

平成 7 年度

ビジネスオブジェクトの動向
～姿を見せ始めたビジネスオブジェクト～

平成 8 年 3 月

新技術動向研究部会

オブジェクト指向に関する調査研究WG

【目 次】

WGメンバー

はじめに

- (1) ビジネスオブジェクトの進展
- (2) 関連技術の動向

オブジェクト指向WG研究部会活動経緯

第1章 オブジェクト指向技術の動向

- 1.1 オブジェクト指向開発方法論
 - (1) はじめに
 - (2) 各世代の概要
 - (3) Fusion Method(第2世代)について
 - (4) Unified Method(第3世代)について
 - (5) まとめ
- 1.2 オブジェクト指向DB技術
 - (1) オブジェクト指向データベースの概念と適用範囲
 - (2) オブジェクト指向データベースの性能について
 - (3) OODBの課題
 - (4) オブジェクト指向データベース製品の動向
 - (5) CORBA環境でのオブジェクト指向データベース
- 1.3 オブジェクト指向CASE
 - (1) CASEツールとオブジェクト指向
 - (2) オブジェクト指向CASEの現状
 - (3) 今後のオブジェクト指向CASEへの期待
- 1.4 オブジェクト指向言語 Java
 - (1) Java の特徴
 - (2) Java 開発支援ツール
 - (3) Java の可能性

第2章 ビジネスオブジェクトの最新動向

- 2.1 ビジネスオブジェクトの動向
 - 2.1.1 ビジネスオブジェクトの標準化動向
 - 2.1.2 ビジネスオブジェクトによるシステム開発
- 2.2 基幹業務パッケージソフトへのビジネスオブジェクト適用

(1) ビジネスオブジェクト適用の背景

(2) ビジネスオブジェクト適用のパッケージソフト「BPCS Client/Server」

はじめに

オブジェクト指向WG研究部会は平成6年度に発足し、エンドユーザーの立場からの「オブジェクト指向技術とは」どのようなものかについて検討を加え、平成7年3月にその結果を中間報告としてとりまとめた。今年度はこの結果を踏まえ、ユーザーの立場から最も期待されるオブジェクト指向の技術である「ビジネス・オブジェクト」を中心に関連技術の最新動向を調査、検討した。

(1) ビジネスオブジェクトの進展

われわれが「ビジネス・オブジェクト」に着目し、その概念、メリット、技術動向を勉強し、臆気ながら理解するに至ってから1年ほどしか経過していない。にもかかわらず、その進展の勢いは目を見張るものがある。例えば、標準化の面では、OMG(オブジェクト マネージメントグループ)において OMG Common Facilities のRFP-4としてビジネスオブジェクトの相互運用性を保証するための標準モデルが提案されている(図1)。

図1 OMG Common Facilities RFP-4で提唱されているAPモデル

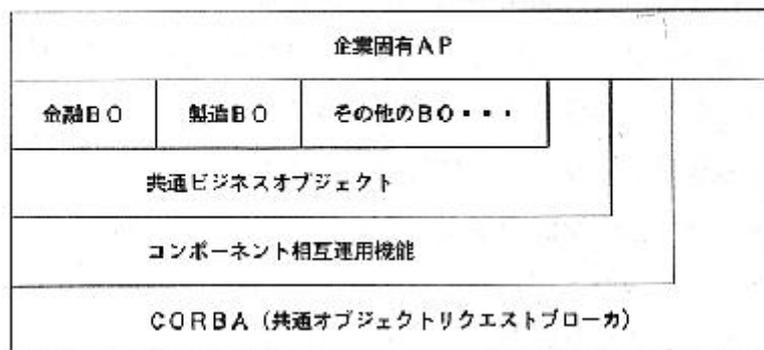


図1

このモデルによれば、各業界ごとのビジネスオブジェクト(BO)の標準が相互運用性を保ったCORBA 環境とコンポーネント相互運用機能の上に提供され、各企業固有のアプリケーションプログラム(AP)はBOを必要に応じて組み合わせ、部分追加、変更することで構築可能となる。BOの規定は製造業の分野で特に進んでいる。

具体的には SEMATECH 社の提唱したCIM(Computer Integrated Manufacturing)業務のフレームワークをベースに標準化が進められている。金融業、通信業のBOの標準化のグループも発足し活動中である。

これら標準化の進展を待ちきれないかのように、オブジェクト指向のプラットフォーム上でビジネスオブジェクトを開発し販売するビジネスがすでに始まっている。

本報告で紹介しているPBCS(SSA社)が世界で初めてビジネスオブジェクトを適用した統合業務パッケージと銘打って昨年末販売を開始したのはその代表例の一つと言えるが、このような傾向は加速度的に広まるものと予想している。

(2)関連技術の動向

ビジネスオブジェクトに関する標準化、製品化がこのように加速度的に進展してきたのは、オブジェクト指向技術を支える関連技術がようやく成熟し裾野を広げてきたことと呼応している。その意味から、関連するオブジェクト指向技術の動向調査を行った。

- ・ オブジェクト指向開発方法論とCASEについては、この分野を引っ張ってきた Booch と Rumbaugh が方法論の統合に合意し、統合方法論に基づく CASEの登場が近いこと、また、標準化に関してはOMGのオブジェクト分析設計SIG(OAD-SIG)の精力的な活動によって、50種類を越すと言われている方法論の相互変換を実現するメタモデルの標準化が今年の秋にも完成する予定であること、などが最も大きな進展であり、オブジェクト指向の分析設計、開発がシームレスに、しかも安心して行える時代の到来を感じている。

- ・ オブジェクト指向DBについては代表的なオブジェクト指向DBMSの製品動向調査を行っているが、小規模でデータ構造が複雑な分野での適用の他に、大規模な事務処理分野への適用を意識したRDBMS機能への接近が顕著な特徴である。オブジェクト指向、分散化の傾向の中でOODBとRDBを適材適所にうまく活用できる製品と標準が整いつつある。

- ・ もう一つ見逃せない技術動向として、インターネットの爆発的な普及とオブジェクト指向を結びつける鍵となるオブジェクト指向プログラミング言語 Java の出現があり、急遽、調査の項目として追加し動向のまとめを行った。

上記のとおり、ビジネスオブジェクトとその関連技術の進展はこの1年全く目を見張るものがあり、調査範囲を絞り込めず、オブジェクト指向WGの報告も若干焦点の定まらない調査報告になったことを反省している。

しかし、調査の結果から断言できることは、オブジェクト指向技術は、その頂点に立つ「ビジネスオブジェクト」さえも、もはや机上の空論や実験室での出来事ではなく、現実のビジネスの世界で堂々と使われる技術や製品になっているということである。この1年は そのことを再確認する1年であったとも言える。同時に、オブジェクト指向技術の動向を調査するという目的で発足した本検討部会の役目は、これで完了できたのではないかと考えている次第である。

最後に、本調査に快くご協力いただいた日本電子計算の高橋氏、日本IBMの沢田氏をはじめ関係各位に本紙を借りて厚くお礼を申し上げます。

〈参考文献〉

- ・ オブジェクト分析と設計(オブジェクト指向 21 の手法の解説と徹底比較)、アンドリュー T. F. ハット編、トッパン、1995
- ・ Object World Expo/Tokyo '95 予講集、1995 年 11 月

第1章 オブジェクト指向技術の動向

1.1 オブジェクト指向開発方法論

(1)はじめに

オブジェクト指向の方法論の現状に関しては急速に多くの手法が提案され、その種類は50を越え、さらに、それぞれが発展を遂げている段階であり、まさに混沌をきわめている状況といえる。

そこで、この方法論の章では、オブジェクト指向方法論の各世代の概要を述べた後に、各世代の代表的な手法について簡単に述べることにしたい(なお、ここで述べる世代の捉え方は羽生田氏の考え方を流用した)。

(2)各世代の概要

1)第1世代

とにかくオブジェクト指向で設計してみるという思想のものと方法論であり、開発経験に基づくというよりも、概念レベルでの手法設計で、実装より上流に重点をおいている。記述するダイアグラムもER図を拡張したオブジェクト図が示されているだけのものが多い。初期のBooch、Cord/Youdon がこれに相当する。

2)第1.5世代

ダイアグラムの表現のなさや、開発手順の不明確さなど、第1世代への不満に対応するべく、「モデリング」という観点を重視し、モデル表現能力を充実するため、視点の異なる複数のダイアグラムを採り入れたり、各フェーズの作業手順と成果物の規定を決めた方法論が開発された。OMT、Shiaer/Mellor、Objectory がこれに相当する。

* OMT:Rumbaugh 等により開発され、内容も実際の開発事例に関する記述に富み、日本語訳の出版も早く、現在、日本で最もメジャーな方法論。

* Shiaer/Mellor:大規模リアルシステムに適用されていた手法をオブジェクト指向に体系化した。

* Objectory:Jacobson のユースケースの考え方を含んでいる。

3)第2世代

第1.5世代では大規模システム開発のための枠組みはモデル概念としては不明確であった。第2世代ではクラスカテゴリ、サブシステム、ユースケースなどを組み込み、上記の問題

への対応とした。また利用経験に基づく改良を行い、より実用化を目指した。Fusion、OMT2がこれに相当する

* Fusion:名前が示すとおり、OMT、Booch、CRCなど、さまざまな手法を融合し、HPが実開発を基に提案した。

* OMT2:OMTの利用経験より、動作記述としてシナリオや事象トレースを採り入れたり、サブシステム分割が可能になるように改良した。

4) 第3世代

従来の方法論の集大成として、記法も仕様もオープンし、分析と設計を一貫してサポートする業界標準を目指すものである。OMTのRumbaugh、ユースケースのJacobsonがBoochのRational社に移ったことにより、より現実化された。その手法であるUnified Methodのバージョン0.8がまもなく発表される予定である。詳細についてはUnified Methodの項で触れたい。なお、記法についてはOMGの中でも検討が進められている。

(3) Fusion Method(第2世代)について

Fusion法はオブジェクト指向ソフトウェア開発に対して体系的なアプローチを提供するために開発された手法で、要求仕様からプログラムの実装を行うフェーズまでを提供している。この方法論は、OMTからオブジェクトモデルとプロセスを、CRCからオブジェクトの交信を、Booch法からオブジェクトの可視性を、FORMAL METHODSから事前・事後条件の考え方を採り入れることにより実現している。

ここでは各フェーズの概要と主な成果物のイメージについて簡単に紹介したい。

1) 分析フェーズ

分析フェーズではシステムがどのように行うかではなく、何を行うのかを分析する。システムの実装方法と分離するためには、実装するマシン側からの観点ではなく、ユーザーの観点から要求される。そのため分析では、システムの外部に見える振る舞いに関する問題領域に着目している。

①オブジェクトモデル

問題領域に存在する概念とそれらの間の関係をとらえることである。クラス、属性、クラス間の関係を示すものであり、OMTのそれとほぼ同様のものである。

②オペレーションモデル

外界とシステム間に発生するシステムオペレーション(入力イベントとそれによる影響)の振る舞いをモデル化したもので、各システムオペレーションの概要、参照項目、更新項目、出力先、事前・事後条件などを記述したモデルである。

③ライフサイクルモデル

システムオペレーション間のシーケンスを定義したものであり、このシーケンス順でなければ、システムは無視し状態を変更しないことになる。基本的にはシナリオより作成される。

2)設計フェーズ

設計フェーズでは分析によって作成された抽象定義を満たすソフトウェア構造を設計する。

①オブジェクト交信グラフ

それぞれのシステムオペレーションを実現するためにオブジェクト間でやり取りされるメッセージの順番を決めるものである。このグラフを作成することにより、各オブジェクトのメソッドが洗い出される。

②可視グラフ

オブジェクト交信グラフではすべてのオブジェクトはすべて相互に認識できメッセージを送り合うことが可能であることを前提にしている。可視グラフではシステム中でのやり取りをどのように実現するかを決めている。つまり、オブジェクト交信グラフ中でのやり取りを必要としているオブジェクトは他のオブジェクトに対してアクセスするために、そのオブジェクトの参照を持つように設計していくことである。

③継承グラフ

オブジェクトモデル、オブジェクト交信グラフ、可視グラフから、クラス間の関係を見て共通部分の抽象化を行うものである。継承グラフはすべてのクラスに対して作成し、すでにここまでの段階で抽象化されている部分に対して新たに作成したクラスがあれば、それらについても作成する。

3)実装フェーズ

Fusionの最終段階で、設計を効率のよい実装に対応させる。複雑な設計上の決定の大半はすでに行われているので、この変換は比較的簡単である。Fusion法では、どのように展開させればよいのかを、コーディング、性能、レビューの3つの局面から述べている。

