

協調ハイパーメディアシステムにおける画面更新方式

5W-5

古林 勝紀 松井 真理夫
NTTデータ通信株式会社

(1) はじめに

リアルタイム型グループウェアシステムにおいて、複数利用者間の並行処理制御は重要な研究課題である。これに関して従来データベースのトランザクション管理の問題として研究されてきた[1]。

現在、多くのデータベース管理システムでは並行処理制御方法として2層施錠方式が採用されている。この方式ではデッドロックが発生した場合、実行中のトランザクションを棄却し、更新されたデータを元の状態にロールバック(後退復帰)させなければならない。しかし、そのトランザクション中に既に出力装置に出力されたデータに関してはそれを取り消すことは、一般に困難である。

現在我々は協調ハイパーメディアシステム[2][3]を開発している。これは従来のハイパーメディアシステムを複数ユーザで同時に利用できるように拡張したものである。本システムでは、並行処理制御のためにオブジェクト指向データベース(OODB)を利用している。本稿では、本システムにおいて、トランザクション中に画面の更新を行った場合に前述の問題を解決する方法について説明する。

(2) 協調ハイパーメディアシステムの特徴

開発中の協調ハイパーメディアシステムは複数のWSをLANで接続し、その内の一台をOODBを有するサーバとし、本システムで扱われるオブジェクトを格納する。以下、このシステムの特徴について説明する。

オブジェクトの共有

本システムで扱われるオブジェクトはネットワークを介して単一のOODBに格納されることにより、複数WS上のユーザがオブジェクトを共有することを実現している。

並行処理制御機能とデッドロックの発生

OODBの並行処理制御機能を利用することにより、複数ユーザの同時アクセスに対してもオブジェクトの整合性を保証することができる。

また、本システムで使用するOODBでは2相施錠方式による並行処理制御を行っている。特に、ユーザがメソッドを動的に定義することを許しているため、トランザクションのデッドロックが発生するこ

とが予測される。

視覚表現の更新

本システムで扱われるオブジェクトは画面上での視覚表現を持っており、これによりオブジェクトのGUIを提供している。本システムでは視覚表現の内容を常に最新に保つておくため、オブジェクトの更新と同時にそれに対応する視覚表現を更新する必要がある。

(3) 画面更新における問題

次に本方式を取り入れずに画面の更新を行った場合の問題について例を用いて説明する。

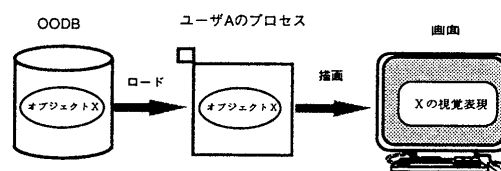


図1 オブジェクトのロードと表示

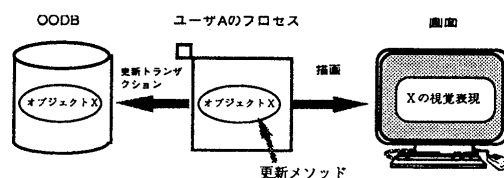


図2 オブジェクトの更新

図1のようにOODB内のオブジェクトXがユーザAによってロードされ、その視覚表現が画面に表示されているとする。次に、ユーザAがオブジェクトXを更新したとする。その場合、図2に示すように、ユーザAからオブジェクトXの更新トランザクションがOODBに送られ、同時に画面上ではオブジェクトXの視覚表現の更新が行われる。

ここでさらにこのユーザAのトランザクションと他のトランザクションがデッドロックを起こし、ユーザAのトランザクションが棄却されたとする。その場合、OODBのロールバック機能によりそのトランザクション内の更新結果は破棄され、オブジェクトXは元の状態に戻される。しかし、そのトランザクション中に実行された描画メソッドにより既にXの視覚表現は更新されている。従ってオブジェクト

