぐに Yes, No のどちらを選択すべきかを判断できなくても、発言が終了するまで判断の猶予があるため自身の発言内容を吟味したうえで Yes, No の判断を行うことができる。

この機能により、指示者の意図をこれまでより正確に反映したリンク情報の付与を行うことができる。すなわち、指示者の意図を正確に反映した議論セグメントの構造化を行うことができ、結果として効率的な議論の閲覧を支援することができる。

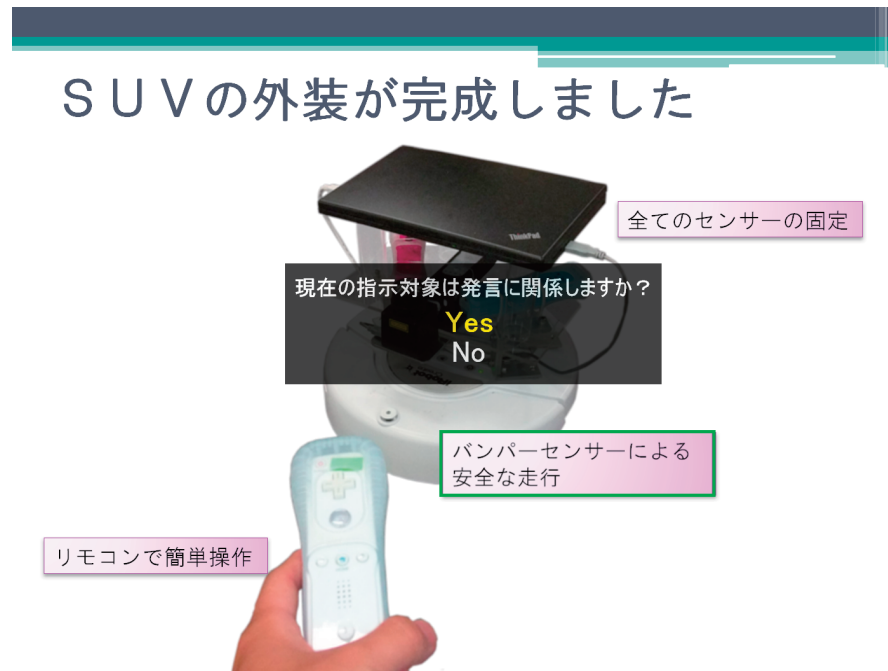


図 3.9: 指示の継承を決定するダイアログ

3.4 指示情報を用いた議論セグメント間の関連付け

本研究では、前節で述べたように同じ議論セグメント内の発言同士を関連付けるだけでなく、複数の議論セグメント間で同じオブジェクトを指示した発言同士も関連付ける。例えば図 3.10 において Id が 3～6 の発言と Id が 7～8 の発言が同じオブジェクトに対して指示を行っていたとする。このとき Id3～6 の発言と、Id7～8 の発言それぞれに前節で述べたようなリンク情報を付与するとともに、Id3～6 の発言群と Id7～8 の発言群間を関連付ける。こうすることで、第 3.1 節で述べたように、ディスカッションブラウザで閲覧する際、単一の議論セグメントを見るだけでなく、関連している他の議論セグメントを閲覧者に提示することで複数の議論の内容を比較でき、新たな発見や理解を促すことができる。

上記のような関連付けを行うためには、各々の議論セグメントで同じものが指示されていることが前提である。しかし、同じオブジェクトを指示したいと思うときに、必ずしもそのオブジェクトが簡単に指示できるとは限らない。例えば指示したいオブジェクトが存在するスライドとは異なるスライド上で議論していた時、まずスライドを切り替えてオブジェクトを独立化しなければならない。このようにオブジェクトを独立化するためだけにスライドを切り替えることは、議論の進行を阻害する可能性がある。なぜなら議論自体はスライドを切り替える前で行われているため、会議参加者は一時的に本来見えていたスライドが見えなくなってしまうからである。そのため現在のスライドを変えることなくかつ簡単な操作

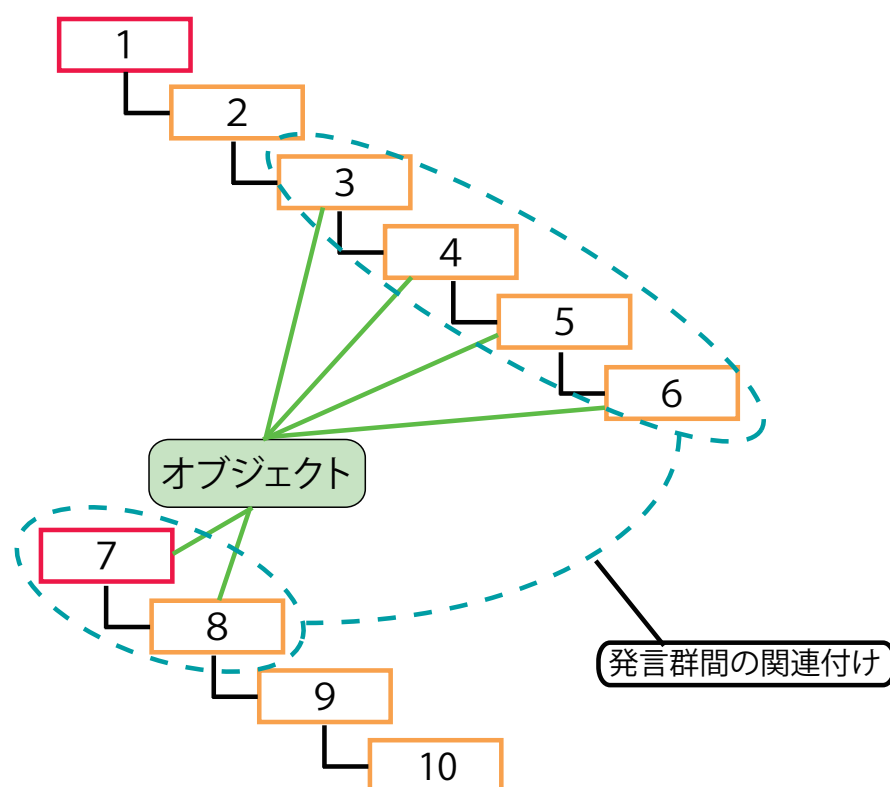


図 3.10: 議論セグメント間の関連付け

でオブジェクトを独立化できることが望ましい。

そこで本研究では矩形選択とは異なるポインタリモコンの操作によって、以前に独立化されたオブジェクトを再選択する機能を実現した。具体的には以下の二つの方法によって独立エレメントを再選択できる。これにより、たとえ異なるスライド上であってもオブジェクトを再選択して表示することが可能となる。

- ポインタリモコンのボタンを長押しすることで直前まで指示されていた独立エレメントを再選択
- スライドごとにリスト表示された独立エレメントの一覧（後述）からポインタリモコンを用いて選択することで独立エレメントを再選択

前者の方法は発言中にポインタリモコンのボタンを長押しすることで、直前に指示されていた独立エレメントを再選択できる機能である。異なる議論セグメント間で同じオブジェクトに対して発言を行っていた場合、多くの場合それらの議論セグメントは直近の議論セグメント同士である可能性が高い。なぜなら直近で行われているため記憶に残りやすく、その記憶をもとにして発言を行うことが容易だからである。そのため直近の議論セグメントで指示されていた独立エレメン

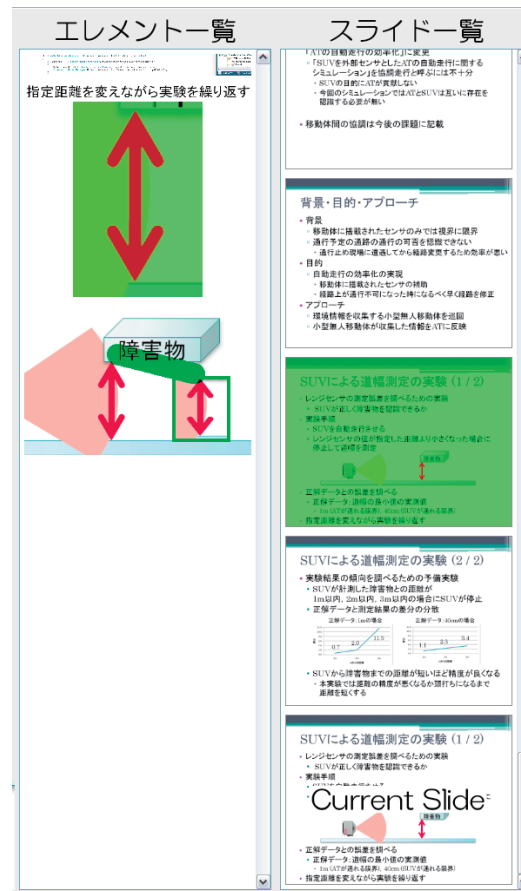


図 3.11: サブスクリーン上からの独立エレメントの再選択

トをポインタリモコンの長押しだけで再選択可能にすることで、矩形選択による選択の手間を省くとともに、同じ独立エレメントへの指示を可能にしている。

後者の機能は、発言中に図 3.11 のようにサブスクリーンに表示されている独立エレメントの一覧からポインタリモコンを用いて再選択したい独立エレメントを選択することで、その独立エレメントをメインスクリーン上で再選択できる機能である。発言者はサブスクリーン上から過去に独立化されたオブジェクトを選択したいと思った場合、まずスライド一覧から目的のオブジェクトが表示されていたスライドを選択する。そうすることで選択されたスライド上で独立化されたオブジェクトのみが一覧に表示されるようになる。その次に絞り込まれた独立エレメントの一覧から目的の独立エレメントを選択することで、メインスクリーン上に独立エレメントを表示することができる。このようにスライド、独立エレメントという順に候補を絞り込んでいくことで、効率的に目的の独立エレメントを発見することができる。

これら二つの方法によるオブジェクトの再選択を可能にすることで、スライドを切り替え、再び矩形選択によって選択するといった手間を省くことができる。こ

れにより、行っている議論を妨げることなく、異なる議論セグメント内で同じオブジェクトを指示している発言群同士を関連付けることができ、結果として議論を閲覧する際に関連する議論を提示するといった議論の閲覧支援が可能となる。

3.5 まとめ

本章では、まず3.1節で、前章で述べた「1つの議論セグメント内でも複数の話題について触れられてしまっているため、議論時間が長い議論セグメントでは効率的な議論内容の閲覧が困難である」という問題点の解決法として着目した、指示動作を伴う指示行為について、その指示行為を用いてどのように議論を構造化するかについて述べた。次に3.2節では、本研究で実際に取得している指示行為に関する指示情報について述べた。ここでは、ポインタリモコンによって提供されている矩形選択、トリミング、下線選択と呼ばれる3つの機能の特徴、そしてそれぞれどのような情報を取得しているかを述べた。そして3.3節では、長い議論セグメントを、閲覧したい話題の議論が行われている部分を選択的に閲覧するために、一つの議論セグメント内で同じオブジェクトを指示している発言は同じ話題について話されている可能性が高いと考え、同じ指示対象を参照している発言間にリンク情報を付与し、同じ話題である発言同士を関連付けることで、議論セグメント内に含まれる強いつながりを持つ発言の集合を判別する手法について述べた。最後の3.4節では異なる複数の議論セグメントの内容を比較することで、新たな発見や理解を促すことが期待できるため、異なる議論セグメント間であっても同じオブジェクトを指示している発言群同士を関連付けることについて述べた。

次章では「継続元の発言と同じ指示対象を参照した場合は、一貫して同じ話題について話しており、それ以外の話題について述べることは少ない」という仮説に対する検証実験について述べる。

第4章 議論の構造化に関する実験と考察

本章では、指示情報を用いた議論の構造化の妥当性を検証するために、ディスカッションレコーダを用いた研究室のゼミにおいて、参加者のポインタリモコンを用いて行った指示情報が議論の構造化にどのように貢献しているかを分析し、考察する。

4.1 分析・検証内容

本研究では、「同じ指示対象を参照している発言は、そうでない発言に比べて同じ話題について話している可能性が高い」という仮説の下に、発言のつながりに関する新たな情報を付与することで議論の構造化を行ってきた。これは同じ話題についての発言同士は、他の発言に比べてより強いつながりがあると考えられるためである。ここでのつながりの強さとは継続元の発言とどれだけ関連のある発言か、という指標である。

