

Title	非同期・分散型会議における協調活動の支援法
Author(s)	門脇, 千恵; 落水, 浩一郎
Citation	Research report (School of Information Science, Japan Advanced Institute of Science and Technology), IS-RR-94-0013S: 1-23
Issue Date	1994-05-12
Type	Technical Report
Text version	publisher
URL	http://hdl.handle.net/10119/8423
Rights	
Description	リサーチレポート（北陸先端科学技術大学院大学情報科学研究科）

非同期・分散型会議における協調活動の支援法

門脇 千恵 落水 浩一郎

1994 年 5 月 12 日

IS-RR-94-0013S

北陸先端科学技術大学院大学

情報科学研究科

〒 923-12 石川県能美郡辰口町旭台 15

kadowaki@jaist.ac.jp

ochimizu@jaist.ac.jp

©Chie Kadowaki, 1994

ISSN 0918-7553

目次

1	はじめに	3
2	概念モデル構築の基本的方針	6
2.1	協調と調整	6
2.2	モデル化のポイント 1（状態把握）	7
2.3	モデル化のポイント 2（調整の要素）	9
3	討議プロセス群と討議空間	11
3.1	時間軸	11
3.2	詳細化の度合軸	11
3.3	成果軸	12
3.4	メタ情報による討議空間の集中管理	12
3.4.1	討議データの集配に関する集中管理	12
3.4.2	討議プロセス走行情報の集中管理	14
4	議事録データ群と議題の蒸し返し	18
5	調整支援	19
5.1	閉塞とは	19
6	まとめ	21

目 次

1	協調と調整	6
2	討議データの 2 状態	8
3	討議データの 2 層管理	9
4	調整理論	10
5	討議プロセスの寿命	12
6	動的再構成の仕組み	16
7	管理頁の例示：再討議の場合	19
8	Groupwarebase を土台とする分散型会議支援	22

1 はじめに

ネットワーク技術の進展に伴い、地理的／時間的に分散したプロジェクトチームによるグループ活動が盛んとなってきた。チームの協調状態が維持されてこそ、こういったプロジェクトの活動は進展が可能であり、その進展は「スムーズなコミュニケーション」によってもたらされる。

「スムーズなコミュニケーション」実現にあたっての一つの攻め口は臨場感の創出であり、テレビ会議システム、マルチメディア・リアルタイム通信、仮想現実感などの研究・技術開発は盛んである。

ネットワークを介したグループコミュニケーションのもう一つの重要な形態として、利用者が地理的に離れており、さらに意見のやりとりに時間的な同期をとらない非同期・分散型の会議がある。技術的にはまだ初歩的な段階であるが、メイリングリストを利用した遠隔会議はその典型である。

この形態は、リアルタイム型より協調状態の維持が難しく、会議の成功にはコーディネータによる調整の度合がよりいっそう濃く影響しやすい。その理由は、

- レスポンスまでの時間差

意思表示をしてからレスポンスを受けとるまでに時間があいてしまうために、当初の考えに変化を来しやすい。このため、意見のやりとりが幾度か続くうちに、当事者間に微妙な意識のずれを生じる。さらに、意見を提案した本人が、思考の構造の変化に気づいていない場合もある。

- 臨場感の欠落

各自が自分のペースで会議に参加可能であるために臨場感を欠きやすく、会議進行役が意見催促を頻繁に行なう必要がある。

しかしながら、解決に時間のかかるような問題を検討するためには、時間束縛に関して柔軟性を持ち、ネットワークを介した意見交換を通じて、ゴールオリエンティッドに収束してゆくこの形態の会議が、さらに発展する必要がある。現状では、その重要性にも関わらず研究は初期の段階である。

このような現状認識のもとに、プロジェクトの活動をサポートするコミュニケーション（非同期・分散型会議）を取り上げ、プロジェクトの協調状態をサポートする土台づくりを目指している。具体的には、「分散した箇所で活動しているメンバーの討議状態を集中的に管理でき、討議状態という情報の共有を可能とする」ための理論的モデルの作成を研究課題としている。

上記研究目的を遂行するにあたって、メイリングリストを利用して実施された一年間程にわたる遠隔会議の事例を参考にした。この会議の特徴は、

1. 会議進行役でかつプロジェクトのまとめ役でもあるコーディネータが存在する。
2. 会議メンバーは、電子メールなどの媒体を通して、会議参加が可能である。

討議の内容面では、

1. プロジェクトの運営的な面をサポートする討論
2. ゴールオリエンティッドな収束型の討論

を対象としていることである。

また、事例分析の結果、以下のような問題点が発生しやすいことがわかった [1]。

1. 会議メンバーは、地理的にも時間的にも分散しており、「会議への参加」という能動的な行為の殆んどは「意見の提出」といった形態でなされる。このため、参加している討議状態全体に関する情報は、会議進行役から与えられなければ分らない。つまり、場合によっては自分が参加している討議の状態把握さえ難しい。自分が参加していない討議の状態把握となると、さらに難しい。
2. 一つの会議で複数の議題に対する討議が並行する場合があります、この進捗を全体的に統括して管理することは困難である。
3. 会議の進行を中断するような閉塞状態が生じる場合がある。
4. 進行役側で意見をまとめ、討議を収束させなければならないので、調整の作業はコーディネータに集中する。

これらの問題点を踏まえた上で、対策および研究の方針を次のように定めた。

1. 問題点1に関しては、一つの討議状態を忠実に表現できるような討議プロセスモデルを作成し、「討議状態」という情報をメンバー全員が共有し、討議に参加しているという臨場感を補う。これは、レスポンスタイムの差から生じる思考のずれを埋めるのにも部分的に役立つと考える。
2. 上記対策1を基に、複数の討議を管理し、会議全体の進捗を把握できるようにする。
3. 問題点3に関しては、上記対策1を基とする閉塞状態の検出方法とその対応策を考察する。
4. 問題点4に関しては、コーディネータの「調整」作業の具体例を一般化し、調整作業を肩代りできるような支援法を検討しなければならない。