Xと同様に、画面も元の状態に戻す必要があるが、既に更新された画面を元に戻すことは非常に困難である。

一方、従来のウィンドウシステムでは、通信コストの削減等のため描画メソッドをバッファにキューイングし、すぐに画面の更新を行わない方法がとられている。しかし、デッドロックが発生したときに、画面の更新が行われない可能性もあるが、常にそれが保証されるわけではない。つまり次の場合、バッファ内の描画メソッドにより画面への更新が行われる。

- (i)例えばウィンドウシステムの往復リクエストのような画面更新を強制的に行うメソッドが実行された場合
- (ii)バッファに一定量以上の情報が蓄積された場合

このように、オブジェクトの更新と画面の更新を同時に行ったのでは、更新された画面を元に戻す必要が生じてくる。

これを解決するために、1トランザクションにおいて更新メソッドを実行した後、描画メソッドを実行する方法が考えられる。しかし、例えば更新メソッド内の中間的な情報を画面に出力したい場合のように、オブジェクトの更新メソッドと描画メソッドを切り分けることは不可能である。特に本システムでは、オブジェクトの更新メソッドと描画メソッドを区別することなくユーザがメソッドを記述できる。

(4)本システムの画面の更新方式

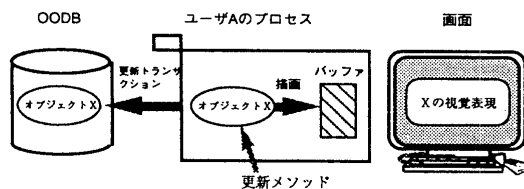


図3 バッファを利用したオブジェクトの更新

この問題を解決するために本システムでは、新たに各ユーザのプロセス毎にバッファを用意する。図3のように一つのトランザクション中に実行された描画メソッドはそこにキューイングされる。そのため実際に画面への描画は行われない。そして、そのトランザクションのコミットが成功したならば、バッファ内のすべての描画メソッドを実際に実行して画面の更新を行う。一方、そのトランザクションのコミットに失敗または途中でデッドロックが発生しロールバックの必要が生じたならば、バッファ内の情報をすべて破棄し画面の更新は行わない。

このように画面の更新とトランザクションのコミットを連携させることにより、更新画面の取り消し処理を避けることができる。この処理の流れを図4に示す。

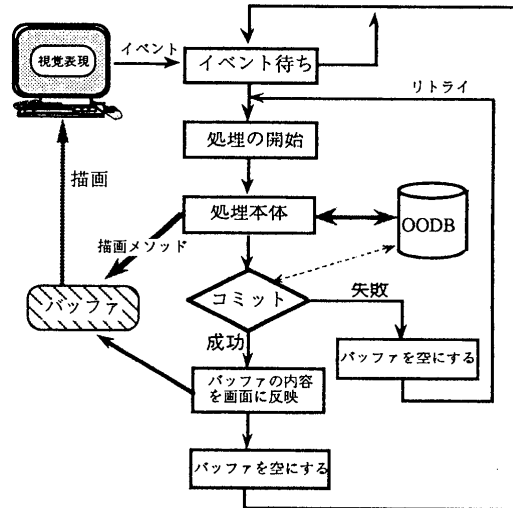


図4 画面更新処理の流れ

またトランザクションのコミットが成功するまで画面の更新を行わないために、強制的に画面更新を行う描画メソッドを使用しないこととする。

(5) おわりに

本稿では、現在開発中の協調ハイパーメディアシステムにおける画面更新の問題点とそれを解決する方式について述べた。この方式は描画メソッドをバッファにキューイングし、トランザクションのコミットと画面の更新を連携させるものである。

今後は、本方式の実装による性能の評価及び他の出力装置への適用の検討を行う予定である。

参考文献

- [1] 西尾：「オブジェクト指向データベースシステムにおける並行処理制御」,情報処理学会学会誌 vol.32,No5,1991.
- [2] 井上、松井、神谷：「グループ意思決定支援システムの開発」,情報処理学会第44回全国大会,3M-1,1992.
- [3] 塩谷、牛田：「グループウェアのためのテストベッドについて」,情報処理学会第44回全国大会,3M-2,1992.