我々は発言を行う際、暗黙的に発言同士のつながりを意識している。例えば、導入発言にすべきか、継続発言にすべきか、という判断を行うことも発言同士のつながりを意識する行為の一つである。今行われている発言との関連性がないと判断した場合は導入発言として、関連性が強いと判断した場合は継続発言として発言を行っている。発言同士のつながりを意識する行為は、継続元の付け替え時にも起こっている。例えば、直近の発言よりも2つ前の発言の方がより関連していると判断した場合は、2つ前の発言へと継続元を変更したうえで発言を行っている。あるいは、自分とほぼ同時に他者が継続発言の予約を行い、他者の発言が先になされたとき、その発言の内容が自分の発言の内容と関連が強いと判断した場合は、予約を挙げた時に継続していた発言ではなく、その発言へと継続元を変更したうえで発言を行っている。このように継続発言を行う際にも、より関連の強い発言がないかを考えた上で発言を行っている。

このように発言同士のつながりを意識しているにもかかわらず、実際の議論セグメントではつながりが強い発言や、逆に繋がりが弱い発言が存在する。それは我々がつながりを意識するのはあくまで自身の発言と個々の発言との相対的な関

連でしかなく、議論セグメント全体で話されていること、すなわち議論セグメントで話されている話題との関連を考慮していないことが原因だと思われる。例えば、一つの発言内で継続元的话题に加えて別の話題を追加していたとする。その後の発言者が、新たに追加された話題について関連する発言を行ったとき、個々の発言同士を見れば同じ話題について発言しているように見えても、議論全体として見たときは話題が逸れていることになる。このような現象が繰り返されることによって、一つの議論セグメント内でも一貫して同じ話題について話している発言群に含まれる発言は互いに強いつながりを持ち、それ以外の発言群に含まれる発言は互いに弱いつながりを持つことになる。

そのため第3章では、同じ指示対象を参照している発言はそうでない発言に比べて同じ話題について話している可能性が高いと仮定して、議論セグメント内に含まれる強いつながりを持つ発言の集合を判別する手法を提案した。指示対象が明示されているときに新たな発言を行う際、常に、表示されている指示対象と関連する内容かどうかを考慮する必要がある。これによって、同じ指示対象を持つ発言群は関連の強い発言、すなわち一貫して同じ話題の発言が行われている傾向が強くなると考えられる。

しかし、同じ指示対象を参照していることと同じ話題が一貫して続くこととの依存関係はまだ十分に検証されていない。そこで本章ではその二つの事象の間に相関性があるかどうかを検証し、付与しているリンク情報が妥当であるかどうかを調べる。具体的には、「継続元の発言と同じ指示対象を参照した場合は、一貫して同じ話題について話しており、それ以外の話題について述べることは少ない」という仮説が成り立つかを検証した。

4.2 実験方法

前節で述べた内容を検証するために、以下の2つの情報を利用する。その後、それらの結果を分析することで仮説の妥当性を検証する。

- ディスカッションレコーダを用いた、ゼミ中に行われる指示情報
- ディスカッションブラウザ上での、各発言者に対してのアンケートによる、継続元発言と話題に関する情報

以下では各々の情報について詳細に述べる。

4.2.1 ディスカッションレコーダによる指示情報の記録

我々は図4.1のように普段行っているゼミでポインタリモコンを使用してもらい、その過程で指示情報を記録した。実験の被験者は教員が3名、学生が10名の

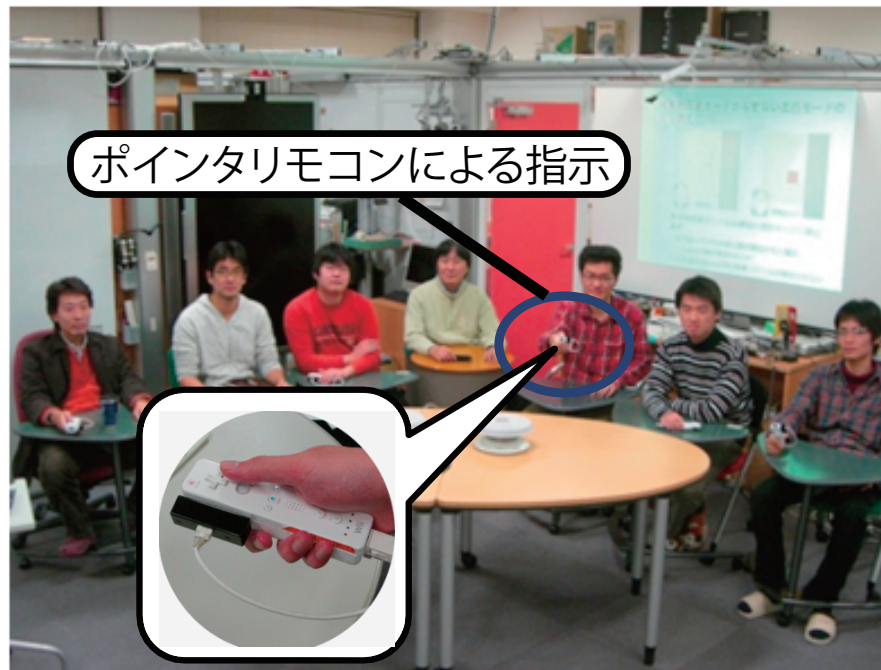


図 4.1: ゼミでポインタリモコンが使用されている様子

13名であり、学生のうちの一人に発表者として発表を行ってもらった。

被験者らには、自身の判断で発言とポインタリモコンによる指示行為を行ってもらった。ただし、スライド内のオブジェクトへの指示を行う場合は、必ずポインタリモコンを用いて指示を行ってもらった。その際、パラグラフや図全体を指示したい場合は矩形選択を、テキストの一部を指示したい場合は下線選択を、イメージの一部やポインタストロークで描かれた図といったような矩形選択や下線選択では指示できないものに対してはトリミング機能でスライドの一部を独立化して指示を行ってもらった。ただし、操作ミスによる指示者が意図していなかった指示行為は考慮しない。

以上のような環境で指示情報を記録した。今回記録した指示情報は3.2節で述べたものと同等である。その後、各発言を、指示行為と無関係な発言、指示を開始あるいは終了した発言、指示された指示対象を継承した発言の3種類に分類し、検証を行った。

4.2.2 アンケートによる発言と話題に関する情報の取得

次に前節のような環境で記録した会議コンテンツに含まれる発言の各発言者に対して、アンケートを行った。アンケートは図4.2のようにディスカッションブラウザ上で回答を行ってもらった。ただし、アンケートの回答を開始する前に、対象となる会議コンテンツの書記を行った人に、書記テキストに不備がないかを確

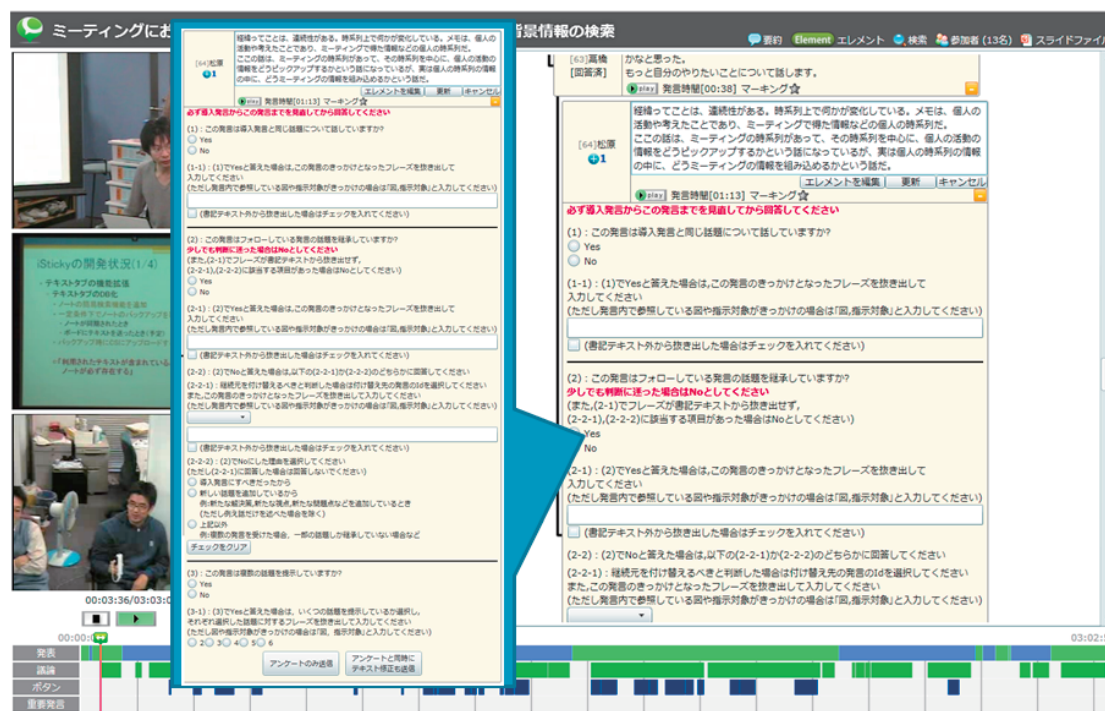


図 4.2: ディスカッションブラウザでのアンケート

認してもらった。これは、より正確な情報をもとにアンケートに回答してもらうためである。

アンケートに回答してもらった被験者は前節で述べた被験者と同じであり、実験対象となる会議コンテンツに含まれる自分の行ったすべての発言に対して、アンケートに回答してもらった。すべての発言に対して一人の人が同じ基準でアンケートに回答せず、各々が自分の発言に対して回答するようにした理由は、発言テキストに記録されなかった発言の意図を加味したうえでアンケートに回答してもらうためである。

また、アンケートに回答する際、必ず、該当する発言が属する議論セグメントの導入発言からその発言までに行われた発言すべてを見直してから回答してもらうようにした。これはそれまでに行われた議論の全容を把握してもらったうえでアンケートに回答してもらうためである。

実際のアンケート項目は大きく以下の3つの設問に分かれている。

- 導入発言との関係を問う設問
- 継続元発言との関係を問う設問
- 発言に含まれる話題の数を問う設問

以下では各設問について詳細に述べる。

(1) : この発言は導入発言と同じ話題について話していますか?

☐ Yes

☐ No

(1-1) : (1)でYesと答えた場合は、この発言のきっかけとなったフレーズを抜き出して入力してください
(ただし発言内で参照している図や指示対象がきっかけの場合は「図,指示対象」と入力してください)

☐ (書記テキスト外から抜き出した場合はチェックを入れてください)

図 4.3: 導入発言との関係を問うアンケート

導入発言との関係を問う設問

この設問は導入発言で提示された話題と同じ話題について発言しているかどうかを問う設問である(図 4.3). ディスカッションマイニングでは新しい話題を提示する場合は導入発言として発言をすることになっているが, 継続発言でも徐々に話題が推移していき, 気づけば導入発言で提示された話題とは異なる話題について話されている, という事象は多くある. そのため, 一つの議論セグメント内においてどの発言までは最低でも導入発言と同じ話題について話されているのかを調査した.

また, 導入発言と同じ話題について話していると判断した場合は, 設問(1-1)で発言のきっかけとなったフレーズを記入してもらうようにした. これには, 本当に同じ話題について話しているのかどうかを判断するための明確な指標とする, という意図がある. このアンケートは, 回答者の判断に依存する部分が多く, それゆえに個人の経験の差や考え方によって回答結果が大きく異なってしまうことが予測される. そのため, きっかけとなったフレーズを記入できるかどうかという指標をもって, できるだけ個人差を吸収するようにしている.

また, 記入するフレーズについては, 可能な限り書記テキストから抜き出して記入してもらうことにした. これは, 書記テキストは書記によって発言内容がある程度要約された形となっているので, 発言を代表することのないようなフレーズはあらかじめ省かれているだろう, という考えからである. 言い換えれば, 書記テキストには書かれておらず, 発言としては言っているものをフレーズとして選ぶ必要がある場合は, 話題が推移している, という判断を行うことができる.

