オブジェクト指向分散環境 OZ++ におけるクラスの拡張

6F－5

新部 裕 (三菱総合研究所*)　音川 英之(シャープ*)
濱崎 陽一(電子技術総合研究所)　鈴木 敬行(シャープビジネスコンピュータ*)
平川 秀忠 (日本ユニシス*)　塚本 享治 (電子技術総合研究所)
* 情報処理振興事業協会「開放型基盤ソフトウェア研究開発評価事業」研究員

1　はじめに

OZ++言語は，オブジェクト指向分散環境 OZ++上でのソフトウェア開発のためのオブジェクト指向プログラミング言語 [2]である．OZ++は電総研で開発された分散システム OZ+[1]を基本としており，バージョンの動的な指定が可能となるようにクラスが管理されることに特徴のあるシステムである．

ここでは，OZ+から拡張した OZ++のクラスについて述べる．

2　従来のクラスの概要

OZ++のクラスでは，オブジェクトのメソッドをすべて実行時に動的に結合し，またメソッドの実行コードもコンパイル時に静的にリンクするのではなく，実行時に必要なコードを動的にロードしリンクする[3]．このような仕組みを利用することによって，メソッドのシグネチャに変更がなく実装のみが変更された場合には，そのメソッドを利用するコードの再コンパイルを行なうことなく，変更されたメソッドの実行コードを利用することが可能である．このように，OZ++では利用するクラスのバージョンの動的な指定が可能なバージョン管理を実現する[4]．さらに多重継承を利用することによって，柔軟なプログラミングが可能である．

ところが，このようなバージョン管理を実現するためには，実行システムはオブジェクトとは別にクラスのメソッドの実行コードに関する情報を管理することが必要である(図1)．したがって OZ++のクラスでは，メソッドの動的結合とクラス情報の管理のためのオーバーヘッドによって実行効率およびメモリ効率の両方の点に問題があり，特に文字列や複素数というような比較的粒度が小さくかつ実行効率が要求されるクラスの実装に不向きであるという欠点を持つことになる．

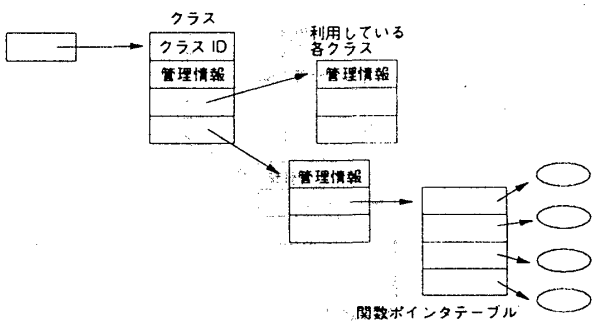

図 1: クラス管理情報の構造

この従来のクラスの欠点を補うため，メソッドをコンパイル時に静的に結合する，スタティッククラスを導入する．さらに，基本型の拡張としてユーザが新たな型を定義するための，レコードを導入する．

3　スタティッククラス

通常のクラスは，柔軟なプログラミングを実現するものであるが，実行効率が要求される場合には問題がある．OZ++のスタティッククラスは，通常のクラスが持つメソッドの動的結合と多重継承をやめ，メソッドを静的結合とすることで実行効率を向上させるクラスである．またスタティッククラスは，継承およびメソッドの結合以外の点(バージョン管理など) では，通常のクラスと同様に利用可能である(表1)．

Extension of Class in OZ++: Object-Oriented Distributed Systems Environment
Yutaka Niibe (Mitubishi Research Institute, Inc.*),
Hideyuki Otokawa (Sharp Corporation*),
Yoichi Hamazaki (Electrotechnical Laboratory),
Takayuki Suzui (Sharp Business Computer Software, Co., Ltd.*),
Hidetada Hirakawa (Nihon Unisys, Ltd. *),
Michiharu Tsukamoto (Electrotechnical Laboratory)
*: Researcher of Evaluation of Open Fundamental Software Technology Project in Information-technology Promotion Agency, Japan

4 レコード

4.1 必要性

OZ++のようにユーザが新たな型を定義する枠組みがクラスのみというオブジェクト指向言語においては，クラス定義が二つの目的で利用されることになる．一つはオブジェクトをモデル化するためであり，もう一つは言語の持つ基本型を拡張する抽象データ型を定義するためである．この後者の場合には，定義されたクラスから生成されるインスタンスはオブジェクトではなく単なるデータであり，このようなオブジェクトではないものを言語においてオブジェクトと同様に表現することは不自然である．

4.2 仕様

レコードは，OZ++においてこのような抽象データ型を表現するためのものである．レコードはデータであるメンバとメンバを操作するためのオペレータを持ち，メンバ，オペレータに対してはクラスのようなアクセス制御はなくすべてレコードの外部に公開されている．オペレータはスタティッククラスのメソッドと同様にコンパイル時に静的に結合される．また，レコードは基本型を拡張するためのものであるという意味から，メンバの型を基本型およびレコードのみに制限し，メソッドの引数および返り値では参照ではなく値として取り扱う．さらに，レコードから生成されるものは，クラスまたはスタティッククラスから生成されるオブジェクトのための部品となるような粒度の小さいものであるため，ネットワーク上で分散透過にアクセスすることはできない(表1).

表 1: クラス，スタティッククラス，レコードの比較

	class	static class	record
継承	多重継承	×(可能)	×
binding	dynamic	static	static
引数/返り値	reference	reference	value
分散透過なアクセス	○	○	×

5 まとめ

現在開発を進めているオブジェクト指向分散環境OZ++のクラスについて，OZ++において利用可能な3種類のユーザ定義の型について述べた

OZ++では，クラスを定義する場合に，オブジェクトの粒度および実行効率に応じてクラスとスタティッククラスの二種類のものを使い分けることが可能である．さらに，基本型を拡張する抽象データ型の定義は，クラスではなくレコードによって行なう．これらによって，目的に応じた柔軟なプログラミングのための枠組を提供する．

本研究において熱心な討論を頂いた，藤野 晃延(富士ゼロックス情報システム)，千葉 滋(東京大学理学部)両氏に感謝する．

本研究は，情報処理振興事業協会「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] 塚本他：「オブジェクト指向開放型分散システムOZ+の研究開発」，電総研彙報，vol. 56，No. 9，Sep. 1992

[2] 西岡他: 「オブジェクト指向分散環境 OZ++ の言語の基本設計」，情報処理学会第46回全国大会，Mar. 1993

[3] 濱崎他: 「オブジェクト指向分散環境 OZ++ の基本設計」，SWoPP '93，Aug. 1993

[4] 吉屋他: 「オブジェクト指向分散環境 OZ++ のクラス管理方式」，情報処理学会第47回全国大会，Oct. n1993