本論文では特に、上記対策の1と2に焦点を絞り、3の一部について考察を行なっている。論文の構成としては、2節において協調活動をサポートするグループウェア開発のための基礎概念である「協調」、「閉塞」と「調整」についてその相互関係を定義する。その方針に基づき、討議が生み出す情報群の構成に関してモデル化のポイントをまとめ、3節以降の各論に対するモデルの枠組を与えている。

2 概念モデル構築の基本的方針

2.1 協調と調整

人間集団は「協調」している状態ではじめて、グループを構成する意味が出てくる。グループを構成するメンバーの得手不得手の分野はさまざまであり、コミュニケーションを繰り返すことにより、各々の持つ欠点を相補いつつ共通の高次のゴールを達成できる。

しかし、グループの共同活動は常に協調しているとは限らず、協調を維持するためには、調整が必要となってくる。まず、この「調整」のきっかけを与える様々な概念を明らかにする必要がある（図1）。

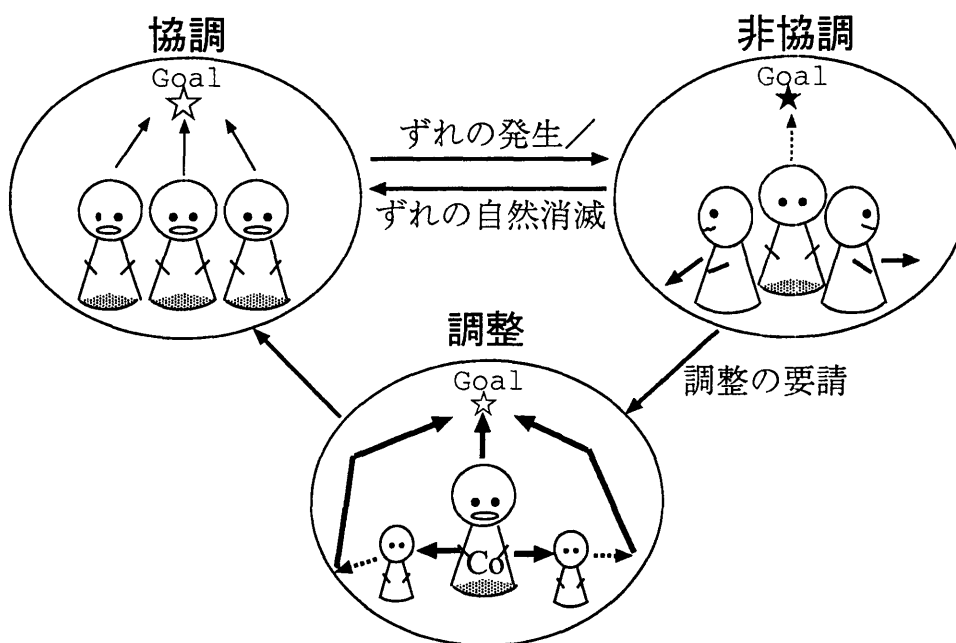


図 1: 協調と調整

図1において、協調とは、グループの形態を保ちながら作業を行ない、最終ゴールに到達しようと活動中である場合に、グループ内のメンバー個々の活動ベクトルが同じ方向にある状態と考えている。ところが、最終ゴールを目指しているうちに、各々のメンバーの持つ知識や技量の格差、あるいは思い入れ、思惑や立場・役割、人間関係など人的要因と、さらに通信手段や共有データベースの破壊、あるいは停電などグループ活動の依存する環境からの物的要因により、この活動ベクトルにずれが生じてくる。

この活動ベクトルのずれを生じる原因の中には、グループ活動の作業状態の維持を極端に阻む要因があり、これを本研究では閉塞要因と呼ぶ。閉塞要因は、自然に取り除かれる場合は少なく（取り除かれる場合はあったとしても、取り除かれるまでに長い時間がかかる可能性が強い）、この要因によって引き起こされた状態で作業が滞ったまま、未解決の

状態を閉塞状態と定義する。特にグループ活動が「会議」に限定される場合には、「何らかの要因により、通常の会議計画が中断もしくは大幅な遅延を起こしている状態」を指す。

生身の人間同士のグループ活動であるのだから、ずれやあるいは、極端な場合には対立が生じて不思議ではない。しかし、活動ベクトルのずれを放置しておいては、いつまでたっても最終ゴールにはたどり着かないだろう。そこで、ずれを生じる要因を取り除く調整が必要となる。

また、場合によっては、要因を取り除くだけでは、解決にはならず、初期の目標を変更し、ゴールの再設定が必要な場合も出てくるだろう。これも調整の大事な要素であると考えている。

調整支援を施すには、さまざまな閉塞要因の種類の中から、その要因を特定しなければならない。そのためには、正確な状態把握が必須となる。

2.2 モデル化のポイント 1（状態把握）

現在、グループウェア分野で、非同期で行なわれるコミュニケーションに関する研究には、構造化電子メール技術を利用した複数のシステムや議論の状態遷移モデルを導入した討論支援ツールであるgIBIS [2] 等が挙げられる。

まず、構造化電子メール技術の場合、会話の追跡支援を実現しているものもあり、討議内容の状態把握という点では評価できる。しかし、半構造化メッセージの場合、定型化した内容のコミュニケーションや定時的な連絡用には効果を発揮できるだろうが、たとえば、ソフトウェア開発の上流工程における要求定義など討議対象の世界の輪郭が明確でないものや、討議によって新たな付加価値を生み出していくような会議には向かないであろう。それは、討議途中でゴールの再設定を余儀なくされたり、討議中に交わされる言葉や資料に発生する因果関係、会議参加者の知識の構造が徐々に変遷を遂げていくからである。

そこで、本研究では、コミュニケーションの内容そのものの表現に主眼は置かず、表現に関してはフレキシビリティを与え、その反面、コミュニケーションプロセスの状態把握に焦点を絞る。すなわち、討議内容の状態よりは、討議進捗状態を重視する。そして、複数のコミュニケーションプロセス間で生じる前後関係や相互関係に柔軟性を与え、プロセスの走行順序の変更、プロセスの追加／除去、プロセスの目標や処理手順の変更等を平易にする。討議内容の状態把握に関しては、各々の討議進行役の判断を通して討議進捗情報に変換し管理を行なう。