(4) Unified Method(第3世代)について

Unified Method は前述のように、見かけだけの違う数多くの方法論を、Booch と Rumbaugh が方法論ウォーに終止符を打つべく提唱された方法論である。

この方法論は、OMTとBooch 法をベースに他のさまざまな手法からのアイデアを採用しており、概念的特徴としては、メタモデルによるモデル概念の見直しと統一、大規模システムに対するカテゴリとその階層化、構造・機能・振る舞いの3モデルの統合、オブジェクト概念とユースケースの統合、分析・設計プロセスの詳細化などがあげられる。ここでは、今までの方法論の中ではあまり述べられていないメタモデルについて簡単にその概念を記述したい。

メタモデルとは、何をモデル化できるか、どのようにモデル化するか、を記述しているとともに、そのシンタックスとセマンティクスを説明している。これの必要性は、作成されたモデルを再利用するためには、作られるモデルのシンタックスとセマンティクスが共通のルールのもとに作成されていなければならないからである。

重要なモデルとしては、クラスモデル、オブジェクトメッセージモデル、ユースケースモデル、状態モデル、操作記述(Use__case に基づいたシステム記述)、物理モデルがあり、個別の詳細はともかく、Fusion Method の成果物とカテゴリはほぼ同様である。

Unified Method は現時点では、「設計上のモデル・成果物」の標準化を整理しつつある段階であり、「プロセス」の標準化まではふれていないが、最終的には唯一の普遍的な方法論を記述しそれを標準化することを目指しているので、今後を見守っていきたい。

(5)まとめ

実際の開発現場でこれらの方法論が利用され、問題点などもフィードバックされて始めており、徐々にではあるが、実用化の方向に進んでいる。また、幸いなことにメジャーな方法論が、独自の方向を示すのではなく、お互いの良い部分を吸収し、統一の方向に向かい始めているなど、非常にありがたいことである。

しかし、利用するわれわれの立場からすると、フレームワークやデザインパターンの有効利用、ビジネスオブジェクトとの関連や、さらにリバース分野での適用などを意識したとき、どのように方法論を有効利用すればよいのかの記述がないなど、まだまだ解決しなければならない点を多く抱えているのが現状のようだ。

〈参考文献〉

- (1)統一方法論を使ったオブジェクト指向分析・設計手法(講演会資料)、羽生田栄一
- (2)オブジェクト・オリエンテッド開発方法論:The Fusion Method 、トッパン

1.2 オブジェクト指向DB技術

(1)オブジェクト指向データベースの概念と適用範囲

オブジェクト指向データベース(OODB)は、第1世代のネットワーク・データベース(ND B)、第2 世代のリレーショナル・データベース(RDB)に代わる、第3世代のデータベースである。従来のデータベースでは扱いにくかったCAD、CASE、マルチメディアのデータが OODBでは容易に扱うことができるようになった。

OODBは、CADのようなデータ構造が複雑で比較的小規模なデータベースの処理に適しており、このような分野での性能の良さはベンチマークテストでも示されている。CADでよく扱われる複合(complex)データでは、オブジェクト指向 DBMS(OODBMS)はリレーショナルDBMS(RDBMS)に比べて平均して数十倍、対象によっては数百倍高速に稼働するというテスト結果が報告されている(1.2 参照)。

OODBMSは、基本機能として「オブジェクト」を扱うことができなければならない。そのためOODBは、オブジェクト定義言語ODL(Object Definishon Language)とオブジェクト問い合わせ言語OML(Object Manipulation Language)を用意している。OMLはRDBでのSQLに相当する。

OODBMSは、一般のDBMSが3大機能としてもっている、メタデータ管理機能、質問処理機能、トランザクション管理機能をもっており、なおかつオブジェクト指向のパラダイムをサポートするので、基本クラスライブラリ、オブジェクト、クラス、継承、メソッドのオーバーロードやオーバーライドなどの機能を実現している。

さらに、OODBMSはCADやマルチメディアといった先進的応用をこなさなければならないので、一般的にバージョン管理、コンフィグレーション管理、チェックイン・チェックアウト機能、ロングトランザクションなどの機能を有している。これらの機能をまとめて表1-2-1に示す。

表1-2-1

表1-2-1 オブジェクト指向DBMSの主な機能

データベース管理システムとしての基本機能	<ul style="list-style-type: none"> ・メタデータ管理 ・問い合わせ処理 ・トランザクション管理
オブジェクト指向パラダイムサポート機能	<ul style="list-style-type: none"> ・基本クラスライブラリ ・オブジェクト ・クラス（階層） ・オーバーロード、オーバーライド他
先進的データベース機能	<ul style="list-style-type: none"> ・バージョン管理 ・コンフィグレーション管理 ・チェックイン・チェックアウト機能 ・ロング・トランザクション機能

日本国内において、OODBの使用はCADやマルチメディアの分野で適用される場合が多いが、他にもさまざまな分野で適用性が試されている。しかしながら、OODBがどのようなアプリケーション開発に適し、どのような分野に不適切なのか、現時点で明確な境界線を引くことは困難である。

ここでオブジェクト指向データベースシステムを理解するためにRDBとOODBの違いを見てみよう。事例として航空機1機の部品展開構造(図1-2-1)を考慮して、それぞれRDBとOODBに展開してみると(図1-2-2)、(図1-2-3)のようになる。RDBでは1機の航空機は複数のリレーションに格納された、複数のタプルの形でしか表現できない。OODBでは「親」部品がその属性値として直接「子」部品を値としてとるので、航空機はその構造のままデータベース化される。したがってオブジェクト指向アプローチでは、部品展開構造をそのまま反映した形で航空機がモデル化されるので、ユーザーにとって理解しやすく扱いやすい。

図 1-2-1 航空機の部品展開構造

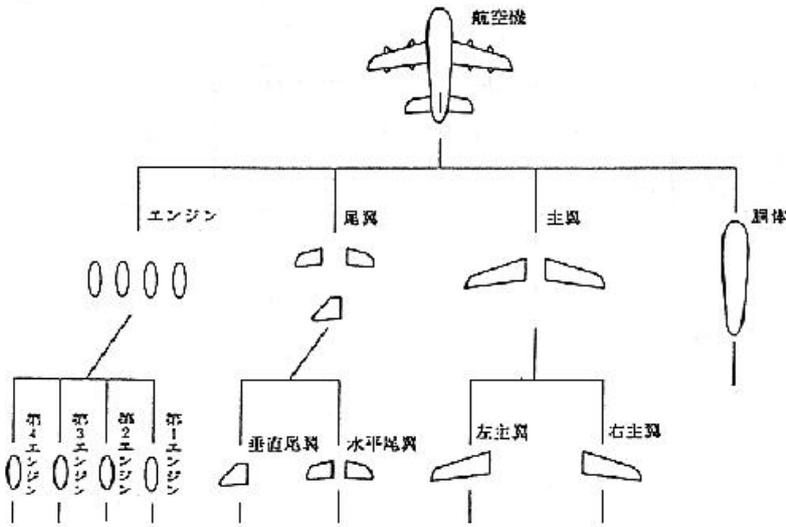


図 1-2-2 部品表 (BOM) による航空機構造のリレーショナルデータベース表現

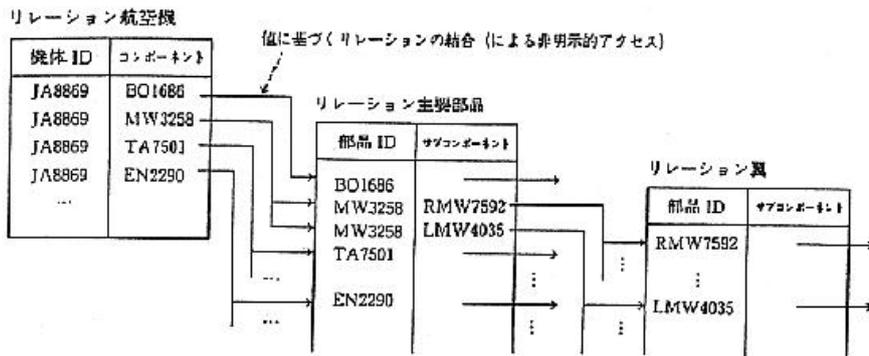


図 1-2-1

図1-2-2 部品表 (BOM) による航空機構造の
リレーショナルデータベース表現

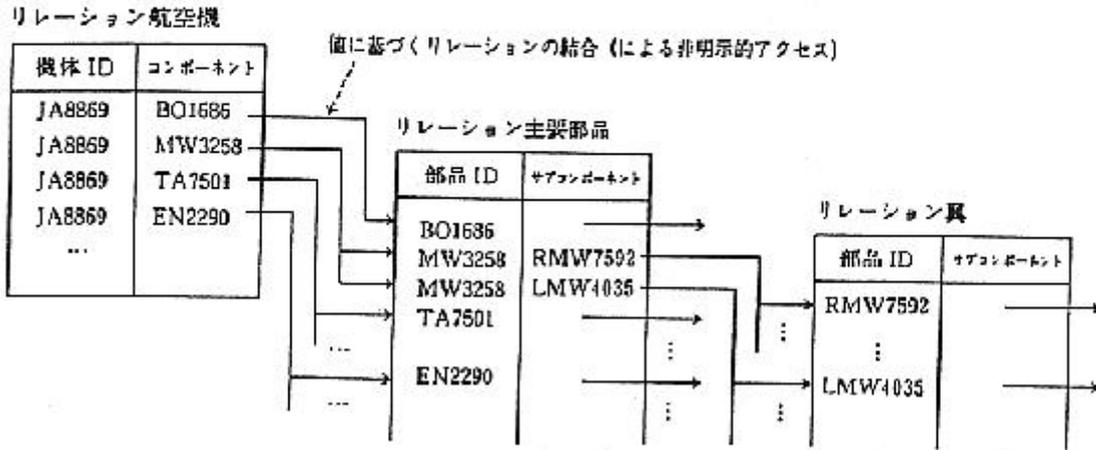


図1-2-2

図1-2-3 オブジェクト指向データベースによる航空機構造の表現

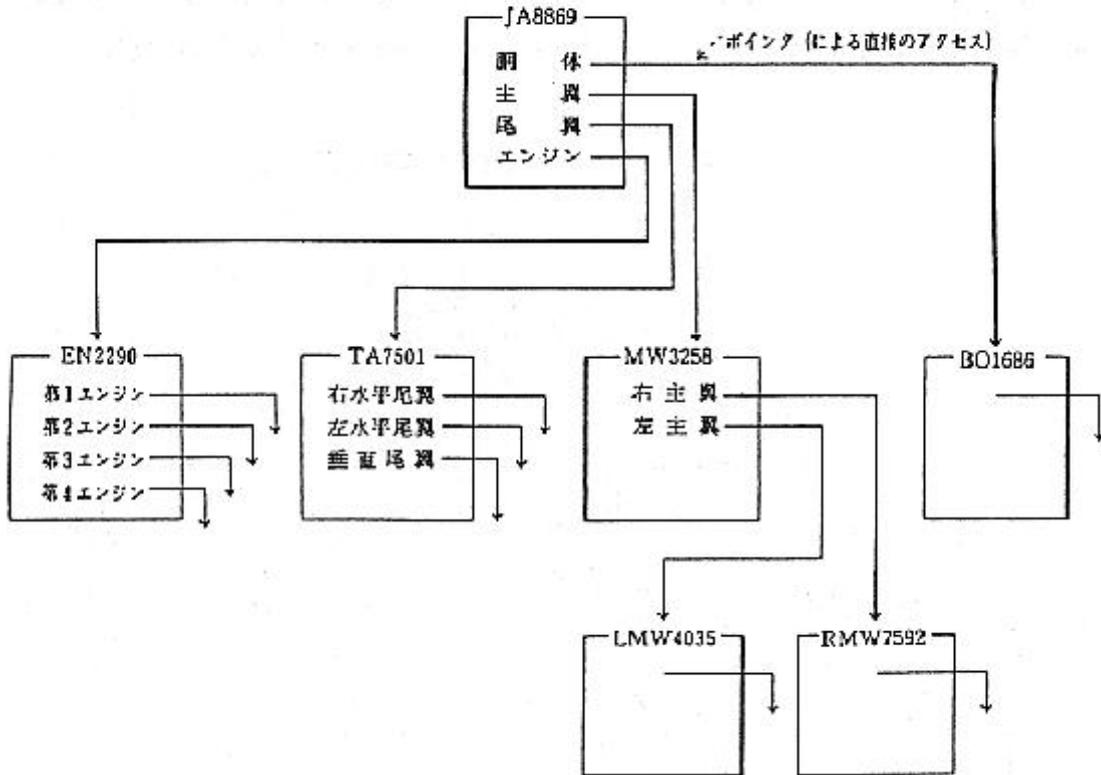


図1-2-3

(2)オブジェクト指向データベースの性能について

オブジェクト指向データベースシステムの性能を評価するため、数々のベンチマークテストが行われてきた。ここでは米国の Sun Microsystem 社での 001 ベンチマークテストを簡単に紹介する。

