

オブジェクト指向モデル記述言語のインタフェース環境の開発

7N-6

水越 功 佐藤達也 細谷伊知郎 高橋大輔 金村星吉 上田賀一

茨城大学 工学部 情報工学科

1 はじめに

近年、視覚的プロトタイピングの需要が高まり、本研究で作成しているオブジェクト指向モデル記述言語（以下、ベース言語）[1]にもそれを実現する機能が望まれる。一方、永続オブジェクトの実現やオブジェクト間の整合性を図るにはデータベースの活用が有効である。

これらの要求を満たすため、ベース言語では Motif ウィジェットセットを用いたグラフィカルユーザインタフェース（GUI）およびリレーショナルデータベースを用いたデータベースインタフェースを作成した。

本発表では、グラフィックスインタフェース、データベースサーバ、およびそれとベース言語間の通信の実現方法を報告する。

2 グラフィックスインタフェース

グラフィックスインタフェースとして実現したい機能は以下の通りである。

- オブジェクトの状態の表示（ブラウザ機能）
- 専用エディタによるプログラム作成
- エディタで作成したプログラムの実行開始指示
- ベース言語を使用したアプリケーション開発

これらを実現するための機構、すなわちウィジェットオブジェクトセットを提供する。このセットは Widget オブジェクトとして実現され、Object, String, Template などのオブジェクトと同列に存在する。

2.1 ウィジェットオブジェクト

ウィジェットオブジェクトは GUI を構築する部品であり、その雛型である。すなわちベース言語上ではウィジェットオブジェクトに対してウィジェット生成を要求し、得られたオブジェクトに対してメッセージを送信することによってウィジェットのカスタマイズを行なう。

画面表示に際して必要であると思われるウィジェットは表 1 の通りである。

表 1: ウィジェットオブジェクト

ウィジェット	説明
Frame	各ウィジェットを集約するウィジェット
Label	ラベルの役割をするウィジェット
Button	何らかの機能を動作させるスイッチの役割をするウィジェット
Editor	ユーザが使用するテキストエディタの役割をするウィジェット
MenuBar	メニューバーの役割をするウィジェット
Pulldown-Menu	プルダウンメニューの役割をするウィジェット
Dialog	様々なメッセージを表示するために現れるダイアログボックスウィジェット
Canvas	図を表示するウィジェット

これらのウィジェットオブジェクトは Motif ツールキットのウィジェットセットを使用して実現する [2]。ウィジェットのカスタマイズは Motif ウィジェットセットのリソースを変更することで行なう。

各ウィジェットオブジェクトはメソッドを属性として持ち、それを呼ぶことによってウィジェット作成、リソース変更/獲得などを行なう。メソッドで扱うリソースは Motif ウィジェットセットのリソースである。

各ウィジェットオブジェクトでは使用している Motif ウィジェットのリソースを獲得/変更できる。

3 データベースインタフェース

本研究ではリレーショナルデータベース [3] である Sybase を使用した。

オブジェクトを格納するデータベースは次の機能を提供しなければならない。

- オブジェクトの格納
- オブジェクトの取り出し
- オブジェクトの削除
- オブジェクトの保護（セキュリティ）
- キーワード、コメントの保存をサポートすることによる再利用性の向上

オブジェクトをデータベースに格納する際にキーワードおよびコメントも格納することにより、オブジェクト再利用時の検索を可能にする。

Development of Interface Facilities on Object-Oriented Model Description Language
 Isao Mizukoshi, Tatsuya Sato, Ichiro Hosoya, Daisuke Takahashi, Seikichi Kanemura, Yoshikazu Ueda
 Department of Computer and Information Sciences, Faculty of Engineering, Ibaraki University

3.1 テーブル

ベース言語からデータベースを扱うために必要なテーブルは表2の通りである。なお、テーブルの参照にはOID, UID, キーワード, コメントを返すビューを使用する。

