特集

コンポーネント指向開発の今後の展望

Prospect of Component-based System Development

あらまし

ネットワークコンピューティング環境の進展に伴い，企業の基幹系システムを支えるサーバコンポーネントへの期待がますます大きくなっている。本稿では，富士通で取り組んでいる，ビジネスオブジェクトのアーキテクチャと，それに基づく開発方法論，およびビジネスオブジェクトの標準化について述べる。

これらの技術開発において，これまでの情報システムの開発経験，データ中心型の方法論の適用など，富士通の経験を集約して進めている。アーキテクチャの特徴は，「コアビジネスモデル」と呼ぶデータベースの構造と制約に伴うアプリケーションの振舞いを集約した概念の導入にある。従来のデータベースを中心に形成されてきた業種のアーキテクチャを，この概念で再整理し，業種・業務に共通なビジネスオブジェクトの型を規定することをねらっている。これを技術基盤として，業務分析から実装までの開発方法論も併せて整備している。

Abstract

The rapid evolution of the network computing environment is placing greater demands on the server components that make up the basic structure of business systems. This paper describes Fujitsu s approach to business component application architecture, development methodologies, and standardization of business objects.

We have been making good progress in this area by drawing on our experience in information system development and data-oriented methodology study. Our architecture is characterized by the concept of the“ Core Business Model,”which represents the application behavior associated with a databases structure and its constraints. The conventional business architecture, which has mainly been based on databases, should be rearranged using this concept to create new business object models common to various applications.

We are also developing a methodology, for the stages from business analysis to implementation, on the above models.

橋本恵二（はしもと　けいじ）

1984年京都大学大学院工学研究科原子核工学専攻博士後期課程了。工学博士。同年富士通入社。以来事務系情報システムの開発技法およびツール開発に従事。
ソリューション技術統括部システム開発技術部

林　恵美子（はやし　えみこ）

1989年日本女子大学文学部教育学科卒。同年富士通入社。以来主として事務系情報システム開発の開発技法とツールの開発と適用に従事。
ソリューション技術統括部システム開発技術部

ま え が き

ネットワークコンピューティング環境の進展に伴い，企業の基幹系システムを支えるサーバコンポーネントへの期待がますます大きくなっている。標準的な業務モデルに従ったコンポーネントは，「ビジネスオブジェクト」と呼ばれ，OMG(Object Management Group)(注1)やサンフランシスコプロジェクト(注2)で先行的な試みが行われている。日本でも，ビジネスオブジェクト推進協議会(CBOP)(注3)が活動を始めている。

ビジネスオブジェクトが実用化する最大の課題は，ビジネスオブジェクトを含むアプリケーションの基本的な構造をいかに規定するかにある。CORBA(Common Object Request Broker Architecture)(注4)やEJB(Enterprise JavaBeans)(注5)という基盤となるコンポーネントモデルが普及しても，その上にビジネスオブジェクトが流通し，各企業がそれらを調達し，組み合わせてアプリケーションを構築できるためには，ビジネスオブジェクトを含むアプリケーションのアーキテクチャが必要になる。このコンポーネントアーキテクチャが，各コンポーネントサプライヤ，コンポーネントを利用する各企業の間でどの程度共有できるかが，ビジネスオブジェクトの適用にとって大きな課題である。

上記のビジネスオブジェクトに関する活動では，様々な角度から努力がなされているが，決して平坦な道ではなさそうである。「日付」「通貨」などの常識的な共通オブジェクトの品揃えと，ERP分野の代表的なビジネスオブジェクトが中心に議論されている。実用的なアーキテクチャを確立するためには，さらに多くの経験とアイデアが動員される必要がある。