しかし, 書記テキストの内容は書記個人の判断に委ねられているため, 本来重要だと思われるフレーズが記述されていない, という事態も発生する. そのため, アンケートでは, 「書記テキスト外から抜き出したかどうか」をチェックしてもらうようにした. これにより書記個人の判断によって記述されなかったフレーズなのかどうかを判断する.

加えて抜き出すフレーズには指示対象と, 指示はされていないが発言内で参照した図を選択できるようにした. これは指示行為を伴う発言では必ずしも発言で言われたことだけがきっかけとなるとは限らないためである. そのため指示対象

(2) : この発言はフォローしている発言の話題を継承していますか?
少しでも判断に迷った場合はNoとしてください
 (また,(2-1)でフレーズが書記テキストから抜き出せず,
 (2-2-1),(2-2-2)に該当する項目があった場合はNoとしてください)

☐ Yes
☐ No

(2-1) : (2)でYesと答えた場合は,この発言のきっかけとなったフレーズを抜き出して入力してください
 (ただし発言内で参照している図や指示対象がきっかけの場合は「図,指示対象」と入力してください)

☐ (書記テキスト外から抜き出した場合はチェックを入れてください)

(2-2) : (2)でNoと答えた場合は,以下の(2-2-1)か(2-2-2)のどちらかに回答してください
 (2-2-1) : 継続元を付け替えるべきと判断した場合は付け替え先の発言のIdを選択してください
 また,この発言のきっかけとなったフレーズを抜き出して入力してください
 (ただし発言内で参照している図や指示対象がきっかけの場合は「図,指示対象」と入力してください)

☐ (書記テキスト外から抜き出した場合はチェックを入れてください)

(2-2-2) : (2)でNoにした理由を選択してください
 (ただし(2-2-1)に回答した場合は回答しないでください)

☐ 導入発言にすべきだったから
☐ 新しい話題を追加しているから
 例:新たな解決策,新たな視点,新たな問題点などを追加しているとき
 (ただし例え話だけを述べた場合を除く)
☐ 上記以外
 例:複数の発言を受けた場合,一部の話題しか継承していない場合など

図 4.4: 継続元発言との関係を問うアンケート

や図が発言のきっかけとなった場合は,「指示対象, 図」と入力してもらうことで対処している。

なお, この設問は, 導入発言との関係を問う設問のため, 導入発言は対象としていない。

継続元発言との関係を問う設問

この設問は, 継続元発言の話題を継承しているかどうかを問う設問である (図 4.4)。この設問によって, 継続元発言と強いつながりにある発言かどうかを判断する。

まず設問 (2) で継続元発言の話題を継承して行っている発言かどうかを調査する。しかし前節でも触れたように, 同じ話題かどうか, 話題を継承しているかどうかといった判断は個人差に大きく依存することが予測される。そのため, 設問 (2) で話題を継承していると判断した場合は, 設問 (2-1) にて前節で挙げた方法と同様に, きっかけとなったフレーズを書記テキストから抜き出してもらうことで,

本当に話題を継承しているかどうかの確認を行ってもらっている。

一方、設問 (2) で話題を継承していないと判断した場合は、設問 (2-2) でその理由を調査する。まず設問 (2-2-1) では、単純に継続元が間違っていた場合であるかどうかを調査する。これは実際の会議中に、構造の付け替えを忘れた、付け替えるべきであることは分かっていたが明確にどこに付け替えればいいのかわからず付け替えなかった、といったような理由から継続元発言の付け替えが行われず、継続元発言の話題を継承していない場合である。アンケートでは、付け替えるべきだったとした発言の Id を選択してもらうとともに、発言のきっかけとなったフレーズを入力してもらっている。

設問 (2-2-2) では、継続元が間違っていたというケース以外で、継続元発言の話題を継承していないとした理由を調査する。アンケートでは「導入発言にすべきだったから」、「新しい話題を追加しているから」、「その他」という3つの選択肢から選ぶようにしている。1つ目の「導入発言にすべきだったから」に該当する発言は、継続元発言と関連はないと判断された発言となる。2つ目の「新しい話題を追加しているから」に該当する発言は、継続元発言と関連が弱いと判断された発言となる。我々が議論を行う際、新たな視点を導入することで考え方の幅を広げたり、新たな問題点や解決策を指摘したりする場合がある。このようなとき、個々の発言間のつながりだけを見れば正しく直前の発言を受けて発言がなされていることが分かる。しかし、議論全体として話題に一貫性があるかどうかを考慮すると、必ずしもそうであるとは言い難い。なぜなら、新たな視点を導入したり、新たな問題点や解決策を指摘したりすることによって、それ以降に続く継続発言で新たに追加された話題について継続することが可能となるためである。そのため、構造としては新たな話題が追加された発言の前後で異なる話題についての議論が行われることになる。それゆえ、新たな話題を追加している発言は継続元発言と弱い関係にあると判断する。3つ目の「その他」に属する発言には、複数の発言や話題を取りまとめた発言が存在する。このような発言の特徴には、継続元とすべき発言が複数存在することが挙げられる。しかし、このような発言を行うには、それまでの議論の流れや各発言の意図を理解する必要があるため、このような発言はあまり多くないと考えられる。そのため、今回のアンケートではその他として分類し、その他と回答された発言に対して個別に分析する程度にとどめる。

なお、この設問は継続元との関係を問う設問であり、導入発言は対象としていない。

発言に含まれる話題の数を問う設問

この設問は、回答の対象となる発言にいくつの話題が含まれているかを問う設問である (図 4.5)。ディスカッションマイニングでは、一つの発言では一つの話題に集中する方が望ましいとしているが、実際の会議の場で厳密にそれを実行する

(3) : この発言は複数の話題を提示していますか?

☐ Yes

☐ No

(3-1) : (3)でYesと答えた場合は、いくつの話題を提示しているか選択し、それぞれ選択した話題に対するフレーズを抜き出して入力してください
(ただし図や指示対象がきっかけの場合は「図, 指示対象」と入力してください)

☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

図 4.5: 発言に含まれる話題の数を問うアンケート

表 4.1: データセット

項目	数
会議コンテンツ数	14
発言数	1066
議論セグメント数	232
エレメント数	353
指示が行われた発言数	489
指示を継承した発言数	306
アンケートに回答された発言数	863

ことは難しい。それは、発言時に自分がいくつの話題を提示しているかを意識することが難しいという理由もあるが、必ず一つの発言では一つの話題しか提示してはいけなと制限してしまうことで、かえって議論そのものを抑制してしまう可能性があるためである。

それゆえに、複数の話題を持つ発言に対して行われた継続発言が、新たに追加された話題に対してのみ関連する発言を行う、といったケースが発生してくる。このようなとき、複数の話題を持つ発言の前後の発言では、異なる話題について話されている可能性が高いと考えられる。そのため、アンケートとしていくつの話題を提示しているかを回答してもらうことで、1つの話題を正しく継承しているのか(継続元発言と強いつながりにある)、あるいは追加された新たな話題を継承しているのか(継続元発言と弱いつながりにある)を判断する。

また、複数の話題が存在すると判断された発言に対しては、その話題を表すフレーズを抜き出してもらうことで、確かに複数の話題が存在していることを確認してもらうようにしている。

表 4.2: 継続元的话题を継承しているかどうかの割合 (%) [継続発言のみ]

	指示行為有		指示行為無	平均
	指示を継承	指示の開始 or 終了		
継続元的话题を継承：Yes	87.95	59.18	78.10	78.65
継続元的话题を継承：No	12.05	40.82	21.90	21.35
合計	100	100	100	100

4.3 結果と考察

今回の実験の対象とした会議コンテンツのデータセットは表 4.1 のようになっている．会議コンテンツ数は 14 で，その中で行われた発言は約 1000 発言，議論セグメント数は約 230 である．これらから，1 会議コンテンツあたり，約 16 の議論セグメントが存在し，発言は約 76 回行われていることが分かる．また，全発言のうち，指示が行われた発言数が約 490 であることから，ほぼ 2 発言に 1 発言は指示行為を伴っていることが分かる．仮説の妥当性が証明された場合，1 つの会議コンテンツに含まれる発言のうち約 3 割に当たる指示を継承している発言に対して，より強いつながりであるという情報を付与することが妥当である，という結論を得る．また，アンケートに回答された発言数は，全発言数の約 8 割であることから，仮説を検証するのに十分なデータが集まったと言える．

次に，指示行為と話題の継承の関係を表 4.2 に示す．この表では，指示の継承の有無と話題の継承の相関性の有無を検証している．指示を継承した場合は，指示行為を伴わない場合と比べ，話題を継承していないとされた割合が低くなっていることから，継続元の発言と同じ指示対象を参照することと，そこで行われた話題が一貫して続くこととの間に依存関係があることが分かる．つまり，指示行為が話題の継承に対して影響を与えていることが分かる．これは，指示対象が存在することで発言に対して何らかの制約が働いているからだと考えられる．我々は，指示対象が存在している時に発言をする場合，自身の発言と指示対象が関連しているかどうかを常に判断している．そのため，指示対象と関連しない継続発言を行う場合は，指示を終了するか，あるいはそのような発言を行わないかのどちらかを行うことが予測される．つまり，指示対象が存在することで，指示対象に関係する発言を行われやすくし，逆に指示対象とは関係のない発言を行われにくくしていると考えられる．

しかし，指示行為を継承しているにもかかわらず，話題を継承していないと判断された発言も少なからず存在している．そのうち，「導入発言にすべきだった」と判断された発言の多くは質問である傾向があることが分かった．特に指示行為

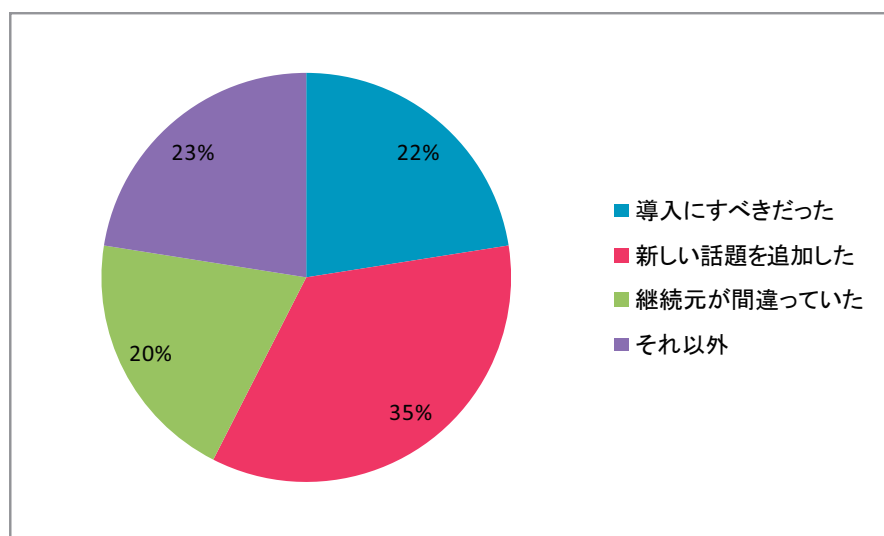


図 4.6: 指示の開始・終了を行った発言の話題を継承していないとされた理由

が伴っている場合は、抽象的な内容が書かれた部分を指示しつつ、具体的な内容について質問している傾向があることが分かった。例えば、「記録システムの運用開始」と書かれた部分を指示したうえで、「記録に用いるマイクの性能は十分なのか」、「記録に用いるカメラはどの位置に設置するのか」といったように、記録システムをより具体化したうえで質問を行っている。このように、指示行為を伴って質問が行われる場合には、今回付与するリンク情報とは異なった属性を持つリンク情報を付与できる可能性がある。例えば、先の例で挙げたように、抽象的な内容を含む指示対象が発言によって具体化されている場合には、指示対象と発言間に「指示対象をより具体的にした」という意味を持つリンク情報が付与できる。今回の指示情報を用いた手法では、リンク情報は付与できるが、抽象具体関係といったような、リンクに対する属性を付与することはできない。しかし、言語解析などの手法を用いることでリンクに対する属性を推定できるとしたら、新たな議論の構造化へとつながるだろう。

