

4P-10

共有アプリケーションにおける クリップボードの機構

坂入 隆

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、アプリケーションを共有して会議を行うシステムが多く開発されている。このように共有されたアプリケーションを共有アプリケーションと呼ぶ。

共有アプリケーションにおいても、通常のアプリケーションで行われている「複写」や「貼付け」といったクリップボード関連の操作が必要となる。クリップボード関連の操作では、ビットマップなどの大きなデータを扱うことが多いので、通信量を少なくする工夫が重要である。本稿では、共有アプリケーションにおける効果的なクリップボードの機構について述べる。

2 従来の方法

共有アプリケーションのクリップボードの機構として、次のような方法がある。

2.1 内部クリップボード

内部クリップボードとは、アプリケーション自身が管理するクリップボードのことである。この方法では、他のアプリケーションにクリップボードを破壊されない。そのため、クリップボード操作のときに、データ自体を送る必要はない。「複写」や「貼付け」といったコマンドのみを接続相手の計算機に送り、それぞれの計算機で複写や貼付けの操作を行えば良い。このため、通信量を減らす事ができる。しかし、他のアプリケーションとの間でデータを交換することが出来ないという欠点がある。

2.2 システムクリップボード

システムクリップボードとは、ウィンドウシステムが管理するクリップボードのことである。この方法では、他のアプリケーションとの間でデータを交換する

ことが出来る。しかし、接続されているそれぞれの計算機で独立にシステムクリップボードが変更されるので、「複写」や「貼付け」といったコマンドのみを接続相手の計算機に送るという方法は採用出来ない。このため、貼付けの操作の時点でデータを送る必要があり、通信量が多くなるという欠点がある[1]。

2.3 内部クリップボードとシステムクリップボードの単純な組合せ

内部クリップボードを使う方法とシステムクリップボードを使う方法の両方を実現し、利用者にメニューなどから明示的にどちらのクリップボードを使うかを選ばせるということも可能である。この方法は、アプリケーション内部クリップボードを使う方法の通信量が少ないという利点と、システムクリップボードを使う方法の他のアプリケーションとの間でデータを交換出来るという利点の両方を持つ。しかし、利用者がどちらのクリップボードを使うかということを示的に指定する必要があり、使いやすさに問題がある。

2.4 共有システムクリップボード

共有システムクリップボードとは、ウィンドウシステムが管理するクリップボードを共有するものである。この方法では、共有アプリケーションは、「複写」や「貼付け」といったコマンドのみを接続相手の計算機に送ることにより、クリップボード操作を実現できる。しかし、接続された計算機のシステムクリップボードの状態を一致させるために、システムクリップボードが変更されるたびに共有システムクリップボードを管理するプログラムがデータを送る必要がある。このため、この方法が最も通信量が多くなる。

3 課題

前章で述べた従来の方法にある問題を解決するのが本稿の目的である。つまり、次の3点の要求を満たすクリップボードの機構を開発することである。

1. 同じアプリケーション内でのクリップボードの操作では、コマンドのやりとりだけを行い通信量を少なくする。

A Clipboard Mechanism for Shared Applications

Takashi SAKAIRI

IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd.

1623-14 Shimotsuruma, Yamato, Kanagawa 242, Japan

2. 他のアプリケーションとの間でのデータの交換を可能にする。
3. 内部クリップボードを使うのか、システムクリップボードを使うのかというような利用者にとって煩わしい指定を不要にする。

4 解決法

本稿での解決法は、内部クリップボードとシステムクリップボードの両方を使う。ただし、どちらのクリップボードを使うのかを利用者に指定させずに、システムが自動的に判別する。そのために、システムクリップボードにシステムクリップボードの中身と内部クリップボードの中身が同じであることを示す印を複写する。

次の処理を行う。fMineというのは、内部クリップボードに自分が複写したデータが含まれているかどうかを判断するためのフラグである。

- アプリケーションを起動したとき、
 1. fMineをfalseに設定する。
- 利用者が「複写」を指示したとき、
 1. 内部クリップボードにデータを複写する。
 2. システムクリップボードを空にする。
 3. システムクリップボードに、内部クリップボードに同じデータがあるという目印を複写する。このとき、アプリケーションのインスタンスを区別する値も複写する。
 4. システムクリップボードにデータ自体を複写する。
 5. fMineをtrueに設定する。
 6. 「複写」というコマンドを別の計算機に送信する。
- 利用者が「貼付け」を指示したとき、
 - システムクリップボードに、内部クリップボードに同じデータがあるという目印があり、アプリケーションのインスタンスを区別する値が自分と同じであり、fMineがtrueのとき、
 1. 内部クリップボードから貼付ける。
 2. 「貼付け」というコマンドを別の計算機に送信する。
 - 上の条件に当てはまらないとき、
 1. システムクリップボードから貼付ける。
 2. 貼付けたデータを別の計算機に送信する。
- 別の計算機から「複写」というコマンドを受信したとき、
 1. 内部クリップボードに複写する。
 2. fMineをfalseに設定する。

- 別の計算機から「貼付け」というコマンドを受信したとき、

1. 内部クリップボードから貼付ける。

- 別の計算機から貼付けるデータを受信したとき、

1. 受信したデータを貼付ける。

ここでは、1つの内部クリップボードを適切に使用するためにfMineというフラグを用いたが、それぞれの利用者ごとに内部クリップボードを用意することも可能である。その場合には、「複写」や「貼付け」のコマンドの処理を実行した利用者によって、どの内部クリップボードを使うかを区別する。

5 実装

グラフィクスエディタを複数の利用者が編集できるようにした共有黒板に、この共有アプリケーション用のクリップボードの機構を実現した。共有黒板では、直線、円、矩形、テキストなどのデータ量の少ない図形と、ビットマップなどのデータ量の多い図形の両方を扱う。

会議システム開発のプラットフォームとしては、Lakes[2]を採用した。Lakesは、OS/2、Windows 3.1及びAIXという複数のOSで稼働する。Lakesでは、2者間だけではなく3者間以上での会議が可能であり、共有黒板も3者間以上で使うことができる。

プログラミング言語は、C++を用いた。また、ユーザインタフェースの部分は、複数のOS上で共通に使えるクラスライブラリを用いたため、1つのソースコードから3つのOSで稼働する共有黒板を作成した。ただし、AIX用のこのクラスライブラリがシステムクリップボードを実現していなかったため、AIX版の共有黒板では本稿で説明した機構を実装していない。

6 おわりに

共有アプリケーションにおける効果的なクリップボードの機構を開発し、この機構を共有黒板に実装することによって有効性を確認した。この機構は共有黒板だけではなく、多くの種類の共有アプリケーションに対して適用することができる。

参考文献

- [1] 田中俊昭, 山田満, 羽鳥好律: “構造化マルチメディア文書を用いた協同編集システムColleague”, 情報処理学会論文誌, Vol. 36, No. 6, pp. 1310-1321 (1995).
- [2] The IBM Lakes Team: “IBM Lakes - An Architecture for Collaborative Networking”, R. Morgan Publishing, (1994).