また、gIBIS は、ネットワーク状の議論の構造を分かりやすく揭示する仕組みを持ち、議論の全体的な流れの中で、一つ一つの意見を追いかけてゆくことが可能である。

一方、討議途中などすべての意見が網構造になってつながっているために、最終結論や複数のノードにまたがった思考の筋の表現が困難であることや、時間的な論点の遷移が分

かりにくいという問題点がある。

本研究は、討議進捗状態の把握を目指すという点では、gIBIS と目標は一致するが、討議データの取り扱い方針に相違がある。

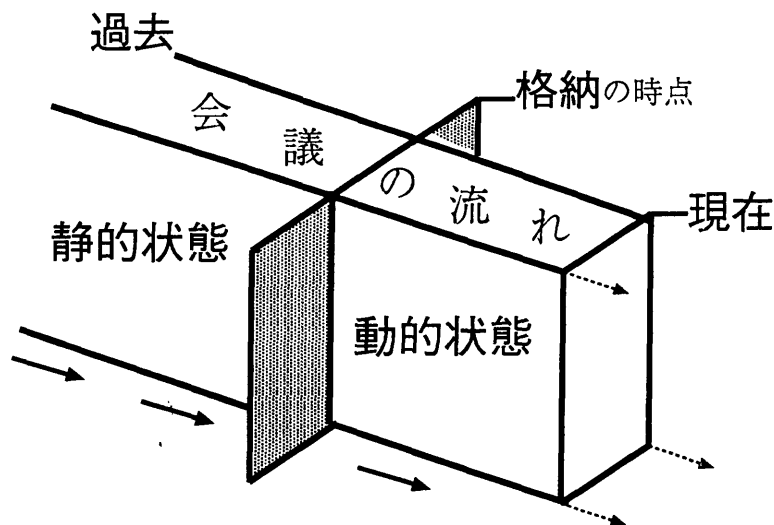


図 2: 討議データの2状態

すなわち、gIBIS のように、討議当初からのデータを一つのデータ群とはせず、「討議データ」を2つの状態に切り分けて定義し（図2）、それぞれのデータ群別に管理を行なう。

つまり、討議内容が変化しつつあるまたは変化する可能性のある動的状態を、gIBIS よりは情報の粒度の粗い討議プロセス群としてモデル化し（図3）、変化がなくなった静的状態を議事録データ群として管理する（図3）。

さらに、動的状態に関しては、討議状態を変化させるトリガーとなっている情報とその変化が生み出す情報（途中成果）を切り分け、静的状態に関しては、結果が生み出されるまでの過程と最終結果とを別管理するという方針である。これらの切り分けにより、

- 討議空間側においては、起動されている討議プロセスは、現在検討中の議題に対するものだけであり、進捗状態と論点を捉えやすい。
- 議事録データ群側では、最終的な結論の表現が明示的である。

過去の議事録間の相互関係の把握は、議事録データ群のブラウジングにより可能である。

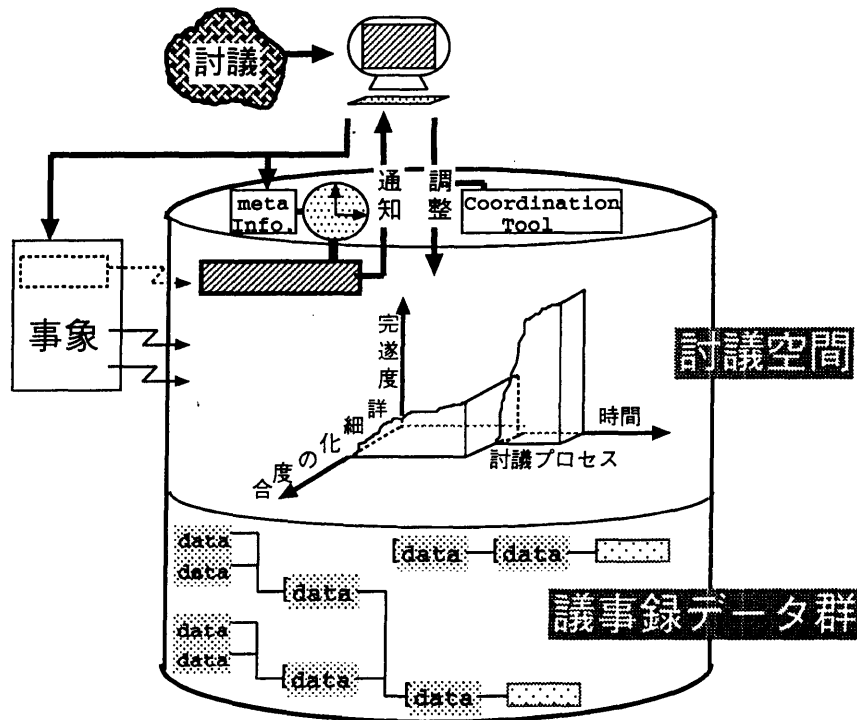


図 3: 討議データの 2 層管理

2.3 モデル化のポイント 2 (調整の要素)

調整という観点に関して、グループウェアに関する研究を総括的に捉える一つの枠組を提供しているのは T.W.Malone による調整理論である [3] [4] (図 4)。

それは、「コミュニケーション」の繰り返しにより、部分的な「グループの意思決定」をし、この部分的な意思決定を構造化する手段として「調整」というレベルを設けている。

さらに、「調整」レベルは、全体ゴールに対するサブゴールの設定である「ゴール分割」、これらサブゴールへのタスクや資源割り当て、サブゴール間の同期・順序づけのプロセスから成り立っている。

