

仮想会議システム (5) 共有アプリケーション[†]

4E-10

坂入 隆、中島 周、安藤 史郎、椎尾 一郎、小林 真

日本アイ・ビー・エム(株) 東京基礎研究所

1. はじめに

複数の計算機を通信回線で接続し、会議などを行うというシステムの研究や開発が多くの機関で進められている[1]。このような会議システムを開発する上では、アプリケーションプログラムを複数の計算機上で、いかに共有するかという点が重要となる。

我々は、OS/2上で稼働する会議システムを開発してきた。このシステムでは、仮想会議室と呼ぶウィンドウに会議の状態を表示し、利用者が簡単に制御できるようになっている。また、会議制御部がすでに開催中の会議に途中から新たな人を参加させることを許している。新しい参加者の計算機上では、それまでの参加者と同じ状態の共有アプリケーションプログラムを起動する事が可能である。本稿では、共有黒板を例として取り上げ、共有アプリケーションプログラムの実現法を述べる。

2. 共有黒板

会議システムにおいては、グラフィクスエディタを共有化した共有黒板が重要な役割を果たしている。この共有黒板に対し、それぞれの参加者が書き込みを行い、会議を進めていく。共有黒板のような共有アプリケーションプログラムでは、参加者間でデータの不整合が起きないようにするために操作権の管理を行う必要がある。また、途中から会議に参加する人のためには、その人の計算機に共有黒板の状態を転送し、共有黒板の状態を同じにする必要がある。

我々の仮想会議システムでは、LEDA[2]というグラフィクスエディタを拡張し、共有黒板として使っている。このLEDAは、グラフィクスエディタとしての機能を一通り持ち、多くの会議システムで採用している共有黒板よりも高機能となっている。また、LEDAは、別のプロセスと通信を行うという機能があるため、共有黒板として使えるように拡張するのは、容易であった。

3. 構成

仮想会議システムの共有黒板の構成を図1に示す。LEDAでは、外部のプロセスと通信を行うためにプロセス間通信DLLを用意している。このプロセス間通信DLLでは、LEDAの上で利用者が行った操作を外部のプロセスに知らせることと、外部のプロセスからLEDAにコマンドを送ることができる。

仮想会議システムの共有黒板としてLEDAを拡張するために、ツール制御DLLを新たに作成した。このDLLが、仮想会議の会議制御部からの会議イベントに対して、適切な処理を行うことにより、共有黒板を制御する。

4. 動作

この章では、共有黒板のツール制御DLLの働きについて述べる。

4.1 初期処理

仮想会議システムが起動されると、それぞれの共有アプリケーションプログラムのツール制御DLLの初期処理が実行される。共有黒板の初期処理では、次のことを行う。

- (a) 参加者間の通信のためのポートの設定とデータを受信したときに実行するルーチンの登録
 - (b) 黒板のプロセス間通信DLLから送られるデータを受信するためのスレッドの生成
 - (c) 会議イベントに対する処理ルーチンの登録
- (a)で登録したルーチンは、別の参加者の操作を自分の共有黒板に反映させる。(b)で生成したスレッドは、自分の操作を自分以外の参加者の共有黒板に反映させる。

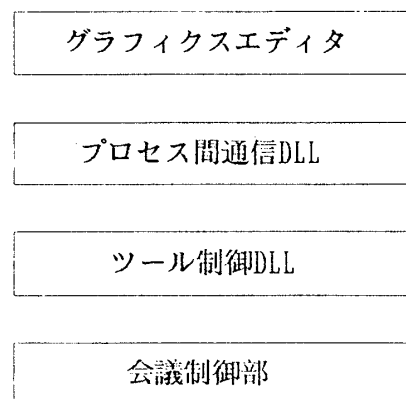


図1: 共有黒板の構成

[†]Virtual Conference System (5) Shared Application
Takashi SAKAIRI, Amane NAKAJIMA, Fumio ANDO, Itiro SHIO, Makoto KOBAYASHI
IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.

