

## オブジェクト指向分散環境OZ++のワークベンチの実現

7L-7

音川英之\*(シャープ) 吉田泰光\*(日本ユニシス)

塚本享治(電子技術総合研究所)

\* 開放型基盤ソフトウェアつくば研究室研究員

## 1 はじめに

OZ++では、広域ネットワーク上においてクラスを共有し、他人の開発したクラスの利用を促進することで、ソフトウェアの部品化および再利用というオブジェクト指向プログラミングの特徴を有効に利用することが可能である[1]。またワークベンチと呼ぶものを中心にした開発環境を提供し、ネットワーク上のクラスを利用したソフトウェアの開発を容易にしている[2]。本論文ではこのワークベンチについて述べる。

## 2 OZ++におけるプログラミングの概要

OZ++では、C言語を元にしたオブジェクト指向プログラミング言語であるOZ++言語によってプログラミングを行う[1]。プログラムはクラス単位に記述し、コンパイルもクラス単位に行う。

## 2.1 クラス

OZ++では、ネットワーク上でのクラスの共有を実現するためにクラスの分散管理を行っている。クラス管理システムでは、クラスにシステム全体で一意的に識別可能なIDを付与し、このIDによって各クラスを管理する[3]。また、クラスの変更に対して利用者側での柔軟な対応を行うため、クラスをバージョンによって管理すると共に、実行時に新旧のバージョンの混在を可能にしている。バージョンはインタフェースと実装を区別して管理している。コンパイル時には、インタフェースのバージョンだけを特定し、インスタンスの生成時に実装のバージョンを決定する。この実装のバージョンを決定する操作をコンフィギュアと呼ぶ。

## 2.2 スクール

OZ++システム全体ではクラスをIDによって識別す  
An Implementation of Workbench in OZ++: an Object-Oriented Distributed Environment  
Hideyuki Otokawa\* (Sharp Corporation),  
Yasumitsu Yoshida\* (Nihon Unisys), and  
Michiharu Tsukamoto (Electrotechnical Laboratory)  
\* Researcher, Tsukuba Laboratory, Open Fundamental Software Technology Project

るが、プログラミング時のソースコード中ではクラス名を記述できる仕組みが必要である。したがってコンパイラはクラス名をIDに変換する必要があり、そのための変換テーブルとしてスクールと呼ぶものを導入している。スクールはクラス名とIDの2項組のリストであり、クラス名に対するプログラミング時の局所的な名称空間を提供するものである。

## 3 ワークベンチ

ワークベンチはOZ++上でのプログラミングのための統合環境である。ワークベンチ上では、上で述べたスクールを単位としてソフトウェアの開発を行う。以下ではワークベンチの機能および実装について述べる。

## 3.1 機能

ワークベンチは開発するソフトウェア単位に用意されるスクールを管理している。ユーザはそこから開発で利用するスクールを選択して、開発作業を行う。ワークベンチからは以下のツールが利用できる。

## ・ブラウザ

指定されたスクール内のクラス名一覧のブラウズを行う。一覧にあるクラスを選択しそのクラスに関する詳細情報(使用するバージョンの指定、インタフェースや実装の内容)等をブラウズすることもできる。またクラス名や使用するバージョンの変更もこのツールで行う。

## ・コンパイラフロントエンド

指定されたスクールを利用して、コンパイルおよびコンフィギュアを行う。

## ・スクールディレクトリブラウザ

スクールの管理システムであるスクールディレクトリのブラウズを行う。またスクールディレクトリ上のスクールをワークベンチに取り込むことができる。

また開発したソフトウェア(クラス)を他人が利用

できるように、スクールをスクールディレクトリ上に公開する操作もワークベンチ上で行う[1]。

ワークベンチを利用したソフトウェアの開発は次のような手順で行う。

1. ワークベンチを起動し、開発に利用するスクールを選択する。
2. ブラウザを起動し、使用するクラスのバージョンやクラス名に応じてスクールの内容を変更する。
3. ソースコードを記述した後、コンパイラフロントエンドを起動して、コンパイルおよびコンフィギュアを行う。