そして、これら一連の活動の中で生み出される情報を共有する手段を与えるのが、「共通オブジェクト認識」のレベルである。

この調整理論の 4 つのレベル分けについては、本研究の考え方と共通する部分が多い。しかし、調整レベルのプロセスに関しては相違がある。つまり、T.W.Malone の方はプランニング的な要素が強く、ゴールが当初から固定されている場合には有効と思える。しかし、討議などのように対象世界が限定されず、その結果についての見通しが立てにくい場合などには適用しがたいと考える。本研究は、グループ作業中に、ゴールが少しずつ変わる場合や全く変わってしまう場合などに適応できるように、「ゴール再設定」という調整の

Maloneらが提唱する調整理論	
レベル	プロセス
調整	ゴール分割
	タスク割当 資源割当
	同期
	順序づけ
グループ意思決定	代替案提案 代替案評価
コミュニケーション	メッセージ転送 ルーティング
共通オブジェクト認識	共通情報の参照・操作

図 4: 調整理論

要素も取り込んでいる。

3 討議プロセス群と討議空間

議題1つ1つは、討議のライフサイクルをもち、これを討議プロセスと呼ぶ。この討議プロセスを駆動するものは事象であり、システム外からの外部事象と、討議プロセスの処理結果を継続する内部事象がある。この討議プロセス群が作り出すのが討議空間であるが、3つの軸（時間軸、詳細化の度合軸、完遂度軸）によって特徴付けられる。

3.1 時間軸

時間軸は一連の討議の流れであり、討議プロセス間の継続関係により定義される。討議プロセスにより生み出された議事内容は、次の討議プロセスに処理結果として継続されてゆく。ところが、受け継がれる途中で他の流れの討議プロセスからの影響を受け（相互依存／排反関係）、議事内容に変化をもたらす場合がある。

a. 継続 討議プロセスの処理結果を引き継いでゆく前後関係。引き継ぎ形態から、継続なし、1対1、分割化、統合化に分類される。この決定は、1つの討議プロセスが持つ前提条件と後件条件という属性による。つまり、ある討議プロセスの後件条件は、継続する討議プロセスの前提条件となる。

b. 相互依存

付加：相手討議プロセスの処理内容が、討議プロセスの処理内容に付け加えられる。ただし、付加されるだけで元の内容の書き換えはおこらない。

変更：討議プロセスの処理内容自体の一部書き換えがおこる。

c. 排反 ある討議プロセスの処理開始により、別のプロセスの処理が必要なくなったり、もしくは相容れない場合。

3.2 詳細化の度合軸

一つの議題について、話し合われる詳細化の量を表したものである。討議空間の構成単位となる一つの討議プロセスは寿命を持ち、5つのステップ（図5）の流れとして捉えられる。

つまり、この軸は一つの討議プロセスの寿命のあいだに、その討議を達成するための議題のミニ議題への詳細化量で表現される。展開された各ミニ議題も図5に示した討議プロセスの形態を持つ。ミニ議題を生成するトリガーは、第3番目の「議論」のステップである。詳細化の起こる要因は、2種類に分かれる。

a. 計画的詳細化：会議開始時点で詳細化が判明

一つの議題では話しの内容が大き過ぎる場合に、予め細分化しておく（この細分化の‘予測’が狂うと、不都合が生じ突発的詳細化を引き起こす）。

b. 突発的詳細化：会議開始後に irregular に発生

その原因としては、以前の討議が不十分なため、また未決定の懸案事項の出現による情報

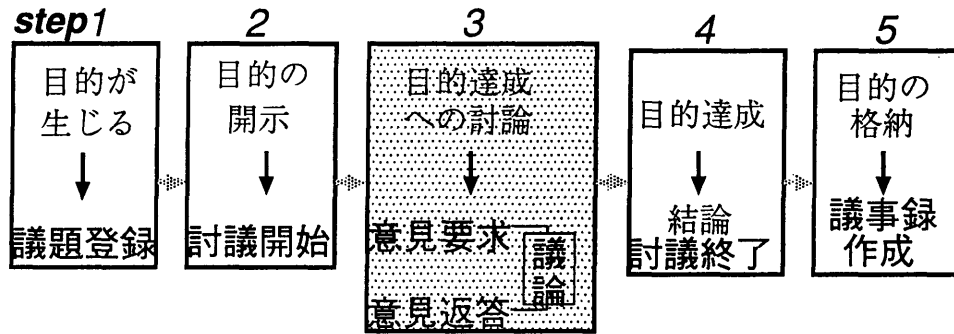


図 5: 討議プロセスの寿命

の追加などが考えられる。

最終的に、詳細化された子討議プロセスは討議終了後、討議結果を親討議プロセスに報告すると同時に、議事録データ群に結果を格納し、消滅する。

3.3 成果軸

討議がどれだけ達成されたかを示す軸である。

完遂度は人間の主観に大きく影響されるため、絶対的尺度を定義することは難しい。そこで現時点では、終了判定者が判断を下しやすいように、＜終了、延期、打ち切り、取消、未開始＞の簡潔な5段階の完遂度を定義している。ここで、＜延期＞とは後日再討議を要するもの、また＜打ち切り＞とは再討議を要さないのものである。完遂度は、後日再討議が必要となるかどうかの一つの判断材料として、あるいは議事録検証の際の資料となる。

これらの討議空間より、たとえば、討議量（時間×詳細化の度合）が異常に多いとか、討議の完遂度が低いというように、会議進捗把握の目安を得ることができる。

3.4 メタ情報による討議空間の集中管理

討議空間には、複数の討議プロセスがあり、これらを集中管理しているのはメタ情報である。メタ情報は討議プロセス運行の管理情報を持つため、討議空間の制御も可能である。また、討議データ（電子メール等）の集配局の役割も行なう。

