論　文

対面式会議を支援する計算機システムの評価実験

海谷　治彦[†]　三浦　信幸[††]　穴井　豪[†††]　江幡　剛[†††]
永岡　洋樹[†††]　佐伯　元司[††††]

Preliminary Experiments of a Computer System for Face-to-Face Meetings

Haruhiko KAIYA[†], Nobuyuki MIURA[††], Go ANAI[†††], Tsuyoshi EBATA[†††], Hiroki NAGAOKA[†††], and Motoshi SAEKI[††††]

あらまし　複数の作業者が対面式会議を何回か繰り返し行なうことで，仕様書などの文書を作成する場合がある．このような会議では，会議中の議論内容を十分に反映した生産物を効率的に作成することが重要となる．本論文では，このような点を支援する場合に計算機をどのように利用すべきかについての考察を行った．そのために，通常の会議と，我々の開発した計算機システムを利用した会議での作業をそれぞれ観察し，会議参加者の発話を中心とした行動と作成された生産物の分析を行なった．その結果，会議参加者の活動を阻害しない範囲で会議履歴の記録/整理をできる限り会議中に行い，会議後に，その履歴をもとに生産物を完成させる利用方法が効果的であることがわかった．また，会議中に計算機から参照しやすいように準備されたアジェンダなどは，無理なく利用できることを確認した．

キーワード　協調作業，グループウェア，ハイパテキスト，ヒューマンインタフェース

1. まえがき

近年，計算機を利用して共同作業を支援するシステムであるグループウェアが数多く開発されている．会議を支援するための電子会議システム[1]の開発運用も盛んに行われており，それぞれのシステムは会議の多種多用な目的や形態を考慮して設計されている．例えば，地域的に分散した会議の支援を行うシステム[2],[3]などは，分散という不利な点を解消し，対面式に近い作業を行うことのできる環境を提供している．また，対面式の会議を支援する電子会議システムは，大型共用スクリーンなどを用いてWYSIWYGを提供することで，参加者の生産物の共有を促進したり[4]，IBISモデルを用いて議論の構造的な進行や記録を促進したり[5],[6]，KJ法[7]を基盤として，柔軟な発想を支援するなど[8]の特徴が見られる．また，会話のモデルを基盤とした基礎研究も盛んである[9]~[11]．

本論文では，知識や立場の異なる複数の作業者が，対面式会議を何回か行い，共通の生産物である文書を作成するための一連を作業を対象とする．そして，そのような作業を支援する場合に計算機システムをどのように利用すべきかについて考察を行う．対象とする作業の例としては，顧客側と製作者側の作業者が集まって，ソフトウェアシステムの要求仕様を決定していく場合などが該当する．そして，このような作業は，以下のような特徴をもつ．

- 会議内でのコミュニケーションは主に発話で行われてるため，議論内容は揮発的であり，また作業には関係ないノイズも多く含まれる．
- 会議開始前にはアジェンダを，会議後には議事録を通常作成するため，会議中の作業と会議後の作業とは密接な関係がある．
- アジェンダや議事録，そして最終的な生産物である文書は，会議での議論内容を反映して作成されることが望ましい．

† 北陸先端科学技術大学院大学 情報科学研究科，石川県
Graduate School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku, Ishikawa-ken, 923-12 Japan
†† 東京工業大学 大学院 電気電子工学専攻, 東京都
Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo, 152 Japan
††† 東京工業大学 工学部 情報工学科, 東京都
Department of Computer Science, Tokyo Institute of Technology, Tokyo, 152 Japan
†††† 東京工業大学 大学院 情報理工学研究科, 東京都
Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo, 152 Japan

このような特徴をもつ作業は，計算機によって以下のように改善することが可能だと思われる．

- 記録機能によって，揮発的な会話情報の消失を防ぐ．
- 会議中にシステムの編集機能によって議論内容の注釈を付加することで，会話などの記録に含まれる重要な情報を際立たせることができる．
- 会議の記録を会議内外で参照する機能によって，議論内容を十分に反映した文書を作成することができる．

これらの機能が実際の作業で有効に働くかどうかを確認するため，利用できる計算機の機能に差がある作業環境を実験的に設定する．そして，それぞれの環境下での作業者の会話や作成される生産物の分析を通して，それぞれの機能の実際の有効性の確認を行う．実験には，上記のような機能を提供し，会議における議事録などの作成を支援するシステム [12] を利用する．続く 2.では，実験の目的と設定について述べ，3.では，本論文で利用したシステムを紹介する．そして，4.で，既存のシステムや研究との比較を交えて，実験結果について考察を行う．

2. 実験の目的と設定

対面式会議を何回か行い文書などを作成する一連の作業を計算機で支援する場合，どのような支援が有効であるかを確認するために，以下のような作業を実験的に設定する．これは，対面式会議を何回か行い文書などを作成する一連を作業の典型的なものであると思われる．

- 作業の目的は，顧客からソフトウェア開発者が要求仕様を獲得し，自然言語で記述されたソフトウェアシステムの仕様書を完成することである．
- 会議の参加者は以下のとおりである．

顧客: 対象システムの要求を出し, 仕様を承認する．作成されるシステムの利用者なども含まれる．

設計者: 顧客の要求を分析し, 生産物である仕様書を記述する．

操作者: 会議における書記に該当する．支援システムを利用する場合は，書記の作業に加えて，そのシステムの操作を行う．

- 会議は会議室において円卓を囲みながら対面式の形態で計画的に数回行う．
- 議事録やアジェンダなどを作成するために，会議前後に文書作成作業が行われる．

本論文で対象とする作業は，以下のような点を計算機で支援することが可能であると思われる．

- 発話などの会議中の情報を機械的に記録する．これによって，一般には記録に残らない情報を残すことができ，その記録をもとに会議の議論内容を十分に反映した文書を作成することが可能となる．
- 会議参加者が明示的に計算機を操作し議論内容を記録し，発話などのノイズの多い情報の中での重要な情報を際立たせることが可能となる．これによって，会議後の文書作成作業を効率化することができ，場合によっては，会議中にリアルタイムに文書を作成すること可能となる．