富士通でも，これまでの多くの情報システム開発の経験に基づき，実用的なアーキテクチャの開発に着手している。従来のCOBOLを中心とした顧客情報システム開発経験の中で，業種・業務ごとに経験的なアーキテクチャが培われてきている。また，90年代当初より，データ中心型の開発方法論とそれに対応する統合的な開発環境を適用し，実業務をオブジェクトを通じてモデル化する文化とそれに基づいてアプリケーションを標準化するノウハウを蓄えてきた。さらに近年，オブジェクト指向言語によるフレームワークの開発と適用を，先進的な商用パッケージの開発に適用している。このような経験を集約し，実用性の高いアーキテクチャを目指している。これまでの情報システム開発で形成されてきた部品化・共通化の概念は，ビジネスオブジェクトとまったく別の方向にあるものではなく，自然な延長にあるべきである。いたずらに新しい用語や枠組みを強調せずに，これまでのアプリケーション概念の深い洞察の上に立った議論が必要であると考えている。

このアーキテクチャの上に，ビジネスオブジェクトを開発する方法論と環境の整備も併せて行っている。データベースを内包するシステムの中核部分に着目し，それにアプリケーションの振舞いを集約する形でビジネスオブジェクトをとらえ，それらを抽象化することによって，業種・業務に共通するビジネスオブジェクトのパターンを追究している。

本稿では，最初にコンポーネントアーキテクチャの基本的なねらいと構造を説明する。つぎに，それをもとにした開発方法論を述べた後，アプリケーションの中核となるコアビジネスモデルの概念とその抽象化・パターン化の取組みを紹介する。最後に今後の展望を述べる。

サーバアプリケーションへのコンポーネント適用の課題

Graphical User Interface(GUI)構築の分野では，ActiveXおよびJavaBeansという基盤となるコンポーネントモデルによって，開発言語や実行ミドルウェアによらない高機能のコンポーネント製品が，独立ベンダから提供されるようになっている。GUIにおけるコンポーネントの普及の条件は，以下の２点にある。

(1)　基盤となるコンポーネントモデルの普及

(2)　Plug&playの効果が画面上で実感できる開発環境

今後，サーバを中心とする本格的なビジネスアプリケーションに，コンポーネントが普及し，適用されるためには，上記の条件以上に，以下の条件を整備する必要がある。

(1)　機種やOSに依存しない，通信やトランザクション管理などの基本的なインフラサービスを提供するミド

(注1) CORBAを推進する非営利団体。大学，主要ソフトウェアベンダ，ユーザ企業など，800以上の団体が加盟している。
http://www.omg.org/

(注2) IBM社が中心となり，Javaによるサーバシステムのコンポーネントアーキテクチャを開発・推進しているプロジェクト。200以上のソフトウェア企業が参加している。
http://www.developer.ibm.com:8080/welcome/java/sfdhandr.html

(注3) ビジネスオブジェクト推進協議会（Consortium for Business Object Promotion）：国内の大学・ソフトウェア関連企業が参加するビジネスオブジェクトの標準化を推進する団体。
http://cbop.tiu.ac.jp

(注4) 機種に依存しない分散オブジェクト間の通信やその他のサービスを提供する機構。OMGが推進している。
http://www.omg.org/news/begin.htm

(注5) Sun Microsystems, Inc. を中心としたベンダグループで制定しているサーバ開発用のコンポーネントとコンテナなどに関する規約。
http://java.sun.com:80/products/ejb/index.html

ルウェア
(2) 基盤となるサーバコンポーネントモデルの普及
(3) サーバコンポーネントを簡便に組み込んで利用できる開発環境
(4) 互いに独立性が高く，自由につないで利用できる，ビジネスロジックまで踏込んだコンポーネントの開発技術

(1)と(2)については，着実に環境が整備されてきている。CORBAは機種に依存しない分散ネットワーク上の通信や実行時サービスの共通仕様である。さらにその上に，EJBという規約が制定され，サーバ上のコンポーネントとコンテナの世界共通インタフェース規約が確立されようとしている。これら仕様に準拠したミドルウェア製品もベンダから順次提供されてきている。[1] また(3)も，開発したコンポーネントの配置(deployment)をビジュアルに行えるツールの提供によって，利用しやすい環境が整ってくる。[2]