3.4.1 討議データの集配に関する集中管理

このメタ情報の集配局を中心とした討議は次の段階を踏んで進められる。

1. 会話の必要性発生。モード選択。参加者側
 - a. 会話モード（会話支援機能のみ）
 - b. 討議モード（進捗管理、討議データ管理等の付加機能有り）

以下 b を選択した場合のみ。

2. 議題申請。参加者側

同時に進行役の推薦も可能。

3. 議題承認。コーディネータ側

討議パターン（進捗チェック用）と進行役を決定。議題を管理ブック（後述）に登録。

4. 討議開始の旨と討議プロセス ID 通知。コーディネータ側

（進行役や議題名、討議の目的、討議メンバ、開始時期等も付記）通知先は選択。

a. 不特定（メイリングリスト全員）

b. 指定された討議メンバ

5. 議題に対して意見提出。参加者側

討議プロセス ID と特別受け取り人を付記。

特別受け取り人とは、討議メンバのうち受け取り人を特に指定したい場合に使用する（メンバ全員も可能。提出意見は、進行役により編集されるため）。

6. メタ情報局にて意見集配。

(a) メタ情報がすべてのメールを収集し、差し出し人のチェックを行なう。

(b) メール of 討議プロセス ID をもとに、進行役を管理ブックより検索し、進行役と特別受け取り人宛にメールをコピーして配送。（この際、オリジナルのメールはメタ情報が副議題毎にバージョン管理を行なう。）

(c) 受けて側では、受信メールを元に返答。

この際、メールの管理は個人次第（ex. 読んだメールは削除可。データ管理の複雑化と誤操作からの解放が目的。誤操作の場合、メールの再配送可。公開可能な情報の問い合わせも可。）。

(d) 討議内容の編集。進行役側

メールの内容を自由（個人裁量）にカット & ペースト。編集作業については、「討議データ note window」で行なえば、save 後に自動的にデータ管理（メタ情報側にて集中管理）を行なう。

7. 進捗チェック。メタ情報側

各「討議データ note window」が更新される毎に、進捗チェックを行なう。もし、討議パターンの変更が要請された場合、それまでのパターンを加味しながら、変更する。

また、進行役が途中で変わった場合は、

- a. 「討議データ note window」が旧進行役からはずされ、passwd も変更。
- b. 「討議データ note window」が新しい進行役宛に設定され、new passwd が通知。
- c. メールの配送先の変更。新進行役に今までの討議結果と討議メンバーに関する情報を通知（希望により、それまでの討議に使用の全メールの配送可）。

進行役については、複数人（MAX 3 人）の指定が可能であり、主たる進行役を決定しておく。「討議データ note window」は進行役全員に設定される（共用）。討議内容の編集は、進行役が自由に行なうが、編集に際して衝突があった場合は、最終権限は主たる進行役が持つ。

3.4.2 討議プロセス走行情報の集中管理

メタ情報は、一つの討議プロセスが生成されてから消滅するまでのプロセスの走行に関する情報を持ち、これを 1 頁の構成単位とする管理ブックを持っている。1 頁あたりの項目は以下の 20 項目であり、コーディネータが入力をする。

項 1：プロセス ID プロセスの識別 ID で会議開催期間を通して一意。

項 2：議題名 討議内容を端的に表す議題が付けられるが、再討議時は元議題と同一名（4 節）。

項 3：副議題名 討議の最終目標が明確な場合に、副議題名として付けておく。討議目標が変わった時点で変更可能ではあるが、副議題毎に討議データはバージョン管理される。

項 4：起動時間（年月日、時刻） 討議プロセスの起動時刻。実際の処理開始は、イベント到着後。

項 5：予定処理時間 各討議プロセス毎に許容される討議時間を割り当てておき、超過した時点でチェック用の割り込みを行なう。

項 6：前提・後件条件 3.1 の a。

項 7：継続先 3.1 節の a。

項 8：処理手順 スクリプトで記述予定 [5]。

項 9：親プロセス ID 直属の親がある場合、親の ID。

項 10：子プロセス ID 詳細化した議題のプロセス ID であり、直属のもののみ（孫以下の情報は持たない。）。

項 11：再討議議題情報 再討議に参照する議事録情報（4 節）。

項 12：討議データの保管情報 実際討議に使用した電子メール等の保管情報。

項 13：相互依存、排反関係情報 3.1 節の b、c。

項 14：進行役

1. コーディネータ, 2. 参加者, 3. 討議プロセス（前提条件成立後、自動的に処理が進む場合）

の3ケースのいずれか。コーディネータには、種々の仕事が集積することや、すべての議題に精通しているわけではないことなどから、一つ一つの議題毎に適任者を進行役にして負荷分散を計る。

項15：パスワード 議事録編集・格納用のパスワードで進行役にのみ公開。

項16：討議パターン 種々の討議パターン(ex. 通知、調停、合意形成 etc.)のうち、これから展開されるであろう討議に一番適合しているものを選ぶ。討議パターンは、図5の第3番目の議論のステップが、それぞれパターンに応じて細分化したものである。たとえば、論点揭示、意見収集、まとめ等に細分化される。この討議パターンの使用目的は、討議の進捗チェックと進捗状況の視覚化のためである。この通りに討議を進めてゆく必要性はない。討議途中で変更可能である。

項17：終了判定者 討議プロセスの終了判定を行なう者。進行役が終了判定者も兼ねるが、複数進行役がいる場合は、主たる進行役が務める。

項18：終了状態 終了判定者が、プロセス内の処理内容を編集して処理結果(議事録)を作成し、さらにその終了状態(3.3節)を下す。終了状態が<終了>以外の場合は、延期、打ち切りや取消になった理由が、処理結果に書き込まれる。

項19：終了時間(年月日、時刻) 終了状態が書き込まれた時間。

項20：結果格納情報 討議が終了したプロセスの処理結果が議事録データ群に書き込まれる際の情報。主に同一議題に関して行なわれた討議の通し番号がカウントアップされてついている。その議題に関して一番最初の討議の場合、初期値の通し番号がセットされる。

討議プロセス自身が持つ情報は、自分自身のプロセスIDと進捗情報のみであり、その他の情報は、その都度管理ブックを参照する。

終了状態が書き込まれると、討議プロセスは必要に応じて処理結果(議事録)を継続もしくは親プロセスへ報告するとともに、<議題名、プロセスID、討議期間、同一議題通し番号、処理結果>の5項目を議事録データ群に格納する。処理結果には、作成者名と終了状態が含まれている。もし、終了判定者以外が作成した議事録を格納しようとした場合、監視機能によりリジェクトされ、コーディネータ側に異常状態が通知される。

