

緩い一貫性に基づいたグループウェアのための 共有オブジェクト空間の実装

2D-1

牛嶋 一智 千葉 滋 益田 隆司

東京大学大学院 理学系研究科 情報科学専攻

1 はじめに

従来のアプリケーションに対するグループウェアの特徴は、図形や文書などのデータオブジェクトを複数のアプリケーションの間で共有し、共有されたオブジェクトを複数のユーザの間の対話的な協調作業を通じて加工してゆく点である。したがって、あるアプリケーションにおいて共有オブジェクトが変更された場合、これを他のアプリケーションにも反映させる必要がある。そこで本研究では、グループウェアの開発基盤として、分散システム内で共有されたオブジェクトの変更をアプリケーションに対して通知する機能を持ったアクティブな共有オブジェクト空間をC++の拡張機構として提供する。

共有オブジェクト空間の実装においては、共有オブジェクトの複製の間の一貫性保持のためのメッセージ通信のオーバーヘッドが問題となる。本システムでは、分散共有メモリで使われる緩い一貫性制御プロトコルを応用し、共有されたデータオブジェクトに対して、アプリケーションの必要に応じた強さの一貫性保持プロトコルを指定することによって、メッセージ通信のオーバーヘッド削減を図る。

2 共有オブジェクト空間

共有オブジェクト空間とは、異なるサイト上で動作するアプリケーションの間でオブジェクトを共有するためのC++の拡張機構である。共有オブジェクト空間内のオブジェクトは、共有されるオブジェクトの変更をアプリケーションに対して通知する機構を持っている。

マルチユーザグラフィカルエディタのように図形オブジェクトを共有するようなグループウェアでは、共有オブジェクトが他のユーザによって変更された場合、暗黙のうちにこれを画面の表示等に反映させる必要がある。このような処理を実現するためには、共有オブジェクトの変更を通知することができるように、共有オブジェクトに一定のアクティビティを持たせる必要がある[3]。

本稿で述べる共有オブジェクト空間では、それぞれのアプリケーションは active replica と呼ばれる共有オブジェクトの複製を保持する。active replica はローカ

ルな共有オブジェクトの変更を他の active replica に対して伝え、また共有オブジェクトが他のサイトで変更された場合、これをアプリケーションに対して通知する機能を持つ。したがって、あるアプリケーションによって共有オブジェクトが変更された場合、その変更は共有オブジェクトの active replica を経由して他のアプリケーションに対して通知されることになる。通知を受けたアプリケーションは、共有オブジェクトの表示の更新や依存関係を持ったデータの再計算を行うことができる。

共有オブジェクト空間は Open C++[2] を用いて実装されており、共有オブジェクトを通常のC++のオブジェクトと同様に定義し、取り扱うことができる。Open C++ はC++に自己反映計算の機構を持たせたプログラミング言語である。Open C++ を利用すると、共有オブジェクトの参照・更新のためのメソッドが起動されたときに、一貫性保持や変更の通知を行うようにC++を拡張することができる。したがってプログラマは、他の複製への変更の通知のための通信を直接記述する必要はなく、分散環境を意識しない通常のプログラムと同じようにしてアプリケーションを記述することができる。

3 緩い一貫性制御プロトコル

共有オブジェクト空間を複製を用いて実装する場合、共有オブジェクトの一貫性制御のためのメッセージ通信のオーバーヘッドの削減が重要となる。

本稿ではアプリケーションの必要に応じて共有オブジェクトの複製の間の一貫性を緩めることにより、全体でやりとりされるメッセージの数を減らし、実行効率を上げることを考える。例えばグラフィックエディタにおいて、ユーザの注目する画面内の図形オブジェクトは強い一貫性制御プロトコルで管理し、そうでない画面外の図形オブジェクトは緩い一貫性制御プロトコルで管理するというを行う。

本システムでは、それぞれの共有オブジェクトに対して強さの異なる以下の3種類の一貫性制御プロトコルを用意している。

Design and Implementation of a Shared Object Space for Groupware using Weak Consistency Protocols
Kazutomo Ushijima, Shigeru Chiba, and Takashi Masuda
Department of Information Science, Graduate School of Science, University of Tokyo

Sequential Consistency Protocol 共有オブジェクトに対する参照操作では複製の値が参照される。更新操作によって変更されたオブジェクトの値は直ちにマルチキャストされ、他の複製に反映され

