ハイパー議事録システムに関する研究

1994 年 3 月 1 日

指導教官 佐伯元司 助教授
提出者　東京工業大学 大学院 理工学研究科
電気電子工学専攻
91D15030 海谷 治彦

目 次

1 はじめに **5**

1.1 背景 . . . 5

1.2 本研究の目的 . . . 6

1.3 関連研究 . . . 7

1.3.1 人的要因 . . . 7

1.3.2 設計理由 . . . 7

1.3.3 支援ツール . . . 8

1.4 本論文の構成 . . . 8

2 実作業の分析 **11**

2.1 分析の方針 . . . 11

2.1.1 分析対象の選択 . . . 11

2.1.2 会議と生産物との対応 . . . 12

2.1.3 分析対象の記録方法 . . . 15

2.2 実際の分析対象 . . . 15

2.3 非効率な議論 . . . 19

2.3.1 分析の目的 . . . 19

2.3.2 分析の手順 . . . 19

2.3.3 分析結果と考察 . . . 21

2.3.4 まとめ . . . 32

2.4 議論内容の生産物からの欠落 . . . 33

2.4.1 分析の目的 . . . 33

2.4.2 分析の手順 . . . 33

2.4.3 分析結果と考察 . . . 35

2.4.4 まとめ . . . 38

2.5 議論の構造とプロダクトの構造の関係 . . . 42

2.5.1 分析の目的 . . . 42

2.5.2 分析の手順 . . . 43

2.5.3 分析結果と考察 . . . 43

2.5.4 まとめ . . . 44

2.6 参加者の役割による会議の分割 . . . 46

2.6.1 分析の目的 . . . 46
2.6.2 分析の手順 . . . 46
2.6.3 分析結果と考察 . . . 48
2.6.4 まとめ . . . 54

3 ハイパー議事録システムの設計 55
3.1 システムの対象となる作業とその問題点 . . . 55
3.2 システムの設計方針 . . . 56
3.3 システムのデータ構造 . . . 57
3.4 Topic の可否の決定 . . . 61
3.4.1 特定の Topic に関する関係の順序 . . . 61
3.4.2 関係の意味 . . . 62
3.5 構造化された会議のデータの例 . . . 62

4 システム利用のための方法 67
4.1 構造化作業の概要 . . . 67
4.2 会議の記録を構造化する手順 . . . 67
4.2.1 会議の構造の作成例 . . . 76
4.3 構造に記述者の提案を付加 . . . 78
4.4 構造からプロダクトを作成 . . . 79

5 ハイパー議事録システムの実現 81
5.1 システムの支援対象と作業の流れ . . . 81
5.1.1 システムの利用者の分類 . . . 81
5.1.2 システムの利用の流れ . . . 81
5.1.3 システムの利用環境 . . . 82
5.2 プロトタイプシステムの機能 . . . 84
5.2.1 参加者管理機能 . . . 84
5.2.2 記録機能 . . . 84
5.2.3 編集機能 . . . 84
5.2.4 検索参照機能 . . . 85
5.2.5 問題警告機能 . . . 85
5.2.6 文書作成機能 . . . 85
5.3 プロトタイプシステムのユーザーインタフェース . . . 86
5.3.1 レコーダ . . . 86
5.3.2 エディタ . . . 88
5.3.3 ビューワ . . . 92
5.4 システムのプログラム構成 . . . 94
5.4.1 データベースサーバー (DBSserver) . . . 95
5.4.2 音声サーバー (GSSserver) . . . 96
5.4.3 ビデオサーバー . . . 96
5.4.4 ユーザーインタフェース . . . 96

6 システムの運用実験と評価 97
6.1 運用実験の方針 . . . 97
6.2 運用実験の設定 . . . 97
6.3 結果 . . . 98
6.4 考察 . . . 103

7 おわりに 109
7.1 まとめ . . . 109
7.2 今後の展望 . . . 110

謝辞 111

参考文献 112

A 運用実験で作成されたプロトタイプ構造 117
A.1 会議 1 の構造 . . . 117
A.2 会議 2 の構造 . . . 120
A.3 会議 3 の構造 . . . 124

B 運用実験で作成された仕様書 131
B.1 要求 . . . 131
B.2 本システムの対象物 . . . 131
B.2.1 本 . . . 131
B.2.2 コピーした論文 . . . 131
B.2.3 電子メディア . . . 132
B.2.4 電子メディアのアーカイブ . . . 132
B.3 登録，公開について . . . 132
B.3.1 データベース . . . 132
B.3.2 公開方法 . . . 132
B.3.3 本の置き場所 . . . 133
B.3.4 貸し出し制限 . . . 133
B.3.5 公開規定 . . . 133
B.4 データ構造 . . . 133
B.4.1 型 . . . 133
B.4.2 identifier . . . 134
B.4.3 タイトル . . . 134
B.4.4 著者 . . . 134
B.4.5 出版社 . . . 134
B.4.6 年度 . . . 134
B.4.7 キーワード . . . 134
B.4.8 場所 . . . 134
B.4.9 備考 . . . 134

B.4.10 所有者 . . . 134
B.4.11 登録者 . . . 135
B.4.12 登録日 . . . 135
B.4.13 借用人 . . . 135
B.4.14 借りた日 . . . 135
B.4.15 返した日 . . . 135
B.5 データベースの操作 . . . 135
B.5.1 登録手続き . . . 135
B.5.2 検索手続き . . . 136
B.5.3 削除手続き . . . 136
B.5.4 修正手続き . . . 137
B.5.5 借用手続き . . . 137
B.5.6 返却手続き . . . 137
B.6 ログについて . . . 137
B.7 インタフェース . . . 137
B.7.1 インタフェース例 (X window 版) . . . 138
B.7.2 インタフェース例 (emacs, terminal 版) . . . 141

第 1 章

はじめに

1.1 背景

近年, 人間の共同作業を計算機によって支援する研究が盛んに行なわれている [1]. それらの研究分野は Computer Supported Cooperative Work (CSCW) と呼ばれ, 人間の共同作業を支援する目的で作られた計算機システムはグループウェアと呼ばれ, そのための国際会議なども数多く開かれている [2, 3, 4, 5].

一般に共同作業は多種多様な形態で遂行される. 例えば, 郵便やファクシミリ通信などによる文書を媒体とした共同作業や, 電話などを用いた打ち合せや, 対面式の会議などの作業形態である. 石井は, 計算機を用いて支援することを想定し, これらの作業形態を, 遠隔分散-非同期型, 遠隔分散-リアルタイム型, 対面-リアルタイム型の 3 つの型に分類している [6]. 遠隔分散-非同期型の作業形態では, 主に電子メールを利用したグループウェアが開発されている [7, 8, 9, 10], [11, 12], [13]. 遠隔分散-リアルタイム型では, マルチメディアなどを利用したビデオ会議システムや, 複数の人間による編集が可能なグループエディタなどが開発されている [14, 15]. 対面-リアルタイム型では, 計算機システムを利用しながら会議を行なう電子会議室システムが主流である [16], [17].

しかし, これらのグループウェアの導入運用は実際には円滑に行なわれていないようである. 例えば, ソフトウェア開発にとって有益なグループウェアとは, 対象となるグループの活動を阻害せず, 支援するものでなければならない. Bill Curtis の報告によれば, いくつかのグループウェア [16], [17] はそのような条件を満たさないためにプロジェクトが中止になったそうである. 特に彼が挙げたグループウェアは共に対面-リアルタイム型であることは注目すべき点である. グループウェアの導入運用が実際には円滑に行なわれないのは, 共同作業の支援や促進において，以下のような問題点があるからである思われる.

第一の問題点は, 従来の方法には共同作業における人的要因に関する考慮が欠けていたことである. 例えば，ソフトウェアの分野では仕様化/設計のために数々の技法 [18], [19], [20], [21], [22] が開発されている. また，投票などの機能に代表される会議による意志決定作業の支援 [17] や，発想支援法 [23, 24] などの方法論も提案されている．しかし, これらを用いたとしても, 人間の行なわなければならない高度で複雑な作業は未解決のまま残されている. また, これらの技法が人間にとって，より複雑で繁雑な作業を強いている危険性もある. 現存の協調作業モデルやツールの多くは, 利用者に特定の作業規則を強いる形のものが多く, それゆえに，利用者が彼らなりの方法で作業を進めて行くことが困難である場合が多い．よって, これらのモデルやツールの基盤としている理論が, いかに優れていたとしても, 実際の作業に適用することが困難なものであれば, その効果はないに等しい. 理想的な作業方法論を適用するのではなく，実際の作業で発生している問題点を取り除くことが，作業の効率を高める支援となると思われる．

第二の問題点は, 共同作業における作業プロセスの情報を有効に利用した支援が行なわれなかったことである. 共同作業プロセス中には生産物として後に残らない多くの情報が含まれている. それらの情報の中で作業の支援に役立つ情報を捜し出す必要がある. 近年では, Design Rationale, すなわち設計理由を生産物と共に記録し, 作業の支援に役立てる方法が注目されている [25], [26]. 特に，対面-リアルタイム型の共同作業では, 設計理由の記録に関して十分に考慮をする必要がある．なぜなら，他の型の作業が，計算機システムを直接の作業媒体とする場合

が多いため，設計理由などの記録が生産物として明文化されやすいのに対し，対面-リアルタイム型の共同作業では, 発話などを媒体の中心とした作業を行なうため，生産物として後に残らない情報が発生し易いからである．対面-リアルタイム型以外の共同作業では，媒体となっている計算機システムを使って作業者間で交換される情報を効率的に管理することも容易である. 例えば電子メールなどの送受信などを管理するのは容易なことである. しかし, 対面-リアルタイム型の共同作業, 特に会議などを計算機を用いて支援する場合は, 生産物として後に残らない多くの情報の記録方法, 管理方法ともに未解明な部分が多い.

1.2 本研究の目的

本研究の目的は，背景で述べた問題点に対する解答の 1 つとして，ハイパー議事録システムを提案することである．ハイパー議事録システムとは，対面-リアルタイム型の共同作業の代表である会議を通して，議事録などの作業者に共通な生産物を作成する作業を支援する計算機システムであり，以下のような特徴を持つ．

1. 会議の履歴を発話情報まで含めて記録し，コンピュータ内に蓄積する．

2. 既に行なわれた会議の履歴を，次の会議で活用できるように構造化する．

3. 構造化された会議の履歴を会議中も含めて随時，検索・参照するためのグラフィカル・ユーザーインターフェースを提供する．

構造化された履歴をアクセスすることにより，参加者は議決内容以外に前回の会議までに結論が出ていない議題や議決の変更にともなう他の議決への影響範囲までも知ることができる．そのため，この構造化された履歴はコンピュータアクセスによって可読な一種の議事録とみなすことができる．通常，会議は時間をおいて複数回行なわれることが多いので，構造化は会議と会議の間に行なうことが一般的と思われるが，会議の進行に会わせてリアルタイムに構造化を行なうことも可能である．

ハイパー議事録システムは以下のような思想の基に設計し実装されている．

1. 複数の作業者が複数回の会議を通して共通の生産物を作成する作業のプロトコル分析を通して明らかにした，計算機システムが支援しなければならない会議の問題点を基に設計がされている．そのために，本システムは，人的要因が考慮し，作業の効率を高めるための支援を行なうことのできるシステムとなり得た．

2. 他の多くの会議支援システムが，いずれも会議中に参加者からのキーボードによるシステムへのテキスト情報入力を基本としているのに対し，本システムは従来の会議と同様に，発言者の声のトーンやジェスチャーといった重要な非テキスト情報をシステムへの入力として直接，扱った．そのために，キーボードに不慣れな参加者も容易にシステムを利用して会議に参加することができ，会議中に発生する設計理由などを記録する方法に関する問題も解消できた．

本システムの適用事例の 1 つとして，顧客と設計者の会議を通して行なわれる場合の多い，ソフトウェア開発の上流工程に位置する要求仕様作成段階での会議などが考えられる．要求仕様段階の作業は，それ以降の工程に大きな影響を与えるため，この段階を支援する本システムの効果は大きい．例えば, Boehm が調査分析した IBM, GTE, TRW などの会社でのソフトウェア開発プロジェクト 63 件では, 要求段階での誤った仮定によって作り出され, 後の段階まで発見されなかった誤りによるコストは, 要求段階を 1 とすると, 設計段階では 3 ~ 6, コード化段階では 10, 開発試験段階では 15 ~ 40, 受入試験段階では 30 ~ 70, 運転段階では 40 ~ 1000 程の相対費用が費やされることが報告されている [27] . また, 彼はソフトウェアの問題を実現後に探し出し修正するのは, 要求や初期の設計段階にそれを行なう場合と比べて 100 倍のコストがかかるとも主張している [28]. よって, この段階での作業における問題点に対する支援は他の段階に対する支援に比べ計り知れない効用が期待できる.

1.3 関連研究

1.3.1 人的要因

Bill Curtis によれば, 共同作業のモデルや支援ツールが受け入れられるためには, チームの活動を阻害せずに, 支援するものでなければならないと主張している [29]. このような問題点を解決するための方法として実際の人間の活動の観察/分析を通して, 有効な支援方法を探求する動きが近年活発に行なわれている. これらの研究で特に必要とされるのは, それぞれの分析から得た知見をどのように支援に生かすかの議論である.

Diane Wakz が行なったオブジェクト指向データベースの設計のために 5 カ月にわたって行なわれた 37 回の会議の分析によると, 設計のための共通のモデルを作り出すのに多くの時間が費やされていることが報告されている [29].

岸本らは, 共同作業における意図の伝達の不具合に注目して, 実際のソフトウェア開発プロジェクトの実装段階の作業者に対して, アンケートを行ない, それの分析を行なった [30]. その分析を通して, 作成対象となっているシステムのイメージに作業者間に差異がある場合, 意図の伝達は円滑には行なわれず, 仕様書のみではシステムの漠然とした輪郭を伝えることができるのみであることを主張している.

Olson らの MCC 社と AC 社のソフトウェアの仕様作成会議の分析によると, かかった時間のおよそ 4 割が設計そのものの議論であり, 3 割がウォークスルーや要約を行なうことによる評価であり, 2 割が純粋な調整活動であることが報告されている [31].

1.3.2 設計理由

設計理由などのプロダクトとして残らない情報を支援に利用する研究が注目されていることは，研究の背景の節でも述べた. この領域では支援対象となる作業をどのようにモデル化し, それをどのような言語で表現するかが研究の中心となっている.

設計理由を記録するためのモデルの代表として IBIS モデル [32, 33] がある. IBIS モデルでは議論の論点に相当する Issue, Issue の解法の候補である Position, Position の理由などに当たる Argument の 3 種類の概念を組み合わせることで議論をモデル化する. IBIS に構成要素を追加した Potts のモデル [34] や, Potts のモデルにさらに拡張を加えた Lee の DRL 言語 [35] などがあるが, モデルを複雑にすれば表現力や推論能力が上がる一方, 実際の作業への適用性が減少するという問題点がある.

設計理由記録のためのモデルの評価基準としては, 利用者が表現したいことをいかに容易に, そして正確に記述できるかという点が大きなポイントとなる. しかし, 吉府らが, 実際の仕様検討会議の対話内容を IBIS および DRL に従い記述した結果, 上記のポイントからはこれら 2 つのモデルはまだ不十分なようである [36]. また, 桑名の分析 [37] の結果から, 理由に関する質問は明示的に行なわれにくいことが考えられるため, これらのモデルの構成要素を実際の作業内の要素とを対応付ける工夫が必要となると思われる.

MacLean らは Question, Option, Criteria の 3 つの構成要素からなる QOC 言語を提案している [38]. この表記は IBIS などに比べ, 問題点の明らかになっている良く理解された分野向きであるとされているが, Question, Option, Criteria は, 順に IBIS における Issue, Position, Argument とほとんど同等であり, IBIS の持つ問題点を継承すると思われる. 桑名らは, この言語を用いた場合の会議と, なにも制約を与えていない会議との比較実験を行なっている. 前者の会議では, 設計判断の根拠や設計解などを複数の観点から総合的な判断をしており, 検討洩れの防止や, 設計品質の向上支援の可能性があると主張している [39, 40].

また, 桑名は, 実際の要求仕様作成会議において技術者が互い行なった質問に注目し, 技術者のどのような情報を知りたいかという観点からの分析を行なっている [37]. この分析の目的は, このような情報の種類の特定を通して, どのようなデータがソフトウェア開発において有用かを考察することである. 特に興味をひく結果は, “なぜ” に該当する理由に関する質問が少ないことである. 彼はこの結果に対して, 理由に関する質問は, 明示的に行なわれるのではなく, 議論の文脈などに織り込まれているのであろうという考察を行なっている. よって, 設計の理由などを支援に利用するために記録するには, 議論の文脈などを考慮する必要があると思われる.

ソフトウェアの分野では, 中島らが PPK 法という設計プロセスの記録と分析のための手法と, その支援ツールを試作している [41, 42]. この方法は基本的には個人作業の記録とるための手法であり, 複数の作業者による共同作業に適用するにはなんからの改善が必要だと考えられる.

浜田らは, ソフトウェアの設計プロセスのノウハウを事例ベース化し, 支援に役立てる研究を進めている [43, 44]. また落水らの Vela も事例ベース推論と帰納推論に基づく設計判断履歴の構築を行なっている [45]. これらは人工知能の分野の技術を応用したものである.

会議中のほとんどの時間を作業者は会話に費やしているという報告 [31] からも明らかなように，会議での情報交換の中心となるのは作業者の行なう発話である．発話データの重要性は古くから指摘されている [46]. また, 既存のグループウェアの中にも会話に関する理論 [47] に基づいて設計されているものも見られる [9].

1.3.3 支援ツール

Colab[16] は 2 ~ 6 人の小グループのための電子会議システムであり, 大型共用スクリーンを備えている. 会議参加者はそれぞれの計算機が与えられ, それを利用して会議を行なう. 特に共用 Window を用いてグループにより同時に利用できるアイディアプロセッシングツール Cognoter は, グループによるアイディア形成, 発表支援, 文書作成などの支援を行なう.

NICK[17] は, 情報の解釈選別機構をもったリポジトリである “pot” を利用することで設計会議の支援を行なう. 会議の参加者は, それぞれの pot を持つことができ, 会議の情報を適宜, 選別して自分の pot に保存し, 自分の pot から必要な情報を適宜とりだして発表することで会議を進めることができる. しかし, 背景でも述べたように Bill Curtis の報告によれば, Colab および NICK は作業の効率を向上するには至らなかったそうである.

IBIS モデル [32, 33] を利用した計算機ツールとしては gIBIS[48], rIBIS[49], itIBIS[50] などが開発されている. これらのツールでは通常，IBIS の構造を 2 次元上の網形式で可視化する支援を行なっている.

また DRL 言語を基にしたツールとして SIBYL が開発されている [51]. これも基本的には DRL 言語の言語要素管理と, グラフィカルブラウザを提供しているものである.

COMICS[52, 53] は, 1.3.1節で紹介した岸本らの研究 [30] をもとに, ソフトウェアの意図を伝達するためのモデルである劇場モデルを具体化したものである. 支援対象は設計からコーディングの下流工程である.

KJ 法 [23] を基盤とした個人用の計算機ツールが開発されており [24], それをグループウェアに拡張する研究が盛んに行なわれている [54, 55, 56, 57]. KJ 法は，文章などが書かれたカードを平面上に配置し，それらをグループ化することによって思考を進める方法論である．よって，会話を中心とした会議形式の作業形態に導入する場合には個人用のツールにはない機能が要求されると思われる.

1.4 本論文の構成

本論文では, ハイパー議事録システムを構築するための基礎研究，設計と実装，そして運用と評価について順を追って紹介する．本論文は全 7 章より構成される.

第 2 章では，ハイパー議事録システムを構築するための基礎研究として，システムの支援対象である実際の対面-リアルタイム型の会議のプロトコル分析について述べる．従来の会議は，参加者の発話によって構成される議論から，会議の生産物である文書などが作成されることに着目し，議論と文書の対応を付けることにより，従来の会議の問題点を抽出するという分析の方針をとっている．分析対象となった 3 つのプロジェクトは，ソフトウェアの仕様書などの文書を複数の作業者が複数の会議を通して作成するプロジェクトである．これらのプロジェクトは，本システムが支援対象としている作業の典型的な例である．実際の現場におけるソフトウェア開発のデータを収集することは困難であるために, これらのプロジェクトは実験的に設定されたものである．この分析では，ビデオカメラを用いて記録した実際の会議の様子を分析対象とするため，会議の参加者の活動を阻害せずに，詳細な分析が可能となっている．分析結果として，会議では，議論結果が会議の出力として落丁してしまう，結論が曖昧なままに会議が終了してしまう，1 つの議決が変更されたときにそれと関連のある議題の再審議を行なわず矛

盾を含む結果が残されたままになる危険があるなどの問題点が，会議の質や効率を下げる原因になっていることがわかった．

第 3 章では, 第 2 章で得られた結果を基にして，ハイパー議事録システムの設計を行なう．ここでの設計とは，会議，会議の生産物そして会議の参加者を統合したデータ構造を構築することである．データ構造は，個体-関係モデルの言語で記述し，その内容は，会議と生産物と作業者を構成する要素間に，支援を行なうために必要な関係をつけたハイパーテキストの構造となっている．このデータ構造に従って会議と生産物と作業者を記述することで，分析で得られた問題点を明確にすることができる．

第 4 章では，実際にハイパー議事録システムを効率的に利用するための方法，特に実際の会議の記録をハイパー議事録としてシステムに入力する手順の 1 つを提案する．第 3 章の設計に従ったシステムは，その環境に応じてあらゆる利用方法が考えられるが，本章で提案する方法に従うことで，容易に会議の記録をハイパー議事録として，構造化することができる．

第 5 章では, 第 3 章の設計に従って実装されたハイパー議事録システムのプロトタイプを紹介する. このプロトタイプシステムは，計算機網に接続された複数の計算機上で稼働するグループウェアである．また，本章では，このプロトタイプシステムのユーザーインタフェースの紹介を通して，利用者に提供される検索機能などを紹介する．

第 6 章では, 第 5 章で紹介したプロトタイプを利用したシステムの運用実験を行なう. この運用実験を通して, 本論文で提案したハイパー議事録システムの有効性の確認，第 4 章で紹介した方法の評価，そしてプロトタイプシステムのインタフェースの評価などを行なう.

最後に, 第 7 章にまとめと今後の展望について述べる.

なお付録 A には, 第 6 章で紹介した運用実験で作成された生産物である文書のプロトタイプの構造を付加し, 付録 B には, この運用実験で最終的に承認された文書を付加した.

第 2 章

実作業の分析

2.1 分析の方針

2.1.1 分析対象の選択

ハイパー議事録システムが提供する支援機能を決定するために，実際の会議の分析を行なう．分析対象は，本システムの適用範囲となる以下のような特徴の会議である．

1. 複数 (5 ~ 6 人程度) の作業者が参加していること.
2. 対面式の会議であること.
3. 複数回の会議が行なわれること．
4. 共通の生産物を作成する作業であること．

対面式でない会議形式 (ビデオ会議システムなど) を分析対象とすることは，従来の会議の問題点以外の要素が含まれる場合があるので，本分析の対象から除外した．ある生産物を作成するための複数回の会議の列をプロジェクトと呼ぶことにする. 分析はプロジェクト単位で行なう.

上記の範囲内においても会議の形態には違いがあり, 会議の問題点を分析するためには, できる限り広範囲な種類の会議を分析対象として選ぶ必要がある. 本論文では, 会議に関する以下の点に注目して分析対象である会議を選択した.

- 作業者の種類:
  会議には種類の違う作業者が参加する場合がある. 作業者の種類がどのような基準で分類されているかによって, 会議の形態が違ってくると考えられる. 本論文では以下のような 2 つの分類基準から作業者の種類が分類されている会議を分析対象とする.
  - 生産物を基にした分類基準: 作業者が担当する生産物の部分から, その種類を決定する分類基準.
  - 役割の違いによる分類基準: 作業者がどのような役割を持ってその会議に参加しているかという観点から, 作業者の種類を決定する分類基準. 例えば, ソフトウェアの仕様を作成する会議の場合は，プロジェクトを管理する作業者, 要求を持つ作業者, 実際に仕様を書く作業者, 議事録を取る作業者, 実装する作業者, 製作されたシステムを利用するであろう作業者などが参加する．

  プロジェクトの途中で作業者の種類が決定される場合があるが, その点については個々の分析中で議論する.
- 生産物の作成方法: ここでの生産物とは会議の出力として作成される文書などを指す．例えば，ソフトウェアの仕様を決定する会議では，仕様書が生産物となり，新しい企画を立てる会議などでは企画書が生産物となる．上記の例に示す通り，通常，生産物は文書として作成される場合が多い．ハイパー議事録が対象とする生産物は，作業者に共通な生産物であるが，その生産物が実際に記述される方法は以下の 2 通りが考えられる．

- 1 人の作業者が記述する.
- 複数の作業者が担当部分を記述する.

- 議事録の作成方法: 議事録も生産物の 1 つと考えることもできるが，議事録は作業の進行を促進するために記述されるため，仕様書や企画書とは性質が異なる．よって，生産物とは分けて考える．その記述方法も，生産物の場合と同様に，以下の 2 通りが考えられる．
  - 1 人の作業者が記述する.
  - 複数の作業者が自分に必要な記録を行なう.

2.1.2 会議と生産物との対応

本分析は, 従来の会議における問題点を明らかにすることが目的である．ハイパー議事録システムは，この分析で得られた問題点の解決を支援するよう設計される．本分析では, 問題点を調べるために会議の中の以下の要素に注目した.

1. 会議内の議論の構造と会議の効率の関係:
   会議において議論される話題に注目して, 会議の流れを分析すると, 支援しなければならない問題点が明らかになるはずである.
2. 会議内の議論の構造と議事録や仕様書などの生産物の構造の関係:
   会議で議論されたことが, 生産物として記述される場合に問題点があるはずである. また, 会議の中で話題が議論される順番は, 生産物中の構造となんらかの関係があるはずである.
3. 会議内の参加者の発話と会議の構造の関係:
   会議内のそれぞれの部分で, どのような役割の作業者が, どのくらいの量の会話を, どのような役割の他の作業者と交わしたかが, その部分の作成すべき生産物と関係があるはずである.

1 が 2.3 節，2 が 2.4 節と 2.5 節，そして 3 が 2.6 節の分析に対応している. 分析で注目する点を図 2.1に整理して示す. 本分析では，複数の作業者が複数回の会議を通して, 文書などの共通の生産物を作成して行く流れを, 会議の順序の構造, 生産物の構造, 参加者の発話という 3 つの要素からとらえている.

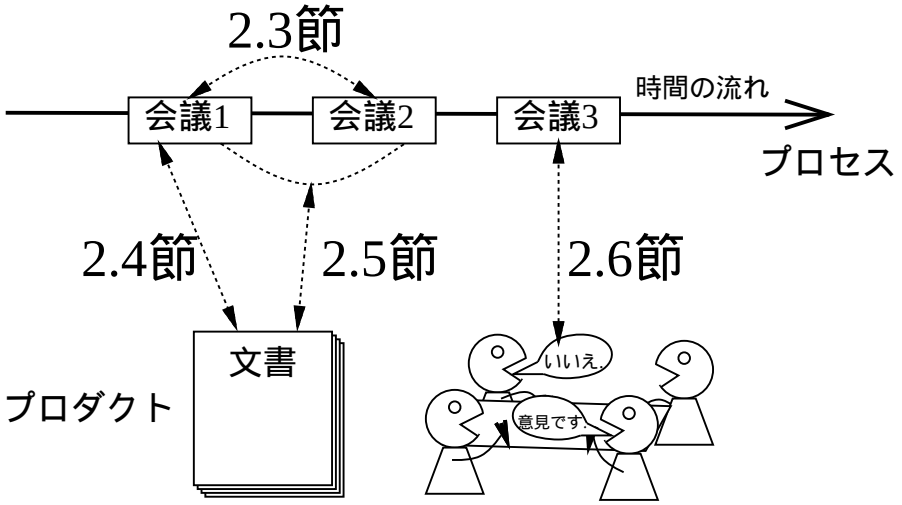


図 2.1: 分析で注目する点

分析の中では会議の列をプロセスと呼び, 会議から作成された文書などの生産物をプロダクトと呼ぶ場合がある. 本研究では, 上記の様に，プロセス間，プロダクト間そしてプロセス-プロダクト間の関係の分析を行なうため，1 つの会議を議論という単位に分割し, 文書を事項という単位に分割して分析を行なう.

プロダクトを事項に分割する場合は, 基本的にはプロダクト中の個々の文章を事項とする. 但し, 文章の内容や流れを考慮して, 1 つの文章を複数に分割したり複数の文章を 1 つにまとめたりする場合がある. 図 2.2は “描画エディタ” というソフトウェアを仕様化するプロジェクトで作成されたプロダクトの 1 つである実際の仕様書の一部である. この仕様書の一部には, 図 2.3に示す 11 個の事項が含まれているとする.

⋮

図形の選択: 1) 1 点を指定して最短距離にある図形を選択. 2) 矩形で括って内部に含まれる/交わる図形を選択. 3) 各図形に小さな枠を出してそれを掴むことで選択.

削除: 利用者に問い合わせはせずに即座に削除. 削除の取消は undo で行なう.

移動: 移動させたい図形を選択し, 移動先を一点指定して作業を完了.

拡大/縮小: 図形を選択し, その外形の矩形を変更することで選択図形の拡大/縮小を行なう.

⋮

図 2.2: 実際の仕様書の一部

1. 図形の選択
2. 図形の選択: 1 点を指定して最短距離にある図形を選択.
3. 図形の選択: 矩形で括って内部に含まれる/交わる図形を選択.
4. 図形の選択: 各図形に小さな枠を出してそれを掴むことで選択.
5. 図形の削除
6. 図形の削除: 利用者に問い合わせはせずに, 即座に削除.
7. 図形の削除: 取消は undo で行なう.
8. 図形の移動
9. 図形の移動: 移動させたい図形を選択し, 移動先を一点指定して作業を完了.
10. 図形の拡大/縮小
11. 図形の拡大/縮小: 図形を選択し, その外形の矩形を変更することで選択図形の拡大/縮小を行なう.

図 2.3: 図 2.2 における事項

会議を議論に分割するために, 会議中の個々の作業者の発話および, それに準ずる行為, 例えば, 黒板に記述を行なうなどを意味を基にグループ化する. 具体的には, 行為中のプロダクトの部分を指す語や話題を開始, 終了させる語などを利用してグループ化する. この発話などのグループを議論とする. もちろん , 冗談, 余談などの部分は議論としてはとりあげない.

会議を議論に分割するためには, 会議のビデオテープによる記録からトランスクリプションを作成する. 会話分析 [58, 59] で用いるトランスクリプションのようにイントネーションなどの発話の物理的な特徴に関するコメントの付加は行なわず, 内容を把握するためのコメントの付加を積極的に行なう. 図 2.4にトランスクリプションの例を示す. 行頭の% はコメントもしくは制御行を示す. 例えば, 黒板などに書かれた内容などを記述する. 行頭が数字もしくは∗ の場合は, 第 1 列が発話の開始された絶対時間を示し, 第 2 列が発話者 (この例では A, B, C の 3 人) を示す. ∗ は直前の発話の直後, もしくは直前の発話に重なって行なわれたことを示す. %cat10 begin と%cat10 end で, 議論の範囲を指定している. 10 は議論の番号である. 発話中で () がついている語を内容を把握するためのキーワードとして用いる.

図 2.5に実際の会議中の議論の例を示す. 図中の A, B, C は発話者を表している. 議論 1 は, 3 番目の C の発話中の「そういえば」という語から話題が変化したとみなし, この発話より “次の議論” が開始されたとする. それ以外の議論中では,「そうしよう」や「そうだね」などの話題の終了を示す語が最後の発話中に見られるので, その

```
%cat10 begin
33.3 A
図形を選んだ後に, 移動とかするわけだけど,
(ラバーバンド) とか出せばわかりやすいんじゃないの?
* B
(ラバーバンド)ってなに?
33.7 A
図形を動かす時の軌跡のようなものだよ.
33.10 C
図形をきっちり書かないで, 外枠の四角だけ書けばいいんじゃないの?
% 黒板に四角の絵を書く.
* B
そうだね.
%cat10 end
```

図 2.4: トランスクリプションの例

議論 1:

A: あと 掴み方 ね. 枠をつくってそこに入ってるものを選択すればいいのじゃないの?
B: 交わっているものを含めて掴めたほうがいいんじゃないの?
C: そういえば, 掴んじゃったものは, どんな風に見えるようにしようか?
色でも変えようか? (次の議論の開始)

議論 2:

A: 削除 の時には, 消していいかどうか利用者に聞こうか?
B: それって, うっとおしいんじゃないの.
A: じゃあ, 消す時は, 有無を言わさず, 消しちゃおう.
B: オーケー.

議論 3:

A: ねえ, それじゃ, 消しちゃ まずいものを 消したら どうするの?
B: そういう場合は undo で復活すればいいじゃない.
A: そうしよう.

議論 4:

A: 図形を選んだ後に, 移動とかするわけだけど, ラバーバンド とか出せばわかりやすいんじゃないの?
B: ラバーバンド ってなに?
A: 図形を動かす時の軌跡のようなものだよ.
C: 図形をきっちり書かないで, 外枠の四角だけ書けばいいんじゃないの?
B: そうだね.

図 2.5: 会議中の議論の例

発話までを 1 つの議論とみなす. 議論と文書内の事項の対応を付けを行なう場合は, 図中で下線をひいたキーワードなどを利用する.

2.1.3 分析対象の記録方法

分析では, 会議による発話などの記録と, 最終的に作成された文書などのプロダクトを利用する. この方法によって, 会議内の作業者の作業を妨げずに会議を記録することができる. 本論文では, 主に会議参加者の発話を分析対象にしているが, 会議中の話者の特定などを行なう場合に映像の記録が役に立つ. また, 黒板などに記述を行なう行為も発話と同等の行為とみなす場合があるためこの場合もビデオによる画像の記録は不可欠である.

分析対象の記録方法としては, 音声レコーダを使う方法や, 作業者にアンケートやインタビューを行なう方法や, 作業者に記録をとるための特別な作業を行なってもらう方法などがある. 音声レコーダを使う方法はビデオカメラを使う方法に包含されている. アンケートやインタビューを行なう方法は被験者の記憶や主観に依存する部分がかなり多いため, 本論文で分析したいデータを収集するには不適切である. 記録のための特別な作業を行なう方法は, それによって会議自身に別の要素が加わってしまうため, 本論文で分析したいデータを収集するには不適切である. 例えば Soloway らは, 被験者に “ひとりごと” をいってもらうことで実験データを収集したが [60], そのような作業は被験者にとって大きな負担となり, 思考作業そのものに影響を与える恐れがあるため最良とは言えない. 特定の記録用ツールやモデル [48] [61] [35] を用いて記録を行なう場合, 作業者の行動を制約したり, 記録したい情報の欠落の恐れがある.

発話を分析するための手法として会話分析 [58, 59] がある. この手法は会話の構造的側面 (会話における間や, 発言の割り込みなど) を主に扱うため, ある程度, 自動的に発話の記録を処理することが期待できるが, 本論文の方法では文書内の内容と発話の内容の対応付けを行なうため, 会話分析の手法は有効ではない.

2.2 実際の分析対象

分析対象としては現実の会議を対象とするのが望ましいので , 現実の会議に近い形の会議の模擬実験を行なった. 複数の作業者が複数回の会議を通して共通の生産物を作成する作業の代表として , ソフトウェアシステムの仕様を作成する作業を分析対象として選択した . 模擬実験の会議での共通の設定は以下の通りである.

- 各々が自由に対話できるように, 作業者に長机を取り囲んで座らせた.
- 長机のそばに白板を置き, 会議中に自由に使用できるようにした.
- 作業者には記録用紙を与え, 会議中に自由にメモを取れるようにした.

実際に会議を行なったのは特に言及しない限り本学情報工学科の学部 4 年から修士 2 年までの学生である. 方針でのべた分類基準から見たプロジェクトの比較を表 2.1に示す. プロジェクト 3(全体) では, プロジェクト 3(設計) で作成された仕様書をプロダクトとしているので, 仕様書の作成人数の欄は空欄である. また, プロジェクト 3(全体) での作業者には調整者が含まれており, 会議の進行がある程度, 管理されているが, 他の会議では会議の進行の管理は明示的に行なっていない. 分析対象となったプロジェクトの概要を表 2.2に示す.

表 2.1: プロジェクトの比較

	プロジェクト 1	プロジェクト 2	プロジェクト 3(全体)	プロジェクト 3(設計)
作業者の種類	生産物	役割	役割	プロダクト
仕様書の作成人数	複数	一人	-	複数
議事録作成方法	個々の作業者	専任の書記	個々の作業者	個々の作業者

図 2.6にプロジェクト 1 の会議の流れの概要を示す. 実際の会議は 3 回行なわれたが, 最終的な仕様書は会議 2 と 3 の間に作成され, 3 回の会議以降に新しい仕様書の作成は行なわれなかった. 表 2.2中の作業者の種類は以下のような内容の作業者を指す.

表 2.2: 分析対象となるプロジェクトの概要

	プロジェクト 1	プロジェクト 2
会議回数	3	4
作業者数	6	5
作業者の種類の内訳	顧客, 設計者 (5)	顧客, 書記, 設計者 (3)
作業者の種類の決定時	会議 1 終了時点	会議 1 開始時点
仕様書の枚数	8	4
仕様書の記述方法	作業者がそれぞれの実装する分担について会議外で記述	設計者の 1 人が会議中にリアルタイムで作成した議事録を会議外で要約して記述
仕様書の決定時	第 2 回会議と第 3 回会議の間	全会議終了後
作成対象	論文の図など作成する目的の描画エディタ	個人がソフトウェアの仕様の記述を行なうためのカード型仕様記述ツール.

	プロジェクト 3(全体)	プロジェクト 3(設計者)
会議回数	4	5
作業者数	8(全員)	2(設計者のみ)
作業者の種類の内訳	顧客, 調整者, 製作者 (4), 設計者 (2)	設計者 (2)
作業者の種類の決定時	会議 1 開始時点	会議 1 終了時点
仕様書の枚数	15	
仕様書の記述方法	設計者がそれぞれの設計する分担について会議外で記述	
仕様書の決定時	第 5 回設計者会議と第 4 回全体会議の間	
作成対象	グループの生活日程を統合的に管理するためのスケジュール管理ソフトウェア	

顧客 : 具体的な要求を提案する作業者であり仕様書の記述は行なわない.

設計者 : 分担された部分の仕様書を記述し, その部分のプログラムの実装を自ら行なう責任を持っている. 具体的には,

入出力部分, 画面部分, 文字部分, 編集部分, 描画部分

の 5 つの部分に分担され, それぞれの部分の担当となっている作業者が個々に仕様書を記述した.

このプロジェクトでは, 個々の作業者は自分の分担となった部分を代表して議論するような特徴が見られると思われるため, 分析対象として選択した.

図 2.7 にプロジェクト 2 の会議の流れの概要を示す. 表 2.2中の作業者の種類は以下のような内容の作業者を指す.

顧客 : プロジェクト 1 と同様に, 具体的な要求を提案する作業者であり仕様書の記述は行なわない.

設計者 : 仕様の記述を行なう作業者. プロジェクト 1 とは異なり, 記述部分の分担は行なっておらず, プログラムの実装を自ら行なう責任もない. 仕様書はプロジェクト終了後に設計者のうちの 1 人が会議中に書記が記述した議事録を構造化した文書である.

書記 : 会議における議論の議事録をとる作業者. 会議中における議事録の内容についての問い合わせに対して随時, 返答する.

このプロジェクトは, 書記による記録の効果の特徴が見られると思われるため, 分析対象として選択した.

図 2.8 にプロジェクト 3 の会議の流れの概要を示す. 会議は大きく分けて以下の 2 種類に分類できる.

設計者会議 : 仕様を設計する設計者のみが参加した会議.

全体会議 : 設計者に加え顧客, 利用者, 製作者を交えた会議.

仕様は利用者に提案するために, 2 人の作業者が分担して作成した文書である. この 2 人の作業者は仕様書を記述するのみであり, プログラムの実装は全体会議に出席した製作者が行なうことになっている. それぞれの作業者の種類の内訳は以下に示す通りであり, () 内に被験者の本業を提示する.

顧客 : ツールの製作依頼者 (大学職員)

設計者 : 発注者の要求に従って仕様書を作成する人 (本学修士 1 年および学部 4 年)

製作者 : 仕様書に従って実際のプログラムを作成する人 (短大情報系学科 2 年)

調整者 ツール作成側の核となって会議の進行を行なう人 (主に専用システムのコーディネートを行なってきた客対応の販売 SE)

仕様の初版は会議 3 にそれぞれの作業者によって議論され, 最終的に仕様書が全体会議に提出された. 仕様書は, 最初の 2 ページのデータ構造に関する記述の部分とそれ以外の操作とインタフェースの部分から構成され, それぞれの部分の担当となっている作業者が別々に記述した文書である. このプロジェクトは, 調整者による調整の効果と, 役割による作業者の対立の特徴がみられると思われるため, 分析対象として選択した.

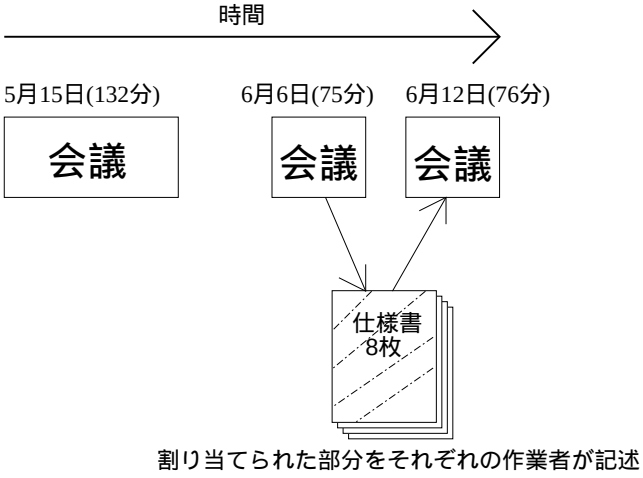


図 2.6: プロジェクト 1 の会議の流れ

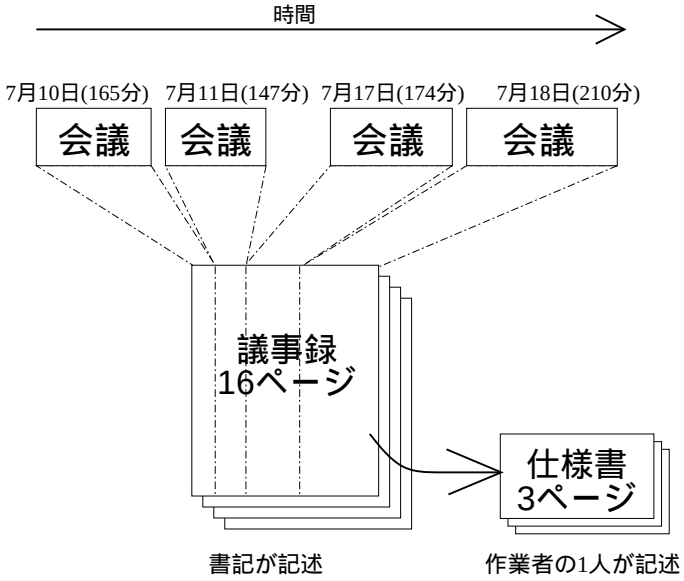


図 2.7: プロジェクト 2 の会議の流れ

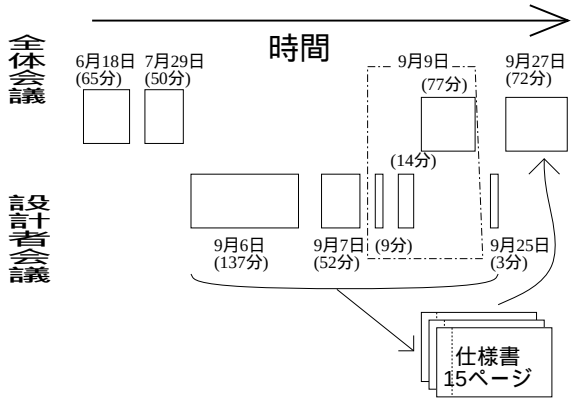


図 2.8: プロジェクト 3 の会議の流れ

2.3 非効率な議論

2.3.1 分析の目的

会議は, 複数の人間がある事項について意見を出し調整を行なって結論を出す活動である. 会議の中には様々な種類の発話が含まれる. それらは, 会議の目的のための議論と, 会議に関係のない冗談, 余談などに大きく分けられる. 後者の冗談, 余談が不必要なことは言うまでもない. なぜなら, 会議の本質には影響しないからである. 一方, 会議の目的のための議論にも不必要なものがあると考えられる. それは, 会議の生成物に貢献しない議論である. ある議論が会議の生成物に影響を及ぼしていない場合, それは最初からなくても構わないはずである. このような議論を非効率な議論と呼ぶことにする. そして, 非効率な議論の発生が少ない会議は, 同じプロダクトを作成するのに要する会議の時間が短縮される. よって, 本分析では, 非効率な議論の発生量や発生の特徴を調べることで, その発生を減少させるための方法を考察する. なお, 本分析では結論の品質自体は考えず, それらの議論はプロダクトとの対応に関する 2.4 節などで議論する .

本論文では以下のような議論を非効率な議論として, 実際の会議の中における, それぞれの時間の占める割合と, その内容的な特徴を調べる. なお図 2.9, 2.10, 2.11 における矩形は議論を表し, 時間は左から右に流れるとする.

同じ議論を繰り返す議論 :

> 前に行なった議論と同じ議論を行なう議論を非効率な議論とする. 図 2.9に示すように, 事項, 結論が共に同じものを指す. この場合, 生成物に含まれている情報は先の議論が終わった時点でできていることになるため, 後者の議論が非効率な議論とみなされる.

結論を後の会議で変更された議論 :

> 後の議論において同じ事項のものがあり, かつ異なる結論を出された場合に発生する. 図 2.10に示すように, 先に行なわれた議論における結論 X は生成物に含まれないため, 先に行なわれた議論を非効率な議論とみなす. 修正, 訂正といった活動も会議での重要な活動であるが, ここではそれらを行なわない会議を理想とする.

結論のない議論 :

> 事項が提示されたものの, 結論が出ない議論を非効率な議論とする. 図 2.11に示すように, 結論がないので, 生成物に貢献しないため, 非効率ととらえる. 結論が出ないことが分かるということも議論の重要な成果であるが, ここでは会議の直接の目的に貢献しないので非効率とする. 例としては, 結論を出すことを先送りにして忘れてしまっていたり, 1 作業者が一方的に発言して次の事項に変化したりする場合である.

2.3.2 分析の手順

本節では以上の 3 つの型の議論を非効率な議論として, 2.2 節で紹介したプロジェクトから, その特徴の分析を行なう. 具体的な作業手順は,

1. 会議の記録から議論を取り出す.
2. 個々の議論の明示的あるいは暗示的な事項を把握する. そして, それの結論がある場合は事項との組として記録する. 結論のない議論はこの時点で検出される.
3. 議論の記録から同じ議論, 結論を後に変更された議論を取り出す. 明示的に結論を変更されていないものも, 観察者が変更されたと判断したものは含まれる.

実際の非効率な議論の例を図 2.12に示す. これは, プロジェクト 3 の第 1 回の設計者会議の冒頭で行なわれた会話である. 最初に「 メインとなるハイパートークを他のプログラムに被せられるか 」という事項を提示されているが, その結論を出さずに「 データベースを使用するのか 」,「 既存のデータベースが存在するのか 」という事項に移ってしまっている. よって最初に提示された事項についての部分が非効率な議論となっている.