討議プロセスは、格納処理が正常終了するとメタ情報に完了通知をし、消滅する。メタ情報側では、この該当プロセスの管理頁を管理ブックから取り外し、メタ情報内の別の場所で保管する。

また、討議の途中で生じたデータは、一固まりのデータに加工し、プロセスIDのタグを付加して、データベース内に保管する。

スタート時点では、会議の議題はプランニングされており、討議プロセス間には明確な前後関係もしくは詳細化関係があり、討議空間を構成している。しかし、討議が進むにつれて、最初の時点では予測できなかった話しの展開がおこり、新たな討議プロセスが必要になったり、討議順が変わったりする。この場合、最初のプランされた討議プロセス群の空間では、現状に即さなくなる。

そこで、討議プロセス群の動的再構成が必要となるが、それが可能となるように柔軟性を与える必要がある。すなわち、

- 討議プロセスの予定外の組み込み自由
- 後続の討議プロセスの追加／付替／除去
- 討議プロセスの処理手順の変更（部分的マージ、一部破棄等）
- 不要になった討議プロセスを抹消

等が可能でなければならない。このような動的再構成は、状況（事象）に応じて駆動可能な事象駆動型討議プロセスにより可能となる。

この動的再構成の情報も、メタ情報に記憶される（図6）。

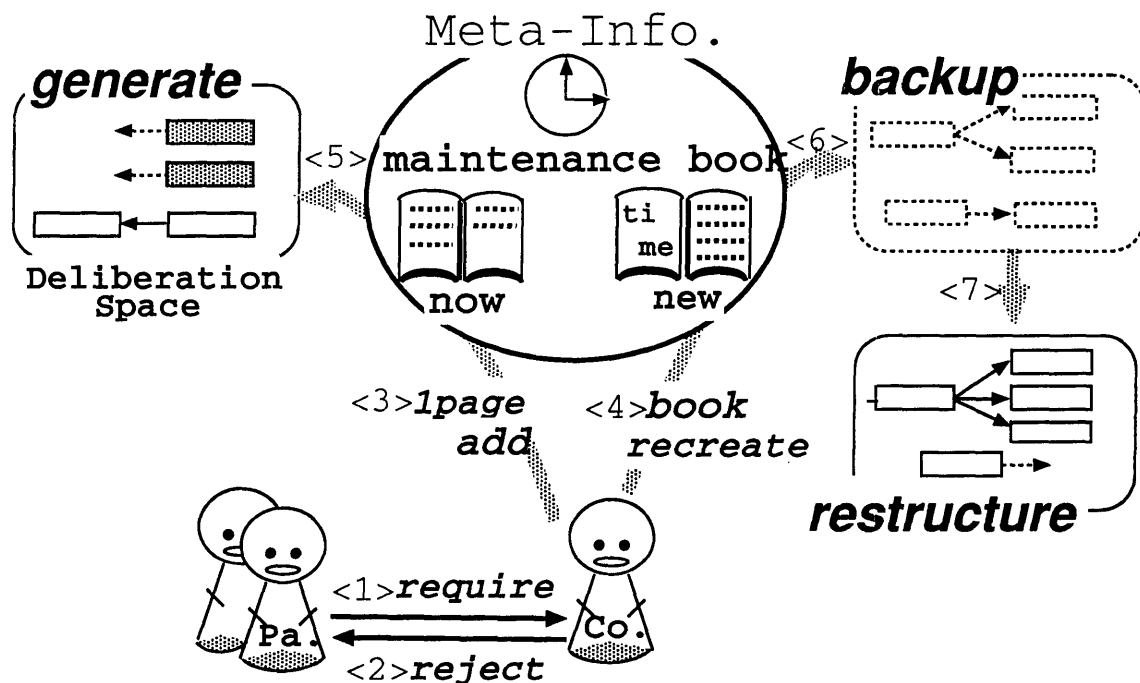


図 6: 動的再構成の仕組み

その手順は、

1. 討議順の変更や新議題が必要になると、コーディネータにその旨を申請（図 6:<1>）。
2. 申請内容について、承認の可否決定（コーディネータ側）（図 6:<2>）。承認すると、
 - 新議題の場合、新しい管理頁を作成し、管理ブックに組み込む（図 6:<3>）。
 - 変更／削除の場合、管理ブックの新版作成。この新版の施行時刻（動的再構成の実施時刻）も付与（図 6:<4>）。
3. メタ情報側では、
 - 新議題の場合、起動時間になった討議プロセスを生成（図 6:<5>）。
 - 変更／削除の場合、動的再構成の実施時刻になると、
 - (a) 討議プロセス群をストップさせ、
 - (b) 管理ブック旧版を含む討議空間情報のバックアップをとり（図 6:<6>）、
 - (c) 管理ブックを旧版から新版に移行し、
 - (d) 討議空間を再スタート（図 6:<7>）。

この動的再構成により、現状の問題により即した討議の維持が可能となる。

ところで、討議の結果がもたらす副産物により、人間の脳に存在する関連情報や知識に関する再構成が起きる場合がある。これは人間内の情報の再認識／再思考の問題であり、討議プロセスの再構成とは直接的には関係ないが、再思考の結果による議題の組み換えをメタ情報に反映することにより、討議プロセス群の動的再構成へと繋がる。

4 議事録データ群と議題の蒸し返し

議事録データ群では、最初に討議の終わった子プロセスの〈議題名、プロセスID、討議期間、同一議題の通し番号、処理結果〉の一组が葉として作成され、さらに自分の親となるプロセスの仮ノードも作成して、これに接続される。親プロセスの討議も終了し議事録が格納されると、仮ノードは親プロセスのデータに置き換えられる。

ところで、会議を行なっていると、同じ議題に対する再討議をどうするかという蒸し返しの問題が発生してくる。これに関しては筆者らは、図5の最後のステップ（議事録格納）により討議プロセスの寿命は尽き、議事録データとしての寿命がスタートすると考える。