また、表 4.2 から指示を開始または終了した発言では話題を継承していない、とされた割合が最も大きくなっている。そのため、指示の開始・終了と話題が継承されないこととの間に依存関係があると考えられる。そこで指示の開始・終了を行った発言と、話題を継承していないとされた理由の内訳を詳細に分析した。その結果が図 4.6 である。図 4.6 の円グラフでは議論セグメントの途中で指示を開始あるいは終了した発言のうち、「導入発言にすべきだった」、「新しい話題を追加した」、「継続元が間違っていた」、「それ以外」と判断された発言の割合を示している。この円グラフから議論の途中で指示を開始、あるいは終了する場合は、新しい話題を追加する可能性が最も高いことが分かる。このように新しい話題が追加された発言で指示が開始・終了する理由は、それまでの議論の俎上に上がっていた話題

表 4.3: 導入発言と同じ話題かどうかの割合 (%) [継続発言のみ]

	指示行為有		指示行為無	平均
	指示を継承	指示の開始 or 終了		
導入発言と同じ話題：Yes	79.02	61.90	63.07	68.60
導入発言と同じ話題：No	20.98	38.10	36.93	31.40
合計	100	100	100	100

とは異なる話題が俎上に上がったために、指示対象もそれに応じて変わったからだ、と考えられる。例えば、システムの構成図が描かれているスライドで、図の一部を指して「このような機能がほしい」、「ここを改善してほしい」といったような議論が行われていたとする。そして、ある発言者が直前の指示を解除し、システムの構成図全体を指示したうえで「システム全体のつながりを考えるとその機能はこのような意味を持つ」という新たな意見を提示したとする。このとき話題の中心は、新たな意見が出されたことによって、システム全体に移り変わるとともに、指示対象もそれに応じてより適切なものへと推移していく。このように、我々は自分の意見をより相手にわかりやすく伝えるために自然とそれまでの指示を解除したり、あるいは新たに指示を開始したりする。このような行為に基づいて議論の構造化を行うことで、より発言者の意図に沿った議論の構造化が可能となるだろう。

次に、導入発言の話題との同一性についての関係を表 4.3 に示す。この表では 1 つの議論セグメント内で、導入発言で提示された話題と異なる議論がどの程度含まれているかを表わしている。この結果も、表 4.2 の結果と同様に、指示を継承している発言の方が、指示を行っていない発言と比べ、導入発言と同じ話題ではない、と回答された割合が低くなっている。このような結果になった理由としては、指示を開始した発言の多くが導入発言であった点であると考えられる。表 4.2 および図 4.6 の結果からもわかるように、指示が開始される場合は新たな話題を追加する場合が多い。そのため新たな話題を切り出す導入発言では必然的に新たな指示を行うことが多い。事実、指示を開始した発言の 64 % が導入発言であった。つまり、導入発言の話題からあまり逸れていない発言で指示が行われていたため、導入発言と同じ話題であると回答された割合が高くなっていると考えられる。さらに、議論の途中で新たな話題を追加するなど、直前の話題を継承しない場合は、指示の継承が行われないため、結果として、指示を継承している発言は導入発言と同じ話題である可能性が高い、という結果になったと考えられる。

最後に、複数の話題を提示することと指示行為との関係性を表 4.4 に示す。ま

表 4.4: 複数の話題を提示するかどうかの割合 (%)

	指示行為有		指示行為無	平均
	指示を継承	指示の開始 or 終了		
複数の話題を提示：No	91.07	93.87	88.87	90.38
複数の話題を提示：Yes	8.93	6.13	11.13	9.62
合計	100	100	100	100

ず、指示を継承した場合と指示行為が無い場合を比べたとき、複数の話題を提示するかどうかの割合に大きな差が見られなかった。この結果から、指示を継承することと、複数の話題を提示することとの間に特に依存関係は無いことが分かった。そのため、1つの発言内で複数の話題が提示されてしまうことで、同じ指示対象を参照していても一貫して同じ話題が話されない、という事態が起こる可能性がある。ところが、表 4.4 の複数の話題を提示する割合の平均が1割を下回っていることから、そもそも複数の話題を提示する発言自体があまり行われなことが分かった。そのため、1つの発言内で複数の話題が提示されることで話題の一貫性が損なわれる、という事態はあまり発生しないと言える。

以上のアンケートの分析結果を踏まえると、継続元の発言と同じ指示対象を参照している場合は、同じ話題について話されている可能性が高く、また1つの発言内で複数の話題が提示されることはあまりない、ということが言える。つまり「継続元の発言と同じ指示対象を参照した場合は、一貫して同じ話題について話しており、それ以外の話題について述べることは少ない」という仮説は妥当であると考えられる。

4.4 まとめ

本章では「同じ指示対象を参照している発言は、そうでない発言に比べて同じ話題について話している可能性が高い」という仮説の検証を行った。具体的な方法は、まずディスカッションレコーダを用いてゼミを行う際に、ポインタリモコンを使用してもらうことで、発言と指示行為の関係を記録した。次に、実験対象のゼミに参加した被験者全員に対し、ディスカッションブラウザ上で自身の発言に関するアンケートに回答してもらった。アンケートの設問は大きく分けて、導入発言との関係を問うもの、継続元発言との関係を問うもの、そして発言に含まれる話題の数を問うものの3つから構成されている。そして、そのアンケートの結果と、指示対象を継承しているかどうか、指示の開始・指示あるいはその継承の

終了が行われているかどうか，あるいは一つの発言内で複数の指示行為が行われているかどうか，を調べることによって，仮説が妥当であることを検証した．次章では指示情報と，実験によって妥当性が証明された，発言間のつながり情報の応用について述べる．

第5章 指示情報を用いた議論の閲覧支援

本章では、本研究の手法で取得した指示情報を用いた、ディスカッションブラウザにおける議論の閲覧支援について述べる。まず5.1節で、指示情報に基づく議論の構造情報を用いた発言群探索について述べる。次に5.2節で発言テキストと指示対象の同時閲覧による発言内容の理解支援について述べる。

5.1 指示情報に基づく議論の構造情報を用いた発言群探索

前述のディスカッションブラウザを用いて効率的に特定の発言を探し、閲覧する方法はいくつか存在する。例えば、特定のキーワードを入力して発言の絞り込みを行ったり、発言者名でフィルタリングを行ったりすることで、探し出し閲覧することができる。しかし、ディスカッションブラウザで、特定の発言と強いつながりのある発言群を効率的に探し出し閲覧することは十分には支援されていない。

例えば、議論セグメントの要約として導入発言と同じ話題について話されている発言群を探し出し閲覧する、といったことはできない。議論セグメントにおける導入発言とは、その議論セグメントで行われた議論の起点となる話題が含まれている発言である。もちろん、議論の過程で新たな話題が追加されることはあるが、導入発言で提示された話題と全く異なる議論が行われていること、つまり、後で導入発言にすべきだったと判断されるような継続発言が行われることは極めてまれであることは、前章の実験から明らかになった。そのため、導入発言でどのような話題が提示され、その話題についてどのような議論がなされたかを知るとは、その議論セグメント全体で大まかにどのような議論が展開されたかを把握する手掛かりとなる。このように、同じ話題について話している発言群を閲覧することは、個別の発言を独立に閲覧することよりも発言が行われた文脈を知ることができるため、より深い議論の理解につながると考えられる。

一方で、発言単体ではなく発言群を閲覧しようとしたとき、その発言群を適切に選択することは一般に困難であると考えられる。発言単体を探し出し、それだけを閲覧する場合ならばそれを閲覧するためにかかる時間もそれほど多くないだ

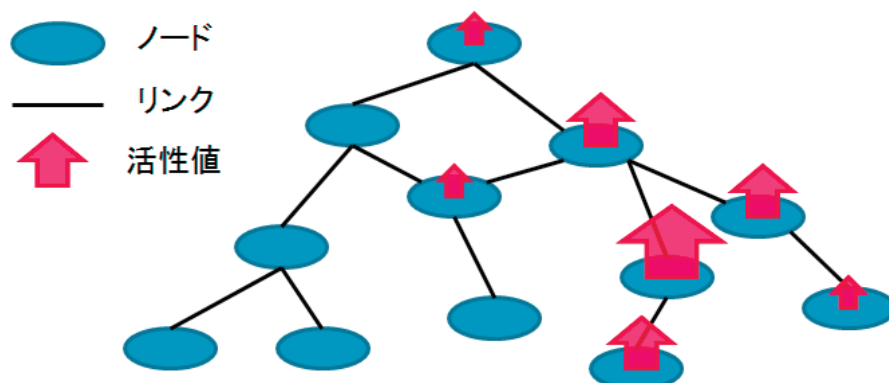


図 5.1: 活性拡散の概念図

ろう。しかし、複数の発言を見るためにはそれらの閲覧にかかるコストも多くなるため、閲覧者がどれだけの時間をかけて閲覧するのかを決めた上で、より目的に沿ったものから順に閲覧できる方が望ましい。しかし、実際には閲覧者が20分で閲覧すると決めても、いくつかの発言を見たら20分になるのかが分からず、気づいたら1時間以上経過していた、となってしまうことがある。そのため、それらの発言群を閲覧することで得られるメリットと、それらを閲覧するのにかかる時間的コストのバランスを考慮した上で、閲覧する時間が決定でき、かつその時間に沿って閲覧できる必要がある。

そこで本研究では、発言群を効率的に閲覧するために、これまでに取得してきた議論構造と、指示情報によって取得した新たなリンク情報を用いて、閲覧の目的に応じた発言群を選別する仕組みをディスカッションブラウザ上に実現した。具体的には、活性拡散アルゴリズムを用いた発言のランキングとフィルタリングを行う。

活性拡散とは1975年にAllan M. CollinsとElizabeth F. Loftusによって提唱された手法である[12]。この手法は、図5.1のようなノードとリンクで表されるネットワークが存在した時、ノードの活性化(活性値)がリンクを通じて近接するノードへ拡散する、というものである。つまり、活性値の高いノードとの距離が近いノードはそれらの活性値も高くなり、逆に距離が遠いノードは活性値も低くなる。この手法を適用することで、ネットワーク内にある全ノードに対する各ノードの相対的な活性値を求めることができる。

具体的な計算方法は以下の通りである。まず、ネットワーク内に存在するノードの数を n としたとき、各ノード間のリンクの有無を表わす行列 A を作成する。

$$A = \begin{pmatrix} a(1,1) & \cdots & a(1,n) \\ \vdots & \ddots & \vdots \\ a(n,1) & \cdots & a(n,n) \end{pmatrix} \quad (5.1)$$

ここで $a(i,j)$ の値は、ノード i とノード j の間にリンクが存在すれば1、存在しなければ0である。

その後、各ノードに与える活性値として以下に示すような行列 C を決定する。

$$C = \begin{pmatrix} c(1) \\ \vdots \\ c(n) \end{pmatrix} \quad (5.2)$$

ここで $c(i)$ の値は、ノード i に与えられる外部入力値である。

その後、以下の漸化式を解くことで t ステップ時におけるネットワーク内の活性値を求める。

$$X_{(0)} = C, X_{(t+1)} = A \cdot X_{(t)} + C \quad (t = 0, 1, 2, \dots) \quad (5.3)$$

$$X_{(t)} = \begin{pmatrix} x(t, 1) \\ \vdots \\ x(t, n) \end{pmatrix} \quad (5.4)$$

ここで $x(t,i)$ の値は、ステップ t におけるノード i の活性値である。

最後に $t \rightarrow \infty$ とすることで最終的なネットワーク内における各ノードの活性値を算出できる。また、 $t \rightarrow \infty$ としたとき、上記の漸化式は以下のように表わすことができる (ただし E は単位行列を表わす)。

$$X = A \cdot X + E \cdot C, \quad X = (E - A)^{-1} + E \cdot C \quad (5.5)$$

本研究では、この活性拡散アルゴリズムを議論構造に適用する。まず、ノードとして各発言を、リンクとして議論セグメントの構造情報を用いる。例えば、ある発言 i が別の継続発言 j によってフォローされていた場合、発言 i と発言 j の間にリンクを付与する。これを繰り返すことで議論セグメントごとに発言のネットワークが形成される。さらに各議論セグメントにおける導入発言を仮想的なノードとリンクさせる。こうすることで会議コンテンツにおけるすべての議論セグメントを1つのネットワーク構造で表現することができる。そして、このネットワーク構造に対して活性拡散アルゴリズムを用いることで、会議コンテンツ内にある全発言に対する各発言の相対的な活性値を求めることができる。