このベンチマークテストは、OODBがRDBに比べて、どのくらい性能がよいか調査する目的で行われた。テストのために構築したデータベースは2万個の部品(レコード)からなり、1個の部品から3本の結線が出て、他の部品とつながっている。したがってトータル6万本の結線があることになる。

ベンチマークテストは3種類のデータベース操作で行われた。

- ① lookup : 部品番号をランダムに指定して、そのレコードを検索する。
- ② traversal : 部品番号をランダムに指定して、その部品につながっているすべての部品データを結線が7段まで求める。
- ③ insert : 新たなノードを、そこから3本の結線も含めて挿入する。

以上の操作に要する時間を計測してテストが行われた。

これらのテストは各々10回繰り返し行い、第1回目のテストをコールド(cold)状態、2回目以降でテスト結果が定常的になった状態をウォーム(warm)と言っている。通常業務でのクライアント作業をみると、必要なデータを読み込んだ状態で行っていることが多いことから、ウォーム状態でのテストが重要である。図1-2-4にベンチマークテストの結果の一例を示す。

図1-2-4 001 ベンチマークテストの結果の一例 (Catt 91)

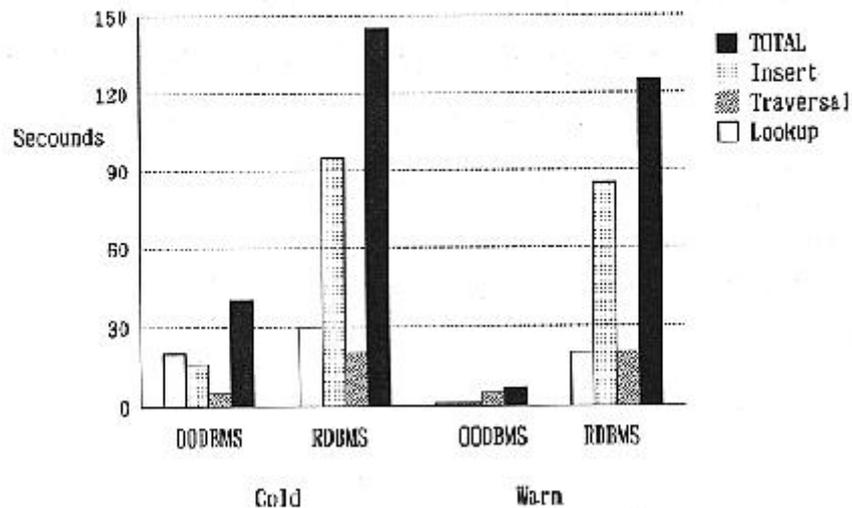


図1-2-4

図中のTOTALはinsert、traversal、lookupの総和である。実験の対象となったDBMSは5つのOODBMSと2つのRDBMSである。図にはそれぞれ最速値がプロットされている。ウォーム状態でのテスト結果をみると、traversalで要した時間はOODBMSでは0.1秒、RDBMSでは84秒となり、性能比はOODBMSが84倍高速ということになる。

これは、OODBMSではデータが対象の構造をポインタを使って格納されるのに対し、RDBMSではデータが部品表の形で格納され、traversal処理を行うにはその深さに応じてリレーション結合を行わなければならない、処理に時間がかかっていることによる。

(3)OODBの課題

本節では、現状のOODBが抱える課題を列挙する。

○関連記述能力の不足

現実世界をオブジェクトとしてモデル化するためには、さまざまな関連が表現できる必要がある。しかし、現状は“has a”、“is a”等の限られた関連しか表現できない。

○時間／空間のモデル化能力がない

オブジェクト間の時間的、空間的關係をきちんと表現できない(例えば“私は、明日の今頃には飛行機に乗って北海道に向かっている”という事象をモデル化できない。これをモデル化するには、飛行機と私が、ある期間だけ関わりをもつということと、“搭乗”という空間的

な関連をモデル化できなくてはならない)。

⇒時間的なモデル化に関しては、参考文献が詳しい。

○宣言的質問言語が未整備

RDBMSに対するSQLのような、問い合わせ言語が整備されていない。

⇒ODMG-93ではOQLという問い合わせ言語が定義された。今後に注目。

○標準オブジェクトモデルがない

すべてのRDBがコッドが提案したデータモデルに基づいているのとは異なり、オブジェクト指向データベースには基準となるデータモデルがない。⇒ODMG-93ではODMG準拠のオブジェクト・データベース管理システムがサポートするオブジェクト・モデルが定義された。今後に注目。

○クラスライブラリが未整備

現状のクラスライブラリは、コンピュータに関連したプリミティブなものばかりであり、ビジネスでの使用に耐える抽象度の高いライブラリが未整備。

⇒OMGではビジネスでの使用に耐える抽象度の高いオブジェクト、すなわちビジネスオブジェクトの標準化を進めている。

○オープン化

他のDBMSとの連携や他オブジェクトシステムとの連携が難しい。

○サクセスストーリーの積み上げが必要

ある程度以上の規模をもった成功事例が少なすぎる。

以上のように課題はあるものの、それに対するアプローチもなされており、状況はどんどん変わっていくものと思われる。

〈参考文献〉

(1)図書館情報大学 増永良文教授 講演資料
“オブジェクト指向データベース技術とその応用”

(2)図書館情報大学 増永良文教授
“マルチメディアデータベースと時間”情報処理 Vol.36 No.5 Page.369-377
95.05

(3)R.G.G.Cattell 編 野口喜洋他訳

“オブジェクト・データベース標準ODMG_93 Release 1.1”共立出版

(4)オブジェクト指向データベース製品の動向

OODBMSでは、RDBでは困難であった「データと手続きとの一体化(カプセル化)」「複雑なデータ構造」「豊富なデータ型」などを利用できる。このため、RDBが大規模でデータ構造が単純な事務処理に適しているのに対し、OODBは小規模でデータ構造が複雑なCAD、CASE、マルチメディアなどの分野に適していることは、前項(1.2)で述べたとおりである。しかし最近、2つのDBMSは徐々に接近し、それらの間の垣根は低くなり、適用領域はオーバーラップしつつある。

OODBMSからRDBMSへの接近の仕方には、大きく2つのアプローチがある。

1つは、OODBMSにRDBMSの機能を持たせ、OODBの適用範囲をRDBの適用範囲にまで広げるとともに、従来のRDBの利用者からの移行を容易にしようとするアプローチである。例えば、UniSQLでは、データベースへの問い合わせ言語としてRDBの問い合わせ言語であるSQL(Structured Programming Language)の基本機能を完全にサポートしているほか、ビュー、トリガ、整合性制約といった先進的な機能もほぼすべて持っている。また、Objectivity/DB、VERSANTも同様にSQLの基本機能をサポートしている。

もう1つのアプローチは、ゲートウェイによりオブジェクト指向言語によるアプリケーション・プログラムを内部でSQLに変換し、OODBへのアクセス方法のままで、RDBへのアクセスを可能にする方法である。ゲートウェイにより既存のRDBのデータをそのまま利用できるようになり、利用者は基本的にはRDBMSとOODBMSのデータを一緒に扱えることになる。例えば、ObjectStore、ONTOS DBではゲートウェイを通し、ORACLE、SybaseといったRDBMSのデータがそのまま利用可能である。また、UniSQL、Objectivity/DB、VERSANTもSQLによる問い合わせにより、既存のRDBMSデータへのアクセスが可能である。

いずれのアプローチでもRDBの領域に踏み込むためには、高信頼で大規模なデータベースが扱えるよう、並行処理制御機能、復旧機能、セキュリティ機能といったDBMS本来の基本機能を充実させるとともに、高トランザクションに耐えうる高速処理の必要があり、OODBMS製品もバージョンアップを重ねるごとにこれらの基本機能が強化されてきている。例えば、Objectivity/DB、UniSQLでは高速化の手法として、RDBMSでは一般的となっているマルチスレッド方式を採用している。表1-2-2に最近のOODBMS製品の概要を示す。

一方、RDBMSもOODBMSへと接近してきている。ISO(国際標準化協会)とANSI(米国規格協会)が共同で標準化を進めているSQLの次期仕様であるSQL3では、オブジェクトを定義するための「抽象データ型(Abstract Data Type)」を導入し、カプセル化機能と

継承機能のサポートを予定している。SQL3をインプリメントしたRDBMS製品の登場も間近い。

〈参考文献〉

- (1)宇田川佳久「オブジェクト指向データベース入門」
ソフト・リサーチ・センター 1992 年
- (2)「日経インテリジェントシステム別冊:オブジェクト指向データベース」
日経BP社 1994 年
- (3)田中淳「実用性高まるオブジェクト指向DB」
日経コンピュータ pp87-89 1995.7.24
- (4)「日経コンピュータ別冊:最新オブジェクト指向実践ガイド」
日経 BP 社、1995 年
- (5)ISO-ANSI Working Draft “Database Language SQL/Foundation(SQL3)”
March 1994
- (6)「Introduction to GemStone」 日本電子計算 1995 年
- (7)「ONTOS」カタログ 日商エレクトロニクス
- (8)「オブジェクト指向データベース入門」 オージス総研
- (9)「ObjectStore 技術解説書 第3版」
オブジェクト・デザイン・ジャパン 1995 年
- (10)「オブジェクト指向と VERSANT ODBMS」 ニチメンデータシステム
- (11)「UniSQL オブジェクト・リレーショナル DBMS」 NTTデータ通信

(5)CORBA環境でのオブジェクト指向データベース

本項の目的は、OODBMSやRDBMSがOMA(オブジェクト・マネージメント・アーキテクチャ)の 中にどのように組み込まれようとしているのかを明らかにすることで、オブジェクト指向を適用したシステムでOODBMSやRDBMSが担おうとしている領域を推し量ろうとするものである。

1)OMAの概念

OMAは、オブジェクト指向技術に関する世界最大の標準化団体であるOMG(オブジェクト・マネジメント・グループ)が提唱するアーキテクチャであり、その目的は、分散環境におけるオブジェクト間のコミュニケーションを特定のプラットフォームや特定の技術からは独立して可能とさせることにある。

OMAでは、CORBA(Common Object Request Broker Architecture)と呼ばれるインフラストラクチャが、異機種システム間にまたがる、オブジェクトの移植性と相互運用性を保証する。(図1-2-5_オブジェクト・マネジメント・アーキテクチャの概念図)

1-2-5 オブジェクト・マネジメント・アーキテクチャの概念図
 (参考文献(3)「共通オブジェクト・リクエスト・ブローカ
 構造と仕様 CORBA1.1」より)

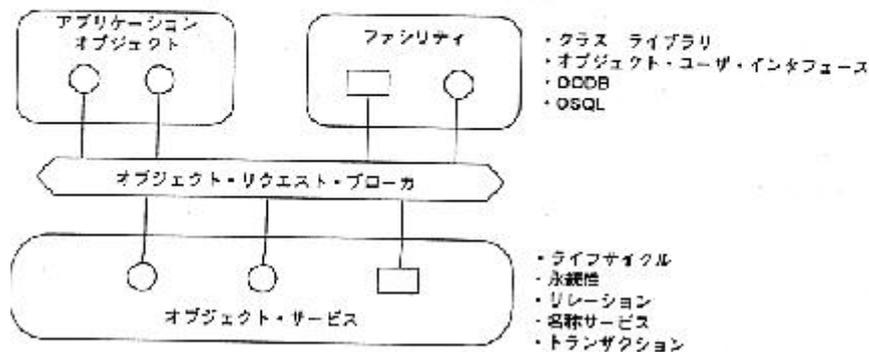


図1-2-5

2) CORBA環境でのオブジェクトの永続性確保

一般にCORBA環境のほとんどのオブジェクトは、オブジェクトを生み出したプロセスが終了しても、その状態を保持する必要がある。CORBAでは、オブジェクトの状態をハードディスク等の不揮発な記憶装置に格納し、必要に応じてオブジェクトの状態をメモリにロードするサービスとしてPOS(Persistent Object Service)を提供している。

POSはデータの記憶形態(ファイル、RDB、OODB、BENTO等)から独立したインタフェースであるため、アプリケーション開発者は、POSを利用することで、記憶形態を気にせずに永続的オブジェクトを作成することが可能となる。(図1-2-6 POS の概念図)