それぞれの記録された情報は，会議中の議論，および会議後で生産物を作成する際に参照可能とする．これは会議中に以前行われた会議の議事録を参照することに該当する．上記の支援機能にもとに，計算機の利用のレベルが異なる以下のような 3 種類の作業形態の実験を設定する．

[作業形態 1] システムを利用しない作業形態．

システムは利用せず，必要であれば黒板などに図を書きながら議論を進める．アジェンダなどは通常の印刷物として用意され，会議の進行に合わせて書記が議事録を記録する．

[作業形態 2] 会議内容の記録のためにシステムを利用する作業形態．

会議中に，システムは会議内容の記録のために利用し，利用者は計算機を利用しない会議に近い形で作業を行う．そして，会議後に，設計者がシステムに記録された会議の内容をもとに仕様書や議事録を作成する．会議後の作業では，前回の会議の記録を整理して生産物を作成する作業のみを行い，アジェンダなどは通常の印刷物として別途用意する．

[作業形態 3] 会議中にシステムの編集機能を積極的に利用する作業形態．

作業形態 2 に加えて，会議中に，それぞれの作業者がシステムの操作し，議論内容の記録や注釈を付ける．可能ならば，リアルタイムに議事録などの文書を作成する．また，会議後の作業では，前回の会議の記録を整理して生産物を作成する作業に加えて，次回の会議のためのアジェンダなどをシステムに入力しておく．

これらの作業形態の実験を以下の観点から分析し，このような会議を計算機で支援する方法について考察する．

● 作業形態 1 から，一般の会議の問題点の中で，計算機によって支援可能な問題を明らかにする．問題点は以下のようなものが考えられる．

– 議論内容が落丁: 会議では議論されたにもかかわらず，最終的な生産物には，その記述がない部分が存在する．これによって，議論内容を反映していない生産物が作成されてしまうと考えられる．

– 同じ議論の繰り返し: 同じ議題に対して同じ結論を与える議論を，繰り返し行う．最終的な生産物を作成する観点からは，議題に対する結論を一度だけ作成するのが最も効率的な会議であると考えられる．

– 結論のない議論: 結論のない議論は，同じ話題の議論を繰り返す原因を作り出し，最後まで結論がない場合，議論に費やした時間が無駄になる場合がある．また，その議論に依存する他の議論の決定に影響を与えるため，生産物内の矛盾を引き起こす恐れもある．

– 結論変更による矛盾の発生: ある部分の結論を変更することによって，同時に再考の必要がある部分が存在する場合がある．もし，その再考を行わなかった場合，生産物内に矛盾が発生する恐れがある．

● 会議中のシステムの機能が，会議後の議事録やアジェンダなどの文書を作成する作業に貢献しているかどうかを調べる．これによって，会議中のシステムの利用が，会議後に議論内容を十分に反映した生産物を作成する作業の効率化に貢献していることを示す．

● システムの利用が会議中の作業にどのような影響を与えるかを調べる．具体的には，計算機からアクセス可能な形で事前に準備された話題が，実際の会議中にどの程度，利用されているかを調べる．これによって，会議中における計算機を利用した情報参照が，実際に有効活用できることを確認する．

上記の分析は，利用するシステム中のデータの構成要素の数，作業時間，作業者へのインタビューなどをもとにして行う．

3. システムの概要

本論文で利用する計算機システムは，3 人から 6 人程度の小人数のグループが，対面式会議と，その会議の準備や議事録作成などの文書作成作業を会議前後に繰り返すことで，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．例えば，ソフトウェア開発の上流工程において，顧客と製作者が会議を通して仕様書を作成する作業や，新しい商品を開発するための企画を関係者の間で議論することにより企画書を作成する作業などに適用することができる．このシステムは以下のような特徴をもつ．

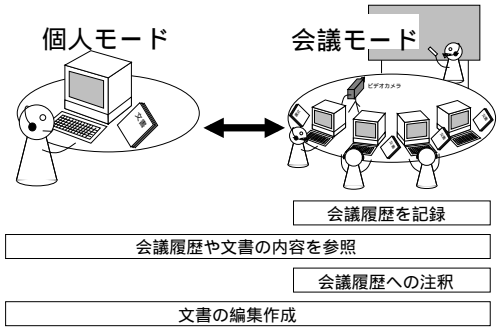

図 1　システムの利用形態
Fig. 1　Using modes of the system.

● 会議中の議論と会議前後の作業の両方を統合的に支援する．

● 作業の生産物である文書と，会議履歴の 2 階層のデータを関係づけて記録している．

● 会議履歴は発話や身振り，板書などのマルチモーダルな情報を含め計算機内に直接に記録する．

● 構造化された文書と会議履歴を整理，編集，検索するためのグラフィカルインタフェースを提供する．

以下にそれぞれの特徴について説明する．

システムを適用する作業の性質から，会議中の作業と会議前後の作業の両方を支援する必要がある．そのため，本システムは図 1に示すような会議モード と個人モード の二つの形態で利用が可能であり，これらの形態の作業を繰り返し行こことで，生産物である文書を作成する．会議モード では，個々の作業者は計算機を与えられた形態で議論を行う．このとき，システムは主に作業者の発話などの記録を行う．また，作業者は個々に与えられた計算機を通して，会議履歴や生産物の内容を参照することができる．また，作業者が計算機利用に慣れているのであれば，議論内容の注釈をテキストとして入力し，後に文書作成作業に役立てることも可能であり，更に，会議中にリアルタイムに文書を作成することも可能である．個人モード では，書記などの作業者が，会議前後に，会議の記録を整理し議事録を作成したり，続く会議のためにアジェンダを作成したりする作業の行う．このとき，システムは文書作成のための記録の参照，および記録の編集，文書の作成の機能を作業者に提供する．

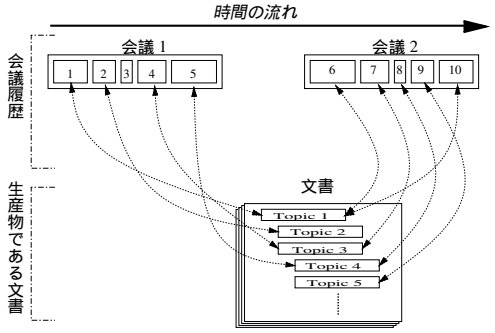

図 2　会議履歴と文書の関係
Fig. 2　Relationship the meeting records and the output-document.