各ノードに与える外部入力値として、ディスカッションレコーダで取得した様々なメタデータを考慮した値を用いる。具体的なメタデータとしては、発言タイプ、

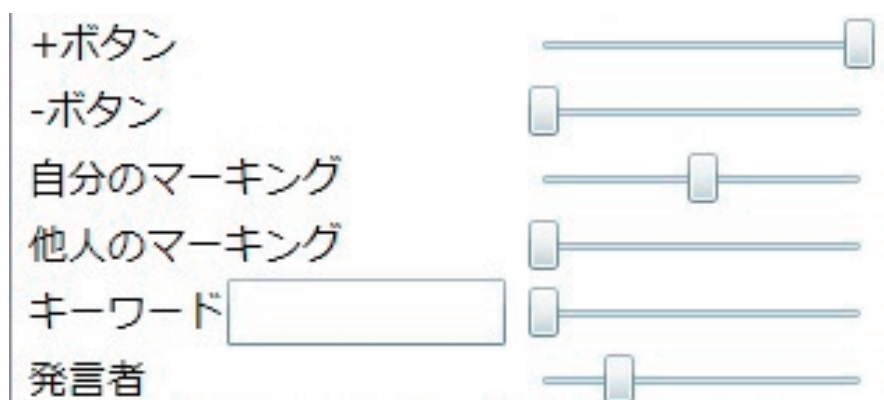


図 5.2: パラメータ入力のインターフェース

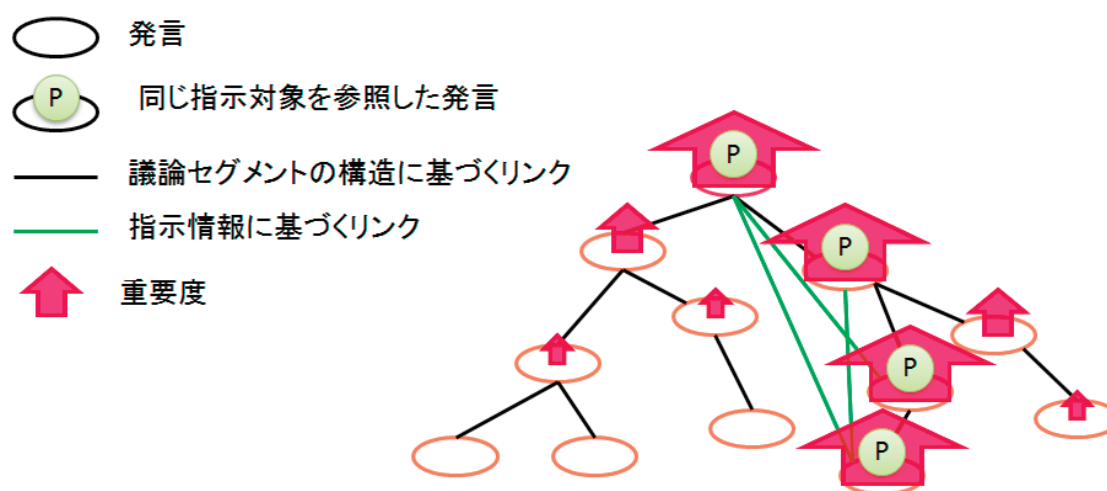


図 5.3: 指示情報に基づくリンクを含む発言のネットワーク

賛成・反対のボタン情報、マーキング情報、発言者名、オブジェクトに含まれるキーワード、といったものがある。

ただし、このうち、導入発言は議論セグメントを代表する発言であるため、他の継続発言より活性値が高くなるように設定してある。そして、閲覧者は各々のメタデータをどれだけ重要視するか、すなわち各パラメータの重要度を図5.2のようなインターフェース上から指定する。これにより、閲覧者は自身の閲覧目的にあった発言群を発見することができる。例えば、導入発言の重要度が高くなるようにすれば導入発言と同じ話題を持つ発言群を、賛成ボタン情報の重要度を高くすれば多くの賛成を得られた発言群をすばやく見つけることができる。

しかし、この手法にはまだ問題が残されている。活性拡散において、各発言の活性値は議論セグメントの構造によって取得されたリンクを通じて拡散される。そのため、たとえ同じ話題について話しているような強いつながりにある発言同士であっても、発言間の距離が長くなれば重要度が下がってしまう。同じ話題につ

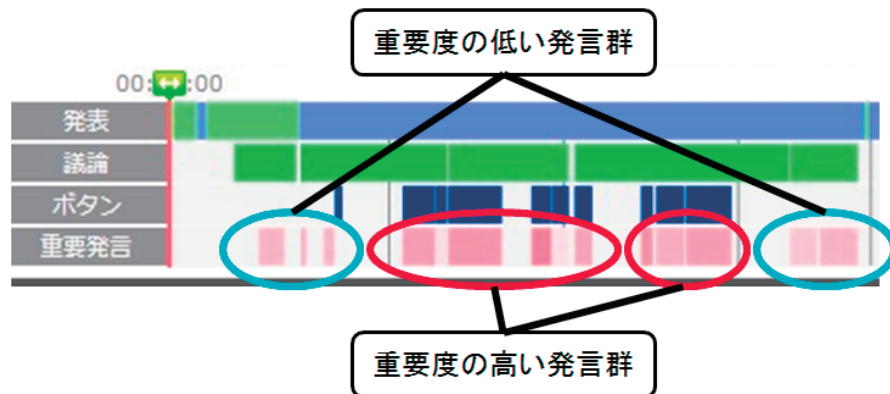


図 5.4: 層状シークバーでのランキング結果表示

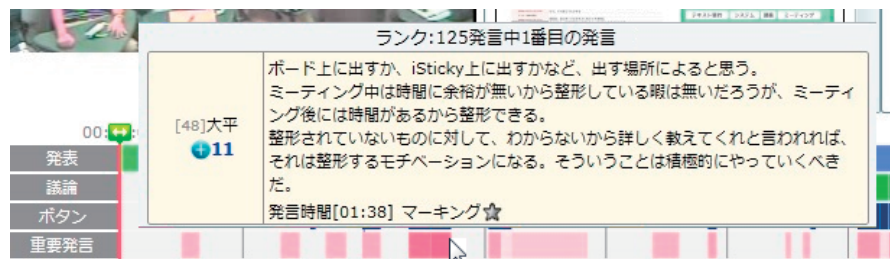


図 5.5: 発言内容とランキング結果の確認

いて話されている発言群は、発言間の距離に依存せず、同程度の活性値を持つことが望ましいと思われる。そこで、図 5.3 のように、前章で検証された指示情報によるリンクを、議論セグメントの構造をもとに生成されたネットワークに追加する。これにより強いつながりにある発言同士のノード間を直接連結することができる。そのため、強いつながりを持っている発言群の活性値をほぼ等しくすることができる。

このように会議コンテンツ内における全発言の相対的な発言の重要度を求めることで、会議コンテンツ全体からみると各発言がどの程度重要なかが決まる。すなわち、閲覧者の目的に沿って会議コンテンツ内の発言をランキングすることができる。そして、このランキングの結果は、図 5.4 のように層状シークバーに反映される。活性拡散アルゴリズムを適用した結果、重要度がより高いと判断された部分はより濃い赤色で表示され、重要度が下がるにつれ徐々に色が薄く表示される。これにより、会議コンテンツ全体から見てどのあたりで行われていた議論が重要とされたかを簡単に把握することができる。そして、重要度が反映された発言に対してマウスカーソルを置くことで、図 5.5 のように、その発言の詳細と、その発言のランキング結果を見ることができる。これにより、議論内容詳細ビューで発言を探すことなく、その発言の内容を知ることができ、その発言が全体から見て何番目に重要だと判断されたかを知ることができる。そして、閲覧したい発言を見つけた場合は、その部分をクリックすることで発言を再生することができる。

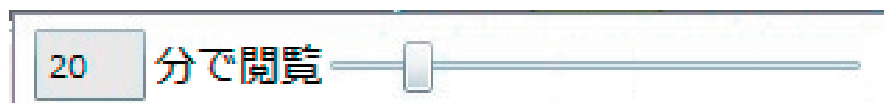


図 5.6: 閲覧時間指定インターフェース

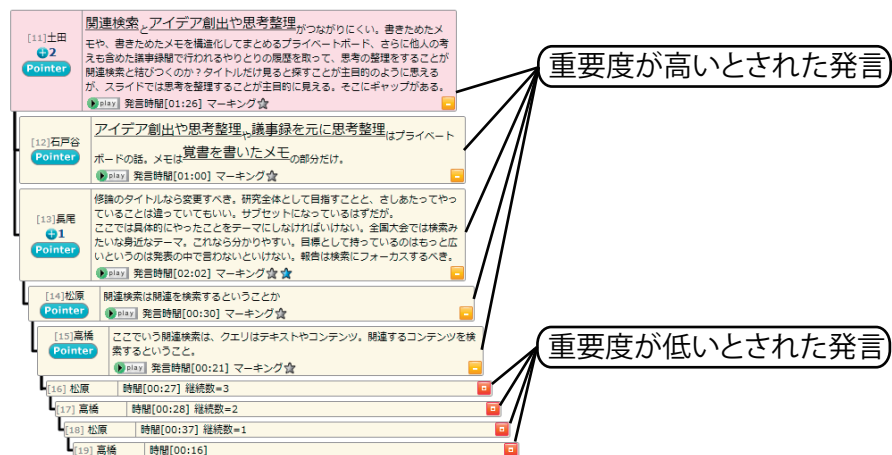


図 5.7: フィルタリングによる発言ノードの展開と縮退

また、重要な発言群が見つかったとしても、それらを全部閲覧した結果、どれだけの時間がかかるか分かりにくい。そこで、図 5.6 のようなインターフェースで閲覧したい時間を指定することで、ランキングされた発言を自動的にフィルタリングできるようにした。閲覧者はスライダーを操作することで、何分で発言を閲覧したいかを指定することができる。そして、指定された時間を閾値としてランキングされた全発言に対してフィルタリングを行う。具体的には、ランキングの高い順から発言時間を加算していき、どの発言で閾値とした時間を超えるかを計算する。そして、その計算の結果は図 5.7 のように議論詳細ビューに反映される。議論詳細ビュー上で、閾値を超過していない発言は展開され、逆に閾値を超過した発言は縮退されて表示される。その後、閲覧者が、展開された発言だけを閲覧していくことで、閲覧可能な重要度の高い発言群を自身が指定した時間内で優先して閲覧することができる。つまり、閲覧時に閲覧時間を制限して、効率的に発言群を閲覧できる。さらにこのフィルタリングの結果を層状シークバーにも反映し、閾値を超えた発言群に対しては重要度の色の濃淡を反映しないようにした。こうすることで、層状シークバー上で発言群を探す場合にも、効率良く行うことができる。

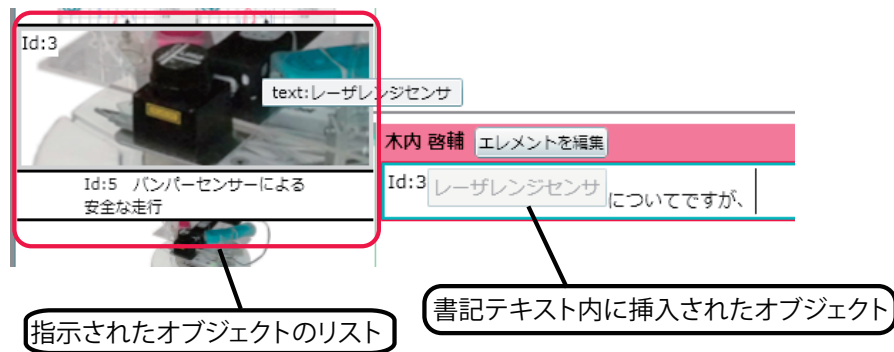


図 5.8: 書記インターフェースでの入力支援と指示情報の記録

5.2 発言と指示対象の同時閲覧

ディスカッションブラウザで会議コンテンツを閲覧する際、主に発言を閲覧することになる。その場合、どの発言が自分にとって有用なのかを判断するもっとも大きな手掛かりは、書記によって書かれたテキストである。書記によって発言内容が適切に要約され書き起こされていることで、発言を探す手掛かりとなるだけでなく、発言を再生して閲覧する際にもその発言内容の理解が促進されるだろう。