問題は(4)である。ビジネスロジックを内包したサーバコンポーネント，いわゆるビジネスオブジェクトが，流通可能なコンポーネントとして提供されなくてはならない。そのためのアプリケーションレベルのアーキテクチャを確立する必要がある。

このアーキテクチャは，アプリケーションを構成するコンポーネントの種類を規定し，各コンポーネントの役割と備えるべきサービス，それをコンポーネント外から利用するインタフェースを規定するものである。これらを規定するには，インタフェースの安定性，業務仕様の固有な部分と共通な部分の見極め，個々のコンポーネントの独立性，といった観点から，多くのビジネスアプリケーションについて深い洞察が必要である。

このような洞察は，プラットフォームや言語の違いはあれ，これまでの情報システム開発の中で，あるいは個々の企業の中で繰り返し行われてきたものである。その結果，オンラインやバッチプログラムの型を標準化し共通化した雛型，データベースアクセス機能を局所化しデータベースを隠蔽する部品，利息計算や日付計算のような計算用の小部品等々の形で蓄積されてきた。コンポーネントアーキテクチャの議論も，このような成果を，オブジェクト指向言語やコンポーネント基盤の上で展開してはじめて確立するものである。

筆者らは現在，基幹系システムで最も平均的なオンライントランザクション処理を想定したコンポーネントアーキテクチャを開発している。その他，例えばワークフロータイプ，データウェアハウスを含む情報活用タイプなども考えられる。これらは，逐次充実していく計画である。

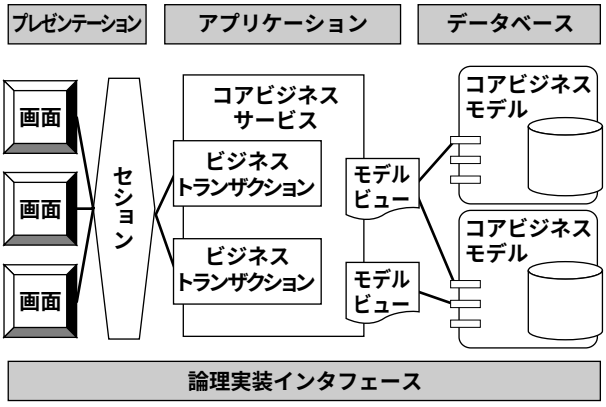

図-1 三層モデルによるビジネスオブジェクトのアーキテクチャ
Fig.1-Architecture for Business Objects based on three-tier model.

オンライントランザクション処理のアーキテクチャ

● アーキテクチャの構成

オンライントランザクション処理のアーキテクチャは，図-1に示すように，プレゼンテーション，アプリケーション，データベースの三層構造をもとにしている。プレゼンテーション層は，ユーザインタフェースに関連するサービスを受け持つ。アプリケーション層は，プレゼンテーション層で発生する様々なリクエストを受け付けるトランザクションのコントローラである。システムの最も中核のデータベース層は，データベースに蓄積されるデータに付随したサービス群を提供する。この基本的な考えに基づいて，各層に配置されるべきコンポーネントの種別，役割，相互のインタフェースを決めている。

このアーキテクチャで最も特徴的なコンポーネントは「コアビジネスモデル」である。このコンポーネントは，データベースに蓄積されるデータが持つ制約やそれに付随するサービスをパックにしたものである。ちょうど，実業務における「台帳」のアナロジになる。例えば，在庫帳を預かる係は，在庫帳によって在庫量を常に最新に管理する役割を持っている。現在の在庫量や引当の要求に対して，限度を超えない制約下で矛盾のない操作を任されている。同様の役割をこのコンポーネントが持つと考える。元来，業務の基本的な構造を，データベースの構造や制約に集約する方法は，データ中心型の方法論の柱である。筆者らは，この考えをコンポーネントアーキテクチャに発展させようと考えている。以下，アーキテクチャの各コンポーネントを説明する。