図 3　システムの GUI の概観
Fig. 3　Overview of GUI.

システム内では図 2に示すように，生産物である文書と会議履歴の 2 段階にデータを分けて記録し，それぞれを関係付けて管理している．文書は and/or 木構造の形式をもち，木構造の葉にあたる部分に自然言語の短文が記述されている．この短文をシステム内では Topic と呼んでいる．また，単一の話題を議論している連続した発話や行為の列を，システム内では Discussion と呼んでおり，その構成要素である発話や行為を Action と呼んでいる．それぞれの Topic はアジェンダとして事前に準備されているか，会議中の Discussion の内容をもとに議事録として作成されるかのどちらかである．図中の会議を表す大きな方形中の番号の付いた方形は Discussion を表している．そして，図中の双方向矢印は，ある Topic がある Discussion で取り上げられたことを示している．例えば，Topic 1 は，Discussion 1, 6, 10 の 3 箇所で議論されている．図中には記述されていないが，双方向矢印には属性を付加することができ，その属性によって，ある Discussion が提案されたままの状態なのか，決定されたか，否定されたかを記録することができる．例えば，Topic 3 は，Discussion 4 で提案され，Discussion 7 で決定されたなどの形で，Topic 3 にかかわる大まかな決定までの推移を記録することができる．また，後に大まかな議論の流れを概観するために，Discussion に対してその要約の文書を注釈として付加する機能がある．このような注釈文をシステム内では Explanation と呼ぶ．

会議履歴の最小単位である Action は，参加者の個々の発話を自動的に計算機内に記録した音声データ，ジェスチャーや黒板などへの記述を手動で記録した画像データである．それぞれの Action には，発話などを行った作業者の名前と，行われた開始/終了時間が自動的に付加される．また，会議中に，それぞれの参加者は明示的に計算機にテキストを入力することで，会議履歴に注釈を付加することができ，このテキストにも Action と同様に記述した作業者と，記述の開始/終了時間が自動的に付加される．

構造化された文書と会議履歴を整理，編集，検索するために，図 3に示すようなグラフィカルインタフェースを提供する．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．以下に図中のインタフェースと機能について説明する．

- Browser (図左下): 文書の木構造を箇条書で表示しているインタフェース．個々の Topic を示す部分から，その Topic が話題となった Discussion を属性付きで列挙することができる．また，個々の Discussion 内の会話などの Action を再生することも可能である．これによって，参加者は議決内容以外に，議決の経緯や，結論が出ているか否かを確認することができる．また，Topic の議論された順序情報を用いて，ある Topic の議決の変更に伴って影響を受けると思われる Topic を自動的に列挙することが可能である [13]．
- Action_editor (図左上): 会議中の議論の流れを時系列的に参照するためのインタフェース．2 次元の表形式になっており，縦軸が会議の参加者，横軸が左から右に時間の流れを示しており，表中の小方形が Action に相当し，その長さは発話などの長さに相当する．この小方形を指示することで，該当する Action の音声や画像を参照することができる．また，いくつ

表 2　実験 1 の結果
Table 2　Results of experiment#1.

会議番号	1	2	3	4	合計
会議時間 (分)	165	147	174	210	696
Discussion 数 (個)	32	67	54	36	189
Topic 数 (個)	23	38	46	23	130

表 5　実験 1 の会議における問題点の割合 (%)
Table 5　Rate of negative factors in experiment#1.

会議番号	1	2	3	4	全体
議論内容が落丁した Discussion	34	27	28	25	28
同じ Discussion の繰り返し	28	43	15	36	31
結論のない Discussion	25	6	7	6	10
落丁した Topic	35	39	30	26	33

かの Action を囲む大きな方形は，Discussion を示し，そこで取り上げられた Topic を列挙することができる．

4.　実験の結果と考察

それぞれの作業形態の実験の規模や複雑さを示すために，表 1には，三つの実験の作成対象となるシステムと作業者に関するデータを示し，表 2から表 4には，それぞれの実験における会議時間や，システム中に作成された Discussion, Topic などの数を示す．表 1中の作業者の人数の欄には，それぞれの作業者の本職を付記している．被験者は情報工学科の学生および職員であるため，計算機操作およびキーボード入力に十分慣れている．表 2に示す実験 1 は，システムを利用しない従来どおりの会議であるため，Discussion, Topic などの数は，会議の様子を録画したビデオ記録をもとに，分析者が推定した．表 3と表 4中の，“会議モードの作業時間” は，実際に会議を行った時間に相当し，“会議後の個人モードの作業時間，前回の分析” は，直前の会議の記録を検索参照しながらシステムで議事録を作成する作業にかかった時間を表し，“会議後の個人モードの作業時間，次回の準備” は，続く会議のためにアジェンダを考えながらシステムに入力した時間に相当する．よって，実験 2 には “会議後の個人モードの作業時間，次回の準備” の行ない．実験 2 の会議 3 は，実際には 1 時間程度の休憩をはさんで，2 回の会議が行われたが，会議の間に作業などは一切行なわれなかったため表 3では 1 回の会議とみなしている．

