# オブジェクト指向分散環境 OZ++ のコンパイルサーバの設計

3H−4

音川 英之 * (シャープ)　西岡 利博 * (三菱総合研究所)
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++システムは、ネットワーク上でオブジェクトの実行に必要なソフトウェアの自動配送を行なう仕組みを備えた、オブジェクト指向分散環境である。OZ++システム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]。

このようにOZ++システムは、ネットワークを利用したソフトウェア開発を可能にするものであるが、ネットワーク形態やシステム環境によっては直接OZ++システムの利用が困難な状況が考えられる。このような場合には、何らかの方法で間接的にOZ++システムを利用できる環境が必要となる。

そこでOZ++システムでは、標準的なネットワーク環境となったWWWのCGIインタフェースを利用し、WWWクライアントをユーザインタフェースとするソフトウェア開発環境を実現するために、コンパイルサーバを提供する。

本稿では、このコンパイルサーバについて述べる。

## 2　OZ++プログラム開発の特徴

OZ++システムではネットワーク上でクラスを共有するためにクラスの分散管理を行なっており、クラス管理システムはクラスを一意に識別可能なIDによって管理している[3]。また、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。

一方、プログラミング時にはソースコード中でクラス名を記述するために、クラスのIDと名前を対応付ける名称空間が必要であり、OZ++システムではこれをスクールと呼んでいる。

A design of a compile server in OZ++: an Object-Oriented Distributed Environment
Hideyuki Otokawa* (Sharp Corporation),
Toshihiro Nishioka* (Mitsubishi Research Institute),
Yasumitsu Yoshida* (Nihon Unisys),
and Michiharu Tsukamoto (Electrotechnical Laboratory)
* Researcher, Tsukuba Laboratory, Open Fundamental Software Technology Project

このようにOZ++システムでは、ソフトウェアの共有を促進するためにいくつかの特殊な仕組みを備えており、これらに関する操作機能を持ったワークベンチと呼ぶ開発環境を提供している。

## 3　コンパイルサーバの目的

ワークベンチを用いることによって、他人の開発した有用なライブラリを再利用しながら、柔軟なソフトウェア開発が可能である。ところが、そのためにはOZ++システムが利用可能な環境が必要となるが、例えば外出先や移動中に職場の環境にアクセスする場合などは、OZ++システムが手元の環境で利用できない可能性がある。そのような場合に、WWWクライアントからコンパイル等のソフトウェアの開発を実現することが、コンパイルサーバを導入する目的である。

またコンパイルサーバを導入することにより、開発するソフトウェアのプラットフォーム環境が手元にない場合でも、WWWクライアントから対象となるプラットフォームのOZ++システムが動作している環境を利用することが可能になる。つまり必要なプラットフォームのOZ++システムを用意することなく、いわゆるソフトウェアのクロス開発を実現することも可能である。

## 4　コンパイルサーバの設計

コンパイルサーバは、WWWクライアントからOZ++システム上で動作するソフトウェアの開発を実現するために、WWWのCGIサーバとして動作するOZ++システム上のオブジェクトである(図1)。コンパイルサーバは、WWWサーバとOZ++システムの間の仲介役として機能し、WWWクライアントからの要求をWWWサーバを通して受け取りそれに応じた処理をOZ++システム上で実行したり、OZ++システム上で実行した結果をWWWサーバを通してWWWクライアントに返すという処理を行なう。

### 4.1　機能

3節で述べた環境を実現するために、コンパイルサーバは以下のような機能を提供する。

- ソースプログラムのコンパイル

  WWWクライアントからCGI経由でソースプログラムを受け取り、それをOZ++システム上でコン

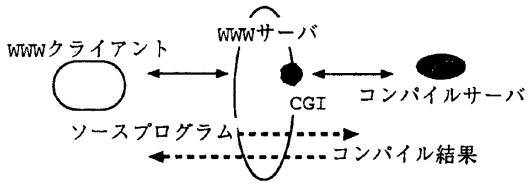


図 1: コンパイルサーバの構成

パイルし、コンパイル結果をWWWサーバ経由でWWWクライアントに返す。

2節で述べたように、OZ++のソフトウェア開発ではクラス名の名称空間をコンパイル時に設定する必要があるが、WWWクライアント上ではできる限り容易にソフトウェア開発が可能となるように、名称空間の設定をコンパイルサーバ側で行ない、WWWクライアントからは固定の名称空間 (使用するクラスには決まった名前が与えられている) を利用する。したがって、コンパイルサーバこの名称空間を設定するためのスクールを管理しなければならない。

また、コンパイルサーバはWWWのCGIサーバとして機能するため、複数のWWWクライアントからのコンパイルの要求を処理しなければなら。これはコンパイルサーバをOZ++システム上のオブジェクトとして実現し、OZ++システムのマルチスレッドの特性を利用した並行処理によって容易に実現可能である。

- コンパイルしたプログラムのリモート実行

  コンパイルしたプログラムを、OZ++システム上で実行し、実行結果をWWWサーバ経由でWWWクライアントに返す。

  ただし、ここで行なう実行というのは開発途中のテスト的な意味のものも含むため、プログラムの安全性が保証されない。つまり、あるプログラムを実行することによって、コンパイルサーバを含む他のOZ++システム上のプログラム (あるいはOZ++システム以外のプログラム) の実行に悪影響を与える可能性があるため、これに対する防御策を講じなければならない。

  OZ++システムでは、プログラム(オブジェクト)をエグゼキュータと呼ぶサブシステム上で実行し、エグゼキュータ単位にメモリ等のプログラムの実行に必要な資源の管理を行なっている。そのため、あるエグゼキュータ上のオブジェクトが他のエグゼキュータに影響を与えるのは、オブジェクト間のメッセージのやり取りのみとなる。したがって、この場合の問題は、開発したプログラムを実行するためのエグゼキュータを別に用意することによって解決することができる。

  ところが、それだけでは実行することによって資源を消費し尽くしてしまうようなプログラムに対しては無防備である。そこで、コンパイルサーバは、実行を開始したオブジェクトを管理し、ある決まった期間以上動作しているオブジェクトを強制的に終了させるなどの仕組みが必要である。

- WWWクライアントとの情報交換

  上で述べたように、コンパイルサーバはWWWクライアントからソースプログラムを受け取ったり、コンパイル結果や実行結果をWWWクライアントに送るという処理が必要である。これは、コンパイルサーバがWWWのCGIインタフェースを利用しているという性格上、HTTPプロトコルによるHTML形式のデータとして行なう。例えば、WWWクライアントから受けとるソースプログラムは、WWWクライアントでフォームによって入力したものをPOSTメソッドでWWWサーバに送信し、コンパイルサーバはそれをWWWサーバから受信する。また、コンパイル結果はコンパイルサーバがHTML形式に整形し、WWWサーバを通してWWWクライアントに送信する。

## 5 まとめ

コンパイルサーバをWWWのCGIサーバとして提供することによって、OZ++システムという分散環境でのソフトウェア開発を、日常利用しているWWWクライアントを利用するだけで可能になる。コンパイルサーバの応用例として、WWWというシステムを利用しているという特徴を活かし、WWWに特化したシステム、例えばWWWの任意のCGIサーバをリモートで開発、制御するという仕組み等を提供することも考えられる。

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