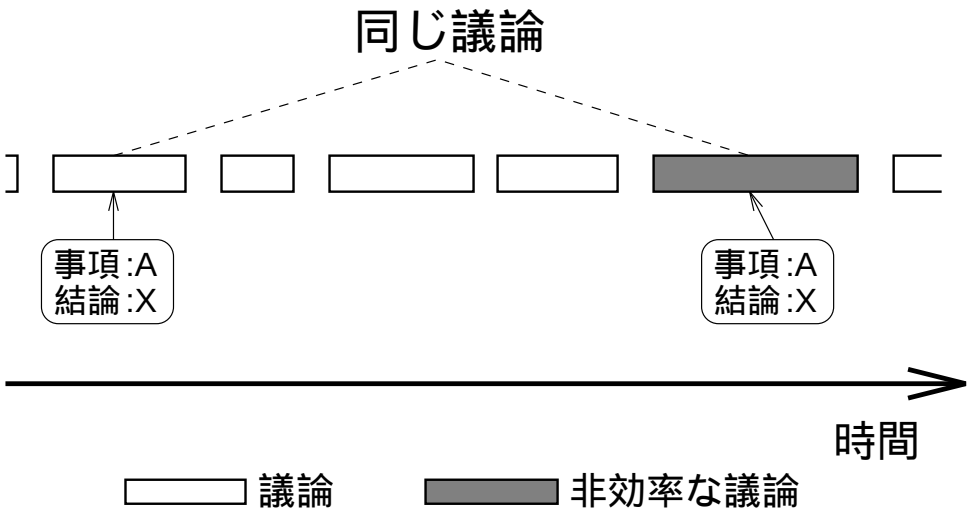


図 2.9: 同じ議論を繰り返す議論

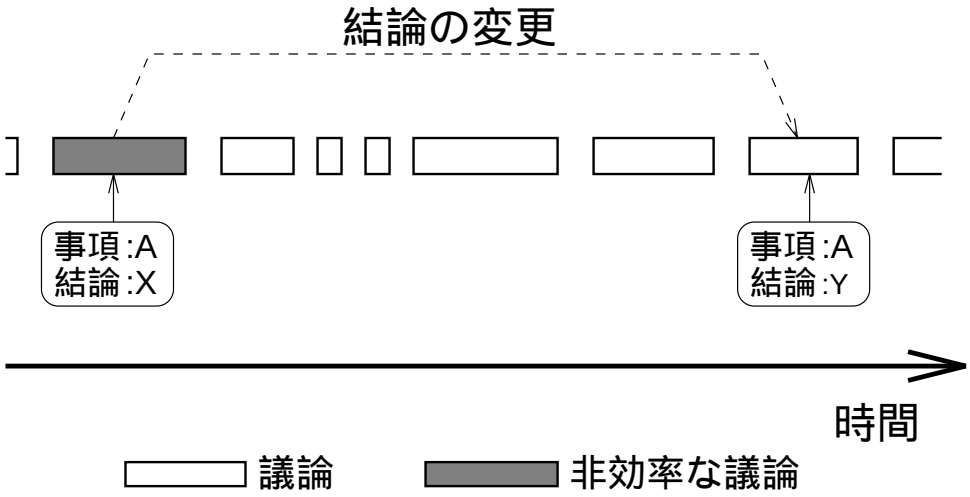


図 2.10: 結論を後の議論で変えられた議論

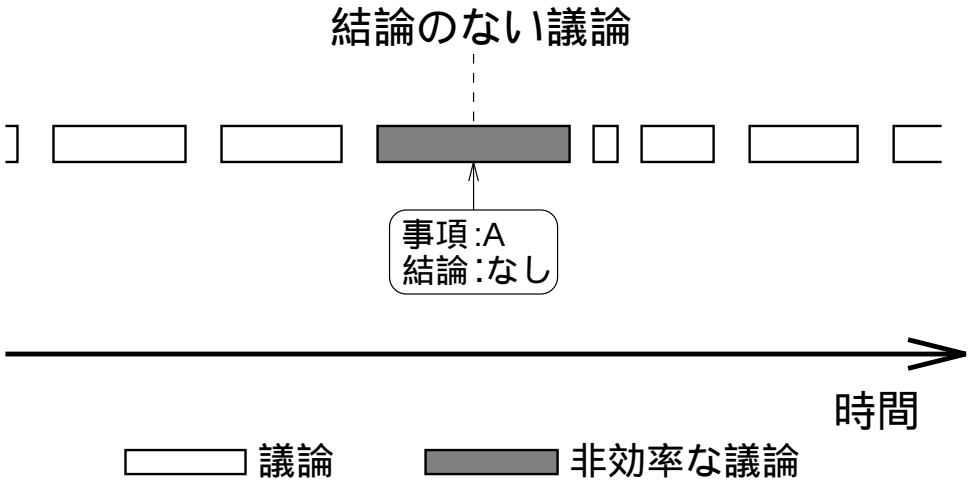


図 2.11: 結論のない議論

i: ハイパートークから, ほかのプログラムからの入出力みたいなのはできるわけ？
かぶせることは.
k: ハイパー トー クでですか？
i: なんか, できるみたいなことがかいてあったけど, あの本にかいてあったのは, 全部さ, 123 の出力してきたファイルを読み込むとかいう形になっていたでしょう？
そうではなくて, 123 そのものにハイパー カー ドを被せることができないのかな？
k: ハイパー カー ドを他のプログラムにとは, どういうことですか？
i: それができなかったら, ほれ, すでにあるデー タベー スを全く利用できないでしょう？
k: デー タベー スなんて使うのですか？
i: だってこれを保存しておくときにさ必要でしょ.
無かったときのことも考えたけど, 既存のデー タベー スがなかったら日付毎に双方向リンクでつなげておこうとかさ.
でも, なんかデー タベー スが必要でしょう？
スケジュー ル, 複数人数分あるから, それに従ってデー タ読み出すわな.
k: 既存のデー タベー スなんてあるのですか？
i: 既存のデー タベー スなんて, マックには腐るほどあるでしょう.

k, i ··· 発話者

図 2.12: 結論のない議論の例

表 2.3: 非効率な議論時間の割合 (%) と議論の件数と図の番号

	同じ議論を繰り返す議論			結論を後の議論で変更する議論			結論のない議論		
	割合	件数	図番号	割合	件数	図番号	割合	件数	図番号
プロジェクト 1	10.7	43	(図 2.13)	7.2	15	(図 2.16)	23.7	72	(図 2.19)
プロジェクト 2	1.2	15	(図 2.14)	0.6	2	(図 2.17)	1.6	14	(図 2.20)
プロジェクト 3	7.7	28	(図 2.15)	3.4	9	(図 2.18)	8.7	16	(図 2.21)

2.3.3 分析結果と考察

表 2.3に非効率な議論時間の割合と, その議論の件数を示す. また表中に示す図の番号 (図 2.13から図 2.21) の図は, それぞれの議論の会議内での分布を示している. 例えば, 図 2.13はプロジェクト 1 の同じ議論を繰り返す議論の分布を表しており, 縦軸に下から上に時間 (秒) をとり, 横軸に事項の番号を示す. また, 縦軸に示す秒は, プロジェクトで複数の会議が開かれた場合は連続しているものとみなし, 会議 1 の開始時点からの時間を示している. 例えば, 番号 2 の事項 ([編集図形で選択, マウスによるクリック] という事項と結論) は, 3100 秒付近と, 9000 秒付近, すなわち会議 2 の 1100 秒付近で同じ議論を繰り返す議論が行なわれていることが, 図 2.13中では楕円で囲んである直線で表現されている. なお, 楕円は説明のために特別に付加したものである.

表 2.3から, 全てのプロジェクトを通して, 結論のない議論, 同じ議論を繰り返す議論, 結論を後の議論で変更する議論の順で多いことがわかる. プロジェクト間の比較では, プロジェクト 1 の結論のない議論がかなり多く (23.7%), プロジェクト 2 が全般に少ない (1.2%, 0.6%, 1.6%).

プロジェクト 1 の特徴 :

プロジェクト 1 はそれぞれの作業者が分担された部分の実装まで責任を持つ点が特徴であるが, 分担が決定されたのは会議 1 の終了時点である. しかし, このプロジェクトでは結論のない議論の多く (72 件中 46 件,64%) は会議 1 で発生している. よって, それぞれの作業者に決定が委ねられたためとは判断できない. しかし, それぞれの作業者がどこを分担するかは別として, 分担を行ない, 分担された部分の実装まで責任を持つ点はプロジェクト開始時点で決定されていたので, 対象ソフトウェアの細かい部分に関しては, 提案程度の形で結論

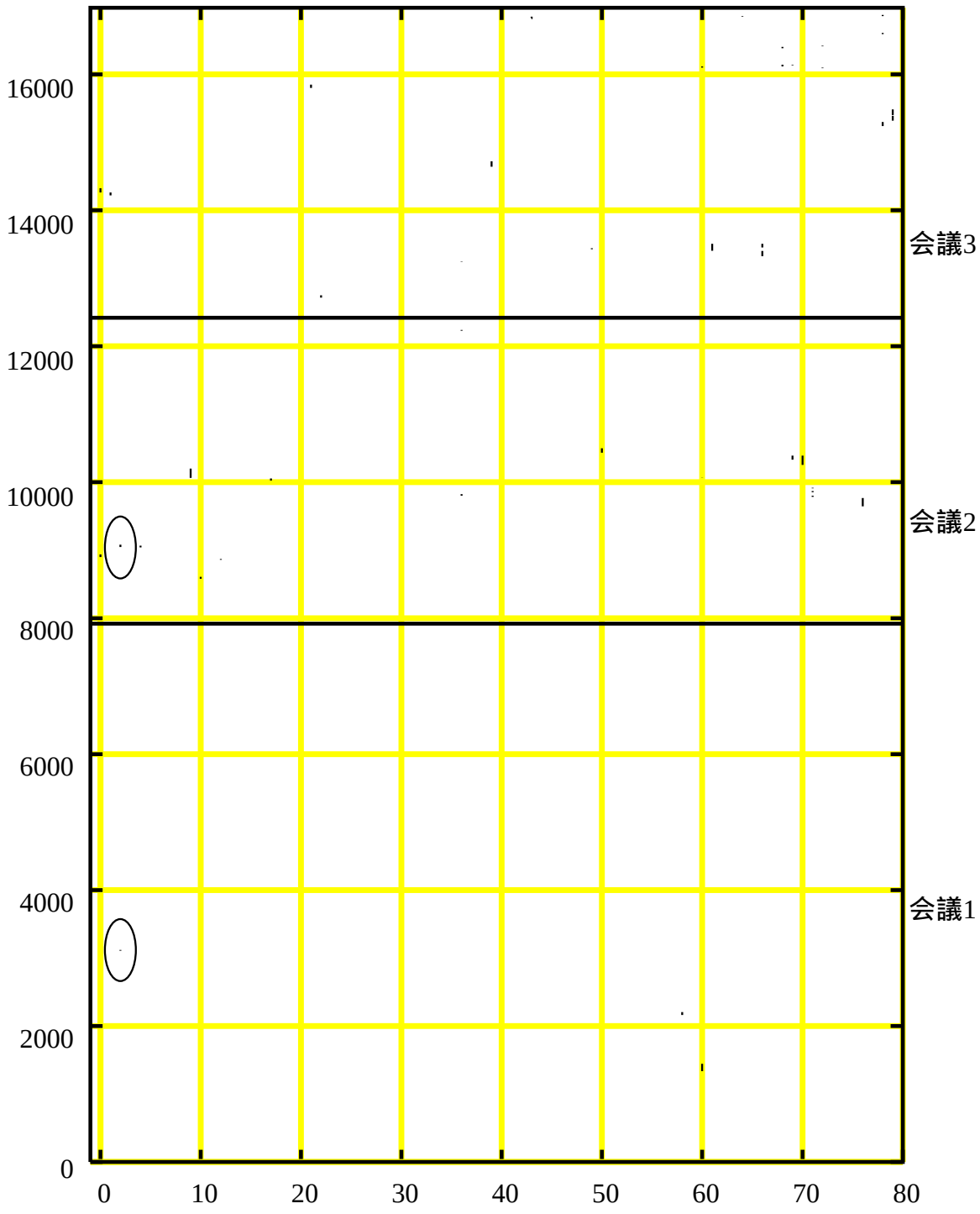


図 2.13: プロジェクト 1 の同じ議論を繰り返す議論の分布

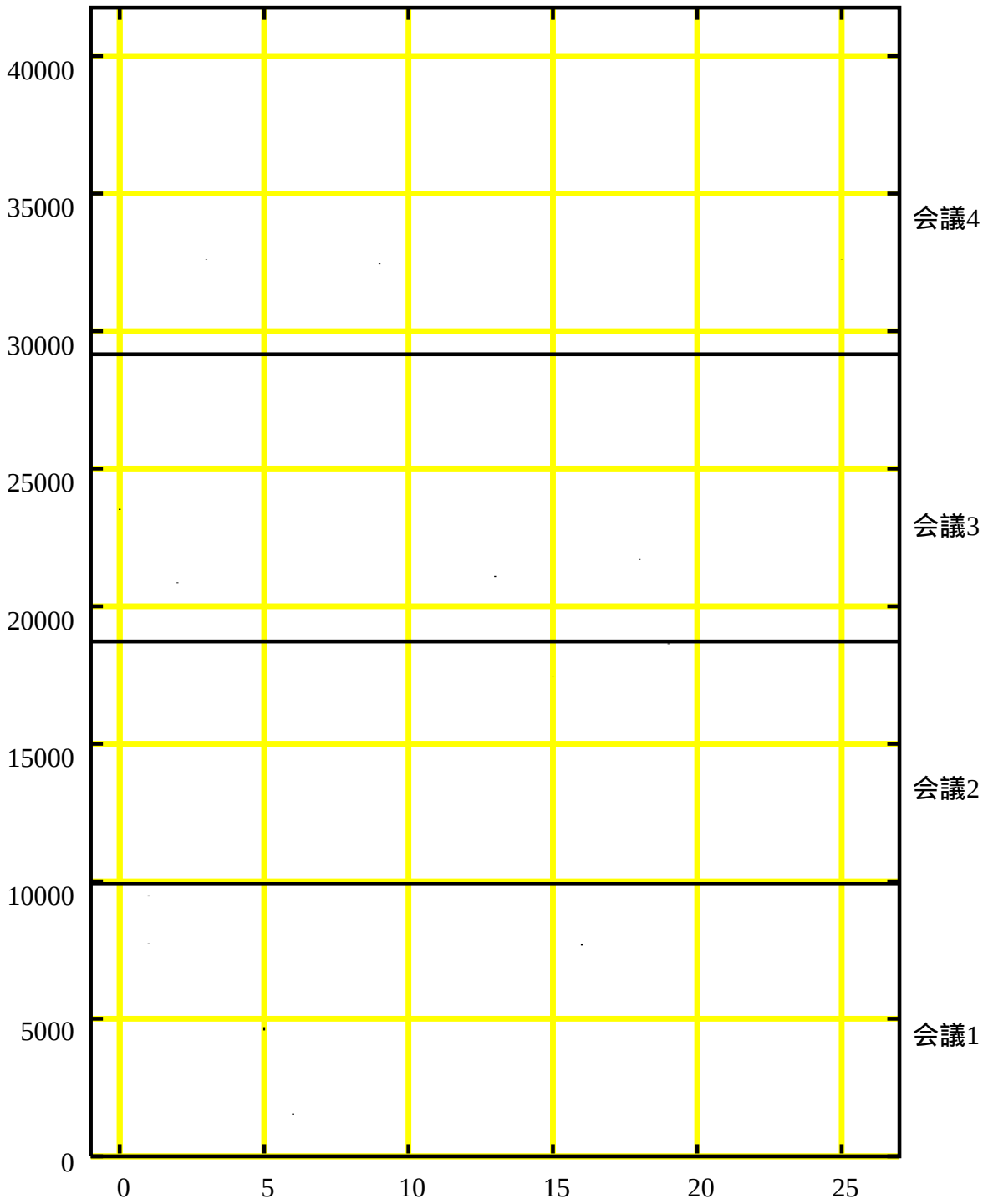


図 2.14: プロジェクト 2 の同じ議論を繰り返す議論の分布

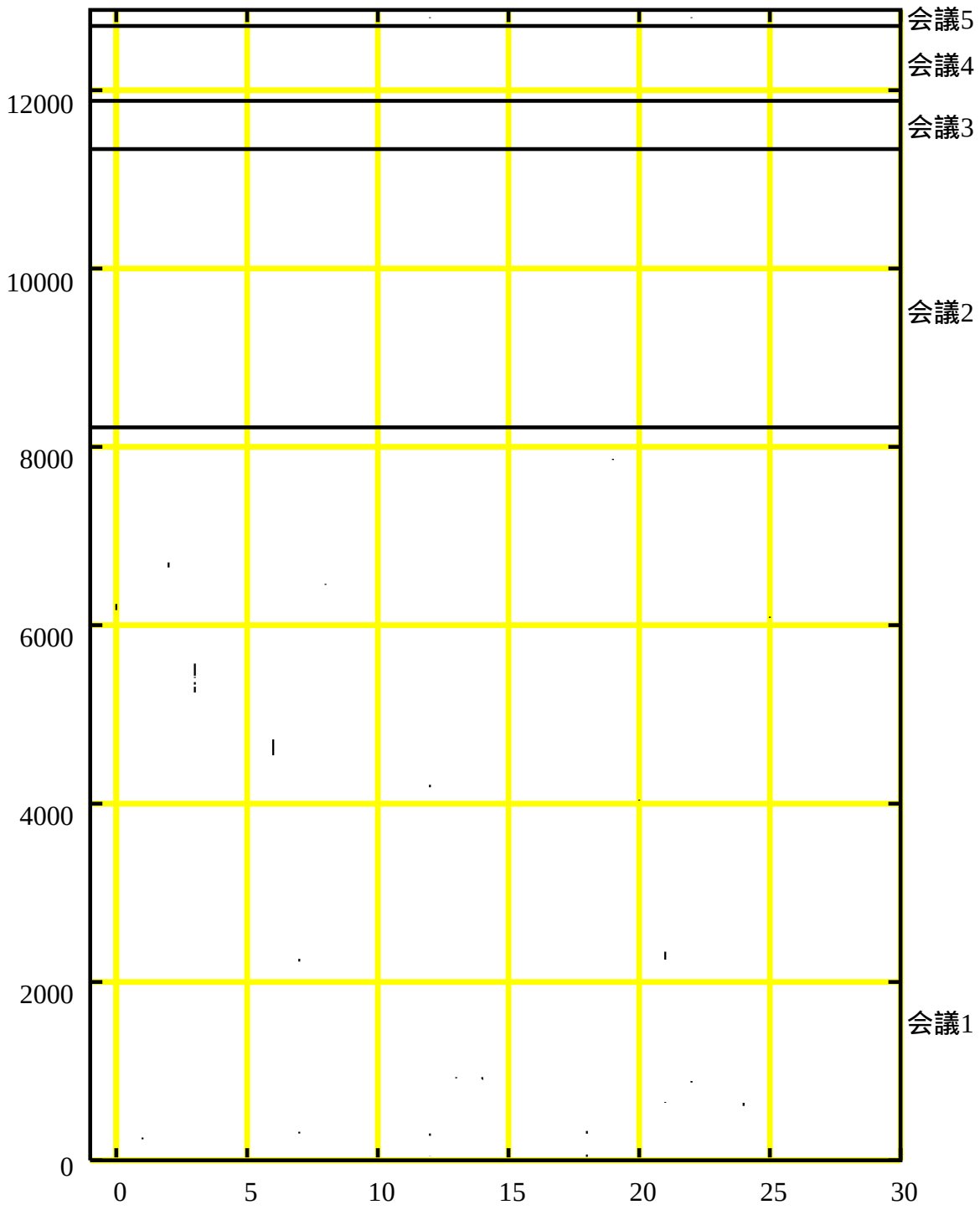


図 2.15: プロジェクト 3 の同じ議論を繰り返す議論の分布

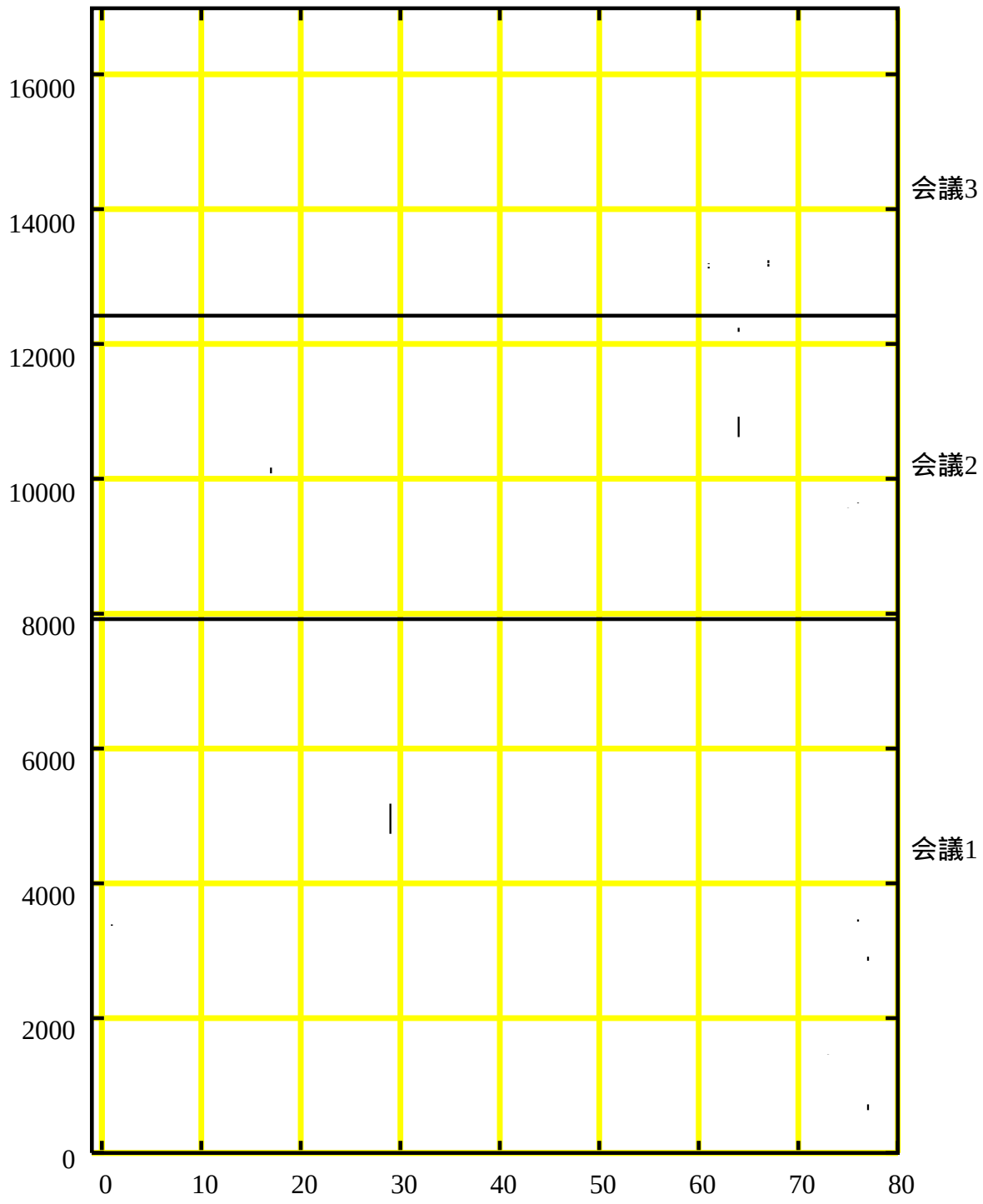


図 2.16: プロジェクト 1 の結論を後の議論で変更する議論の分布

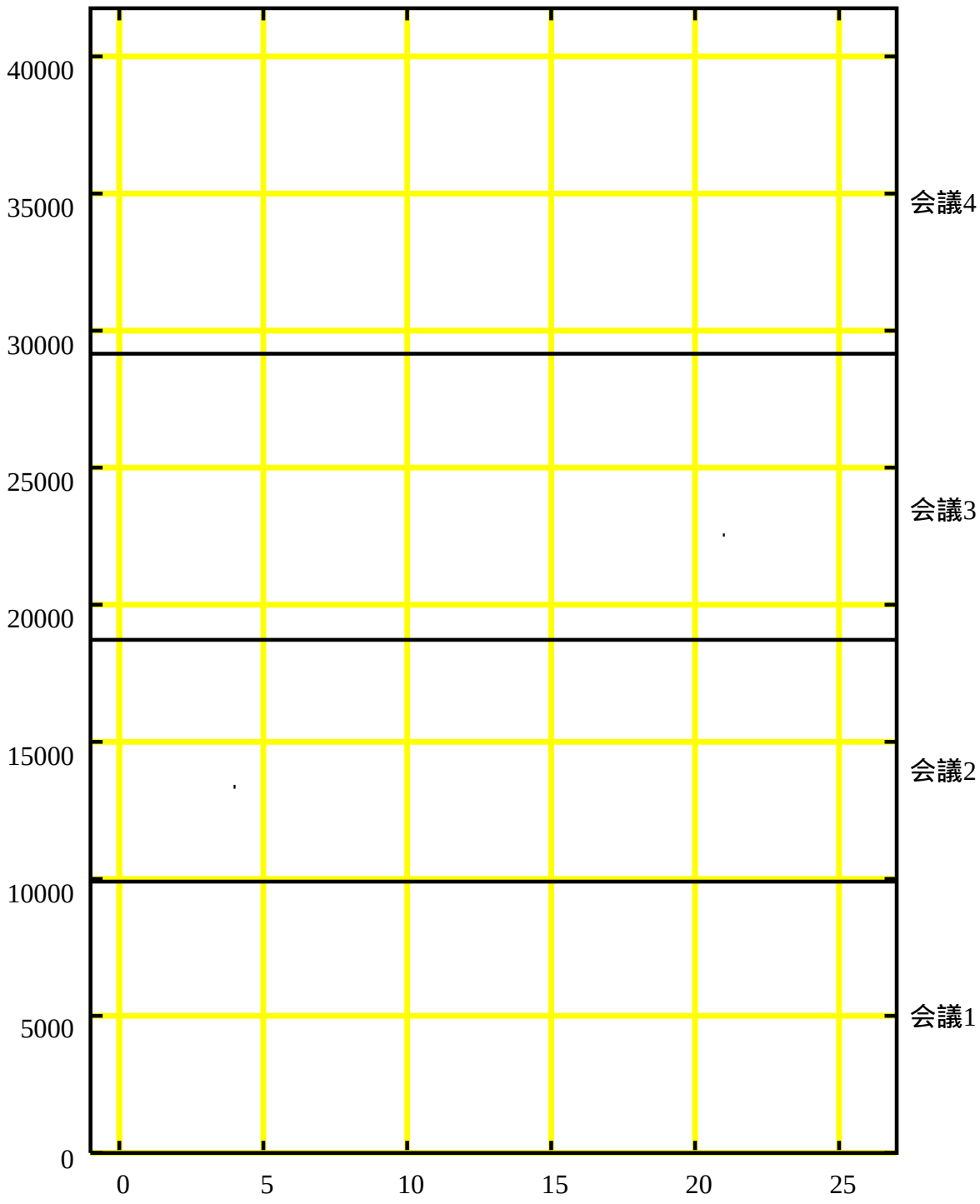


図 2.17: プロジェクト 2 の結論を後の議論で変更する議論の分布

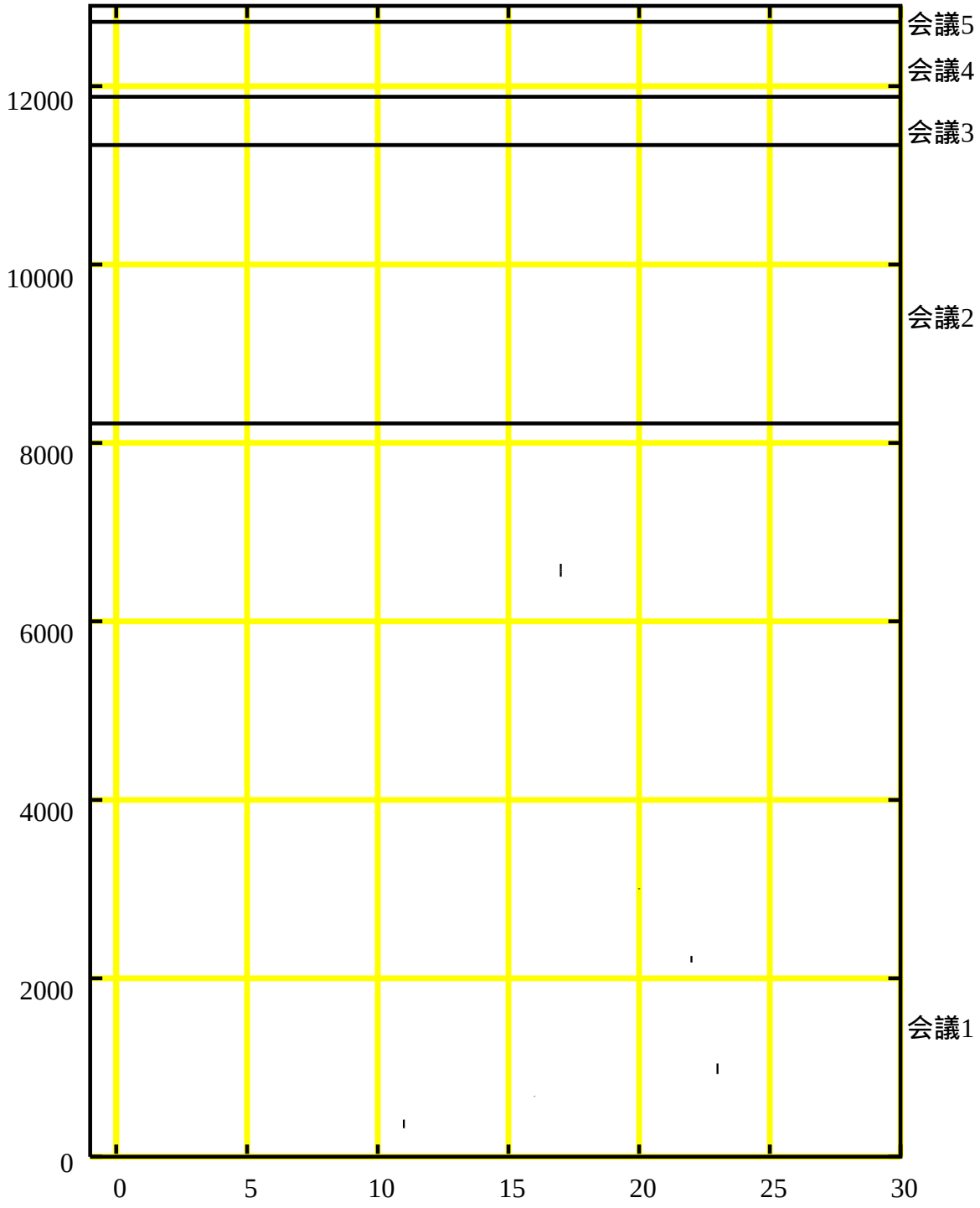


図 2.18: プロジェクト 3 の結論を後の議論で変更する議論の分布

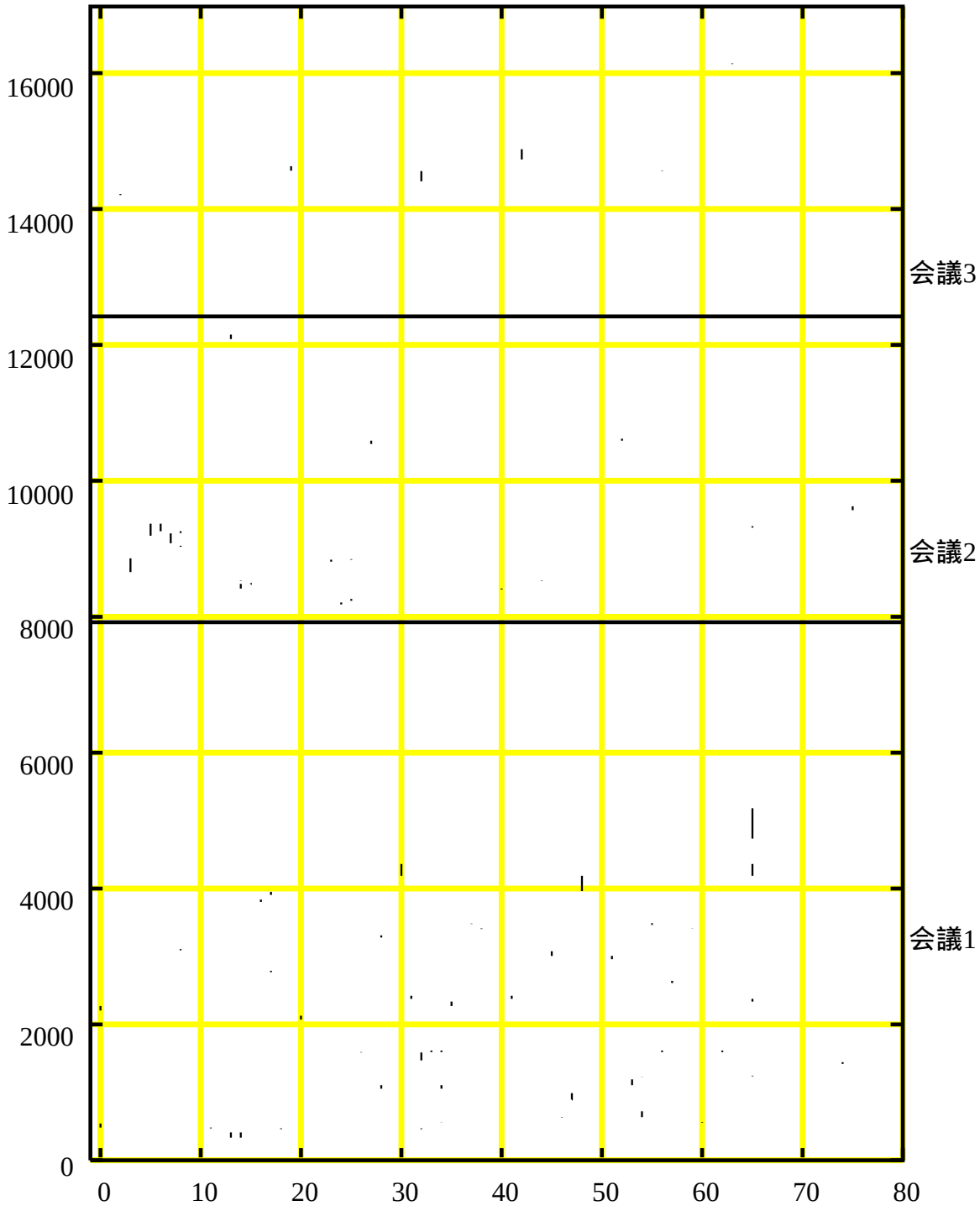


図 2.19: プロジェクト 1 の結論のない議論の分布

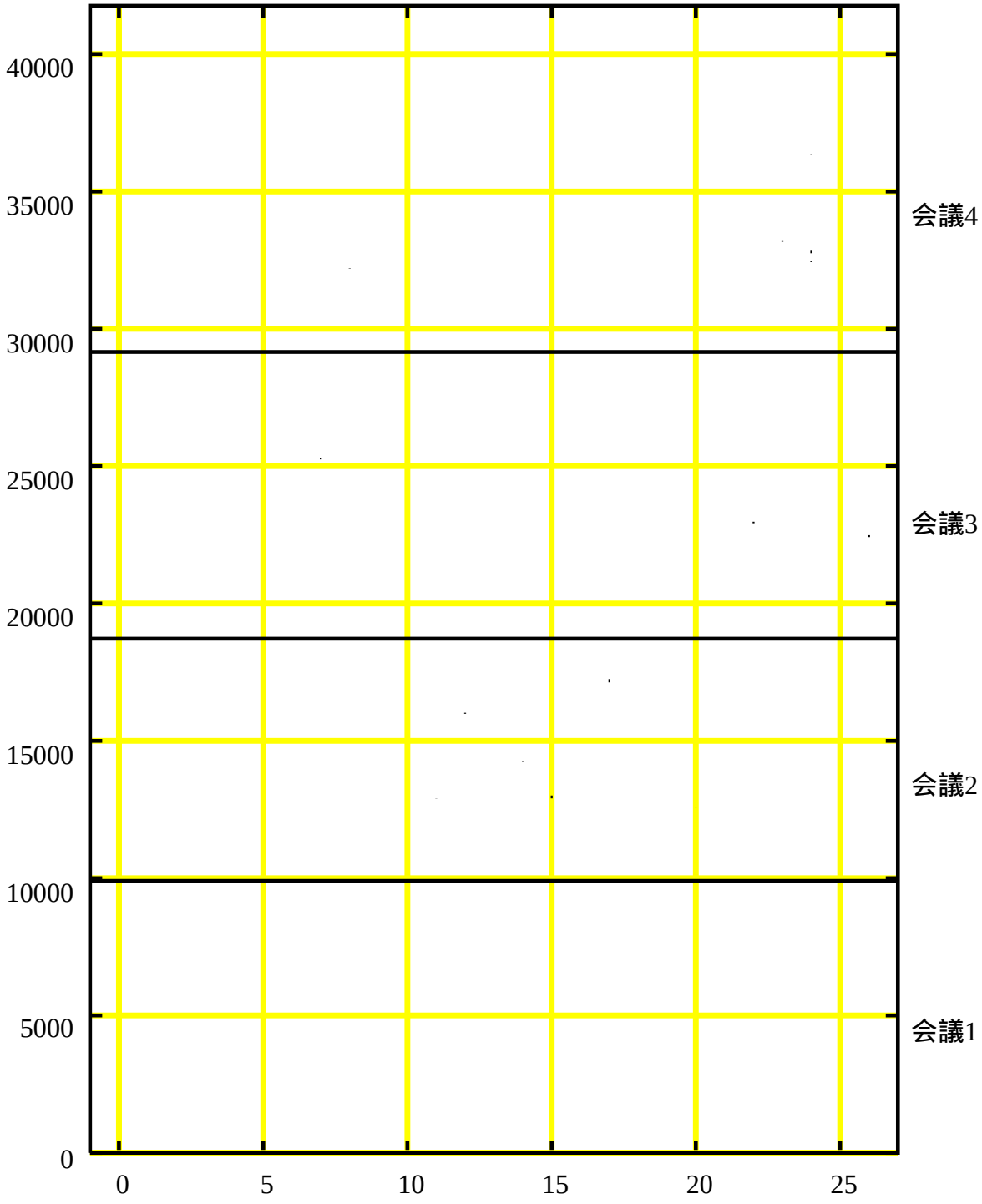


図 2.20: プロジェクト 2 の結論のない議論の分布

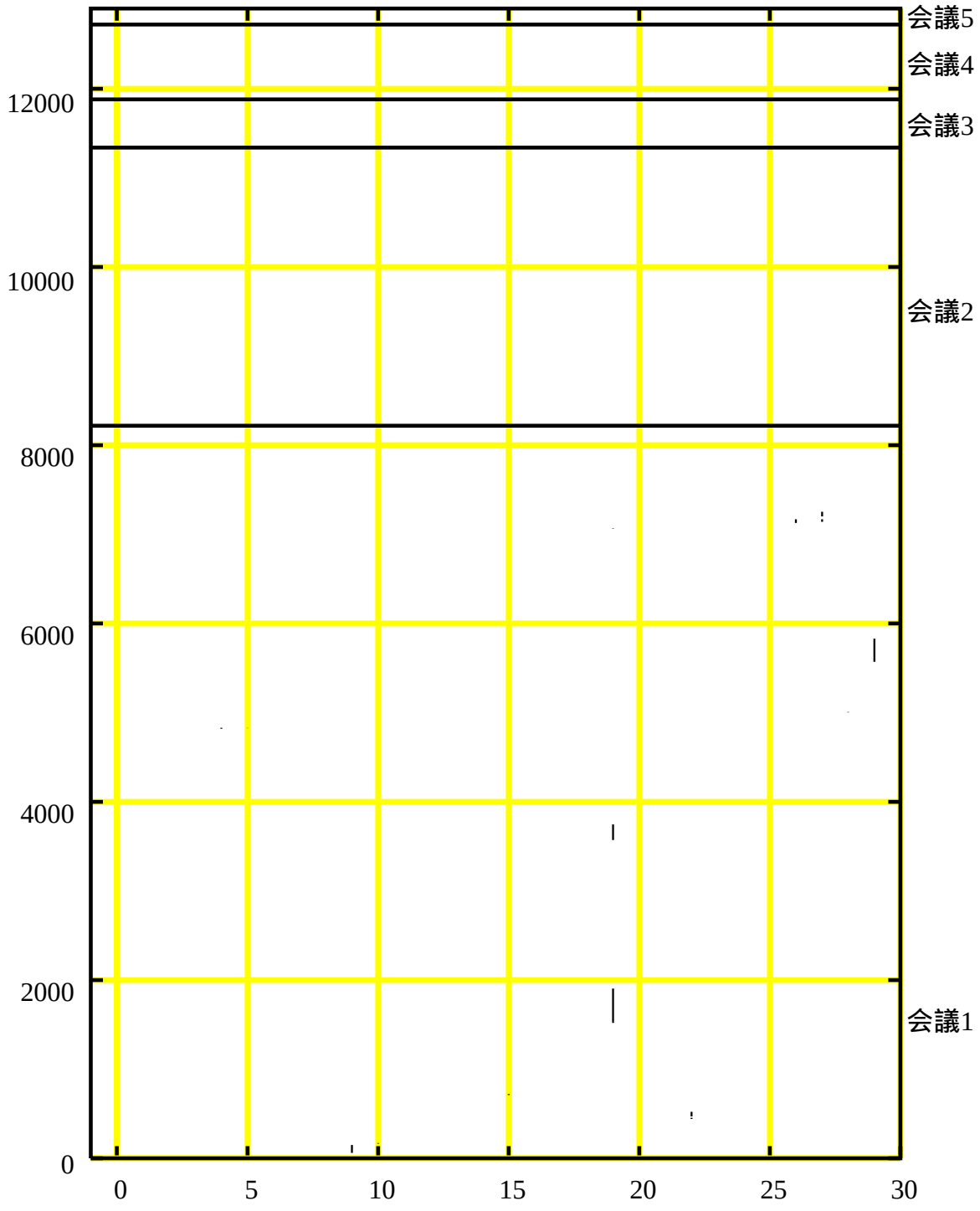


図 2.21: プロジェクト 3 の結論のない議論の分布

を出さなかったと考えられる. 事実, プロジェクト 1 の会議 1 で結論が出なかった議論は対象ソフトウェアの細かい機能に関する部分に関する議論がほとんどであった.

プロジェクト 2 の特徴 :

プロジェクト 2 は書記が参加していたことが大きな特徴である. 書記が発話した議論を調べた結果を表 2.4に示す. 例えば, 全議論数 189 個の内の, およそ 20%に相当する 39 個の議論で書記が発話を行なっている. 書記が発話しても非効率な議論が行なわれる場合があるが, 書記が結論を利用者に提示することは非効率な議論の抑制になっている. ちなにみ, 書記が発話しても行なわれた非効率な議論は, 結論を変更した議論である.

また, 書記の発話が増加するのは, プロジェクトの後半 (このプロジェクトでは会議 4) であるという特徴も見られる. これは, プロジェクトの後半では決定した事項のレビューを行なうという直観と良く一致している.

表 2.4: プロジェクト 2 の議論と書記の発話の関係

	会議 1	会議 2	会議 3	会議 4	合計
議論の数	32	67	54	36	189
非効率な議論の数	5	9	8	9	31
書記が発話した議論	0	7	6	26	39
書記が発話した非効率な議論	0	0	1	0	1

同じ議論の繰り返し :

同じ議論を繰り返す議論の分布の特徴は図 2.13, 図 2.14, 図 2.15からは特別に見られない. 内容の特徴を調べると,

- 確認のために前に行なった議論を繰り返した.
- 前の議論を本当に忘れていて同じ議論を繰り返した.

などが見られた.

結論を後の議論で変更 :

結論を後の議論で変更する議論は事例が少ないこともあり, 図 2.16, 図 2.17, 図 2.18からは特徴が見られない. プロジェクト 1 の結論を後の議論で変更する議論は少し値が大きいが, それは図 2.16の分布に示されるとおり, かなり長い 2 箇所の議論が結論を後の議論で変更する議論となっているからである. その 2 箇所のそれぞれの議論の内容は以下の通りである.

- 29 番 500 秒付近: [塗りつぶし図形の回転, 一緒に動く](会議 1,1:18:55-1:26:22) が, [塗りつぶし図形の回転, 塗ったところは回転せずにそのまま](会議 1,1:26:22-1:26:50) に変更されたものである.
- 64 番 11000 秒付近: [出力形式,Xwindow の著名なツールに準拠](会議 2,45:0-50:0) が, [出力形式, 直接プリンタに出す形式](会議 3,1:13:50-1:13:57) に変更されたものである.

結論のない議論 :

結論のない議論は, 図 2.19, 図 2.20, 図 2.21からプロジェクトの前半が分布が多いことがわかる. これは, 初期の段階ではブレインストーミングなどを行ない, アイディアを提案するのみにとどめ, 決定はしないためであると考えられる. 内容の特徴を調べると,

- 一作業者が提案したのみで他の事項に議論が移動した.
- 明示的に結論を先送りにした.

などが見られた.

2.3.4 まとめ

- プロダクトを基に作業分担を行なう場合は, 会議の初期の段階での結論のない議論である提案が多い.
- 書記の議論での発話は, 非効率な議論の抑制を期待できる. また, 書記の発話はプロジェクト後半に多い.
- 議論の繰り返しは, プロジェクト 2 での書記のように議事録などの記録をとることによって, 解決できると思われる. よって, 会議の記録が簡単に行なえるような支援が必要であると考えられれる.
- 結論を後で変更する議論は, どのプロジェクトでも数値的にかなり小さく, 無視できる範囲だと考えられる. しかし, 実際に変更が行なわれた場合, 変更された部分から影響を受ける他の部分も再度議論し直す必要があると考えられるが, その点については, 2.5節で議論する.
- 結論のない議論は, プロジェクト前半に数多く発生しており, 決定ではない提案段階のアイディアなどが多くみられる. これらを後半の決定をしなければいけない段階で効率的に利用するために, 記録をする必要があると考えられる. また, 決定事項と提案事項を分けるための機構が支援のためには必要だと考えられる.
- 全体の議事録を取っているプロジェクト 2 の場合は, その議事録が全体の合意の記述の代りをしており, 非効率な議論の発生を抑制していると考えられる.

2.4 議論内容の生産物からの欠落

2.4.1 分析の目的

高品質な会議では議論された内容が十分に生産物に反映されている必要がある . よって, 会議の中で議論されている内容が生産物から欠落している現象があれば, それを支援することで減少させなければならない. 会議における情報の欠落を防止するためには, そこから生産されるプロダクトとの対応からプロセスの特性を明らかにする必要がある. 本分析では, 生産物である文書から欠落してしまった会議中の情報の特性を明らかにするため, プロダクトとプロセスとの対応付けを行なう. これによって, どのような種類の情報が, どんな場合に文書から欠落するかが明らかになり, その特性に従った会議の支援を行なうことが可能となると考えられる.

会議での議論内容はできる限り多く文書に反映されることが望ましいと考えられる. よって, 会議で議論されたにも関わらず文書には記述がない部分の発生, すなわち議論内容の欠落を防止する必要がある. 我々は文書の部分とその部分に関する会議中の議論の対応付けによって得られる情報から, 議論内容の欠落の発生を防ぐ手がかりが得られると考えた. その理由は, 文書中の個々の部分の中に欠落しやすい特徴があれば, それに即した支援を行なうことが期待できるからである. ここでは議論内容の欠落の防止に役立つ情報が, 文書と会議の対応付けから獲得できることを述べる.

以下のような情報を獲得し作業者に提示することで, 会議で議論されたにも関わらず文書には記述がない部分の発生を防止できると考えられる. そして, 会議で議論されたにも関わらず文書には記述がない部分の発生の割合が高ければ, その支援の効果は大きいものとなる.

1. 個々の文書の部分の内で欠落しやすい特徴があれば, それに即した支援を行なうことが期待できる. よって, その特徴を獲得, 提示することは有効な支援となりうる.
2. 文書内の, ある部分に関する議論の量は, その部分の欠落のし易さに関係があると思われる. 直観的には会議中に議論される回数が少ない場合には欠落しやすいと考えられる. よって, 文書内のある部分が会議中で議論される回数を作業者に対して提示することが有効な支援となりうる.
3. 文書内のある部分に関する議論が複数ある場合, 会議内におけるそれらの時間的な分布は, その部分の欠落のし易さに関係があると思われる. 直観的にはある部分に関する議論が会議中において時間的に広範囲に分布している場合, その部分は欠落しにくと考えられる. よって, 文書内のある部分に関する議論の分布を作業者に提示することが有効な支援となりうる.
4. 時間的に隣接している議論で議論されている部分は互いに関係があると考えられる. 例えば, 正常動作についての議論の直後に例外動作についての議論が行なわれる場合などが考えられる. よって, 会議中の議論の隣接関係を利用して, 文書の部分間の関係を作業者に提示することが, 欠落しやすい部分を予測するための支援となりうる.

これらの情報は, 文書の部分とその部分に関する会議中の議論を対応付けることで獲得することができる. なお本分析では, 文書とその作成会議の対応に関する問題点を議論するので, 文書内の事項間の関係から生じる不備, 欠落, 矛盾点などは扱わない.

2.4.2 分析の手順

以下に示す手順に従い, 文書の分析単位である事項と, 会議の分析単位である議論を決定し, その対応付けを行なう.

1. 文書を事項に分割する.
2. 会議を議論に分割する.
3. 会議の日程や作業者の作業分担などの作成対象となるソフトウェアに関する部分以外の議論は除外する. この分析は文書を基に分析を行なうため, 対象システム自身に直接関係のない部分は対象外として扱う.

4. 議論の内容がある事項に関する議論である場合に, その議論と事項の対応を付ける. 1 つの事項は複数の議論と対応付けられても良い.
5. 議論と対応が付かなかった事項を除外する. このような事項は, 会議での議論を通して作成されたのもではないので, 本分析の対象外として扱う.
6. 事項と対応が付かなかった議論から, その内容を示す文章を作成する. ここで作成された文章を欠落事項と呼ぶ. 欠落事項と, それを作成する基になった議論との対応を付ける. ここでの議論と対応させることのできる欠落事項が既に作成されている場合は, 新しい欠落事項は作成せずにその既存の欠落事項と対応付けを行なう.
7. ある事項に対応する議論の中で, 最も時間的に遅い議論においてその事項が否決されている場合, その事項と対応する議論の両方を除外する.
8. 以上で除外されなかった議論全てを分析対象とする. 1. で作成した事項と 6. で作成した欠落事項を合わせたものの内で除外されなかったものを分析対象とする.

図 2.22に文書と会議の対応の例を示す. 実線の矩形で囲まれた事項 1 から 4 は, 実際の文書に含まれていたものであり, その中の事項 2 は, 会議中の議論に出現しないため除外している. 事項 a, b は文書に記述はない欠落事項であり, 会議の議論を基に作成されたものである. 図中の 8 つの議論のうち, 議論 3 は会議の日程などについての議論なので除外とする. 除外されていない 7 つの議論の内, 議論 5, 6, 7 は欠落事項 a,b と対応する. また, この例では 40%(2/5) の事項が欠落しており, 欠落していない事項では平均 1.3 回 (4/3) の議論と対応しているのに対して, 欠落している事項では平均 1.5 回 (3/2) の議論と対応している. 事項 3 と事項 a を除いては, 事項と議論は 1 対 1 対応である. 事項 3 は議論 2,8 と対応しており, 議論 2 は会議の前半に行なわれており議論 8 は一番最後に行なわれている. また, 事項 a は議論 5,7 に対応しており, この 2 つの議論は比較的時間的に近接して行なわれている. 事項 3 に対応している 2 つの議論を, 時間的に広範囲に分布している議論と呼ぶ場合がある.