4.1　実験データの評価

本論文の実験で作成対象となったのは，すべてインタラクティブシステムであるため，顧客の要求がシステム化しようとする作業の手順に従って設計者に説明され，その実現性などについて設計者と順次議論する形で議論が進められた．作業の目的は，対象領域を十分に理解した顧客が，対象領域を知らない設計者にこのようなシステムの仕様の作成を依頼する形をとっている．

このように，それぞれの実験でのシステムの種類や作業目的は実験間に差異はないと思われるが，システムの規模，参加者の経験値などは完全に同一レベルではない．また，本論文では 2.で設定した 3 種類の作業形態の作業をそれぞれ 1 データ毎に実験し分析対象としている．よって，これらのデータをもとに定量的な結論を導くのは困難であるため，それぞれの実験の分析から，2.で示した作業形態それぞれの特徴を定性的に議論する．よって，分析で求めた各種の割合を，単純にそれぞれの実験間の比較には利用できないが，大まかな傾向を議論することは可能かと思われる．

実験データの実数が大きく異なるのが，実験 1 と他の二つの実験の差である．例えば，実験 1 の仕様書のページ数は他に比べ少ないにもかかわらず，会議に費やした時間はもっとも長い．これは，実験 1 のみが計算機を利用しておらず，作業を通して利用可能な会議の記録が他に比べ少なかったために，このようなデータになったと思われる．同様の理由で，実験 1 での一つの Discussion の平均的長さは他と比べ長いものになっている．

4.2　通常の会議の問題点の支援

実験 1 で発生している問題点を明らかにし，計算機システムを適用することによってどの程度，それらの問題点が解決できるかを議論する．表 5に実験 1 の問題点の割合を示す．それぞれの割合は，Discussion の場合も Topic の場合も，表 2に示す個数を全体として，該当する性質をもつものの個数の割合を表示している．例えば，表 5の「議論内容が落丁した Discussion」の「会議番号 1」の 34 %とは，表 2に示した実験 1 会議 1 に出現する Discussion の個数 32 個のうち，34 %にあたる 11 個の Discussion の内容が文書として残らなかったことを示している．実際の議論の中には，表 5に示す問題点の中の複数の問題点に該当する議論がある．例えば，同じ議論が繰り返された上に，その内容が生産物から落丁したり，結論のない議論を，何回も繰り返すなどの事例が通常見られる．しかし，それぞれの問題点によって起こる会議や生産物上の不具合は，それぞれ種類が異なるため，上記のような分類を行った．よって，表 5の列方向の割合の和は 100 %にはな

表 1　三つの実験の概要
Table 1　Overview of three experiments.

	実験 1	実験 2	実験 3
作成対象	カード型のソフトウェアの仕様を記述するためのシステム	大学の研究室での文献や計算機データの管理システム	国際会議のプログラム委員長の業務支援システム
仕様書のページ数（A4 用紙）	4	12	10
顧客人数（本職）	2（大学 4 年生）	2（大学職員, 大学院生）	1（大学職員）
設計者人数（本職）	2（大学 4 年生，大学院生）	1（大学院生）	1（大学院生）
操作者人数（本職）	1（大学 4 年生）	1（大学 4 年生）	1（大学院生）

表 3　実験 2 の結果
Table 3　Results of experiment #2.

会議番号	1	2	3	合計
会議モードの作業時間（分）	60	96	125	281
会議後の個人モードの作業時間，前回の分析（時間）	10	10	7.5	27.5
Discussion 数（個）	40	43	68	151
Topic 数（個）	127	65	65	257

表 4　実験 3 の結果
Table 4　Results of experiment #3.

会議番号	会議前	1	2	3	合計
会議モードの作業時間（分）	-	31	50	30	111
会議後の個人モードの作業時間，前回の分析（時間）	-	2	2	3.5	7.5
会議後の個人モードの作業時間，次回の準備（時間）	-	3	1	0	4
Discussion 数（個）	-	22	22	24	68
Topic 数（個）	58	83	39	7	187

らない．

表 5の 1 行目に “議論内容が落丁した Discussion” を示す．書記が参加しているにもかかわらず，実験 1 では議論内容の落丁が全体で 28 %あり，行われた 4 回の会議それぞれでも同様の値をとっている．また，表 5の 4 行目の “落丁した Topic” は 33 %ある．このような問題点は，一部の会議で行われるようにテープレコーダ装置などによって会議の様子を記録し，その記録をもとに会議後に議事録などの文書を作成する作業を行えば，かなりの部分が解決されると思われる．しかし，このような記録を参照する作業はかなりの手間が必要であると思われるため，記録参照を効率的に行う支援が必要となる．一例として，我々の行った会議内容のプロコトコル分析 [13] の作業の紹介する．この作業では，会話分析 [14] の手法を一部利用し，ビデオテープに記録された会議の会話を含めた内容を発生した順番にテキストに書きおこし，その書きおこしたテキストをもとに議論を認識し，議事録のような会議内容の要約を作成する．実際に行なった作業時間は，前者の作業が会議時間のおよそ 50 倍かかり，後者がおよそ 30 倍かかり，ばく大な労力を必要とする．この作業はテープレコーダで記録した会話から議事録を作成する場合に比べ，かなり詳細な作業を行うため，単純には一般の書記が議事録を作成する作業とは比較できないが，会議履歴から文書を作成することが困難であることの一端を示している．

表 5の 2 行目に示すように，実験 1 では，同じ議論を繰り返すような非効率な作業は，会議全体を平均して 31 %程度見られる．会議は一般に冗長な議論を含むものであり，例えば同じ議論を随所で繰り返すことは，その話題の重要性や周辺の話題との関連を表している場合がある．しかし，冗長な議論を最小にすることは，会議の進行を効率化につながると思われる．実際に，これらの議論の内容を調べてみると，可否などの決定が曖昧な結論のない議論がいくつか見られる．このような問題点は，過去の同じ話題に関する議論に関する情報，例えば，先送りになった経緯や，現在の決定に至った理由などが参照できれば，減少させることができると思われる．

