

分散オブジェクト基盤上の OODB アプリケーション開発支援ツールの

4 A a - 3

試作と評価

北野 拓哉

NEC C&C メディア研究所

1. はじめに

WWW やインターネット上の分散オブジェクト基盤を用いてデータベース(DB)を利用するアプリケーション(AP)が増えてきた。このような AP の開発を支援するために、筆者らは PERCIO[1]の DB スキーマを分散オブジェクトとしてラッピングするコードの自動生成ツールを開発した。本ツールは、分散オブジェクト基盤の一つである COM/DCOM のオートメーションオブジェクト(AtO)を対象にした。DB スキーマのメンバ変数やメソッドをそれぞれオートメーションプロパティ/メソッドとして外部の AP に公開する。

DB スキーマが AtO としての API を持つことは DB のオープン性を高め、さまざまな分散 DB-AP の開発・実行を実現し可能性を広げる。VB などオートメーションをサポートする言語を用い、インタプリタの環境で開発可能となる。AtO は EXE, DLL, OCX の各実行形式で作成可能であるが、これらの形式を活かした分散オブジェクトのさまざまな組合せによる AP の構築・実行が可能となる。

本稿では本ツールの AtO のコード生成方式を説明してから、従来の分散 DB-AP の開発・実行方式と EXE/DLL 形式の AtO を用いる方式との違いを性能比較も踏まえて説明し、考察する。

2. AtO のコードの自動生成方式

一般に AtO の開発では、ユーザはオブジェクト記述言語(ODL)によってタイプライブラリを作成したり、AtO 特有のコードを開発言語で記述しなければならない。本ツールはオートメーションとして公開対象とする DB と、ユーザからの簡単なオプションの設定から AtO の開発に必要なコードを自動生成する。

本ツールの他にも以前からこのような開発労力を軽減するツールは存在する。ObjectStore[2]は DB スキーマ定義とは別の、DB オブジェクトを AtO として公開するための型定義を記述する型記述ファイル(OST)とそのコンパイラを用意している。ユーザは OST を作成しコンパイルすれば、AtO のための ODL の定義ファイルと開発言語 C++のコードを得ることができる。

本ツールは OST のような DB オブジェクトの情報

はユーザが記述するのではなく、既存の DB から実行時に取得する。その取得したスキーマ情報を基に、AtO の細かな情報設定に関しユーザにオプションで取捨選択をしてもらった後、本ツールは ODL 定義ファイルと AtO のコードを自動生成する。ユーザに AtO コーディングの特別な知識と労力を要求することなく、既存の DB をそのまま AtO として拡張するツールといえる。

なお、本ツールが自動生成の対象としている DB スキーマのクラスはユーザ定義のクラスである。PERCIO が本来有する操作言語 C++ API[1]のクラスに関しては、AtO のコードをあらかじめ用意して対応している。もっともユーザ定義のクラスは C++ API を用いて定義されるので、自動生成には C++ API の AtO コードが必要である。

3 分散 DB-AP の開発・実行方式の比較

● 従来の分散 DB-AP 開発・実行方式

PERCIO は従来より分散 DB-AP を開発する環境として、クライアントとサーバにそれぞれ専用の DBMS をインストールして分散 DB を管理する環境を用意している[1] (図 1)。この開発環境は PERCIO に限らず多くの DBMS でも同様である。

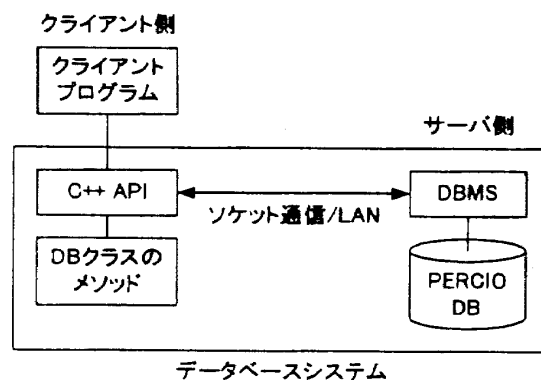


図 1 PERCIO の分散 DB-AP の環境

● EXE 形式の分散 DB-AP 開発・実行方式

従来の分散 DB-AP の環境はクライアント側にも DBMS 専用の環境をインストールする必要があり、インターネット向きであるといえる。本ツールの目的の一つに、クライアント側にこのような専用の環境を必要とせずに分散 DB-AP の開発・実行環境を提供することがある。そこで AtO を用いて DB をラッピン