図1-2-6 POSの概念図

(参考文献(1) "The Essential Distributed Objects Survival Guide" より)

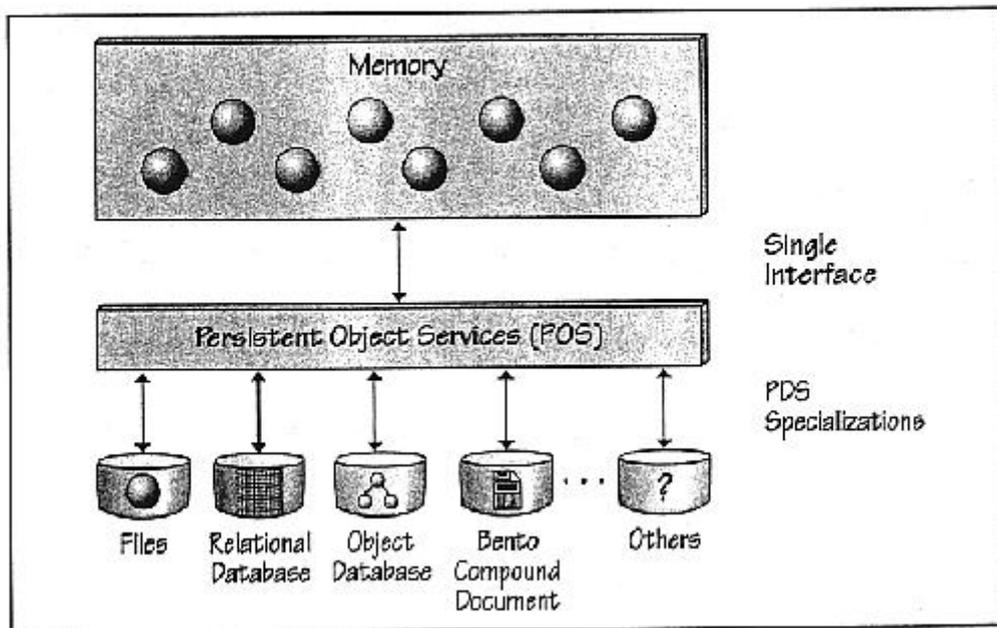


図1-2-6

3) POSの構造

POSは図1-2-7に示すような4種類のコンポーネントから構成されている。

図1-2-7 POSの構造

(参考文献(1) "The Essential Distributed Objects Survival Guide" より)

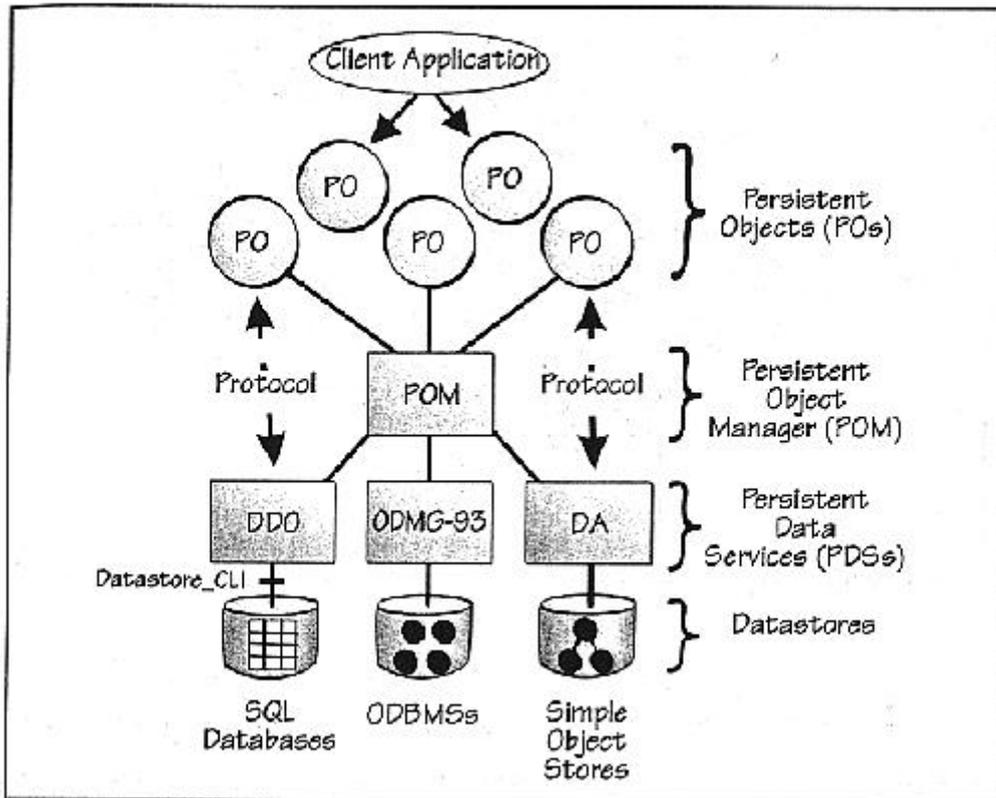


図1-2-7

- PO (Persistent Object)
永続性のあるオブジェクトであり一般にサーバーと呼ばれるものに相当する。
- POM (Persistent Object Manager)
POからの要求を適切なPDS (Persistent Data Service) に伝えるルータとしての役割を持っている。また、POMはPOとPDSを隔離することで、新たなPDSのプラグアンドプレイを可能としている。
- PDS (Persistent Data Service)
特定の記憶形態に依存して実装されているが、インタフェース自体は記憶形態から独立したものであり、実装を隠蔽している。
- DATASTORE
RDBMSやOODBMSのこと。

永続サービスの提供者と永続オブジェクトの開発者がこうした枠組みを利用して開発を行う限りにおいては、特定の技術やプラットフォームに依存しないシステム開発が可能となる。

しかしながら、RDBMSやOODBMSはパフォーマンスを向上させるため、独自のプロトコルを定め、POからの直接アクセスもサポートしている。特にOODBMSはC++の構造体のような粒度の小さい数百万のオブジェクトを並列に扱うことがその役割であるため、POSの枠組みの中では、オーバーヘッドが大きすぎて、OODBMSのメリットを発揮できない。

DDO (The Dynamic Data Object) やODMG-93はRDBMSやOODBMSとPOがPOMを介さずにアクセスするためのプロトコルである。これらのプロトコルはPDS固有のプロトコルであるため、こうしたプロトコルを使用しているPOは異なる環境では動作しない。

4) ODMG-93 が提唱するオブジェクトアダプタ

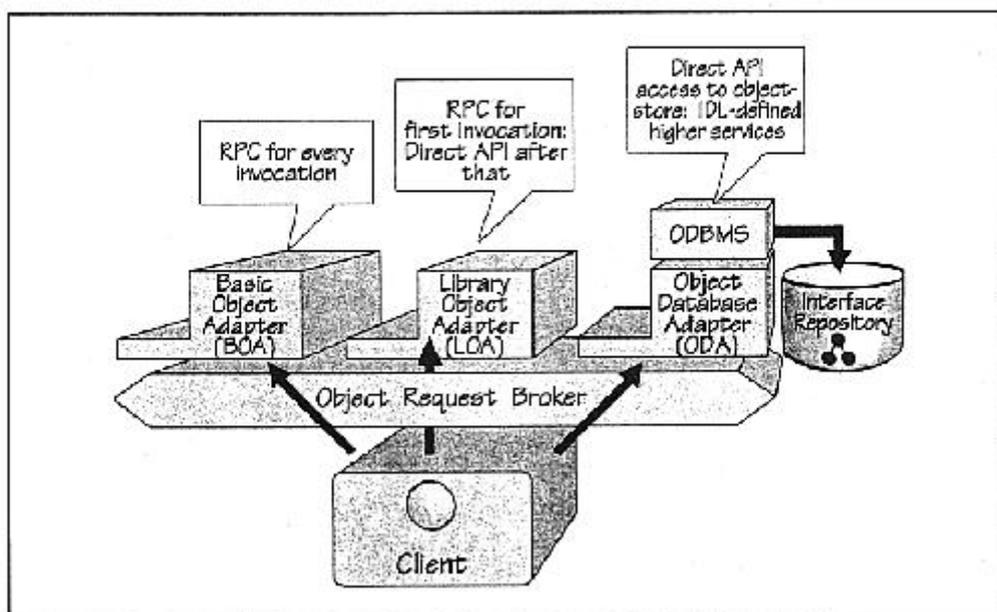
オブジェクトアダプタは、CORBA環境でサーバー側において、クライアントが呼び出したオブジェクトを起動し、メッセージを渡す役目を担うコンポーネントである。

現状、CORBAが採用しているBOA (BASIC OBJECT ADAPTER)は、オブジェクトにメッセージを渡すために毎回プロセス間通信を行う。これはフレキシビリティの向上を狙っているわけであるが、OODBMSにとっては、オーバーヘッドが大きすぎ、デメリットの方が大きい。

そのためOODBMSの主要ベンダーで構成されるODMG (Object Database Management Group)は、オブジェクトデータベースの標準ODMG-93の中で、ORBにOODBMSが管理しているオブジェクトの一部の参照情報を登録することでORBからダイレクトにオブジェクトにアクセスする仕組み、ODA (OBJECT DATABASE ADAPTER)を提唱している。(図1-2-8 オブジェクトアダプタの概念図)

図1-2-8 オブジェクトアダプタの概念図

(参考文献(1) "The Essential Distributed Objects Survival Guide" より)



5)まとめ

- RDBMSとOODBMSはどちらもCORBA環境で利用可能である。
- POSを利用することでパフォーマンスは犠牲になるが個々のDBMSから独立したオブジェクトを作成することが可能となる。逆に、オブジェクトの独立性よりもパフォーマンスを重視し、DBに依存したプロトコルを使用することも可能。
- オーバーヘッドにはプロセス間通信によるものと、オブジェクトの状態を格納するための処理(RDBにオブジェクトの状態を格納するには、オブジェクトの属性値をテーブル構造に合わせて分解する必要がある)に起因するものがある。

最後に主観的なコメントを添える。

個々のDBMSは対象としているオブジェクトのサイズが異なる。すなわちOODBMSはC++の構造体のような小さいオブジェクトを数多く扱おうとしており、RDBMSはコンパウンドドキュメントのようにサイズの大きな少数のオブジェクトを扱うことを狙っているように思われる。

この推理が正しければ、着目しているアプリケーション処理に必要なオブジェクトの数の大小によって、適するDBMSを選択する必要性や、逆の場合(すなわち利用するDBMSが決まっている場合)、設計結果がDBMSに依存して大きく変わるであろう。

具体的には、ワークシートを全体で1つのオブジェクトとしてとらえると、オブジェクトの数はたかだか1つでしかないが、個々のセルをオブジェクトとしてとらえると1つのワークシート上のオブジェクトは簡単に数万件に達してしまい、利用するDBMSによっては良好なパフォーマンスで永続サービスを提供することが難しくなってしまうことが想像される。

〈参考文献〉

(1)Robert Orfali, Dan Harkey, Jeri Edwards 著 “The Essential Distributed Objects Survival Guide” John Wiley & Sons, Inc.