上記の議論より，会議での記録は，議論内容を十分に反映した文書と作るという点，および，会議での冗長な議論をできる限り防止するという点で有効であると思われる．また，ソフトウェアの要求獲得などでの要求の決定が行われる経緯や理由などは，開発の後段

階にあたる設計や保守，改版などの作業における有効な資料となり得る．同様の考えに基づき，設計理由を記録することに注目した IBIS [5] や QOC [15] などのシステムやモデルがある．しかし，これらのシステムでは，記録したデータの網構造が複雑になり，探索や参照が困難になることが知られている．このような問題点を回避するには，会議と生産物という構造に注目し，決定事項と理由とを同一レベルに記録せずに，会議履歴と生産物の 2 段階に分けて記録し相互に参照可能な仕組みを提供することが有効であると思われる．

この分析における結論がないと思われる議論は，表 5の 3 行目に示すように，およそ 10 % ある．その内容を調べると，結論を先送りにすることを繰り返した結果発生していた．このような問題点は，議決事項が決定されているか未決なのかを管理し，利用者に対して，その状態を提示する機能によって減少させることが可能であると思われる．

また，実験 1 では結論があいまいなために，議論での矛盾が発生している部分が見られた．例えば，この実験で作成しようとしているシステム中のハイパテキスト上のリンクが，作業履歴なのか，作業者が任意にはったリンクなのか明確な合意を得ずに，作業を進めているため，個々のリンク操作の意味を決定する場合に顧客-設計者間の矛盾が生じている．このような問題点に関しては，IBIS [5] や QOC [15] などに見られる，選択肢を明示する形式化をもつシステムが有効であると思われる．本論文で利用したシステムでも，グラフィカルインタフェースによって，このような選択的な問題があることが明確に表示されるため (例えば図 3, Browser 中の Topic16,17,18 など)，このような問題があいまいなまま放置されることを防ぐことができる．

これらの問題点は，参加者の技量に大きく依存する．しかし，これらは会話などの人間の基本的な活動に起因するため，程度の差こそあれ，一般に発生する問題点だと思われる．更に，ソフトウェア開発に限定すれば，対象とする生産物の複雑さから，上記のような問題点は発生しやすいと考えられる．また，ごく一部の企業を除いては，アドホックな開発過程を経るのがほとんどであることが報告されている [16]．よって，学生を被験者とした我々の実験が実際の多くの作業と大きく異なることはないと思われる．

4.3 会議外の作業に関する支援

実験 3 は，できる限り会議モードにおいて生産物の作成作業を行うことを目指したが，実際には，会議後の個人モードの作業を効率化するためのデータを入力する程度の作業しか行われなかった．しかも，会議参加者全員が編集作業を行うことが可能だったにもかかわらず，その作業のほとんどを実質的に議論に加わらない操作者が行っていた．これについて以下のような理由が考えられる．従来の電子会議システム [1] ~ [3], [5], [8], [18] の多くは，計算機操作と会話の情報を統合する傾向が少ない．例えば，論理的な作業は，計算機上のワードプロセッサや黒板，スプレットシートなどで行い，会話や遠隔地などでは相手の表情が見えることなどは，補助的な通信路としての役割をになっており，論理的な作業の枠組みの外に置かれている．よって，このような会議システムでは，会話などの情報はインフォーマルなものとして会議の記録などには残らないか，残っても利用されないのがのどちらかである．また，会話の理論 [14], [19] を利用したシステム [10], [20] も，会話の構造的な側面を Mail などの会話以外の計算機上の操作に適用している場合がほとんどである．しかし，文献 [21] の報告や，我々の行ってきた分析から，会議では会話の担う量的および質的な役割は非常に大きい．よって，会議において，キーボード操作を含む計算機の操作と，会話や板書の両上が利用可能な場合，若しくは後者で前者の機能を代行できる場合，後者が優先的に利用されると思われる．

よって，会議中の会話を中心とした記録と，その記録の会議後に利用する技術は重要になってくる．これは，一部の会議で行われるようにテープレコーダ装置を用いて会議の会話内容を記録し詳細な議事録を作成するような作業をいかに軽減するかに帰着する．これらの作業は以下のような理由で多大な労力がかかると思われる．

- 会議中に記述した文字による注釈情報と，会話などの会議履歴を関係づけて記録することができない．
- 会議履歴の参照する場合は，順次，記録を再生することがほとんどで，記録全体を大まかに見渡す方法がない．
- 会議履歴を再生して議事録などの文書を作成する場合に，会議履歴の上の情報を構造化して参照しやすくすることができない．
- 会議履歴を再生して議事録などの文書を作成する場合には，同じ場所を繰り返し再生して聞き直す場

合が多いが，この作業が大きな負担となる．

これらの問題点は，現在利用できる計算機の機能でもいくつか解決できる．例えば，計算機中に会議の記録を蓄積し，会議の記録を順次だけでなく，ランダムに参照するようなシステムは容易に構築できる．また，発話の区切れを認識し，限られた語ならは，その語を認識することが可能である[17]．よって，会議内容の記録に関しては，かなりの程度，計算機システム導入による効用が期待できる．

上記の会議履歴の処理に関する問題点は，計算機により自動的に処理可能なものと，会議参加者がある程度操作する必要なものに大別される．実験 2, 3 で利用したシステムは，3.で説明したように，これらの機能を一部が利用可能となっている．表 3に結果を示した実験 2 は，主に計算機で自動的に処理可能な機能を利用し，表 4に結果を示した実験 3 では，それに加えて参加者のある程度の操作も行った場合の結果である．しかし，上記の議論に示したように，実際には，実験 3 の設定は実験 2 に近いものになっている．これは，会議モードの作業を分析するのにかかる時間が，実験 2 の場合，281 分当り 27.5 時間で，およそ 6 倍の時間がかかり，実験 3 の場合，111 分当り 7.5 時間で，およそ 4 倍の時間がかかることなどから，4.2で紹介したビデオ記録の整理作業ほどの差はないにしても，会議中の参加者の操作が，会議後の作業の効率化に貢献することを示唆している．

