Ⅴ 分散オブジェクト

学習目標
① 汎用的オブジェクトの意味を理解させる。
② 分散オブジェクトの概要を理解させる。

1 汎用的オブジェクト

これまで説明してきたオブジェクト指向は、「オブジェクトに対してメッセージを送り、オブジェクトは受け取ったメッセージによって依頼された操作（メソッド）を実行する」というものであった。このような考え方は、「古典的オブジェクト」と呼ばれる。

古典的と称される理由は、メッセージを送るオブジェクトは、必ずクラスに属していなければならないという制約をもっているからである。そのため、オブジェクトと関数の世界は明確に分離されていた。

現在は、この制約を打破した「汎用的オブジェクト」という考え方が登場している。これは、メッセージをオブジェクトに送るという概念を汎用化し、一般の関数も同様に扱えるようにしたものである。すなわち、「オブジェクトにメッセージを送るとオブジェクトはメソッドを実行し結果を返す」というオブジェクトの概念と、「関数に引数を与えて呼び出すと関数は処理を実行し結果を返す」という関数の概念を汎用化し、「入力パラメタを演算に与えると演算は実行され出力パラメタを返す」と定義し直す。これによって、オブジェクトと関数の世界を一体化することができる。

例えば、Class1というクラスにprintfというメソッドがあり、Class2というクラスにもprintfというメソッドがあるものとする。一方、Ｃ＋＋言語にも、printfという関数が存在する。

この場合、三つのprintfを区別するためには、古典的オブジェクトでは、リストⅤ－１のようにオブジェクト名を指定したり、スコープ演算子を使用したりすることになる。

```
Class1 o_Object1;
Class2 o_Object2;
o_Object1.printf("私はクラス１のメソッドです。¥n");
o_Object2.printf("私はクラス２のメソッドです。¥n");
::printf("私は関数です。¥n");
```

リストⅤ－１　古典的オブジェクト

一方、汎用的オブジェクトでは、三つのprintfは、同じ名前なので一つのグループとして定義する。そして、これを総称定義と呼ぶ。汎用オブジェクトでは、三つのprintfを区別するために、リストⅤ－２のようにprintfの第１引数に工夫をする。そして、printfが呼ばれたら、総称定義の三つの中の一つに特定する。特定の方法は、もし第１引数がオブジェクトであれば、それはオブジェクトのメソッドであると考え、もし第１引数がオブジェクトでなければ、それは関数であると考えるというものである。

```
Class1 o_Object1;
Class2 o_Object2;
printf(o_Object1,"私はクラス１のメソッドです。¥n");
printf(o_Object2,"私はクラス２のメソッドです。¥n");
printf("私は関数です。¥n");
```

リストＶ－２　汎用的オブジェクト（イメージ）

汎用的オブジェクトを実現したのが、ＣＬＯＳ（Common Lisp Object System）であり、Lisp本来の関数の世界と、オブジェクトの世界を一体化することに成功している。

2 分散オブジェクト

分散オブジェクトとは、ネットワーク上に分散しているオブジェクトのことである。分散オブジェクトにおいては、複数のオブジェクトのアドレス空間が同じであるか異なるかにかかわらず、オブジェクト間のメッセージ交換が可能であり、相手のオブジェクトの所在を全く意識することがないというのが最大の特徴である。そこで、ネットワークが必須のものとなった現在のコンピュータの世界では、必要不可欠な技術であるといえる。

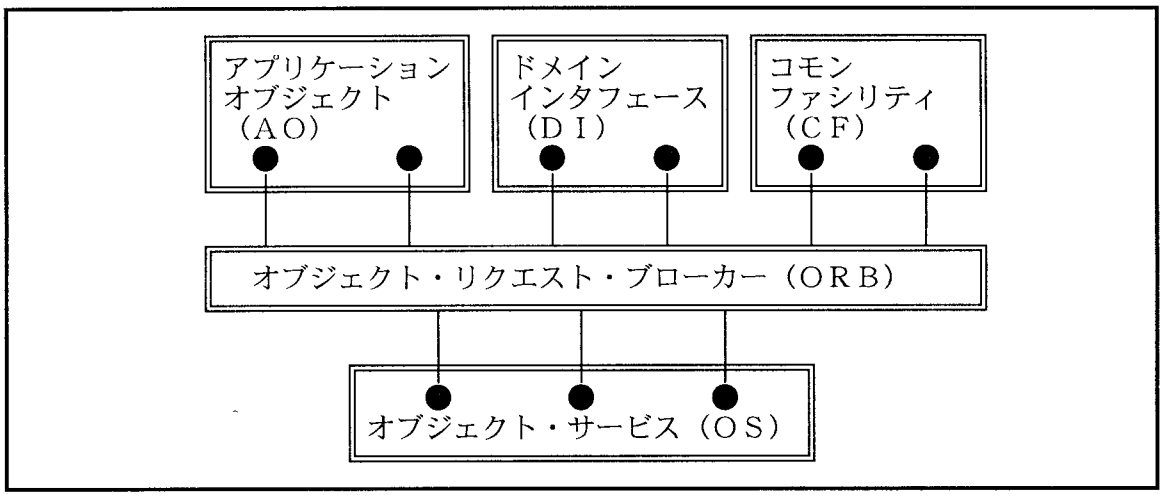
例えば、データベースというサービスを、分散オブジェクトで実装したとする。すると、データベースサービスがネットワーク上のどこにあるかということを意識せずに、データベースのサービスを自由に使用することができる。