### 3.1 実装

ワークベンチは図1のように3つの部分からなる

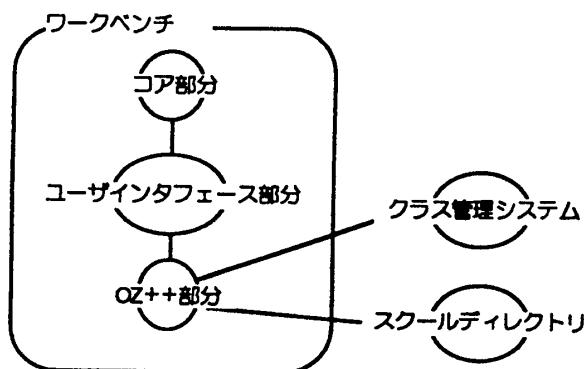


図1: ワークベンチの構成

・コア部分は、ユーザインタフェース部分から起動されるUNIX上のプロセスとして動作し、コンパイルやスクール内容の変更等の主要機能を行う。クラス管理システム等のOZ++上で動作しているものとのやり取りは、ユーザインタフェース部分を介して行う。ブラウザ等のツールの主要機能もこの部分に実装している。

・ユーザインタフェース部分は、OZ++部分から起動されるUNIXプロセスとして動作し、OZ++部分およびコア部分の機能の呼び出しを行う。この部分はX-Window上のGUIツールキットであるTcl/Tkにより実装している。

・OZ++部分は、コンパイル結果のクラス管理システムへの登録等、クラス管理システムやスクールディレクトリ等との情報のやり取りを行う。

このような実装方法を選択したことには二つの理由がある。一つはこのワークベンチ以外に、OZ++システムのブートストラップ環境を構築するための開発環境を提供するためであり、OZ++部分だけ

を除いたものがUNIX上でそのまま利用できるようにしている。一方、コア部分もOZ++部分に取り込んだプロトタイプも作成したが、コンパイル等の処理ではUNIX上のツールを利用するためにプロセス間通信を多用することになり、ユーザの操作上の体感速度が著しく遅くなったため採用しなかった。

また、Tcl/Tkをユーザインタフェースに用いたアプリケーションの開発を容易にするために、OZ++とTcl/Tkとのやり取りを簡単に記述できるライブラリを開発し、ワークベンチの実装でも利用している。

### 3.3 協調開発の支援

現在のワークベンチに、幾つかのクラスをまとめてモジュールとして扱う機能や、そのモジュールを単位としたプロジェクト管理の機能を導入することにより、複数人によるソフトウェア開発において次のような支援が可能である。

- ・あるクラスのインタフェースが変更された場合、そのインタフェースに依存しているクラスの開発者に、変更されたクラス名や変更の内容を通知する。
- ・各モジュールの都合に応じて使用するクラスのバージョンを指定して、動作試験を行うことを可能にする。

## 4 まとめ

本論文では、OZ++システムの開発環境であるワークベンチについて述べた。現在このワークベンチを含むOZ++システム第1版の実装が終了し、その評価および拡張、改良を行っている。今後、3.3節で述べた協調支援機能を実装する予定である。

この研究は情報処理振興事業協会(IPA)が実施している「開放型基盤ソフトウェア研究開発評価事業」の一環として行われたものである。

## 参考文献

1. 音川他, 「オブジェクト指向分散環境OZ++のプログラミングパラダイム」, SWoPP'95, Aug. 1995
2. 籠他, 「オブジェクト指向分散環境OZ++の開発環境ワークベンチ」, 情報処理学会第50回全国大会, Mar. 1995
3. 新部他, 「OZ++コンパイラによるクラスの版管理」, SWoPP'94, Aug. 1994