(1) コアビジネスモデルとモデルビュー

データベース層の設計要素で，業務上，常に整合性を

とっておかなくてはならないデータの集まりをコアビジネスモデルという。これは通常，一緒に更新されるような，ライフサイクルが同じビジネスエンティティのグループである。例えば，「受注」は，一般にヘッダと明細というエンティティ構造になる。明細を追加するとヘッダの情報も同時に変更を受ける。この制約は，元来，受注データ自身が持っている業務的な制約である。そういう関連の深いエンティティをまとめて一つのコアビジネスモデルと認識する。コアビジネスモデルは内部にあるデータ（ビジネスエンティティ）を隠蔽し，可能な操作を限定することでモデル全体の整合性維持に責任を持つ。

一般のビジネスオブジェクトの議論では，個々のビジネスエンティティをコンポーネントの単位と想定しているように見える。しかし，筆者らの経験では，いくつかのエンティティの協同的な振舞いの中に，典型的なビジネスルールがある。個々のエンティティだけをコンポーネント化しても，そのような典型的なビジネスの振舞いはコンポーネントの外に散逸してしまい，トリビアルなコンポーネントにしかならない可能性が高い。

コアビジネスモデルに対して，外から操作を依頼するときの引数や返値になるデータを，モデルビューという。これは，実業務の「伝票」に相当する。在庫係に仕事を依頼するとき，入庫伝票や出庫伝票が渡される。そのアナロジである。

モデルビューは，実業務の伝票と同様，固有のデータ構造とそれに付随するビジネスルールを内包している。例えば，伝票内のデータ項目間の正当性のルール，ヘッダに対する明細行の制約などである。もともと伝票は，実業務の中でさまざまな機能を持って設計されている。例えば「納品伝票」と「受領書」がワンライティングでできているような場合，これらは別々の伝票ではなく，同じ一つの対象としてモデル化したほうが素直である。重要なビジネスルールもこれらの伝票に共通にある場合が多い。例えば「伝票一回の受注では商品を５件まで指定可能」という制約は，二つの伝票で共通している。

(2)　セション

ユーザが一連のコンピュータとの会話を完了し，業務的な目的を達成する単位である。例えば，受注業務のように，在庫の確認→受注の新規登録→受注確認票の印刷など一連の会話の単位である。このコンポーネントは，ユーザ会話をコントロールするための画面制御を行い，後述のビジネストランザクションを利用して業務目的を達成する。

一般に，この部分は，ユーザの操作性要求，画面レスポンス，などを考慮して，会話手順が設計される。そのため，極めて要求の多様性が大きく，汎用的にコンポーネント化することは難しいと思われる。そのため，会話制御を外から与えてコントロールするようなツールがむしろ有益であると考えている。

(3)　ビジネストランザクションとコアビジネスサービス

上述のセションは，ユーザ要求による多様性が極めて大きい部分である。これを吸収するためにビジネストランザクションという概念を導入する。これは，会話手順などに依存しない純粋に業務的なイベントに対応する。例えば，セション「受注処理」では，「在庫数量を確認し，あれば受注内容を登録する。変更の場合も同様」というような一連の会話が設計されているとする。ビジネストランザクションは，「在庫確認」「新規受注登録」「受注変更」の三つを設定する。これらのビジネストランザクションは，たとえ他のセション（例えば，出荷実績処理セションで実績に合わせて受注変更が発生するような場合）でも，その文脈を知らないように設定し，共通化を図ることができる。

ビジネストランザクションは，自分が起動される文脈を知らず，状態概念を持たない。伝票のデータ構造に従って，モデルに対する最小作用素まで分解する役割を持ち，理想的にはそれ以上のビジネスルールは持たないように設計できるはずである。したがって，コンポーネントとしては，「N階層型更新」パターンのような型で，固有のロジックのほとんどないスケルトン的なもので提供可能と考えられる。