図形の生成や移動などの大部分の操作は、単純に対応する黒板にデータを送るだけである。しかし、ファイルを黒板に読み込む場合には、自動的に自分以外の参加者にファイルを転送する。また、ビットマップを黒板に貼り付ける場合には、データ量を減らすために圧縮を行う。

(c) で登録したルーチンでは、共有化や非共有化といった会議イベントに対して適切な処理を行う。処理の具体的な内容は、4.2 節で述べる。

4.2 会議イベントに対する処理

ツール制御 DLL では、会議イベントに対して次のような処理を行う。なお、会議の参加者の中に一人だけ議長という特別の役割を持った人がいる。

TERMINATE

仮想会議システムを終了する時に発生するイベントである。実行中の黒板のプロセスを終了し、ポートの登録を解除し、黒板のプロセス間通信用 DLL から送られるデータを受信するためのスレッドを削除する。

NEW

黒板を生成する時に発生するイベントである。新しい黒板が生成したことを記録する。しかし、この時点では黒板の起動はしない。

OPEN

黒板を開く時に発生するイベントである。黒板のプロセスを起動する。また、このイベントの前に BEGIN_SHARE のイベントを受け取っていれば、直ちに共有化する。

CLOSE

黒板を閉じる時に発生するイベントである。黒板のプロセスを終了する。

DELETE

黒板を削除する時に発生するイベントである。黒板のプロセスが起動中なら終了し、黒板を削除する。

BEGIN_SHARE

黒板を共有化する時に発生するイベントである。すでに黒板のプロセスが起動していれば、共有化する。議長なら、操作権を作成する。

END_SHARE

黒板を非共有化する時に発生するイベントである。すでに黒板のプロセスが起動していれば、非共有化する。議長なら、操作権を削除する。

ENTERING

会議に新たな参加者がこれから加わろうとする時に発生するイベントである。一時的に、黒板に対する入力を禁止する。この入力禁止は、新しい参加者の黒板がそれまでの参加者の黒板と同じ状態になった時点で解除される。

ENTERED

会議に新たな参加者が加わった時に時に発生するイベントである。会議の議長は、現在の黒板の状態をファイルに書き出し、新しい参加者に送る。

LEAVING

会議から参加者がこれから退出しようとする時に発生するイベントである。退出する本人の黒板を非共有化する。また、一時的に、黒板に対する入力を禁止する。

LEAVED

会議から参加者が退出した時に発生するイベントである。黒板の入力禁止を解除する。

FLOOR_REGISTERED

操作権が登録された時に発生するイベントである。操作権識別子を記憶しておく。

FLOOR_UNREGISTERED

操作権の登録が解除された時に発生するイベントである。操作権識別子の記憶を取り消す。

FLOOR_OBTAINED

操作権を得た時に発生するイベントである。黒板に対する入力を可能にする。

FLOOR_RELEASED

操作権を失った時に発生するイベントである。黒板に対する入力を不可能にする。

4. おわりに

本稿では、共有黒板と例として仮想会議システムにおける共有アプリケーションプログラムの実現法について述べた。ツール制御 DLL が、様々な会議イベントに応じた処理をグラフィクスエディタに対して行うことにより共有黒板を実現した。この方法は、共有黒板を新たに作成する方法と比較し、簡単に行うことができた。

黒板の生成、削除、共有化、非共有化、会議への途中参加、会議からの途中退出、操作権の移動などは、仮想会議室を利用者が操作することにより行う。利用者の操作に応じて、仮想会議室は、会議制御部に対し会議イベントを発生させる。共有アプリケーションプログラムは、それぞれの会議イベントが発生したときに行う処理を登録しておく。このような一般的な機構があるために、新たな共有アプリケーションを追加することが可能である。

参考文献

- [1] C. A. Ellis, S. J. Gibbs, and G. L. Rein, "Groupware - Some Issues and Experiences," *Communications of the ACM*, Vol. 34, No. 1, pp. 39-58, Jan. 1991.
- [2] Y. Mima, "A Visual Programming Environment for Programming by Example Abstraction," *IEEE Workshop on Visual Languages*, Oct. 1991.