表 2: テーブル

テーブル	説明
base_table	他のテーブルの操作の際に参照されるテーブル
attribute_table	オブジェクトとその属性オブジェクトをカラムとするテーブル
value_table	属性値のテーブル
ref_oid_table	あるユーザが参照できるOIDが列挙されているテーブル
listing_table	あるオブジェクトの属性が列挙されているテーブル

3.2 ストアドプロシジャ

ストアドプロシジャとはSQLとフロー制御言語の集合である。ストアドプロシジャを使用することによって以下のことができる。

- パラメータの取得
- 他のプロシジャの呼び出し
- 呼び出し側プロシジャに終了ステータスを送信
- 呼び出し側プロシジャにパラメータを送信

データベース関連は、すべき作業が決まっているので表3に示すストアドプロシジャをSQLサーバ中に作成しておき、テーブルに値を挿入/変更する際に使用する。

表 3: ストアドプロシジャ

プロシジャ	説明
make_table	テーブルを作成し、初期化する
insert_base	base_table に値を挿入する
insert_attribute	base_table を参照しながら attribute_table に値を挿入する
insert_value	base_table を参照しながら value_table に値を挿入する
use_ref_oid_table	自分が参照可能なOIDを ref_oid_table に返す
listing	あるオブジェクトの属性をすべて listing_table に返す

4 データベースサーバ

ベース言語が動作しているマシンとSQLサーバが動作しているマシンが異なる場合、マシン間の通信(プロセス間通信) [4] が必要になる。

本研究では通信プロトコルとしてTCPを使用した。データのやりとりは図1のようになっている。

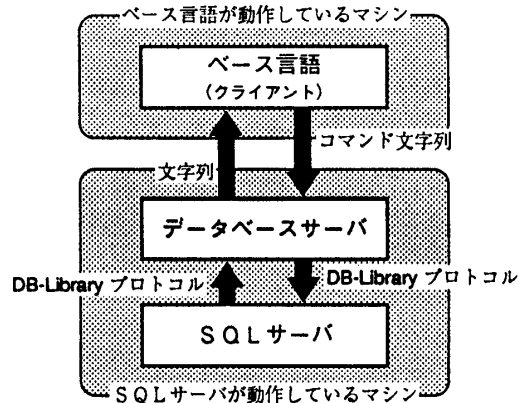


図 1: クライアント/サーバ間通信

4.1 SQL コマンドの処理

実際にデータベースを使用する際はデータベースサーバがベース言語から適当な引数を受け取り、SQLサーバにSQLコマンドを送信することによってストアドプロシジャを起動する。データベースサーバとSQLサーバの間の通信プロトコルはデータベースライブラリによって決められている。

データベースサーバがSQLサーバから受け取った実行結果は、ベース言語が利用しやすい形の文字列にパックしてベース言語に送信する。

5 おわりに

オブジェクト指向インタプリタ言語上にグラフィックス/データベースインタフェースを実装した。

各部門には次のような問題がある。

- グラフィックスインタフェース: Motif ツールキットのウィジェットセットを使用したことによるベース言語の実行速度の低下
- データベースインタフェース: オブジェクトのバージョン管理や複数ユーザへの対応
- データベースサーバ: データ送信失敗時の処理

今後はこれらの問題を解決し、よりよいインタフェース作りを目指していきたい。

参考文献

- [1] 中野喜之, 小飼 敬, 上田賀一ほか: “オブジェクト指向モジュール記述言語のインタプリタ処理系の開発”, 情報処理学会第52回全国大会 7N-7 (1996)
- [2] Eric F. Johnson, Kevin Reichard, 中西隆 監訳: “Motif 1.2 パワープログラミング”, 技術評論社 (1995)
- [3] 増永良文: “リレーショナルデータベース入門”, サイエンス社 (1991)
- [4] W. Richard Stevens, 篠田陽一 訳: “UNIX ネットワークプログラミング”, トッパン (1992)