2S－1　インタフェース指向によるビジネスオブジェクトフレームワークの実現アーキテクチャ

松塚 貴英, 金谷 延幸, 原 裕貴, 大久保 隆夫, 上原 三八
株式会社富士通研究所

1　はじめに

柔軟な企業イントラネットを構築するためのビジネスオブジェクトフレームワークの実現アーキテクチャを提案する。このためには、既存のデータベースや分散オブジェクト基盤をビジネスオブジェクトの概念で統一的に扱える枠組が必要である。これをインタフェース指向のアプローチで実現した。

2　ビジネスオブジェクトフレームワーク

我々は、インターネット上で多人数による業務作業の効率化を実現する次世代エンタープライズシステムの研究を行なっている[1]。このシステムでは、業務を規定する企業モデルを定義し、そのモデルに基づいて業務支援を行なう。この際、メインフレームで管理される基幹システムの情報と部門サーバで管理される業務情報をリンクするなど、分散環境で既存資産を積極的に活用することが求められる。

これを実現するフレームワークとして、ビジネスオブジェクトに基づくフレームワークが議論されている[2]。ビジネスオブジェクトは、人や組織、タスクなどの業務の要素を表現し、システムはビジネスオブジェクトのインタラクションにより処理を行なう。

業務を実現するアプリケーションはビジネスオブジェクトの操作により実装する。ビジネスオブジェクトのデータベース透過性やサーバ透過性を高めることにより、業務アプリケーションの実装コスト、保守コストを下げることができる(図1)。

3　インタフェース指向

ビジネスオブジェクトフレームワークを実現するためには、さまざまな既存のアプリケーションやデータベースを統合しながら、それらに格納されているデータを透過的に扱えることが必要である。

これを実現するアプローチとして、「インタフェース指向」を提案する。インタフェース指向とは、オブジェクト指向の上位概念であり、ある機能を表すインタフェースを定義し、それをオブジェクトに実装することによってそのオブジェクトがサポートする機能を表現するものである。

An Architecture of Business Object Framework with Interface Oriented Paradigm
Takahide MATSUTSUKA, markn@flab.fujitsu.co.jp
Fujitsu Laboratories Ltd.

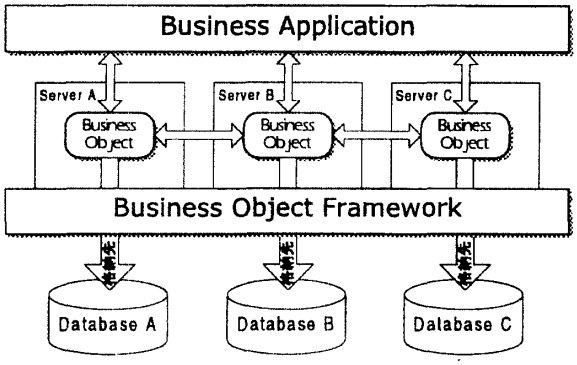

図 1: ビジネスオブジェクトフレームワーク

1. インタフェースは、メンバに抽象メソッドのみを持つクラスで表現される。実装を持たず、インスタンス化はできない。
2. クラスは、1個以上のインタフェースを実装する。
3. オブジェクトをアクセスする際は、必ずインタフェースによりアクセスする。

オブジェクト指向では通常、継承を使って再利用を行ない、これにより機能がオブジェクトに集約されていく。一方、インタフェース指向では機能をカテゴライズし、インタフェースに分散する。

これにより、通常のオブジェクト指向に対して次の利点がある。

- オブジェクトの使用者は、インタフェースでのみアクセスが許されるため、自然なオブジェクトのカプセル化が実現する。
- オブジェクトの機能を実行時に判別できる。あるオブジェクトのインタフェースから、キャストによってそのオブジェクトが他にサポートするインタフェースを得るが、この際、キャストに成功するか否かで機能を問い合わせることができる。

4　実現アーキテクチャ

前節で述べたインタフェース指向のパラダイムを用いたビジネスオブジェクトフレームワークの実現アーキテクチャとして、次のような構造を提案する(図2)。

このアーキテクチャは、3つの層に分かれ、以下のように構成される。

論理層 実現層の基底オブジェクトを使ってさまざまな種類のビジネスオブジェクトを実装する。ビジネ

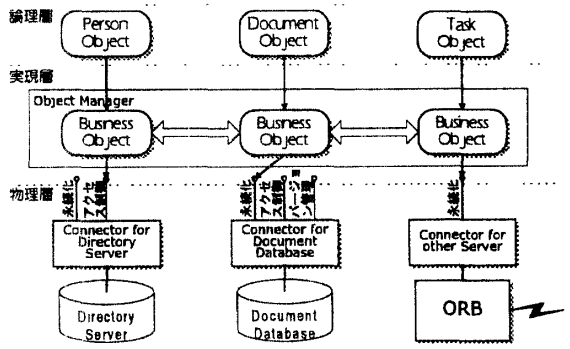

図 2: 実現アーキテクチャ

スオブジェクトにはメソッドにより動作が実装されるが、オブジェクトが持つデータ (プロパティ) へのアクセスは実現層に委譲され、透過性の管理などが行なわれる。アプリケーションはこの層のオブジェクトを利用して業務論理を処理する。