グし、AtO のレベルで C/S 間のデータ通信を実現させる方法を用いた。本ツールは、インターネットでの分散 DB-AP 開発・実行環境としても利用可能である。

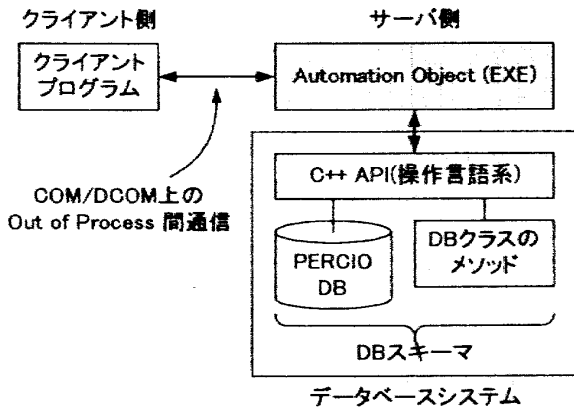


図 2 EXE 形式を用いた分散 DB-AP

EXE 形式の AtO は、Out of Process のリモートオートメーションで操作可能である。クライアント側には DB スキーマが AtO としてそのまま公開される。よって、あたかもそこに DB があるかのようにプログラムを記述することができる。DB スキーマやそのメソッドなどデータベースシステムに関する開発・実行と、そのクライアント AP の開発・実行とに分散させることができる。

● DLL 形式の分散 DB-AP 開発・実行方式

DLL 形式の AtO は、他の AP の In-Process として実行させる。一般に In-Process はプロセス間通信より高速にデータ通信を行える。図 2 のサーバプログラムの開発に、EXE 形式と同様、DB スキーマをオートメーションとして操作するプログラムの記述が可能である。クライアント側はサーバプログラム自体を AtO として利用し、クライアントプログラムを開発する。

● 性能比較

性能評価テストプログラム(TP)により、通常の C++ API, EXE 形式, DLL 形式のそれぞれの性能を測定した。本稿で対象とする TP のデータはある DB クラスのオブジェクト 1500 件で、その全件探索と条件検索(100 件がヒットする)の実行時間を測定した。使用したマシンはサーバ側が PentiumII 266MHz, クライアント側が Pentium 120MHz, OS は双方とも WindowsNT4.0 である。

サーバマシンのみのスタンドアロンで C++ API と DLL 形式の性能比較を行った。結果は C++ API が全件探索で 2.98 秒, 条件検索で 0.30 秒, DLL 形式はそれぞれ 5.63 秒, 0.38 秒であった。DLL 形式は C++ API に対し全件探索で約 50%, 条件検索で約 20%性能が低下することが分かった。

C/S 型の環境で C++ API(図 1)と EXE 形式(図 2)の性

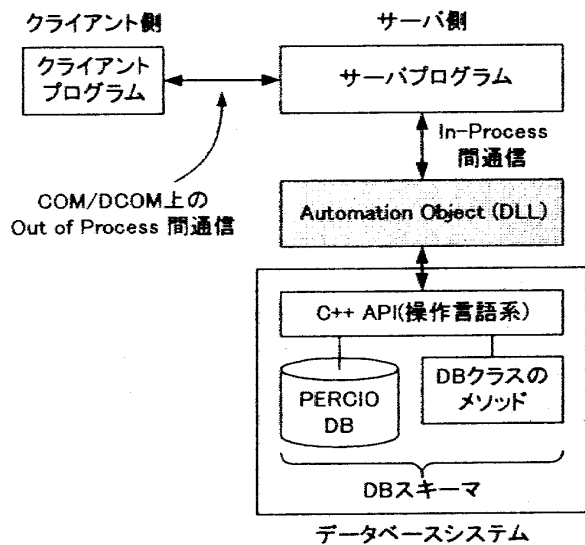


図 3 DLL 形式を用いた分散 DB-AP

能比較を行った。結果は C++ API が全件探索で 26.16 秒, 条件検索で 2.72 秒, EXE 形式はそれぞれ 53.07 秒, 3.07 秒であった。EXE 形式は C++ API に対し全件探索で約 50%, 条件検索で約 10%性能低下することが分かった。

EXE 形式(図 2)と DLL 形式(図 3)の性能比較を行った。結果は EXE 形式が全件探索, 条件検索で先ほど示した 53.07 秒と 3.07 秒, DLL 形式はその約 1/10 の 5.59 秒, 0.39 秒であった。この DLL 形式の性能は、最速の環境であろうスタンドアロンの C++ API と比べても全件探索で 3 秒以下, 条件検索で 0.1 秒以下の差しかないことが分かった。

4. まとめ

本稿では本ツールの AtO のコード生成方式と、その AtO を用いる分散 DB-AP の開発・実行方式とその性能に関して述べた。DB のスキーマレベルでクライアントに公開することはクライアント側における AP の拡張可能性, カスタマイズ性を高めるが、高い性能は出せない。AP レベルでオートメーション通信を行えばスタンドアロンとほとんど変わらぬ性能が出せるが、反面、特定の機能しかクライアント側に公開できず AP は固定的となる。従来の分散 DB-AP の形態を含め、クライアントの環境条件や性能要求, AP の拡張性も考慮した上で分散 DB-AP の設計を行う必要があるといえる。

参考文献

[1] NEC Corp., "PERCIO オブジェクト指向データベース管理システム", http://www.ace.comp.nec.co.jp/product/db/percio/p_main.htm, 1996-1997.
 [2] Object Design, Inc., "ObjectStore PSE Pro for ActiveX, version 2.0", <http://www.odi.com/content/products/psax/pseproax.html>, 1997.