ところで、分散オブジェクトについては、現在、ＣＯＲＢＡとＤＣＯＭという二つの大きな流れがある。これら二つの技術について説明する。

(1) ＣＯＲＢＡ

ＣＯＲＢＡ（Common Object Request Broker Architecture）とは、ＯＭＧ（Object Management Group）が制定する分散オブジェクト技術の標準の名前である。ＯＭＧは、１９８９年に設立された企業・大学・団体から構成される非営利目的の組織であり、分散オブジェクト環境におけるインフラストラクチャの提供をその目的としている。その発足メンバには、Ｓｕｎ、ＨＰ、Ｕｎｉｓｙｓといった有力なＵＮＩＸワークステーションメーカが含まれている。

図Ｖ－１にＣＯＲＢＡのアーキテクチャを示す。



図Ⅴ－１　ＣＯＲＢＡのアーキテクチャ

次に、各部分について説明する。

イ　オブジェクト・リクエスト・ブローカー（ＯＲＢ）

オブジェクトに関するリクエストの作成やレスポンスの受取といった通信機能を提供する部分である。

ロ　オブジェクト・サービス（ＯＳ）

ＣＯＲＢＡサービスとも呼ばれている。オブジェクトのインタフェースをもつサービスが集合したものであり、オブジェクトの実現のための基本的な機能を提供する部分である。すでに実装されたＯＳもあれば、まだ実装されていないＯＳもある。以下に示すのは、既にＯＭＧの会員企業の手により実装されたＯＳである。

・ ネーミング
・ イベント
・ 永続オブジェクト
・ ライフ・スタイル
・ コンカレンシー（同時実行性）
・ リレーションシップ（関係）
・ トランザクション
　　：

ハ　コモン・ファシリティ（ＣＦ）

ＣＯＲＢＡファシリティとも呼ばれている。アプリケーションにとって役に立つ汎用的な機能を提供するオブジェクトの集合の部分である。次に説明するＤＩが業界別の視点に基づいているのに対し、ＣＦは技術的な視点に立っているところが相違点である。

以下に示す４種類のファシリティに分けられる。

・ ユーザ・インタフェース
・ インフォメーション・マネジメント

・ システム・マネジメント
・ タスク・マネジメント

ニ ドメイン・インタフェース（ＤＩ）

業界別の視点に基づいたファシリティの集合の部分である。以下に示すようなファシリティがある。

・ フィナンシャル・ドメイン
・ ビジネス・オブジェクト・ドメイン
・ マニュファクチャリング・ドメイン
：

ホ アプリケーション・オブジェクト（ＡＯ）

従来のアプリケーションプログラムに相当するオブジェクトの集合の部分である。

このように、ＣＯＲＢＡは数多くの要素から構成される大規模なアーキテクチャであることがわかる。このＣＯＲＢＡに従って、ＯＭＧの参加企業は、ＯＲＢ（Object Request Broker）製品を作成しており、その数は１９９７年現在で約２０に達している。

ところで、コモンファシリティ（ＣＦ）の一部である複合ドキュメントの規格として決まったのが、ＯｐｅｎＤＯＣである。これについては、既に製品化が進行中である。

しかし、ＣＯＲＢＡのオブジェクト・サービス（ＯＳ）はまだ未完成の状態にあり、しかも、現在既に作成されたサービスも動作が重すぎるという批判がある。そのため、例え今後、すべてのサービスを実装したとして、それがはたして実用的なものになるかどうかは疑問の残るところである。

(2) ＤＣＯＭ

ＤＣＯＭ（Distributed Component Object Model）とは、マイクロソフトが制定するＷｉｎｄｏｗｓのための分散オブジェクト技術の名前である。

その説明をする前に、ＣＯＭ（Component Object Model）について説明しておく必要がある。ＣＯＭは、アプリケーションの外に存在するオブジェクトへのアクセスを可能にするための技術であり、マイクロソフトが複合ドキュメントの実現のために制定したＯＬＥ（Object Linking and Embedding）規格の中心をなすものである。

ＯＬＥにより、Ｗｉｎｄｏｗｓ上の二つのプログラムの間で、ドキュメントのカット・アンド・ペーストが容易に行えるようになった。ＯＬＥはその後、ＯＬＥ２へと発展し、更にインターネットの急速な普及に歩調を合わせて、ＡｃｔｉｖｅＸへと発展していったが、ＣＯＭの重要性は今も変わらない。

ただし、ＣＯＭには、アプリケーションからアクセスするオブジェクトは、アプリケーションと同じコンピュータ上にある必要があるという制約がある。

一方、ＤＣＯＭは、ＣＯＭが発展したものである。ＣＯＭのもっていた制約を取り払い、オブジェクトは、アプリケーションと異なるコンピュータ上にあるものでもよくなった。これによってＤＣＯＭを用いると、ネットワーク経由でオブジェクトへアクセスすることが可能になった。

また、ＤＣＯＭは既にＷｉｎｄｏｗｓＮＴ４．０から製品に組み込まれており、ＣＯＭからの拡張が自然なため、今後Ｗｉｎｄｏｗｓパソコン上で普及することが予想される。

ところで、ＤＣＯＭは、Ｗｉｎｄｏｗｓというプラットホームに大きく依存しており、Ｗｉｎｄｏｗｓ以外のプラットホームでは、動作しない技術であると誤解されていることがある。しかし、ＤＣＯＭもＡｃｔｉｖｅＸも、プラットホームには依存しない技術である。

そこで、マイクロソフトは、ＡｃｔｉｖｅＸをあらゆるプラットホームに普及させようと活発な活動を行っている。