しかし、会議中に発言内容を適切に要約し書き起こすことは困難である。その困難さの要因の一つは、話し言葉特有の指示語を書き言葉にして入力することである。例えば、スライド内のパラグラフや図を指示して「この部分について～」や「この図について～」といったように指示行為を伴う発言が行われたとする。前者の場合、書記は指示されたパラグラフや図の内容を要約して入力する必要がある。このような手間は会議中に発言を要約して書き起こす際に大きな負担になると考えられ、その結果として書記テキストの質を下げることになると考えられる。また、後者の場合、書記はその図が何であるかを判断しテキストとして入力する。しかし、その際に入力されたテキストが、必ずしも十分に理解できる説明になっているとは限らない。時には、テキストで書かれているよりも図そのものを閲覧できた方が理解しやすい場合もある。いずれの場合にしても、指示対象を書き起こすことが困難であると判断し、そのまま「この部分」や「この図」と入力されることもありえる。このように入力された書記テキストは、ディスカッションブラウザで発言を閲覧する際に、発言内容の理解を妨げる。

そのため、会議中はできる限り書記の負担を軽減し、かつディスカッションブラウザでは発言内容が理解しやすい状態になっていることが望ましい。そこで、本研究では指示対象の参照情報を書記テキスト内に直接挿入できる仕組みを実現し、さらに挿入された指示対象の情報をディスカッションブラウザで適切に参照できる仕組みを実現した。

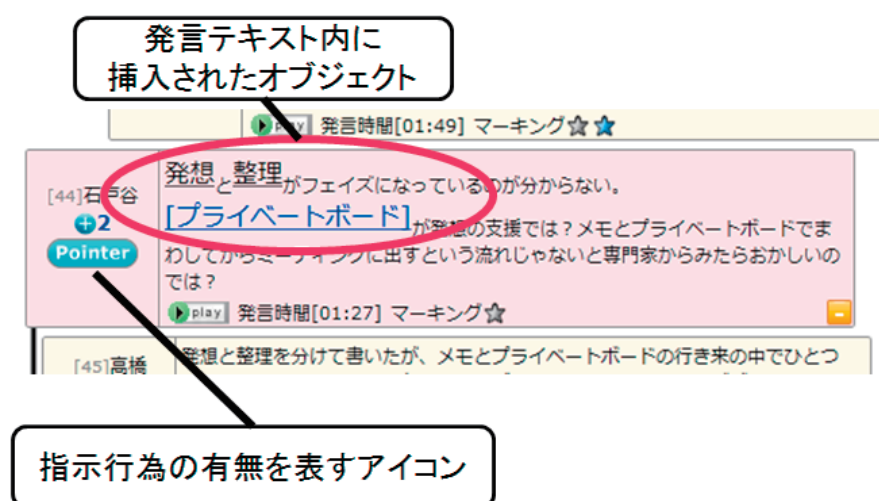


図 5.9: テキストでオブジェクトの情報を表示した発言ノード

参加者が会議中にポインタリモコンを用いた指示行為を行った場合、書記インタフェースでは、図 5.8 のように、指示されたオブジェクトの情報がリスト表示され、リスト内の項目をクリックすることで書記テキスト内に挿入できる。その際、指示されたオブジェクトが図やトリミングされた独立エレメントの場合は、画像としてリスト表示される。一方、下線選択によるテキストに対する指示や、矩形選択によるパラグラフに対する指示のように、指示対象にテキスト情報が含まれるものは、そのテキストが表示される。テキストが長すぎて表示しきれない場合でも、リストの項目に対してマウスオーバーすることで、そのテキストの全文が表示されるようになっている。このように、指示対象の種類によって適切な表示を行うことで、発言に挿入すべきオブジェクトが何であるかを分かりやすい形で表示している。

そして、挿入されたオブジェクトは書記テキスト内に埋め込まれ、図やトリミングされた独立エレメントといった、テキスト情報を持たないオブジェクトに対してはその場で書記がオブジェクトにテキスト情報を付与することによってテキスト情報を関連付けることができる。ただし、テキスト情報を持たないオブジェクトが挿入されたにもかかわらず、テキスト情報が入力されていない場合、書記テキストとして不自然な文章が構成されてしまう。そのため、単にテキスト情報を持たないオブジェクトに対してテキスト情報を関連付けるだけでなく、書記に対して「○○の図」といったように指示語を適切に書き起こすように促すことができる。またこのとき、他の発言内ですでに挿入された同一のオブジェクトが存在していた場合は、自動的にオブジェクトに関連付けられたテキストも置換される。一方でパラグラフや下線選択によって指示されたテキストの一部など、すでにテキスト情報をもつオブジェクトは、テキスト情報が関連付けられた状態で挿入される。こうすることで指示対象の内容を容易に入力することができる。



図 5.10: 画像でオブジェクトの情報を表示した発言ノード

このようにしてオブジェクトの情報が入力された書記テキストは、ディスカッションブラウザ上で図 5.9 のように表示される。その発言内で指示行為が行われた発言のノードには「Pointer」というアイコンが表示され、その発言で指示行為が行われたかどうかを知ることができる。そして、図やトリミングされた独立エレメントのような画像に関連付けられたテキストは青色で、パラグラフや下線選択によって指示されたテキストは黒色で、関連付けられたテキスト情報が表示される。こうすることで、どのオブジェクトが指示され、発言内で参照されたのかを大まかに知ることができる。加えて、挿入されたオブジェクトに対してマウスカーソルを合わせることで、図やトリミングされた独立エレメントの画像や、パラグラフや下線選択によって指示されたテキストを見ることができる。

ただし、書記テキストを閲覧する際、必ずしもすべてテキストとして書かれていることがその発言内容を理解するのに適しているとは限らない。例えば、スライド内にある図を指示して「この黒いデバイスは何ですか？」という質問が行われたとする。このとき、テキストとして「黒いデバイス」と書かれているよりも、実際にそのデバイスが画像として閲覧できる方が望ましい。そこで本研究では、図 5.10 のように書記テキスト内に画像を表示できるようにした。これにより、発言内で指示されているものが何なのかを直感的に知ることができる。こうすることで、例えば、指示された図は覚えているがそれがどの議論で行われたか、何と呼んでいたかを正確に覚えていない時にでも、指示対象が画像として閲覧できるので、簡単にその画像に関連した発言や議論を探し出すことができる。

5.3 まとめ

本章では、取得した指示情報を用いた、ディスカッションブラウザにおける議論の閲覧支援について述べた。5.1 節では指示情報に基づく議論の構造情報を用いた活性拡散アルゴリズムによる発言群のランキングとフィルタリングについて述

べた．この仕組みを用いて効率的に議論を閲覧できる仕組みを実現した．5.2節では指示されたオブジェクトの情報を発言テキスト内に挿入できる仕組みと，その情報が埋め込まれた発言テキストをディスカッションブラウザ上で閲覧できる仕組みについて述べた．これにより，会議中においては書記の入力の負担を軽減する仕組みを実現するとともに，ディスカッションブラウザでの発言閲覧時には閲覧者の発言内容の理解を促す仕組みを実現した．

次章では，本研究で提案した手法に関連する研究について述べる．

第6章 関連研究

6.1 指示情報の取得に関する研究

6.1.1 講義における教材中の指示対象の抽出

通信ネットワークの広帯域化により、大学教育の場においてもネットワークを通じた講義の配信が行われるようになっている。特に遠隔講義や講義アーカイブにおいて、映像や表示教材の組み合わせた講義コンテンツの自動生成や検索を目指して、講義の自動撮影等が行われている [13, 14]。これらを行うために講義室内での講師の位置や表示教材の更新タイミング等に注目されていたが、丸谷らはこれら以外にも講師が教材中で指示する対象 (指示対象) があるとし、指示棒先端の軌跡、講師の体の位置、講師の体の向きと指示動作との関係を整理し、それに基づきセンサ情報から自動で表示スライド内の指示対象を抽出する方法を提案している [10]。抽出した指示対象を用いることで、講演風景の自動撮影時に指示対象を確実に撮影できるようカメラを移動させたり、講義コンテンツの提示の際に指示対象を強調表示したりすることで、講義コンテンツ利用者の理解支援を目指している。この研究では、強調表示のように高精度な抽出が要求されるような用途に関しては、この手法の指示対象の抽出精度が十分ではないため、抽出精度の向上を今後の課題としている。

一方で、本研究における指示対象は矩形選択やトリミングの機能によって誤りの少ない指示を可能にしている。加えて会議中にも強調表示され、さらに必要に応じて拡大や移動が可能であるため、このような用途にも十分利用可能であると考えられる。

6.1.2 会議コンテンツの協調的提示のためのレーザーポインタ情報の抽出

大平らは、ディスカッションマイニングにおいて、プレゼンテーション資料が表示されたスクリーン映像を記録するカメラと、レーザーポインタ検出用のカメラを利用することで、レーザーポインタの指示対象を取得し、会議コンテンツに含まれる、スライド情報、発言者名、発言テキストと、レーザーポインタの指示

対象を、映像の再生に合わせて同期的に提示する手法を提案した [15]。しかし、検出精度がスクリーンと指示者の相対的な位置関係に依存し、場合によっては極端に精度が下がるという問題や、指示動作の開始・終了時間の正確な値が取得できないという問題があった。

本研究ではポインタリモコンという新たなポインティングデバイスを開発し用いることでこれらの問題を解決している。加えて従来のレーザーポインタでは不可能であったポインタの軌跡を描画する、といったことも可能となっている。

6.1.3 講義講演シーン検索のためのレーザーポインタ情報の抽出

仲野らは、教育コンテンツの統合、蓄積、及び統合コンテンツに対する高度な検索機能を実現するシステムである UPRISE (Unified Presentation Slide Retrieval by Impression Search Engine) を提案している [16]。UPRISE では、動画ストリームを資料スライドの切替タイミングによってシーンという単位に分割し、各シーンとそこで使用された資料スライドを対応付けることでそれらを統合する。また、各シーンに対して、対応する資料スライドの文字・構造情報、シーンの時間長の情報などから検索用インデックスを作成することで、高度な検索を可能としている。これらに加え、仲野らは、講師が用いたレーザーポインタの照射情報に着目し、その情報を統合した、より高度な検索を実現する手法を提案している。具体的には、レーザーポインタの照射対象を行単位で扱い、行に対するレーザーポインタ照射の回数と時間という観点でレーザーポインタ情報を抽出することで、どのキーワードを指示していたかを取得し検索に用いている。この手法では、レーザーポインタの情報から抽出している点で汎用性がある一方で、適合率が6割程度しかないため情報に誤りを含む、図の部分を示した場合に情報を取得できない、などの問題がある。

しかし、本研究の手法では、専用のポインティングデバイスを用いることで汎用性が劣る半面、図の部分の正確に指示できる、テキストを指示した場合はそれに含まれる文字情報を自動で取得できるなど、多くの指示情報を正確に取得することができる。

6.2 議論の構造化に関する研究

6.2.1 gIBIS

議論の内容を構造化・可視化することで閲覧者の議論内容に対する理解を促すことが期待できる。その代表的な研究例として挙げられるのが、議論支援グループウェアの IBIS (Issue Based Information Systems) である。IBIS では、共同作業

時の問題解決において発言を、Issue(問題)、Position(立場)、Argument(賛否)の3種類に分類し、それらの関係をリンクとするグラフ構造によって議論を構造化する。そして、IBISをベースとしてGUIによる議論支援を行うシステムがgIBISである[17]。gIBISは議論構造を表すグラフを表示するブラウザ、グラフを構成するノードに関する情報を表示するノードインデックスウィンドウといったコンポーネントから構成されており、ユーザはこれらのコンポーネントを利用して直感的な操作で議論構造の編集を行う。

IBISやgIBISは議論を構造化し、それを可視化することで議論の内容を把握しやすくしている。しかし、議論で発生する発言すべてをこれらのモデルで提案している発言のタイプのいずれかに分類することが難しいという問題点がある。そのため、ディスカッションマイニングでは、発言を新しい話題かどうかという単純な視点から「導入」と「継続」の2つのタイプに分類し、会議における議論を議論セグメントとして構造化していくことを議論の構造化のベースとしている。こうすることで各発言者が議論の構造化に掛けるコストを軽減している。また、本研究ではさらに発言のタイプからではなく指示情報をもとにした構造化も行うことで、複数の発言間にまたがるような関係性も取得できる。

