

オブジェクト指向に基づいた分散プログラム開発環境

7H-4

何 千山 田中英彦
東京大学工学部

1 はじめに

計算機ネットワーク技術の発達により、ワークステーションはネットワークによって連結され、リモート計算機で提供される機能(データ、プログラム等)をアクセスすることは十分可能になってきた。非常に多くの計算機機能が提供されている場合、それをユーザの立場がどのように利用することができるか、また、資源管理の面から見ると、どのようにこれらの膨大な量の計算機機能を扱うかは、解決しなければならない問題である。ユーザにとって既にある分散資源を組み合わせる自由を利用できるプログラム開発環境をサポートするためのシステムは是非必要である。本論文で考察するシステムでは、以上の問題を解決するために、計算機が提供する機能をオブジェクトとして抽象化し、サービスベースで管理する。さらにユーザは分散型言語を使用してこれらのオブジェクトを組み合わせることにより、容易に分散応用システムを構築することができる。

2 オブジェクト指向に基づいたサービスのモデル

計算機がユーザにサービスを提供するという事は、「あるデータに対し、何らかの作用(アクション)を施し、結果を返す」ということである。さらに、この結果を新たなデータとして、それに対して別の作用を施していくこともある。オブジェクト指向の概念で言うと、このデータとそれに対する作用を一つのオブジェクトとすることができる。しかし、このオブジェクトは受動的なものであり、オブジェクトに対する処理、すなわちプロセスが必要になる。プロセスでオブジェクトを実行することにより処理が進む。ここでは、オブジェクトとプロセスを一つのサービスと呼ぶことにする。ユーザはサービスを組み合わせることにより応用システムを構築することができる。上述のモデルは、通常のプログラミング言語に対しても適用できる。したがって、サービスを受けるために、プログラムを書く作家と同様の過程が必要となる。このため、サービスの記述や要求のための言語が必要となるが、上述のモデルは、プログラミング言語との自然な融合を示すものである。

分散環境では、特に解決しなければならない問題は分散透明性の問題である。ユーザにサービスを提供する場合、なるべくデータ、プログラムの存在する場所などをユーザに見せなく、そのかわりに、システムはそれを管理する。ここで考える一つの方法としては、オブジェクトは、データとそれに対する作用のほかに、データ、作用の属性も内含する。すなわち、各具体的なデータについて、そのデータの存在場所、データタイプなどの属性、及びそのデータに対する作用、作用の属性などを含めて一つのオブジェクトとして抽象化して記述することにより、分散資源の内部の細かい実装については隠され、その外部的な機能、仕様だけユーザに見せることができる。さらに、オブジェクトの組み合わせるにより、適応なサービスを提供することができる。そうすると、オブジェクトを記述し、オブジェクトを組み合わせるために、適応な分散言語と処理系が必要となる。このように、オブジェクトを効率的に管理するシステムとこれらのオブジェクトを扱う分散言語により、よりよい分散プログラム環境をユーザに与えることができる。

3 システムの構成

従来、われわれがサービスベースシステム(SBS)を開発しつつある[1]。SBSは、ユーザに便利、有効な分散環境を提供することを目的とする。既存の計算機の機能、計算機で蓄えられるデータやプログラムを組み合わせ、容易に新しい機能をユーザに提供することができるのが、SBSの特徴である。オブジェクト指向の概念をSBSに導入し、さらに、SBSの上に分散したオブジェクトを容易に扱うことができる分散プログラミング言語を設置することにより、分散プログラム環境を構築する。(図1)

サービスベースで管理するオブジェクトは、三つのビューに分けられる。それぞれは外部ビュー、概念ビュー、内部ビューである。外部ビューは、個々の応用プログラム(ユーザ)の立場から見たオブジェクトのビューであり、このビューでは、オブジェクトをサービスの形式でユーザに提供し、分散性を意識させない、概念ビューは、データ、作用を抽象化し、属性を含むオブジェクトのビューである。分散性はここで吸収する。内部ビューは、概念ビューを計算機の世界に写像し、データ、作用、プロセスをいかにして計算機上に実現するか、すなわち、データ、作用、プロセスの実体のビューである。

