

協調ハイパーメディアシステムにおける共有オブジェクト更新伝播方法

5W-4

塩谷素彦、古林勝紀

NTTデータ通信株式会社

1 はじめに

我々はグループウェアの基盤となる、複数ユーザ共同利用型協調ハイパーメディアシステム(Cooperative HyperMedia System:以下CHMS)を開発中である。CHMSでは複数のプロセスで共有されるオブジェクトの最新の内容を画面に表示する。そのため、オブジェクトの更新時に、更新伝播を行う必要がある。その際、即時に各プロセスに伝播させると、更新が失敗した場合に元の状態に戻すことが大きなオーバーヘッドとなる。本論文では、このオーバーヘッドを減少させ、OODB transaction 管理との整合性を保つための効率のよい更新伝播方法を説明する。

2. CHMSの特徴

2.1. 画面への表示

CHMSは、データ部分とその表現方法を別個に管理している。従って、同じデータを複数箇所に異なる表現形式で画面に表示することが出来る。[1]

2.2. オブジェクトの共有

CHMSでは、共有可能なオブジェクトはパーシステントとし、OODB中に格納している。各ユーザ毎にあるプロセスは、利用するオブジェクトをメモリにロードする。同期型協調作業でオブジェクトを共有するためには、各メモリ中の同一オブジェクトは内容が同一であり、画面もその内容を反映していなければならない。

共有されているオブジェクトを用いて複数ユーザが同期型作業をする場合、同時に行なわれた異なる更新が同じオブジェクトを変更しようとする場合がある。このような更新をデータベースに対して行うとトランザクションのアポートにより、ロールバックする必要が生じる。

3. 更新伝播

CHMSにおける更新伝播には

- 1) データの無矛盾性
- 2) 即時性

が要求される。

データの無矛盾性とは、すべてのユーザプロセスの管理するメモリ中とOODB内で同じオブジェクトの内容が等しいこと、オブジェクト間に定義された制約が満たされていることをいう。

即時性とは、更新が成功した時点で、ユーザが明示的に操作することなく、すみやかに伝播が行われることを指す。

つまり、あるプロセスがオブジェクトの内容を更新した場合、その内容はデータベースと他のプロセスに伝わらなくてはならない。

3.1. 従来方法

従来一般的な方法のうち、ユーザの操作によらずに、更新伝播を行なうものは、データの更新が起こった時、直ちにその更新結果をそのデータが表示されているすべての画面に反映していた。しかし、この方法では更新が失敗した場合、更新の取消に大きなオーバーヘッドがかかる。

また、従来方法では具体的な画像情報をやりとりするものもあるが、CHMSでは個々のプロセスで表示画像は異なるので、そのやりとりは無意味である。

3.2. CHMSでの方法

すべての更新は、OODBにコミットすることで管理し、複数の更新間の無矛盾性を保証する。従って、変更を行ったプロセスはまず、OODBに変更を通知する。しかし、データベースには、他のプロセスに変更後の内容を通知する機構がないので、OODBにコミットしただけでは、更新を行なったプロセスのメモリ中の内容と他のプロセスのメモリの内容が同一にはならない。

そこで、我々の方法では、すべてのオブジェクトにIDをつけている。オブジェクトを変更した場合は、まず、OODBへのコミットをかけ、終了するまで変更したオブジェクトのIDをプロセス内でキューイングし、他のプロセスには通知しない。コミットが成

功した時点でオブジェクトIDを他のプロセスに通知する。(図1)コミットが失敗した場合はキューイングしていた情報を破棄するだけなので、オーバーヘッドがかからない。更新通知を受け取った各プロセスは、該当オブジェクトをOODBから再ロードし、画面を更新する。

プロセス間では、オブジェクトIDのみを通信するので、通信量が少ない。また、ユーザの明示的な操作無しに各プロセスが管理するオブジェクトは必ずOODB中の状態と同一になるので、即時性と各プロセス間の無矛盾が保証される。

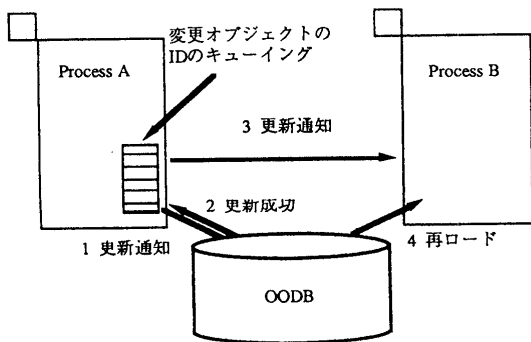


図1 オブジェクトの更新の順序

3.2.1. 連続した更新

ひとつのオブジェクトの更新が連続して行われた場合、最初の更新における伝播が行われた時点で続く更新が完了している場合がある。(図2)この場合、オブジェクトIDだけを通信するので最初の更新による伝播で各プロセスが再ロードした結果が、次の更新の結果となる場合がある。(図3 Aによる更新の通知の結果、Process CはBによる更新後の状態となる。)そして、続く更新の伝播は、意味の無いものになってしまう。しかし、OODB内で保証された矛盾がない結果なので問題はない。

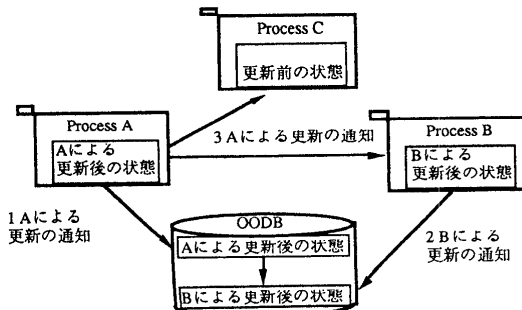


図2連続した更新が起こった場合(1)

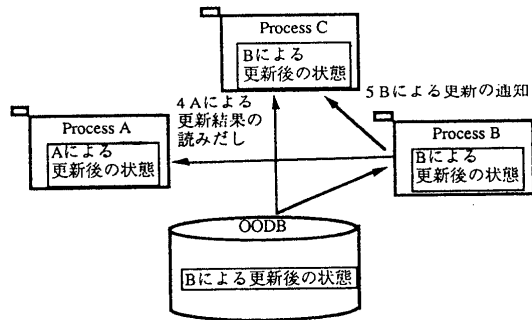


図3連続した更新が起こった場合(2)

4. 課題

情報量が多いオブジェクトの場合、OODBからロードすることにコストがかかる可能性がある。この場合、変更があった属性のみをロードする方法や、変更内容を通信する方法が考えられる。これらの方法は、通信コストやデータの無矛盾性とのトレードオフを考慮して選択する必要がある。

5. まとめ

本稿では、我々が現在開発中の協調ハイパーメディアシステムにおける複数プロセス間の共有オブジェクト更新伝播方法について述べた。

参考文献

[1]牛田「共有オブジェクトのGUIフレームワークについて」,情報処理学会第45回全国大会