る。変更通知はシリアルライズされ、すべての複製は通知メッセージを同じ順序で受けとる。共有オブジェクトに対する最も強い一貫性が保証される。

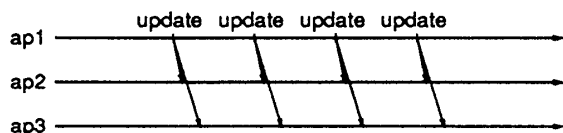


図 1: Sequential Consistency Protocol

Delayed Update Protocol 共有オブジェクトに対する参照操作では複製の値が参照される。更新操作の時は、複製の更新に先立ち acquire 操作で update lock を取得する。このとき他の複製への参照は許されるが、変更は acquire 操作でブロックされる。変更されたオブジェクトの値は release 操作により他の複製に反映され、ロックも解除される。ロックされている共有オブジェクトに対する参照操作では最新の値が得られるわけではない。

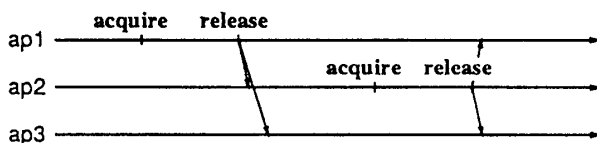


図 2: Delayed Update Protocol

Merged Update Protocol ほぼ 2 番目のプロトコルと同様であるが、acquire 操作でブロックされることはない。また、あるアプリケーションで release 操作が発行される前に他のアプリケーションで acquire 操作が発行されると、先行する release 操作での変更は無視され、最後に release 操作を発行したアプリケーションによる変更だけが他の複製に反映される。

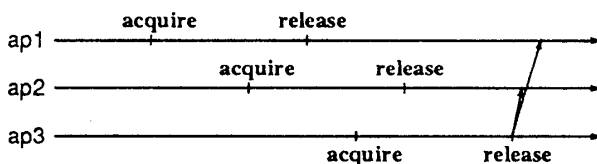


図 3: Merged Update Protocol

はじめの 2 つのプロトコルは他の論文ですでに提案されたものである [1]。3 番目のプロトコルは、とくに強い一貫性の必要とされないグループウェア向けに本稿で提案するプロトコルである。3 番目のプロトコルは、2 番目のプロトコルと比べると、release 操作に伴うマルチキャストを最後の release 操作時の 1 回に抑制して、発生するメッセージの数を減らしている。また、更新開始時の acquire 操作がブロックしないので、各アプリケーションでは中断されることなく共有データオブジェクトを編集することができる。

グラフィカルエディタにおいて共有された図面を複数メンバーで編集する場면을例にとると、Delayed Update Protocol は各メンバーの担当領域が重なっていて排他制御を行いたい場合、Merged Update Protocol はメンバーが互いに議論しながら同じ領域を編集していて、速い応答が要求されるような場合に利用することができる。

4 実行時性能の測定

複数のユーザが共有された図形オブジェクトを画面上で同時に編集することのできるマルチユーザドローイングツールを共有オブジェクト空間を利用して作成し、緩い一貫性制御プロトコルの性能の比較を行った。

4 台のサイト上のアプリケーションから共有されている図形オブジェクトに対して、一定の変更操作をポアソン分布に従った時間間隔でそれぞれ 30 回実行し、それぞれのサイトで発生したメッセージの数と acquire, release, update 操作に要した累積時間の平均を計測した*。

プロトコル	メッセージ数	累積時間
Sequential	6000	1.199(sec)
Delayed Update	360	0.893(sec)
Merged Update	298	0.350(sec)

表 1: 計測結果

一貫性を緩めるにしたがって、共有オブジェクトに対して保証される性質が弱くなるが、変更操作に要するメッセージ数・累積時間をともに減少させることができる。

5 おわりに

グループウェアの開発支援の基盤システムとして、アクティビティを持った共有オブジェクト空間を実装した。また、グループウェア向けの緩い一貫性制御プロトコルを提案し性能の比較を行った。

参考文献

- [1] J. K. Bennett, J. B. Carter, and W. Zwaenepoel. Munin: Distributed shared memory based on type-specific memory coherence. In *the 2nd ACM Symp. on Principles & Practice of Parallel Programming*, pages 168-176, March 1990.
- [2] S. Chiba and T. Masuda. Designing an extensible distributed language with a meta-level architecture. In *Proc. of the 7th European Conference on Object-Oriented Programming*, pages 482-501, 1993.
- [3] Prasun Dewan and Rajiv Choudhary. A high-level and flexible framework for implementing multiuser user interfaces. *ACM Transactions on Information System*, pages 345-380, October 1992.

*Sun-4/ELC,Ethernet