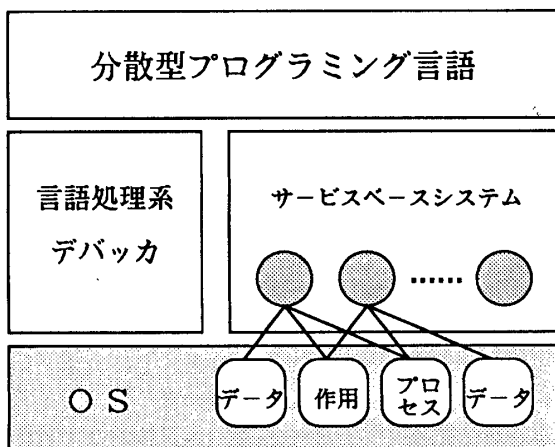


図 1: システムの構成

サービスベースは、以上の三層ビューでオブジェクトを管理する。内部ビューでは、実際のデータ、作用、プロセスを既存の OS で管理すれば、システムの構築は行いやすくなる。外部ビューは、応用プログラミング言語の形式に依存する部分があるので、そのインタフェスをうまく作らなければならない。概念ビューでオブジェクトの記述については次の章で述べる。

4 オブジェクトの記述と管理

各ノードでは、自ノードで利用する分散オブジェクトについてだけ記述し、オブジェクトディレクトリに置く。このように、一部のノードに情報が集中したり、また、不必要な情報まで持つようなことはなくなる。応用プログラムからオブジェクトを呼び出す場合、システムはオブジェクトディレクトリの情報に基づいて実際のデータと作用を探す。

OS で管理するデータ、作用は独立ではなく、何らかの形で組み合わせることができる。データを中心にし、その属性、施される作用を記述することにする。あるオブジェクトは、特定のクラスに属し、自分の属しているクラスの記述に従う。クラスの階層関係、クラスのインスタンス変数、メソッドの継承により、効率的に管理することができる。

記述例として UNIX のファイルシステムを考えてみる。UNIX のファイル（データ）の種類を階層的に分類することができる。例えば、図 2 で示すように、C プログラムファイル或いは latex ファイルは、text ファイルの一つの種類である。text ファイルの属性、text ファイルに対して施されるコマンド（作用）は、C プログラムファイルにも同じように適用である。C プログラムファイルでは、自分の特有の属性、施される作用（例

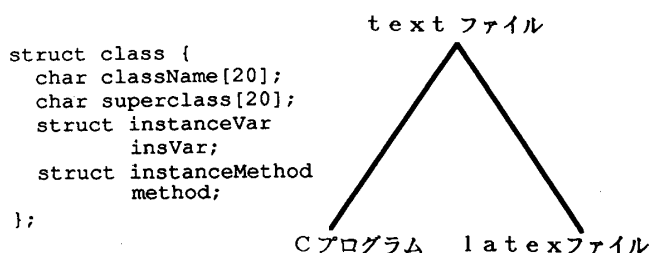


図 2: クラスと継承

例えば、コンパイルできる)を追加すれば、text ファイルの全部の特性を継承することができる。このように、クラスの導入することにより、大量の分散資源を簡単に記述することができる。現在、C 言語で UNIX の上で記述方法を詳しく検討している。オブジェクトディレクトリのクラス構成は、図 2 のようになり、インスタンス変数は属性に相当し、メソッドはデータに施す作用である。

5 分散プログラミング言語

以上で述べたように、データ、作用をオブジェクトとしてサービスベースで管理し、ユーザは分散プログラミング言語を使用してそれを扱う。データ、作用を記述するために、オブジェクト、プロセスを扱うために、分散型プログラミング言語とその処理系は重要である。サービスベースで管理するオブジェクトはネットワークに分散され、各オブジェクトは並列に実行される場合もある。したがって、分散プログラミング言語は特に並列プロセス記述機能とオブジェクト間通信メカニズムをサポートしなければならない。また、分散型言語のデバッグ環境も必要である。現在、上述のシステム構成に適する言語の仕様を設計している。

6 おわりに

本論文では、オブジェクト指向の概念に基づき、よい分散プログラム開発環境を構築するためのサービスモデル、システムの構成を考察した。現在、オブジェクト指向に基づいた分散システムは各所で開発されつつあるが、このシステムの特徴は、サービスの組み合わせることができ、システムの拡張性があることである。今後は、今回考察したシステムモデルの詳しい仕様を決定し、実験システムの実装を進めていく予定である。

参考文献

- [1] 何 千山、田中英彦、「LAN に適したサービスベースシステムの構成と実装」、情報処理学会マルチメディアと分散処理研究会、1988.5.20.