そこで，実験 3 の個人モードの作業を行った設計者に対するインタビューを通して，会議中のシステムの機能によって効率化されたと思われる作業を調べた．これによって，どのようなシステムの機能が会議中に必要かを推測するこができる．

- Discussion を区切る作業: 会議内の議論を，会議から作成される文書と対応づけるために，このシステムでは，一つの話題を継続して議論している会議内の時区間を Discussion という単位に分割する．この作業が，議論内容の落丁を防ぐための基本的なデータとなる．通常，Discussion を区切る際には，議論者の発話などを繰り返し参照し内容的に判断する必要があり，その作業には多くの時間が必要となる．
- Discussion に Explanation を付加する作業: システムで議事録を作成する作業者は，まず Discussion を認識し，その内容をまとめた文章に相当する Explanation を各 Discussion に付加してよい．Explanation は文書化された Discussion の内容であるため，既存の Topic と Discussion を関係付けたり，新規の Topic を作成する作業を行う場合の手掛りとなる．よって，Explanation の作成の効率化は，構造化の作業の効率化に大きく貢献すると予想される．
- Discussion と Topic との関係をづける作業: Discussion と Topic との間の関係を作成することで，特定の話題が複数の箇所で議論された場合でも，会議全体の流れの中でのその話題に関する議論の経緯を概観することができる．通常，この関係は，会議の参加者の発話などの Action と，Topic のリストを丹念に参照して関係付けなければならない．
- Topic 間に関係を付ける作業: Topic 間の関係は，and/or 木構造であり，最終的に作成される文書などの生産物のプロトタイプとなるよう箇条書きの文書を想定している．よって，通常は，システムで議事録を作成する作業者（この実験の場合は設計者）が，個人モードでこの関係を決定する．
- Topic の文章を作成する作業: Topic は会議内の Action に含まれるキー語をもとに作成するが，通常は，会議の参加者の発話などの Action を丹念に参照して作成しなければならない．

表 6　実験 3 の個人モードでの作業の効率化の割合（%）
Table 6　Rate of efficiency in experiment#3(outside meetings).

会議番号	1	2	3	全体
会議を Discussion に分割	22	50	38	40
Discussion に Explanation を付加	-	-	71	71
Discussion と Topic を関係づける	36	27	63	43
Topic 間に関係を付ける	51	24	39	36
Topic の作成	7	8	14	8

表 6に，実験 3 の個人モードの作業で効率化できた作業の割合を示す．これは，実験 3 の個人モードの作業において，ほとんど議論内容を聞き直さずに行うことのできた作業の数の割合である．この割合は，実際に作業を行った操作者にインタビューを行うことで得た．インタビューから，これらの作業の効率化に貢献したのは，会議中において，操作者が意図的に行った独言に近い発話などの Action や，編集機能によって操作者が記述した注釈などであることがわかった．以下に，それぞれの作業別に考察を行う．

- Discussion を区切る: 表 6の “会議を Discussion に分割” の行に，区切った Discussion の中で，操作者の Action をヒントとして区切ることのできた Discussion の割合を示す．この結果より，操作者の

Action や注釈が，およそ 40 %の Discussion を区切る作業に貢献していることがわかる．通常は丹念に発話などを再生する必要があるにもかかわらず，実験 3 では，図 3中の action editor などを利用して，操作者の Action や注釈を参照することで，Discussion の開始/終了点を容易に推定することができ，Discussion を区切る作業の負担を大幅に軽減できてた．

また，会議中における発話の時間分布の中で連続して発話していない作業者の発話を目安としても，Discussion の分割ができたそうである．これは，例えば，顧客が連続して自分の要求を説明している際に，話題の区切れ目で，設計者が不明な部分の質問などを行うためである．このような発話内容ではなく，話者と話時間に関する形式的な特徴から，ある程度内容の区切れなどを予想できることは，実際の会話履歴の分析の研究 [14], [22] でも明らかにされており，計算機支援の可能性の一つと考えられる．実験で利用したシステムでは，図 3中の action editor などを通して，話者と話時間に関する形式的な特徴の目視が可能となる．

- Discussion に Explanation を付加: ある Discussion の時区間に含まれる操作者の Action や注釈は，そのまま，若しくは編集することで Explanation として利用することができる．表 6の 2 行目に，操作者の Action や注釈がもとになって作成された Explanation の割合を示す．

表に示すように，会議 1 と 2 では，注釈が付けられなかったため，Explanation としての利用が行われなかったが，会議 3 では操作者の Action や注釈のうちのおよそ 7 割を Explanation として利用することができている．

しかし，全体として見れば Explanation が付加されている Discussion 自体がそれほど多くなく，個人モードにおける作業の効率化にそれほど大きな影響は与えないと思われる．Explanation が付加されている Discussion が少ない理由は，Discussion と関係をもつ Topic 自身が，その Discussion の内容の説明となる場合が多いからであると考えられる．

