

協調作業における作業権限及び作業状態の管理

杉山 圭司 横田 裕介 垂水 浩幸 上林 彌彦

京都大学情報学研究科

1. はじめに

計算機による分散環境下での協調作業の研究として我々の研究室ではデータベースの技術を用いた、協調作業基盤システム VIEW Media の構築を行っている。これは協調作業を行うアプリケーションに必要な機能を提供するシステムであり、このシステムの機能を用いて様々な種類のアプリケーションを構築することができる。

VIEW Media の機能の有効性を検証するためにダイバートを行うアプリケーションを実装し、実験を行った。その結果、このシステムが提供する非WYSIWIS 機能[1]の有効性を確認することができた。しかし、協調作業における作業権限の管理機構の不足、作業状態の遷移機能への要求などが見つかった。

本稿では、協調作業において、各利用者の作業権限を設定・管理し、自動的に作業権限、作業状態を遷移させる機能の必要性を述べ、それを実現するために作業状態オブジェクトを VIEW Media に組み込むことを提案する。

2. 協調作業支援基盤システム VIEW Media

VIEW Media は基本構成要素としてハイパーメディア資料、利用者、環境を持ち、それぞれをオブジェクトとして扱う。環境は各基本構成要素、及びチャットなど、利用者間のコミュニケーションを可能にする装置を内部に含むことができ、各オブジェクトを管理することが可能である。また、環境オブジェクトを内部に含むこともでき、環境の階層構造を形成することが可能になる。

環境は協調作業に必要なオブジェクトを内包することから部屋概念[2]のように作業空間として機能する。

資料の共有や個別化、各利用者の権限のコントロールは環境を用いて行われる。例えば同じ共有資料でも環境による個別化を行うことで、環境が異なると資料の一部の見え方が異なるといった処理を行うことが可能になる。環境の階層化がなされている場合、上位環境での変更を下位環境に波及させることも可能である。また、環境による権限の管理によって、同じ環境、もしくは異なる環境に属する利用者間のチャットの利用を制限するといったこともできる。

Authorization and State Control Mechanisms for Cooperative Work
Keiji SUGIYAMA, Yusuke YOKOTA,
Hiroyuki TARUMI, and Yahiko KAMBAYASHI
Graduate School of Informatics, Kyoto University

環境による資料および利用者の管理と、協調作業の階層構造化を組み合わせることで複雑な環境下での協調作業への支援を実現している。

3. アプリケーションの実装と実験からの分析

VIEW Media の提供機能の有効性を確認するために、システム上に、分散環境でダイバートを行うアプリケーションを実装し、それを用いた評価実験を行った。ダイバートを実装したのは少人数での作業、グループ分けがされる会議といった性質を持ち、評価実験を行いやすいためであるが、ダイバートは一般の会議の性質を持っており、このアプリケーションを用いてほとんどの会議を行うことが可能である。

3.1 ダイバートと評価実験

ダイバートとはある議題について、肯定派、否定派に別れて議論を行い、その議論の優劣を議長や審判が判断する知的競技である。図1に環境を用いたダイバートの作業構造の例を示す。

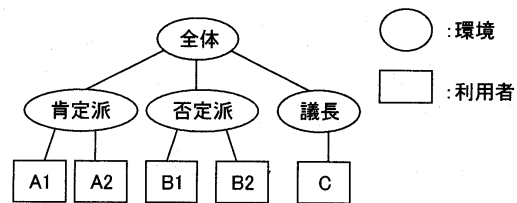


図1 ダイバートの作業構造

ダイバートは複数の作業の段階を持ち、全員が参加する段階と各グループ内の作業を行う段階などがある。この実験の結果、VIEW Media が環境を用いて実現している特徴的な機能の有効性を確認することができた[1]。しかし、ダイバートでの作業権限管理を行うには VIEW Media の提供機能では不足であることが判明した。

3.2 複数の作業空間に対応する作業権限管理機構

ダイバートでは次のような状況が生じる。

- ・ 全員が参加する作業時には、グループ内での作業をすることはできない。
- ・ 作戦時間では全体での作業を行うことはできない
- ・ 議長は例外的に、いつでも全体と議長環境での作業を行うことができる。

このように、協調作業全体として適切な作業状態を実現するためには、作業の段階にあわせて、全体環境・各グループ環境などの複数の環境（作業空間）での作業権限

を適切に設定しなくてはならない。これを実現するには作業の管理者が作業の段階に合うように、全ての作業空間（環境）での各利用者の権限の設定を行うことが必要になる。ディベートでは作業がどのように進行するのかがあらかじめ判明しており、作業の進行に合わせて各環境での利用者の作業権限を自動的に変更する機能を実装した。この機能は管理者の手間を軽減するだけでなく、作業権限の設定ミスをなくす効果もある。