6.2.2 音声認識された議事録からの議論マップの自動生成

趙らは、議論の流れやテーマを常に把握しておけるようにするために、音声認識された議事録から文脈中で周りの多くの語と繋がりを持ち、かつよく登場する語を「テーマキーワード」として取り出し、それらを使って議論の流れや概要をグラフ化したものを「議論マップ」として表わすことを目指している[3]。彼らはテーマキーワードを抽出する方法として、まず音声認識によって発言をテキストデータにし、次に形態素解析を行ってその中から名詞のみを取り出してノイズ除去を行うことでテーマキーワードの候補を抽出している。その後、多くの語と関連を持っている語、特定の語と関連の強い語という二つの観点に基づいて、テーマキーワード候補からテーマキーワードを抽出し、図6.1のような議論マップを生成している。しかし、このようにして生成された議論マップでは、議論全体が何について話していたのかを把握することはできるが、個々の議論でどのような話がなされたのかを詳細に知ることはできない。加えて、実際の議論でどのような順番でテーマキーワードが推移したかがわからず、時間に沿った議論の流れを把握することは難しいと推測される。

一方、本研究では、ディスカッションブラウザにおいて議論全体を把握したい場合は層状シークバーで、個々の議論について詳細に見たい場合は議論詳細ビューで閲覧することができる。また、層状シークバー、議論詳細ビューでは発言が時間順に閲覧できるため、どのようにして議論が推移していったかについても容易に把握できる。また、指示情報を用いた活性拡散による議論セグメントの要約が

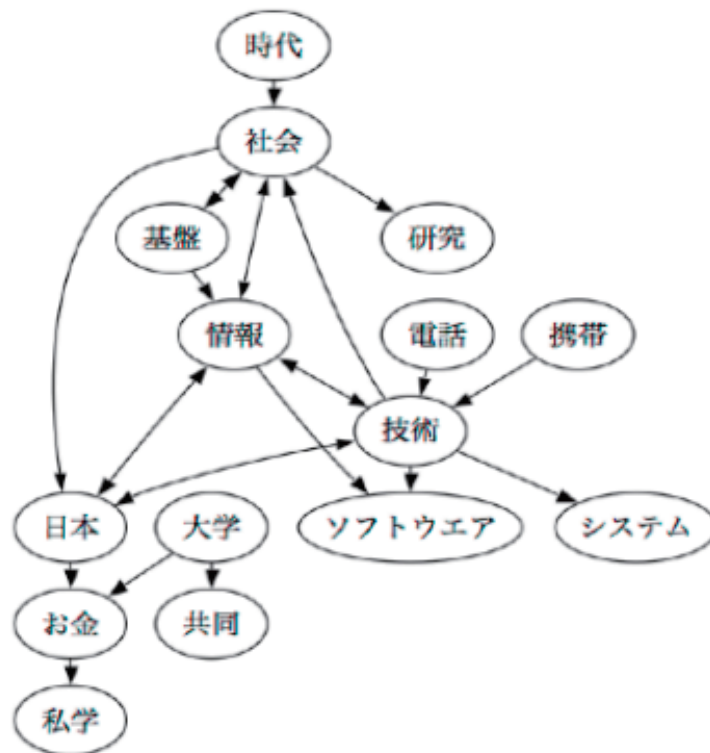


図 6.1: 議論マップの例

可能なため、個々の議論でどのような話が行われていたかを把握することも容易である。

6.2.3 議論構造の可視化による論点の発見と理解

松村らは、議事録における議論の流れを構造化して視覚的に表示することにより議論全体の流れを読者が把握することを支援することを試みている [18]. 「議論を発展させるトリガとなるような話題」を興味深い話題と考え、議論の字面に基づいた表層的な解析により議論の流れを捉える手法を提案している. 具体的には、文書中の語彙的結束性に基づいて話題の境界を自動的に求めることで議論を構造化し、さらに影響の普及モデル IDM (Influence Diffusion Model) を適用することで議論の発展に強く影響を与えた話題を同定し、その結果を図 6.2 のように可視化している.

しかし、この手法はあくまで議論の字面に基づいた表層的な解析でしか議論を捉えておらず、発言が行われたときの発言者の意図などは考慮されていない. そのため、議論の流れに沿って正しく可視化しないと理解しづらく、誤った理解を与えてしまう可能性がある. さらに、構造のみを表示しているため、どのような

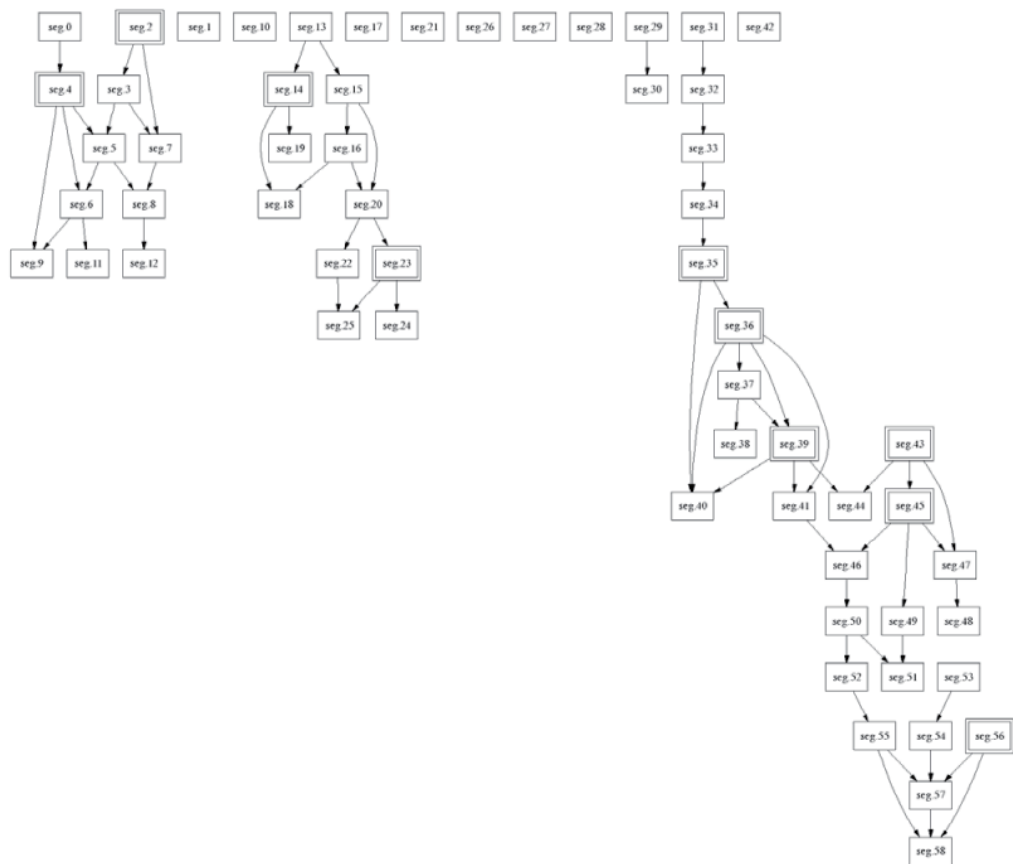


図 6.2: 構造化マップの例

議論が行われていたのかを直感的に知ることができない。

本研究では、発言者の判断のもとに発言を「導入」、「継続」のタイプ分けを行い、さらに発言者が行った指示行為をもとに構造化を行っている。つまり、発言者の意図に沿った構造化が行われるので、議事録の閲覧時に不適切な構造化によって誤った理解を与える、といった問題は発生しない。さらに構造化情報とともに各発言の発言者名や発言テキストといった情報を可視化しているため、議論内容をより直感的に理解することができる。

6.2.4 ミーティングにおける会話のグラフ構造化

Rienks らは、議論図と呼ばれる、発言を表わすノードと発言同士の関係を表わすリンクによって構成されるダイアグラムを作成することで議論を構造化する手法を提案している [19]。彼らは発言を、問題の Yes-No を問う発言 (Yes-No issue)、未解決の問題について話す発言 (Open issue)、A か B を問うような発言 (A/B issue)、のように問題提起の仕方によって5つのタイプに分けている。そして、発言ノード同士

第7章 まとめと今後の課題

7.1 まとめ

我々は、ディスカッションマイニングと呼ばれる、人間同士の知識の交換の場であるミーティングから、映像・音声情報やテキスト情報といった実世界情報を獲得し、それらを半自動的に構造化することによって再利用可能な知識を抽出する技術の研究・開発を行ってきた。ディスカッションマイニングの成果の一つであるディスカッションレコーダでは、会議中に行われた発言を、新しい話題を切り出すことを意味する「導入」と、直前の発言を受けて発言を行うことを意味する「継続」という二つのタイプに分類し、議論セグメントを半自動的に生成していくことで議論の構造化を行ってきた。そして、構造化された議論や会議に用いられたスライド、会議中の映像・音声などは会議コンテンツとして記録され、ディスカッションブラウザによって閲覧できる。

しかし、従来の議論セグメントでは、1つの議論セグメント内で常に同じ話題について話されているとは限らず、複数の話題を含むケースが多く存在することが分かってきた。そのため、発言に対するメタデータが存在していても、閲覧者にとって有用な発言を選定・閲覧することにコストがかかったり、たとえ閲覧者が閲覧したいと思った発言を見つけたとしても、その内容を理解するために必要な過不足ない発言群を選定することが困難であり、効率的な議論の閲覧が行えない、という問題があった。

そこで本研究では、ミーティング中に行われる指示行為に着目し、同じ指示対象を参照して行われた発言同士は、同じ話題について話されている可能性が高いと考えた。指示に関する情報を取得して記録し、それらの情報を用いることで議論セグメントをより詳細な話題ごとに構造化し、その妥当性について確認するために実験を行った。

本研究では、ポインタリモコンを用いてスライド内のオブジェクトを柔軟に指示できるようにして、その情報を取得して記録する仕組みを実現した。具体的にはまずスライド内に含まれるオブジェクトを矩形選択によって指示をできるようにした。その際、指示対象をスライドショーの操作とは独立して移動や拡大が可能なエレメントとして独立化させることで、従来の矩形選択では不可能だった複数のスライドにまたがる指示を可能にした。さらにトリミング機能を実現することで図の一部やグラフの一部など任意の領域の指示を可能にした。そして、テキス

トをOCRによって取得したテキスト情報を用いて下線選択をできるようにした。

このようなオブジェクトの指示方法を提供するとともに指示に関する様々なメタデータも取得した。具体的にはオブジェクトの座標情報をはじめとして、指示者のIdや、指示対象のスライド内におけるオブジェクトのId、オブジェクトの画像、オブジェクトに含まれるテキスト情報、といったものである。加えてポインタリモコンによる指示では指示対象を常在化することで、どのオブジェクトが誰によって指示されているのかを可視化した。これにより明確な指示の開始・終了時間の取得も可能にした。

そして取得したメタデータを用いて議論の構造化を行った。我々は一つの議論セグメント内において、同じ指示対象を指示して行われている発言同士は、同じ話題について話している可能性が高いとし、それぞれの発言と指示対象の間に新たなリンク情報を付与した。そして指示対象を介してそれぞれの発言を関連付けることで、議論セグメントを再構造化した。しかし、指示情報をもとにリンク情報を付与する際、各発言の発言者が自分の発言と指示対象は本当に関係あるのかどうかをきちんと判断しないと正しい構造化ができない、という問題があった。そこで発言の開始時に現在の指示対象と発言内容が関係しているかどうかを問うダイアログを表示し、指示を継続するかどうかを判断させることでこの問題を解決した。

しかし、上述のように議論セグメントの再構造化を行ったが、同じ指示対象を参照していることと同じ話題が一貫して続くこととの依存関係は十分に検証されていないなかったため、指示対象を介してそれぞれの発言を関連付けることの妥当性が示されていないかった。