(2)OMG編著 オブジェクト・マネジメント研究会翻訳 “オブジェクト・マネジメント・アーキテクチャ・ガイド OMAG2.0”

オブジェクト・マネジメント研究会 創研プランニング

(3)OMG編著 工学博士 相磯秀夫監訳 “共通オブジェクト・リクエスト・ブローカ 構造と仕様 CORBA1.1”

オブジェクト・マネジメント研究会 創研プランニング

(4)R. G. G. Cattell 編 野口喜洋他訳 “オブジェクト・データベース標準ODMG__93 Release 1.1” 共立出版

1.3 オブジェクト指向CASE

(1)CASEツールとオブジェクト指向

CASEツールは10年以上前からいろいろ開発されてきているが、お絵描きツールの的なものが多かったため大幅な生産性向上には結び付けられず、企業にはあまり導入されていなかった。しかし、オブジェクト指向という概念を得て、今後非常に大きな期待が持てるようになってきた。何故なら、これから開発されるであろうCASEツールは当然のごとく、そのCASEツール自体もオブジェクト指向を使って開発されることが予想される。データと手続きをカプセル化したオブジェクトという扱いやすい存在を得たことにより、CASEツールはソフトウェア開発の上流工程と下流工程をシームレスにつなぐことが可能となる。これはCASEツールの世界では画期的な出来事であり、分析・設計結果からソース・コードを生成し、さらにリバース・エンジニアリング等を含めたスパイラル型（ラウンドトリップ型）の開発を可能にする夢のような統合CASEツールの誕生を予感させるものである。

(2)オブジェクト指向CASEの現状

オブジェクト指向の開発方法論は上流から下流まで成果物をうまく継承できる特徴を持つことから、CASEツールはその方法論と併せて提案されている。現在、オブジェクト指向分析、設計方法論は数多く提案されており、どの方法論も主流になるべく改良や合体を繰り返している。方法論の普及度を知るバロメーターの1つがCASEツールでの採用の割合となっている。また、CASEツール開発側もできるだけ自社の指示する方法論をメジャーな存在にするべく方法論の提唱者を自社に取り込み方法論の統一化を図ったりしている。米ラショナル社はOMT法のジェームス・ランボー氏とBooch法のグラディ・ブーチ氏に加えて、ユースケース概念を考案したアイバー・ヤコブソン氏までも招き入れ、統一方法論の作成にとりかかっている。これにより記述力に優れ設計面に強いBooch法と、分析から設計、プログラミングまでをフルにカバーはしているがどちらかというと分析寄りなOMT法が合体するという、非常に魅力的な組み合わせが実現したわけである。現段階ではそれぞれに長所短所を併せ持っている各手法が、各々の長所を取り込んで進化していくであろう将来の方法論には大きな期待を寄せることができる。CASEが実用に耐えうるものとなっていくためには、方法論側もこのように改善をくり返し欠点をなくしていく必要がある。

現在、表1-3-1、表1-3-2のようなCASEツールが販売されている。技術の進歩により最近ではPCで使える商品も出てきているが、現状ではWSで稼動するものが主流である。価格は数十万円から数百万円までのものが大半である。

表1-3-1

表1-3-1 主な上流CASEツール

製品名	開発会社	販売会社	手法	概要	稼動環境、価格
SLP/DNT	米インテリテック・ディベロップメント・インターナショナル	エフ・シー・システム	DNT	ユースケースを併用した分析が可能	SPARCstation, HP9000, RS/6000, DECstation 250万円
SEWB3/ROAD	日立製作所	日立製作所	DNT	DLE対応	
LDV/DNT	依ペリコ	CRC 総合研究所	DNT	C++のオブジェクトを生成	
ODSPACE/BS	NEC	NEC	DNT	国産初のツール	
Rose	米ラショナル・ソフトウェア	オージェス総研	Booch DNT	C++コード生成と逆解析機能	Windows3.1, OS/2, SPARCstation, HP9000, RS/6000 34.8万円 UNIX用は85万円
Objectteam	米キャット・インテグレーション	東洋テクニカ	Shlaer-Mellor	分析結果をシミュレーション実行可能	SPARCstation, HP9000, RS/6000 309.7万円から
QMW	米インテリコ	日本科学技術研研移所	Martin/Odell	下流CASEのKappaと連携可能	SPARCstation, HP9000 490万円 (Kappaを含む)
Ptech	米 Ptech	日商エレクトロニクス	Martin/Odell	ビジネスアプリケーション開発向き	

表1-3-2

表1-3-2 主な統合CASEツール

製品名	開発会社	販売会社	概要	稼動環境、価格
Cheetah	米ニューロンテック	ニューロンテック・ジャパン	OpenInterface や NEXPERT OBJECT と組み合わせ可	
Forta	米オメガ・ソフトウェア	三菱商事	大規模 C/S 向けツール、独自の 4GL を提供	
Galaxy	米ビジュアル・ソフトウェア	オージェス総研	異機種が混在する C/S 向けツール	
HOLON/VP	NEC	NEC	ソフト部品を組み合わせてアプリケーションを開発できる。	Windows3.1, Windows NT 標準 7.8万円 オプション 29.8万円
NEXTSTEP	米ネットコンピュータ	キヤノン販売	GUIアプリケーション構築ツール	SPARCstation, HP9000, IBM PC 互換機 59.8万円
ObjectIQ	日立製作所	日立製作所	オブジェクトとA-Mを併用してアプリケーションを開発	SPARCstation, HP9000 93万円から Windows3.1 (実行版のみ) 14.7万円
Obsydian	米オメガ	オメガ・ジャパン	ジェームズ・ボーン氏の C/S 開発方法論 TCSM に対応	
SAPIENS	イスラエル・サピエンス・テクノロジー	サピエンス・ジャパン	汎用機向けツール	
TIPPLER	日本ユニクス	日本ユニクス	オブジェクト、ルール、MINI-TEXTを併用して開発ができる	
Visual Works	米ウィークウェア・デザイン・ワーク	三井物産など	Smalltalk を使った統合開発環境	

上表1-3-2にある統合CASEと称している製品は、分析、設計の上流までを統合しているわけではなく、実際には統合開発環境(下流の開発工程)を提供しているにすぎない。

これらのツール(上表1-3-1、上表1-3-2)が支援しているのは、図形エディタ、プロトタイピング、プログラム生成、リポジトリなどの機能であり、解析系の機能は、ほんの一部に存在しているだけである。中にはお絵描き工具的なものや、従来からある構造化分析・設計を支援していたツールにオブジェクト指向を付け足したようなツールも含まれている。しかし数年前と比べれば機能的には増えてきており、C++や Smalltalk といったオブジェクト指向では主要な開発言語については、単にコード生成を行うだけでなく、ソースコードから設計モデルを抽出できるリバース・エンジニアリングの機能も加わり、双方向エンジニアリングの機能を備えた上流CASEツールもいくつか登場してきている。

また、上表1-3-1にある上流CASEと位置付けられた製品の中には、他のツールと組み合わせることによって下流と連携をとれるツールが出てきている。例えばOMWは Kappa へ設計情報を渡しシステムの実装までが可能となっている。また、Rose はOODBの Varsant をリポジトリに利用したり、FORTE や Objectivity/DB のコード生成を行える。このように他社のツールと連携できるようにする動きも活発になってきている。国内の大手メーカーもオブジェクト指向の将来性に目を付け開発に乗り出してきているが、国産の方法論もなく、日本ではまだまだオブジェクト指向に対する意識が低いことから、まずは最先端を行っている米国でツールを育てて日本に逆輸入していこうといった動きもある。

(3) 今後のオブジェクト指向CASEへの期待

一般にオブジェクト指向の分析・設計は今までの手法に比べて難しいと言われている。これは方法論自体が未成熟なせいもあるが、やはり世の中にはそんなにおいしい話はないわけで、「どんなに経験の浅いSEでも簡単に分析・設計ができる魔法のような手法などは存在しない」というのが事実のようである。どんなにCASEツールが便利になったとしても方法論をマスターせずに分析・設計を行えばおかしなモデルを作り、かえって生産性を落としてしまうことになってしまうのである。しかし難しいからといって壁を乗り越えずにあきらめる必要はない。プロトタイピング機能によるスパイラル型の開発が行えるようになればシステムを簡単に成長させていけるからである。しかし、これからのCASEには壁を少しでも低くするために、単に上流と下流をうまくつなぎラウンドトリップを可能にするという機能以外に、次のような機能を期待したい。

○リポジトリ

オブジェクト、クラス・ライブラリを管理し、部品化再利用を支援

○チーム開発機能

大規模なシステムの開発をするときに複数の設計者による開発を支援

○ガイダンス、ナビゲーション機能

次の手順の誘導やオブジェクト獲得、クラス・ライブラリ設計を支援

これからは、ビジネス・オブジェクトやエンジニアリング・フレームワークといったオブジェクト指向の部品もどんどん世の中に出まわってきて、部品を一から 作らずにすむ時代が訪れることから、CASEを取り巻く環境も、より便利な方向に進んで行くことに疑いの余地はない。CASEが方法論と共に進化し知的な ツールとなったとき、導入する企業は飛躍的に増加し、初めて実用化の時代を迎えるに違いない。

〈参考文献〉

- (1)本位田真一、山城明宏「オブジェクト指向システム開発」 日経BP社
- (2)田中 淳ほか「最新オブジェクト指向実践ガイド」 日経BP社

1. 4 オブジェクト指向言語 Java

ここではオブジェクト指向言語として今最も注目され期待されている言語、Java を取り上げることとする。

コンピュータの世界におけるキーワードは、「パソコン」「インターネット」である。昨今の価格競争によるハードウェアの低価格化と「Windows 95」日本語版の発売により、パソコンの出荷台数は大きな伸びを示している。

一方、全世界的規模のネットワーク「インターネット」がビジネスの世界で注目を浴びており、ユーザー数 も増大、現在では 8000 万人とも言われている。パソコンを使って、インターネットに接続し、世界のさまざまな情報を手に入れたり、自分の作ったものを全世界に公開する時代がやってきたことで、ビジネスはインターネットを前提とするようになってきた。

インターネット上にホームページを作成するには、これまではHTMLが使われてきたが、この言語は手軽である反面、絵を動かす、音を鳴らすなどといった複雑なことはできなかった。

そこで登場したのが、HotJava というブラウザと、このブラウザの上で動作させることのできるプログラム言語、Java である。この Java の出現により、WWW業界は大きく変わろうとしており、これを開発したサン・マイクロシステムズ社は、インターネットの世界でトップを走るほどの勢いである。

(図1-4-1 業界標準を狙う 4 大勢力)

(ファイルが見つからない)

図1-4-1

(1)Java の特徴

Java 言語は、SmallTalk、HyperTalk、C++、Tcl/Tk 等と同じような「オブジェクト指向プログラミング言語(OOP)」である。C++が持つ多重継承のような、便利だが、あまり需要がないような機能をできる限り削除し、スタンドアロンでも動作可能にするために、ベーシックインタプリターとクラスサポートを約 40KBに抑えてある。また他のOOP同様、クラスライブラリも用意されているが、Javaの世界では、API(Application Programming Interface)と呼ぶ。

Javaの最大のメリットは、これまで困難とされてきたイベント駆動型のプログラム、いわゆる「会話型」のプログラムを開発できることにある。その他、Javaは次のような性格をもっている。

- ①単純で無駄がない
- ②習得しやすい
- ③開発効率が高い
- ④中間コードを生成し、実行する
- ⑤マシンの機種にとらわれない
- ⑥どのようなプラットフォーム(Windows、AppleTALK 等)でも動作が可能である
- ⑦複数のアプリケーションを同時に処理できる

(2)Java 開発支援ツール

これまでに述べてきたように、Javaはオブジェクト指向言語の中では非常に簡単なイメージがあるが、その基本構造はC++と同じである。したがって、誰でもが書けるといった言語ではない。しかし、コンピュータ上に迅速にロードできる非常に小型アプレット(プログラム)を開発することができるため、ソフトウェアの新手法として注目されており、特にマルチメディア関連製品ではJavaが必要となってきた。これによりプログラマでなくてもプログラムを開発できる環境が必要となってくるのである。

いち早く開発用ツールのリリースを発表したのは「ポーランド社」である。その他、続々と開発支援ツールが発表されることが予想されることから、誰もがJavaを書き、Javaアプレットを作る時代が来るであろう。

なお、現在わかっている開発支援ツールは表1-4-1のとおりである。

表1-4-1

表1-4-1 Java開発支援ツール

会社名	コードネーム	製品名
ポーランド社	Latte	
シマンテック社	Espresso	Cafe
マイクロソフト社		VisualBasicScript

(3) Java の可能性

Java は 500ドルパソコンの分野で活躍するであろうと言われている。500ドルパソコンは機能を絞り込んだもので、32ビットMPU、8メガバイトの記憶容量程度のもので、これに組み込むソフトウェアは Java で開発したものが適している。Java によりパソコンが家電製品やゲーム機と同じ扱いになる時代がやってくる。

またイベント駆動型のプログラムの開発が可能となることで、ネットワークの世界にもリアル性を持ち込むことが可能となってきた。これはビジネスの世界、とりわけ情報を提供するようなビジネスにおいては注目に値するものと思われる。

Java 言語が普及するにつれ、さまざまな発想でプログラムが開発され、ビジネスの可能性の枠を大きく広げていくことが予想され、今後の各社の動向に期待したいものである。

今回の研究サブテーマである「ビジネス・オブジェクト」では、一体どんな言語が採用されるのであろうか。Java が企業の情報ネットワーク構築に使われることとなるのか等、今後の展開に注目していかなければならないであろう。

〈参考文献〉

- (1)「図解 60 分でわかる Java」 伊藤友八郎、河野光雄著 PHP研究所
- (2)「あまくち Java 言語」 RESET編 高木和夫著 同文書院
- (3)「Java/HotJava ようこそジャバ・ザ・ワールドへ」
菅野政孝編 NTTデータ通信 Java 研究会著 カットシステム
- (4)「Java/HotJava を知る インターネットの新しい世界」
JohnDecember 著 松田晃一訳 ブレンティスホール出版発行 トッパン発売
- (5)「Java 入門」
有我成城、衛藤敏寿、佐藤治、白神一久、西村利浩、村上列著 翔泳社