作業の構造が複雑になり、より多くの作業空間での権限の管理を必要とする場合、このような機能の重要性は増大する。ディベートに限らず、作業の一定の段階で、誰がどのような作業空間でどのような作業権限を持つかが予測できている例は少なくない。よって、作業の権限の設定情報が得られている場合に、複数の作業空間での作業権限の変更を自動的に行うための機能の有効性は普遍的な性質を持つと考えられる。

4. 状態遷移機構

以下では、状態オブジェクトという概念を導入し、複数の作業空間の作業権限を自動的に変更する機能の実現について述べる。また、協調作業の状態を遷移させ、作業権限を設定する際に条件判定を行い、遷移結果や作業権限の設定結果を変更する機能についても述べる。

4.1. 状態オブジェクト

自動的に各利用者の作業権限を設定するためには各作業空間の権限設定の情報などを記述し、管理しておく必要がある。この機能を実現するために状態オブジェクトを用意し、このオブジェクト内に必要な情報を記述する。このオブジェクトは次のような情報を持つ。

・ 作業空間の構造

協調作業で用いられる環境、環境間の親子関係、環境に属する利用者の情報を格納する。ディベートの例では全体での議論を行う作業状態では、全体環境、肯定側環境、否定側環境、議長環境が用いられること、全体環境は他の環境の親子環境であることの情報と、議長環境に属する利用者の情報が格納される。

・ 各作業空間での作業権限の設定情報

各々の作業空間内での作業権限の設定情報が与えられる。ディベートの全体での議論の段階では、発言する利用者に対して全体環境での発言権・資料使用权などが設定される。同時に各グループでの作業ができないような作業権限の設定情報が記述されている。また議長には全ての環境において管理権限が設定されている。

・ 他の状態からの遷移条件

この状態に遷移するための制約条件が記述される。ディベートの例では、一連の作業の流れは一定であり肯定側立論の状態に遷移できるのは「現在の状態が一回目の作戦時間」に限られている。これにより間違った状態への

遷移を防ぐことが可能になる。

・ 遷移先の状態及び遷移条件の指定

どのような状態に遷移することができるのかの指定を行う。ディベートでは各段階で、遷移先は一つしかないが、遷移先が複数考えられるような協調作業も考えられる。例えば、教育をおこなうアプリケーションにおいて、授業状態からの遷移先として、「学生の平均点が60点以上であれば自由行動状態に遷移、それ以外のときは補習授業状態に遷移」といった指定方法も考えられる。このような場合に遷移先を複数指定できること、遷移先を条件判定によって遷移先を変更できることは有効性をもつ。以上が、状態オブジェクトの内容であり、この状態オブジェクトを生成・登録する機能、状態オブジェクトにしたがって、環境間の関係を変更し、環境内の作業権限の設定を変更する機能をVIEW Mediaは基本機能として、新たに提供する。

4.2 条件判定による権限設定

4.1で状態オブジェクトを記述するときに、状態遷移先を条件判定によって選択することを述べたが、これは各環境内の作業権限の指定を行うときにも用いる。環境内部のオブジェクトの状態によって条件判定を行い、環境内の作業権限の設定を変更する。例えば、利用者Aが教師環境内に存在する場合は全体環境の管理権限を利用者Aに与えるが、利用者Aが教師環境に不在の場合は利用者Bに全体環境での管理権限を与える、といった権限の指定が可能になる。

この機能を利用することで、些細な設定の違いがあるだけの状態オブジェクトを生成する必要がなくなる。また、状態オブジェクトを導入することで、「作業状態に応じたシステムの挙動」などを実現することが可能になる。

5. まとめと今後の課題

アプリケーションの実装と実験の経験から、協調作業での利用者の作業権限を設定する機構の重要性を認識し、それを実現するための作業状態の遷移機構について述べた。これにより、多数の作業空間を用いる協調作業において権限管理の手間を軽減することができる。

今後は、定型的な作業を支援するための状態遷移機構と、柔軟な作業構造・作業状態の変更機能の統合をめざしていく必要がある。

参考文献

- [1], Yusuke Yokota, Keiji Sugiyama, Hiroyuki Tarumi, Yahiko Kambayashi.: Evaluation of Dynamic Representation of Roles by Environment Model. Springer, Proc. CODAS'99, pp88-99 (1999)
- [2], Saul Greenberg, Mark Roseman.: Using a Room Metaphor to Ease Transitions in Groupware. Technical Report 98/611/02 (1998)