そこで、「継続元の発言と同じ指示対象を参照した場合は、一貫して同じ話題について話しており、それ以外の話題について述べることは少ない」という仮説を立て検証を行った。まず、ディスカッションレコーダを用いた研究室のゼミにおいて普段通りにポインタリモコンを使用してもらい、その過程で指示情報を記録した。その後、発言者に対してディスカッションブラウザ上で、導入発言と同じ話題について話しているか、直前の話題を継承しているか、いくつの話題を提供しているか、という3つの設問に回答してもらった。アンケート調査の結果、継続元の発言と同じ指示対象を参照していた場合は、指示行為が行われていない場合と比べ、一貫して同じ話題について話す可能性が高い、ということが分かった。これにより、同じ指示対象を参照していることと同じ話題が一貫して続くことの間には依存関係があると考えられ、指示対象を介してそれぞれの発言を関連付けることは妥当であると判断した。

最後に、我々は取得した指示情報と指示情報を用いた構造化の結果を用いて、2種類の議論閲覧支援手法を実現した。1つは、活性拡散アルゴリズムを用いて議論のフィルタリングとランキングを行う仕組みである。本研究で付与した新たなリンク情報を考慮することで、閲覧者の目的に沿った話題について話されている発

言群を適切に選別することを可能にした．もう1つは会議中に指示されたオブジェクトの情報を簡単に書記テキスト内に挿入できる機能を実装し，さらにディスカッションブラウザ上で発言を閲覧する際に挿入されたオブジェクトをテキストや画像として閲覧できる仕組みを実現した．これにより会議中は書記の入力の手間を軽減するとともに，会議後は発言内容の理解を促すことを可能にした．

7.2 今後の課題

7.2.1 スライド外のオブジェクトに対する指示

本研究では，矩形選択や下線選択によってスライド内の様々なオブジェクトへの指示を可能にした．しかし，我々が研究室などのゼミで発表を行う場合，必ずしも指示対象のすべてがスライド内に存在しているとは限らない．例えば，システムのデモを行ったり，実験結果を示すビデオを流したりしたとき，指示対象はデモで見せた1機能であったり，あるいはビデオの1シーンであったりする場合が多い．このようなとき，我々が使用しているポインタリモコンでは指示が行えず，結果として指示情報の取得が行えない，という問題がある．

この問題の解決法として，現在行っているデモの様子やビデオそのものを独立エレメントとして取り込む，という方法が考えられる．本研究ではこれまで独立エレメントではスライド内の一部を切り出し，見た目上は画像のみを扱ってきた．しかし今後は画像だけでなく，映像も扱えるようにすることで上記の問題を解決できると考える．例えば，実際に行うデモならばそれらの様子をリアルタイムにWebカメラ等で撮影してストリーミング配信する映像を，ビデオならばその映像を独立エレメント内で流すことで，ポインタリモコンによる指示が可能となる．しかしこの解決法でも，映像に映っているものを指示したいのか，あるいは一連の動作そのものを指示したいのかなど様々な指示の状況が考えられるためそれらすべてに対応する必要があるなど，課題も多いと考えられる．

7.2.2 指示対象の種別に基づいたリンクの意味付け

本研究では，発言中における指示行為の有無に基づいて新たなリンク情報を追加した．加えて，指示対象がイメージであるか，テキストであるか，あるいはストロークであるか，といった指示対象の種別に関する情報も記録している．しかし，その指示対象の種別によって，発言にどのような違いがあるか，例えば，ストロークを指示した場合はより具体的な話をする傾向がある，イメージを指示した場合は新たな話題を追加する傾向がある，といったことはまだ十分に分析されていない．

このような指示対象の種別を考慮することは、本研究で付与したリンク情報に対して属性を付与することを可能にすると考えられる。例えば、指示対象がテキストと図の場合を比べたとする。テキストを指示している場合、文章が書かれているため、その内容の解釈の仕方もほぼ一意に定まると考えられる。一方で図はそれだけで多くの情報が含まれているため、人によって着眼点や解釈の仕方も異なってくると考えられる。そのため、テキストを指示して行われた発言よりも、図を指示して行われた発言の方がより新たな話題を提示しやすいと考えられる。ゆえに、付与するリンク情報にも図を指示したことによって付与されたものに関しては「新たな話題を提示しやすい」といった属性を付与することができる。このように指示対象の種別によってそれぞれ属性を付与することで、活性拡散アルゴリズムを用いてフィルタリングやランキングを行う際、付与された属性の意味を考慮しパラメータを適切に変更することで、より閲覧者の目的に沿った発言群を提示することが可能になるだろう。

7.2.3 指示領域に基づいた発言群同士の関連付け

本研究で付与した新たなリンク情報は、同一のものを指示したかどうかを考慮しているが、あるオブジェクトの一部を指示したのか、あるいは逆に複数のオブジェクトを一つの独立エレメントとして指示したのか、といった指示領域は考慮していない。

このような指示領域を考慮することは、発言群の関連付けを行う手掛かりになると考えられる。例えば、スライドとしてあるシステムの全体図が表示されていたとする。このとき初めはシステムの一部を指示して個別の機能についての議論が行われていたが、途中からシステム全体を指示して、個々の機能の繋がりについて議論が行われたとする。このようなとき、指示対象はシステムの一部とそれを含むシステム全体図の二つがある。そしてそこで行われた議論はシステムの一部からシステム全体へと、より話しの視点が大きくなっている。この例のように複数の指示対象の指示領域が包含関係にあるとき、そこで行われた議論も、話の視点がより広がった、あるいは具体的になった、抽象的になったなど、何らかの関係を持つのではないかと考えられる。もちろんこれらの考えはまだ十分に分析されていないため推測の域を出ないが、仮にこれらの考え方が正しかった場合、指示情報を用いて付与されたリンク情報によって選別された発言群同士を関係付けることができる。そして発言群同士を関連付けることで、例えばある発言群を閲覧している際に、その発言群で行われた議論とほぼ同じ話題をより詳細化した議論が行われている発言群を提示したり、より広い視点で議論が行われた発言群を提示したりすることが可能となるだろう。

謝辞

本研究を遂行するにあたり、指導教員である長尾確教授をはじめとして、数多くの方々に御支援、御協力を頂きました。ここに心から深く感謝の意を表します。

本研究を進めるにあたり、指導教員である長尾確教授には、研究に対する姿勢や心構えといった基礎的な考え方から、研究に対する貴重な御意見、論文執筆に関する御指導など、大変御世話になりました。心より御礼申し上げます。

松原茂樹准教授には、研究室全体のゼミにおいて貴重な御意見、御指摘を頂き、大変御世話になりました。心より御礼申し上げます。

大平茂輝助教には、研究の本質的なことから、技術的なことまで幅広く貴重な御指導、御意見を頂き、大変御世話になりました。心より御礼申し上げます。

そして、ディスカッションマイニングプロジェクトのリーダーである土田貴裕さんには、論文や発表資料の添削を始め、研究の進め方やプログラミングに関する様々なアドバイスや御指導を頂き、大変御世話になりました。ここに御礼申し上げます。

石戸谷顕太郎さんには、研究に対する取り組み方やプログラミングに関する様々なアドバイスや御指導を頂き、大変御世話になりました。ここに御礼申し上げます。

同プロジェクトメンバーである、森直史さん、高橋勲さん、磯貝邦昭さんには、プロジェクトゼミや研究室全体のゼミにおいて研究に関する様々な御意見を頂いたことに加え、研究の相談にのって頂くなど、大変御世話になりました。ありがとうございました。

同研究科修士課程の井上泰佑さん、山本圭介さん、学部生の渡邊賢さん、棚瀬達央さんには、ゼミ等で貴重な御意見を頂いたことに加え、研究室での活動の中でも大変御世話になりました。ありがとうございました。

研究室のOBであり、本研究の基礎を築いた清水元規さんには、研究室在籍中にゼミ等で貴重な御意見を頂くなど、大変御世話になりました。ありがとうございました。

長尾研究室秘書である鈴木美苗さん、土井ひとみさんには、研究生活や学生生活の様々な面で御世話になりました。ありがとうございました。

最後に、陰ながら見守っていただき、日々の生活を支えて頂いた両親にも最大限の感謝の気持ちをここに表します。ありがとうございました。

参考文献

- [1] 本橋 洋介, 坂上 秀和, 内藤 圭三. 会議支援システムを用いた会議内容情報の再利用性の考察. 情報処理学会研究報告, Vol.2009-GN-72, No.4, 2009.
- [2] 倉本 到, 野田 潤, 藤本 典幸, 荻原 兼一. 会合における備忘録をもとに一時記録を検索参照する会合情報記録検索システム ReSPoM. 情報処理学会論文誌, Vol.41, No.10, 2000.
- [3] 趙 雲超, 松村 真宏, 谷内田 正彦. 音声認識された議事録からの議論マップ自動生成. The 20th Annual Conference of the Japanese Society for Artificial Intelligence, 2006.
- [4] Vedran Sabol, Christian Gutl, Thomas Neidhart, Andreas Juffinger, Werner Klieber, Michael Granitzer. Visualization Metaphors for Multi-modal Meeting Data. In Proceedings of the 6th Workshop of the Multimedia Metadata Community (WMSRM07), Aachen, Germany.
- [5] Christian Gutl. Enhancements of Meeting Information Management and Application for Knowledge Access and Learning Activities. Journal of Universal Computer Science, Vol.14, No.10, 2008 .
- [6] Nagao, K., Kaji, K., Yamamoto, D. and Tomobe, H.. Discussion Mining: Annotation-Based Knowledge Discovery from Real World Activities. Proc. of the Fifth Pacific-Rim Conference on Multimedia (PCM 2004). pp.522-531, 2004.
- [7] 友部 博教, 土田 貴裕, 大平 茂輝, 長尾 確. ディスカッションメディア：会議コンテンツの構造化と効率的な閲覧システム. 第 21 回人工知能学会全国大会論文集, 人工知能学会, 2F3-5, 2007.
- [8] 土田 貴裕, 大平 茂輝, 長尾 確. 対面式会議コンテンツの作成と議論中におけるメタデータの可視化, 情報処理学会論文誌 特集「インタラクションの基盤技術, デザインおよび応用」, Vol.51, No.2, pp.404-416, 2010.
- [9] 清水 元規, 木内 啓輔, 土田 貴裕, 大平 茂輝, 長尾 確, ディスカッションマイニングにおける発言中の指示対象の取得とその応用, 情報処理学会第 72 回全国大会, 2010.

- [10] 丸谷 宜史, 西口 敏司, 角所 考, 美濃 導彦. 講義における教材中の指示対象の抽出, 電子情報通信学会, IEICE, Vol.J90-D, No.5, pp.1238-1248, 2007.
- [11] 桑畑 幸博. 目に見える議論. PHP 研究所. 2008.
- [12] A.M. Collins and E.F. Loftus: A Spreading Activation Theory of Semantic Processing. *Psychological Review*, Vol.82, No.6, pp.407-428, 1975.
- [13] 先山 卓郎, 大野 直樹, 椋木 雅之, 池田 克夫. 遠隔会議における講義状況に応じた送信映像選択. 信学論 (D-11), Vol.J84-D- , No.2, pp.248-257, 2001.
- [14] 亀田 能成, 宮崎 英明, 美濃 導彦, 倉塚 重善, 辻本 雅之. 講師追跡によるカメラ映像の自動切り替え. 情処学春期全大, Vol.2, pp.219-220, 1999.
- [15] 大平 茂輝, 長尾 確. レーザーポインタ指示情報に基づく会議コンテンツの協調的提示手法の提案. 情報処理学会第 68 回全国大会. 2006.
- [16] 仲野 亘, 小林 隆志, 直井 聡, 横田 治夫. 講義講演シーン検索におけるレーザーポインタ情報の活用法. 電子情報通信学会論文誌 D-I, Vol.J91-D, No.03, pp.654-666, 2008.
- [17] Conklin, J. and Begeman, M.L.. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *Proc. of CSCW' 88*, pp.140-152, 1988.
- [18] 松村 真宏, 大澤 幸生, 石塚 満. 議論構造の可視化による論点の発見と理解. 日本ファジィ学会誌, Vol.15, No.5, pp.554-564, 2003.
- [19] Rienks, R., Heylen, D., and van der Weijden, E. Argument diagramming of meeting conversations. In A. Vinciarelli and J.-M. Odobez, editors, *In Proceedings of the Multimodal Multiparty Meeting Processing Workshop at the 7th International Conference on Multimodal Interfaces (ICMI 2005)*, October 3-7, Trento, Italy, pp.85-92, 2005.