グループ演算を用いた協調作業の可視性とフローの制御*

3M－7

菊田 裕次　岩井原 瑞穂
九州大学 大学院 システム情報科学研究科 情報工学専攻

1　はじめに

コンピュータ及びネットワークを介して行なわれる議論の手段として、メッセージ駆動型の NetNews や Mailing List、チャット型の IRC といった多くのアプリケーションが存在する.

前者はあるメッセージに対するフォローメッセージの投稿によって議論が進むためフォロー関係が明確にわかるが、あるひとつのメッセージに複数のフォローが行なわると、フォローメッセージがどのメッセージに対するフォローであるかというフォロー関係を木構造で表現した場合、その枝が複雑に分裂してしまう. さらに、それぞれの枝の異なる議論の流れで同様の議論が起こってしまうことがあり、多くの無駄が生じる.

後者は、ある特定の参加者間で議論が行なわれ、その参加者グループの会話は時間軸に沿って並べられる. 発言は直列に並べられるため、議論の分散は最小限に押えることができるので発言のフォロー関係を意識しない程度の小さな議論には有効である. しかし、フォロー関係はメンバが文脈によって判断しなければならないなど、議論の流れを把握することが困難である.

また、対話構造の可視化を目的とした研究 [1] がある. これはメンバの発言それぞれを中心として木構造を表現しているが、実際は問題点や提案など、議論の内容に着目することも重要である. なぜなら、後から議論の流れを閲覧する際には、誰が発言したか、もしくは発言の内容が何かより議論内容がどのように推移したかが問題となるからである.

このように議論の流れに着目すると、議論の構造の変化に従ってメンバのグループ構造を変更させていくことが必要となってくる.

これらを考慮して、我々はグループディスカッション支援システム "GODSS(group operating discussion support system)" を構築しようとしている. "GODSS" の目標とするものは、次のとおりである.

- 議論の内容による議論構造の表現
- 議論の分散の抑制
- 議論の可視性の制御
- グループの入れ子構造を利用した議論の支援

本稿では、メッセージ駆動型及びチャット型コミュニケーションの両方の利点を活かしながらそれぞれの欠点を補うグループディスカッション支援システムについて検討する.

2　議論の構成要素

議論の構成要素として、次の要素を定義する.

2.1　メンバ

メンバは、議論の参加者である. 次のように分類する. すなわち、グループの指導者で、グループに対する操作を行なうことができるリーダ、一般のメンバ、そして発言権を持たないウォッチャである. ウォッチャは、議論の内容を閲覧することはできるが、発言することはできない.

2.2　グループ

グループは、議論を共同で行なう一人以上のメンバの集まりである. グループ内にグループを持つことにより、入れ子構造となる. このとき、親グループ内に子グループがあるという. 最上位のグループをルートグループという.

2.3　メッセージ

メッセージは、問題提起や結論など、議論の内容の情報を持つオブジェクトである. 親グループは議決すべき事項をメッセージとして子グループに渡し、子グループによって議論された結果は再びメッセージとして親グループに返される. このメッセージの集合によって議論の流れを把握することができる.

3　GODSS の概要

GODSS の機能概要を以下に示す. GODSS は、メッセージ駆動型とチャット型それぞれの利点を採用し、それぞれの欠点を補完している.

3.1　グループ制御機能

従来のメッセージ駆動型のアプリケーションでは、問題提起のメッセージに対してさらなる問題提起や解決方法を示すフォローメッセージがメンバによって次々に返され、このフォロー関係を木構造として表現する. しかし、これはある一人の発言に対する別の一人の発言が連なるという構造であり、表現される木構造は意味的な並び方をしていない.

議論の内容に基づいてこの構造を表現するということは、問題に対しては解決策の検討が行なわれ、その結果としての解決策がフォローとして連結されるということである.

このとき挙げられた解決策の案それぞれに対して妥当性の検討が行なわれる場合もある. 解決策の検討を行なう議論グループ（親グループ）には、解決策の案の数だけの子グループが作成され、その子グループの中で議論が行なわれ、その結果得られたメッセージを親グループに返し、それをもとにさらに議論を進めることができる.

3.2　チャット機能

先に述べたグループ制御機能を用いることにより、議論を細分化することができる. こうして小さな単位に分割したとき、フォロー関係を意識せずに議論を進めることが可能となると考えられる. そこで、小グループでの意思決定に適した簡易なコミュニケーション手段としてチャット機能を採用し、細分化したグループ内での議論に用いる. これを用いて行なった議論の結果をそのグループの結論とし、親グループにメッセージとして返す.

3.3　可視性制御機能

議論が進行中であり、結果が出ていないときには、その議論に参加していないメンバには議論の進行状況を見せたくないという要求が考えられる. また、議論には参加させないが、進行状況を見せることは許可するという場合もある. このような要求を満たすために、グループ毎に可視性を制御する機能を有する. この可視性は、各グループのリーダによって操作することができる. グループでの議論を閲覧可能にするかどうかは、リーダの意思もしくはグループ全体の意思により決定する.

3.4　フロー表示機能

議論の流れを表示する機能である. 図 1のように表示する.