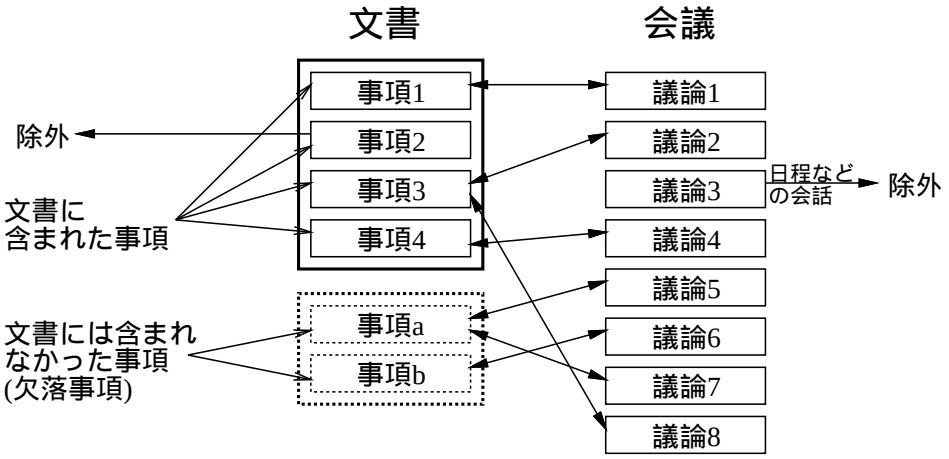


図 2.22: 文書と会議の対応

具体的な文書と会議の対応付けの例を, 2.1.2 節で紹介した文書の例 (図 2.2と図 2.3) と, 議論の例 (図 2.5) を利用して説明する. 表 2.5に議論と事項の対応を示す. 議論 4 に対応する事項が文書内に存在しないため, 例えば,

表 2.5: 議論と事項の対応

図 2.5中の議論番号	1 2 3 4
図 2.3中の事項番号	3 6 7 該当なし

図形の移動: ラバーバンドは同形の図形を用いても良いが, 外形の矩形表示程度で問題ない.

のような事項を議論 4 から作成する.

2.4.3 分析結果と考察

1. 欠落事項の割合:

   表 2.6に, それぞれのプロジェクトにおける事項の欠落とそれに関する議論の割合を示す. 表の第 3 列 (全個数) が分析の対象となる事項全ての個数であり, 図 2.22での事項 1,3,4,a,b に対応する. 第 2 列 (欠落した個数) は第 3 列の事項の内の欠落事項の個数であり, 図 2.22では事項 a,b に対応する. 第 4 列 (全回数) は会議中で分析対象となる議論の数であり図 2.22では議論 1-2,4-8 に対応する. 第 5 列 (欠落事項に関する議論数) は第 4 列の議論の内で文書に反映されなかった議論の数であり図 2.22では議論 5,6,7 に対応する. 例えば, プロジェクト 2 では, 分析対象となる事項は 130 個あり, そのうちの 43 個は文書には現れなかった欠落事項である. また, 分析対象となる議論は 189 個あり, そのうちの 53 個は文書には反映されなかった議論である.

   表 2.6: 結果の比較

	事項		議論	
	欠落した個数 (%)	全個数	全回数	欠落事項に関する議論数 (%)
プロジェクト 1	82(62)	132	217	125(58)
プロジェクト 2	43(33)	130	189	53(28)
プロジェクト 3	37(39)	96	133	46(35)

   この表より, どの種類のプロジェクトでもかなりの事項の欠落があることが分かる. プロジェクト 2 は欠落の割合が低く, 議事記録を専門に行なう作業者の存在が会議における情報の欠落を防ぐという直観の根拠となりうる. プロジェクト 1 と 3 では, 作業者の種類の分類方法を共にプロダクトを基にしているにもかかわらず, 欠落の割合がかなり異なっている. これは, それぞれの分担に対する作業者の責任が異るためであると考えられる. 具体的には, 会議の設定として, プロジェクト 1 ではソフトウェアをプログラミングする部分まで分担の責任に含まれているのに対し, プロジェクト 3 では文書を作成する部分までの責任が要求されている. よって, プロジェクト 1 では作業者自身が理解していればよい部分の仕様は文書として文書化されず, プロジェクト 3 では後段階の作業者のために議論された内容が比較的詳細に文書化されたため, このような差異が生じたと考えられる.

2. 個々の欠落事項の内容的な特徴: 決定が不明確な事項

   会議の中では個々の作業者の発言によって提案されてはいるが, 他の作業者によるそれに対する意見などがない決定か未定か不明確な議論がいくつか見らた. 一般には会議の決定事項は会議参加者の合意によって行なわれると考えられる. 2.3節では, プロセスのみに注目する観点から, このような議論を結論のない議論と定義したが, 実際の文書には, このような決定か未定かが不明確な事項も記述される場合がある. このような決定が不明確な事項は当然, 仕様から欠落しやすいと考えられる. 表 2.7に, 会議において決定が不明確な事項の割合を, 文書から欠落している事項とそうでない事項とに分けて示す. 括弧内の数値は事項の実個数である . 例えば, プロジェクト 1 の欠落事項 82 個の内の, 74.4%にあたる 61 個は会議においての決定が不明確であった事項である. 最右列には , 欠落事項と欠落していない事項を合わせた全事項の中の , 不明確な事項の割合を示している . 例えば , プロジェクト 1 では , 全事項の中のおよそ半数に当たる , 49.2%が不明確であったことが示されている .

   どのプロジェクトでも, 欠落事項のほとんどが決定か否かが不明確である場合が多いことがわかる. よって, 提案のままの決定が不明確な事項などの発生を抑制することが欠落事項の防止につながると考えられる.

3. 個々の欠落事項の内容的な特徴: 孤立した事項.

   新しい事項を文書に追加する場合, 追加する適当な位置を特定しにくい場合や, 文書中の複数の部分と関係があるため追加する場所を一意に特定しにくい場合がある. このような性質を持つ事項を孤立した事項と呼ぶことにする. 文書から欠落してしまった事項は, 孤立した事項であるために, 文書にとりこまれなかった可能性がある. もし, この仮定が正しければ, 孤立した事項となるような事項に注意を払うことが, 文書からの議

表 2.7: 会議において決定が不明確な事項の割合 (%)

	欠落した事項		欠落していない事項		全体
プロジェクト 1	74.4	(61)	8.0	(4)	49.2
プロジェクト 2	65.1	(28)	12.6	(11)	30.0
プロジェクト 3	40.5	(15)	3.4	(2)	17.7

論内容の欠落の防止となる. よって, 実際に欠落してしまった事項中の孤立した事項の割合を調べる. 分析のために, 以下の 3 種類の孤立した事項に注目する.

追加位置の特定が不可 : 文書内に併記するのにふさわしいと思われる場所が存在しない欠落事項.
例えば, プロジェクト 1 の描画エディタでは,
入出力部分, 画面部分, 文字部分, 編集部分, 描画部分
のように文書の分担を行なっているが, “建築用などの用途による特殊化” を特定の部分に含めることが困難である.

追加位置の欠落 : 追加位置の特定が不可な事項ではあるが, 他の欠落事項を文書に追加することにより, 併記するのにふさわしいと思われる場所を確保できる事項.
例えば, プロジェクト 1 の描画エディタでは, 文書内に “文字の修飾” という事項が存在しないため, “フォントの切替え”, “下線” などの事項を特定の部分に含めることが困難である.

追加位置の一意特定が不可 : 文書内に併記するのにふさわしいと思われる場所が複数存在する欠落事項.
例えば, プロジェクト 1 の描画エディタでは, “塗りつぶし図形を回転させると塗った部分も回転する” などの事項は, 描画部分の “塗りつぶし図形” もしくは編集部分の “回転” のどちらにも併記できるため, 特定の部分に含めることが困難である.

表 2.8 に仕様の中で孤立した事項になりうる欠落事項の割合を示す. 括弧内の数値は事項の実個数である . 例えば, プロジェクト 1 の欠落事項 82 個の中の 39 個に当たる 47.6%が, 追加位置の一意特定が不可である孤立した事項である. ただし , 最右列のみ , 欠落していない事項を含めた全事項の中での , 仕様の中で孤立した事項の割合を示している . 仕様書から欠落していない事項は , その定義から , 全て孤立していない事項とみなす .

仕様の中で孤立した事項になりうる欠落事項の割合は, 全てのプロジェクトで多いとは言えないがいくつかの特徴が見られる.

- プロジェクト 1 の追加位置の一意特定が不可な孤立した事項がかなり多い. プロジェクト 1 では, 対象システムを 5 つの部分に分割して, それぞれの作業者が仕様を記述することになっていたため, 誰の領域の問題かが不明確な部分は, 互いに自分の領域の問題ではないと判断して, 文書として記述をおこなわなかったことが考えられる.
- プロジェクト 3 の追加位置の欠落している孤立した事項が多い. 内容を調べると, “行事の飛びいり参加者の扱い” と, “グループユーザー” に関する事項が欠落したため, これらの事項下に記述されると思われる事項が欠落してしまったことが, 追加位置の欠落している孤立した事項の発生の原因となっている.

表 2.8: 仕様の中で孤立した事項になりうる欠落事項の割合 (%)

	追加位置の特定が不可	追加位置の欠落	追加位置の一意特定が不可	全事項中での割合
プロジェクト 1	3.7 (3)	4.9 (4)	47.6 (39)	34.8 (46)
プロジェクト 2	9.3 (4)	2.3 (1)	9.3 (4)	6.9 (9)
プロジェクト 3	8.1 (3)	27.0 (10)	8.1 (3)	16.7 (16)

4. 個々の欠落事項の内容的な特徴: その他

個々の欠落した事項の内容的な特徴として以下のようなものが見られた. 具体的にこれらの特徴を持った欠落事項の数を表 2.9に示す. 例えば, プロジェクト 3 において特徴 3 に該当する欠落事項は 5 事例あった.

表 2.9: 個々の欠落事項の内容的な特徴

	全欠落事項数	特徴 1	特徴 2	特徴 3	特徴 4	特徴 5
プロジェクト 1	82	15	4	3	7	1
プロジェクト 2	43	2	7	3	1	4
プロジェクト 3	37	8	4	5	1	2
合計	162	25	15	11	9	7

特徴 **1.** 例外処理に関する欠落

通常処理以外の処理に関する事項. 特に, 他の機能と組み合わせることによって生じる場合が多い.

特徴 **2.** 暗黙的な部分の欠落

作業者の操作に応じて, 状態を変えなければいけない部分や, 機能の初期状態, 終了状態での処理などに関する事項.

特徴 **3.** 用語の定義に関する欠落

用語の言い替えなどが文書から欠落していたり, 文書には用語のみの記述があり, その内容が欠落している場合.

特徴 **4.** 複数の選択肢を持つ機能の欠落

ある機能を決定する場合に複数の案が提案された事項.

特徴 **5.** 他の機能で補間できる部分

よって, これらの特徴を持った事項の議論に注目し, その事項を記録することが有効だと思われる.

5. 事項に対応する議論の量:

表 2.10に事項当たりの議論数を, その事項が文書から欠落した場合と欠落していない場合に分けて示す. 表の第 2 列は 1 つの欠落事項に対応する議論の回数の平均を示し, 第 3 列は 1 つの欠落していない事項に対応する議論の回数の平均を示している. また, 表の第 4 列は全ての欠落事項の内で 1 回のみの議論と対応のつく事項の割合を示し, 表の第 5 列は全ての欠落していない事項の内で 1 回のみの議論と対応のつく事項の割合を示している. 例えば, プロジェクト 2 の欠落事項は, 平均して 1.2 回議論されており, 全ての欠落事項 43 個の内の 35 個, すなわち 81%が 1 回のみの議論で議論された事項であることを示している. それぞれのプロジェクトとも文書から欠落した 1 事項当たりの平均議論数は欠落していない事項 1 つ当たりの平均議論数に比べてそれほど大きな差ではないが小さく, 複数回議論された事項は文書から欠落しにくいという直観には反しない. また, 表の第 4,5 列から, 欠落事項は欠落していない事項に比べ, 1 回のみ議論された事項の割合が高いことがわかる. よって, 議論の回数が少ない場合の事項は欠落し易い傾向があり, 特に 1 回のみ議論された事項は, その傾向がより強いと言える. よって, 議論された回数の少ない事項を提示することで, 作業者に対しその部分の欠落に注意を払わせることが可能であると思われる.

表 2.10: 事項当たりの議論数

	1 事項当たりの平均議論回数		1 回のみ議論された事項の割合 (%)	
	欠落した事項	欠落していない事項	欠落した事項	欠落していない事項
プロジェクト 1	1.5	1.8	67(55/82)	46(23/50)
プロジェクト 2	1.2	1.6	81(35/43)	67(58/87)
プロジェクト 3	1.2	1.5	81(30/37)	73(43/59)

6. 事項に対応する議論の時間的な分布:

   図 2.23に各事項に対応する議論の時間的な分布を 3 つのプロジェクト別と欠落事項に対応する議論とそうでない議論別の 6 つのグラフに分けて示す. 個々のグラフ中の横軸は時間の流れ (秒) を示し, プロジェクトに含まれる複数回の会議を連結して表示している. 縦軸は事項を表しており, 例えば図中の左上の “プロジェクト 1” の “欠落していない事項” の議論に関するグラフでは 50 個の事項と, 合計時間がおよそ 12600 秒である 2 回の会議を対象としていることがわかる. グラフ中のプロットは議論に対応し, 同一事項に関する議論は見やすさのために点線で結んである. 例えば, “プロジェクト 2” は 4 回の会議から成るためグラフは横軸方向に 4 分割されている. “プロジェクト 2” の “欠落している事項” の議論のグラフでは, 番号 3, 10, 11, 26, 29, 40, 42, 43 の 8 個の事項が, この 4 つの会議を通して複数回議論がなされているが, グラフから目視で分布が分かるのは 3, 10, 26, 40 の 4 つ程度である. それに対して, “欠落していない事項” のグラフでは 29 個の事項が, この 4 つの会議を通して複数回議論がなされているが, その内のかなりの事項に関する議論が, この 4 つの会議中に広範囲に分布しているのが分かる. プロジェクト 1 を除いては, 明らかに欠落していない事項に関する議論の方が, 時間的に広い範囲に分布していることが分かる. よって広範囲に分布する議論と対応のつく事項は欠落しにくいと考えられる. プロジェクト 1 の場合, 欠落事項に関する議論もそうでない事項に関する議論も, ほぼ同程度に分布しているが, 母集団となっている事項の数が欠落事項に関する議論の方が多いため, 同程度に分布してしまった. よって, 議論の分布などを図 2.23のように提示することにより, 広範囲に分布していない事項に対して作業者が注意を払う助けとなると思われる.

7. 欠落事項と他の事項との関係の特徴:

   欠落事項間の関係を, それぞれの事項に対応する議論の時間的な隣接関係を基にして特徴を調べる. 図 2.24にそれぞれのプロジェクト中の議論の時間分布を示す. プロジェクト 2 のみ 10 分間隔で時間を区切り, その他は 5 分間隔で時間を区切っている. 欠落した事項に関する議論の分布も欠落していない議論の分布も, 議論が集中している箇所がいくつか見られるが, 全体的にはそれらの発生はそれほど偏っていない. 特に, プロジェクト 1 と 3 での作業者の種類の決定 (会議 1 の終了時) の前後においても, 欠落の発生に関する大きな差異はみられない. プロジェクト 1 では欠落した事項に関する議論の集中する部分が数箇所 (1200-1500, 3300-3600, 4500-4800, 9600-9900 など) あるが, それらに含まれる議論の内容を調べると “文字” に関する部分であり, この部分の担当者に依存して欠落が発生したと考えられる. プロジェクト 2 では, 欠落していない事項に関する議論が集中している箇所があるが (14400-16200 など), そこでは主にシステムの動きを追跡することで議論を行なっている. 以下に議論の時間的な順序や隣接関係による欠落の特徴を示し, その位置を図 2.24に示す.

   特徴 **1.** 一般的な機能の議論の後に, その特殊な例や例外が続くと後者が欠落する. (4 事例)
   特徴 **2.** 機能の組合せについての議論では, その組合せに関する事項が欠落する. (5 事例)
   特徴 **3.** ある機能の議論の後に, その機能から連想された部分が議論されると後者が欠落する. 例えばユーザインタフェースを介して関係する部分など. (5 事例)
   特徴 **4.** 付加的な機能が後に続いて議論されると, それが欠落する. (3 事例)

   よって, 会議中の議論と議論の隣接関係などを手がかりに, 関連のある事項を提示することで, 議論内容の欠落を予防することが可能であると思われる.

2.4.4 まとめ

ソフトウェアの仕様作成会議における生成物としての文書と会議内の議論を対応付けることで, 文書に含まれなければならない情報の多くが文書となる時点で欠落していることを, いくつかの異なった種類のソフトウェア開発プロジェクトの分析を通して明らかにした. 分析の結果を以下に列挙する.

- 議論内においての決定の不明確な事項ほど欠落しやすい傾向にある.
- 文書中で追加する適当な位置を特定しにくい事項や, 文書中で追加する場所を一意に特定しにくい事項などの孤立する事項の欠落がある程度見られる.

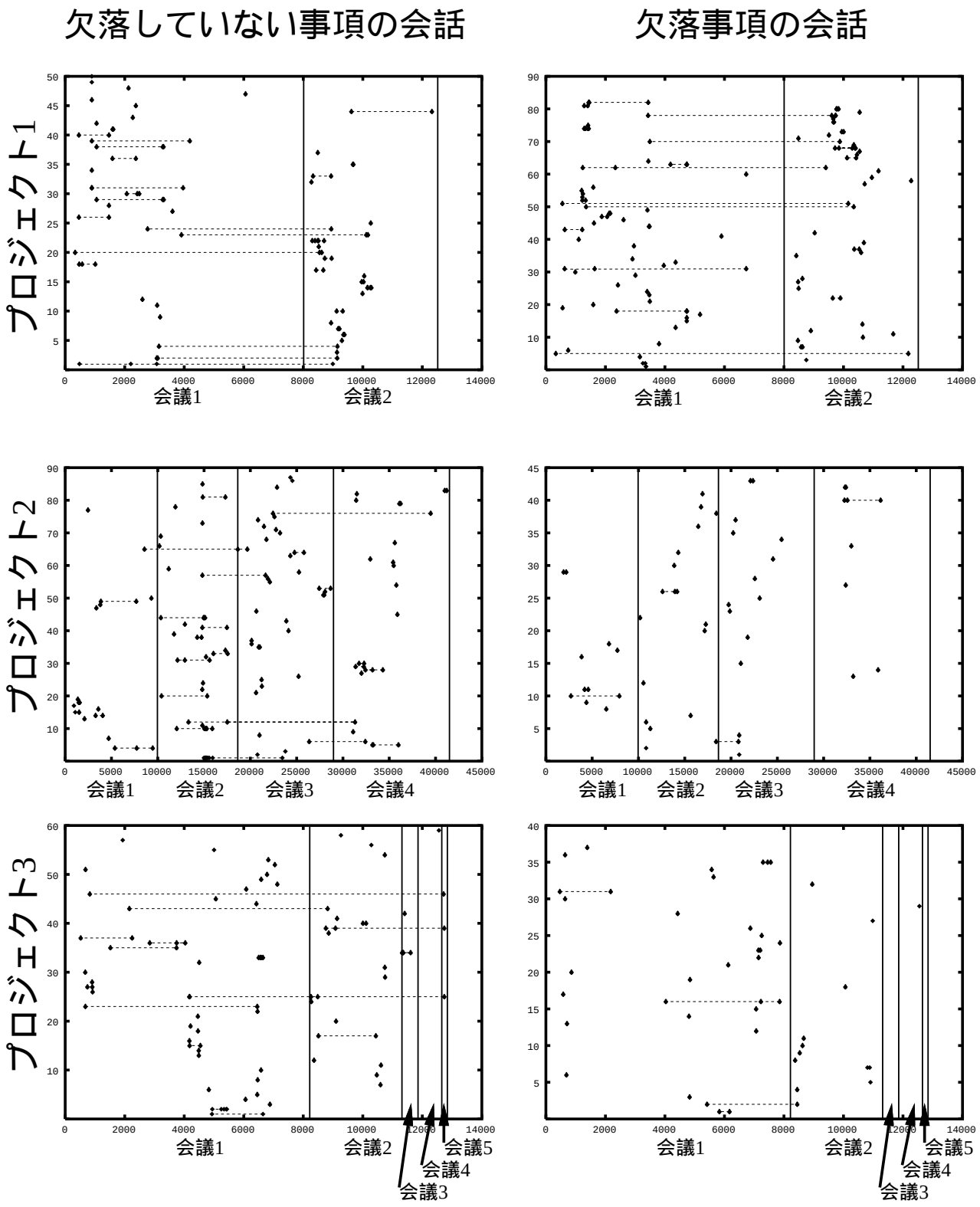


図 2.23: 各事項に対応する議論の時間的な分布

プロジェクト1

```
会議1
# -  300  0+ 0= 0
# -  600  3+ 6= 9 ---******
# -  900  3+ 7=10 ---*******
# - 1200  2+ 4= 6 --****
# - 1500 18+ 3=21 ------------------***
# - 1800  4+ 3= 7 ----***
# - 2100  2+ 1= 3 --*
# - 2400  4+ 5= 9 ----*****
# - 2700  2+ 3= 5 --***
# - 3000  2+ 1= 3 --*
# - 3300  3+ 8=11 ---********
# - 3600 13+ 2=15 -------------**
# - 3900  1+ 1= 2 -*
# - 4200  2+ 3= 5 --***
# - 4500  2+ 0= 2 --
# - 4800  8+ 0= 8 --------
# - 5100  0+ 0= 0
# - 5400  1+ 0= 1 -
# - 5700  0+ 0= 0
# - 6000  1+ 0= 1 -
# - 6300  0+ 1= 1 *
# - 6600  0+ 0= 0
# - 6900  2+ 0= 2 --
# - 7200  0+ 0= 0
# - 7500  0+ 0= 0
# - 7800  0+ 0= 0
# - 8100  0+ 0= 0

会議2
# - 8400  0+ 4= 4 ****
# - 8700  8+ 9=17 --------*********
# - 9000  2+ 6= 8 --******
# - 9300  1+ 7= 8 -*******
# - 9600  2+ 3= 5 --***
# - 9900 14+ 3=17 --------------***
# -10200  4+ 8=12 ----********
# -10500  8+ 3=11 --------***
# -10800  8+ 0= 8 --------
# -11100  1+ 0= 1 -
# -11400  1+ 0= 1 -
# -11700  1+ 0= 1 -
# -12000  0+ 0= 0
# -12300  2+ 0= 2 --
# -12600  0+ 1= 1 *
```

プロジェクト2

```
会議1
# -  600  0+ 0= 0
# - 1200  0+ 2= 2 **
# - 1800  0+ 4= 4 ****
# - 2400  2+ 1= 3 --*
# - 3000  1+ 1= 2 -*
# - 3600  0+ 3= 3 ***
# - 4200  2+ 3= 5 --***
# - 4800  2+ 1= 3 --*
# - 5400  0+ 1= 1 *
# - 6000  0+ 0= 0
# - 6600  1+ 0= 1 -
# - 7200  1+ 0= 1 -
# - 7800  1+ 2= 3 -**
# - 8400  1+ 0= 1 -
# - 9000  0+ 1= 1 *
# - 9600  0+ 2= 2 **

会議2
# -10200  1+ 1= 2 -*
# -10800  1+ 3= 4 -***
# -11400  3+ 1= 4 ---*
# -12000  0+ 2= 2 **
# -12600  1+ 2= 3 -**
# -13200  0+ 2= 2 **
# -13800  0+ 1= 1 *
# -14400  4+ 1= 5 ----*
# -15000  0+11=11 **********
# -15600  0+16=16 ****************
# -16200  1+ 3= 4 -***
# -16800  2+ 0= 2 --
# -17400  3+ 2= 5 ---**
# -18000  0+ 3= 3 ***
# -18600  2+ 0= 2 --
# -19200  0+ 1= 1 *

会議3
# -19800  1+ 1= 2 -*
# -20400  2+ 2= 4 --**
# -21000  4+ 6=10 ----******
# -21600  1+ 4= 5 -****
# -22200  2+ 4= 6 --****
# -22800  2+ 3= 5 --***
# -23400  1+ 2= 3 -**
# -24000  0+ 3= 3 ***
# -24600  1+ 4= 5 -****
# -25200  0+ 1= 1 *
# -25800  1+ 3= 4 -***
# -26400  0+ 1= 1 *
# -27000  0+ 0= 0
# -27600  0+ 1= 1 *
# -28200  0+ 3= 3 ***
# -28800  0+ 1= 1 *
# -29400  0+ 0= 0
# -30000  0+ 0= 0
# -30600  0+ 0= 0

会議4
# -31200  0+ 1= 1 *
# -31800  0+ 5= 5 *****
# -32400  3+ 3= 6 ---***
# -33000  3+ 3= 6 ---***
# -33600  1+ 3= 4 -***
# -34200  0+ 0= 0
# -34800  0+ 1= 1 *
# -35400  0+ 0= 0
# -36000  1+ 6= 7 -******
# -36600  1+ 2= 3 -**
# -37200  0+ 0= 0
# -37800  0+ 0= 0
# -38400  0+ 0= 0
# -39000  0+ 0= 0
# -39600  0+ 1= 1 *
# -40200  0+ 0= 0
# -40800  0+ 0= 0
# -41400  0+ 2= 2 **
```

プロジェクト3

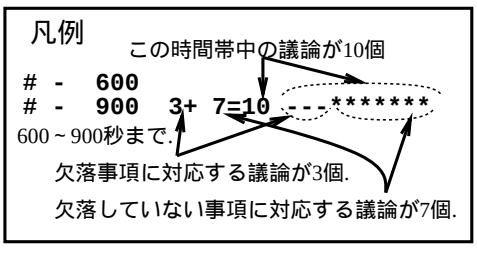
```
会議1
# -  300  0+ 0= 0
# -  600  2+ 1= 3 --*
# -  900  5+ 5=10 -----*****
# - 1200  0+ 3= 3 ***
# - 1500  1+ 0= 1 -
# - 1800  0+ 1= 1 *
# - 2100  0+ 1= 1 *
# - 2400  1+ 2= 3 -**
# - 2700  0+ 0= 0
# - 3000  0+ 1= 1 *
# - 3300  0+ 0= 0
# - 3600  0+ 0= 0
# - 3900  0+ 2= 2 **
# - 4200  1+ 5= 6 -*****
# - 4500  1+ 5= 6 -*****
# - 4800  0+ 2= 2 **
# - 5100  3+ 5= 8 ---*****
# - 5400  0+ 2= 2 **
# - 5700  3+ 2= 5 ---**
# - 6000  1+ 0= 1 -
# - 6300  2+ 2= 4 --**
# - 6600  0+10=10 **********
# - 6900  1+ 5= 6 -*****
# - 7200  4+ 2= 6 ----**
# - 7500  5+ 0= 5 -----
# - 7800  1+ 0= 1 -
# - 8100  2+ 0= 2 --

会議2
# - 8400  1+ 3= 4 -***
# - 8700  5+ 2= 7 -----**
# - 9000  1+ 3= 4 -***
# - 9300  0+ 4= 4 ****
# - 9600  0+ 0= 0
# - 9900  0+ 0= 0
# -10200  1+ 2= 3 -**
# -10500  0+ 3= 3 ***
# -10800  0+ 5= 5 *****
# -11100  4+ 0= 4 ----

会議3
# -11400  0+ 3= 3 ***
# -11700  0+ 2= 2 **
# -12000  0+ 0= 0
# -12300  0+ 0= 0

会議4
# -12600  1+ 1= 2 -*

会議5
# -12900  0+ 3= 3 ***
```



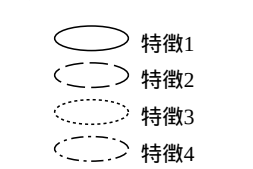


図 2.24: 議論の分布

- 会議内の時間的に広範囲に分布した複数の議論において議論された事項ほど, 文書から欠落することが少ない.
- いくつかの個々の欠落事項に関する特徴と時間的に隣接して議論されている.

これらの結果より, 以下のような支援が有効であると考えられる.

- 決定の不明確な部分は不明確であるままに記録し, 後に明確化する支援.
- 文書内で孤立する事項の記録/管理を行なう. もしくは, その発生を他の部分と常に関係つけることで, 孤立する部分の発生を防ぐ支援を行なう.
- 会議中の議論で出現した事項を後に再び議論できるように記録し, それが会議の中で決定事項か否かを管理する支援. 特に, 1 回のみ議論された事項や, 内容的に今回得られた特徴を持つ事項などを記録し, 作業者に再提示することが有効であると考えられる.
- 会議内で時間的に隣接して議論される事項同士は内容的に関係がある場合があり, その関係を用いて事項の欠落を防止する支援. この点に関しては 2.5節で改めて分析を行なう.

2.5 議論の構造とプロダクトの構造の関係

2.5.1 分析の目的

発話履歴には，会議中の人間のごく自然な思考の過程が含まれると考えられる．特に，会議の参加者が意図的にあるいは無意識に行う話題とする事項の変遷は，その前後の事項間に内容的な関連がある．よって, このような人間のごく自然な思考過程や事項間の関連性は, プロダクトである文書などの文書の理解・記述や, プロセスである会議の支援に役立てることが可能だと思われる．本分析では, 発話履歴のうち，特に事項の変遷に着目し，仕様作成会議の発話履歴と文書の構造との関係の分析を行なう．

図 2.25の例のように発話履歴中の事項の変遷にもいくつかのパターンが考えられる．そして, そのパターンによって事項の関係の強さが異なるものと考えられる．そこで, 発話履歴中の事項の変遷をもとに 3 種類の事項の隣接を以下のように定義し, 発話履歴から得られる事項間の関係の強さを示す指標とする．

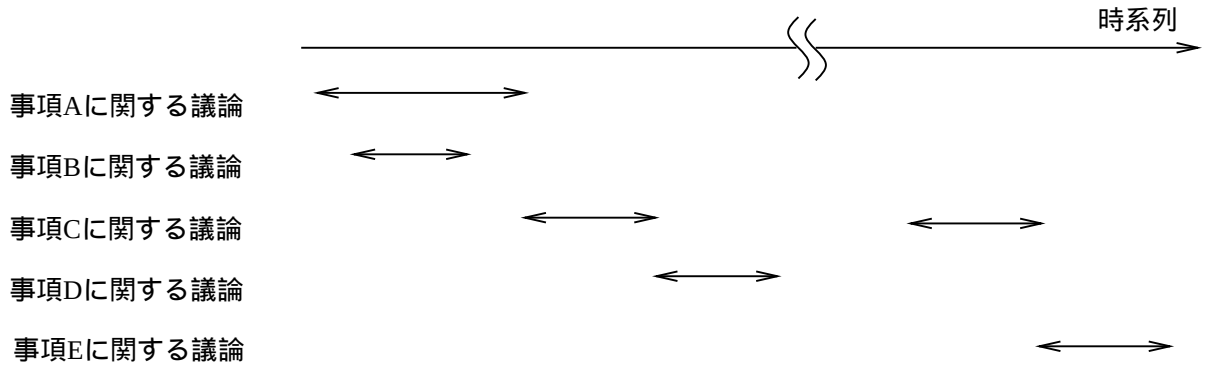


図 2.25: 発話履歴の例

包含型隣接 : 事項 A に関する議論と事項 B に関する議論のように，事項 A について話されている最中に別の事項 B が話されているような場合の，事項 A に関する議論と事項 B に関する議論との隣接を包含型隣接と呼ぶ．

直接型隣接 : 事項 A に関する議論と事項 C に関する議論，事項 C に関する議論と事項 D に関する議論のような隣接を直接型隣接と呼ぶ．

なお，包含型隣接も，直接型隣接に含むことにする．ただし，事項 A に関する議論と事項 B に関する議論の隣接の例で言えば，事項 B に関する議論が始まる時点で事項 A から事項 B へ一度事項が転換し，さらに事項 B に関する議論が終る時点で事項 B から事項 A へ再び事項が転換しているため，包含型隣接 1 回は直接型隣接 2 回と考える．

間接型隣接 : 事項 A に関する議論と事項 C に関する議論，事項 C に関する議論と事項 E に関する議論がそれぞれ直接型隣接しているとする．ただし，事項 A に関する議論と事項 E に関する議論は隣接しているとは限らないものとする．この 2 つの隣接関係に推移律を適用して得られる，事項 A に関する議論と事項 E に関する議論のような隣接を間接型隣接と呼ぶ．

推移律を用いるため, 2 つの事項を関係付けるパスは一般には複数存在する．最短のパスの長さを可到達パス長と呼ぶ場合がある．

なお, これらを単に事項の隣接と呼ぶ場合があり, 隣接関係にある事項の対を隣接対と呼ぶ場合がある．

文書の構造を示す指標として “事項の影響対” を導入する．以下のいずれかの条件が成立する事項 A と事項 B を, 事項の影響対と定義する．

- 事項 A の内容を変更もしくは事項 A を削除した場合には，それに合わせて事項 B の内容を変更するか，事項 B を削除しなければならない．

- 事項 B の内容を変更もしくは事項 B を削除した場合には，それに合わせて事項 A の内容を変更するか，事項 A を削除しなければならない．

このような事項の対は仕様書や議事録などの文書中に明示的に併記される必要はないと思われるが, 文書に記述してある仕様を理解するための重要な情報であると思われる. よって, 仕様書や議事録などの文書の中にも何らかの方法で, この情報が記述されていることが望ましいと考えられる.

事項の隣接対と事項の影響対との関係に関して以下のような予想ができる.

予測 **1** ：包含型隣接のある隣接対は，影響しあう可能性が非常に高い．

予測 **2** ：直接型隣接のある隣接対は，影響しあう可能性が高い．特に，隣接する回数が多いほど影響する可能性が高い．

予測 **3** ：間接型隣接のある隣接対は，一般的には影響しあう可能性が低い．特に，可到達パス長が多いほど影響する可能性が低い．

これらの予測が正しければ, 会議における事項の変遷を利用して, 仕様書などの文書の理解の支援を行なうことができる.

2.5.2 分析の手順

1. 仕様を複数の事項に区切り，各事項について発話されている議論を作成する．
   (a) 会議で作られた文書の章構成などを基に，事項を選定し，各事項のキー語を定める．
   (b) 会議で取り上げられたが，何らかの理由で文書には記述がないような事項も，キー語を定めて 1 つの事項として取り入れる．これを文書外事項と呼ぶ.
   (c) キー語を基に，各事項について言及している議論を決定する．
2. 議論の隣接から，包含型隣接および直接型隣接のある隣接対のリストを作成する．ただし，会議日程調整などの文書そのものと直接関係のない事項は除外事項とし，隣接の対象としない．
3. 直接型隣接のある隣接対のリストから間接型隣接のある隣接対のリストを作成する．
4. ここまでの分析作業に関与していない第三者に，文書と各事項のキー語の一覧を見せて，影響し合う事項の対を決めてもらう.
5. 分析の結果得られた隣接対と第三者が決めた影響のある事項の対とが一致する割合を調べる．

2.5.3 分析結果と考察

分析対象となった事項と議論の数を表 2.11に示す. 総事項数は仕様外事項数を含めた全ての事項の数を示す. 文書外事項および除外事項は手順で述べた通りである. 考え得る隣接対の数は分析対象とする相異なる事項の対の数を示している. もちろん, 実際に仕様書などの文書や会議には発生していない対も含まれている. 事項の影響対の数は上記の影響対にあてはまると第三者が判断した事項の対の数である.

前述の予測を確かめるために, 以下の 2 つの指標を導入する.

正答率 ：実際に会議に発生した事項の隣接対の内で, 影響対であると判断された対の割合.

この割合が低ければ, 文書内における影響の強さの観点から, 虚偽とみなされる情報を, 会議の事項の変遷から得られた事項対に関する情報が含むことを示す.

獲得率 ：実際に影響対であると判断された事項の対の内で, 会議の隣接対から獲得できた対の割合.

この割合が高ければ, 文書内における影響の強さに関する情報を, 会議の事項の変遷から, 多く得られたことを示す.

表 2.11: 事項と事項の対と議論の数

	プロジェクト 1	プロジェクト 2	プロジェクト 3
総事項数 (仕様外事項を含む)	22	23	22
文書外事項	0	1	3
除外事項数	1	1	3
分析対象とする事項数	21	22	19
考え得る隣接対の数 ((事項数^2 − 事項数)/2)	210	231	171
事項の影響対の数	68	153	71
議論の数	62	186	154

表 2.12: 包含関係隣接のある隣接対と影響対の関係

	隣接対数	隣接対の内の影響対数	正答率	獲得率
プロジェクト 1	5	5	100.0	7.4
プロジェクト 2	25	22	88.0	14.4
プロジェクト 3	35	28	80.0	39.4

予測 1 の包含型隣接対に関する結果を表 2.12に示す. 包含型隣接対は，獲得率が低いため，ある事項を修正・削除した際に，チェックすべき影響対を得ることには適さない．また，プロジェクトによって若干のバラツキがあるが，一般に他の型に比べて正答率は高い．

予測 2 の直接型隣接対に関する結果を表 2.13に示す. 直接型隣接対は，隣接回数が少なくなるにつれて正答率は下がるが，隣接回数 1 回以上の隣接対による獲得率はプロジェクト 1 を除けば，ほぼ 50%と比較的高い．

予測 3 の間接型隣接対に関する結果を表 2.14に示す. 間接型隣接対は，獲得率は高いが，正答率が低く，ほとんど実用的に利用することはできない．

なお，3 つのプロジェクトとも，会議の進行に関して特に制限を加えておらず，議長などもおいていないため，議論の発散傾向が見られた．特に，プロジェクト 1 の直接型隣接対による，影響対の獲得率が 30.9%と，プロジェクト 2 の 51.0%，プロジェクト 3 の 56.3%と比べてかなり低いのは，プロジェクト 1 の会議が議事の進行を全く制御しないブレインストーミング的な会議だったため，議論があちこちに発散していて，本質的な関連のない事項の変遷が多く含まれているためである．議論の進行を適切に制御することにより，正答率や獲得率は高くなると考えられる．

2.5.4 まとめ

本分析から発話履歴における事項の変遷は, プロダクトである仕様書などの文書内の影響の波及の観点から, ある程度有効であることがわかった. 具体的には以下に示す通りである.

- 包含型隣接の情報から, プロダクト内の影響の波及に関するかなり正確な情報を獲得することができる.
- 直接型隣接の情報から, プロダクト内の影響の波及に関するかなり多くの情報を獲得することができる.

プロダクトの影響の波及の情報を発話履歴から獲得することで, プロダクトの理解や記述を支援することが可能だと思われる. 例えば,

- 発話履歴からプロダクトの改善を行うような場合には，包含型隣接対が利用できる.
  包含型隣接対が少ない場合には，隣接回数の多い直接型隣接対を合わせて利用する.
- 発話履歴から影響し合う事項の対をできるだけ多く獲得したい場合は, 直接型隣接対を用いる方が良い．

などが考えられる.

表 2.13: 直接型隣接対と影響対の関係

		隣接対数	隣接対の中の影響対数	正答率	獲得率
プロジェクト 1	隣接回数 4 回以上	1	1	100.0	1.5
	隣接回数 3 回以上	3	2	66.7	2.9
	隣接回数 2 回以上	9	8	88.9	11.8
	隣接回数 1 回以上	30	21	70.0	30.9
プロジェクト 2	隣接回数 6 回以上	3	2	66.7	1.3
	隣接回数 5 回以上	9	8	88.9	5.2
	隣接回数 4 回以上	14	13	92.6	8.5
	隣接回数 3 回以上	27	24	88.9	15.7
	隣接回数 2 回以上	45	40	88.9	26.1
	隣接回数 1 回以上	87	78	89.7	51.0
プロジェクト 3	隣接回数 15 回以上	1	1	100.0	1.4
	隣接回数 13 回以上	2	2	100.0	2.8
	隣接回数 12 回以上	3	3	100.0	4.2
	隣接回数 10 回以上	4	4	100.0	5.6
	隣接回数 9 回以上	6	6	100.0	8.5
	隣接回数 8 回以上	9	8	88.9	11.3
	隣接回数 7 回以上	12	10	83.3	14.1
	隣接回数 6 回以上	14	12	85.7	16.9
	隣接回数 5 回以上	15	13	86.7	18.3
	隣接回数 4 回以上	21	18	85.7	25.3
	隣接回数 3 回以上	26	22	84.6	31.0
	隣接回数 2 回以上	38	29	76.3	40.8
	隣接回数 1 回以上	55	40	72.7	56.3

表 2.14: 間接型隣接対と影響対の関係

		隣接対数	隣接対の中の影響対数	正答率	獲得率
プロジェクト 1	可到達パス長 1 以下	30	21	70.0	30.9
	可到達パス長 2 以下	73	39	55.7	57.4
	可到達パス長 3 以下	112	51	45.5	75.0
	可到達パス長 4 以下	145	60	41.4	60.9
	可到達パス長 5 以下	165	63	38.2	92.6
	可到達パス長 6 以下	171	64	37.4	94.1
	可到達パス長 7 以下	172	64	37.2	94.1
プロジェクト 2	可到達パス長 1 以下	87	78	89.7	51.0
	可到達パス長 2 以下	215	143	66.5	93.5
	可到達パス長 3 以下	231	153	66.2	100.0
プロジェクト 3	可到達パス長 1 以下	55	40	72.7	56.3
	可到達パス長 2 以下	144	67	46.5	94.4
	可到達パス長 3 以下	171	71	41.5	100.0

2.6 参加者の役割による会議の分割

2.6.1 分析の目的

1 つの会議の中でも，議論の目的が異なるため，注目する生産物が異なる部分がある．例えば，ソフトウェアの仕様を作成する会議の中でも，製品の発注, プロジェクトのスケジュール, 仕様書の説明といった部分があり，それぞれに，発注書，スケジュール表，仕様書など，注目する生産物の種類が異なる．このような会議の部分をステップと呼ぶことにする．もし, 会議中のそれぞれのステップに特徴があれば, それを用いて会議をステップに分割し, ステップ毎の文書の作成の支援が可能になる.

会議の進行に特に重要なものは作業者の発話であり, 発話の発話者, 開始時刻と終了時刻は機械的に判定可能である．そして, それぞれの作業者は顧客, 設計者, 調整者などの役割を持って会議に参加している．また, ステップが変わると作業者の発話の量が変化すると考えられる．よって, ステップ毎に作業者毎の発話の時間の長さと発話回数を調査する．もし, あるステップにおいて, 特定の作業者の発話の時間の長さや回数に著しい偏りが見られた場合, その偏りがステップの特徴となる.

また, 特定の作業者が交互に発話を行なっている場合は, それらの作業者間で対話を行なっていると考えられる．この対話の偏りもステップによって異なると考えられる．よって, ステップ毎の作業者間の対話の量を調査する．もし, あるステップにおいて, 特定の作業者間の対話の量に著しい偏りが見られた場合, その偏りがステップの特徴となる.

2.6.2 分析の手順

発話, 対話の偏りを調べるために, ビデオに記録した発話を以下の順序で整理する.

1. 会議の内容を見て, ステップで区切る.
2. 計算機処理のため, ビデオ記録の発話をリスト化する.
3. 視覚的に偏りを調べるため, リスト化した発話データを時間軸上に表示する.
4. 対話の偏りを調べるため, 発話者の順序関係を表にする.

以下にこの分析で作成するデータを紹介する.

1. 発話リスト

   全ての発話について発話者名と発話の開始時刻と終了時刻を 1 秒単位で記録したものである．この分析は発話の形式的な特徴を重視するため, 時間は詳細に調べるが, 個々の発話内容については詳しくは調べない．図 2.26に実際の発話リストの例を示す．左から発話開始時刻, 発話終了時刻, 発話者を記述する．例えば最上段では, 0 分 12 秒から 2 分 30 秒まで作業者 a が発話した, ということを表す.

```
00.12  02.30  a
02.30  02.35  b
02.35  03.36  a
03.36  03.47  c
03.47  04.15  b
04.19  04.43  c
04.43  04.50  a
04.50  05.04  b
05.07  05.15  d
```

図 2.26: 発話リストの例

発話には, 質問に対する返事などの会議において求められるもの, 口癖のような相槌といった発話時間が短いものがある. 本分析では原則として, 長さが 1 秒未満の発話は分析対象としなかった. ただし, 返事は会議に貢献するものなので, たとえ長さが 1 秒未満の発話であったとしても, 1 秒間の発話として処理する. 一方, 相槌は発話として記録しない.

会議に関係しない発話がなされる時間もある. 会議に関係しない発話がなされる時間を “除外” の発話者名で記録する. “除外” は, 会議に関係しない話題が議論された時や冗談などで生じ, 会議を中断させる原因となっている.

また, 発話は話し始めた時間, 話し終った時間をそれぞれ開始時刻, 終了時刻とするべきである. しかし話し終りで声が小さくなったり, 他の人の発話に遮られて終ったりすると終了時刻は正確には検出できない. また, 発話の多くは前の発話を受けて行なわれると考えられるので, 声が聞こえなくなった時間から次の発話までの間が 2 秒未満であれば, 後の方の発話の開始時刻を前の発話の終了時刻とする. ただし 2 秒以上の間がある時には, 声が聞こえなくなった時間を終了時刻とする.

2. 発話グラフ

発話リストを視覚的に表したものが発話グラフである. 発話グラフを用いると発話の偏りを視覚的に調べることができる.

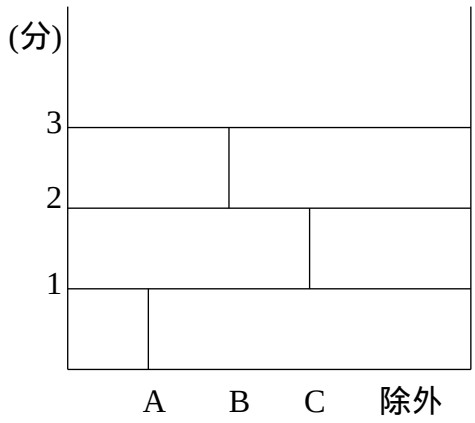


図 2.27: 発話グラフ

図 2.27に示すように, 縦軸に時間を上向きにとり, 横軸に各作業者名と “除外” を置いたグラフを考える. そこに発話リストからデータを取り出し, 横軸を発話者名に合わせ, 縦方向に発話開始時刻から発話終了時刻まで線分を引き, 発話の表示とする. 例えば図は 0 分から 1 分まで A が発話し, 1 分から 2 分まで C が発話し, 2 分から 3 分まで B が発話したことを表す. 以下全ての発話データについてそれを行なうことにより, 会議中の各人の発話の多さや長さとその時間遷移を表せる.

3. 発話表

作業者間の関係の強さの偏りについて調べるために, 発話者の順序関係を表にする. 発話の順序については, 以下の式を見たす時, A を B の直前の発話とする. ただし, 発話 A, B, C の開始時刻, 終了時刻を, それぞれ Ast, Aend, Bst, Bend, Cst, Cend とする.

$$(\mathrm{Ast} < \mathrm{Bst}) \wedge \neg\exists \mathrm{C}((\mathrm{Aend} < \mathrm{Cst} < \mathrm{Bst}) \vee (\mathrm{Aend} < \mathrm{Cend} < \mathrm{Bst}))$$

直観的には,「 A, B の間に他の発話が入っていない.」ということである. ただし, 1 つの発話に直前の発話が複数ある場合は, 発話内容を調べてどちらか一方を選ぶ. また, 最初になされた発話には前の発話がないので, 直前の発話を “除外” とする.

2 人の作業者 A, B が対話を行なっている場合, それぞれの作業者の発話は以下のような順序になっている場合が多い.

表 2.15: 発話表

回数	作業者 A	作業者 B	作業者 C
作業者 A	1	7	0
作業者 B	6	0	3
作業者 C	1	1	0
除外	0	1	0
合計	8	9	3

総発話回数：20 回

A, B, A, B, A, B ···

よって, ある時間の範囲において, A の発話が B の発話の直前である割合と, B の発話が A の発話の直前である割合が, 他の発話の遷移の割合よりも大きければ, この 2 人の作業者 A, B は対話を行なっていると解釈する. 実際に対話を行なっているかどうかは, 発話の順序における 1 次の遷移ではなく, ある程度の長さのある遷移列を調べ, さらに, 発話内容を調べる必要がある. しかし, 本分析ではデータの整理の簡略化のために, 1 次の発話の遷移のみを対象とする.

発話表は, 発話の順序における 1 次の遷移を表にしたものであり, ある 2 人の作業者間の対話の割合を示す指標となる. 発話表は発話の回数を表すものと発話の時間を表すものの二種類を作成する. 発話回数による発話表は, ある作業者の発話の次に行なわれた発話の回数を作業者毎に調べて表にしたものである. 例えば, 表 2.15では, B の後に A が 6 回, B が 0 回, C が 3 回の発話を行なったことを時間による発話表は, ある作業者の発話の次に行なわれた発話の時間を作業者毎に調べて表にしたものである. 発話表には, 作業者名を縦横に並べ, 縦の欄には “除外” と “合計” も加える. 横軸は次の発話者を表す. 数値は, “合計” を除く縦の欄の発話者の直後に横の欄の発話者が発話した回数または時間 (秒) の合計値を表す.