- Discussion と Topic を関係づける: 表 6の 3 行目に示すように，実験 3 では操作者が会議モードで行った Action や注釈を参考に，これらの関係のおよそ 4 割を作成することができた．例えば，操作者の「査読者に関する話」という注釈をもとに，“査読者” に該当する Topic と，その注釈を時間的に包含している Discussion との関係を作成する場合を考えてみる．作業者は，まず “査読者” というキー語によって既存の Topic を検索し，現在の議論に出現している “査読者” と，既存の Topic である “査読者” が概念的に同じであることを確認するために既存の Topic 間の構造を参照し，場合によっては，既存の Topic である “査読者” が話題となった Discussion を再確認する必要がある．このように，この作業では人間が行わざるを得ない部分が多い．よって，計算機を支援に用いる場合には，例えば，概念が同じであるか否かに対するヒントを機械的に計算するための指標や，それを理解しやすい形で表示するインタフェースなどが必要となると思われる．実験で利用したシステムでは，図 3に示すような Browser によって，Topic 間の構造を目視でき，必要ならば，各 Topic が議論された Discussion を検索することが可能である．
- Topic 間に関係を付ける: 実験 3 では，「委員名の検索に関する話」という操作者の Action をもとに「委員名」「検索」などの言葉のある Topic を検索し，関係を作成することができた．このように操作者の Action や注釈によって Topic 間の関係を付けることができた割合は，表 6の 4 行目に示すように，およそ 36 %程度であり，ある程度，議事録作成者の支援になったと思われる．
- Topic の文章を作成する作業: 会議モードで操作者が明示的にキー語を Action や注釈として記録すれば，この作業の負担は軽減できる．しかし，表 6の 5 行目に示すように，操作者の作業によって作成できた Topic は 8 %程度である．これは，以下の三つの理由が考えられる．第 1 に，Topic はアジェンダとして準備されている場合はそれが利用されること，第 2 に，Action として Topic のもとになる単文を会議中に独言のように発話すること自身に抵抗があること，第 3 に，抵抗がないように注釈をキー入力を行ったとしても漢字などの入力を含めると議論の進行についていけなくなる場合があることなどが考えられる．第 1 の理由に関しては次節で議論する．

4.4 会議中の作業に関する支援

実験 3 の会議モードにおいてシステムがどのように運用され，それらが会議モードにおける作業にどのように影響を与えたかを分析する．会議モードでは，以下に示すようなシステムの機能が利用された．

- アジェンダとしての Topic を見ながら議論

実験 3 では，会議のためのアジェンダは，すべてシステムに Topic の構造として事前に入力してあり，そ

表 7　実験 3 の会議モードにおけるシステムの利用
Table 7　Rate of tool-use in experiment#3(in meetings).

会議番号	1	2	3
会議モードの作業時間 (分)	31	50	30
会議モードで Topic を参照しながらの作業時間の割合 (%)	71	94	80
延べ Topic 数 (個)	58	141	180
会議内で参照された Topic 数の割合 (%)	100	57	17
該当会議における新出 Topic 数 (個)	58	83	39
会議内で参照された新出 Topic 数の割合 (%)	100	96	80

れらを会議中に見ながら，議論する形式を取った．表 7の 2 行目に示すように，およそ 7 から 9 割の時間は，Topic を参照しながら会議を行った．すなわち，会議中の多くの時間は，過去に議論された内容や，事前に準備されたアジェンダをもとに議論が行われているのである．4 行目には，会議内で参照された Topic 数の割合，すなわち，それぞれの会議の開始時点でシステム内に存在する Topic のうちで，その会議内で参照された Topic の数の割合を示している．これは，会議の回を追うごとに，100, 57, 17 %と割合が下がっている．6 行目には，会議内で参照された新出 Topic の数の割合を示す．新出 Topic とは，直前の会議には存在せず，直前の会議と，注目している会議の間の個人モードで作成された Topic である．この割合は，どの会議も 8 割以上であることから，新規に用意されたアジェンダにあたる Topic や，前回の会議をもとに作成された Topic を軸として議論が進んでいることがわかる．また，4 行目と 6 行目の結果から，前々回の会議に発生した Topic などのむし返しなどは少ないことがわかる．

よって，計算機に準備されたデータを参照する程度の作業は，会議内でも問題なく行うことができることがわかり，会議前に計算機からアクセス可能な形のアジェンダなどは，会議を進める上で有効利用できることがわかった．通常，会議の開始以前に特定の作業者によってアジェンダは準備されるため，会議中の議論を進める作業とは異なり，アジェンダを準備する作業で計算機を利用することの障壁は小さいと思われる．

- 前回までの議論内容を聞き直す

顧客や設計者が前回までの会議での発言内容を正しく覚えていないために，議論が滞ってしまうような場合に，Topic をもとに過去の議論内容を検索し，そこでの発話を確認する作業が行われた．実際に聞き直しが行なわれた Topic 数の割合は，1 %程度しかなく，実際に聞き直しにかかった時間も，会議全体の 2 %程度であり，かなり少ない．聞き直す場所を探す場合，「前の会議の前半で... 」などのあいまいな時間の順序情報を用いてる場合が多かった．実験システムが提供するような，図 3に示す action editor などの，話者と発話時間を 2 次元のグラフ表現にしたインタフェースなどは，このような曖昧な検索に有効である．

- Topic の変更に伴う影響範囲を検索

ある Topic を変更あるいは削除した場合には，他の Topic を変更あるいは削除しなければならない可能性が非常に高い．その際に，変更あるいは削除しなければならない Topic を探し出す作業は，一般には困難である．しかし，隣接して議論された内容 (ここでは Topic) はこのような変更の影響範囲内であることが極めて高いことから [13]，本システムでは，このような Topic を列挙する機能がある．列挙された Topic を見て，変更あるいは削除を行えば，効率良く議論が進み，また，議決の矛盾を防止することが可能となっている．実験 3 において，実際に変更が行われた Topic は 1 事例のみであった．この場合に，影響があるとシステムの予想した Topic は四つあり，会議参加者は，この四つの Topic すべてを調査した結果，そのすべてに対して変更を行う必要があると判断した．よって，議論の隣接情報を用いた影響の予測は有効であると思われる．

5. むすび

本論文では，対面式の会議を通して，文書などの生産物の作成を行う作業に計算機システムを適用する場合に考慮する点を明らかにするために，三つの異なった型の会議の実験を行った．そして，議論内容を十分に反映した生産物を作成するという観点から，その実験の分析を行った．その結果，以下のような点を考慮して計算機システムを適用することで，作業を効率化できると思われる．