第2章 ビジネスオブジェクトの最新動向

2.1 ビジネスオブジェクトの動向

ビジネスオブジェクトという言葉は、昨年度の報告にも述べたが、オブジェクトの異種分散環境での相互運用性を解決する目的で1989年、Sun、HP、Data General、American Airlinesなどの企業11社が設立した非営利のコンソーシアムであるOMGにBOMSIGができあがった1993末にでき上がった言葉であり、「人、物、場所、コンセプトなどビジネス領域に存在する現実の世界の事象をオブジェクトの形で表現したもの」と言える。

なぜビジネスオブジェクトの考え方が出現してきたかといえ、世界的なビジネス競争が激化する中で企業が生き残っていくための商品開発、サービス提供の迅速化、高品質化を支える情報処理システムの構築、導入には、極端なことを言えば、1週間の単位での短期化が求められるようになったためといえる。すなわち、これまでのシステム開発の発想を根本的に転換し、企業の核となるコアビジネス以外についてはすべて、ビジネスそのものまでも市場から調達してでも短期にシステムを開発してBPR(ビジネス・プロセス・リエンジニアリング)を実現しなければならないとの考え方である。

しかし、考え方は理解できるとしても、そんなに急にビジネスオブジェクトが実用化され市場から調達できる時代が来ることはないのではないかと考えるのが常識で、われわれも大いに期待はするけれどもその実用化の時期については昨年度までは同様の認識であった。ところが、今年度1年間、ビジネスオブジェクトの技術動向を追って見ると、オブジェクト指向技術と、そのインフラの急激な発達により、実用化の速度が加速され、このビジネスオブジェクトによる社内システム構築の成否が企業の命運を分ける時期が間近に来ているのではないかと考えられるようになっている。

現在、どの業種を見ても生産性の最も低い部門は営業・販売・財務・人事など、これまでホワイトカラーと呼ばれる社員が属している部門であることは、指摘されて久しい。

ホワイトカラーの多くが資料や報告書の作成という本来の仕事ではない間接業務に多くの時間をとられているが、それらは伝票などの1次情報を機械的に、整理・分類・集計する作業がほとんどであり、コンビニエンスストアのPOSシステムなどから明かなように、かなりの部分をシステム化・自動化できる。

問題は、システムの構築と保守に必要なとされる時間と費用にあった。

これまでの手法によるシステムの構築では、これらのホワイトカラーの間接業務をシステム化するには、構築したシステムから得られる恩恵以上の投資が必要とされた。特に、刻々と

変化する現実に適応するようにシステムを運用・保守していくためには、システムの構築に必要な以上の投資を要求されることも少なくなかった。

しかしオブジェクト技術の発達により、ホワイトカラー業務のシステム化に必要なとされるコストは劇的に低下することが明らかになっている。

米国での石油業界及び半導体業界のオブジェクト指向をベースにしたシステム開発から発見されたことは、システム構築に必要なオブジェクトの 80%はすべての企業で共通であり、残りの 15%は同じ業界であれば共通であるということである。

つまり、自社の所属する業界にビジネスオブジェクトが構築されていれば、システムの構築・運用・保守に必要なコストの 95%までは負担する必要がなくなり、これまでの開発費の 20 分の1にあたる5%の投資により新しいシステムを構築することが可能になるということである。

これまでは自社の扱う製品や顧客の変化への対応は、情報システムよりも人間の柔軟性の方が優れていた。

しかしビジネスオブジェクトは現実と等価な情報モデルによりシステムを構築しているため、システムの変更は現実の変化とほぼ同程度でよく、これまでのシステムとは比較にならないほどの柔軟性を持っている。

したがって、ビジネスオブジェクトを使用すれば柔軟性の高いシステムを低コストで実現でき、ホワイトカラーの生産性を飛躍的に向上させることができる。ビジネスオブジェクトによる社内システムの構築に成功すれば、ホワイトカラー社員はその本質的な業務のみに、その労力を集中できるようになるのである。

次に、今後のビジネスの形について考えてみたい。

アメリカの大手製薬会社のイーライ・リリー社は、世界 120 か国の拠点にインターナル・ウェブ・サーバーを置き、社内情報の共有化に成功した。これがイントラ(インターネット版LAN)と呼ばれる思想を確立させ、世界中のどの場所においても情報の強い共有化(密接なビジネス)が可能であることを知らしめた。

またインターネットのモールなどからもわかるように、電子ネットワーク上での取引がほとんどの分野において可能になっている。そして最近話題になっている Java が、その根本からオブジェクト指向技術によって作られているように、ネットワーク上でのやりとりも、そのほとんどがオブジェクト化されつつある。

したがって、ビジネスオブジェクトの標準化が進めば、取引の多くがその標準化されたネットワーク上の商品オブジェクトによって行われるようになるのではないだろうか。

その場合、CALSなども当然、ビジネスオブジェクトによるネットワークビジネスの一部にとりこまれてしまうだろう。

今後、ネットワークビジネスは商取引の重要な部分となることは間違いなく、自社の商品をネットワークにのせたり、有用な商品・情報をいち早く手に入れるためにも、ネットワーク上のビジネスオブジェクトを扱える社内システムの構築が必須のものになると考えられる。

以上述べたように、ホワイトカラーの生産性を向上させ、ネットワークビジネスによりビジネスチャンスを拡大し、自社の競争力を維持するためにも、ビジネスオブジェクトによる社内システムの構築を急ぐべき時期に来ている、というのがわれわれの結論である。

ビジネスオブジェクトの標準化はOGMを中心としてかなりオープンな形で行われており、その参加企業や標準化の進捗を見ても、近い将来、ビジネスオブジェクトがビジネスの中心となることは間違いなく、ビジネスオブジェクトに対応した社内システムの構築の成否が社運を分けるというのも、誇張でもなんでもない。

本章では、急速に姿を現し始めたビジネスオブジェクトについて、以下に標準化の動向と製品化状況についての調査結果を述べる。

(注1) OGM……Object Management Group

オブジェクト指向の用語と、4GLの形式と、インタフェース標準において、ベンダー間での広範囲な同意の確立を意図する、有力企業の集まり。
事実上の、オブジェクト指向の標準化機構。

(注2) BOMSIG……Business Object Mangement Special Interest Group

OGMの中のビジネスオブジェクトの検討グループ。

最新のOGMの体制では Business Object Domain Task Force となった。このTFから出される報告が、おそらく、業界標準のビジネスオブジェクトの規格となる。

(注3) OGMの活動状況

1989年4月	OMG設立	(参加企業 11社)
1990年末	OMAG 1.0リリース	
	OMAG…Object Mangement Architecture Guide	
	オブジェクトモデルと参照モデルを通じて、OMGの技術理念と路線を定めると同時に、OMGにおける仕様化の制度的な手続きを明らかにしたもの。	
1991年末	CORBA 1.1リリース	

1992年 中	OMAG 2.0 リリース
1993年 末	BOMSIG設立
1994年 末	CORBA 2.0 リリース
1995年 中	OMG Japan SIG設立
1996年現在	OMG参加企業 約 500 社

2. 1. 1 ビジネスオブジェクトの標準化動向

1)ビジネスオブジェクトの標準化は、OMGのBOMSIG(現在はBODTF…Business Object Domain Task Force に組織改正)を中心において精力的に行われている。

BODTFはOMGの Domain in Technology Committee の配下で活動している。

本TCにはこの他、

- Financial DTF.
- Telecommunication DTF.
- CORBAmed DTF (以前の Healthcare SIG)
- Manufacturing DTF

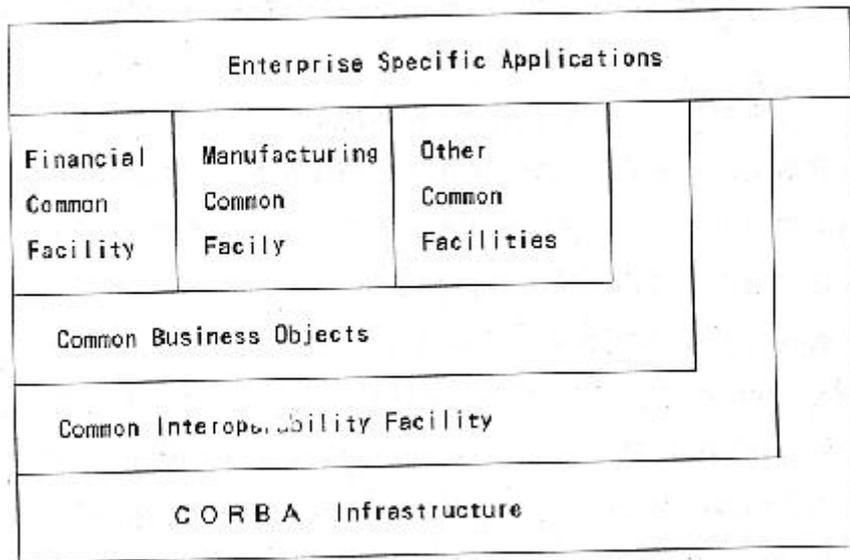
等のDTFが設置され、各ビジネス・ドメインごとのビジネスオブジェクトの標準化を進めている。

OMGの標準化は、1つのプロセス(案件)につき1サイクル2～3年で処理されており、標準として採択された内容も現実的で明確である。

2)OMG共通ファシリティRFP4 (Request For Proposal)

ビジネスオブジェクトの相互運用性のための標準化提案を産業界から募集するためのRFP:Common Business Objects and Component Interoperability Facility の発行が可決されている。このRFPでは図2-1-1に示すように企業固有の業務が共通ビジネスオブジェクト(Common Business Object)とドメイン共通ビジネスオブジェクト(Domain Common Facility)を利用して実現され、CORBA環境上で相互運用性を保証されるアーキテクチャモデル(OMA準拠)となっている。

図2-1-1 OMGのCommon FacilitiesのRFP-4のAPモデル



(J. サザーランド、Object World Expo/ Tokyo '95)

図2-1-1

これまでの流れを見てもわかるように、完全に標準化された規格・仕様に基づいてシステム構築を行える時代が、すぐに到来することが期待される。

2.1.2 ビジネスオブジェクトによるシステム開発

1) オブジェクト指向により開発したシステムの汎用性の高さを示す事例として、下記の2つが著名である。

○テキサス・インスツルメンツ社とセマテック社の協同開発による半導体製造プロセス管理システム

○油田開発と石油採掘に関係する企業76社が参加した、PDSCによるビジネス・オブジェクト・モデル開発

これらはどちらも、同業種の企業が企業の基幹となる大規模システムの開発を協力して行ったものであり、オブジェクト指向に基づいてシステム開発を行うならば、各企業の独自性を損なうことなくシステムの協同開発が可能であることを示している。また下記の資料には、22個の、実現され、実際に稼働中の(または稼働した)オブジェクト指向によるシステムの著名な開発事例が掲載されている。

○「オブジェクト・レッスン」

著者:トム・ラブ 訳:寺島哲史

出版社:プレントイスホール・トッパン

さまざまな資料や出版物を見ても、オブジェクト指向による開発の生産性の高さと、開発されたシステムの柔軟性や再利用性の高さが現場の担当者から報告されている。

2) ビジネスオブジェクト構築ツールの動向

CORBAによる分散オブジェクト実行環境を待たずに、ユーザーインタフェース、ビジネスプロセス、データベースアクセスの3つの処理を完全に分離し、柔軟かつオープンなシステムの構築を実現する3層(3-tired)アーキテクチャのアプリケーションを容易に構築可能とするオブジェクト指向開発ツールが台頭している。米国 Forte 社の Forte、Dynasty 社の Dynasty などがある先駆者であるが、マイクロソフト社の Visuls Basic 等のGUI構築ツールメーカーやOracle社のOracle等のRDBMSソフトメーカーもこぞって3層構築のための機能の追加を宣言している。

Visuls Basic の普及とともにGUI部品市場が形成されたように、ビジネスオブジェクトはオブジェクト指向と3層アーキテクチャをサポートし、市場占有率が高く、デファクト標準と見られるAP構築ツール群の上で構築され、市販、流通することが予想される。