回数による発話表では, 返事などの時間的に短い発話も 1 として数えられるため, 作業者間の対話の偏りを調べるために有効であると考えられる. それに対し, 発話時間による発話表では, 発話を, その発話時間で重み付けを行なっているため, 対話という観点よりも, むしろ個々の作業者の発話量の偏りを調べるのに有効であると考えられる.

表 2.15の例は発話回数に関する発話表である. 例えば, 表中の 3 行 2 列の 6 は, B の発話の直後に A が発話した回数が 6 回であることを表している.

4. 対話係数

発話グラフや発話表を用いると, 発話や対話の偏りについて調べることができる. 作業者間の発話表より以下の式によって対話係数を算出する. A の直後に B が発話した量を talkAB, B の直後に A が発話した量を talkBA, 総発話量を talkALL, A と B の対話係数を talk とすると,

$$\frac{\mathrm{talkAB} + \mathrm{talkBA}}{\mathrm{talkALL}} = \mathrm{talk}$$

対話係数 talk が大きければ A と B の対話が多いと考えることができる. 例えば表 2.15の例では, A と B の発話回数での対話係数が 0.65 になり, A と B の対話が多いと言える.

2.6.3 分析結果と考察

2.6.2節で紹介したデータを実際の会議の記録から算出することで, これらのデータがステップに関する特徴を示していることを示す. 分析対象としては 2.2 節で紹介したプロジェクト 3 における全体会議を利用する. その理由は, 他のプロジェクトでは参加していた作業者の種類も少なく, 会議が単一のステップで構成されていたからである. よって, 分析の目的がステップ毎の発話, 対話の偏りを調べる目的にはプロジェクト 3 の全体会議が適していると考えられる. プロジェクト 3 の全体会議は内容的に以下のステップに分類できた.

発注ステップ : 顧客が製作を依頼するツールの説明を行なう会議の部分.
スケジュールステップ : 今後の開発スケジュールの作成を行なう会議の部分.
機能提案ステップ : ツールに付加する機能などの提案を行なう会議の部分.
仕様説明ステップ : 設計者による仕様書の説明を行なう会議の部分.

このステップ名を用いて全体会議のおおまかな経過を書くと以下のようになる.

6 月 18 日 第 1 回全体会議: 前半は発注ステップ, 後半はスケジュールステップ.
7 月 29 日 第 2 回全体会議: 大部分が機能提案ステップで, 他にスケジュールステップなど.
9 月 9 日 第 3 回全体会議: 大部分が仕様説明ステップで, 他にスケジュールステップなど.
9 月 27 日 第 4 回全体会議: 仕様書についての仕様説明ステップ.

第 1 回全体会議の発注ステップ (28 分 44 秒) およびスケジュールステップ (32 分 48 秒), 第 2 回全体会議の機能提案ステップ (39 分 20 秒), 第 3 回全体会議の仕様説明ステップ (48 分 38 秒) に関して前節の分析法を適用する.

ただし, 製作者の 4 名は全体的に発話が少なかったので, 1 つにまとめる. 設計者の 2 名は発話が多く, 設計者同士の対話もあるので別々に表示する. しかし, 役割別の分析では, 設計者 2 名の発話量を合計して使用する.

以下に各ステップごとの分析結果を述べる.

発注ステップ :

図 2.28に, このステップの発話グラフを示し, 表 2.16に, このステップの回数と時間による発話表を示す. 発注ステップは, 調整者による 8 分にわたる長い発話とその前後で三つに分けることができる. この発話自体の内容はあまり重要ではなく, あまりに長く, 特殊なデータである. 主観的ではあるが, この発話はステップの分析に障害となると考え, これを “除外” として整理した.

表 2.16から明らかなように, 顧客と調整者の対話が多い. 発話回数での対話係数は 0.63, 発話時間での対話係数は 0.61 である. これは, 発注ステップでは顧客の要求を調整者が対話によって引き出すことが多いからだと考えられる.

スケジュールステップ :

図 2.29に, このステップの発話グラフを示し, 表 2.17に, このステップの回数と時間による発話表を示す. スケジュールステップはグラフ, 表から調整者の発話量が多いことが分かる. 発話回数は全体の 37%で他の発話者の 2 倍程度だが, 平均発話時間が 23.5 秒/回と全体平均の約 2 倍であるため, 発話時間においては全体の 75%を占める. これはスケジュールを作成する時に, 調整者が全員の都合を聞いてそれをまとめる仕事を行なうからである. つまり, 調整者と他の作業者との関係が深く, 調整者以外の作業者同士の関係はあまりない.

機能提案ステップ :

図 2.30に, このステップの発話グラフを示し, 表 2.18に, このステップの回数と時間による発話表を示す. 機能提案ステップでは, あまり偏りが生じていない. 発話回数では設計者, 調整者, 顧客がそれぞれ全体の 33%, 28%, 34%を占め, ほぼ同程度といえる. 一方, 発話時間では設計者, 調整者, 顧客がそれぞれ 23%, 50%, 23%を占め, 調整者の発話が多い. しかし, それでも他のステップと比較すると偏りは小さいといえる. 対話量でも, 設計者と調整者, 調整者と顧客, 顧客と設計者の組合せを見ると, 発話回数での対話係数がそれぞれ 0.22, 0.28, 0.26 となり, 特定の偏りは見られない. これは機能に関する議論を行なう上で, 多くの人の自由な提案が求められたためだと思われる.

仕様説明ステップ :

図 2.31に, このステップの発話グラフを示し, 表 2.19に, このステップの回数と時間による発話表を示す. 仕様説明ステップでは設計者の発話が多い. 特に発話時間では設計者の発話は全体の 65%を占めている. 対話は, 発話時間での対話係数が設計者と顧客では 0.38, 設計者と調整者では 0.30 となる. ここから設計者と調整者, 顧客の対話が多く, 特に設計者と顧客のつながりが強いといえる. 設計者の発話が多いのは, 設計者が自分たちの作った仕様書の説明をするためである. また設計者と顧客との対話が多いのは, そのツールの仕様について顧客が十分に理解するべく質問を多く発するためだと思われる.

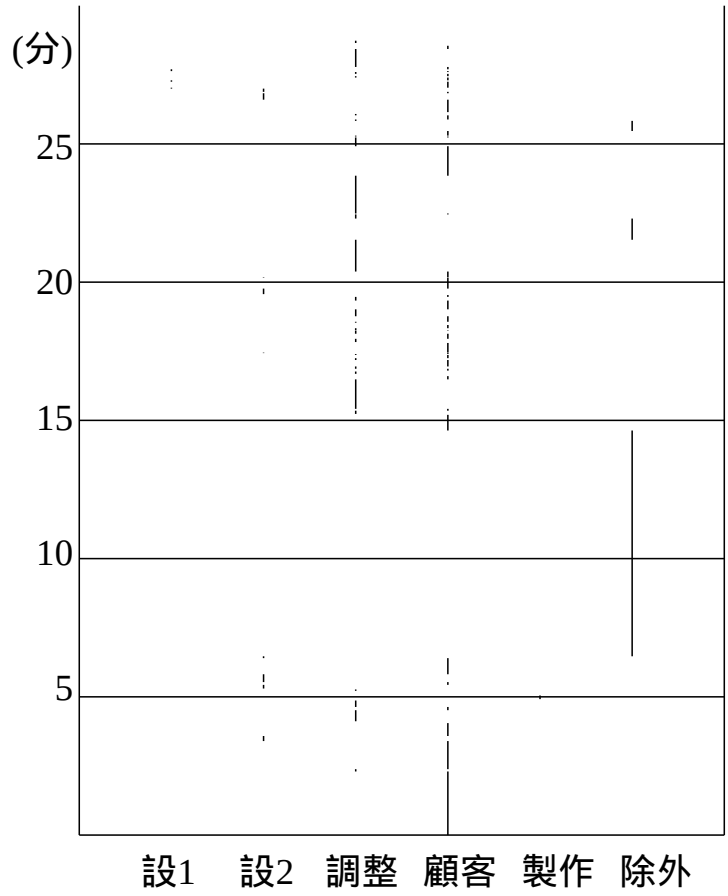


図 2.28: 発注ステップの発話グラフ

表 2.16: 発注ステップの発話表

回数	設 1	設 2	調整	顧客	製作
設 1	0	0	0	3	0
設 2	1	0	0	7	0
調整	0	1	0	23	1
顧客	2	8	24	0	0
製作	0	0	1	0	0
除外	0	0	2	2	0
合計	3	9	27	35	1

時間	設 1	設 2	調整	顧客	製作
設 1	0	0	0	24	0
設 2	2	0	0	129	0
調整	0	8	0	288	8
顧客	7	67	399	0	0
製作	0	0	3	0	0
除外	0	0	12	172	0
合計	9	75	414	613	8

総発話回数：75 回
総発話時間：1119 秒
平均発話時間：14.9 秒/回

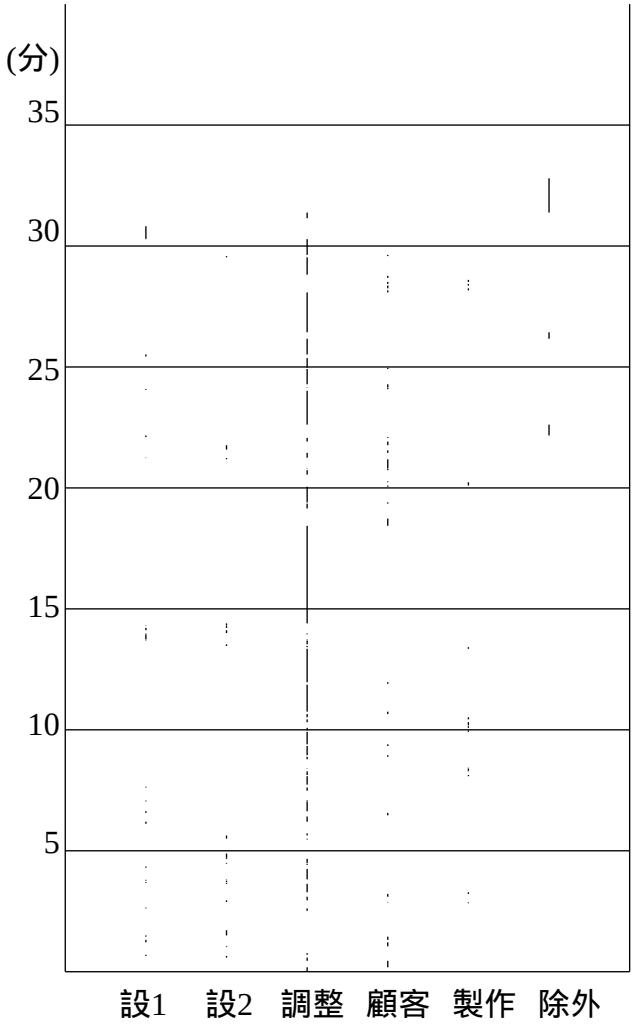


図 2.29: スケジュールステップの発話グラフ

表 2.17: スケジュールステップの発話回数表

回数	設 1	設 2	調整	顧客	製作
設 1	0	6	12	2	1
設 2	6	0	8	5	0
調整	12	8	2	17	8
顧客	4	3	15	0	5
製作	0	0	9	5	3
除外	0	0	3	0	0
合計	22	17	49	27	17

時間	設 1	設 2	調整	顧客	製作
設 1	0	35	144	12	2
設 2	15	0	309	22	0
調整	38	28	16	120	23
顧客	17	18	429	0	26
製作	0	0	60	20	14
除外	0	0	194	0	0
合計	70	81	1152	174	65

総発話回数：132 回
総発話時間：1542 秒
平均発話時間：11.7 秒/回

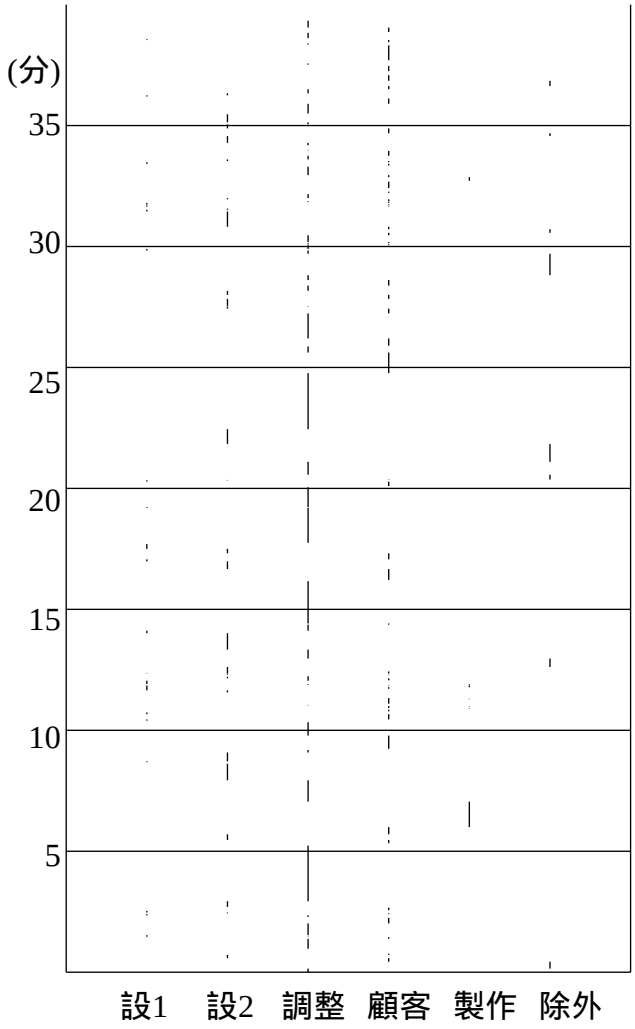


図 2.30: 機能提案ステップの発話グラフ

表 2.18: 機能提案ステップの発話回数表

回数	設 1	設 2	調整	顧客	製作
設 1	0	5	9	9	0
設 2	8	0	10	6	0
調整	6	6	0	24	0
顧客	8	14	16	2	7
製作	0	0	1	4	0
除外	0	1	4	4	0
合計	22	26	40	49	7

時間	設 1	設 2	調整	顧客	製作
設 1	0	47	216	52	0
設 2	46	0	355	34	0
調整	22	139	0	318	0
顧客	25	147	324	17	81
製作	0	0	53	16	0
除外	0	37	71	41	0
合計	93	370	1019	478	81

総発話回数：144 回
総発話時間：2041 秒
平均発話時間：14.2 秒/回

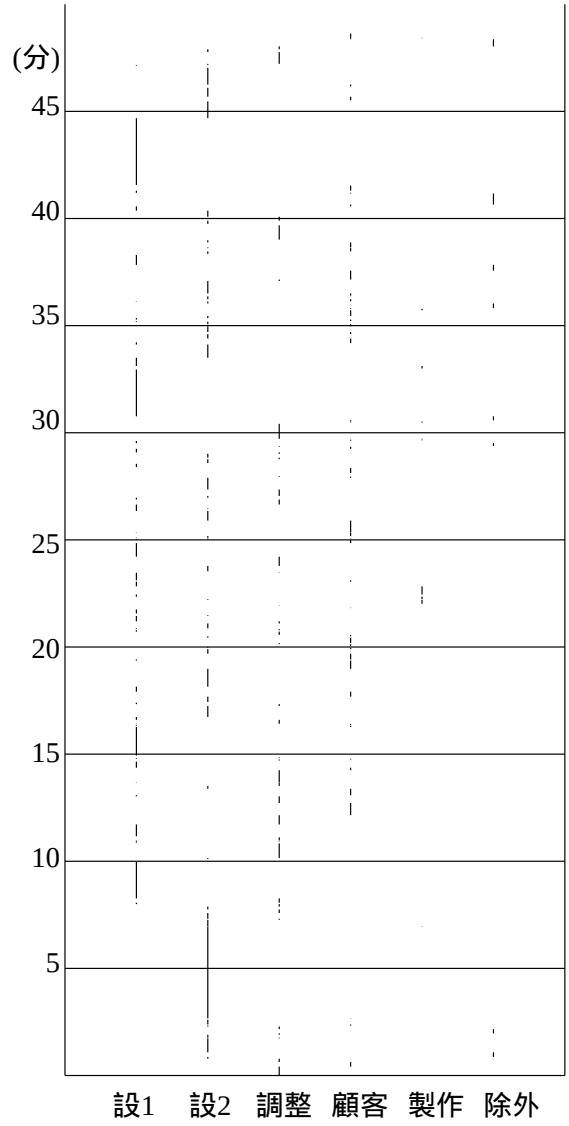


図 2.31: 仕様説明ステップの発話グラフ

表 2.19: 仕様説明ステップの発話回数表

回数	設 1	設 2	調整	顧客	製作
設 1	0	16	12	15	2
設 2	8	0	18	19	2
調整	16	12	0	7	2
顧客	14	17	7	0	5
製作	3	2	1	5	0
除外	3	2	2	2	0
合計	44	49	40	48	11

時間	設 1	設 2	調整	顧客	製作
設 1	0	239	103	137	30
設 2	54	0	248	139	8
調整	312	134	0	110	13
顧客	270	481	45	0	9
製作	42	21	42	20	0
除外	165	44	32	5	0
合計	843	916	470	411	60

総発話回数：192 回
総発話時間：2700 秒
平均発話時間：14.1 秒/回

2.6.4 まとめ

会議中のステップには会話の偏りがある程度あることがわかった. 例えば，ソフトウェアの仕様を作成する会議では，それぞれのステップで以下のような特徴が見られた．

発注ステップ : 顧客と調整者の対話が中心.

スケジュールステップ : 調整者と他の作業者との対話が中心.

機能提案ステップ : 特定の偏りがない.

仕様説明ステップ : 設計者が発話. 対話は設計者-顧客, 設計者-調整者が多い.

よって，ソフトウェアの仕様を作成する会議中に上記のような発話の特徴が見られた場合は, その特徴を持つステップに必要な文書などのプロダクトに注目させるような支援が有効であると考えられる．本分析では，ソフトウェアの仕様作成に特化した性質を用いていないため，ソフトウェアの仕様を作成する以外の会議，例えば企画を作成する会議などもでも，同様の性質があると思われる．

第 3 章

ハイパー議事録システムの設計

本章では第 2 章における実際の会議の分析結果を基に, 従来の会議の問題点を表現するためのデータ構造の定義を行なう. この構造を基に支援のための方法やシステムの実装が可能となる.

3.1 システムの対象となる作業とその問題点

ハイパー議事録システムが対象とする作業は，第 2 章の分析対象となった作業と同一であることは言うまでもないが, ここで簡単にまとめておく.

- 複数の作業者による対面式の会議から構成される .
- 会議は通常複数回行なわれる .
- 作業を通して共通の生産物が作成される .
- 生産物は通常，文書であり，自然言語で記述される.
- 文書は, 用紙の上の文章や図として記述するため, その構造は文章などを節とする木構造となる. そして, 文書は記述されている順番に読まれる.
- 文書は通常, 会議と会議の間で要約・記述される.
- 会議での作業者の提案は, 会議開始以前に, それぞれの参加者が準備してきたものを含む.

第 2 章における分析結果より, 実際の会議ではいくつかの問題点が存在することがわかった.

- 議論内容が生産された議事録や仕様書などの文書から欠落する.
- 決定が不明瞭な点が発生している.
- 議論中で文書全体から孤立するような部分が文書から欠落しやすい.
- 非効率的な議論がある.
- 文書内の一部を変更した場合に, その影響を受けて同様に変更を行なわなければならない部分がある. それらの部分は会議中において時間的に連続して議論されている場合が多い.

また, それ以外に以下のような特徴が得られた. 以下の特徴は会議における問題点ではないが, 会議の支援を考える場合に有効な情報となる.

- 会議中には, 発注ステップ, 説明ステップ, スケジュールステップなどの内容的に特徴を持った部分がある. 段階の種類毎に作業者の発話パターンに特徴がある.

3.2 システムの設計方針

上記のような問題点を解決し, システムが作業の支援を行なうためには, 以下のような機能が必要であると考えられる.

- 会議中の議論の内容をできる限り記録することで欠落を防止する機能.
- 決定がなされていないような部分の発生を防止する機能.
- 文書全体から孤立するような部分の発生を防ぐ機能.
- 会議と, そこから生産される生産物を対応つけて管理し, 非効率な議論の発生を防止する機能.
- 互いに影響を及ぼす文書の部分同士を管理し, 一方が変更されたら, 他方を検査するようにするれば, 文書の変更による矛盾の発生を検出する機能.
- 会議を段階毎に管理し, ステップ毎に作成する必要のあるプロダクト, 例えば発注文書, 仕様書, スケジュール表などの作成・参照を作業者に促す機能.

上記のような機能を実現するためには, 通常の会議での作業に加えて, 以下のような作業を行なわなければならない.

1. 会議の内容を, できるだけ完全に記録する.
2. 会議のために準備された意見を保存する.
3. 記録した会議の内容を, 上記の問題点を検索しやすい形で構造化する.
4. 構造化された会議の内容を, 会議の進行に従って会議の参加者に提供する.
5. 会議の内容を基に仕様書や議事録などの文書を作成する.

会議の内容の記録を行なうことは, 会議中以外では不可能である. また, 問題点を検索しやすい形で構造化された会議の記録を会議の進行に従って検索する作業も会議外では不可能である. しかし, 記録した会議の内容の構造化や, 文書の作成は, 会議の外でも行なうことが可能である. むしろ, これらの作業は会議外で行なわれることが普通であると考えられる. また, 会議で議論するための話題や, それに対する意見は, 個々の会議の開始以前に作業者が, あらかじめ準備してくることが普通である.

よって, 本論文の支援方式では, 作業形態を以下のように大きく 2 種類に分類する.

会議モード : 複数の作業者が参加する会議形式の作業形態. 口頭を中心とした議論を行ない, それらの記録を記録する. 必要に応じて, 構造化された会議の記録を参照する.

個人モード : 作業者が会議の記録の構造化や必要な文書の作成を行なう. また, 会議で議論すべき提案などを準備する.

図 3.1に単純な作業の流れの例を示す.

1. 個人モードで会議で議論するための提案などを準備する. 必要ならば文書も作成する.
2. 会議モードにおいて, 議論を行なう. この場合において, 作業者の発話や黒板への記述などを記録する.
3. 個人モードにおいて, 記録された会議の記録を, 支援のために構造化する. さらに付加したい情報があれば, その付加を行なう.
4. 個人モードにおいて, プロダクトとなる文書を作成する.
5. 会議モード, もしくは, 個人モードにおいて, 作成された文書の確認を行ない作業を終了する.

続く 3.3 節では, 会議を構造化するためのデータ構造を紹介する. また, 第 4 章では, 会議の記録を構造化するための方法を紹介する. さらに, 第 5 章では, 会議の記録, 構造化, 検索を行なうための方法の一例として作成した計算機システムを紹介する.

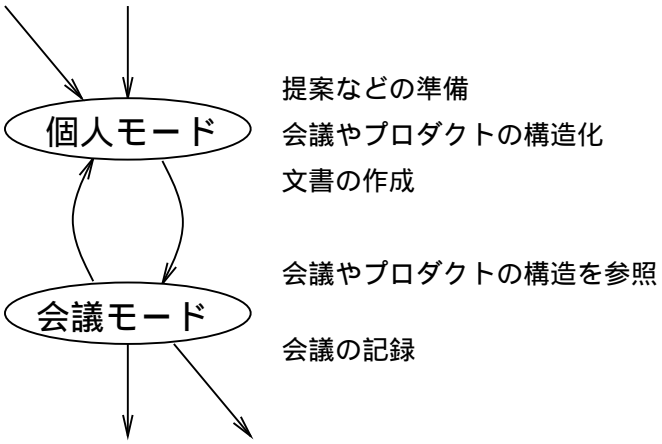


図 3.1: 作業の流れの例

3.3 システムのデータ構造

3.2 節の方針の一解法として, 会議を生産物や作業者と関係付けて図 3.2に示すデータ構造で記述し , ここでのデータ構造を操作することによって, 支援方式を提案する. この節では, データ構造についてのみ説明し, 操作については第 4 章以降で議論する. 構造を記述する言語として個体-関係モデルの記述方法を利用し, 個体を矩形で表現し, 関係を矢印で表現する. 図 3.2中の凡例は直観的には,

主語 が 目的語など を 述語 する.

を意味しており, 以下の説明では,

(主語, 述語, 目的語など)

と略記する場合がある. 関係は一対一から多対多の 4 つの種類があり, 図 3.2中では矢印の形状によって区別している. このソフトウェアの会議の構造は大きく分けて以下の 3 つの部分から構成されている.

参加者の構造**:** 図 3.2の最左列. 会議にどのような役割を持った作業者が参加しているかを構造化して表現.
会議履歴の構造**:** 図 3.2の中央列. 会議がどのような段階を経て進められるかを構造化して表現.
生産物の構造**:** 図 3.2の最右列. 会議で作成される生産物を構造化して表現.

参加者の構造は以下の個体, 関係で構成する.

1. Worker: 会議に参加する作業者を表す. 具体的な値は, 実際の作業者の名前などである.
2. Role: 作業者の役割を表す. 具体的な値の例としては,

   顧客 : 作成対象システムを発注する役割.
   設計者 : 設計仕様書を記述する役割.
   製作者 : 実際にソフトウェアをプログラミングする役割.
   調整者 : 会議の進行を制御する役割.
   書記 : 会議の記録をとり議事録を記述する役割.

   などである.
3. (Worker,play,Role):
   Worker は役割を割り当てられている関係を表現する. ある役割に複数の人数が割り当てられる場合があるため, n:1 対応になっている.

会話履歴の構造は以下の個体, 関係で構成する.

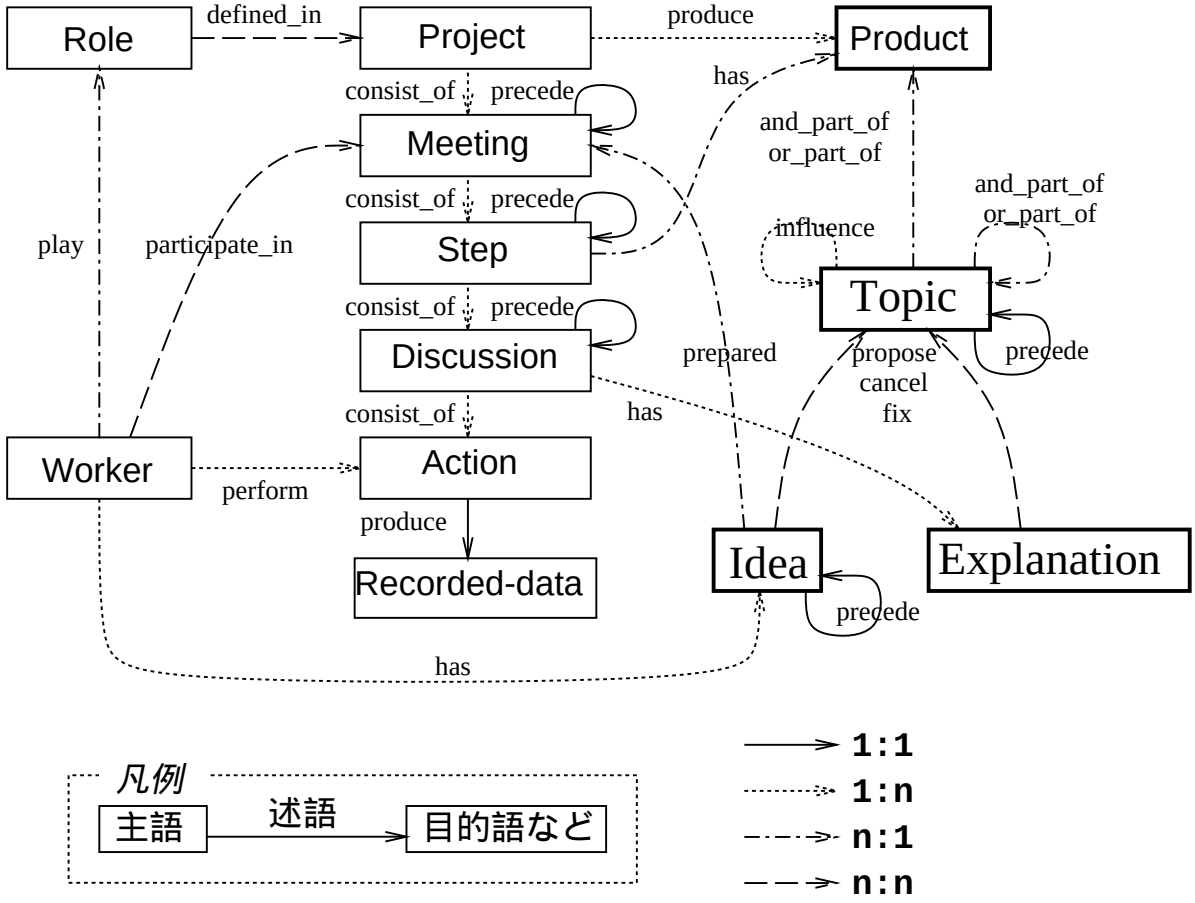


図 3.2: 会議のデータ構造

1. Project: 対象とするプロジェクトを表す. 具体的な値は, “描画エディタ作成プロジェクト”, “ハイパーテキストによる仕様記述ツール作成プロジェクト” などのプロジェクト名をとる.
2. Meeting: 会議を表す. 具体的な値は会議の識別子である. 図中には省略したが, それぞれの値は属性として,

   開始, 終了日時.

   などを持つ.
3. Step: 2.6 節の分析で述べた会議の目的や内容で分割した部分を表す. 具体的な値は, その部分の識別子である. 図中には省略したが, それぞれの値は属性として,

   開始, 終了の日時

   を持つ.

   実際の Step の同定の方法は, 2.6 節の分析に用いた方法に従うが, 詳しくは第 5 章で紹介する.
4. Discussion: 会議中における完結した発話などの行為の列をグループ化した部分を示す. 具体的な値は, その部分の識別子である. それぞれの部分は, その部分だけ聞いて何を話しているのかが理解可能であるように分割する. 図中には省略したが, それぞれの値は属性として,

   開始, 終了の日時

   を持つ.
5. Action: 個々の Worker が行なった発話, 黒板に板書を行なうなどの行為を示す. 具体的な値は, その行為の識別子である. 図中には省略したが, それぞれの値は属性として,

   - 行為の開始, 終了の日時

- 行為の種類: 音声, 画像など.

を持つ.

6. Recorded-data: Action の内容を直接, 音声や画像で記録したデータを示す. 具体的な値は, 音声や画像そのものである.
7. (Action,produce,Recorded-data):
   Recorded-data は Action によって作成される.
8. (Project,consist_of,Meeting), (Meeting,consist_of,Step),
   (Step,consist_of,Discussion), (Discussion,consist_of,Action):
   会議履歴は階層的に構成されていることを表す. Meeting が Step から構成されることは, 2.6 節の分析結果を理由とする.
9. (Meeting,precede,Meeting), (Step,precede,Step), (Discussion,precede,Discussion):
   Meeting, Step, Discussion には全順序があることを示す. すなわち, このデータ構造では, 並行に行なわれる Meeting などを対象としないことを表す. precede の順序を示すため, $d_1 \rightarrow d_2$ と略記する場合があり, これは "d_1の後に d_2が続く" ことを表す.
   Discussion の順序付けは, 2.5 節における議論構造とプロダクトの関係のために行なった. また, 2.4 節の分析における未決事項の発生の防止にも利用する. なお, Meeting, Step の順序付けは, Discussion を順序付けるために行なった.

参加者の構造と会議履歴の構造は以下のような関係を持つ.

1. (Role,defined_in,Project):
   Role は Project 内において定義されている. よって会議によって Role が変わることはない.
2. (Worker,participate_in,Meeting):
   Worker は個々の会議に参加する. 全ての会議に全ての Worker が参加するとは限らない.
3. (Worker,perform,Action):
   Action を行なうのは Worker である.

生産物の構造は以下の個体, 関係で構成する.

1. Product: 会議から作成される生産物を示す. 具体的な値の例としては, 仕様書, 議事録, スケジュール表, 説明書などである.
2. Topic: 議事録や仕様書などの文書の章のタイトルや, それぞれの章に入る文章を示す. その値は識別子である. それぞれの値は，文書の章のタイトルや, それぞれの章に入いる文章や図に相当する属性を持つ.
3. Explanation: 会議において Product として具体化されなかった Topic に関する説明を，具体化した個体を示す．その値は識別子である. それぞれの値は，説明文や図に相当する属性を持つ.
   この個体は Discussion などから得られる設計理由や説明を，文書として明示的に表現したり, 会議開始以前に個々の作業者が提案のために準備した Topic の説明を明示的に表現することに利用する. これによって, 桑名 [37] の主張による明示的でない理由を明示的に記述することができ, 2.3 節の分析における非効率な議論の発生の防止が可能となる.
4. Idea: 個々の作業者が会議のために準備する提案などの基になる考え. Idea がとり得る値の領域は Explanation と同じである.
5. (Idea,precede,Idea): Idea 間には順序がある．これは個々の Worker が Idea を考え付いた順序に相当する．
6. (Topic,influence,Topic):
   Topic は影響を及ぼす Topic と関係を持つ. この関係を持つことによって, 2.5 節で得られた特徴である Topic 間の修正や変更の波及を調べることができる.
   実際にこの関係を構築するための方法は, 2.5 節で利用した方法を用いるが, 詳しくは第 5 章で紹介する.

7. (Topic,or_part_of,Topic), (Topic,or_part_of,Product):
   Topic 間の部分-全体関係を表す. これによって, 選択的な部分や, 互いに補間できる部分を指定する.
8. (Topic,and_part_of,Topic), (Topic,and_part_of,Prodcut):
   Topic 間の部分-全体関係を表す. 共通の Topic の部分となっている兄弟の Topic の関係は and もしくは or として揃ってなければならない.
   and_part_of と or_part_of で構成される Product 以下の Topic の構造は, 木構造に限定する. すなわち,

$$\forall a(a \text{ part_of } b \wedge a \text{ part_of } c \rightarrow b = c)$$

   である. ただし, $a, b, c \in \text{Topic} \cup \text{Product}$ で, part_of は, and_part_of または or_part_of である.
9. (Topic,precede,Topic):
   Topic には全順序関係がある. これは, 最終的に生成する Product である文書が, 文章や図の列であることを反映している.

会議履歴の構造と生産物の構造, 及び, 作業者と生産物の構造は以下のような関係を持つ. また, Topic と Explanation, Topic と Idea の関係についても説明する.

1. (Project,produce,Product):
   Project の最終アウトプットは Product である. 複数の Product が作成される場合がある.
2. (Step,has,Product):
   それぞれの Step は作成すべき Product を持つ. これによって, 発注 Step では発注文書, 仕様説明 Step では仕様書, スケジュール Step ではスケジュール表などに注目することができる.
3. (Discussion,has,Explanation)
   Discussion は, その説明として Explanation を持つ. これによって, 桑名 [37] の主張による明示的でない理由を明示的に記述することができる.
   Discussion は全順序を持つため, 同じ Discussion に属さない Explanation も全順序を持つ.
4. (Explanation,propose,Topic), (Explanation,cancel,Topic), (Explanation,fix,Topic):
   前述の (Discussion,has,Explanation) の関係と組み合わせて, Discussion において Topic が言及されることを表現する. これによって, 2.3 節で述べた非効率な議論を防止する. 言及の種類は以下の 3 種類とする. 詳細な説明は, 続く 3.4 節で述べる.

   propose : Product の一部となる Topic を提案する.
   cancel : 提案されている Topic を取り消す.
   fix : 提案されている Topic を決定とする. もしくは, 提案されている Topic の取り消しを決定とする. これによって, 2.4 節の分析でふれた決定の不明確な事項の発生を防止する.

   これらの関係に関しては以下のような条件を付加する.

   (a) ある Discussion と関係を持つ Explanation と, 1 つの Topic との関係が複数あるのは, 以下の 2 つの場合のみである. そして, それぞれの場合において以下に示す関係間の順序をつける. ただし, e_1 は Explanation, t_1 は Topic とする.
   - e_1 propose $t_1 \wedge e_1$ fix t_1, 順序は prpose → fix.
     直観的な意味は提案されて, 即, 決定されたことを表す.
   - e_1 cancel $t_1 \wedge e_1$ fix t_1, 順序は cancel → fix.
     直観的な意味は提案されて, 即, 否定されたことを表す.

   (b) 上記以外で, ある Discussion と関係を持つ Explanation と, 1 つの Topic の関係は, propose, cancel, fix の中のたかだか 1 つである.
5. (Worker,has,Idea):
   作業者は会議のために準備した提案など基になる考えを持つ.

6. (Idea,prepared,Meeting):

   提案などの基になる考えは, ある会議のために準備される.

7. (Idea,propose,Topic), (Idea,cancel,Topic), (Idea,fix,Topic):

   前述の (Worker,has,Idea), (Idea,prepared,Meeting) の関係と組み合わせて, ある Worker がある会議のために準備した Topic について言及することを表す. それぞれの関係の内容と条件は (Explanation, 関係,Topic) の場合と同様である.

以上のようなデータ構造を用いて会議を記述することで, 3.1 節に示した問題点を表現することができる.

3.4 Topic の可否の決定

ある Topic が議論の中で, どのように言及されたかは , Topic と Explanation の関係で表現されている . また , 会議外で , どのような意見として準備されたかは , Topic と Idea の関係で表現されている . 本システムでは , Topic が , ある時点でどのような決定がされているかを , これらの関係の列から計算する . この節では, Topic と Explanation の関係 , Topic と Idea の関係の意味と, それらの関係の列について説明する .

3.4.1 特定の Topic に関する関係の順序

Topic と Explanation の間の関係は propose, fix, cancel のいずれかである. そして, ある 1 つの Topic と Explanation の間の関係は全順序を持つ. その理由を示す.

1. Meeting, Step, Discussion は全順序を持つ.
2. ある Discussion と関係を持つ複数の Explanation と, 1 つの Topic との関係が複数あるのは, 以下の 2 つの場合のみである. そして, それぞれの場合において以下に示す関係間の順序をつける. ただし, e_1 は Explanation, t_1 は Topic とする.

   - e_1 propose $t_1 \wedge e_1$ fix t_1, 順序は prpose $\rightarrow$ fix.
   - e_1 cancel $t_1 \wedge e_1$ fix t_1, 順序は cancel $\rightarrow$ fix.

   よって, 同じ Discussion と関係を持つ Explanation の中で, ある 1 つの Topic と関係を持つ Explanation は全順序を持つ.
3. 上記以外で, ある Discussion と関係を持つ Explanation と, 1 つの Topic の関係は, propose, cancel, fix の中のたかだか 1 つである.
4. 相異なる Discussion と関係を持つ Explanation の中で, ある 1 つの Topic と関係を持つ Explanation は, 関係している Discussion の全順序に従って全順序を持つ.
5. よって, ある 1 つの Topic と関係している Explanation は全順序を持つ.
6. よって, 1 つの Topic と Explanation の間の関係は全順序を持つ.

同様に , Idea と Topic の関係を以下のような規則で順序をつける .

1. Idea 間には順序がある .
2. (Idea,prepared,Meeting) であり, (Meeting,consist_of,Step), (Step,precede,Step), (Step,consist_of,Discussion), (Discussion,precede,Discussion) から, Idea の順序と Explanation の順序を連接することができる.
3. よって, propose, cancel, fix を全順序つけることができる.

以後, propose, cancel, fix の関係間の順序を, $\rightarrow$ や, precde で表現する場合がある.

3.4.2 関係の意味

Topic と Explanation の関係は 3 種類存在するが, その意味的な区別から次の 5 つのパターンが考えられる.

(1) **propose:** Topic を肯定的に “提案” する Explanation .

(Explanation,propose,Topic)

過去に同様の肯定的な “提案” がされている場合にはその Topic を “参照” する Explanation とみなし , すでにこの肯定的な “提案” が決定事項となっている場合は , その決定の “確認” とする .

これは新たな Topic や既存の Topic について , 肯定的な意見が述べられているが , はっきりとした結論が導かれていない場合, またはすでに可決の結論が導かれている Topic についての結論の確認や , すでに否決の結論が導かれている Topic についてその結論を転覆するような肯定的な意見が述べられた Explanation である .

(2) **cancel:** Topic を否定的に “提案” する Explanation .

(Explanation,cancel,Topic)

過去に同様の否定的な “提案” がされている場合にはその Topic を “参照” する Explanation とみなし , すでにこの否定的な “提案” が決定事項となっている場合は , その決定の “確認” とする .

これは新たな Topic や既存の Topic について , 否定的な意見が述べられているが , はっきりとした結論が導かれていない場合, またはすでに否決の結論が導かれている Topic についての結論の確認や , すでに可決の結論が導かれている Topic についてその結論を転覆するような否定的な意見が述べられた Explanation である .

(3) **fix:** Topic を決定する Explanation.

(Explanation,fix,Topic)

既に提案されている Topic を決定事項とする場合の議論を記述する形式である.

これは, 過去に提案された Topic をまとめて検査して決定する場合などに利用できる.

(4) **propose → fix:** Topic が肯定的に提案され決定される. 以後, これを “可決” と呼ぶ場合もある.

(Explanation,propose,Topic) precede (Explanation,fix,Topic)

(5) **cancel → fix:** Topic が否定的に提案され決定される. 以後, これを “否決” と呼ぶ場合もある.

(Explanation,cancel,Topic) precede (Explanation,fix,Topic)

これらの仕組みは, Topic に対して可決/否決などの属性を持たせる方式と同等である. すなわち, このようにある 1 つの Topic と Explanation の関係の列を考えることによって, それらの属性が持つべき情報を表現できるため, Topic には可決/否決などに相当する属性を持たせていない.

ある Topic の関係の列が以下のような場合, その Topic は決定していないとし, そうでない場合は, その Topic は決定しているとする.

1. 関係の列に fix が含まれていない場合.
2. fix の直前の関係の種類と, fix の後にある関係の種類が異なる場合.

3.5 構造化された会議のデータの例

上記のデータ構造を用いることで, 3.1 節で述べた問題点や特徴を表現することができる例を示す.

- 議論内容の欠落の防止:
  会議における個々の発話などの行為の音声や画像のデータを Recorded-data として直接記録するために, 会議で行なわれた行為の記録の洩れはなくなる. そして, Action, Discussion, Step, Meeting, Project の順番で会議を階層的に内容を整理し, 実際の記録から効率的に生産物の部分を構築してゆくことが可能となる. 構築する方法はシステムに利用者に委ねられているが , 効率的と思われる方法の 1 つを第 4 章で提案する .

- 文書の不明瞭な点の発生の防止:
  全ての Topic は最終的には可決・否決が決定されなければならない. 本論文では, 決定されていない Topic を不明瞭な Topic とする.
  図 3.3 に例を示す. Topic1 は,

  propose, propose

  という関係の列を持ち fix をふくまないのでため, これは不明瞭な Topic である. Topic2 は,

  propose, fix, propose

  という列を持つので決定された Topic である. Topic3 は,

  propose, fix, cancel

  という列を持ち, fix の前後で関係の種類が異なるため不明瞭な Topic となる.

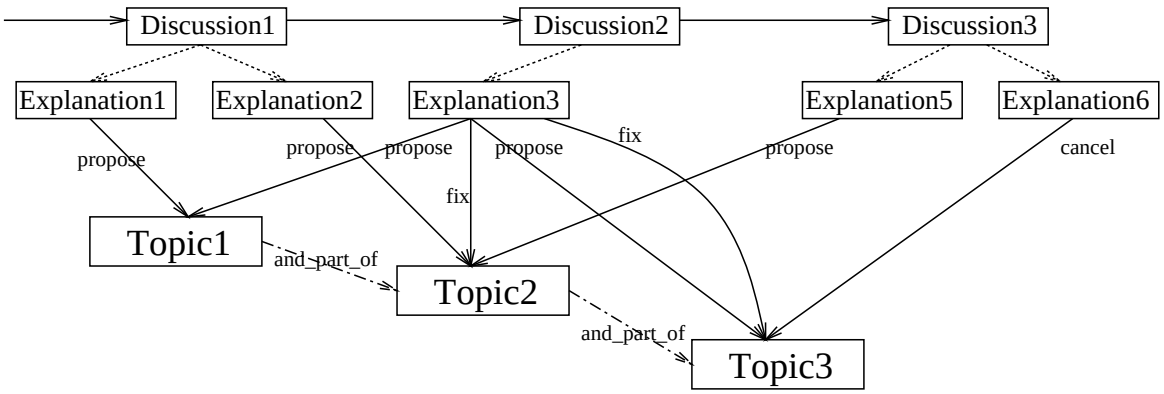


図 3.3: 構造化された会議と不明瞭な点の発生防止

- 議論中で文書全体から孤立するような部分:
  全ての取り消されていない Topic は最終的には, ある Product から and_part_of もしくは or_part_of でたどることができなければならない. 全ての Topic は Discussion と Explanation の関係をたどることによって列挙でき, それらの中で Product からたどれないものは孤立する部分である. 図 3.4に孤立する部分の例を示す.

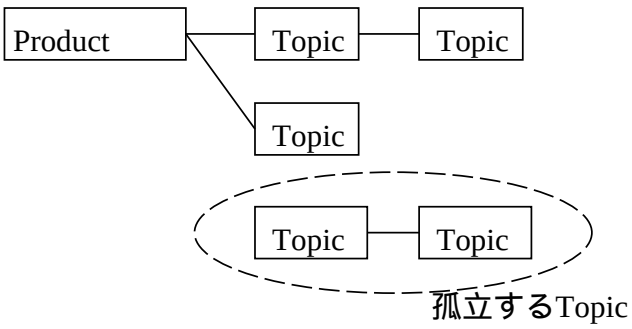


図 3.4: 孤立する部分の例

- 非効率な議論の防止:
  Topic と Explanation の関係の順序を調べることで, 同じ議論の繰り返し (図 3.5) や, 結論のない議論 (図 3.6) や, 決定の度重なる変更 (図 3.7) などを防止できる. 特に, 決定の度重なる変更などが行なわれそうな場合に, Discussion や, その内容を要約した Explanation が決定理由を示す役割をすれば, 無駄な決定の変更などが防止可能だと思われる.

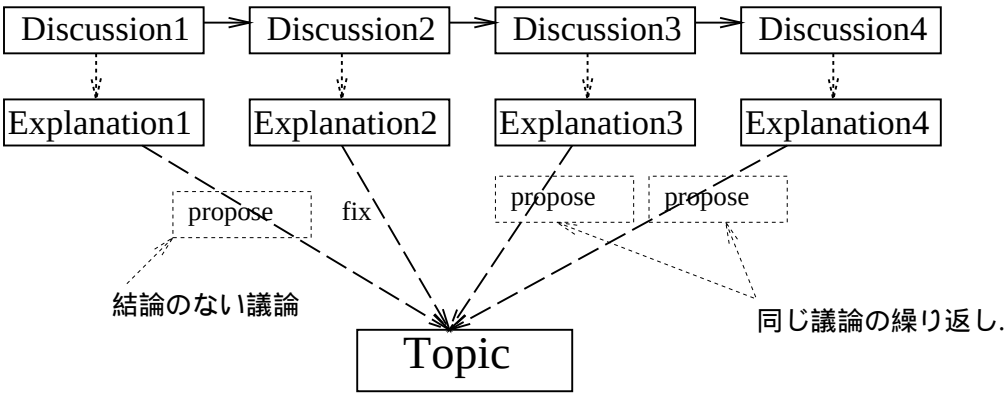


図 3.5: 同じ議論の繰り返しの例

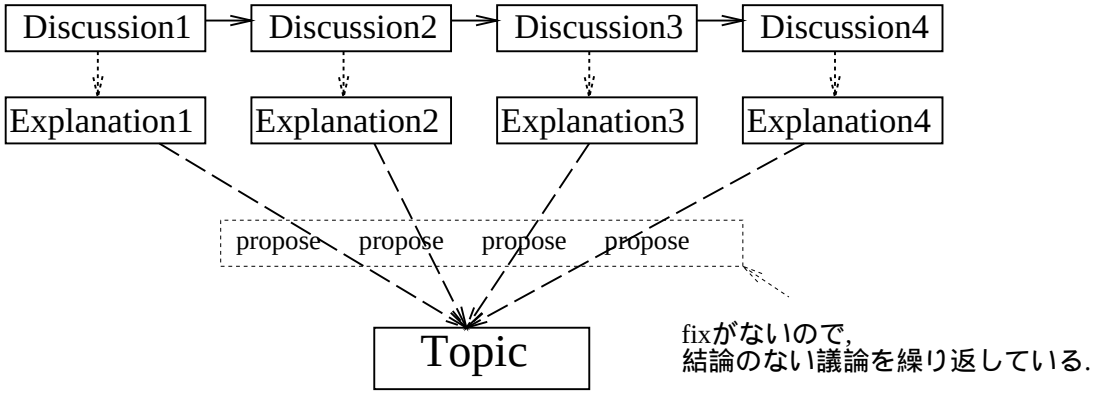


図 3.6: 結論のない議論の例

- 生産物内の一部を変更した場合に, その影響を受けて同様に変更を行なわなければならない部分:
  関係 influence によって, ある Topic の変更や修正によって, 同様に変更しなければならない可能性のある Topic がわかる. 例えば, 図 3.8において, Discussion2 の時点では Topic2, 3 ともに決定事項となっているが, Discussion3 において, Topic3 の決定が覆されている. よって, Topic3 と influence 関係にある Topic2 に関しても再考の必要があると考えられる.
- 会議中の 発注ステップ, 説明ステップ, スケジュールテップなどの内容的に特徴を持った部分
  Step と Product の has 関係によって, ステップ毎に作成する必要のある生産物 , 例えば発注文書, 仕様書, スケジュール表などの作成・参照を作業者に促すことができる. 例えば, 図 3.9 では, Meeting に, Step1 から 3 の 3 つの Step が含まれており, Step1 と Step3 は, Product の 1 つである仕様書, Step2 はスケジュール表と関係を持っている. そして, 個々の Step に含まれる Discussion と関係のある Explanation は, Step が指定した Product 下の Topic と関係がついている.