---

* "View and flow control in a collaborative work with group operation", Yuji Kikuta and Mizuho Iwaihara, Graduate School of Information Science and Electrical Engineering, Kyushu University

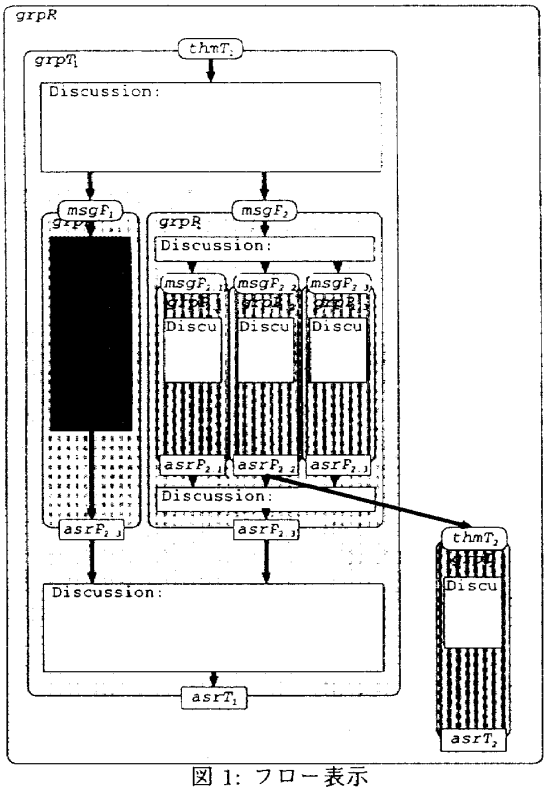

図 1: フロー表示

入れ子構造になっている部分は、まず親グループが表示される．子グループの情報を得たいときはその親グループを指定し、表示を指示する．

4 GODSS を用いた議論の進行

実際に GODSS を利用して議論を行なう例について述べる．

1.ルートグループ ($grpR$) に決議すべきテーマ ($thmT_1$) が与えられる．

2.$thmT_1$に対処するためのグループ ($grpT_1$) が作成され、$grpT_1$にメンバが招集される．

3.$grpT_1$のリーダとメンバにより、チャット機能を用いた議論が開始される．

4.この議論では、$thmT_1$には問題点が二つ (P_1, P_2) あり、それぞれをまず解決する必要があるということになる．

5.リーダは、$grpR$内に子グループ ($grpP_1$, $grpP_2$) を作成し、$grpR$のメンバから$grpP_1$及び$grpP_2$にメンバを割り振る．ある一人のメンバが複数のグループに所属することは許可される．

6.$grpR$から$grpP_1$, $grpP_2$にメッセージとして$msgP_1$, $msgP_2$を渡し、子グループは議論を始める．

7.$grpP_2$は議論を公開するが、$grpP_1$はその問題の性質から議論の内容を公表しないことにした．

8.$grpP_2$は問題 P2 を解決するために、さらに 3 つの問題 $P_{2.1}$, $P_{2.2}$, $P_{2.3}$を解決する必要があると判断し、子グループとして$grpP_{2.1}$, $grpP_{2.2}$, $grpP_{2.3}$を作成し、$P_{2.1}$, $P_{2.2}$, $P_{2.3}$をメッセージ$msgP_{2.1}$, $msgP_{2.2}$, $msgP_{2.3}$として渡す．

9.$grpP_{2.1}$, $grpP_{2.2}$, $grpP_{2.3}$は解決策としてそれぞれメッセージ$asrP_{2.1}$, $asrP_{2.2}$, $asrP_{2.3}$を親グループである$grpP_2$に返す．このうち、$asrP_{2.2}$については、大きなテーマ$thmT_2$として扱うことにし、$grpP_{2.2}$は親グループ$grpP_2$から切り離され、独立したグループとしてさらに議論を進めることにする．

10.$grpP_2$は$asrP_{2.1}$, $asrP_{2.2}$, $asrP_{2.3}$をもとに問題 P2 の解決策を議論し、結果をメッセージ$asrP_2$として親グループである$grpT_1$に返す．同様に$grpP_1$からもメッセージ aP1 が返される．

11.$asrP_1$, $asrP_2$をもとに、$grpT_1$において議論が行なわれ、テーマ$thmT_1$の決議が終了する．

12.その結果をメッセージ$asrT_1$として$grpR$に返す．

グループ間を往来したメッセージを集めてみると、図 2のようになる．すなわち、議論の内容の遷移はメッセージの集合によって明確に理解することができる．

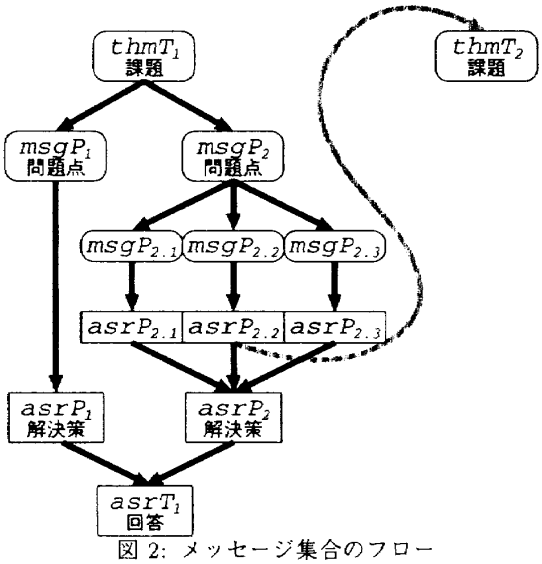

図 2: メッセージ集合のフロー

5 おわりに

本稿では、メッセージ駆動型及びチャット型コミュニケーションの両方の利点を活かしながらそれぞれの欠点を補うグループディスカッション支援システムのモデルについて述べた．入れ子構造のグループ演算を導入することにより、可視性の制御を可能にし、フロー表現において議論の流れをより明確にすることができる．

本システムでは、利用者の議論を支援するのみであるが、今後はさらに一般的な協調作業に応用し、オブジェクトに対する作業を取り扱えるようなモデルを構築する予定である．

参考文献

[1]久寿居 大, 石黒 義英, 宮下 敏昭,“電子掲示版システムにおける対話構造の可視化”, 情報処理学会第 55 回全国大会講演論文集, 4Y-02, 1997