すなわち、議題の蒸し返しとは、同一議題に対する新しい討議プロセスのスタートであり、必要があれば、元の議題に対する議事録結果を参照することで再討議を進めてゆく。

この同一議題に関する元討議プロセスと新討議プロセスの参照関係も、メタ情報内の管理ブックにより定義付けされる（管理頁の項11）。

議題の蒸し返しの管理法について、簡単な例を用いて説明する。

まず、図7の状態に至るまでの経過として、Pid 1は、Pid 1.1と1.2の子プロセスを持ち、さらにPid 1.2は3つのプロセスに詳細化されていたとする。この時、6枚の管理頁が作成されており、親討議プロセスが1つ、子プロセスが2つ、孫プロセスが3つ走っている。その後、Pid 1.2に関する討議が全て終り、議事録データ群に格納される。終了した管理頁4枚分は、現行管理ブックから外され、メタ情報の別の場所で保管される。

後日、Pid 1.2の議題について蒸し返しが生じたとする（図7）。

この再討議の手順とは、

1. 会議参加者側で再討議が必要になると、コーディネータにその旨を申請。
2. 申請内容について、終了した管理頁の中から元議題に関する情報を検索し、承認の可否を決定（コーディネータ側）
3. 承認すると、その議題に関して新しい管理頁（図7:Pid 1.3）を作成し、さらに同一議題の通し番号（図7:bbb0）と元議題のプロセスID（図7:1.2）も書き加える。
4. 起動時間になり、生成された討議プロセス（図7:Pid 1.3）は、議事録データ群中の元プロセス（図7:Pid 1.2）の議事録を参照しながら討議を進めて行く。
5. 討議終了後は、同一議題の通し番号がカウントアップ（この場合、bbb1の値）され、管理頁（図7斜線部）に更新される。

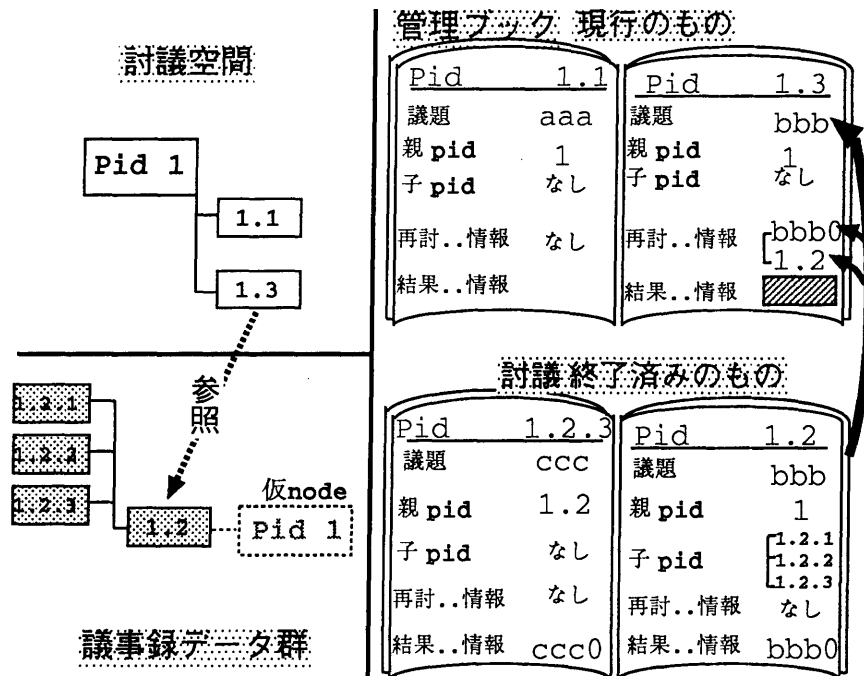


図 7: 管理頁の例示：再討議の場合

5 調整支援

本節では前節までに得られた概念モデルをもとに、プロジェクトの協調状態の維持を支援する方法について部分的ではあるが考察をおこなった。

5.1 閉塞とは

活動ベクトルのずれを生じる原因の中には、プロジェクト活動の作業状態の維持を極端に阻み、自然には除去されにくい要因があり、これを閉塞要因と呼ぶ。閉塞要因によって引き起こされた未解決状態を閉塞状態と定義する。この状態を放置しておけばプロジェクト活動は停止してしまう。そこで、コーディネータによる調整支援を施し、閉塞状態からの脱却を計る。メイリングリストを利用した会議事例においては、

- 外部／内部事象が長時間未到着状態。
- 討議プロセスの処理中断、中止状態。
- 無駄な詳細化により、子プロセスからの処理結果が未回収状態。

などの閉塞状態が考えられる。これらへの対応策としては、討議プロセスは知能を持たないため、現在置かれている状態が閉塞状態か否かを判断する能力はない。よって何らかの基準を設けて、その基準を過ぎた時点で、コーディネータに通知する必要がある。

まず、通知基準として会議の物的資源である「時間」の消費が考えられる。具体的には、管理頁の討議予定時間のチェックをメタ情報が行なうことで実現する。また、別の通知基準として、一つの議題に対する詳細化のレベル数が考えられる。ただし、この基準は必ずしも閉塞状態と直結する概念ではないので、今後の検討が必要である。

6 まとめ

本論文では、プロジェクトメンバ間で生じる非同期型のコミュニケーションを2つの状態に切り分け、動的状態を討議プロセス群、静的状態を議事録データ群としてモデル化し、さらに討議プロセス群がおりなす討議空間を詳細化、継続等の関係を用いて構造化した（詳細は文献[6]を参照のこと）。そして、この討議空間において時間に関して発生する閉塞状態の検出は可能である。

この概念モデルをさらに充実させるためには、議事録データ群の具体的なデータ構造についての考察が必要である。さらに、モデルに従って作成されるプロトタイプ（試作品）によりその有効性を確認するようなアプローチが必要である。概念モデルが充実した時点で、プロトタイプを作成し、実質的な評価を行なう予定である。