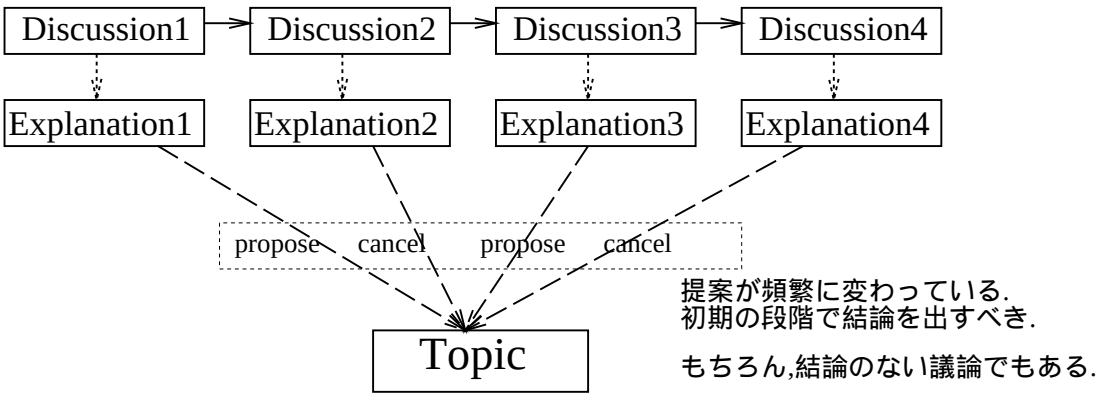


図 3.7: 決定の度重なる変更の例

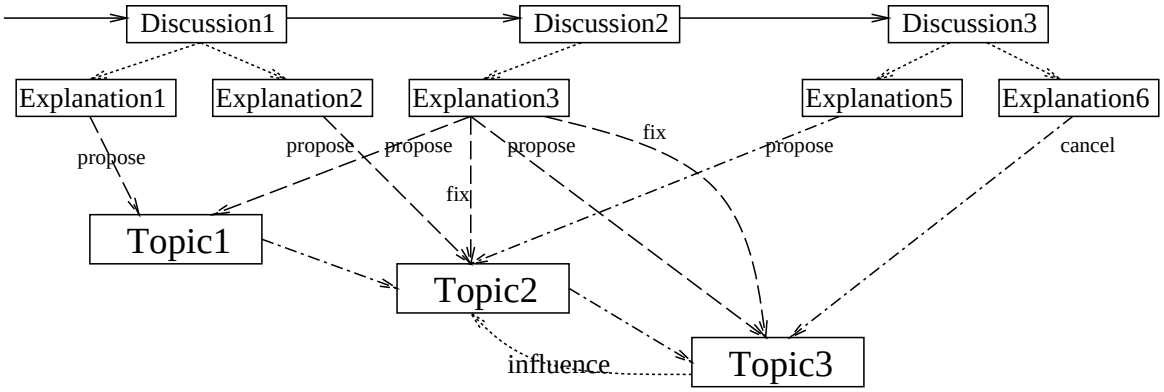


図 3.8: 構造化された会議と影響の及ぶ箇所の検査

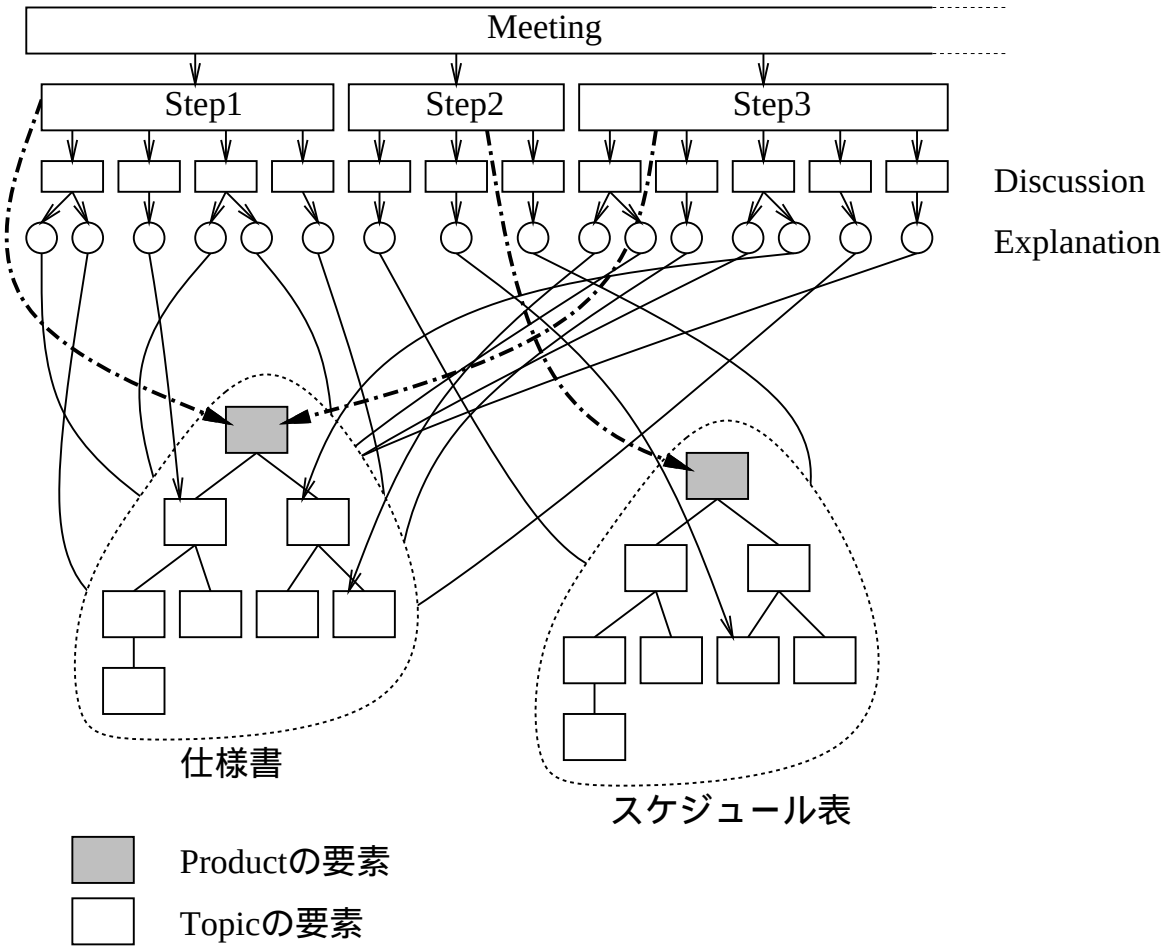


図 3.9: Step と Product の関係の例

第 4 章

システム利用のための方法

この章では, 第 3 章で紹介したデータ構造に従って会議の履歴を構造化し, 会議の履歴から仕様書や議事録などの文書を作成するための効率的な方法の 1 つを提案する . この方法を利用することで第 5 章で紹介するような実現されたシステムを円滑に利用することができる. 構造化された会議のデータを参照/検索する方法は, 第 5 章で紹介するプロトタイプシステムのユーザーインタフェースを通して説明する.

4.1 構造化作業の概要

第 3 章で紹介した構造を用いて仕様などの文書作成作業を行なうことによって, 通常の作業では行なわない作業を行なうことになる. しかし, この通常行なわない作業を行なうことによって, 第 2 章で明らかになった問題点を解決し , 作業の促進を行なうことができる . 図 4.1に, プロダクトの作成者から見た作業の流れを示す. また, 図 4.2には, データ構造から見た作業の流れを示す.

ここで述べる文書作成作業は , 会議の参加者の中の一人の文書作成者によって会議外で行なうことを原則としている . もちろん書記のような役割の作業者が会議の進行に沿って行なっても良いが , その作業の全てを会議中に行なうことは困難であるので , その場合でも作業の一部は会議外で行なうことになる . また , 前回までの会議の記録の処理は必ず次回の会議までに終えておく . そして次回の会議において, 作成された会議の構造化されたデータを利用して不明瞭な事項や未決事項などの問題点の解消を行なう.

1 回目の会議では , 会議履歴の記録だけを行ない , 会議後の文書作成作業で構造データを作る . そして 2 回目以降の会議では , その構造データと議事録や仕様書などの文書である議事録を入力とする . 2 回目以降の文書作成作業では , それ以前の会議履歴と構造データを基にして構造化を行なう . これらの作業を繰り返して , 最終的に不明瞭な事項や未決事項などの問題点を解決し, プロダクトである議事録や仕様書などの文書が作成され, プロジェクトは終了する.

図 4.2に示すように, データ構造から作業の流れを見た場合, プロダクトである議事録や仕様書を作成する作業において, 図中の太線の矢印が通る部分, すなわち Step, Discussion, Explanation, Topic は, 作業者が内容を見て構築して行かなければならない. そして, Topic 部分の構造を基にプロダクトである議事録や仕様書などの文書を作成する. 図中では• の印のあるその他の構造の要素は, 自明のものか, 機械的に構造化することが可能な部分である. 続く節では, 作業者が内容を見て構築しなければならない部分の構造化の手順を紹介する.

4.2 会議の記録を構造化する手順

以下に示すような手順に従って段階的に, 会議の記録とプロダクトを構造化して行く.

手順 1 Meeting を Step に分割

手順 2 Step を Discussion に分割

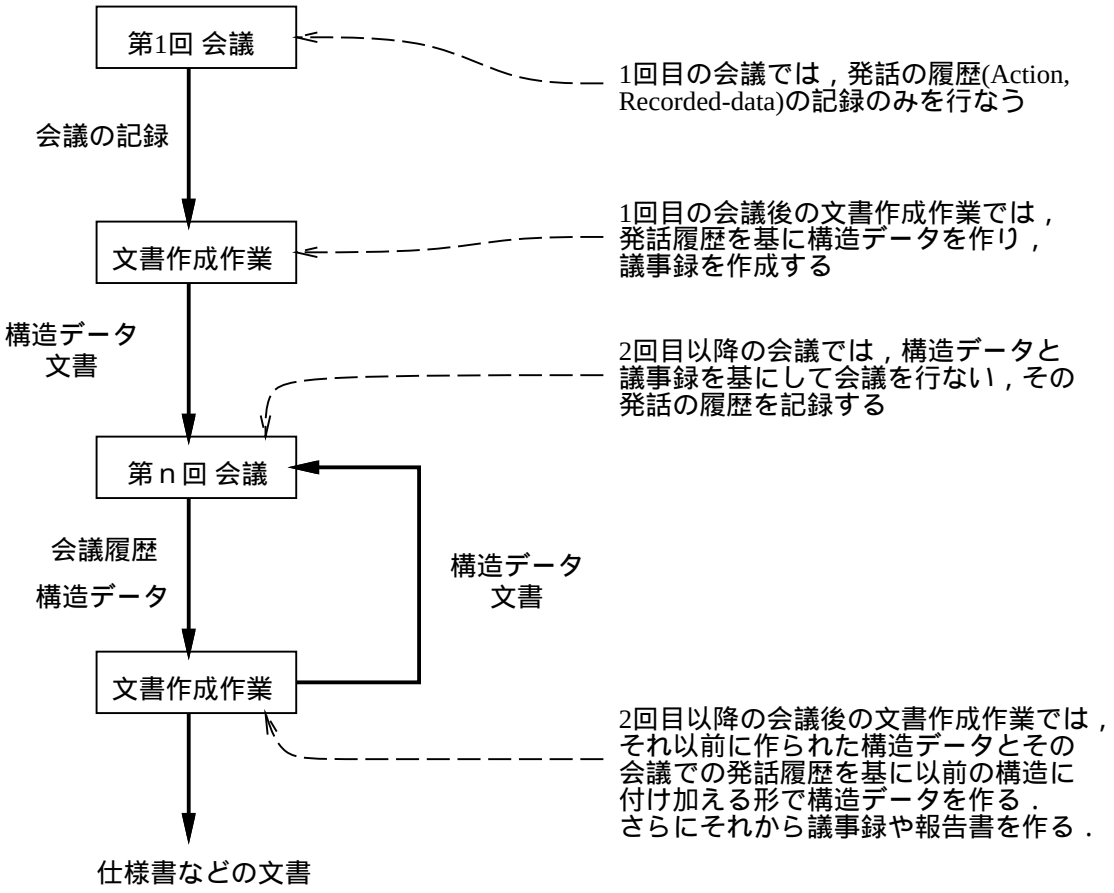


図 4.1: プロダクト作成者から見た作業の流れ

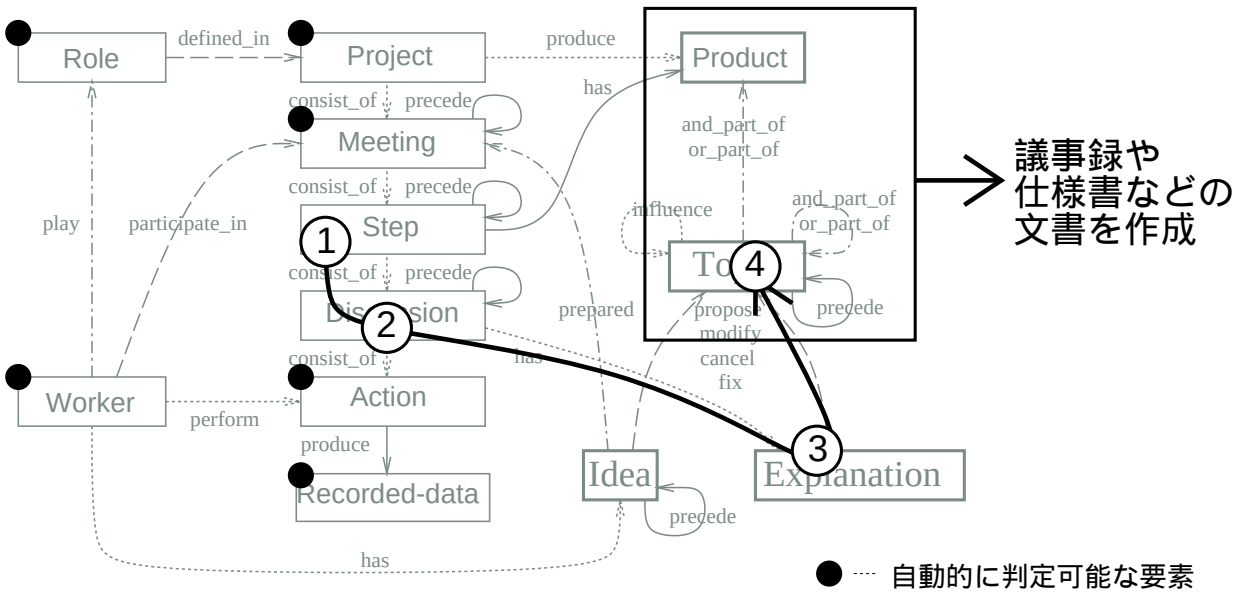


図 4.2: データ構造から見た作業の流れ

手順 3 Discussion から Explanation を抽出
手順 4 Explanation から Topic を抽出
手順 5 Topic 間の階層関係の決定
手順 6 Topic 間の順序関係の決定
手順 7 Topic 間の影響関係の決定

以下に個々の手順の内容を説明する.

手順 1: Meeting を Step に分割

この段階では,

Step, (Meeting,consist_of,Step), Product, (Step,has,Product), (Step,precede,Step)

の値を作成する. 最後の (Step,precede,Step) は, 実時間の順序により自動的に決定される.

作業者が行なわなければならない第一の作業は, Meeting を Step に分割することである. 分割する際には Step 間には時間的な重複のないようにする. 分割を行なう目安として, 2.6節の分析で利用した発話の偏りの情報を利用する. さらに, 会議中に参加者が記録した Step の区切れの情報も利用する.

同時に, それぞれの Step が, どの種類の Product を作成するために利用するかを決定する. 例えば, 仕様書, 議事録, スケジュール表 などである. これは, それぞれの区切られた Step の内容を見て決定を行なうが, 2.6節の分析で得られた発話の偏りの特徴も参考にする.

以降の作業では, 同じ Product の要素と has 関係を持つ Step 毎に作業を進める. すなわち, 対象システムの仕様に関する議論をしている Step だけを集め構造化をし, 次に, スケジュールに関する議論をしている Step だけを集めて構造化を行なう.

手順 2: Step を Discussion に分割

この段階では,

(Step,consist_of,Discussion), Discussion, (Discussion,precede,Discussion), (Discussion,consist_of,Action)

の値を作成する.

各 Step 中の記録 (Action) を，議論のまとまりと思われる部分を Discussion として区切る．

この段階では，後に文書の章・節やその説明文になるものと会議中の発話との対応をつけるためのものであるため，後に Discussion を検索した場合にもその Discussion だけで意味が通じるように区切る必要がある．その際，ある 1 つの Topic について議論されている部分だけを切り出して Discussion とすることが望ましいが, 現実的にはいくつかの Topic について時間的に重複して議論されるようなことが多い. よって, 数個の Topic について議論されている部分として Discussion が区切られることが普通である．1 つの Discussion の目安としては，数十秒間 ~ 2, 3 分間程度である．また Discussion を区切る際には，話題を展開するようなキーワード (「次に」「それでは」など) を目安にする．

Discussion に含まれる Action の開始時間の内の最も小さいものを Discussion の開始時間とし, Discussion に含まれる Action の開始時間の内の最も大きいものを Discussion の開始時間とする. Discussion 間の順序は,

1. 開始時間の大きい Discussion を, より後に続く Discussion とする.
2. 開始時間が同じ場合は, 終了時間の大きい Discussion を, より後に続く Discussion とする.
3. 開始・終了共に等しい場合は, それらを同一視する.

このようにして Discussion に全順序をつける.

ここまでの手順で，図 4.3のような構造が作成される. なお, この図では, (Step,has,Product) の関係と precde の関係は省略してある. precde の関係は左から右に順序付けられているとする.

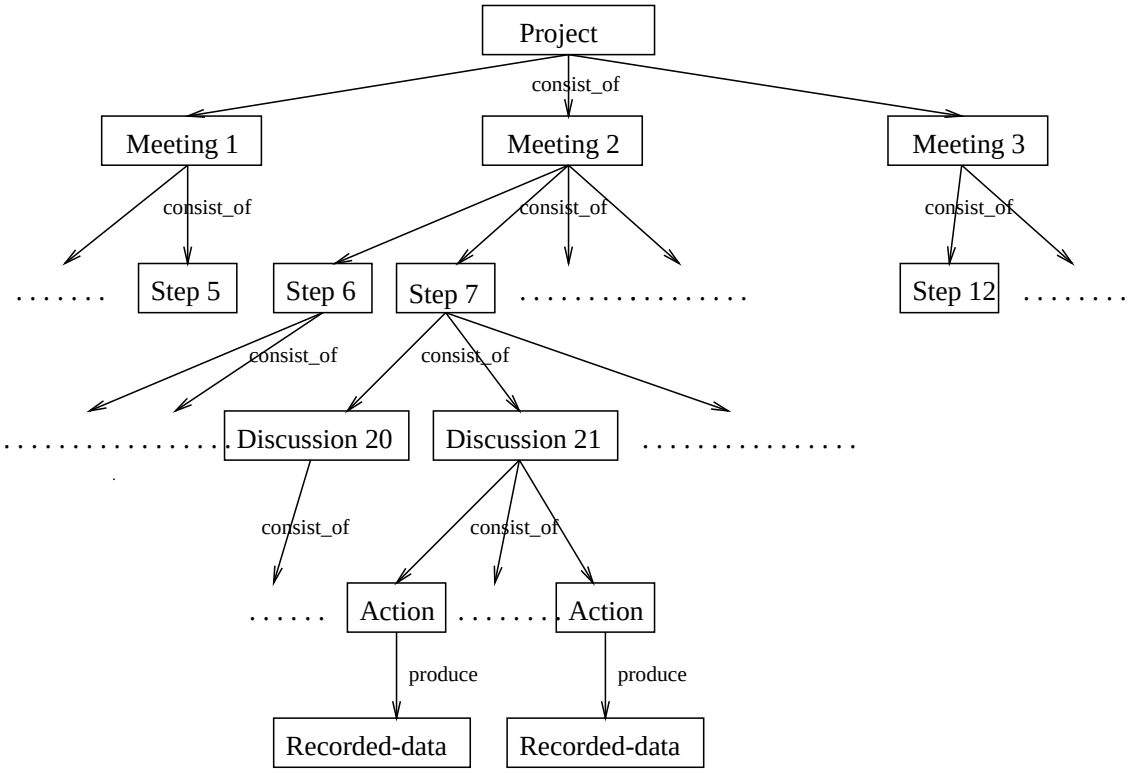


図 4.3: 手順 2 までの作業結果例

手順 3: Explanation の抽出

この段階では,

(Discussion,has,Explanation), Explanation

の値を作成する.

Explanation は, Product である議事録や仕様書などの文書に直接含める必要のない説明や理由を記録するために導入された. Product に直接含める必要のないような情報でも, 会議の効率的な進行を行なうためには有効なので, 記録として残すのである. そして, 構造を作成する作業の中では, Topic を作成するための目安となるラベルとして使われ, 議論を行なう段階では非効率な議論を行なわないために過去の議論のラベルに関するラベルとしても用いられる. よって, Explanation は, 時間的に分割のできない “論理的に” 分割した Discussion であると考えて良い.

1 つの Discussion は, 一般に複数の Explanation を持つが, Discussion の分割でも述べたように, 1 つの Discussion に含まれる Explanation の数を最小にし, かつ, その部分に含まれる Action を参照することによって, 何を話しているのか分かるように Discussion を分割する必要がある. また, 3.3 節に示した Discussion, Explanation, Topic に関する制約条件を守る必要がある.

手順 4: Explanation から Topic を抽出

ある Topic が議論の中でどのように扱われたかの意味を表す部分が, Topic と Explanation の関係である. そして, 個々の Topic が現在, 可決なのか否決なのか, 提案程度で決定が不明確なのかは, その Topic が Explanation との間に持つ関係の順序列から決定される.

Topic を作成する上で, 注意する点は，“否定語” と “可決・否決” の取り扱いである . 例えば「○ ○ は，不必要である .」ということが決定された場合，Topic を ○ ○ とするとこの Topic は “否決” されたとみなされ，逆に

Topic を ○ ○ の不必要性 などとするとこの Topic は “可決” されたとみなされる．このようなことは不都合であるので，Topic 名にはいわゆる “否定語” は極力使わないようにする．以下に Explanation から Topic を抽出する手順を示す.

(1.1) 抽出した Topic に該当する名がそれ以前に出現していない場合:

(1.1.1) 抽出した Topic に該当する名を持つ新しい Topic の要素を生成する.

(1.1.2) 文書作成者が Explanation の意味的な分類から，
(Explanation,propose,Topic) もしくは (Explanation,cancel,Topic)
とする．直観的には, 出された提案を登録することに該当する.

(1.1.3) もし, その Explanation で即, 決定事項となった場合は,
(Explanation,fix,Topic) または (Explanation,fix,Topic)
とする．これは, 提案が決定事項になったことを表す.

(1.2) 抽出した Topic 名がそれ以前に出現している場合:

(1.2.1) 既存の Topic の構造や Explanation の内容を調べることで, 抽出した Topic と同一の既存の Topic が存在するかどうかを調べる.

(1.2.2) もし同一な Topic が存在しない場合, (1.1.1) 以降の操作を行なう.

(1.2.3) 同一であると判断した場合は，(1.1.2) 以降の操作を行なう. 直観的には, 過去に提案された Topic の繰り返しであるということになる.

手順 5: Topic 間の階層構造の決定

Topic 間の階層構造は，会議中で明確に Topic 間の構造について議論されている場合を除き, 原則的には文書作成者によって構築する．しかしながら，議論中で明確に Topic 間の階層構造について議論された場合には, その議論に従う形で階層構造を構築する．

1 つの Topic に関する Explanation の場合

文書作成者の判断により Product または既存の Topic’ との関係を作成する.

(Topic,part_of,Specification) または (Topic,part_of,Topic’) または (Topic”,and_part_of,Topic)

ただし, Topic” は他の Topic などの部分ではないとする. 他の Topic などの部分とならない場合は, 独立 Topic とする．part_of は, 適宜, and_part_of, or_part_of を選択する.

複数の Topic と関係する Explanation の場合

Topic の木構造に関する判断のための情報を, Topic が共通に関係している Explanation の内容から得る. Explanation だけでは不十分な場合は, Explanation の関係する Discussion を通して実際の発話などのデータを参照する. 以下に, いくつかのパターンと例を示す.

Topic の部分関係

(1) 明示的にある Topic の部分として Topic を肯定的に “提案” する場合:
過去に同様の肯定的な “提案” がされている場合にはその Topic を “参照” する Explanation とみなし，すでにこの肯定的な “提案” が決定事項となっている場合は，その決定の “確認” とする．
これはある Topic の部分としての新たな Topic や既存の Topic について，肯定的な意見が述べられているが，はっきりとした結論が導かれていない場合．またはすでに可決の結論が導かれている Topic についての結論の確認や，すでに否決の結論が導かれている Topic についてその結論を転覆するような肯定的な意見が述べられた場合である．
具体的には,

Explanation 1:
「検索ウィンドウには削除ボタンがあったほうが良いのではないか .」

などの発話を核とする Discussion があるとする.

この場合は検索ウィンドウを肯定的に “提案” し，また検索ウィンドウの部分として削除ボタンを肯定的に “提案” し, それらの間に and_part_of 関係を張っていると解釈する. すなわち以下の通りである.

(Explanation1, propose, 検索ウィンドウ)
(Explanation1, propose, 検索ウィンドウ)
(削除ボタン, and_part_of, 検索ウィンドウ)

この場合の構造は図 4.4のようになる.

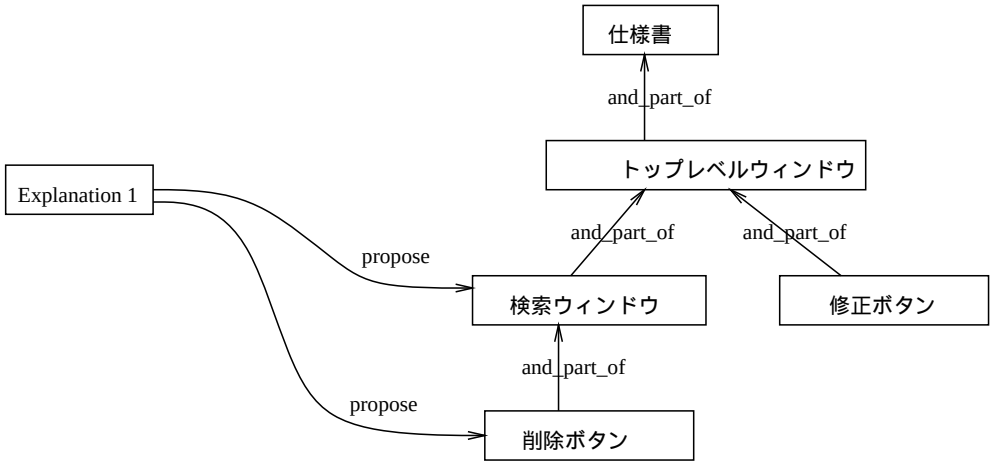


図 4.4: 部分関係の作成の例

(2) 明示的にある Topic の部分として Topic を否定的に “提案” する場合:

過去に同様の否定的な “提案” がされている場合にはその Topic を “参照” する Explanation とみなし，すでにこの否定的な “提案” が決定事項となっている場合は，その決定の “確認” とする．

これはある Topic の部分としての新たな Topic や既存の Topic について， 否定的な意見が述べられているが，はっきりとした結論が導かれていない場合．またはすでに否決の結論が導かれている Topic についての結論の確認や，すでに可決の結論が導かれている Topic についてその結論を転覆するような否定的な意見が述べられた場合である．

具体的には，

Explanation 2 「検索ウィンドウには削除ボタンはいらないと思う .」

などである．
この場合は検索ウィンドウを肯定的に “参照” し，また検索ウィンドウの部分としての削除ボタンを否定的に “参照” していると解釈する．

(Explanation2, propose, 検索ウィンドウ)
(Explanation2, cancel, 削除ボタン)

この場合の構造は図 4.5となる．

(3) 明示的にある Topic が, ある Topic の部分であることを “可決” する場合:

具体的には，

Explanation 3 「検索ウィンドウには削除ボタンを付けることに決定 .」

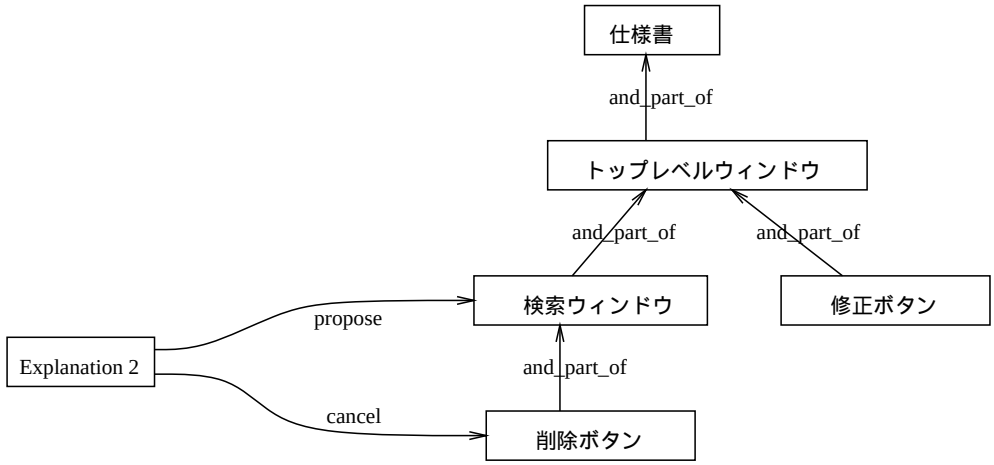


図 4.5: 部分関係の削除の例

などである．

この場合は 検索ウィンドウ を肯定的に “参照” し，また 検索ウィンドウ の部分としての 削除ボタン を肯定的に “提案” し, 決定していると解釈する．

(Explanation3,propose, 検索ウィンドウ)

(Explanation3,propose, 削除ボタン ,

(Explanation3,fix, 削除ボタン)

この場合の構造は図 4.6となる．

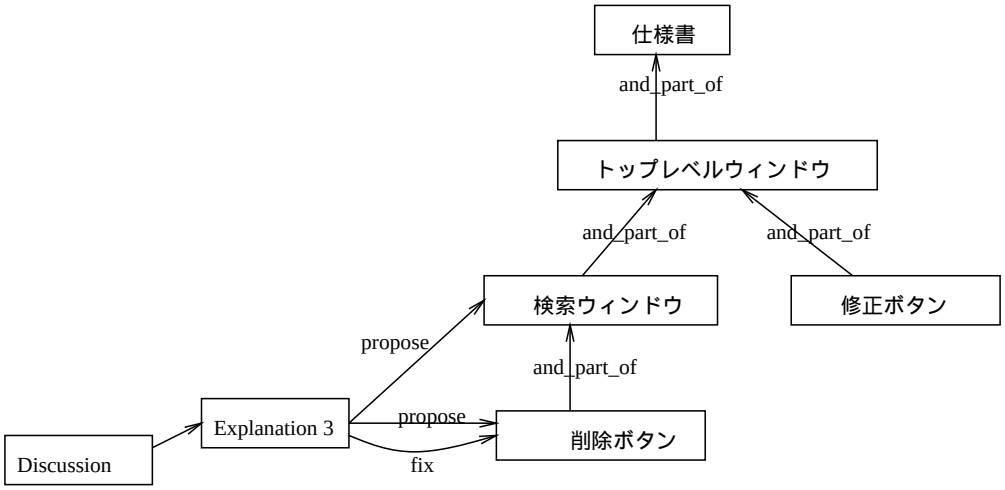


図 4.6: 部分関係の可決の例

(4) 明示的にある Topic が, ある Topic の部分であることを “否決” する場合:

具体的には，

Explanation 4

「 トップレベルウィンドウ には 修正ボタン はつけないことに決定 .」

などである . この場合は トップレベルウィンドウ を肯定的に “参照” し，また トップレベルウィンドウ の部分としての 修正ボタン を “否決” していると解釈する．

(Explanation4,propose,トップレベルウィンドウ)
(Explanation4,cancel,修正ボタン)
(Explanation4,fix,修正ボタン)

この場合の構造は図 4.7となる．

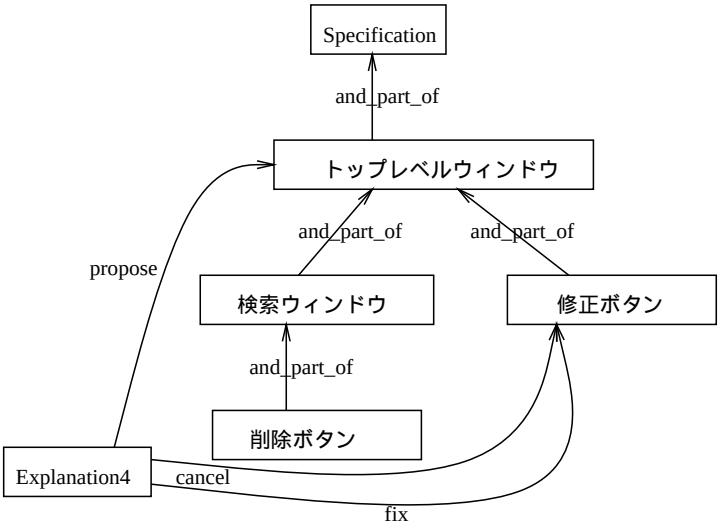


図 4.7: 部分関係の否決の例

また議論中で明示的に Topic の構造中での移動に関する議論が述べられることも考えられる．これは，(2)+(1) と考える．

具体的には，

Explanation 5:
「削除ボタンは，検索ウィンドウではなくて
トップレベルウィンドウに付けた方が良いのではないか？」

などであり，この場合は

「検索ウィンドウには削除ボタンはいらないのではないか？」＋
「トップレベルウィンドウに削除ボタンを付けた方が良いのではないか？」

のように解釈する．

(Explanation5,propose,検索ウィンドウ) (Explanation5,cancel,削除ボタン)
(Explanation5,propose,トップレベルウィンドウ) (Explanation5,propose,削除ボタン)

この場合の構造は図 4.8となる．

Topic の並列・選択肢的な関係

議論の中で

「A 案と B 案のどちらにしましょうか？」

などのように 2 つの Topic が選択肢的に列挙されることがある．その場合それ以降の議論において普通はそのどちらか 1 つを “可決” し，もう 1 つの方を “否決” するような形で進むことが期待される．しかしながら，多くの

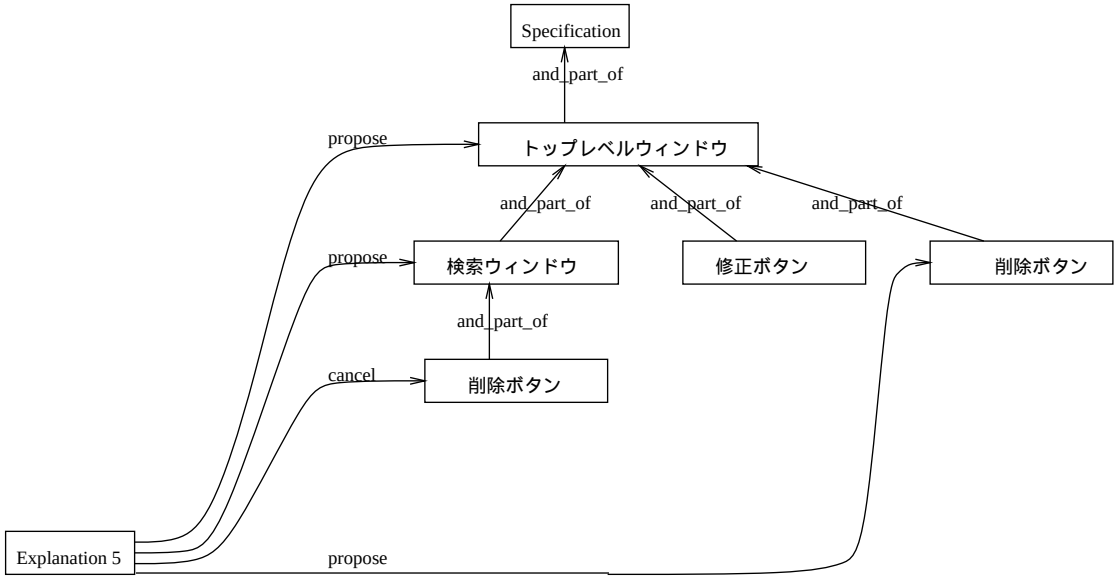


図 4.8: Topic の移動に関する議論の例

場合は明確な一方の選択が行なわれず暗黙の内にもう一方を否決した形で出力文書内にも片方しか記述が残らない場合が多い．また選択肢的に列挙されていても，それ以降の議論においてその両方を “可決” または “否決” するような結論が導かれることもあり得る．

そのため，特に区別して or_part_of 関係を用いる．
A 案, B 案の共通の親を 親 Topic とすると，図 4.9のようになる．

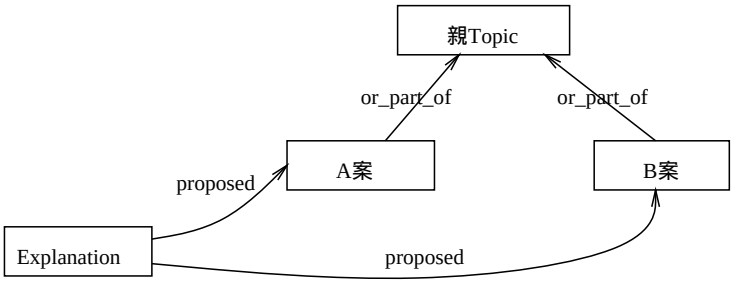


図 4.9: or_part_of の例

ただし, 3.3 節で述べた and_part_of, or_part_of に関する条件を満たす必要がある. また, 部分となる Topic を持たない Topic に, 部分関係をつける場合, and_part_of 関係を選択する.

手順 6: Topic 間の順序関係の決定

Topic は最終的に文書の一部として利用されるため, 文書内の記述の順番を決定する必要がある. これは一般には, 会議外において文書記述者が決定するが, 会議で明示的に指定去れた場合は, それを反映する必要がある. 例えば, 図 4.10 の例では,

(検索ウインドウ ,precede, 修正ボタン)

の順番で文書化されることが議論で明示されたことを示す.

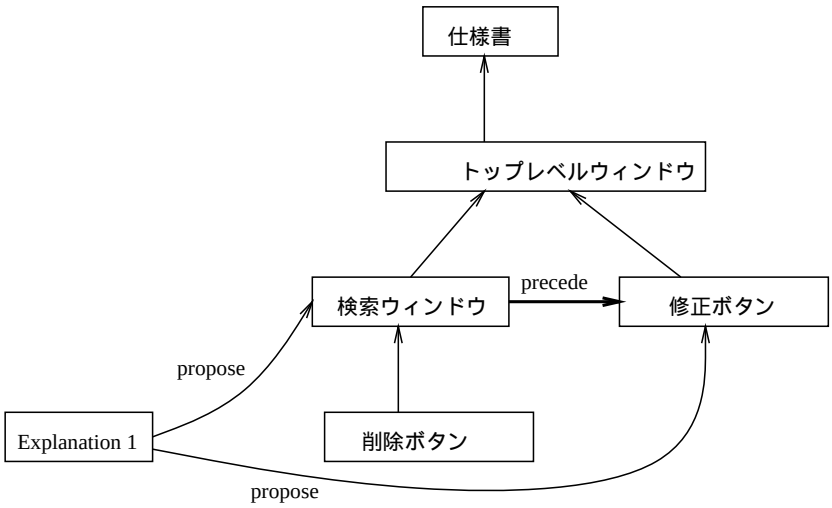


図 4.10: Topic 間の順序の例

手順 7: Topic 間の影響関係の決定

この関係は, 2.5 節の分析結果を基に以下のように決定する.

1. 同じ Discussion と関係を持つ Explanation と関係を持つ Topic は, 相互に関係を張る.
2. $(\mathrm{d}_1, \mathrm{precede}, \mathrm{d}_2)$ であるような Discussion d_1, d_2 において, d_1 と関係を持つ Explanation と関係を持つ Topic と, d_2 と関係を持つ Explanation と関係を持つ Topic の間に相互に関係を張る.

よって, この関係は自動的に計算可能である.

4.2.1 会議の構造の作成例

ここでは，これまでに述べた議事録や仕様書などの文書の構造の構築法のうち手順 4 以降の具体例をあげて説明する．この例では, 会議中に次のような内容の Discussion1 から 4 が時間的に連続して行なわれたとする．Specification は, Product の一種である.

Discussion 1: この議論において, 新規の Topic 1 が提案され, 参加者に承認されて決定事項となった.

Topic 1 は Specification の部分とすることが妥当であると, 作成者が判断した.

手順 **4** 議論内容より,

Explanation 1 と Topic 名として Topic 1 を抽出．

Topic1 が提案されたので,

(Explanation1,propose,Topic1)

として Topic 1 を生成する．

この Topic1 は即座に決定事項となったので,

(Explanation1,fix,Topic1)

とした.

手順 **5** 作成者の判断から,

(Topic 1,and_part_of,Specification)

当然, (Dicussion1,has,Explanation1) である.

Discussion 2: この議論で, 内容的に見ると独立した 2 つの Topic 1, Topic 2 についてふれられた.

文書作成者は, Topic 1 は既存の Topic に同一のものあると判断し, Topic 2 は新規だと判断した. さらに, Topic 1 と Topic 2 には部分関係はないと判断した.

このような場合は，出来れば Topic 1 について議論している部分と Topic 2 について議論している部分とを別の Discussion にすることが望ましい．しかしながら実際には別々に区切ることが難しく，このようになる場合がある．

手順 **4** Explanation 2 より Topic 名として Topic 1 を，Explanation 3 より Topic 名として Topic 2 を抽出．それぞれの Topic 間の構造を調べる．

Topic 1 は既存の Topic で，確認と解釈し，

(Explanation2,propose,Topic1)

とする．

Topic 2 は新規の Topic で，肯定的な提案がされたので，

(Explanation3,propose,Topic2)

とする．

手順 **5** ： Topic2 は Topic1 と関係を持たないため，独立 Topic となる．

手順 **7** ： Discussion2 を基準として，(Topic1,influence,Topic2)，(Topic2,influence,Topic1) とする．

図 4.11に Discussion2 の構造化後の会議のデータを示す．

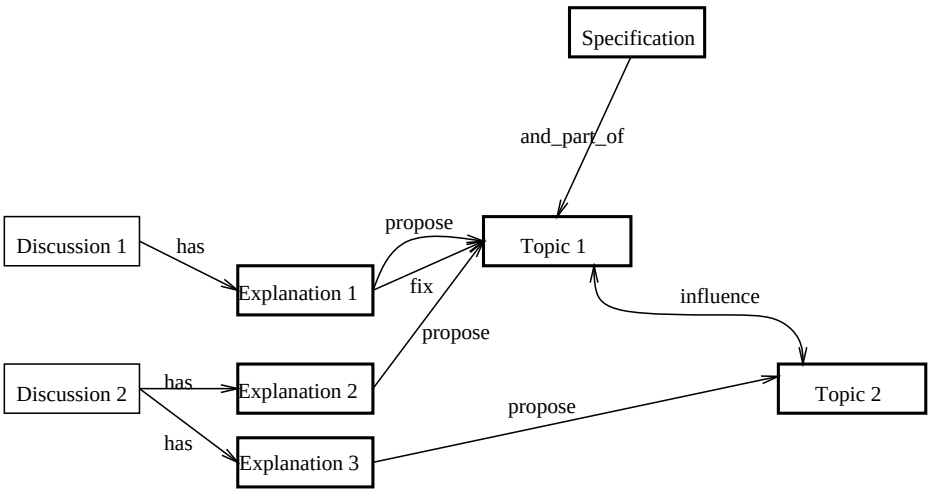


図 4.11: Discussion2 の構造化後の会議のデータ

Discussion 3: Topic 2 は Topic 3 の部分であることがす議論が行なわれ，Topic 3 が必要であることが決定された．

文書作成者により Topic 2 は既存のものであり，Topic 3 は新規のものであり，かつ Specification の部分とすることが妥当であると判断した．

手順 **4** Topic 名 Topic 2 と Topic 3 に対して，

Topic 3 に相当する Topic の要素を作成する．

Topic 2 は既存の Topic であり，肯定的な参照が会議で行なわれたとして，

(Explanation4,propose,Topic2)

とする．

Topic 3 は新規の Topic として提案されたため，

(Explanation4,propose,Topic3)

となり，決定事項となったので，

(Explanation4,fix,Topic3)

手順 **5** ここでは，2 つの Topic の関係に関する議論である．また Topic 2 は既存の Topic で，Topic 3 は新規の Topic である．

議論で明確に述べられた事実より,
(Topic 2,and_part_of,Topic 3)
文書作成者の判断より,
(Topic 3,and_part_of,Specification)

手順 7 Discussion3 より, (Topic2,influence,Topic3), (Topic3,influence,Topic2) の関係を作成する.

図 4.12に, Discussion3 の構造化後の会議のデータを示す.

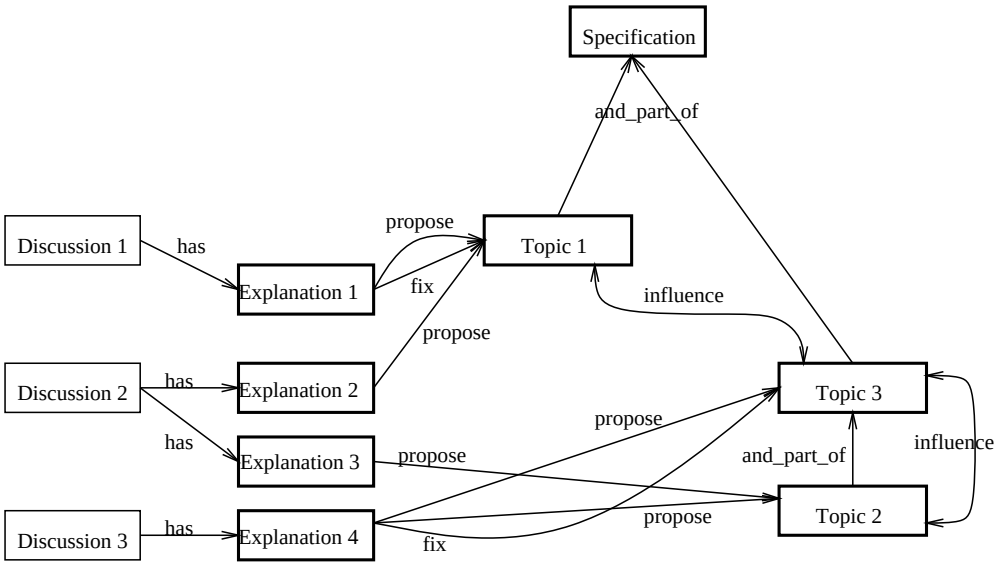


図 4.12: Discussion3 の構造化後の会議のデータ

Discussion 4: 既存の Topic 3 に関する議論が行なわれ，“否定的な意見” が述べられたが明確な結論は導かれなかった．

手順 4 Explanation 5 を作成し, Topic 3 と関係を作成する.
(Explanation5,cancel,Topic3)
Topic 3 は既存の Topic である. これによって, Topic3 に関する関係の列は,
propose, fix, cancel
となるため, Topic3 に関する決定事項である事実は消滅する.

手順 7 Topic3 に関する変更が発生したため, (Topic3,influence,Topic2) の関係を利用し, Topic2 に関する確認作業をする必要がある.

図 4.13に, Discussion4 の構造化後の会議のデータを示す. Topic3 に関する関係の列を点線の矩形で囲んで強調してある.

4.3 構造に記述者の提案を付加

会議外で，作業者が会議で提案された以外のことを生産物として追加する作業が必要である. 例えば，ソフトウェアシステムの設計者などが，会議で一任された部分の試案を次回の会議まで作成する場合などがこれに当てはまる．そのための要素としてデータ構造に Idea が導入された．3.4 節で述べたように，Idea と Explanation に関する関係は同様の意味を持つものが多いため, それぞれ対応付けて説明する.

(Worker,has,Idea) と (Discussion,has,Explanation):

後者は議論の中に, 提案の基となる要素が含まれていることを意味するが, 前者は文書作成者自身に, 提案の基となる要素が含まれていることを意味する.