実現層 ビジネスオブジェクトの基底オブジェクトを実装する。ビジネスオブジェクトのプロパティの操作は、ここで永続性や透過性に関わる変換が行なわれて、物理層のオブジェクトに委譲される。
また、これらの基底オブジェクトと物理層を管理するオブジェクトマネージャを実装する。オブジェクトマネージャは、オブジェクトが実際に格納されているコネクタを判別したり、コネクタの動的登録を行なったりする。

物理層 使用するデータベースや分散オブジェクト基盤ごとのコネクタを実装する。コネクタはコネクタオブジェクト (実現層の基底オブジェクトに 1 対 1 対応する) とコネクタマネージャ(コネクタオブジェクトを管理する) から構成され、コネクタオブジェクトは自分が提示できるインタフェースを 1 つ以上実装する。プロパティの操作はここでデータベースやアプリケーションとの入出力となる。

あるビジネスオブジェクトは次のように実現される。まず、動作とデータを分離し、動作をメソッドとして論理層にインプリメントする。この動作は、プロパティの操作などのより原子的な操作に変換され、実現層の基底オブジェクトのメソッドを呼び出す。基底オブジェクトでは、その動作が可能かどうかをコネクタオブジェクトが実装しているインタフェースから判断し、可能であればコネクタオブジェクトのメソッドを呼び出す。インタフェースが実装されておらず、動作が不可能の場合は論理層に例外を返す。

コネクタオブジェクトが実装するインタフェースは、プロパティ操作、永続化、アクセス制御、バージョン管理機構などであり、あとから追加が可能である。

5 実装例

これまで述べたアーキテクチャに従ったビジネスオブジェクトとコネクタの実装例を図 3に示す。

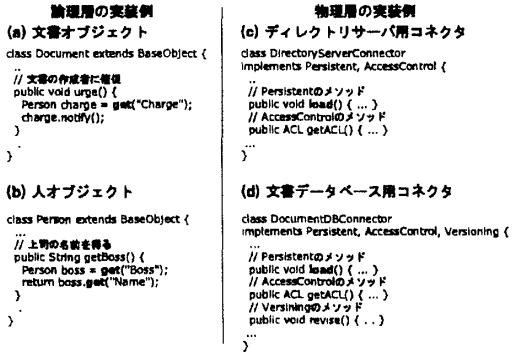

図 3: 実装例

この例では、人オブジェクトはディレクトリサーバに保存され、文書オブジェクトは文書データベースに保存されているが、論理層の実装ではそれは隠蔽され、実現層の基底オブジェクトのメソッド **get** が呼ばれている。すると基底オブジェクトでは、保存先のコネクタオブジェクトの永続化インタフェースを取得し、**load** メソッドを呼ぶことによって **get** を実現する。このように、論理層と物理層の実装が分離し、透過性・保守性が高くなっていることが分かる。

また物理層に新たなデータベース用のコネクタを実装し、オブジェクトマネージャに登録することにより、論理層の実装を変更せずにデータベースを動的に追加することも可能である。追加されたデータベースは、新たなビジネスオブジェクトを作成したり、オブジェクトのコピーを作成したりする際にオブジェクトマネージャにより自動的に使用される。

6 おわりに

ビジネスオブジェクトフレームワークの実現アーキテクチャをインタフェース指向で構築する手法を提案した。現在、これまで述べたアーキテクチャに従い、分散企業情報システムのプロトタイプを実装している。今後は、コネクタにキャッシュやレプリケーションの機能を実装することによって、マルチサーバ環境下においてパフォーマンスや信頼性を向上させる研究を行なう予定である。

参考文献

[1] 上原 三八, 大規模共同作業支援のためのモデル化と実現アーキテクチャ, 電子情報通信学会論文誌, Vol.J80-D, No.7, 1997.

[2] OMG Business Object Domain Task Force BODTF-RFP 1 Submission, Business Object Facility Revision 1.0, Data Access Technologies, 1997.