- 会議中の議論内容を生産物に反映するためには，会議履歴を記録し，それをもとに生産物を作成するこ

とが有効である．また，履歴の記録および参照作業は計算機システムを用いて効率化することができる．

- 会話を記録を中心とした会議履歴を後に参照する場合の便宜のために，計算機による注釈情報を会議参加者が付加することが有効である．この場合，議論に参加しない書記のようなシステムのオペレータがこのような編集操作を行いやすい．また，情報を参照する程度のシステムの操作であれば，議論に参加している作業者も行うことが可能であると思われる．
- 会議履歴から生産物を作成する場合の便宜のための，会議履歴の表示方法が必要である．例えば，会議中の作業者毎の発話の時系列的な分布を自動的に作成し表示することで，ある程度の話題の区切れなどを推測することができる．
- アジェンダは会議前に準備が可能であるため，会議中に作成してもよい議事録と異なり，十分な時間をかけて計算機システム内に準備することが可能である．また，その記述を参照しながら続く会議を行うことで，議事の進行の効率に貢献する．

本論文では対面式の会議を対象としているが，地域的に分散した会議を対象とする場合，参加者の見ぶりや視線などの利用が必要だと思われる．これらは，既存の地域的に分散した会議の支援を行うシステム [2], [3] などと組み合わせることで可能となると思われる．また，適用分野毎の背景知識などを，システム内に準備し，会議中に適宜，その背景知識に言及することで，より高度な支援を行うことが可能だと思われる．

謝辞　本研究を進めるにあたり, 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および助言を頂いた同研究所 ユーザ指向ソフトウェアプロセスグループのメンバに深く感謝致します．また，論文の初稿に対して有効な助言を戴いた北陸先端科学技術大学院大学の落水浩一郎教授に深く感謝致します．

文　献

[1] J.F. Nunamaker, A.R. Dennis, J.S. Balacich, D.R. Vogel, and J.F. George, Electronic meeting systems to support group work. *Commun. ACM*, Vol. 34, No. 7, pp. 40–61, Jul. 1991.

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

[3] Kazuo Watanabe, Shiro Sakata Kazutoshi Maeno, Hideyuki Fukuoka, and Toyoko Ohmori. Distributed Multiparty Desktop Conferencing System: MERMAID. In *CSCW90*, pp. 27–38, Oct. 1990.

[4] 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. *Commun. ACM*, Vol. 30, No. 1, pp. 32–47, Jan. 1987.

[5] Jeff Conklin and Michael L. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. In *CSCW86*, Dec. 1986.

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

[7] 川喜田二郎. 発想法. 中公新書, 1967.

[8] 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.

[9] Colin Potts and Kenji Takahashi. An active hypertext model for system requirements. In *7th International Workshop on Software Specification and Design*, pp. 62–68, Redondo Beach,CA, Dec. 1993.

[10] Colin Potts, Kenji Takahashi, and Annie I. Anton. Inquiry-based requirements analysis. *IEEE Software*, Vol. 11, No. 2, pp. 21–32, Mar. 1994.

[11] 土井晃一, 大森晃, 蓬莱尚幸, 渡辺勇, 片山佳則. 発話行為論に基づく要求獲得パラダイムの一提案. 第 47 回 (平成 5 年度後期) 全国大会 講演論文集 (3), p. 125. 情報処理学会, Oct. 1993.

[12] Haruhiko Kaiya and Motoshi Saeki. A Groupware for Face-to-face Meetings to Develop Software Specifications. In *InfoScience '93, International Conference organized by Korea Information Science Society*, pp. 691–698. Korea Information Science Society, Oct. 1993.

[13] 三浦信幸, 海谷治彦, 佐伯元司. 仕様作成会議の発話履歴と仕様書の構造に関する分析. 情処学研報 94-HI-53, Vol. 94, No. 23, pp. 71–78, Mar. 1994. ヒューマンインタフェース研究会.

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

[15] 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.

[16] Watts S. Humphrey and David H. Kitson. A comparison of u.s. and japanese software process maturity. In *13th international conference on software engineering*, pp. 38–49, May 1991.

[17] 伊藤克亘, 速水悟, 田中穂積. 音素文脈依存モデルと高速な探索手法を用いた連続音声認識. 信学論, Vol. J75-D-II, No. 6, pp. 1023–1030, Jun. 1992.

[18] Joseph S. Valacich, Alan R. Dennis, and J. F. Nunamaker. Jr. Electronic meeting support: the GroupSystems concept. *Int. J. Man-Machine Studies*, Vol. 34, No. 2, pp. 261–282, Feb. 1991.

[19] J.R. Searle. *Speech Acts*. Cambridge University Press, 1969.
[20] Terry Winograd. Where the action is. *BYTE*, Vol. 13, No. 13, Dec. 1988.
[21] 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.
[22] 西正博, 海谷治彦, 佐伯元司. ソフトウェアの発注者-開発者会議におけるインタラクションの分析. 情処学研報 92-HI-41, Vol. 92, No. 15, pp. 17–24, Mar. 1992. ヒューマンインタフェース研究会.

（平成 7 年 4 月 24 日受付，11 月 24 日再受付）

海谷　治彦

平成 6 年東工大大学院電気電子工学専攻博士課程修了．博士（工学）．現在，北陸先端科学技術大学院大学情報科学研究科助手．

三浦　信幸

平成 7 年東工大大学院電気電子工学専攻修士課程修了．現在，日本電信電話株式会社ソフトウェア研究所 広域コンピューティング研究部勤務．

穴井　　豪

昭和 45 年生．平成 6 年東工大工学部電気電子工学科卒業．

江幡　　剛

昭和 46 年生．平成 6 年東工大工学部電気電子工学科卒業．

永岡　洋樹

昭和 45 年生．平成 6 年東工大工学部電気電子工学科卒業．

佐伯　元司　（正員）

昭和 58 年東工大大学院情報工学専攻博士課程修了．現在，東京工業大学大学院情報理工学研究科計算工学専攻助教授