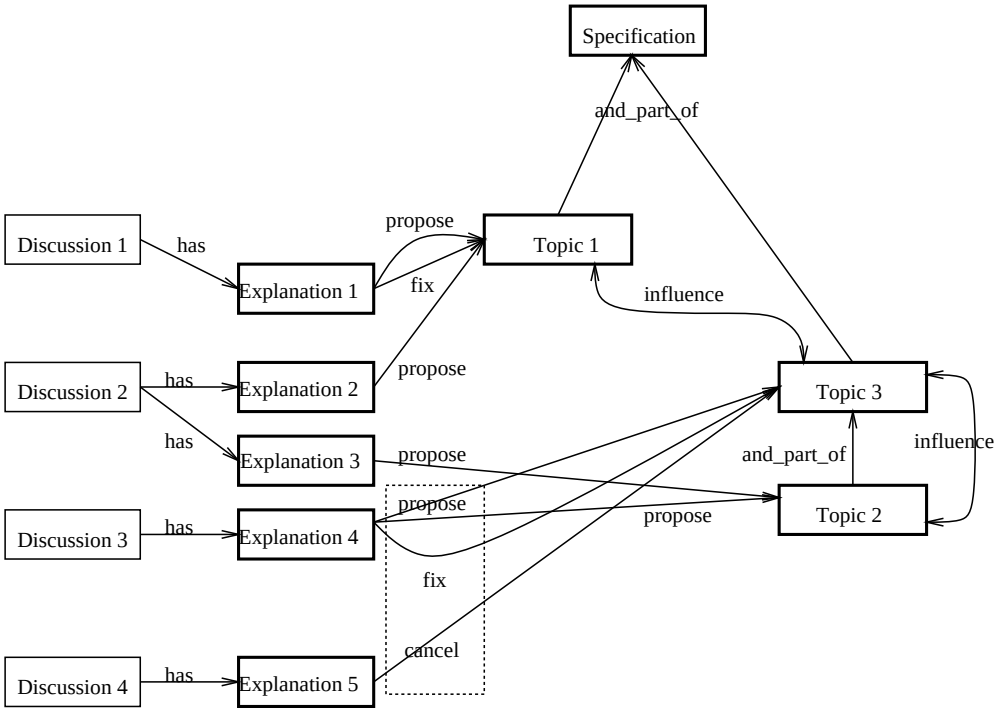


図 4.13: Discussion4 の構造化後の会議のデータ

(Idea,propose,Topic) と (Explanation,propose,Topic):

後者は, 議論による提案の基となる要素が, 新たな提案などを行なうことを意味するのに対し, 前者は, 個人による提案の基となる要素が, 新たな提案などを行なうことを意味する.

cancel, fix も同様である.

このようにして, Explanation と関係を持つ Topic を議論から得られたプロダクトの部分とし, Idea と関係を持つ Topic を作業者が準備したプロダクトの部分とする. よって, 議論を基にして Topic を構造化するのと同様に，会議の始まる前に Topic を追加することが可能となる.

実際には，最終的な決定は会議で行ない, 作業者は意見を準備するにとどめることが望ましいと考えられるので,

(Idea,fix,Topic)

の関係は用いないことが望ましい. また, 複数の作業者が並行して Idea を作成する作業を行なう場合，既存の特定の Topic と Idea の間の関係に順序をつけることが困難となる．よって，作業を並行して行なう場合は，1 つの Topic に関して複数の作業者が Idea を準備することを抑制する必要がある．

4.4 構造からプロダクトを作成

以上のような方法で，文書の構造に当たる Topic の階層構造を構築した後に，実際の印刷物としての文書を作成する．実際の Product としては,

仕様書, 議事録, スケジュール表

などが考えられ, それぞれについて文書を作成する. 文書化の手順を以下に示すが, 手順を追う毎に人間による手間が増加するため, 作成する Product の種類によって, 手順途中でやめるかどうかを判断する.

1. 構造から自動的に文書のプロトタイプを作成:

構造における Topic 間の and_part_of 関係, or_part_of 関係, precede 関係, および, それぞれの Topic と Explanation の間の関係の列を基に, その Topic の可決や否決の情報を作成する. 実例としては, 付録 A に添付したような文書となる.

この文書は会議のいかなる時点でも作成可能であり, 文書化のための人間の作業は必要がない. また, 第 5 章で紹介するプロトタイプシステムのグラフィカルブラウザの表示内容とまったく同じになる.

2. 文書のプロトタイプのフィルタリング:

   プロトタイプの段階では, 個々の Topic に関して以下に示す付加情報を持っている.

   - 決定されているか否かのフラグ.
   - その Topic が肯定されているか否定されているかのフラグ.
   - 構造内で孤立している Topic のリスト.

   これらの情報を用いてプロトタイプのフィルタリングを行なうことができる. 例えば,

   - 肯定的に決定されている Topic のみ取りだし, 仕様書とする. 否定された Topic はシステムに取り込まない部分なので, 必要な情報ではないという観点の, もっとも狭い意味の仕様書である.
   - 決定されていない Topic と孤立している Topic だけ取りだし, 議題一覧表 (agenda) とする.

   などが考えられる. この段階では, どのような情報が必要かのみを作業者は入力すれば良い.

3. プロトタイプの文書化:

   プロトタイプを実際に人間の読む文書に直す作業である. 例えば, Product 部分を “タイトル”, Product の直接の部分となっている Topic を “章”, その下のレベルを “節” のようにして文書化する.

   この段階では Explanation に記述されている内容や, 実際の Discussion の内容を参照することで, より詳細な文書が作成できる.

第 5 章

ハイパー議事録システムの実現

本章では第 3 章に従って, 実際の会議の支援に利用するためのシステムのプロトタイプを紹介する.

5.1 システムの支援対象と作業の流れ

ハイパー議事録システムは , 会議モードと個人モードの作業を支援することは第 3 章で述べた. それぞれの作業形態において必要な機能は共通な部分が多いため, プロトタイプシステムでは , 実際のシステムの機能やインタフェースはどちらの作業形態でも利用可能であるように配慮する.

5.1.1 システムの利用者の分類

このシステムの利用者を以下のように分類する. 会議モードにおける利用者は,

参加者 : 会議に参加する作業者全員.

議論者 : 会議中に実際に作成する生産物に関する議論をする作業者. 顧客, 利用者, 設計者, 製作者などは通常この分類に含まれる.

操作者 (オペレータ): 会議中のシステム全体の動きを管理する作業者. 会議中における板書などを記録する作業を兼任する場合がある.

に分類できる. 個人モードの利用者は,

文書作成者 : 会議外で構造化作業をする作業者. 一般には構造化対象である会議に参加していた設計者などが, ここでの作業を行なう.

参照者 : 会議外で過去の会議のデータを参照する作業者. すでに会議のデータが構造化されている場合は, その構造を利用して会議内容を参照することも可能である. 顧客が最終的に作成された生産物のチェックをする場合などがこれに当てはまる.

に分類できる.

5.1.2 システムの利用の流れ

また, 実際にプロトタイプシステムを運用する状況を考えてみる. 図 5.1に典型的なシステムの利用の流れを示す. このような作業の流れの中で, システムにどのような機能が必要となるかを概観する.

1. 会議において, 参加する Worker を登録を行なう (参加管理機能).
2. 会議において, Worker の発話や黒板への記述などを半自動的に記録する (記録機能).

3. 会議外において, Worker が前回の会議の記録の構造化を行なうためのインタフェースを提供し (編集機能), 次の会議に提示すべき文書を Worker が作成する支援を行なう (文書作成支援機能).
4. 会議外において, 前回の会議の記録の確認を Worker が行なうためのインタフェースを提供する (検索参照機能). 例えば, すでにその会議が構造化されている場合, その構造を簡易に参照できるインタフェースを提供する. また, Meeting を Step に分割するための対話の分布などの統計情報を提供する.
5. 会議において, Worker が現在議論されている Topic と関連ある過去のデータを検索したり, 決定か否かの確認や, 変更を行なった場合の影響が波及する範囲などを参照するためのインタフェースを提供する (検索参照機能).
6. 最終的な生産物を Worker が作成するための文書のプロトタイプを作成する (文書作成支援機能).

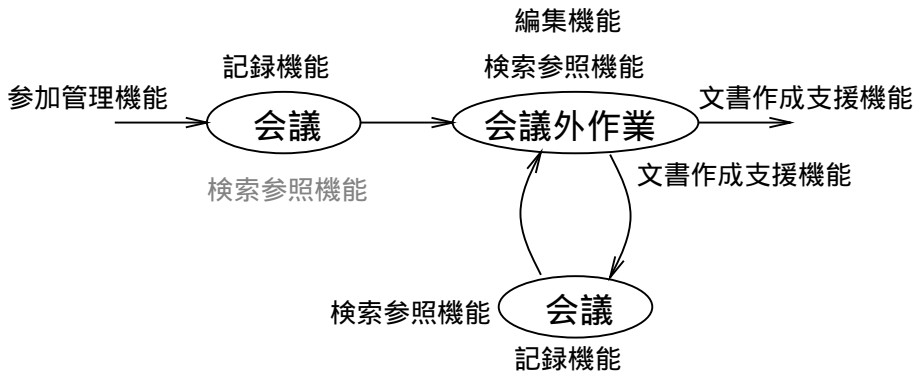


図 5.1: システムを用いた作業例

5.1.3 システムの利用環境

図 5.2に会議モードの利用環境の概観を示す. 作業者はテーブルを囲みそれぞれの計算機を利用することができる. 利用するハードウェアは,

- ヘッドフォン: 過去の Discussion の記録を参照するために個々の利用者に配布される. ワークステーションに接続されており, ワークステーションからの操作により, 必要な音声が参照可能である.
- マイク: 現在の会議の利用者自身の発話を記録するために個々の利用者に配布される. ヘッドフォンと一体化している. ワークステーションに接続されており, 自動もしくは手動で発話を記録することができる.
- ワークステーション: 情報の共有のためにネットワークに接続されている. 構造化したデータの参照や編集に利用する. ビデオカメラで記録した過去の黒板の記述などはワークステーションの画面上に表示することで参照できる. 発話を行なうのみの利用者はほとんど利用する必要はない.
- ビデオカメラ: 利用者の黒板への記述などを記録する. ワークステーションと接続されており, 記録はシステムによって管理される.
- 黒板: 利用者のプレゼンテーションのために用意される. 上記のビデオカメラによって, 必要な黒板の画像を記録することができる.
- 文書: システムのデータを基にして作成されたものも含めて文書が配布される場合がある.

以上のような利用環境において利用者はシステムの機能を用いながら会議を進めて行くことができる.

図 5.3に個人モードの利用環境の概観を示す. 利用するハードウェアは会議モードとほぼ同じであるが, マイクやビデオカメラなどの入力装置は原則として利用しない. 以上のような利用環境において個々の作業者は, 会議において議論された情報の構造化や, 構造化された生産物のデータを会議履歴のデータと対応付けて参照などを行なうことができる.

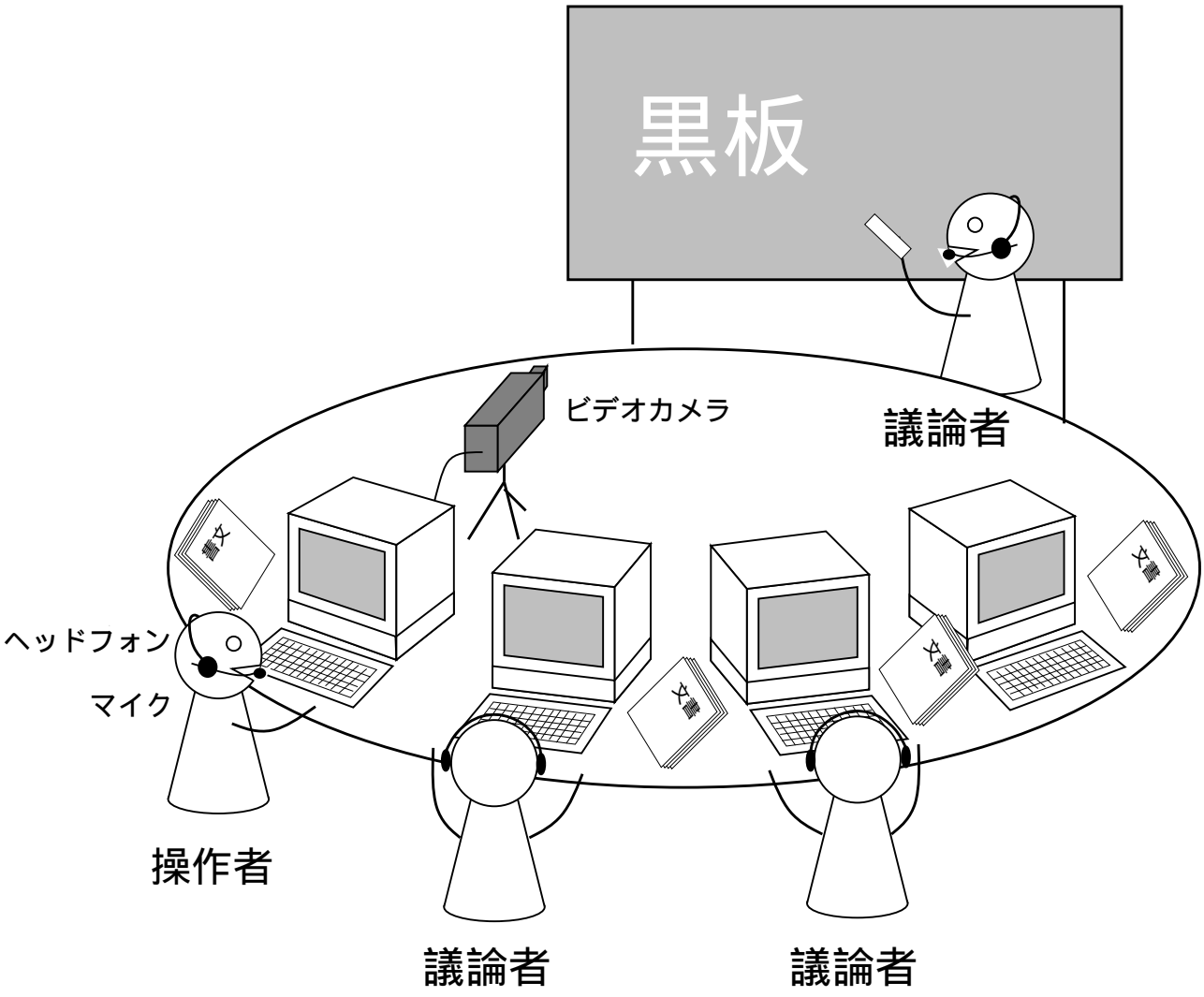


図 5.2: システムの利用環境 (会議モード)

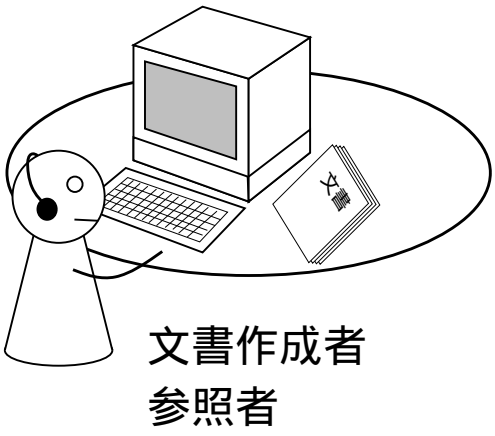


図 5.3: システムの利用環境 (個人モード)

5.2 プロトタイプシステムの機能

5.1節で概観した作業の流れをふまえ, 第 3 章で紹介したデータ構造に従って計算機によるプロトタイプシステムを作成する場合には, 以下のような機能が必要であると考えられる.

1. 会議における Worker の参加管理機能.
2. 会議における Recorded-data の記録機能.
3. Recorded-data を構造化し Discussion, Step などの会議履歴のデータを構築するための編集機能.
4. 会議履歴のデータから Topic などの生産物のデータを段階的に構築するための編集機能.
5. 構造化された会議履歴や生産物のデータの検索参照機能. 対話の分布などの統計情報の表示も含む.
6. 構造化された会議履歴や生産物のデータを用いて, 会議や生産物の問題点を示す問題警告機能.
7. 生産物の構造から, 文書の基となるデータを生成する文書作成支援機能.

続く節では, それぞれの機能の設計と実現について述べる.

5.2.1 参加者管理機能

このシステムは複数の作業者によって利用されるため, どのような Worker が会議に参加しているのか, 記録した発話はどの Worker によるものかといった, 参加者の管理をする機能が必要である. この機能によって, 個々の作業者の会議の参加や, 会議中の Action などを管理することができる.

具体的この機能の実現は, 集中管理を行なうデータベースプログラムによって行ない, 個々の作業者がインタフェースプログラムを利用することによって, 認識する方式をとる.

5.2.2 記録機能

議論された内容が, 生産された議事録や仕様書などの文書から欠落するようなことを防止するために, 会議中の議論の内容をできる限り記録し, 会議履歴として管理できる必要がある. これらの記録は, 会議で議論をしている作業者の活動を妨げないような形で行なわなければならない. また, 検索や文書作成の段階のために, できるだけ議論を直接記録するのが望ましい.

よって, この機能は以下のような方針で実現をする.

1. 個々の作業者の発話などを, 直接音声データとして計算機に記録する.
2. 同様に個々の作業者の動作や黒板などへの記述などを, 直接画像データとして計算機に記録する.
3. 誰がいつ発話を行なったかなどの情報を自動的に個々のデータに付加.

5.2.3 編集機能

記録した会議履歴をデータ構造に従って構造化し, プロダクトを作成し, 対応付けることで, 分析から得られた問題点は解決できる. しかし, 例えば, 第 4 章の方法に従ったとしても, 作業者が内容を判断して作業を行なわなければならない部分が多数存在する. よって, そのような作業を円滑に進めるための強力な編集機能が必要である.

よって, この機能は以下のような方針で実現をする.

1. 文書作成作業の手がかりとなるような情報を会議中に, 適宜付加する機能, 例えば, 議論の切れ目など, 会議履歴にメモができる機能などをつける.
2. 会議履歴のデータを構造化する場合, 該当する議論の音声や画像データを容易にアクセスできる.
3. プロダクトのデータを構造化する場合, 関係する会議履歴のデータを容易にアクセスできる.
4. 複数の作業者で同時に編集する方式は採用せず, 設計者などの作業者 1 名が会議履歴およびプロダクトの構造を編集する. ただし, 1. の構造化のための手がかりとなる情報は, 個々に編集できることにする.

5.2.4 検索参照機能

構造化された会議履歴や議事録や仕様書などの文書のデータ構造の検索参照が簡易にできるインタフェースを提供し, 効率的な議論, 文書作成作業の促進をする. 主な機能を以下に示す.

- 構造化したデータの参照や検索.
- ある Topic に関する過去の Discussion の内容を Explanation として参照.
- ある Topic に関する過去の Discussion の内容を音声/画像データとして参照.
- Topic の変更によって影響を受ける他の部分の提示機能.
- Topic に関する fix か fix でないかの表示.
- ある Discussion から生産された構造 (Explanation,Topic) の参照.
- ある区間での作業者の対話の分布などを表示する機能.

検索参照機能実現の注意点として,

1. 複数の作業者で, 同一のプロダクトを作成するために, 作業者によって情報の食い違いが出ないように参照を行なうことができる必要がある.
2. 表示しているデータが, 編集作業によって変わった時, すぐに更新できるようにする.
3. 会議履歴の構造は, 時間に沿った形で, 一目でわかるように表示する.
4. Action に対応する Recorded_data は容易に参照できなくてはいけない.
5. 議事録や仕様書などの文書の構造を一目で見られるような機能が必要.

5.2.5 問題警告機能

すでに終了した会議の記録を構造化した結果, 作成中の議事録や仕様書などの文書に不備な点, 矛盾発生の可能性がある点が発生した場合に，それらを指摘する．

- 未決な Topic の指摘.
- 不明瞭な Topic の指摘.
- Topic の変更や修正によって矛盾が発生する部分の指摘．

である.

問題警告機能実現の注意点として,

1. 未決な議論である決定されていない Topic の指摘と, 決定するかどうかの確認.
2. 不明瞭な議論としてプロダクトの構造から孤立する Topic の指摘, その扱いについて議論を促す.
3. それぞれの Topic に関する propose, fix, cancel の関係を調査し, その Topic に関する決定が変化したかどうかを調べる. もし, ある Topic の決定に変化が生じた場合には, その Topic と影響関係にある Topic を, 再検討するように警告を行なう.

が考えられる.

5.2.6 文書作成機能

文書作成に関しては, 付録 A に示すような, プロトタイプ文書をシステムが作成し, あとは文書作成者がテキストエディタなどを使って文書を作成する方式をとる.

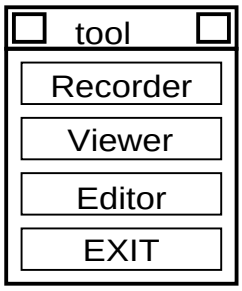


図 5.4: システムのメインボード

5.3 プロトタイプシステムのユーザーインタフェース

個々の作業者が利用するシステムのユーザーインタフェースを紹介する. このインタフェースは, 会議モード, 個人モード, 作業者の役割に関係なく共通である. インタフェースは Multi-Window System を用た階層型のメニュー形式であり, マウスによるボタン押し, およびキーボード打鍵によって操作を行なう.

プロトタイプシステムを立ち上げると, 図 5.4に示すウィンドウが開かれる. 各ボタンは次のような機能を持つ.

- Recorder ボタン: 記録機能を行なうレコーダが立ち上がる.
- Viewer ボタン: 検索機能を行なうビューワが立ち上がる.
- Editor ボタン: 編集機能を行なうエディタが立ち上がる.
- EXIT ボタン: インタフェースの終了.

5.3.1 レコーダ

会議履歴を記録するシステム (図 5.5) である. 各ボタンは次のような機能を持つ.

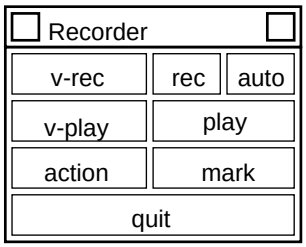


図 5.5: レコーダ

- v-rec ボタンで, 画像録画モードになる.
- v-play ボタンで, 画像再生モードになる.
- rec ボタンで, 音声録音モードになる.
- auto ボタンで, 自動音声録音モードになる.
- play ボタンで, 音声再生モードになる.
- action ボタンで, 現在の会議の履歴を見るウィンドウ (Action_viewer) が開かれる.
- mark ボタンで, 会議履歴にメモをつける.
- quit ボタンで, 終了する.

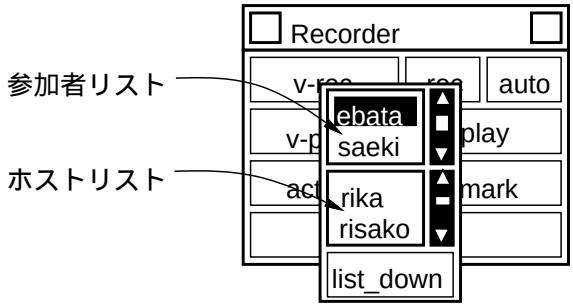


図 5.6: 画像録画

画像録画モード

画像を記録する. このモードを指定すると, 現在会議に参加している利用者のリストと, ビデオカメラが接続されている計算機のリストが表示される (図 5.6). 画像を記録する時は, まず, 誰による図示かと言う意味で利用者をリストから選ぶ. そして, どのビデオカメラの画像を記録するかリストで指定すると, そのビデオカメラの画像が指定した利用者による Action として記録される. 参加者リストは, デフォルトでは自分に選択されている. list_down ボタンでこのモードを終了する.

画像再生モード

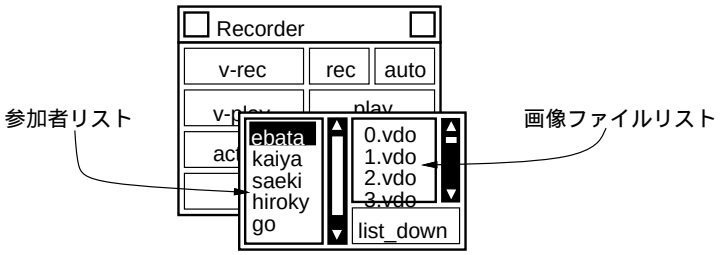


図 5.7: 画像再生

記録した画像データを再生, 画面に表示する. このモードを指定すると, 誰の図示かを指定する参加者リストと, 選択されている利用者による図示の画像データの id (ファイル名) のリストが表示される (図 5.7). 利用者を選択し, 指定した利用者の画像データリストから id を選択することで, 画像が表示される. 表示された画像を消す時は, その画像上でマウスをクリックする. 利用者リストは, デフォルトでは自分に選択されている. list_down ボタンでこのモードを終了する.

音声録音モード

音声を手動で録音する. 作業者それぞれに与えられているマイクロフォンから音声が録音される. ボタンを押すと録音, もう一度押すと停止で, 録音中はボタンがへこんだ状態になっている. 録音された音声データは, 録音した作業者の Action として記録される. list_down ボタンでこのモードを終了する.

自動音声録音モード

音声を自動で録音する. 利用者が何か発話することで, 自動的にそれぞれの作業者のマイクから録音し, 発話をやめると停止する. 通常, 会議ではこのモードにしておく. ボタンを押すと自動録音モード, もう一度押すと解除で, 自動録音モード 中はボタンが押された状態になっている.

音声再生モード

記録した音声データを再生する. このモードを指定すると, 誰の発話かを指定する参加者リストと, 選択されている利用者による発話の音声データの id (ファイル名) のリストが表示される. 利用者を選択し, 指定した利用者の音声データリストから id を選択することで, 音声が再生される. 再生された音声を中断する時は, list_down を押す. 参加者リストは, デフォルトでは音声を再生した利用者に選択されている. list_down ボタンでこのモードを終了する. 操作は画像再生と同じである.

会議履歴参照ウィンドウ (**Action_viewer**)

図 5.8: Action_viewer

現在進行中の会議のそれまでの Action (発話・図示) を利用者ごとに分けて時系列で表示する (図 5.8). Action は棒グラフのように表示され, その長さが発話の長さに対応する. 各利用者において上段に表示される Action は図示, 下段に表示される Action は発話となっている. また, 各 Action はボタンになっていて, 押すことで発話または図示の音声・画像を再生し参照できる. down ボタンで終了する.

メモ

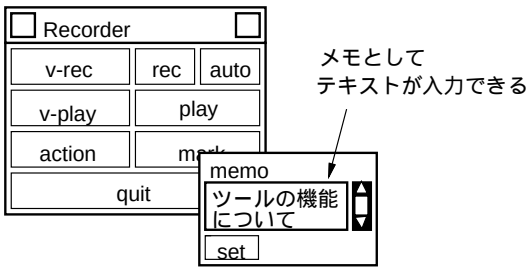


図 5.9: 会議中にメモを付加

議論の切れ目や議論のポイントなど, 会議外の文書作成作業の手がかりになるようなメモを会議履歴に加える. ボタンを押すとテキスト入力を受け付けるウィンドウが開き (図 5.9), set ボタンでその時間でのメモとして記録される. メモのテキストは特に入力しなくても構わない. 付けられたメモは, Explanation として , 議論と関係をつけることができる . メモをつけるのは基本的に操作者が行なうこととするが, 他の利用者も発話の切れ目など明らかな場合は自分でつけても構わない.

5.3.2 エディタ

会議履歴から議事録や仕様書などの文書のデータ構造を構築するための部分である . エディタを立ち上げると, まず, 過去の Meeting のリストが表示される. リストからエディットする会議を指定すると, その会議の Action_editor (前出の Action_viewer と基本的な機能は同じ) が表示され (図 5.10), つぎのような作業ができる.

- set_dis ボタンで, Discussion 登録モードになる.

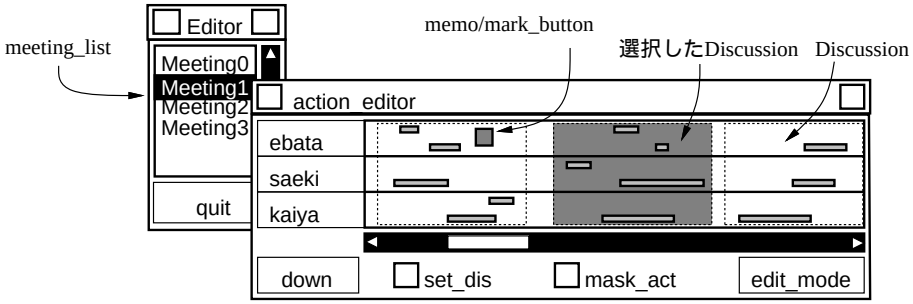


図 5.10: Action_editor

- mask_act ボタンで, Action のマスクモードになる.
- Discussion 上でマウスの真中ボタンを押すと, その Discussion を指定できる. 重複がある場合は続けて真中ボタンを押して選択する. 指定した Discussion は色が反転する.
- edit_mode ボタンで, 指定した Discussion に関する登録/検索モードになる.
- Action_editor 上の任意の場所でマウスの右ボタンを押すと, その場所にメモを書き込むことができる. 登録方法や機能はレコーダの mark ボタンと同じ.

discussion 登録

Discussion を登録する. set_dis を押すと Discussion 登録モードとなり, 登録したい Discussion の中の最初と最後の Action を Action_button で指定すると, その Discussion は枠で囲まれデータベースに登録される. Discussion を 1 つ登録するとこのモードは解除される. また, Discussion の登録の途中, set_dis ボタンを押すとこのモードは中断され, 解除される.

Action のマスク

無意味な Action を Action_viewer や Action_editor で表示させなくする機能である. act_mask ボタンを押して Action マスクモードにして, Action_button でマスクする Action を指定する. このモードは 1 つの Action をマスクすると解除される, また, マスクしなくても act_mask ボタンをもう一度押すことで解除できる.

無意味な Action とは, 無録音の音声データや「あ~」とか「う~」とかいった議論に関係ない意味のない発話などのことをいう. 特に会議で自動録音で記録を行なった場合, 録音レベルや話し方などによってこのような無意味な Action が履歴に多く記録されてしまう. そのため, 次回会議などに情報として見やすいように余分な Action は整理した方が好ましい.

指定した Discussion に関する登録/検索モード

本システムで扱うデータ構造は, 議事録や仕様書などの文書のデータ構造のそれぞれが, 会議のある部分の議論 (Discussion) と対応づけられている. このモードはある Discussion において, それに対応する議事録や仕様書などの文書の構造を作ってゆくためのものである. 議事録や仕様書などの文書の構造はすべてこの作業の繰り返しでつくられる.

このモードを立ち上げると, 図 5.11のようなウィンドウが開かれ, 次のような作業ができる.

- Topic ボタンで, Topic の登録
- Explanation ボタンで, Explanation の登録
- Relation ボタンで, part_of 関係の登録

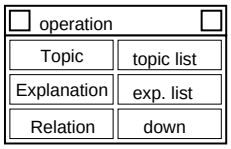


図 5.11: Operation

- topic list ボタンで，現在注目している Discussion で言及されている topic を列挙
- exp. list ボタンで，現在注目している Discussion で言及されている explanation を列挙．

Topic の登録

Explanation を指定して, そこで議論される Topic について propose, cancel, fix を決めて登録する. 図 5.12のようなウィンドウが開かれ, 次のような手順で登録する.

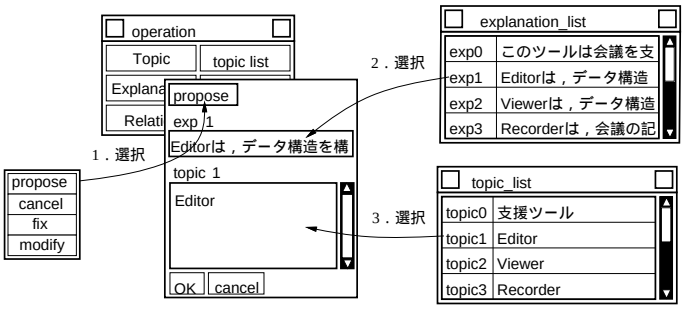


図 5.12: Topic の登録

1. propose, cancel, fix を選択肢から選ぶ. fix されている Topic の propose, cancel を変更した場合, fix は自動的に解除される.
2. Explanation を Explanation_list から選択する. この場合の Explanation_list は指定した Discussion の Explanation のみ.
3. Topic を入力する.
   - 既存の Topic に関する議論の場合, Topic_list から選択する.
   - 新出の Topic に関する議論の場合, キーボードで直接入力する.
4. OK ボタンで登録, cancel ボタンで取り消し.
5. modify ボタンで，任意の Topic のラベルを変更することができる．この場合は，Explanation などを選択する必要はない．

次のような操作を行なう場合, 警告, エラーウィンドウを開いて, 作業者に確認, 警告を行なう.

- 子 Topic を持つ既存の Topic を cancel するとき, 子 Topic の存在を警告する.
- 子 Topic を持つ既存の Topic を cancel するとき, 子 Topic の扱いについて確認をする.

Explanation の登録

図 5.13のようなウィンドウが開かれ, Discussion の要約 (Explanation) を登録する. 手順としては,

1. 指定してある Discussion の要約を, キーボードから直接入力する.

2. Explanation 登録をやめる時は cancel ボタン.
3. 入力が終ったら OK ボタンで登録.
4. Explanation の内容 (文字列) の変更もここから選択される．

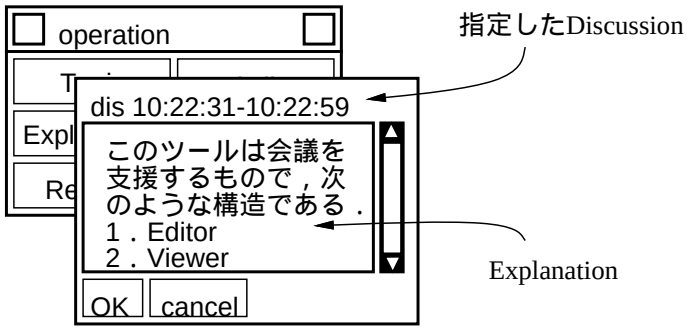


図 5.13: Explanation の登録

関係の登録

議事録や仕様書などの文書の木構造を構築するために，Topic 間の関係を登録する．

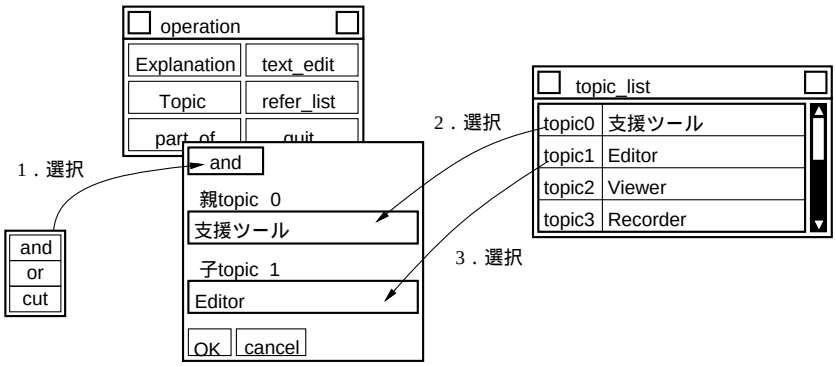


図 5.14: 関係の登録

図 5.14に示すインタフェースを通して，次のような手順で登録する.

1. and, or, cut を選択枝から選ぶ. and は and_part_of 関係であり，or は or_part_of 関係である．cut で既存の関係を切断する．
2. 親の Topic を Topic_list から選択する.
3. 子の Topic を Topic_list から選択する. ただし，and, or の場合は，親の Topic を含めた親の Topic の祖先に当たる Topic と，既に，親の Topic を持つ Topic は選択肢から除外される．cut の場合は，既存の関係のみを表示するような選択肢になる．
4. OK ボタンで登録され, cancel ボタンで取り消される．

topic list, exp. list

現在注目している Discussion で言及された Topic や Explanation を図 5.15のように列挙する．これは編集機能ではないが，既にある Topic や Explanation の内容を参照して，新しい Topic などを登録する場合の支援となる．

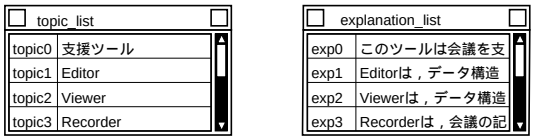


図 5.15: topic list と exp.list

5.3.3 ビューワ

会議中または文書作成作業において，ハイパー議事録を検索・参照するためのインタフェースである．このインタフェースは，図 5.16に示すブラウザを中心に，ハイパー議事録のデータを提供する．現在作成中の議事録や仕様書などの文書の構造を表示する. ブラウザは以下のような情報を表示している．

図 5.16: Browser

- 決定 (fix) されている Topic の表示: 図中の Topic10, 112, 113, 114 のように，Topic 番号の縁を強調表示．
- 否定 (cancel) されている Topic の表示: 図中の Topic28 のように，Topic 番号を X 28 のように表示．
- それ以外の Topic は提案されているが，決定されていない Topic.
- or_part_of 関係の表示: 図中の Topic15 の子に当たる Topic16, 17, 18 のように縦線を付加．
- and_part_of 関係は上記以外の縦線が付加されていない関係．
- 孤立している Topic の表示: Topic28, 33 のように横線を付加．

また，以下のような機能を提供している．

- search ボタン: Topic の番号によるブラウザ上のサーチ．
- word search ボタン: Topic の文字列によるブラウザ上のサーチ．

- redrawボタン: 編集作業によって Topic の構造の変化した場合の，再表示機能の On/Off．デフォルトでは On になっている．
- printボタン: 文書の構造をプリンタに出力．
- Topic 番号のボタン: 指定された Topic に関する詳細な情報を表示するための Topic window を図 5.17のように表示する．

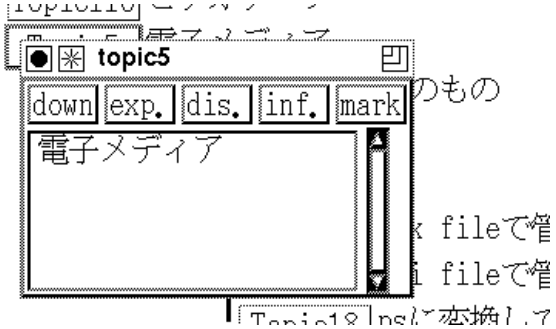


図 5.17: Topic window

以下に Topic window から呼び出すことのできる表示について説明する．

Explanation リスト

ある Topic に関する Explanation を列挙するために，図 5.18のような Explanation リストが実装されている．これによって，Topic に関する文書化された説明をブラウザから得ることができる．図 5.18の例では，topic17 の Topic window から，この topic に関する説明である exp4 と exp26 が列挙し，特に exp4 の全体を表示している．

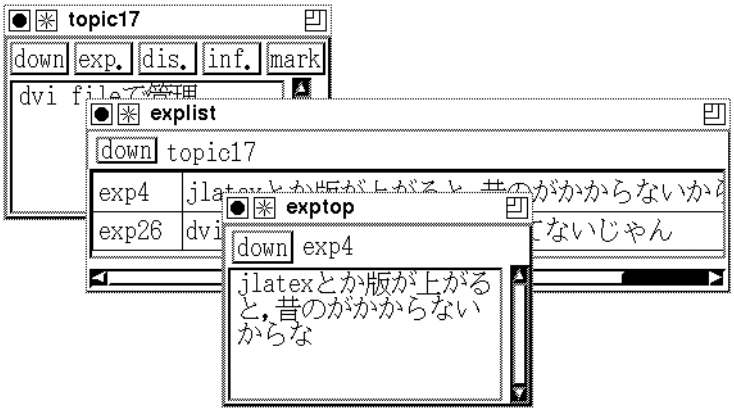


図 5.18: Explanation リスト

Discussion リスト

ある Topic に関する Discussion を列挙するために，図 5.19のような Discussion リストが実装されている．これによって，Topic が議論された部分をブラウザから得ることができる．図 5.19の例では，topic124 に関する議論が，会議 1 の 16:27:26 から 16:28:32 と，16:45:47 から 16:50:23 に行なわれていることが表示され，特に後者の議論の action viewer をよびだしている．また，会議 1 の 16:27:26 から 16:28:32 の議論では，Topic123, 124, 125, 18 が提案されていることと，16:45:47-16:50:23 の議論では，Topic123 が提案され，Topic124, 129 が否定されていることが，(O)，(X) の文字で示されている．決定の場合は，(F) の文字で示される．

図 5.19: Discussion リスト

influence リスト

ある Topic の変更などによって影響を受ける Topic のリストを表示する図 5.20のような influence リストが実装されている．これによって，ある Topic を変更しようとした場合に，影響の及ぶ範囲をブラウザから確認することができる．図 5.20の例では，topic150 によって影響を受ける Topic150(自分自身), 151, 42, 152, 180, 181 の 6 個の Topic が列挙されている．

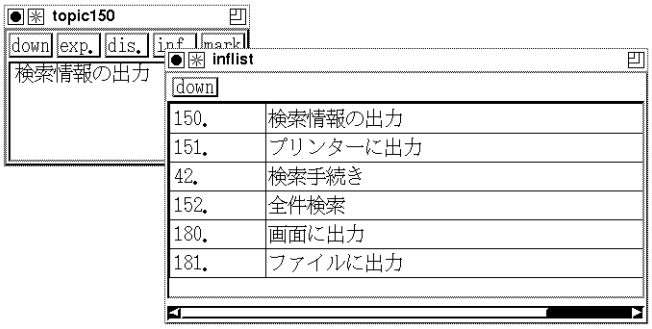


図 5.20: influence リスト

5.4 システムのプログラム構成

本プロトタイプシステムは，計算機ネットワークに接続された複数の計算機上で稼働する複数のプログラムから構成されている．図 5.21に，その構成を示す．システムはサーバー・クライアントモデルを用いた通信によって個々のプログラムがデータを交換しながら稼働する．

図に示すように，システムの一般の利用者は以下のプログラムを起動する.

音声サーバー, ユーザーインタフェース

これらのプログラムによって画像や音声のデータをハイパー議事録システムを通して記録検索できる．

また，これらに加えて操作者は以下に示すプログラムも起動する．

データベースサーバー, ビデオサーバー

以下, それぞれのプログラムの概要を述べる．

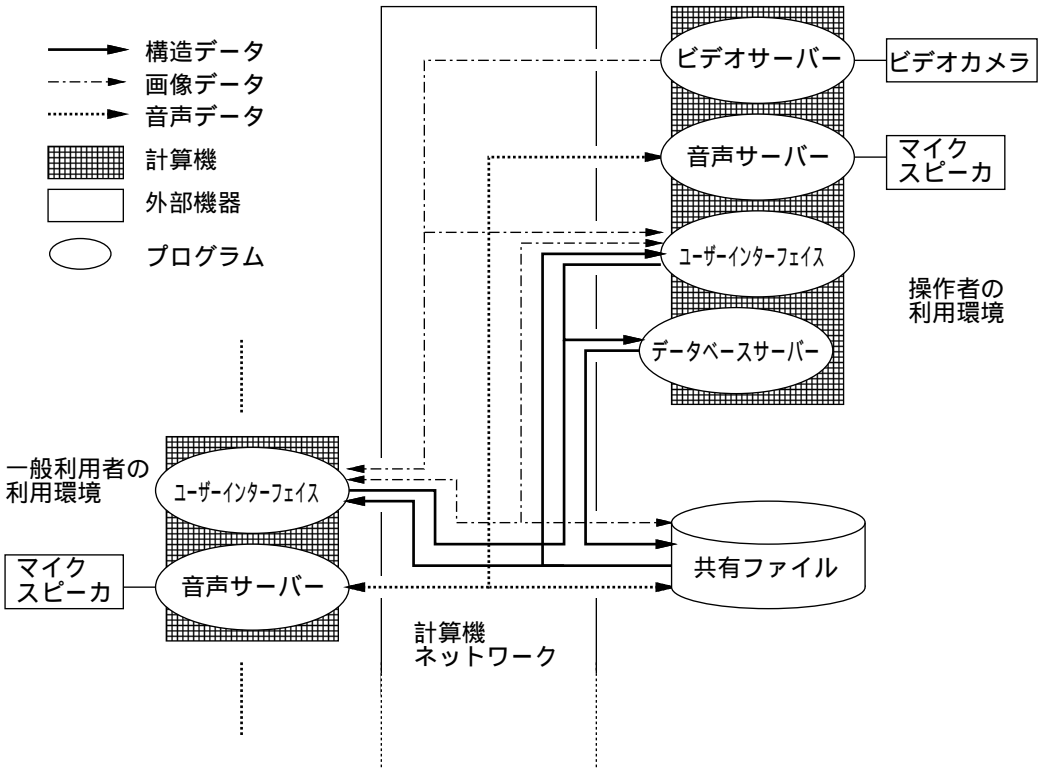


図 5.21: プログラム間の関係

5.4.1 データベースサーバー (DBSserver)

会議履歴や議事録や仕様書などの文書などのデータ構造を管理・検索するデータベースである．図 5.21に示すように，画像や音声などのデータは，他のプログラムが直接に保存し，データベースサーバーでは，個々のデータの識別子と関係を管理する．作業者共通の生産物を作成するため，生産物の構造のデータの書き込みに関して，排他制御を行なっているが，読み出しに対しては個々のユーザーインタフェースが自由に行なうことができるような構成になっている．ハイパー議事録システムを使用するときには，必ず 1 つ起動させなければならない. 図 5.21 中では操作者の計算機上で稼働しているように記述されているが，ネットワーク上の任意の計算機で動作させることができる．

起動方法は,

```
DBSserver [-option] dir
```

で, option として-v を指定することで，ユーザーインタフェースなどの各クライアントとのデータ通信情報や, エラーなどのシステム管理用の情報を表示することができる. dir は支援する会議のデータを保管する共有ファイルの位置を指す. 個人モードでの作業なども同様に共有ファイルを指定して作業する. データベースサーバーの終了方法は,

Ctrl-C のキーを押すか，または DBSserver -k *hostname*

hostname はデータベースサーバーが起動している計算機名である. データベースを終了すると自動的にすべてのクライアントは終了される. クライアントとは，ユーザーインタフェース, ビデオサーバーであり，データベースサーバーは, 任意数のユーザーインタフェース, ビデオサーバーを管理することができる．また, クライアントはデータベースサーバーが動作している間の任意の時刻に，データベースサーバーとの接続や切断を行なうことができる．これによって, 個々の作業者が会議の途中参加や途中退出が可能となっている．

5.4.2 音声サーバー (GSSserver)

音声の再生/録音用のサーバーであり，実際の音声データを共有ファイルに記録する作業も行なう．後述のユーザーインタフェースを起動させる計算機上には，必ず稼働させる必要がある．起動方法は,

```
GSSserver [-option]
```

であり, option として-v を指定することで，音声の録音状況や, エラーなどのシステム管理用の情報を表示することができる. 音声サーバーは，

Ctrl-C のキーを押すか，または GSSserver -k *hostname*

で終了できる．*hostname* は音声サーバーが起動している計算機の名前である.

音声サーバーは, 本プロトタイプシステム専用ではなく, 汎用のプログラムとして設計されている. これによって，計算機毎に異なる音声記録再生方法を吸収している．実際の音声のデータはデータベースサーバーを介さず，自身で計算機に記録する．本プロトタイプシステムでは, 図 5.21に示すように，同一の計算機で動作するユーザーインタフェースと対にして利用する．

5.4.3 ビデオサーバー

ビデオカメラを通して図示などの画像を録画するためのもので, データベースサーバーに対してクライアントとして接続される. カメラの映像はカメラを接続している計算機上のディスプレイに表示される. ビデオサーバーが起動されていない場合, 画像の録画はできない. 起動方法は,

```
video [-option] hostname
```

であり, option を-v を指定すると，エラーなどのシステム管理用の情報の表示をする. *hostname* はデータベースサーバーが起動している計算機名である. ビデオサーバーは 1 つのデータベースサーバーに対して複数個接続することができる．図 5.21に示すように，音声サーバーと異なり，画像データの記録は個々のユーザーインタフェースが行ない，ビデオサーバーはユーザーインタフェースに対して静止画のデータを転送を行なう．

5.4.4 ユーザーインタフェース

データベースサーバーに対しての記録検索をおこなうプログラムであり，5.3 節に示したグラフィカル・ユーザーインターフェースを利用者に提供している．利用者はユーザーインタフェースを用いて会議履歴を記録したり, 各種データを参照したりする. 文書作成もユーザーインタフェースを用いて行なう. ユーザーインタフェースの起動方法は,

```
tool [-option] hostname
```

であり, option として-v を指定することで，データベースとのデータ通信情報や, エラーなどのシステム管理用の情報を表示することができる. *hostname* はデータベースサーバーが起動している計算機名である.

第 6 章

システムの運用実験と評価

ハイパー議事録システムの有効性を確認するために，第 5 章で紹介したプロトタイプシステムの運用実験を行なう．

6.1 運用実験の方針

第 4 章で紹介した方法, および, 第 5 章で紹介したプロトタイプシステムを用いることで以下のような効果が期待できる.

- 議論内容が会議の出力として生産物から欠落することの防止．
- 結論が曖昧なまま会議が終ってしまう部分の発生の防止．
- 1 つの議決が変更された時に，それと関連のある議題の再審議を行なわず矛盾が発生することの防止．

これらを運用実験から明らかにする.

また, 実際にシステムを利用することにより従来の計算機を用いない会議とは異なった要素が発生するものと考えられる. そのような要素を実際の運用を通して明らかにする.

6.2 運用実験の設定

本運用実験では以下のような作業者と作業の流れを設定して行なった. この設定はシステムを利用する操作者を除けば, 一般的なソフトウェアの開発のための会議の設定である. 会議を行なった被験者にはこれ以上の制約や指示は与えていない.

顧客: (2 名) 対象システムの要求を出し, 仕様を承認する作業者. 大学職員および大学院学生.

設計者: (1 名) 顧客の要求を分析し, 個人モードで仕様を記述する作業者. 大学院学生.

操作者: (1 名) システムを導入したがために必要になった作業を行なう作業者. 大学学生. このプロトタイプシステムの製作者の一人.

1. 会議 1: 顧客が対象システムに対する要求を口頭で提案し, 設計者が不明な点などの質問をそのつど行ない, 要求の明確化作業を行なう.

   入力 : なし
   出力 : システムに記録された会議のデータ

2. 記述作業 1: 設計者が第 4 章の方法に従って, 会議の記録の構造化をシステムで行なう.

   入力 : システムに記録された会議 1 のデータ

出力 : 構造化された会議 1 とプロダクトのデータ

3. 会議 2: 設計者が構造化した会議とプロダクトのデータを基に, 議論を進める.

   入力 : 構造化された会議とプロダクトのデータ, 会議 1 までのプロダクトの階層構造の印刷物. (付録 A中の会議 1 の構造).
   出力 : 構造化された会議 1 とプロダクトのデータ, システムに記録された会議 2 のデータ.