米国では10年以上前からオブジェクト指向による開発の成功事例が、規模の大小に関わらず、数多く伝えられている。

日本でもオブジェクト指向に基づいてシステム開発を実施する時期にきているのではないかと思われる。

2.2 基幹業務パッケージソフトへのビジネスオブジェクト適用

(1) ビジネスオブジェクト適用の背景

1) ビジネスオブジェクトのコンセプトとパッケージソフト導入の目的

ビジネスオブジェクトは、昨年のオブジェクト指向WGの報告書にあるように、BPR(ビジネス・プロセス・リエンジニアリング)を実施する企業の中で、激しい企業間競争に打ち勝つための戦略策定の中から出てきた話である。つまり差別的有利性を発揮すべきコアビジネスにシステム開発を含んだ経営資源を集中化させるが、コアビジネス以外は特徴を出す必要はなく廉価に処理、システム化できるならビジネスプロセスまで市販品を購入してくればよいという考えが、ビジネスオブジェクトという概念が発生した原点である。ただ、この考え方は企業が基幹業務パッケージソフトを導入するときの動機と同一の考えである。そういう意味

において、廉価に早く対象業務のシステム化を行うことが目的であり、ビジネスオブジェクトの適用が 目的ではないことは当然である。

2) パッケージソフトへのビジネスオブジェクト適用でメリットとデメリット

①パッケージソフト開発側にとって

パッケージソフト会社にとってのメリットとしては、カスタマイズが柔軟にできることを最大のセールスポイントとすることが可能と思える。そこではコアビジネスに近い部分でカスタマイズが多くなるモジュール、例えば生産管理モジュールなどでの適用が一番メリットが大きいのではないかと思う。また販売・物流でも企業により千差万別のビジネスプロセスを持っており、オブジェクト指向によるパッケージソフトはメリットが大きいと言われている。

②パッケージ導入企業にとって

ビジネスオブジェクト適用のパッケージ導入による最大のメリットは、パッケージのカスタマイズの容易さとシステム変更に対して柔軟な対応が可能である点にあると言えるだろう。またカスタマイズ及びシステム変更がない業務としては会計業務などに限定されると言え、パッケージの導入に関してビジネスオブジェクト適用のパッケージソフトを採用することは大きなメリットとなるだろう。

次に、世界で初めて統合業務パッケージソフトに対してビジネスオブジェクトを適用したといわれている、System Software Associates, Inc (以下SSA)によるパッケージソフト「BPCS Client/Server (以下BPCS)」を紹介する。

(2) ビジネスオブジェクト適用のパッケージソフト「BPCS Client/Server」

パッケージソフト「BPCS Client/Server」はSSA開発、販売(国内ではSSAジャパン Ltdが販売)している統合業務パッケージソフトであり、生産管理、物流管理、販売管理、財務・会計が適用範囲である。以下、BPCSのシステムアーキテクチャを中心に説明する。

1) BPCSシステムアーキテクチャの特徴

BPCSの Client/Server を基礎としたアーキテクチャの特徴は次の4点である。

①メッセージ・ベースの分散オブジェクトアーキテクチャ

②ORB (オブジェクト・リクエスト・ブローカー) ベースのシステムとして、業界標準であるCORBA準拠分散オブジェクトサービスを基盤としたシステムアーキテクチャ

③リポジトリをベースとしたシステム生成

ビジネスオブジェクトとしてのオブジェクト、ビジネスルール、データモデルをリポジトリとし

てオペレーティングシステム、データベース、クライアント環境から独立させて管理され、そのリポジトリから実行モジュール、データベース生成が行われる。

④第2世代の Client/Server テクノロジーの利用

エンタープライズシステムの多量なトランザクションにも対応可能な Client /Server モデルを提供する。そこではクライアントとデータベース間におけるビジネスロジックを適切な分散を行うことが可能なシステム構成である。

2) BPCSのオブジェクトモデル

BPCSの Client/Server アーキテクチャでの基盤となることはオブジェクト指向であることである。ここでは、重要な要素である分散コンピューティングを意識しないでアーキテクチャを分類、説明する。

①BPCSのオブジェクトレイア

BPCSのオブジェクトモデルは、企業活動で存在する人・場所・物をオブジェクトとしてモデル化する。そしてその振る舞いがカプセル化されたエンティティが、企業活動をコントロールし起きたことを記録する。ただ、ビジネスオブジェクトとして大事なものは、ユーザーの日々の仕事で使っている物のイメージでビジネスオブジェクトを扱うことができることである。そこでBPCSでは、ビジネス・オブジェクト・レイヤとインフォメーション・エンジニアリング・レイヤの2つのレイヤ分けが行われている。

ビジネス・オブジェクト・レイヤは、ユーザーの概念ビューとして定義される。それはユーザーの端末 操作や処理中の状態管理に対して責任を持つレイヤであり、クライアントワークステーションに配置される。次にインフォメーションレイヤは、企業資源の ビューとして通常E/Rモデルとして定義される。このレイヤは通常サーバーに配置され、企業資源の管理と保護に責任をもつ。また、このレイヤだけが資源管理 (RDBなどのシステム資源) に対してアクセスが可能である。

ユーザーのビジネスオブジェクトのイメージは、インフォメーション・エンジニアリングとしての E/Rモデルではなく、クライアントに存在するビジネスオブジェクトモデルである。例として売り上げオーダーの場合、E/Rモデルでは顧客やアイテムと いった多くの関連を持った複合的なエンティティの組み合わせで定義されるが、ユーザーにとってはヘッダーの情報と行がアイテムで埋められた売上フォームそのものである。(図2-2-1 BPCS Client/Sever Business Object Models and E/R Models)

図 2-2-1 BPCS Client/Server Business Object Models and E/R Models

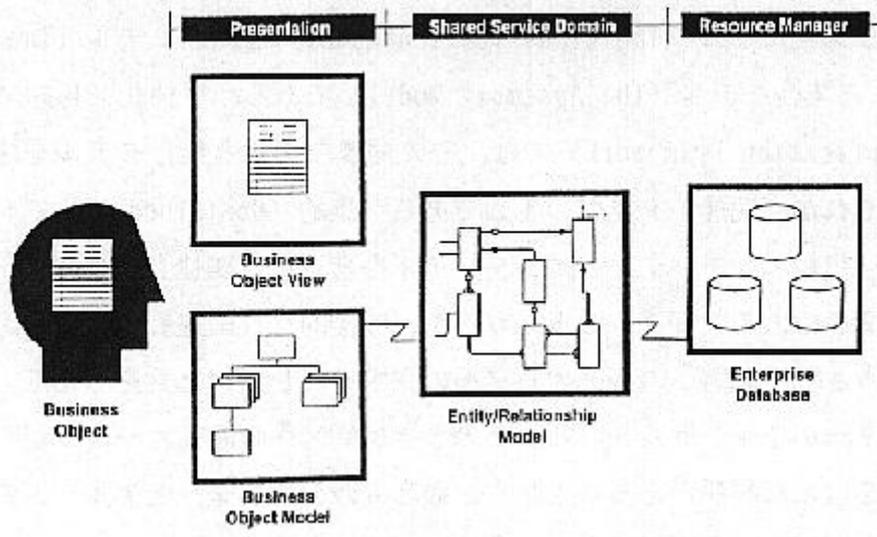


図2-2-1

②共有サービス領域におけるインフォメーション・エンジニアリング

共有サービス領域(Shared Service Domain)は、ビジネスエンティティ(人、場所、物など)、属性(オブジェクト固有の属性)、要素(オブジェクト間で共有可能な属性)、関連により構成されたレイヤとして位置づけられている。(図 2-2-2 BPCS Client/Server Class Hierarchy)

図 2-2-2 BPCS Client/Server Class Hierarchy

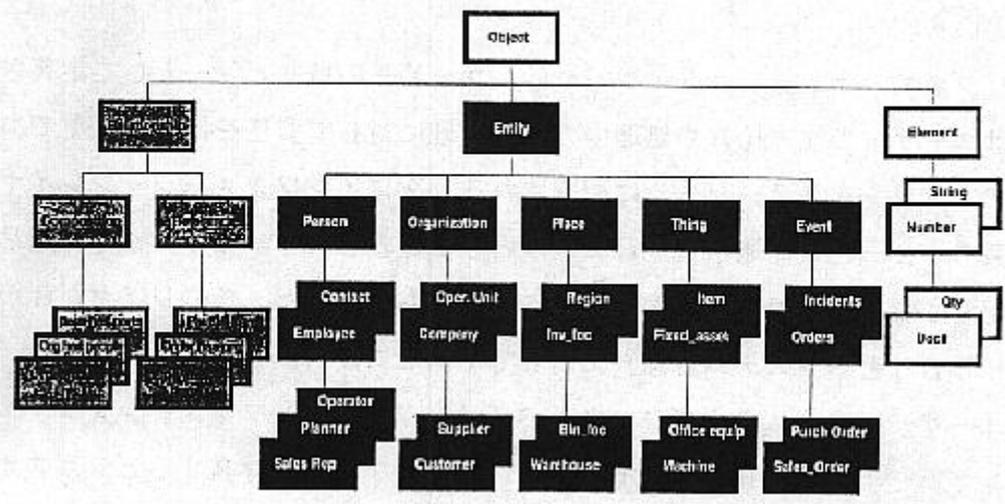


図 2-2-2

特に「関連」は、企業のモデル化に対してビジネスエンティティ間の定義をする基盤となる。「関連」は、論理的な仕事の単位やトランザクションを含んだ複合オブジェクト階層の構成定義として、またビジネスプロセス(例えば、オーダー明細が品目を参照する)の中でのエンティティ間の相互作用の結合も「関連」が行う。

ほとんどのオブジェクト指向方法論では、次の3つのモデル化を行っている。それは、クラス化モデル (The Classification Model)と関連モデル(The Associative Model)と集合モデル (The Assembly Model)である。単純化を目的にしたBPCSの「Application Framework」では、その関連モデルと集合モデルを統合したものであり、それが「関連」タイプの1つである「集約(Contained-in)」を 実現させる。さらに、集約エンティティとエンティティを含んだ「関連」との間と、ビジネスプロセスに関係のある「関連」のトリガーを、間接的な「関連」と させる。例として、キャンセルされた顧客への販売オーダーの在庫を戻したいときを考える。まず顧客の販売オーダーがキャンセルされる。そのときは顧客への 販売オーダーが明細行の複数アイテムを含んだ関係があることから、顧客の販売オーダーとアイテムの在庫を関係づけることができ、在庫の処理を行うことが可 能となる。

BPCSの「Application Framework」は、構成(Composition)と参照 (Reference)の「関連」の抽象化クラスを提供している。まず、構成の「関連」とは、複合的なオブジェクト(仕事のある一塊や処理方針などを含む)の階層構成として定義される。例えばオーダーヘッダーがオーダーの明細行の内容を含ん でいるような構成の「関連」である。参照の「関連」は、ビジネスプロセスの中で相互作用だけがあるような(構成される関係ではない)エンティティ間の関連 として定義されている。それはオーダーの明細行と品目マスターとの間の参照の関連が それにあたる。(図 2-2-3 Business Entity Resaionship)

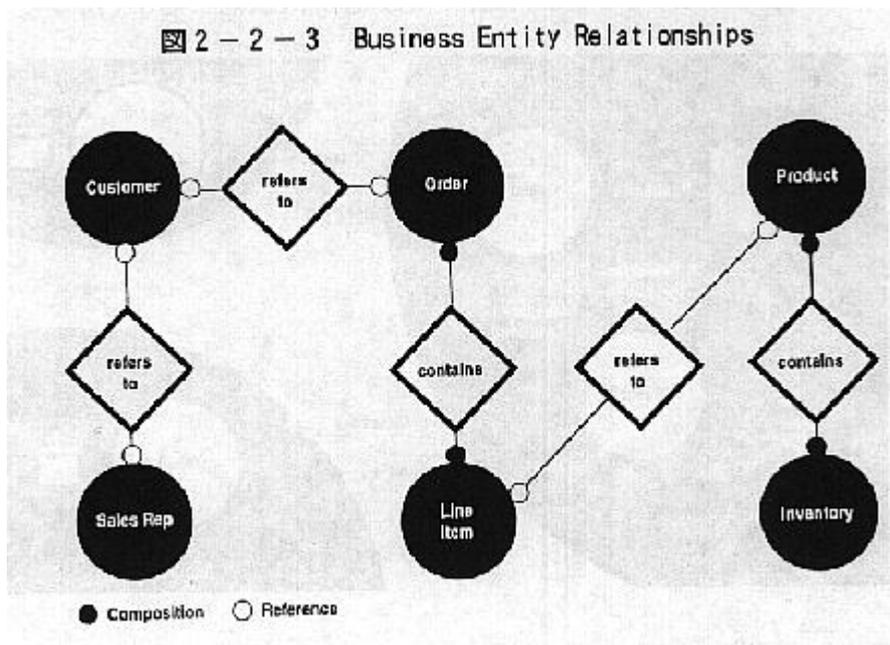


図 2-2-3

③プレゼンテーション領域におけるビジネスオブジェクトモデル

BPCSにおけるビジネスオブジェクトモデルは、ユーザーが直接操作してもよいオブジェクトである。またビジネスオブジェクトビューは、ビジネスオブジェクトを含んだ情報をただ表示するだけのGUIのウィンドウである。