最後にシステム化にあたって留意すべき点および研究の方向性についてまとめる。

- 討議進捗情報の共有

討議空間や討議の進捗状況を視覚化し、そのイメージを共有することで、討議進捗に関する同一認識や臨場感が持てるようにしたい。非同期の会議と言えども、意見交換のタイミングがずれているだけであり、会議存続中はリアルタイムに討議関連の情報が得られてもいいはずである。この視覚化により、常にリアルタイムな仮想討議空間を供給することを目指したい。

また、進捗情報の共有により、メンバーの会議への参加意識の向上も計れるのではないかと推察する。この全体管理がうまく行くようになれば、コーディネータの特権的な操作（強制終了や議題の認可 etc.）は、しだいに必要性が減ってくると推察できる。

- コミュニケーションの対象範囲

非同期型コミュニケーションの扱いが可能であれば、意見交換のインターバルが短くなった形態であるリアルタイム型コミュニケーションも、システムに組み込み可能であると考ええる。そこで、同期／非同期のコミュニケーション手段を状況に応じて使い分けることが可能になるようなシステムを目指したい。

それは、お互いの時間が許す限りはリアルタイムでお互いの顔を見ながら会話した方が、目の動きや口調、顔の表情など微妙なニュアンスが伝わり易いからである。そして、時間的に制約のあるキーパーソンは、非同期コミュニケーションツールを用いて適切なコメントを与えるという具合である。同期／非同期のコミュニケーション手段を状況に応じて使い分けることが可能になれば、たとえば、ソフトウェア分散開発の様々な側面で生じるコミュニケーションの型との対応が容易にとれるようになる。

- グループウェアベースによる討議情報統括管理

リアルタイム／非リアルタイム型のいずれのコミュニケーションの場合も、会議全体

の進捗状況、また自分の参加している討議の状態やさらには自分の参加していない他の討議に関する情報を把握することが、円滑に会議を進めてゆくキーポイントとなる。これらの討議進捗情報やその最終結果を管理し、プロジェクト活動のキーポイントとなる協調や調整という概念を活かすようなグループウェアベースを開発することが最終的な目標である（図8）。

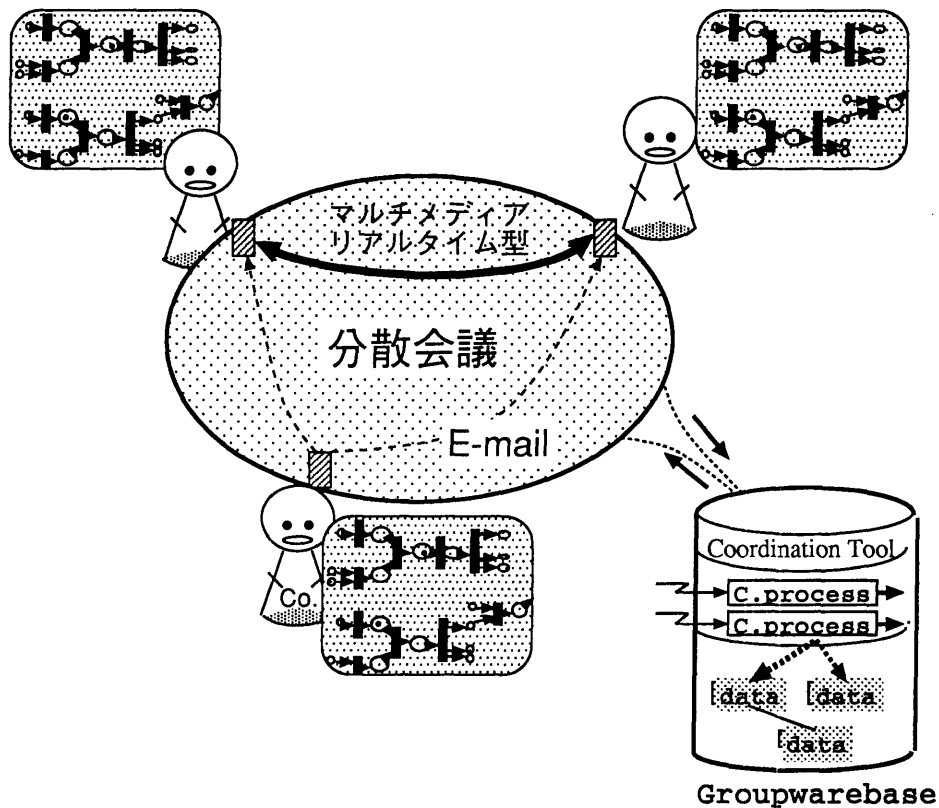


図 8: Groupwarebase を土台とする分散型会議支援

この研究をさらに発展させることで、時間という壁あるいは地理的に離れているという疎外感を感じさせない討議の世界を構築したい。そして、このようなアプローチを通じ、分散開発環境下における開発チーム構成員間の協調活動を支援するための基礎理論へと発展させる必要がある [7]。

参考文献

- [1] 門脇 千恵 , 落水 浩一郎 : 先取り実行や遅延回復機能を有するソフトウェアプロセスの実行制御方式, 日本ソフトウェア科学会 第1回ソフトウェアプロセス研究会資料 SP93-1-7, pp.61-65 (1993)
- [2] Jeff Conklin and Michael L. Begeman : gIBIS:A Hypertext Tool for Exploratory Policy Discussion, CSCW '88 , pp.140-152 (1988)
- [3] グループワーク支援システムの研究開発報告書 C S C W, 財団法人日本情報処理開発協会,5 章 (1993)
- [4] Thomas W.Malone and Kevin Crowston : Toward an Interdisciplinary Theory of Coordination, MIT Technology Paper, (1992)
- [5] Roger C.Schank : ダイナミック・メモリ, 近代科学社, (1988)
- [6] 門脇 千恵 : グループ活動の討議プロセスによるモデル化と調整支援への応用, 北陸先端科学技術大学院大学 情報科学研究科 修士論文, (1994)
- [7] 門脇 千恵 , 落水 浩一郎 : 非同期分散型会議の事象駆動型討議プロセスによるモデル化と調整支援への応用, 情報処理学会 第96回ソフトウェア工学研究会資料 96-25, pp.193-200 (1994)