4. 記述作業 2: 設計者が第 4 章の方法に従って, 会議の記録の構造化をシステムで行なう. および仕様書の原型となる文書を自分が構造化したデータから作成.

   入力 : 構造化された会議 1 とプロダクトのデータ, システムに記録された会議 2 のデータ
   出力 : 構造化された会議 1,2 とプロダクトのデータ, 会議 2 までのプロダクトの階層構造の印刷物.

5. 会議 3: 仕様記述者の構造を利用し, 設計者の仕様書のレビューを行なう. 内容が問題なければ顧客は承認する.

   入力 : 構造化された会議 1,2 とプロダクトのデータ, 会議 2 までのプロダクトの階層構造の印刷物.
   出力 : 構造化された会議 1,2 とプロダクトのデータ, システムに記録された会議 3 のデータ.

6. 記述作業 3: レビューの結果を文書にまとめ, 最終的な仕様書を作成する.

   入力 : 構造化された会議 1,2 とプロダクトのデータ, システムに記録された会議 3 のデータ.
   出力 : 構造化された会議 1,2,3 とプロダクトのデータ, 会議 3 までのプロダクトの階層構造の印刷物. プロダクトの構造を基に作成した仕様書 (LaTeX[62] 形式[1]).

7. 個々の作業者が文書に承認して終了.

全ての会議と記述作業でシステムを利用可能な状態である.

なお, 本運用実験で作成することになったソフトウェアは,

「 大学の研究室での文献や計算機データの管理システム 」

である.

6.3 結果

図 6.1に実際に行なわれた会議の流れを示す. 図は上から下に会議の流れをとり, 横軸に作業者をとっている. 個々の段階の特徴は以下に示す通りである.

会議 **1:** 操作者以外は, システムの記録機能を主に利用していた. 記録機能によって, 会話内容は全てシステムに記録されているため, 2 回目以降における議論の欠落が防ぐことができた. また, 作業は普通の黒板を用いたため, 会議の最初に記述したものは当然消されてしまっているが, 画像データを会議の途中でも個々の作業者が自由に参照できるため, 役だった. 図 6.2に, この会議での利用者の典型的な画面例を示す.

記述作業 **1:** 図 6.3 に, この段階での典型的な作業画面例を示す. 設計者は, 第 4 章の方法に従って構造を構築していった. システムの記録を基に不明な点を明らかにする際に, 会議 1 中に操作者が付加した議論の区切れのメモが役立った.

設計者は構造の構築を通して, 欠けている部分などを明らかにすることができた. 作業にかかった時間は, 図 6.1に示す通り, 延べ 10 時間である.

顧客 2 名もそれぞれに, 発話記録を整理することで, 会議 1 の内容の確認や, 次回の会議で話す話題などを考えたが, 第 4 章に示されるような方法ではなく, かなり自由に会議の整理を行なった. この作業を通して, 自分の主張などが明確に伝えられていないことがわかった. 顧客の作業時間は, 図 6.1に示すように, それぞれ 2 時間, 5 時間半である.

[1] 設計者が最も慣れている文書整形システム

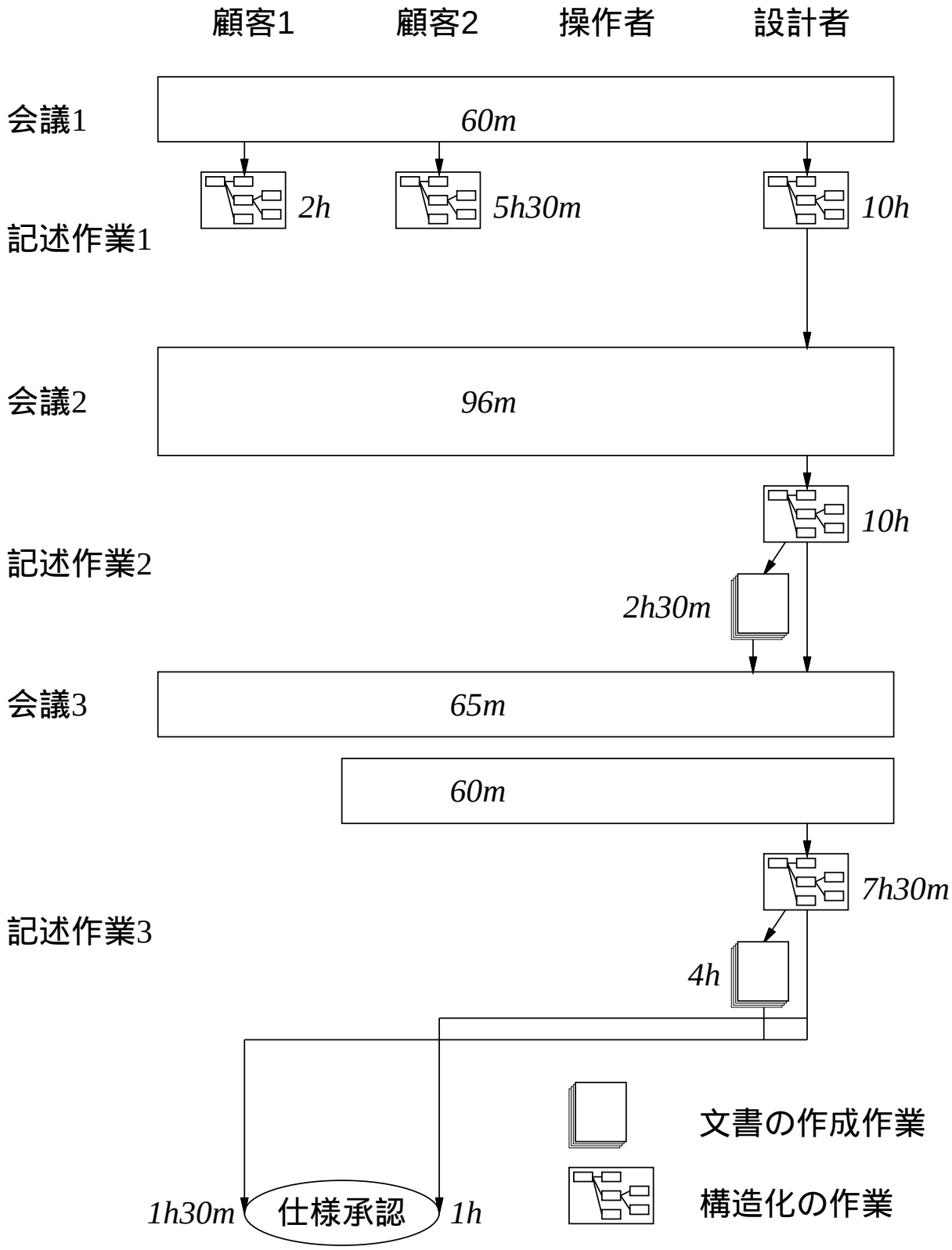


図 6.1: 実験中の会議の流れの概要

会議 **2:** この作業では, 個々の作業者が用意してきた意見を, 順番に議論する形で会議が進められ, 会議 1 での Topic と関連がある場合は, 図 6.4や図 6.5 のインタフェースを通して会議 1 のプロダクトや議論を参照することで効率的な議論を行なうことができた. そして, 設計者が次回に仕様書をまとめてくることを決め会議を終了した.

記述作業 **2** 会議 1 後の記述作業と同様に, 図 6.3に示すようなインタフェースを通して, 構造化の作業を行ない, 文書のプロトタイプ (付録 Aの会議 2 の文書) を自動生成し, そのプロトタイプの形式を整え, 不足している情報を付加することで, 仕様書の作成を行なった.

会議 **3:** およそ 1 時間の休憩を間にとり 2 回の会議を行なった. 休憩後の会議では顧客の 1 名は出席しなかった. この段階では, 設計者の作成した文書をレビューする形で会議が進められ, 適宜, 過去の議論を参照することでき, 作成した仕様を顧客に対して納得させることに役だった.

記述作業 **3** 会議 3 で修正/追加された点を考慮して, 最終的な文書を設計者が作成した. これは LaTeX 形式で記述されたが, Topic の階層構造を利用して簡単に LaTeX 形式に直すことができた. さらに, Topic に記述された文字列のみでは不足した情報を Explanation, Discussion から取り出すことで文書に情報を付加することができた.

この作業によって作成された仕様書を会議の構造データと合わせて顧客がそれぞれ確認作業を行ない, 仕様は承認された.

表 6.1に実験結果を示す. 表の第 2, 第 3 行は, 実験が行なわれた日と会議時間であり, 第 4 行から 8 行が実際にシステムに登録された会議履歴に関するデータの数, 第 9 行から 15 行がプロダクトに関するデータの数である. プロダクトに関するデータは, 前回までの会議のデータに追加を行なってデータを構築してゆくために, “延べ”, すなわちプロジェクト開始から, その会議終了までの累積の値で表示している.

“Discussion 数”, “Action 数”, “Explanation 数” の行は, それぞれの会議で作成された数を示している. “Explanation 数/Discussion 数”, すなわち 1 つの Discussion 当たりの Explanation の数は, 平均して 2.5 である. “Topic を複数持つ Explanation” の行の意味は, どの程度, 会議内の議論から明示的に Topic 間の関係を作成することができるかを示している. 例えば, このプロジェクト全体で作成された Explanation の数 376 個に対し, およそ 16% に当たる 62 個の Explanation から, Topic 間の関係を明示的に付けることができたことになる.

“延べ Topic 数” は, それぞれの会議までに作成された Topic の数を示している. 例えば, 会議 2 までの時点では, 192 個の Topic が作成されていることになる. 本論文の支援方式では, 否決された Topic も結論として取り扱うため, システムによる編集誤り以外で Topic の数が減ることはあり得ない. “最大深さ” は, Topic の木構造における根である Product の要素から, 葉である Topic の要素までの関係の数の最大値を示している. “平均 influence 数” は, Topic1 つ当たりの影響の及ぶ Topic の数の平均である. 例えば, 会議 1 の平均 influence 数は, 10.4 であるため, 会議 2 で, ある Topic が変更された場合, およそ 10 個の Topic について同時に変更すべきか否かを調べる必要があることを示している. “延べ肯定 Topic 数” は, 肯定的な提案の数を示している. およそ 84%が肯定的な提案であり, それ以外が「...しない」というような否定的な提案内容となっている. 第 4 章の方法に従い, Topic の内容は否定語は利用されていない (付録 A参照). “延べ決定 Topic 数” は, その会議までに決定事項となっている Topic 数を示している. “延べ孤立構造数” は, プロダクトの木構造と part_of 関係を持たず, 孤立してしまっている Topic もしくは, 部分構造の数を示している.

次に, 図 6.6から図 6.9にそれぞれの会議中における Discussion と Topic の分布を示す. それぞれの図は横軸に会議の進行時間を表し, 縦軸に Topic の識別番号をとったグラフ上に, 個々の Topic が議論された Discussion を線分で表現し, その分布を示している. なお, それぞれの Topic の識別番号と, その内容の対応は, 付録 A の会議 3 の構造中の対応と同一である. Topic の識別番号は, Topic が提案された順番に対応している. グラフ中で, 縦軸の値が等しい線分は, 同じ Topic に関する Discussion 群を表しており, 横軸の値が等しい線分は, 同じ Discussion で議論された Topic 群を表している.

個々の線分の端点の記号と線の種類によって, その内容を表 6.2のように区別している.

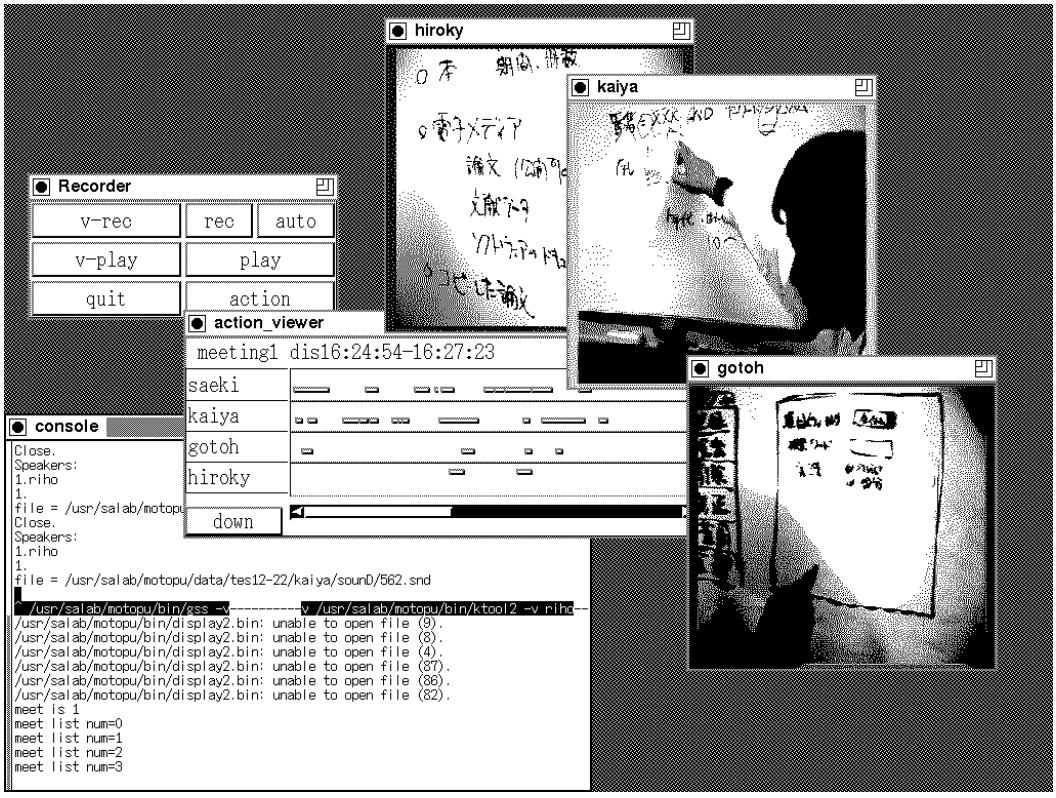


図 6.2: 記録時の画面例

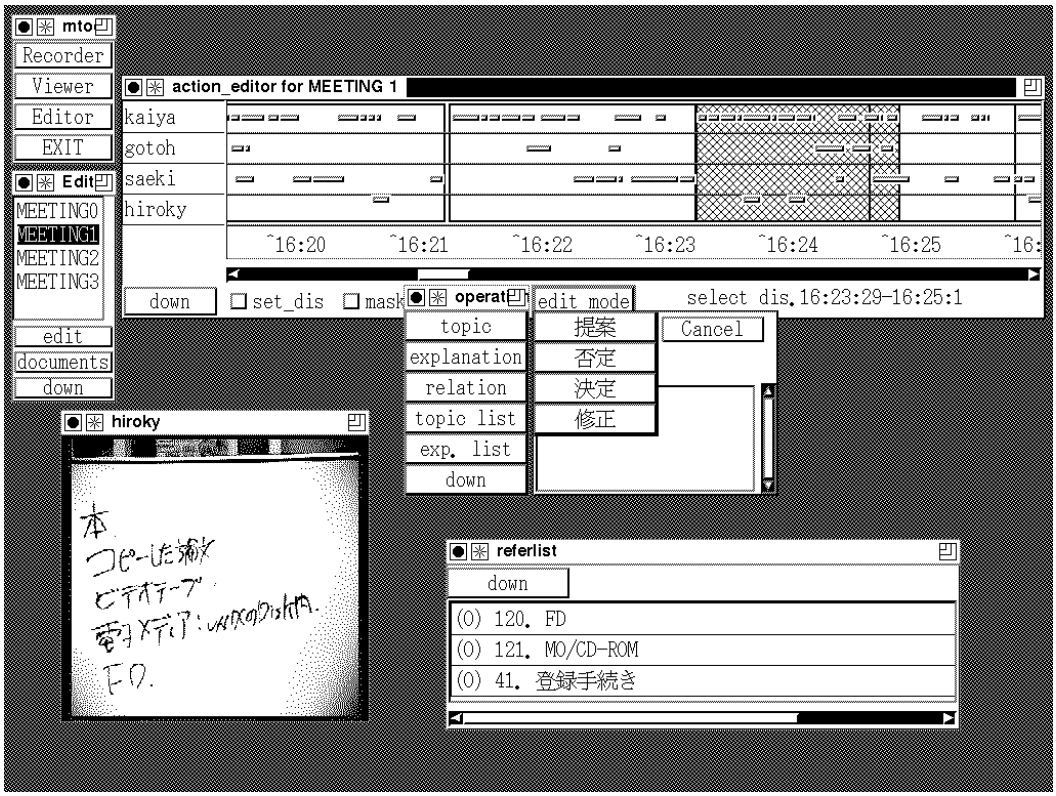


図 6.3: 編集時の画面例

図 6.4: 検索時の画面例 1

図 6.5: 検索時の画面例 2

表 6.1: 実験結果

会議番号	1	2	3 の 1	3 の 2	合計
日	12 月 22 日	1 月 6 日	1 月 18 日		
時間長 (分)	60	96	65	60	281
Discussion 数	40	43	37	31	151
Action 数	505	1141	827	841	3314
Explanation 数	166	90	63	57	376
Explanation 数/Discussion 数	4.2	2	1.7	1.8	2.5
Topic を複数持つ Explanation 数	29	13	10	10	62
延べ Topic 数	127	192	218	257	257
最大深さ	5	6	6	6	6
平均 influence 数	10.4	1.9	1.0		7.3
延べ肯定 Topic 数	102	155	178	217	217
延べ決定 Topic 数	105	182	216	257	257
延べ決定 Topic 数 (%)	82.7	94.8	99.1	100.0	100.0
延べ孤立構造数	12	4	0	0	0

表 6.2: グラフ中の線分の意味

種類	記号	内容
端点	■	決定が行なわれた
	+	同じ議論の繰り返しが行なわれた
	◆	システムによる確認が行なわれた
線	–	Topic は肯定的な提案がされている
	...	Topic は否定的な提案がされている

6.4 考察

- 議論内容が会議の出力として生産物から欠落することの防止:

  会議における発話の記録と対応付けて，生産物の部分を個人モードで作成していったため，欠落の発生はないと思われる．第 2 章で分析したプロジェクトでは，表 2.6より，およそ 3 から 6 割の事項が欠落している．しかし，図 6.1に示すように，個人モードでの構造化の作業に会議時間の 10 倍程度の時間がかかるため，個人モードでの作業者 (ここでは設計者) の負担が，かなり増大したと思われる．この部分はシステムのインタフェースの向上により，かなりの改善ができることが設計者へのインタビューから得られた．例えば，本プロトタイプのブラウザは，主に記録されたデータを参照するのみであるが，ブラウザ上で Topic 間の構造を変更したり，Topic の内容に当たる文字列を変更したりする作業が可能となれば，個人モードの作業時間をかなり短縮できるようである．

- 未決定な部分の発生防止:

  表 6.1の延べ決定 Topic 数の割合から, 最終的には全ての提案された Topic に対して決定が行なわれている. それに対して，第 2 章で分析したプロジェクトでは，表 2.7などの結果から，およそ，50 ~ 82% が決定されており，それ以外の部分は未決定のままに終っている．よって，このシステムによって，未決定な部分の発生は十分に防止できたと考えられる.

- 不明瞭な部分の発生防止:

表 6.1の延べ孤立構造数は, 最終的には, 0 になっており, 全ての提案は, 最終的に仕様全体の部分に取り込むことができた. それに対して，第 2 章で分析したプロジェクトでは，表 2.8などの結果から，およそ，7 ~ 35% が最終的に仕様全体から孤立したままであった．よって，このシステムによって，不明瞭な部分の発生は十分防止できたと考えられる.

- 効率的な議論の促進:
  図 6.6から図 6.9の議論の分布から, 1 つの Topic に関する議論の繰り返しはそれほど多くないことが分かる．実際には，1 つの Topic 当たりの平均議論回数は, 1.9 回である．議論の内容を調べると，提案程度で結論を出していない議論は，およそ 25% あるが，表 6.1 に示すように，最終的には，それら全てに対して結論が出されているので，ここでの 25%の結論のない議論による非効率な作業の影響は，回復されているといえる．
- 文書内の矛盾の発生の防止:
  会議中に 15 Topic の肯定/否定の変更があった. 図 6.6から図 6.9中に番号の付加してある線分が変更のあった議論に当たる. これらの変更時点において, influence 関係を用いて矛盾の発生の危険がある Topic を表 6.3 に示す. 表の縦方向の罫線で区切られている Topic は，同じ Discussion 内で変更があった Topic のグループである．例えば，番号 55, 66, 68 は同じ Discussion 内で変更が起こっている．表中の第 5 列の数値は，それぞれの変更が発生した Discussion において，矛盾を起こす可能性のある Topic の数である．この分析結果より，4.6 ~ 18.8% の矛盾を引き起こす可能性のある Topic を influence 関係によって，獲得することができ，仕様書内の矛盾の発生の防止になっていると思われる．

表 6.3: influence 関係の利用度

A	B	C	D	C/D(%)
58	44 54 55 56 57 59 60 61 65	9	65	13.8
59	44 54 55 56 57 58 60 61 65	9	65	13.8
55	44 54 56 57 58 59 60 61 65	13	69	18.8
66	7 8 67 68			
68	7 8 66 67			
56	54 55 57 58 59 60 61	7	77	9.1
31	11 20 24 25 26 27 28 29 30	9	78	11.5
90	3 20 21 22 86 87 88 89 91 92 93 94 95 96	14	126	11.1
35	3 12 17 18 19 32 33 34 36 37 38 39 40	13	129	10.1
27	11 20 24 25 26 28 29 30 31	9	141	6.4
105	82 83 84 85 98 102 103 104 106 107	9	150	6.0
106	82 83 84 85 98 102 103 104 105 107			
143	142 144	7	153	4.6
148	82 144 145 146 147 149			
28	2 11 20 24 25 26 27 29 30 31 69 78	12	191	6.3

ただし，表中の A ~ D は，

A: 変更があった Topic の番号

B: 変更があった Topic と influence 関係にある Topic の番号

C: 変更があった Topic と influence 関係にある Topic の総数

D: A の Topic に変更があった議論での Topic の総数

- 第 2 章で行なった分析は, ビデオテープなどを利用して, このシステムを利用した作業と類似した作業を行なった. よって, 分析作業に費やした時間から, システムを利用せずこの方法で仕様書を作成する手間を予想する. その場合, 記録を作成する作業は会議の時間の約 50 倍かかり, 記録を構造化するために約 30 倍の時間がかかると推定できる.

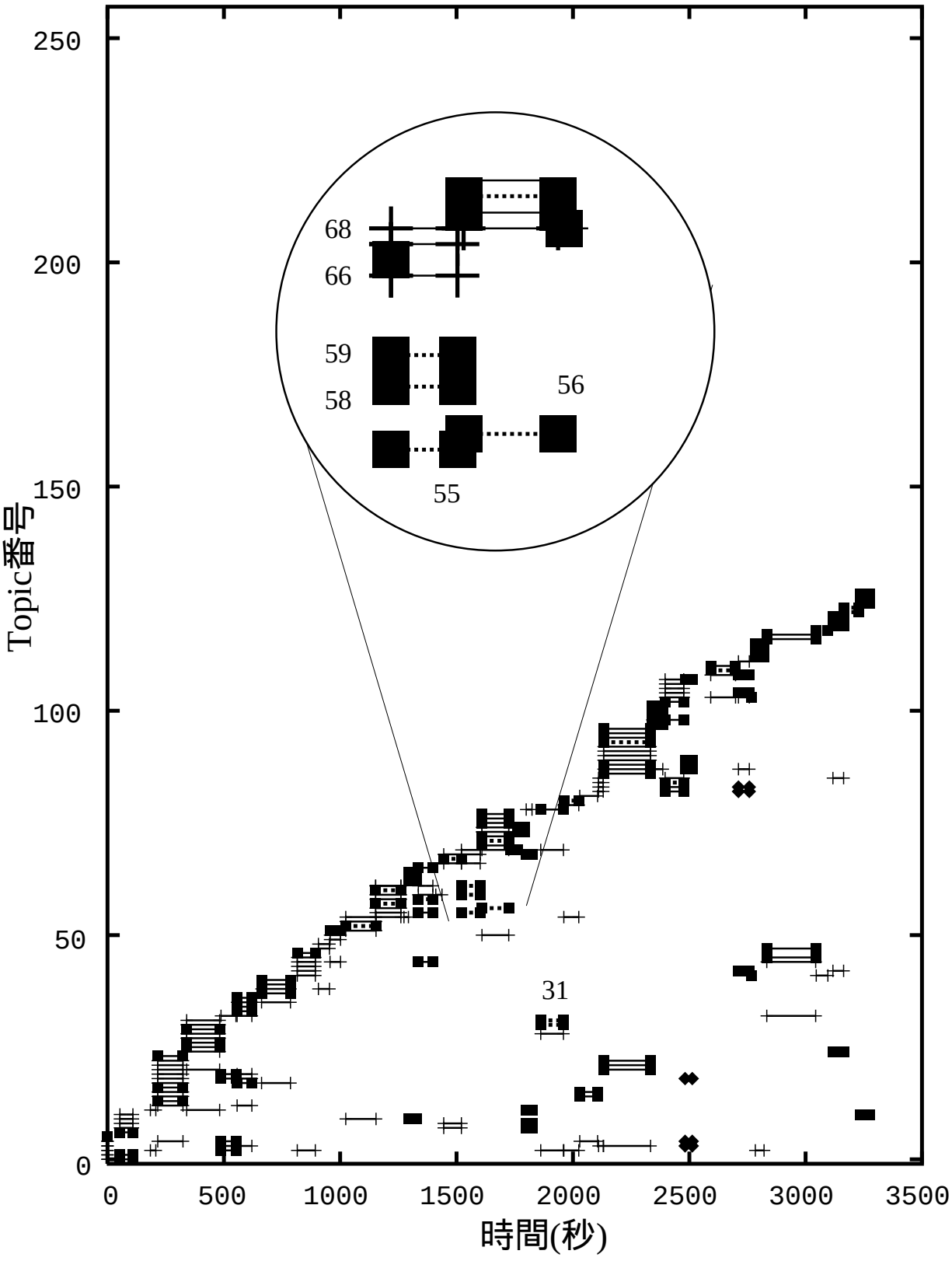


図 6.6: 会議 1 中の Discussion と Topic の分布

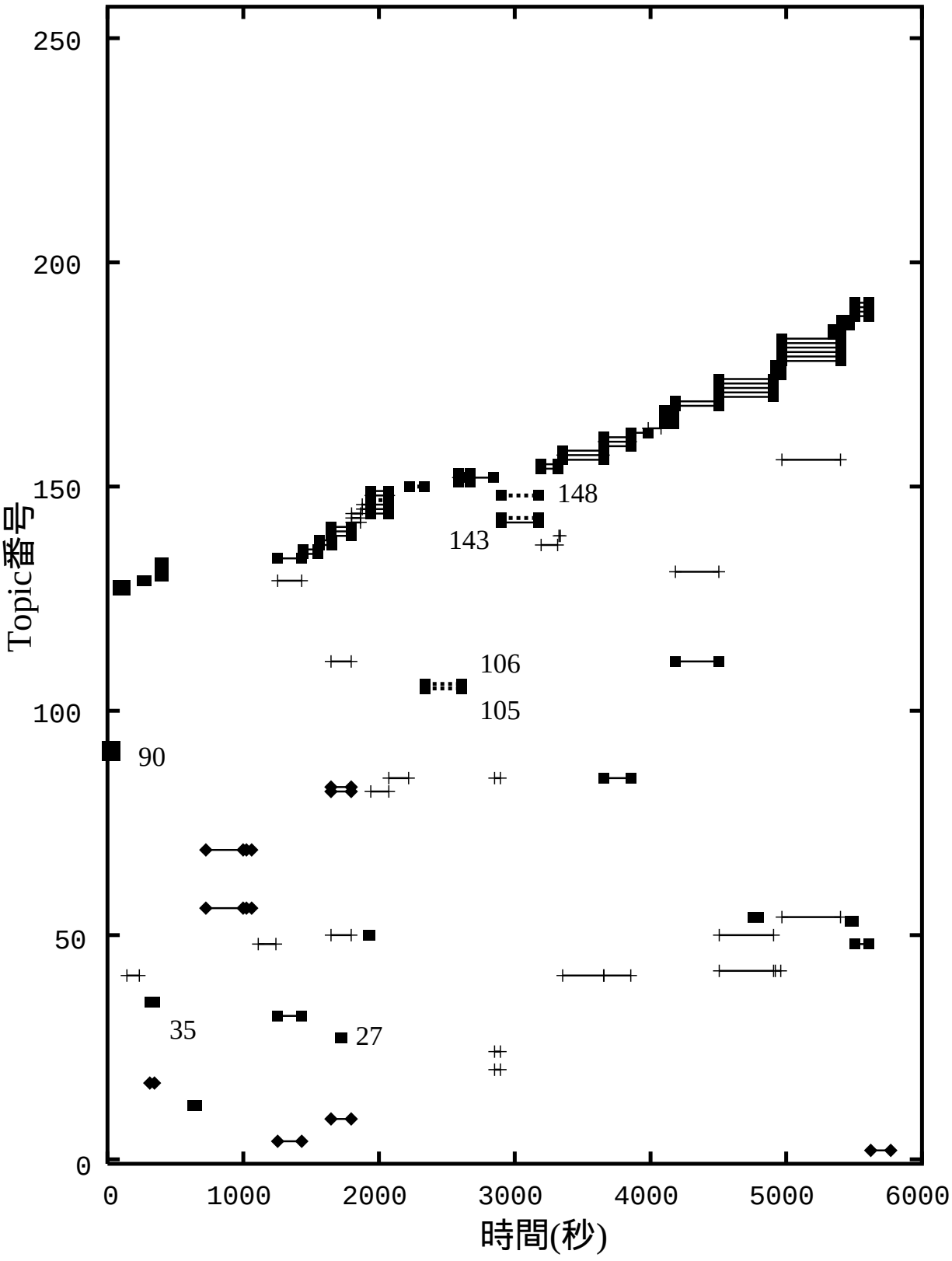


図 6.7: 会議 2 中の Discussion と Topic の分布

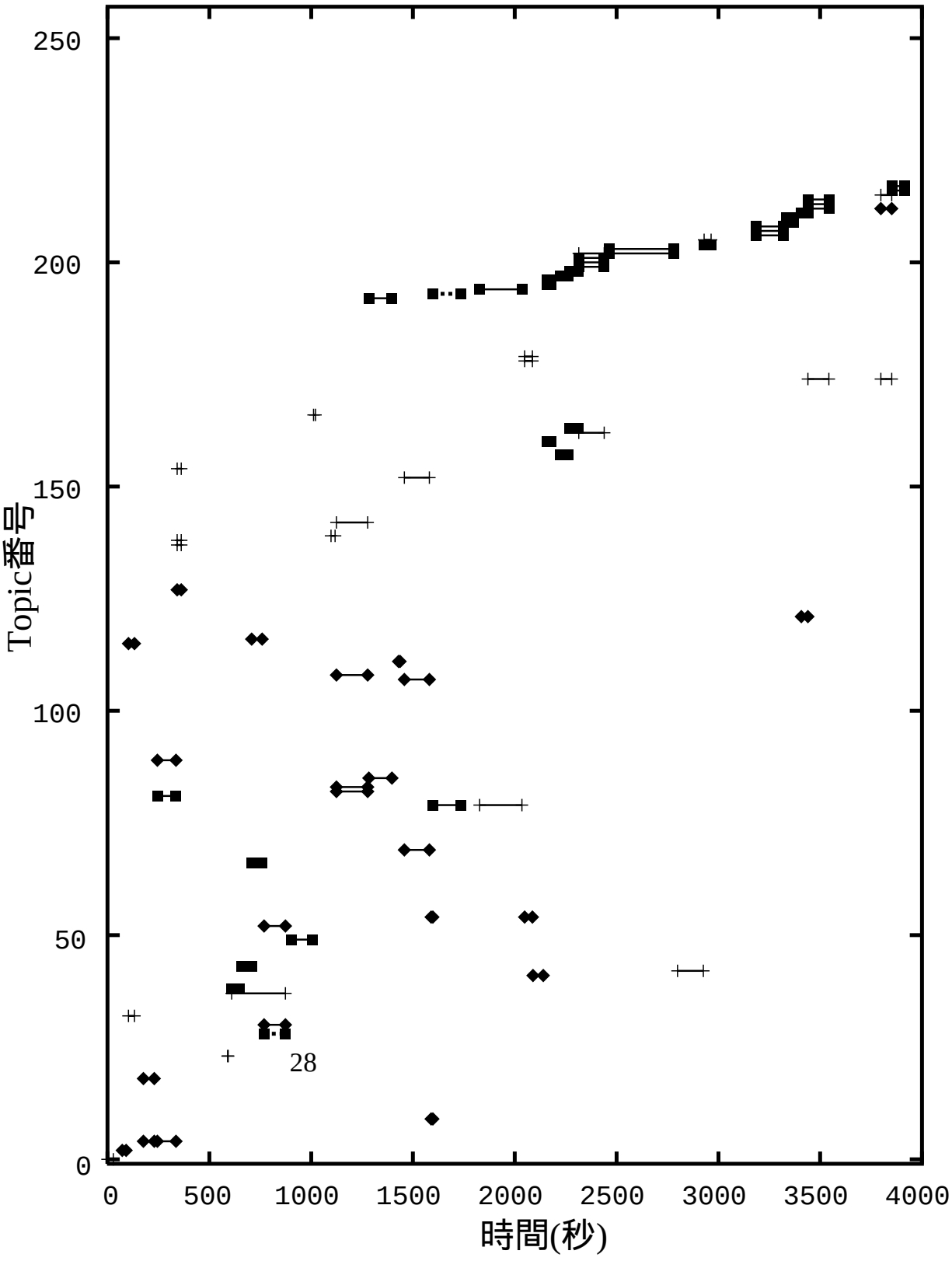


図 6.8: 会議 3 その 1 中の Discussion と Topic の分布

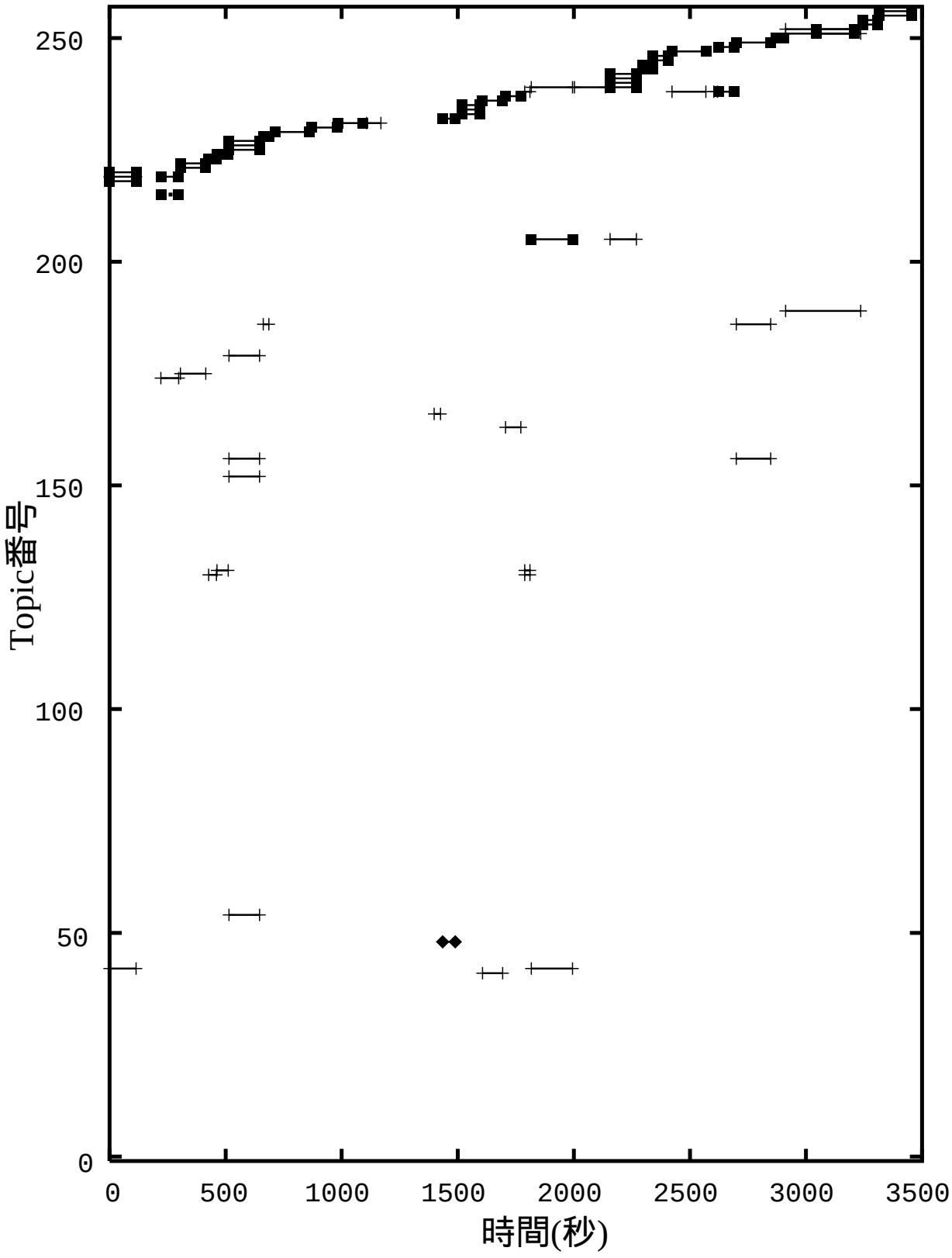


図 6.9: 会議 3 その 2 中の Discussion と Topic の分布

第 7 章

おわりに

7.1 まとめ

本研究では, 実際の会議中の作業のプロトコル分析を通して, 従来の会議における問題点を明らかにした. そして, その分析結果を基に，ハイパー議事録システムを構築し，適用実験を通してシステムの有効性を確認した．本システムは，複数の作業者が複数回の会議を通して共通の生産物を作成する作業を支援する計算機システムである．

分析は，会議で行なわれる発話を中心とした行為と, 議事録や仕様書などの生産物との関係に注目し，その対応をとることによって, 従来の会議での問題点を明らかにすることを目標として行なった．実際に得られた問題点は以下の通りである.

- 議論内容が生産された議事録や仕様書などの文書から欠落する.
- 議論中で仕様全体から孤立するような部分が文書から欠落しやすい.
- 同じ議論を繰り返したり，結論のない議論を行なうなどの，非効率的な議論がある.
- 文書内の一部を変更した場合に, その影響を受けて同様に変更を行なわなければならない部分がある. それらの部分は会議中において時間的に連続して議論されている場合が多い.

また, それ以外に以下のような特徴が得られた.

- 1 つの会議中でも，作成対象となる文書は適宜入れ替わる．例えば，ソフトウェアの仕様作成を行なう会議などでは，発注文書，基本調査書，要求 (外部) 仕様書，設計 (内部) 仕様書，スケジュール表など注目する文書の種類が適宜入れ替わり，1 つの会議で 1 種類の文書のみに注目しているわけではない．本分析では，同一の文書に注目している会議内の部分をステップと呼び，それぞれのステップでは，注目している文書の種類によって，作業者の発話のパターンが異なることが分かった．

これらの分析の結果を踏まえて，ハイパー議事録システムのデータ構造を考案し, その構造を構築するための方法を考案し, その方法を実行するためのシステムのプロトタイプを計算機ネットワーク上に作成した. 本システムの特徴は以下の通りである．

- 会議の履歴を発話情報まで含めて記録し，コンピュータ内に蓄積する．
- 既に行なわれた会議の履歴を次の会議で活用できるように構造化する．
- 構造化された会議の履歴を会議中も含めて随時検索・参照するためのグラフィカル・ユーザーインターフェースを提供する．

本論文で提案したシステムは，システムの運用を補助する操作者 (オペレータ) 以外は，計算機を利用しない従来の会議と同様に作業するだけでも良く, 実作業への適用性が高い. 設計した会議を記録するためのデータ構造も, 理論の上から構築された非現実的な構造ではなく, 作業を支援するという観点から, 必要な部分のみに注目している極めて現実的な構造となっている.

作成したプロトタイプシステムを用いた運用実験から，本システムが実作業の分析で得た問題点の多くを解決していることが分かった．

7.2 今後の展望

本研究では, 複数の作業者が複数回の会議を通して共通の生産物を作成して行く作業を支援するハイパー議事録システムを構築した．

第 4 章における利用の方法でも示したように，実際のハイパー議事録の構築のほとんどは会議と会議の間の個人モードの作業として行なわなければならない．しかし，この作業も会議中にリアルタイムに処理することで，より効率的な会議を行なうことが可能だと思われる．このような作業方式をとる場合, 会議内で, より多くの “普通でない” 作業を, “より多くの” 作業者に強いる可能性が高いため，本研究の指針の 1 つである実際の作業への適用し易さという基準に反する恐れがある．会議中にリアルタイムでハイパー議事録を構築する方法を確立するためには，プロトタイプシステムの運用などを通して，ユーザーインタフェースなどの改良と組み合わせて議論してゆく必要がある．

本研究は, 対面型の作業を前提としたが, 高速計算機ネットワークを利用した非対面型の共同作業などにも対応する必要があると思われる. 電子メールなどを利用した遠隔分散-非同期型のシステム [63] との接続により，より広範囲な作業形態への適用が可能となる．

本システムは，個々の適用分野特有のドメイン知識 [18],[22] や，投票機能などに代表される意志決定支援法 [64], [17] や，発想支援法 [23],[24] などを取り込んでいない．これらの方法は，落丁や矛盾のない生産物の作成を支援する本システムの目的と相矛盾するものではなく，本システムと既存のドメイン知識，意志決定支援法，発想支援法と組み合わせることによって，いっそうの効果が期待できると思われる．

謝辞

本研究を行なうにあたり終始変わらぬ御指導を賜わりました佐伯元司助教授には心から深く感謝申し上げます. また本研究に関与された佐伯研究室の諸氏にも御礼を申し上げます.

共同研究をさせていただいた産能短期大学の本間学氏に深く感謝致します. そして, 実験に参加していただいた産能短期大学の学生の皆様に深く感謝いたします.

実務の観点から貴重な助言をしていただいたキヤノン販売株式会社の岸本英樹氏に深く感謝いたします.

本研究を進めるにあたり, 大森晃博士 (現 東京理科大学工学部経営工学科助教授) をはじめとする富士通株式会社 国際情報社会科学研究所 (現 株式会社 富士通研究所 情報社会科学研究所) での社会科学アプローチセミナーを通して貴重な議論及び助言を戴いた同研究所 ユーザー指向ソフトウェアプロセスグループのメンバーに深く感謝致します.

古宮誠一氏をはじめとする情報処理振興事業協会 (IPA) ソフトウェア設計過程分析とモデル化の調査研究ワーキング委員会の委員の皆様に深く感謝致します.

本位田真一博士をはじめとする情報処理振興事業協会 (IPA) 新ソフトウェア構造化モデルの研究開発事業プロジェクトの皆様に深く感謝致します.

システムのプロトタイプの開発に数多くの助言をしていただいた通産省工業技術院 電子技術総合研究所 (ETL) の伊藤克亘博士に深く感謝いたします.

参考文献

[1] I. Greif. *Computer-Supported Cooperative Work*. Morgan Kaufmann Publishers, 1988.

[2] ACM SIGCHI & SIGOIS. *CSCW86, Proceedings of the Conference on Computer-Suported Cooperative Work*, 1986.

[3] ACM SIGCHI & SIGOIS. *CSCW88, Proceedings of the Conference on Computer-Suported Cooperative Work*, Sep. 1988.

[4] ACM SIGCHI & SIGOIS. *CSCW'90, Proceedings of the Conference on Computer-Suported Cooperative Work*, Oct. 1990.

[5] L.Bannon, M.Robinson, and K.Schmidt, editors. *ECSCW91, Proceedings of The 2nd European Conference on Computer Supported Cooperative Work*, Sep. 1991.

[6] 石井裕. グループウェア技術の研究動向. 情報処理, Vol. 30, No. 12, pp. 1502–1508, Dec. 1989.

[7] Terry Winograd. A language perspective on the design of coopearative work. In *CSCW'86 Proceedings*, Dec. 1986.

[8] Terry Winograd. A language/action perspective on the design of cooperative work. *Journal Of Human-Computer Interaction*, Vol. 3, , 1987.

[9] Terry Winograd. Where the action is. *BYTE*, Vol. 13, No. 13, Dec. 1988.

[10] Terry Winograd and Fernand Flores. *Understanding Computers and Cognition*. Ablex Publishing Corporation, Norwood, N.J., 1986.

[11] T.W. Malone, K.R. Grant, K.Y. Lai, R. Rao, and D. Rosenblitt. Semistructured messages are surprisingly usefull for computer-supported coordination. *ACM Transaction on Office Information System*, Vol. 5, No. 2, pp. 115–131, Apr. 1987.

[12] Kum yew Lai, Thomas W. Malone, and Keh chiang Yu. Object Lens: A Spreadsheet for Cooerative Work. *ACM Transaction on Office Information System*, Vol. 6, No. 4, pp. 332–353, Oct. 1988.

[13] Uta Pankoke-Babaz. *Computer Based Group Communication, the AMIGO activity model*. Ellice Horwood Limited, England, 1989.

[14] Hiroshi Ishii. TeamWorkstation: Towards a Seamless Shared Workspace. In *CSCW90*, pp. 13–26, Oct. 1990.

[15] C.A.Ellis, S.J.Gibbs, and G.L.Rein. Design and Use of Group Editor. *MCC Technical Report STP*, Vol. 263, No. 88, 1990. about GROVE.

[16] Mark Stefik, Gregg Foster, Daiel G. Bobraw, Kenneth kahn, Stan Lanning, and Lucy Suchman. BEYOND THE CHALKBOAD: computer supported for collaboration and problem solving in meetings. *Communications of the ACM*, Vol. 30, No. 1, pp. 32–47, Jan. 1987.

[17] Peter Cook, Clarence Ellis, Mike Graf, Gail Rein, and Tom Smith. Project Nick: Meetings Augmentation and Analysis. *ACM Transaction on Office Information System*, Vol. 5, No. 2, Apr. 1987.

[18] M.A. Jackson. *System Development*. Prentice-Hall, 1983.

[19] T. Demarco. *Structured Analysis and System Specification*. Yourdon Press, 1978.

[20] Grady Booch. Object-oriented development. *IEEE transactions on software engineering*, Vol. SE-12, No. 2, pp. 211–221, Feb. 1986.

[21] Sally Shlaer and Stephan J. Mellor. *Object-oriented Systems Analysis*. Prentice-Hall, 1988.

[22] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.

[23] 川喜田二郎. 発想法. 中公新書, 1967.

[24] 小山雅庸, 河合和久, 大岩元. カード操作ツール KJ エディタの実現と評価. コンピュータソフトウェア, Vol. 9, No. 5, pp. 38–53, 1992.

[25] 垂水浩幸. グループウェアのソフトウェア開発への応用. 情報処理, Vol. 33, No. 1, pp. 22–31, Jan. 1992.

[26] P.Moran Thomas, editor. *HUMAN-COMPUTER INTERACTION*, Vol. 6 No. 3 & 4. LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS, Hillsdale, New Jersey, 1991.

[27] Barry W. Boehm. *Software Engineering Economics*. Englewood Cliffs,N.J.:Prentice-Hall, 1981.

[28] B.Boehm. Industrial software metrics top ten list. *IEEE Software*, Sep. 1987.

[29] Bill Curtis. Implication from empirical studies of the software design process. In *Proceedings of an International Conference organized by the IPSJ to communicate the 30th Anniversary*, 1990.

[30] 岸本三江. ソフトウェア生産プロセスにおけるインターラクションの分析. 情報処理学会第 35 回全国大会論文集, pp. 1141–1142, 1987 年後期.

[31] Gary M. Olson, Judith S. Olson, Mark R. Carter, and Marianne Storrosten. Small group design meetings: An analysis of collaboration. *Human-Compter Interaction*, Vol. 7, No. 3, pp. pp.347–374, 1992.

[32] H.Rittel and W.Kunz. Issues as elements of information systems. Working paper #131, Institut fur Gundlagen der Planung I.A. University of Stuttgart.

[33] H. Rittel and M.Webber. Dilemmas in a general theory of planning. *Policy Sciences*, Vol. 4, , 1973.

[34] C. Potts and G. Bruns. Recording the Reasons for Design Decisions. In *10th International Conference on Software Engineering*, pp. 418–427, 1988.

[35] J. Lee. Extending the Potts and Bruns Model for Recording Design Rationale. In *13th International Conference on Software Engineering*, pp. 114–125, 1991.

[36] 吉府研治, 垂水浩幸, 西田幸雄. 設計理由記録モデルの比較. 情報処理学会第 44 回全国大会論文集, pp. 5–351, Apr. 1992.

[37] Eiji Kuwana and James D. Herbsleb. Representing knowledge in requirements engineering: An empirical study of what software engineering need to know. In *Proceedings of the IEEE International Symposium On Requirments Engineering*, pp. 273–276, Jan. 1993.

[38] MacLean,A., Young,R.M., Bellotti,V.M.E., and Moran,T.P. Questions,options,and criteria: Elements of design space analysis. *HUMAN-COMPUTER INTERACTION*, Vol. 6, No. 3 & 4, pp. 201–250, 1991.

[39] 尾上裕子, 桑名栄二. 設計者間のコミュニケーション構造モデルの一考察. 情報処理学会グループウェア工学研究会, Vol. 3, No. 1, Dec. 1992.

[40] Yuko Onoe and Eiji Kuwana. Communication analysis of argument structure-based collaborative design. In *Joint Conference on Software Engineering '93*, pp. 159–166. IPSJ and KISS, Nov. 1993.

