

分散環境におけるオブジェクト指向データベースの実現に関する一方式

4C-4

前田和昭*, 野呂昌満**

*中部大学 経営情報学部 経営情報学科

**南山大学 経営学部 情報管理学科

1. はじめに

ソフトウェア開発に携わる開発者は、プログラム・仕様書・マニュアル等を共有しながら開発を進めていく。また、ネットワーク環境を利用して開発を進めることも多い。そこで、ソフトウェア開発支援環境がネットワーク環境下での共有化機構を持つ必要がでてくる。われわれは、ソフトウェア開発支援環境を対象にしたオブジェクト指向データベース管理システムLeO [1,2]の開発を進めている。LeOの特徴は、オブジェクト指向データベースの技術と分散環境におけるデータ共有化機構の技術とを融合したところにある。本論文では、LeOが取り入れている、分散環境におけるオブジェクト共有化のための実現方式について述べる。

2. 分散環境向けソフトウェア開発支援環境

最近のソフトウェア開発支援環境(以下、SDEと略記する)は、データベース(以下、DBと略記する)を中心に置き、それを各ツールが共通に利用するという構造を持つ。このSDE用DBとして、オブジェクト指向データベース(以下、OODBと略記する)がいくつか提案されている。SDE用DBは、ソフトウェア開発の途上で作成されるもの全て(プログラム・マニュアル・データフロー図等)を保存するためのものである。これらは、木構造やグラフ構造のような複雑なデータ構造を用いて表現することが多い。OODBを利用すると、このような複雑なデータ構造を扱うことが可能となる。われわれは、「OODBとは、補助記憶中のDBに格納されたオブジェクトを操作・管理する機能を持ったオブジェクト指向プログラミングシステムである」と考えている。

An Implementation Method of Object-Oriented Database Systems in Distributed Environments

Kazuaki Maeda *, Masami Noro **

*Chubu University, **Nanzan University

OODBが提供するプログラミングシステムを利用してソフトウェアツールを開発すれば、プログラム・マニュアル・データフロー図等をオブジェクトを使って統一的に扱うことが可能となる。

分散処理の研究の一分野として、分散環境でのデータ共有化機構を備えたシステムが多数提案されている。これらの実現方式は、DSM (Distributed Shared Memory) とDOM (Distributed Object Memory) の二つに分類することができる[3]。DSMは、ページ化による仮想記憶機構の考えを応用し、ネットワーク上で仮想的な単一アドレス空間を作り出す。DOMは、プログラミング言語で扱うオブジェクトを最小単位として共有の機構を提供する。

DSMやDOMの手法を用いたシステムは、ネットワークに接続している複数の計算機の主記憶の共有を可能にするが、そこで扱うデータは永続性を持たない。また、OODBは、オブジェクトに永続性を持たせることを可能にしているが、分散環境での利用を積極的に支援するものではない。そこで、OODBの技術と分散環境におけるデータ共有化機構の技術とを融合することで、分散環境で永続性を持つ共有可能なオブジェクトを扱うことができると考えた。われわれは、この考えに基づきオブジェクト指向データベース管理システムLeOを開発している。

3. 実現方式

SDEで扱うオブジェクトには、

- 1) 永続性を持つものが多い。
- 2) 特定のオブジェクトが複数のプロセスから同時に頻繁に変更の要求を受けることはまれである。
- 3) オブジェクトの大きさに、ばらつきがある。
- 4) 特定のプロジェクト専用で共有されるオブジェクトがある。

という特徴がある。そこで、以下のような方式をLeOに取り入れた。

- 1) メモリマップ機構を使って永続性を実現する。これにより、主記憶に存在するオブジェクトと補助記憶に存在するオブジェクトが同じ操作体系でアクセス可能になる。
- 2) 分散環境でのオブジェクト共有化の実現のために、DSMの機構を取り入れる。DSMで議論されている偽共有 (false sharing) の問題は、オブジェクトのページへの配置を工夫することで、その頻度を下げる。
- 3) 関係するオブジェクトをなるべく同一ページ内に置くように工夫する。そのために、オブジェクトをクラスタ化する機能を持たせる。関係しているオブジェクトを同じクラスタに入れるようにユーザが指示し、ページ割り当てのヒントにする。
- 4) オブジェクト共有の範囲を特定のプロジェクト用に限定できるようにする。

LeOが扱うオブジェクトは、ネットワーク上で唯一であるOID (Object ID) を持つ。LeOが作り出すネットワーク上の仮想的なオブジェクト空間では、オブジェクトを参照するためにOIDを用いる。このOIDを管理するために、ネットワーク上にOIDサーバを置く。OIDサーバは、OIDからページ番号への写像を行なう。OIDサーバが保有するOIDテーブルには、実際に使われているページ番号やページ内オフセット等の情報が保持される。

仮想的なオブジェクト空間を実現するために、LeOはブロードキャストを使いオブジェクトの一貫性を保っている。ネットワークに接続している各計算機の記憶管理は、LeO Managerが行なう。LeO Managerは、全てが対等の関係を保ちながら仮想共有オブジェクト空間を管理する。また、占有の領域と共有の領域は別々に管理する。メモリ管理の単位は、固定長のページを利用している。特定の計算機が所有しないページにアクセスした場合、そのページの所有者を探すために、ブロードキャストメッセージを送信する。その返事として該当するページを返し、そのページをプログラムがアクセスできるように設定を行なう。

LeO Managerが受け取るブロードキャストに矛盾がないようにするためには、全てのブロードキャストが同じ順番に到着することを保障する必要がある。

そこで、ブロードキャストメッセージに、BID (Broadcast ID) を持たせ、メッセージの順番を明記する。このBIDの割り当ては、BIDサーバが行ない、実際には連続した数値を使う。ブロードキャストは、以下の手順で行なう。

1. オブジェクトがLeO Managerにメッセージ渡す。
2. LeO Managerがメッセージを受信すると、そのオブジェクトを一時停止させる。
3. LeO ManagerがBIDサーバに1対1のメッセージを送る。
4. BIDサーバは、メッセージにBIDを付加しメッセージをブロードキャストする。
5. 送信元のLeO Managerがブロードキャスト通信を受信した後、矛盾がなければオブジェクトの実行を再開する。

LeO Managerが矛盾する順番でメッセージを受け取った場合は、再送信の要求を出すことで到着順を保障する。

4. まとめ

本論文は、分散環境におけるオブジェクト共有化のための実現方法について述べた。この方法は、ネットワークに接続している各計算機の主記憶を仮想的なアドレス空間に見せるものである。さらに、メモリマップ技術を利用し、オブジェクトの永続性を実現しようというものである。現在、LeOの開発を進めており、今後、性能評価や各種拡張を行なう予定である。

参考文献

- [1] 前田和昭, 野呂昌満, "カスタマイズ可能なソフトウェア開発支援環境用データベース管理システムの設計," ソフトウェアシンポジウム'92, pp.E28-E34, 1992.
- [2] 前田和昭, 野呂昌満, "分散環境におけるオブジェクト共有化機構," ソフトウェアシンポジウム'93, pp.78-83, 1993.
- [3] Michael J. Feeley and Henry M. Levy, "Distributed Shared Memory with Versioned Objects," Proceedings of ACM Conference on Object Oriented Programming Systems, Languages and Applications, pp.247-262, 1992.