次に、ビジネスオブジェクトの共有サービス領域におけるエンティティとの関係では、ビジネスオブジェクトビューへ表示している情報の更新の仲介役を担っている。そこでは共有サービス領域から送られてきたデータは、通常、クライアントワークステーション上にキャッシュされた状態である。そしてユーザーがビューを通じて更新のアクションをとったときには、ビジネスオブジェクトモデルが更新トランザクション生成を行う。

3) BPCSのメッセージング・アーキテクチャ

BPCSのメッセージングの基盤は、特徴でも触れているようにORBベースのCORBA準拠分散オブジェクトサービスとして実現されている。ビジネスオブジェクト間のすべてのインターフェースにはメッセージングが利用されており、アプリケーションの柔軟性を実現している。そこではオブジェクト間のメッセージのやり取りは、マシンが異なってもプラットフォームが異なっても行うことが可能であり、オブジェクトが完全にカプセル化されている。

(図2-2-4 Distributed Messaging Between Objects)

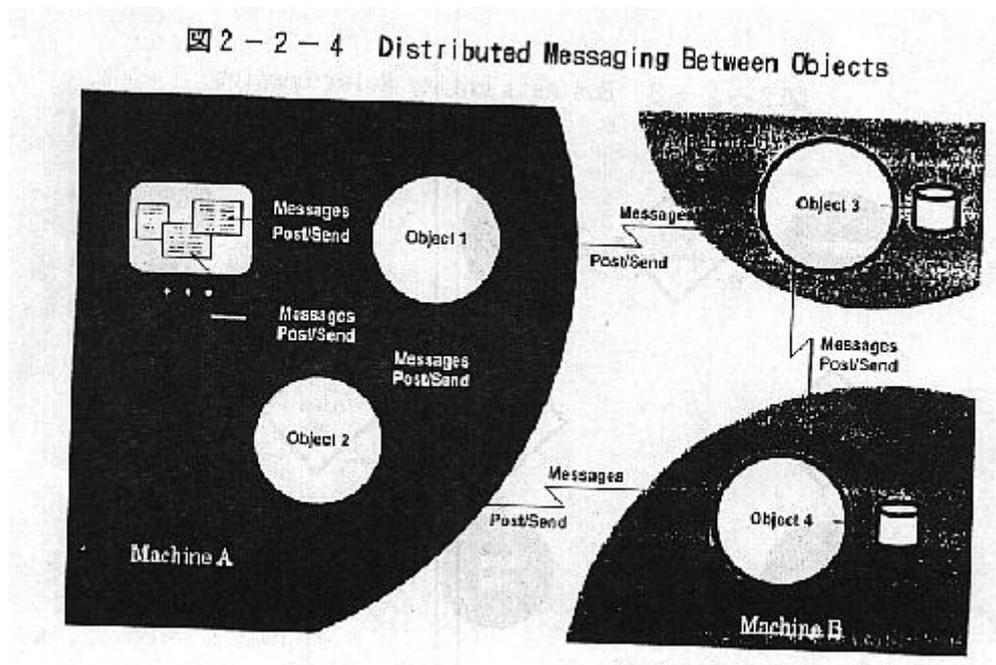


図2-2-4

4) BPCSの分散オブジェクト・アーキテクチャ

BPCSの分散オブジェクト・アーキテクチャは、次の3つに分割される。

- ①プレゼンテーション領域(PLD:The Presentation Logic Domain)
- ②共有サービス領域(SSD:The Shared Service Domain)
- ③資源管理(The Resource Manager)

これは一般的にC/SSモデルとして3階層構造と呼ばれているアーキテクチャである。PLDはユーザーの操作と作業中の状態の管理が対象範囲である。またSSDは企業の資源管理が対象範囲である。最後に資源管理はデータの永続記憶としてのリレーショナル・データベースである(ただしBPCS提供のコンポーネント範囲は、プレゼンテーション領域と共有サービス領域となる)。

①プレゼンテーション領域(PLD:The Presentation Logic Domain)

PLDは、アプリケーションの構成とシステム基盤の構成からなっている。(図2-2-5 BPCS Client/Sever Distributed Objects Architecture)

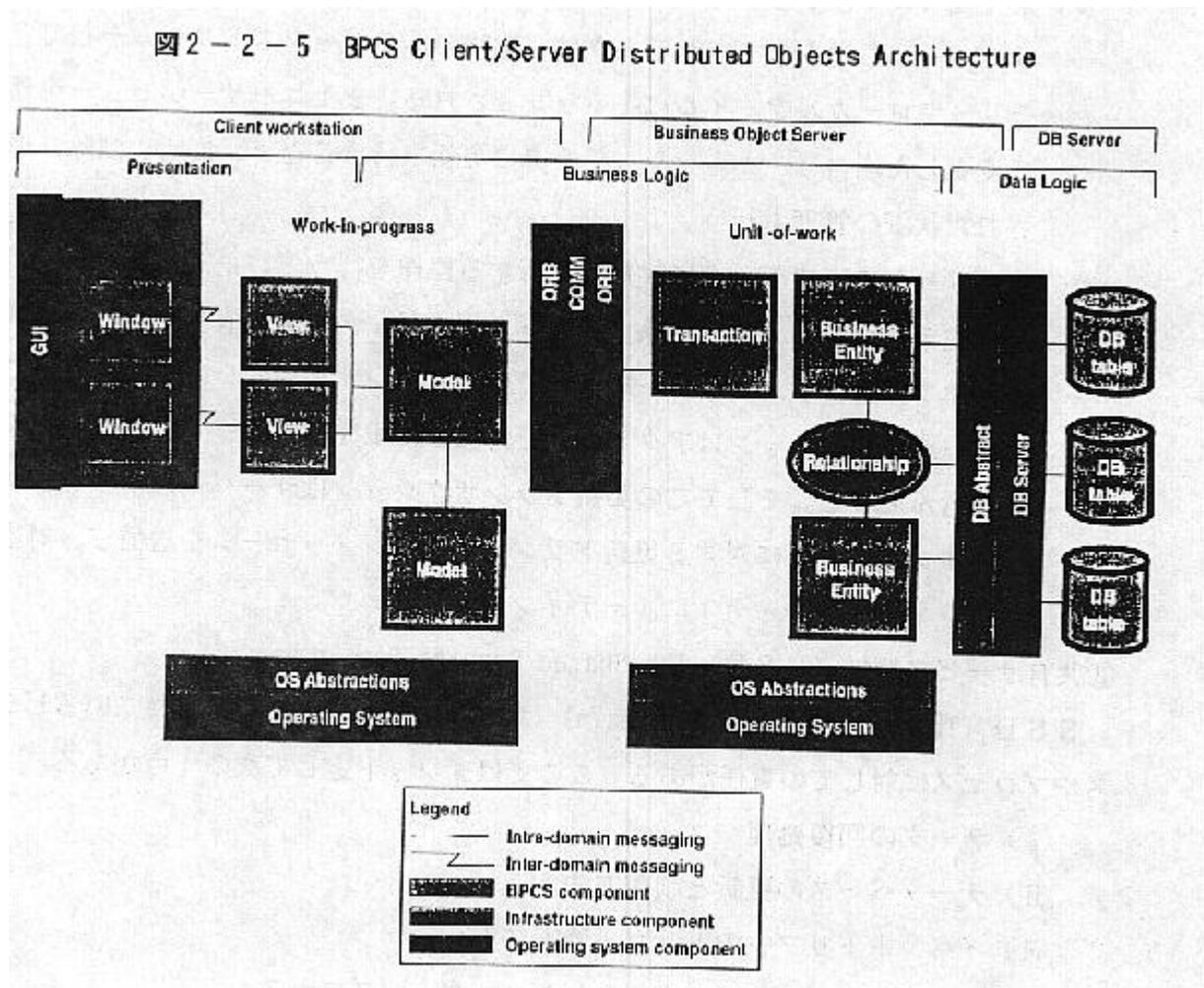


図2-2-5

アプリケーション構成は以下4項目で構成されている。

i) ビジネス・オブジェクト・モデル	ビジネスオブジェクトとして制御の中心部分。ユーザーはこのモデルを通じてサーバーに対してのデータ要求や更新トランザクション生成などを行う。
ii) ロジックのGUIビュー	このビューはウィンドウ全体とウィンドウ部品の管理を行う。ウィンドウとしてはユーザーのアクション(例えば入力フィールドにデータを入力したり、メニューを選択したりするアクション)に対して応答する。またサーバーへ応答を求めないようなビジネスロジックもその対象である。
iii) アプリケーションデータの要素	要素はユーザーの入力するそれぞれのウィンドウ部品が保証する範囲や正しいフォーマットで保証されたものである。
iv) 未完成(コミットされていない状態)データ	<p>サーバーから送られてきたデータは、意味のあるデータストリームとして、メモリーやローカルファイルにキャッシュされる。またユーザーがビューを通じて変更した場合は、キャッシュデータの更新も行うことと、更新、削除、追加された状況の管理も行う。</p> <p>ユーザーが未完成データに対して更新する操作を行うと、キャッシュされている未完成データを通じて、関係するデータオブジェクトを生成する。その生成されたオブジェクトがサーバーを通じて更新トランザクションに応答する。この更新トランザクションはアクションを付加して未完成のデータの管理下のデータを含んでいる。そしてこの更新トランザクションはサーバーのトランザクションオブジェクトに対する更新トランザクションメッセージを送信し、対応するビジネスエンティティにルーティングされていく。</p>

②共有サービス領域(SSD:The Shared Service Domain)

SSD(The Shared Service Domain)は、複数ユーザーに共有されているビジネスやプロセスに対しての責任がある。ここではメソッドとして次の4点がある。

- i)データの回復処理
- ii)データベースの更新と制御方法
- iii)イベントドリブンのトリガー確実な実行
- iv)ワークステーションで実行されたユーザーのプロセス

またSSDにおけるアプリケーションのコンポーネントとして次の5点がある。

- i)ビジネスエンティティ
- ii)関連
- iii)トランザクション
- iv)要素
- v)データベース

最後にSSDのシステム基盤として次の7点がある。

- i)アプリケーションコンポーネントのベースクラス
- ii)メッセージメカニズム
- iii)データベース抽象化
- iv)オブジェクトライフサイクル管理
- v)ネーミングサービス
- vi)イベントサービス
- vii)OSサービス抽象化

5)BPCSのオブジェクト指向性の評価

ビジネスオブジェクトがオブジェクト指向な製品であるか否かについての評価基準の1つとして、以下のオブジェクト指向の特徴的3要素で評価したい(表2-2-1)。ただし、この評価要素は、あくまでオブジェクト指向的か否かの評価であり、ビジネスオブジェクト利用に対する効果の評価ではないことは認識していただきたい。BPCSでは、表に示した3要素のすべてが実現されている。継承については、提供されたクラスをユーザーがGUI開発ツールによりサブクラス化して 再利用可能である。またカプセル化はメッセージングで、特にビジネス・オブジェクト間ではメッセージングを利用することで実現されている。

表2-2-1

表 2 - 2 - 1

評価要素	要素の説明	期待出来る効果
1. 継承	類似した構成要素集合(以下クラスとする)間の関連として上下関係を持たせ下位クラスは上位クラスの性質を引継ぎかつ新たな性質を追加出来るメカニズム。	再利用性向上
2. カプセル化	データと手続きをまとめた管理を行う事。オブジェクト内のデータおよび手続きに対しての外界からのアクセスを制御可能とする。それにより外部からのデータアクセスを手続きを通してしかアクセス出来ないようにコントロールすることが可能となる。	仕様変更が少工数化 オブジェクトの内部データを隠蔽し手続きだけを外部に開放する事で、オブジェクトの仕様と実装が分離
3. ポリモフィズム	各クラス間で手続き名を共有化し、実行時にオブジェクトが属するクラスメソッドが動的に実行させるメカニズム。	システムの拡張性 新規クラスの追加にたいして修正が不要となる

〈参考文献〉

- (1)SSA BPCS Client/Server Distributed Object Computing Architecture
- (2)95年度JUASオブジェクト指向WG報告書
- (3)Robert Orfali Essential Client/Server Survival Guide 富士ソフト出版
- (4)Robert E. Shelton BPR and Business Objects 講演資料