[41] 中島毅, 田村直樹, 藤岡卓, 上原憲二, 高野彰. PPK 法 : ソフトウェア設計プロセスの記録と分析の手法. ソフトウェア工学研究会, Vol. 67, No. 2, Jul. 1989.

[42] 田村直樹, 中島毅, 藤岡卓, 上原憲二, 高野彰. ハイパテキストを用いた設計プロセス支援ツールの試作. ソフトウェア工学研究会, Vol. 68, No. 7, Sep. 1989.

[43] 安達久人, 竹中豊文, 浜田雅樹. 設計プロセスを利用した修正支援法について. 情報処理学会ソフトウェア工学研究会, Vol. 80, No. 16, pp. 127–134, Jul. 1991.

[44] 浜田雅樹, 安達久人, 竹中豊文. 設計プロセスの蓄積・利用による設計支援法について. 情報処理学会ソフトウェア工学研究会, Vol. 80, No. 17, pp. 127–134, Jul. 1991.

[45] 落水浩一郎ほか. ソフトウェア開発における協調支援環境 Vela. 情報処理学会第 41 回全国大会 5, pp. 149–162, Sep. 1990.

[46] K.A.Ericsson and H.A.Simon. Verbal reports as data. *The Psychological Review*, Vol. 87, No. 3, pp. 215–251, 1980.

[47] J.R. Searle. *Speech Acts*. Cambridge University Press, 1969.

[48] Jeff Conklin and Michael L. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. In *CSCW'86 Proceedings*, Dec. 1986.

[49] G. L. Rein and C. A. Ellis. rIBIS: a real-time group hypertext system. *International Journal of Man-Machine Studies*, Vol. 34, No. 3, pp. 349–368, Feb. 1991.

[50] Yakemovic,KCB and Conklin,J. Report on a development project use of an issue-based information system. In *Proceedings of the Conference on Computer Suported Cooperative Work*, pp. 105–118, New York, 1990. ACM. about itISIS.

[51] Jintae Lee. SIBYL: A Tool for Managing Group Decision Rationale. In *CSCW'90 proceedings*, pp. 79–92, Oct. 1990.

[52] 岸本三江, 西田正吾, 後藤柳一郎. ソフトウェア意思伝達支援ツール COMICS(1),(2). 情報処理学会第 37 回全国大会論文集, pp. 857–860, 1988 年後期.

[53] 中谷三江. 劇場モデルに基づいたソフトウェア意図伝達支援ツール COMICS. 情報処理学会論文誌, Vol. 31, No. 1, pp. 124–135, Jan. 1990.

[54] Naohiko Takeda, Akichika Shiomi, Kazuhisa Kawai, and Hajime Ohiwa. Requirement Analysis by the KJ editor. In *IEEE International Symposium On Requirements Engineering*, pp. 98–101. IEEE, IEEE Computer Society Press, Jan. 1993.

[55] Naohiko Takeda, Akichika Shiomi, Kazuhisa Kawai, and Hajime Ohiwa. Preliminary experiment with a distributed and networking card-handling tool named kj-editor. In *Human-Computer Interaction: Software and Hardware Interfaces Proceedings of the 5th International Conference on Human-Computer Interaction*. ELSEVIER, Aug. 1993.

[56] 杉山公造. 収束的思考支援ツールの研究開発動向-KJ 法を参考とした支援を中心にして-. 人工知能学会誌, Vol. 8, No. 5, pp. 32–38, Sep. 1993.

[57] 河合和久, 塩見彰睦, 竹田尚彦, 大岩元. 協調作業支援機能を持ったカード操作ツール KJ エディタの評価実験. 人工知能学会誌, Vol. 8, No. 5, pp. 47–56, Sep. 1993.

[58] Charles Goodwin and John Heritage. Conversation analysis. *Annual Reviews Anthropoly*, Vol. 19, pp. 283–307, 1990.

[59] Stephen C. Levinson. *Pragmatics*, chapter 6, pp. 284–370. Cambridge Univ. Press, 1983. about Conversation Analysis.

[60] E. Soloway and K.Ehrich. Empirical studies of programming knowledge. *IEEE Trans. on Soft. Eng.*, Vol. 10, No. 10, pp. 595–609, 1984.

[61] C. Potts. A Generic Model for Representing Design Methods. In *11th International Conference on Software Engineering*, pp. 217–226, May 1989.

[62] Leslie Lamport. *A document Prepartion System* LaTeX *user's guide & reference manual*. addison-wesley publishing company, 1985.

[63] Motoshi Saeki, Kazuhisa Iguchi, and Masanori Shinohara. Supporting tool for cooperative specification processes. In *SEKE'93: The 5th International Conference on Software Engineering and Knowledge Engineering*, pp. 351–354, Jun. 1993.

[64] Kenneth L. Kraemer and John Leslie King. Computer-based systems for cooperative work and group decision making. *ACM Computing Surveys*, Vol. 20, No. 2, pp. 115–146, Jun. 1988.

付録 A

運用実験で作成されたプロトタイプ構造

ツールによって作成された会議の構造を提示する. 1 行が 1 つの Topic に対応しており, それぞれの列の意味は,

階層の深さ 肯定か否定か 決定か否か Topic 番号 Topic の内容

であり, 表示形式は,

階層の深さ : . の数で表示. ただし, - の場合は孤立 Topic.
肯定か否定か : O: 肯定, X: 否定.
決定か否か : F: 決定. -: 未決定.
Topic 番号 : 重複しない数字.
Topic の内容 : 文字列.

である.

A.1 会議 1 の構造

. O F 0 要求
.. O F 1 本の管理
... O F 5 論文, 電子メディアも含めて, 研究室のものを管理する.
.. O F 10 検索
... O F 124 誰が借りているか
... O F 125 論文の一覧
... O F 126 本の一覧
.. X F 123 知識ベース的なもの
- O F 4 電子メディア
.. O F 6 電子的な論文
... O F 13 TeX
.... O F 14 卒論
.... O F 15 修論
.... O F 16 D 論
... O F 18 文献データ
.... O - 12 BIBTeX
..... O F 33 自動
..... O F 34 力まかせ
.... O F 36 文献一覧表

.. O - 81 project 毎のソフトウェア
... O F 19 ソフトウェアのドキュメント
... O F 87 ソフトウェア本体
... O F 88 実験データ
.... O F 98 不定型データ
..... O F 99 フォーマット
..... O F 100 テープ起こし
..... O F 101 画像データ
... O F 89 ハードコピー
... O F 94 分野毎
.... O F 97 コマンド
.... O F 96 卒論などは，今までは年度毎に管理されていたが，分野毎の方が良い．
- O F 9 所有者
.. O F 51 所有権
.. O F 62 個人
.. O F 63 研究室
- O F 82 タイトル
.. O F 102 ツール名
- O F 83 著者
- O F 24 分野
.. O F 20 project
... O F 21 METHODBASE
... O F 22 MOTOPU
... O F 25 FORMALMETHOD
... O F 26 グループウェア
... O F 86 LOTOS
... O F 95 カーデット
... O F 103 project 名
. O - 27 図書館の分類
. O F 47 対象物
.. O F 2 本
... O F 7 図書館登録の本
... O F 8 個人の本
... O F 78 学会誌
... O F 115 マニュアル
.. O F 3 論文
.. O F 11 分類
... O F 29 内容
.. O F 17 公開
... O F 23 対象とするのは，対象物のうちの，公開しても差し支えないものである．
- O - 32 コピーした論文
. O - 35 非公開
. O F 37 登録
.. O - 28 貸しだし制限
... X F 30 期間
... X F 31 冊数

.. O - 38 データベース
... X - 53 管理者
... O F 40 データを一箇所にまとめて保存する.
.. O - 43 公開方法
... O F 44 個人のもの
.... O F 39 個人の所有物は, 各自が公開してもいいものだけを任意に登録する.
... O F 45 共有のもの
.... O F 46 共有のものについては到着次第登録する.
.. X F 52 公開規定
... X F 55 プロテクション
.... X F 57 特定の人のみ公開
.... X F 58 本人の許可
.... X F 59 すべての人に公開
.... X F 60 すべての人に非公開
.... X F 61 特定の人に非公開
.. X - 66 本の置き場所
... X F 67 自分の本棚
... X F 68 共有の本棚
.. O F 116 複数ある場合
... O F 117 論文のコピーなどで重要そうなものは, 共有用に一部作っておくと便利.
. O F 41 登録手続き
. O F 42 検索手続き
. O - 48 インタフェース
. O - 49 データ構造
.. O - 50 すべてに共通な情報
... O F 108 何年度か?
.... O F 110 論文などの場合, 何年度のものか.
- O - 54 借用人
.. O F 64 研究室のメンバー
.. O F 72 一般民間人
... O F 65 図書館の本は研究室外の人にも貸すことがある.
.. O - 79 オンライン
... X F 80 オンライン（ディスク内のもの）についての借用人.
. X F 56 借用条件
- O F 69 コメント
.. O F 70 借りる時
... O F 76 本などを借りるときは, その本の「備考」の欄の貸し出し条件を見てから借りる.
.. X F 71 守らない場合
... O F 77 備考欄の貸し出し条件が守られなくても, 計算機システムでは対処しようがないので, 守ってもらうしかない.
.. O F 73 持ちだし禁
.. O F 74 コピー禁
.. O F 75 貸し出し制限（条件）などを明記する.
. X F 84 検索法
- O - 85 キーワード
. O - 90 お約束の文章

. O - 91 謝辞
. O - 92 論文の部品
. X F 93 年度毎
. O - 105 ローダー
. O - 106 リーダー
- O F 107 project 特有データ
.. O F 104 abstract のようなもの
. X F 109 キー
- O - 111 場所
. X F 112 記録するデータ
. X F 113 日付
. X F 114 名前
. X F 118 バーコード
- O F 121 一覧を出す
.. O F 119 50 音
.. O F 120 アルファベット
. X F 122 機能

A.2 会議 2 の構造

. O F 0 要求
.. O F 1 本の管理
... O F 5 論文, 電子メディアも含めて, 研究室のものを管理する.
.. O F 10 検索
... O F 124 誰が借りているか
... O F 125 論文の一覧
... O F 126 本の一覧
.. X F 123 知識ベース的なもの
. X F 27 図書館の分類
. O F 47 対象物
.. O F 2 本
... O F 7 図書館登録の本
... O F 8 個人の本
... O F 78 学会誌
... O F 115 マニュアル
... O F 155 マンガ
.. O F 3 論文
.. O F 4 電子メディア
... O F 6 電子的な論文
.... O F 13 TeX
..... O F 14 卒論
..... O F 15 修論
..... O F 16 D 論
.... O F 18 文献データ
..... O F 12 BIBTeX
...... O F 33 自動

..... O F 34 力まかせ
..... O F 36 文献一覧表
... O - 81 project 毎のソフトウェア
.... O F 19 ソフトウェアのドキュメント
.... O F 87 ソフトウェア本体
.... O F 88 実験データ
..... O F 98 不定型データ
..... O F 99 フォーマット
..... O F 100 テープ起こし
..... O F 101 画像データ
.... O F 89 ハードコピー
.... O F 94 分野毎
..... O F 97 コマンド
..... O F 96 卒論などは，今までは年度毎に管理されていたが，分野毎の方が良い．
... O F 134 UNIX の Disk 内のもの
.. O F 11 分類
... O F 29 内容
.. O F 17 公開
... O F 23 対象とするのは，対象物のうちの，公開しても差し支えないものである．
.. O F 32 コピーした論文
.. O F 127 電子メディアのアーカイブ
... O F 128 本の場合とやり方は同じ．(8mm テープに identifier をふるなどする．)
.. O F 129 ビデオテープ
.. O F 135 物理的形状
... O F 136 物理的形状で分類すると探すときに便利だが，明記する置き場所で判断することにする．
.. O F 137 FD
.. O F 138 MO/CD-ROM
.. O F 154 CD
. X F 35 非公開
. O F 37 登録
.. O - 28 貸しだし制限
... X F 30 期間
... X F 31 冊数
.. O - 38 データベース
... X F 53 管理者
... O F 40 データを一箇所にまとめて保存する．
.. O - 43 公開方法
... O F 44 個人のもの
.... O F 39 個人の所有物は，各自が公開してもいいものだけを任意に登録する．
... O F 45 共有のもの
.... O F 46 共有のものについては到着次第登録する．
.. X F 52 公開規定
... X F 55 プロテクション
.... X F 57 特定の人のみ公開
.... X F 58 本人の許可
.... X F 59 すべての人に公開

.... X F 60 すべての人に非公開
.... X F 61 特定の人に非公開
.. X - 66 本の置き場所
... X F 67 自分の本棚
... X F 68 共有の本棚
.. O F 116 複数ある場合
... O F 117 論文のコピーなどで重要そうなものは, 共有用に一部作っておくと便利.
. O F 41 登録手続き
.. O - 157 identifier
... O F 158 図書館登録の本に限る
.. O F 159 登録事項を入力
... O - 160 default 値の設定?
.... O F 161 型や, キーワードなどは, あらかじめ決められた値からの選択も出来るようにする.
... O F 162 全角、半角等の問題
.. O - 163 複数ある場合はそれぞれ登録
. O F 42 検索手続き
.. O F 170 インクリメンタル
.. O F 171 AND
.. O F 172 OR
.. O F 173 NOT
.. O F 174 ソート
.. O F 175 検索情報の出力
... O F 176 プリンターに出力
.. O F 177 全件検索
. O F 48 インタフェース
.. O F 188 Xwindow
.. O F 189 ターミナル
.. O F 190 emacs
.. O F 191 いろいろな環境で使えるように 3 種類ほど インタフェースを用意する.
. O - 49 データ構造
.. O F 50 すべてに共通な情報
... O F 9 所有者
.... O F 51 所有権
.... O F 62 個人
.... O F 63 研究室
... O F 82 タイトル
.... O F 102 ツール名
.... O F 144 雑誌
..... O F 149 雑誌の場合は, 雑誌名,volume,number でまとめてタイトルとする.
.... O F 145 volume
.... O F 146 number
... O F 83 著者
... O F 54 借用人
.... O F 64 研究室のメンバー
.... O F 72 一般民間人
..... O F 65 図書館の本は研究室外の人にも貸すことがある.

.... O - 79 オンライン
..... X F 80 オンライン（ディスク内のもの）についての借用人.
.... O F 178 名前
..... O F 183 default では loginname とする.
.... O F 179 所在
..... O F 182 借用人が研究室外の人の場合は名前のところに連絡先も入力してもらう.
... O F 85 キーワード
.... O F 24 分野
..... O F 20 project
..... O F 21 METHODBASE
..... O F 22 MOTOPU
..... O F 25 FORMALMETHOD
..... O F 26 グループウェア
..... O F 86 LOTOS
..... O F 95 カーデット
..... O F 103 project 名
... O F 108 何年度か?
.... O F 110 論文などの場合, 何年度のものか.
... O F 111 場所
.... O F 141 対象物が置いてある場所. 例えば, 本は本棚, 電子メディアはディレクトリ名など.
... O F 139 型
.... O F 140 本か, コピーか, ビデオテープか, など.
... O F 142 出版社
... X F 143 edition
... X F 148 組織名
... O F 164 登録者
... O F 165 登録日
... O F 184 借りた日
... O F 185 返した日
. X F 56 借用条件
- O F 69 コメント
.. O F 70 借りる時
... O F 76 本などを借りるときは, その本の「備考」の欄の貸し出し条件を見てから借りる.
.. X F 71 守らない場合
... O F 77 備考欄の貸し出し条件が守られなくても, 計算機システムでは対処しようがないので, 守ってもらうしかない.
.. O F 73 持ちだし禁
.. O F 74 コピー禁
.. O F 75 貸し出し制限（条件）などを明記する.
. X F 84 検索法
. X F 90 お約束の文章
. X F 91 謝辞
. X F 92 論文の部品
. X F 93 年度毎
. X F 105 ローダー
. X F 106 リーダー

- O F 107 project 特有データ
.. O F 104 abstract のようなもの
. X F 109 キー
. X F 112 記録するデータ
. X F 113 日付
. X F 114 名前
. X F 118 バーコード
- O F 121 一覧を出す
.. O F 119 50 音
.. O F 120 アルファベット
. X F 122 機能
. O F 130 削除手続き
.. O F 168 所有者か登録者のみ可
.. O F 132 登録してある物が古くなって捨てる場合など, 登録事項の削除を行なう.
. O F 131 修正手続き
.. O F 169 所有者か登録者のみ可
.. O F 133 登録してある物の置き場所や所有者など, 登録事項が変更された場合, 修正を行なう.
. X F 147 シリーズ名
. X F 150 同型資料
. X F 151 読むための説明
- O F 152 備考
.. O F 153 ローダーやリーダーについての情報をつける.
. O F 156 借用手続き
.. O F 180 検索後に借用する
. O F 186 返却手続き
.. O F 187 検索後に返却する
. O F 166 ログ
.. O F 167 登録者と登録日のログを残す.
.. O F 181 借用, 返却についてのログも残す.(借用人, 借りた日, 返した日)

A.3 会議 3 の構造

. O F 0 要求
.. O F 1 本の管理
... O F 5 論文, 電子メディアも含めて, 研究室のものを管理する.
.. O F 10 検索
... O F 124 誰が借りているか
... O F 125 論文の一覧
... O F 126 本の一覧
.. X F 123 知識ベース的なもの
. X F 27 図書館の分類
. O F 47 対象物
.. O F 2 本
... O F 7 図書館登録の本
... O F 8 個人の本
... O F 78 学会誌

... O F 115 マニュアル
... O F 155 マンガ
.. O F 3 論文
.. O F 4 電子メディア
... O F 6 電子的な論文
.... O F 13 TeX
..... O F 14 卒論
..... O F 15 修論
..... O F 16 D 論
.... O F 18 文献データ
..... O F 12 BIBTeX
...... O F 33 自動
...... O F 34 力まかせ
..... O F 36 文献一覧表
... O F 81 project 毎のソフトウェア
.... O F 19 ソフトウェアのドキュメント
.... O F 87 ソフトウェア本体
.... O F 88 実験データ
..... O F 98 不定型データ
...... O F 99 フォーマット
...... O F 100 テープ起こし
...... O F 101 画像データ
.... O F 89 ハードコピー
.... O F 94 分野毎
..... O F 97 コマンド
..... O F 96 卒論などは, 今までは年度毎に管理されていたが, 分野毎の方が良い.
... O F 134 UNIX の Disk 内のもの
.. O F 11 分類
... O F 29 内容
.. O F 17 公開
... O F 23 対象とするのは, 対象物のうちの, 公開しても差し支えないものである.
.. O F 32 コピーした論文
.. O F 127 電子メディアのアーカイブ
... O F 128 本の場合とやり方は同じ.(8mm テープに identifier をふるなどする.)
.. O F 129 ビデオテープ
.. O F 135 物理的形状
... O F 136 物理的形状で分類すると探すときに便利だが, 明記する置き場所で判断することにする.
.. O F 137 FD
.. O F 138 MO/CD-ROM
.. O F 154 CD
. X F 35 非公開
. O F 37 登録
.. X F 28 貸しだし制限
... X F 30 期間
... X F 31 冊数
.. O F 38 データベース

... X F 53 管理者
... O F 40 データを一箇所にまとめて保存する.
.. O F 43 公開方法
... O F 44 個人のもの
.... O F 39 個人の所有物は, 各自が公開してもいいものだけを任意に登録する.
... O F 45 共有のもの
.... O F 46 共有のものについては到着次第登録する.
.. X F 52 公開規定
... X F 55 プロテクション
.... X F 57 特定の人のみ公開
.... X F 58 本人の許可
.... X F 59 すべての人に公開
.... X F 60 すべての人に非公開
.... X F 61 特定の人に非公開
.. X F 66 本の置き場所
... X F 67 自分の本棚
... X F 68 共有の本棚
.. O F 116 複数ある場合
... O F 117 論文のコピーなどで重要そうなものは, 共有用に一部作っておくと便利.
. O F 41 登録手続き
.. O F 157 identifier
... O F 158 図書館登録の本に限る
.. O F 159 登録事項を入力
... O F 160 default 値の設定?
.... O F 161 型や, キーワードなどは, あらかじめ決められた値からの選択も出来るようにする.
.... O F 195 借用人の default 値は , loginname.
.... O F 196 登録者は loginname , 登録日 , 借りた日 , 返した日は当日の日付 .
... O F 162 全角、半角等の問題
.... O F 199 有り得るパターンを自動的に検索
.... O F 200 使用者が考慮して検索
.... O F 201 推奨する入力形式の設定
.... O F 202 フォーマッティング
.... O F 203 部分マッチングによる検索
.. O F 163 複数ある場合はそれぞれ登録
... O F 198 借りる人が別であるため .
. O F 42 検索手続き
.. O F 170 インクリメンタル
.. O F 171 AND
.. O F 172 OR
.. O F 173 NOT
.. O F 174 ソート
... O F 212 フィールドを指定
.... X F 215 逆ソート
.... O F 214 フィールド毎に順序づけを決めておく .
... O F 217 キーワードを複数入力できる場合の問題
... O F 213 一覧を出す時の並び順 .

.. O F 175 検索情報の出力
... O F 176 プリンターに出力
... O F 221 画面に出力
... O F 222 ファイルに出力
.. O F 177 全件検索
... O F 121 一覧を出す
.... O F 119 50 音
.... O F 120 アルファベット
.... O F 211 条件なしの一覧や，検索条件に合うものの一覧を出す．
.. O F 204 検索の入力方法
... O F 205 式形式で入力
.. O F 206 strict マッチ
.. O F 207 部分マッチ
.. O F 208 正規表現
.. O F 209 フィールド指定なし
... O F 210 どこのフィールドでもいいから，ある語が含まれるもの，などの検索．
.. O F 218 各フィールドで検索可能
.. O F 219 範囲指定による検索
... O F 220 日付関係のフィールドは，範囲指定で検索できるようにする．
.. O F 231 修正，削除，借用，返却手続き
. O F 48 インタフェース
.. O F 188 Xwindow
.. O F 189 ターミナル
... O F 251 curses(画面制御) を使うもの
... O F 252 コマンドラインから入力するもの
... O F 254 Xwindow 版と同じイメージにする．
.. O F 190 emacs
... O F 250 Xwindow 版と同じイメージにする．
.. O F 232 UNIX 上
.. O F 233 インタフェースの概観 (Xwindow 版)
... O F 234 トップレベル
.... O F 236 登録手続きのインタフェース
..... O F 237 登録済みデータのコピー
.... O F 239 検索手続きのインタフェース
..... O F 238 削除，修正手続きのインタフェース
...... O F 248 修正を選ぶとデータが編集可能になる．
..... O F 240 ボタンを利用して入力
...... O F 242 この場合も入力式を表示するようにする．
..... O F 241 式を直接入力
..... O F 243 UNDO ボタン
...... O F 244 検索の範囲を一段階戻す場合に使う．
..... O F 245 PRINT ボタン
..... O F 246 SAVE ボタン
..... O F 249 借用，返却手続きのインタフェース
.... O F 235 修正，削除，借用，返却を検索と分ける．
.... O F 247 修正，削除，借用，返却を検索の下に持っていく．

... O F 255 一画面前に戻るボタン
... O F 256 終了ボタン
.. O F 253 階層型のメニュー
.. O F 191 いろいろな環境で使えるように 3 種類ほど インタフェースを用意する.
. O F 49 データ構造
.. O F 50 すべてに共通な情報
... O F 9 所有者
.... O F 51 所有権
.... O F 62 個人
.... O F 63 研究室
... O F 82 タイトル
.... O F 102 ツール名
.... O F 144 雑誌
..... O F 149 雑誌の場合は, 雑誌名,volume,number でまとめてタイトルとする.
.... O F 145 volume
.... O F 146 number
... O F 83 著者
... O F 54 借用人
.... O F 64 研究室のメンバー
.... O F 72 一般民間人
..... O F 65 図書館の本は研究室外の人にも貸すことがある.
.... O F 79 オンライン
..... X F 193 ゲストアカウント
...... O F 194 研究室外の人は , 代わりの人が手続きをする.
..... X F 80 オンライン (ディスク内のもの) についての借用人.
.... O F 178 名前
..... O F 183 default では loginname とする.
.... O F 179 所在
..... O F 182 借用人が研究室外の人の場合は名前のところに連絡先も入力してもらう.
... O F 85 キーワード
.... O F 24 分野
..... O F 20 project
...... O F 21 METHODBASE
...... O F 22 MOTOPU
...... O F 25 FORMALMETHOD
...... O F 26 グループウェア
...... O F 86 LOTOS
...... O F 95 カーデット
...... O F 103 project 名
.... O F 216 キーワードの個数
.... O F 192 キーワードは , あらかじめメニューなどでいくつか用意する .
... O F 108 何年度か?
.... O F 110 論文などの場合, 何年度のものか.
... O F 111 場所
.... O F 141 対象物が置いてある場所. 例えば, 本は本棚, 電子メディアはディレクトリ名など.
... O F 139 型

.... O F 140 本か, コピーか, ビデオテープか, など.
... O F 142 出版社
... X F 143 edition
... X F 148 組織名
... O F 152 備考
.... O F 69 コメント
..... O F 70 借りる時
..... O F 76 本などを借りるときは, その本の「備考」の欄の貸し出し条件を見てから借りる.
..... X F 71 守らない場合
..... O F 77 備考欄の貸し出し条件が守られなくても, 計算機システムでは対処しようがないので, 守ってもらうしかない.
..... O F 73 持ちだし禁
..... O F 74 コピー禁
..... O F 75 貸し出し制限 (条件) などを明記する.
.... O F 107 project 特有データ
..... O F 104 abstract のようなもの
.... O F 153 ローダーやリーダーについての情報をつける.
... O F 164 登録者
... O F 165 登録日
... O F 184 借りた日
... O F 185 返した日
... O F 197 図書館の本の場合, identifier の項目も必要.
. X F 56 借用条件
. X F 84 検索法
. X F 90 お約束の文章
. X F 91 謝辞
. X F 92 論文の部品
. X F 93 年度毎
. X F 105 ローダー
. X F 106 リーダー
. X F 109 キー
. X F 112 記録するデータ
. X F 113 日付
. X F 114 名前
. X F 118 バーコード
. X F 122 機能
. O F 130 削除手続き
.. O F 168 所有者か登録者のみ可
.. O F 223 検索して削除
.. O F 132 登録してある物が古くなって捨てる場合など, 登録事項の削除を行なう.
. O F 131 修正手続き
.. O F 169 所有者か登録者のみ可
.. O F 224 検索して修正
.. O F 133 登録してある物の置き場所や所有者など, 登録事項が変更された場合, 修正を行なう.
. X F 147 シリーズ名
. X F 150 同型資料

. X F 151 読むための説明
. O F 156 借用手続き
.. O F 225 借用人の名前の入力
.. O F 180 検索後に借用する
.. O F 226 物理的な対象物を借りる時に行なう．
.. O F 227 備考の貸しだし条件を参照する．
.. O F 230 借用時の検索では，すでに貸し出し済みのものも検索できる．
. O F 186 返却手続き
.. O F 187 検索後に返却する
.. O F 228 借用人を確認してから返却．
.. O F 229 返却は原則として借用人本人が行なうが，借用人以外が手続きすることも可能．
. O F 166 ログ
.. O F 167 登録者と登録日のログを残す．
.. O F 181 借用，返却についてのログも残す．(借用人，借りた日，返した日)

付録 B

運用実験で作成された仕様書

設計者が最終的に作成し, 顧客によって承認された仕様書である.

研究室用データベースシステム 仕様書

B.1　要求

本研究室用データベースシステムに対する要求．

本の管理

- 研究室内の本，論文，その他を計算機上のものも含めて管理する．

検索

- 誰が借りているかを分かるようにして，それらのものが失くならないようにする．
- 目的の本，論文，その他を検索して探すことができるようにする．

B.2　本システムの対象物

本システムの対象となる対象物は以下のものである．

本, コピーした論文, 電子メディア, 電子メディアのアーカイブ

ただし，実際に対象となるのは，対象物の中で公開しても差し支えないものだけである．

B.2.1　本

本とは，一般に本と呼ばれるもののことであり，次のようなものを含む．

図書館登録の本, 個人の本, 論文，学会誌, マニュアル, マンガ

B.2.2　コピーした論文

コピーした論文とは，論文誌などを全部，または一部コピーしたものである．製本したものも含む．

B.2.3 電子メディア

電子メディアとは，研究室の計算機システムのディスク上にあるもののことであり，電子的な論文と project 毎のソフトウェアを含む．

電子的な論文

TeX で書かれているもの TeX で書かれているものとは，次のような論文等である．

卒論, 修論, D 論

文献データ 文献データとして，BIBTeX のデータ管理も行なう．この BIBTeX のデータは自動生成などは考えないこととする．

project 毎のソフトウェア

project 毎のソフトウェアとは以下のもののことである．

ソフトウェア本体 佐伯研で作られたツール等である．

ソフトウェアのドキュメント 上記のツール等についての説明などを記したものである．

画面のハードコピー 上記のツール等の実行画面などのハードコピーである．

実験データ これは，仕様作成会議を録画したビデオテープからテープ起こししたデータや，画像データ，音声データなど，不定型なデータである．

B.2.4 電子メディアのアーカイブ

電子メディアのアーカイブとは，以下のようなもののことである．これらは本の場合と扱い方は同じである (テープに identifier をふるなどする．)

ビデオテープ, FD, MO/CD-ROM, CD

B.3 登録，公開について

本システムで基本となる，登録，公開に関することについての，いくつかの決定事項．

B.3.1 データベース

データは一箇所にまとめて保存する．

データベースの管理者

管理者をつけるか，つけないかは implement する人の判断によるが，基本的には必要ない．

B.3.2 公開方法

本システムへの対象物の登録，公開は以下のように扱う．

個人のもの　個人の所有物については，各自が公開してもいいものだけを任意に登録，公開する．すなわち，登録することは公開を意味する．ただし，一部に制限を付加することは可能．

共有のもの　研究室共有のものについては，基本的に全て公開すべきものであるので，到着次第全て登録，公開する．

B.3.3　本の置き場所

本システムでは，全てのものについてプロテクションなどは考えないので，本の置き場所についても特に規定はしない．個人のものは自分の本棚でもいいし，共有の本棚でもよい．ただし，共有のものは原則として共有の本棚に置く．

B.3.4　貸し出し制限

登録，公開してある対象物を貸し出す場合，次のような貸し出し制限が考えられるが，本システムは，基本的に研究室内での利用を考えているので，以下のように特に制限は与えない．

- 期間
  特に制限しない．ただし，出来る限り早く返すということを心がけてもらう．
- 冊数
  これも一度に借りられる冊数は特に制限せず，必要であれば何冊借りても良い．

B.3.5　公開規定

プロテクション

公開規定について考える場合，プロテクションとして，

- すべての人に公開
- すべての人に非公開
- 特定の人のみ公開 (本人の許可)

といったことが考えられるが，物理的な対象物のプロテクションは計算機システムでは支援できないので，本システムでは，基本的にプロテクションは設定せず，貸し出し条件のコメント程度にする．

B.4　データ構造

本システムのデータ構造については，検索に必要な情報をもとにした，対象物すべてに共通な情報として，以下のものがある．ここで挙げるものが登録，検索等の手続きにおけるフィールドとなる．

ただし，対象物によって当てはまるデータがない項目もある (この場合，入力時は空白でよい)．

> 型, identifier, タイトル, 著者, 出版者, 年度 , キーワード , 場所, 備考, 所有者, 登録者, 登録日, 借用人, 借りた日, 返した日

B.4.1　型

本か，コピーした論文か，ビデオテープか，など，対象物の種類を表す．

B.4.2 identifier

identifier とは，図書館登録の本についている ID 番号のことである．そのためこれは図書館登録の本に限る．

B.4.3 タイトル

本などのタイトルであり，ソフトウェアの場合はツール名である．ただし，雑誌の場合は，雑誌名，volume, number をまとめてタイトルとする．

B.4.4 著者

B.4.5 出版社

B.4.6 年度

B.4.7 キーワード

検索の時にキーとなるような単語である．例えば，本などの場合は分野名，研究室のソフトウェアなどの場合は project 名，などとする．

project 名の例としては，次のようなものが挙げられる．

グループウェア, METHODBASE, MOTOPU, FORMAL METHOD, LOTOS, カーデット

B.4.8 場所

対象物が置いてある場所．例えば，本は本棚，電子メディアはディレクトリ名など．

B.4.9 備考

登録事項の一つとして，メモのように多目的に使うことができるものである．次のようなものを備考としてつけることが推奨される．

貸し出し条件に関するコメント

持ちだし禁，コピー禁，または貸し出しを特定の人に制限するなど，貸し出し条件などを明記するために使う．

借りる時　本などを借りるときは，その対象物の備考の欄の貸し出し条件を見てから借りる．

守らない場合　備考の欄の貸し出し条件が守られなくても，物理的な対象物については計算機システムでは対処しようがないため，これは紳士協定的なものであり守ってもらうしかない．

project 特有データ

プロジェクト特有のデータについては，abstract のようなものをつけたり，ローダーやリーダーについての情報 (README ファイルについてなど) をつけるために使う．

B.4.10 所有者

所有者は，その対象物の所有権を持つ者である．個人のものであれば，個人名，共有のものであれば，研究室，などである．

B.4.11 登録者

B.4.12 登録日

B.4.13 借用人

借用人については，一般には研究室のメンバーであるが，図書館の本は研究室外の人にも貸すことがあるので，研究室外の人も対象とする．また，オンライン (ディスク上のもの) については，借用人は考えないこととする．

名前　借用人の名前は，default では login name，すなわち本システムを現在使用している人とする．

所在　借用人が研究室外の人の場合は，所在，連絡先も入力する．

B.4.14 借りた日

B.4.15 返した日

B.5 データベースの操作

B.5.1 登録手続き

登録手続きは，本システムに新しく対象物を登録する場合に行う手続きである．

登録は，基本的にデータ構造のところで挙げた登録事項を一つ一つ入力していく．具体的な手続きはインタフェースによる．

default 値の設定

入力の便宜をはかるため，型や，キーワード などは，あらかじめ決められた値からの選択も出来るようにする．また，借用人は login name を default 値とし，さらに

- 登録者は login name
- 登録日，借りた日，返した日は当日の日付

とする．

identifier

図書館登録の本に限り，図書館における ID 番号 (identifier) も登録することとし，借用，返却手続きの場合にその identifier を入力することによって，目的の本を検索できるようにする．

本などが複数ある場合

本，コピーした論文などで，同じものが複数ある場合，それらが公開可能な場合はそれぞれを別のものと見て登録する．(登録時の便宜はインタフェースで考慮する)

全角、半角等の問題

タイトル，著者などを入力する場合，全角で入力するか，半角で入力するかなどによって，検索のときに問題が出てくる．

このため対処方法として，

- 登録時に入力された内容は，自動的にフォーマッティング (例えば全て半角へ変換) してから登録されるようにする．
- 部分マッチングによる検索を行う．

とする．

B.5.2 検索手続き

検索手続きは，目的の対象物を探すときに行う手続きである．

検索手続きについては，普通の基本的な検索が出来るようにする．ただし，普通の基本的な検索とは，以下のような検索を出来るものとする．

各フィールドでの検索　特定のフィールドを対象とした検索である．

全フィールドでの検索　特定のフィールドを指定せず，全てのフィールドを対象とした検索である．

これらのそれぞれの場合について

- インクリメンタルサーチ
- AND, OR, NOT 検索
- strict マッチング
- 部分マッチング
- 全件検索による一覧

による検索を可能とする．

また，年度，登録日，借りた日，返した日の日付を表すフィールドに関しては，

- 範囲指定による検索 (範囲指定の方法は決められた式形式とする)

も可能とする．

ソート　検索手続きによって一覧を出す場合，フィールドを指定することによって各フィールドを対象としてソートして表示することが出来る．

ソートの順序づけは，各フィールド毎に一通り (例えば，タイトルはアルファベット順というように) に決められているとし，逆ソートは考えない．

検索情報の出力

検索情報は以下のような出力が可能である．

- 画面に出力
- プリンターに出力
- テキストファイルに出力

B.5.3 削除手続き

登録してある対象物が古くなって捨てる場合など，登録事項の削除を行う．

具体的な手続きは

1. 削除すべき対象物を検索する．
2. 削除の操作を行い削除する．

この削除手続きは，所有者か登録者のみ可とする．

B.5.4 修正手続き

修正手続きは，登録してある対象物の置き場所や所有者など，登録事項が変更された場合に行う手続きである．

具体的な手続きは

1. 修正すべき対象物を検索する．
2. 修正の操作を行ったあとで修正する．

この修正手続きは，所有者か登録者のみ可とする．

B.5.5 借用手続き

借用手続きは，本システムに登録されている対象物 (ただし，物理的な対象物に限る) を借りる場合に行う手続きである．

具体的な手続きは

1. 借りたい対象物を検索する．
2. 備考を読み，貸し出し条件を確認する．
3. 借用人を入力し，借用の操作を行なう．

B.5.6 返却手続き

返却手続きは，本システムに登録されている対象物を，借用手続きを行って借りていた場合，その本を返却する場合に行う手続きである．

具体的な手続きは

1. 借りていた対象物を検索する．
2. 自分が借用人になっていること (実際に借用していた対象物であること) を確認する．
3. 返却の操作を行い，借用人が空欄になったことを確認する．

返却は原則として借用人本人が行うが，借用人以外が手続きすることも可能とする．

B.6 ログについて

本システムは管理，運用のための情報として，ログをファイルに残すこととする．その残すべきログの内容は次のものである．

- 登録に関して，登録者と登録日
- 借用，返却に関して，借用人，借りた日，返した日

B.7 インタフェース

本システムのインタフェースとしては，登録，借用手続きなどが簡単に，使いやすくできるようにする．また，X window の環境だけでなく，PC-98 端末からも利用できるように，以下の 3 種類のインタフェースを用意する．ただし，これらの 3 種類のインターフェイスは基本的に同様の操作方法で扱えるようにする．

X window, emacs, terminal

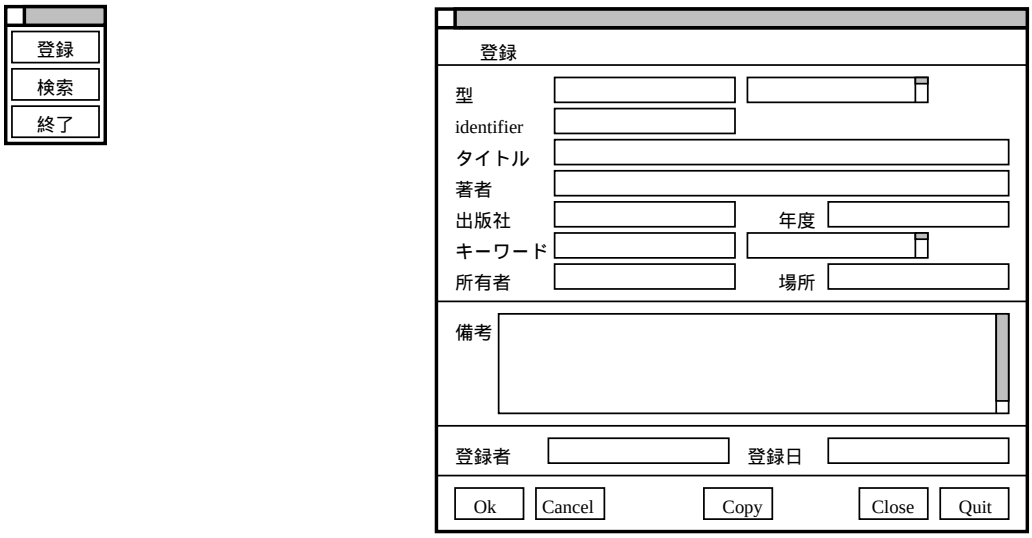


図 B.1: トップレベル，登録

B.7.1 インタフェース例 (X window 版)

X window 版のインタフェースの例．

トップレベル

- 登録，検索，または終了を選択する．

概観は図 B.1

登録

登録のインタフェースは

- 登録事項を一つ一つ入力していく．
- 選択による入力が可能なフィールドがある．
- 入力する必要がないフィールドがある．
- Copy ボタンは登録済みのデータを再利用するときに使う．

また，

- Close ボタンは，一画面前に戻る．
- Quit ボタンは，終了する．

で，これらは以下同様である．

概観は図 B.1．

検索

検索のインタフェースは

1. 検索対象とするフィールドを選択する．
2. 検索するワードを入力する．
3. 検索方法を選択する．

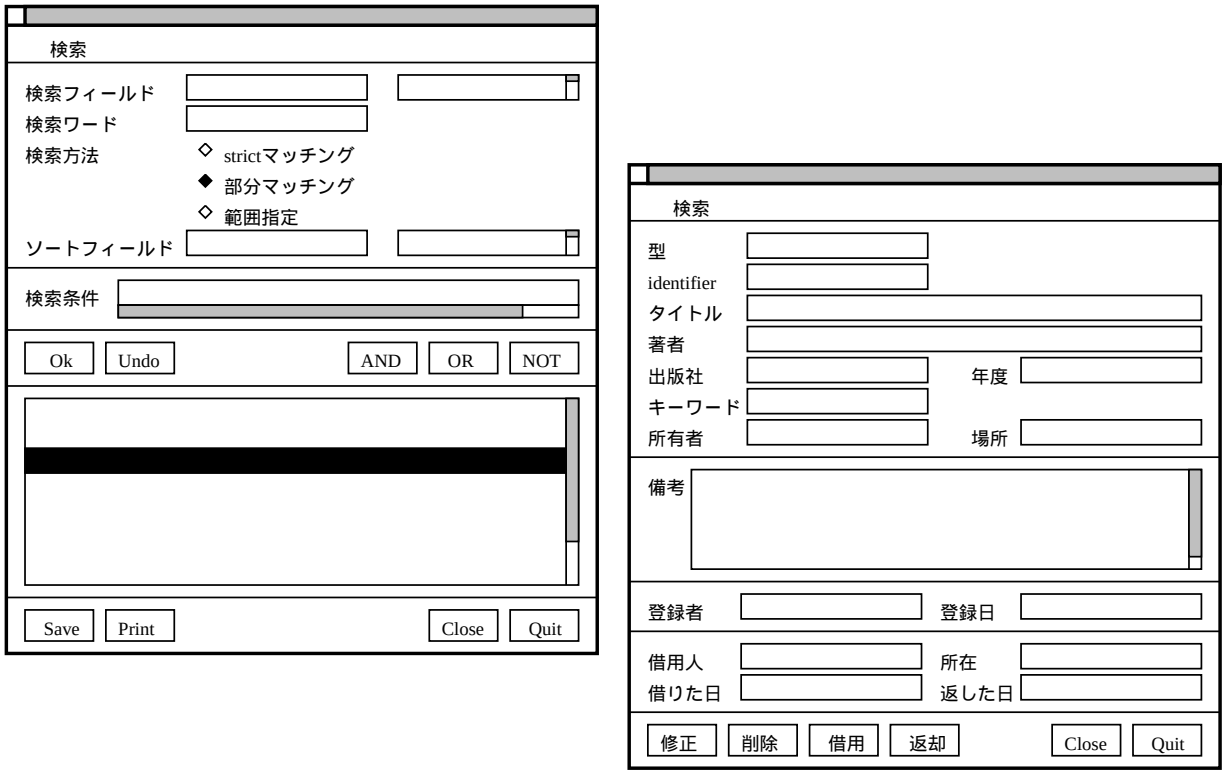


図 B.2: 検索

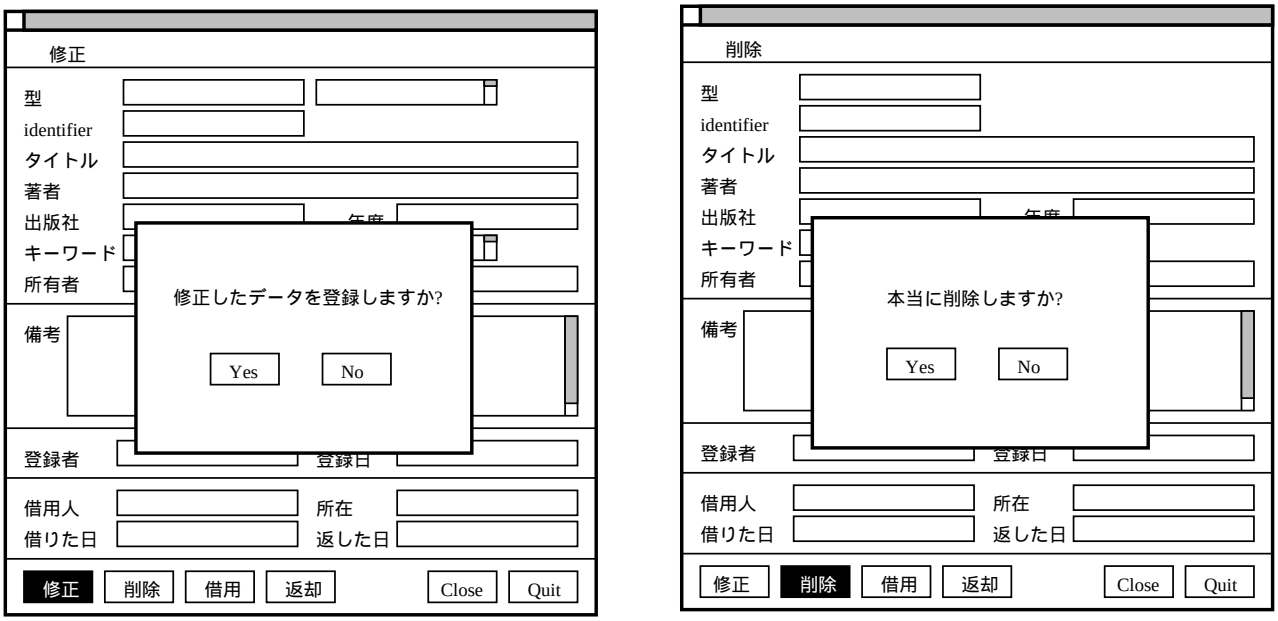


図 B.3: 修正，削除

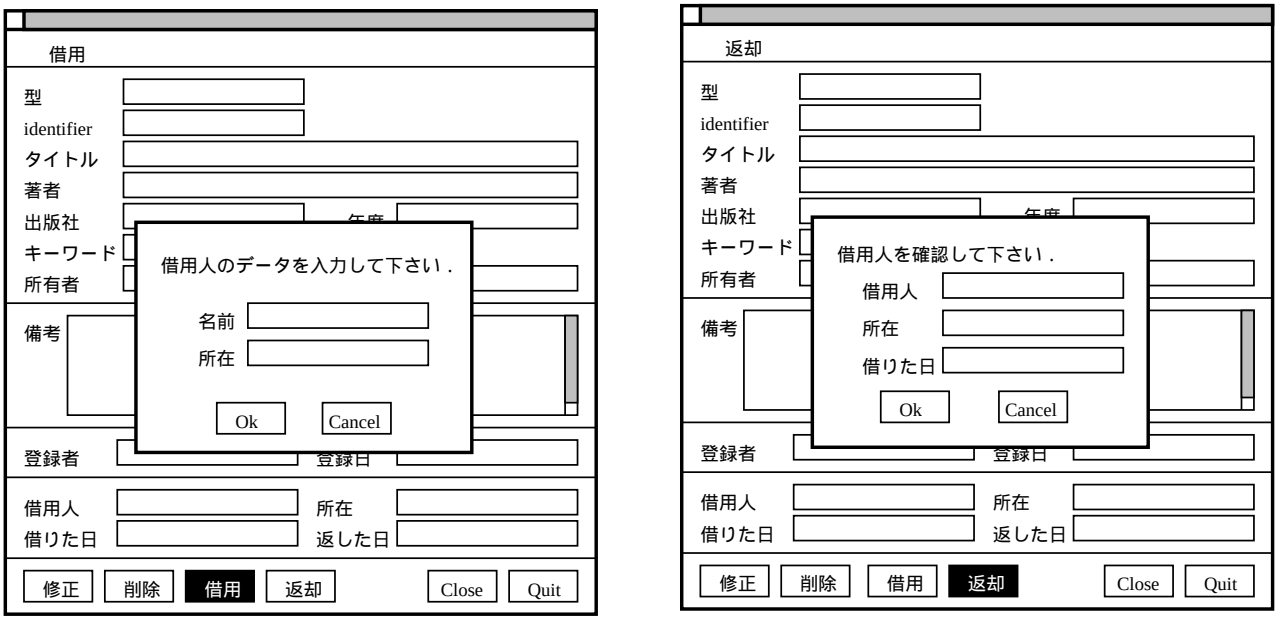


図 B.4: 借用，返却

4. 表示するときのソートの対象となるフィールドを選択する．
5. 必要であれば NOT を選択する．
6. Ok で検索を実行する．

また，

- AND, OR を選択した後で上記の操作を繰り返すことによって，検索条件を付け加えて引続き検索することが可能．
- 入力した検索条件式は検索条件の欄にも表示される．ここに直接入力することも可能．
- Undo ボタンは検索の範囲を一段階戻す場合に使う．
- Print ボタンはプリンタに出力する．
- Save ボタンはファイルにセーブする．
- 検索した結果，修正，削除，借用，返却手続きを行うことが出来る．

概観は図 B.2．

修正

修正手続きの通りで，修正を選択すると表示されているデータが編集可能になる．

概観は図 B.3．

削除

削除手続きの通り．概観は図 B.3．

借用

借用手続きの通り．概観は図 B.4．

返却

返却手続きの通り．概観は図 B.4 ．

B.7.2 インタフェース例 (emacs, terminal 版)

emacs, terminal 版は X window 版に準拠して，階層型メニュー形式のインターフェイスとする．

以上