コアビジネスサービスはビジネストランザクションを業務機能単位にまとめるためのコンポーネントで，ビジネストランザクションはコアビジネスサービスのメソッドになる。これは，クライアントからのサービスを簡潔にするための工夫である。

(4)　その他の課題

コアビジネスモデルやモデルビュー以外に，ダイナミックな振舞いが実業務には存在する。これは，例えば「出荷実績が上がった後，月締めをはさんで受注の取消しは受け付けない」というようなビジネスルールである。このようなルールは，一つのコアビジネスモデルに持込むのは難しいし，個々のセションやビジネストランザクションのルールでもない。むしろ，異なるセションにまたがるルールである。

このようなルールは，ワークフロー的なモデルでとらえるのが自然であろう。筆者らは，現在この種のルールが，他のコンポーネントとうまく独立に扱えるか，検討

を進めている。

● コンポーネント指向開発方法論

－ComponentAA/BRMODELLING－

上記のようなアーキテクチャを中核とし，業務分析から一連の開発方法を規定している。この方法論ComponentAA/BRMODELLING(Component Application Architecture/Business Rule Modelling：以下，CAA/BRMODELLING)は，図-2に示すように，開発の作業手順，ドキュメント体系を定めており，コンポーネント指向の開発者に対して，アーキテクチャに従った開発を支援する。以下に，この方法論における開発作業上の特徴とドキュメント体系を紹介する。

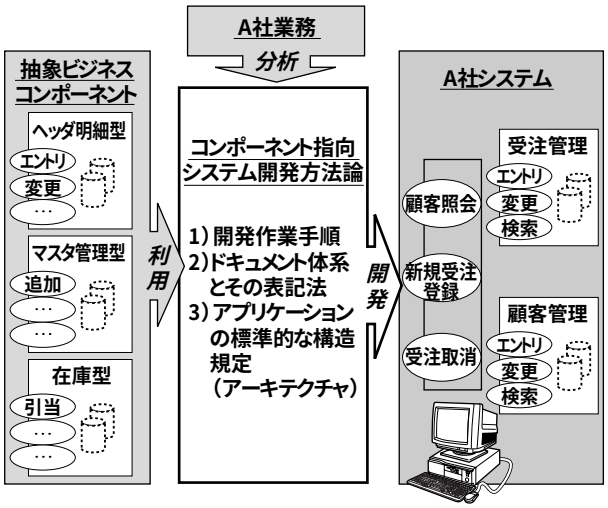

図-2 コンポーネント指向システム開発方法論
Fig.2-Component-based application development methodology.

● 開発作業手順

アーキテクチャの構成で述べたアーキテクチャをベースに，システム企画工程から，運用テスト工程までの作業項目(47項目)の手順と，インプット，アウトプットを規定する。システム開発作業標準SDEM90上のカバー範囲を図-3に示す。以下，分析・設計工程を中心に特徴を挙げる。

(1) システム分析工程

システム分析工程は，10年の実績のあるデータ中心型の方法論AA/BRMOLDELLING[(3)]のシステム分析技術を発展・継承し，これに，オブジェクト指向の標準的表記法であるUMLを取り込んでいる。ビジネスプロセスの分析に始まり，ビジネスエンティティの構造とプロセスとの相互作用のモデル化を手順に従って実施できるようになっている。従来との違いは，ビジネスエンティティの構造分析にある。従来は，対象とする業務領域に対して自由なモデル化を行ってきたが，後述するコアビジネスモデルのパターンを前提として，分析を進める方法を採っている。

(2) ユーザインタフェース設計工程

利用者から見える外部仕様を確定するとともに，アーキテクチャをあてはめて実装クラスの外部仕様を決定する。このときも，システム分析工程で適用を決定したパターンをもとに，実装図を作成する。

(3) システム構造設計工程

各メソッドの内部仕様を決定し，コンポーネントのカ

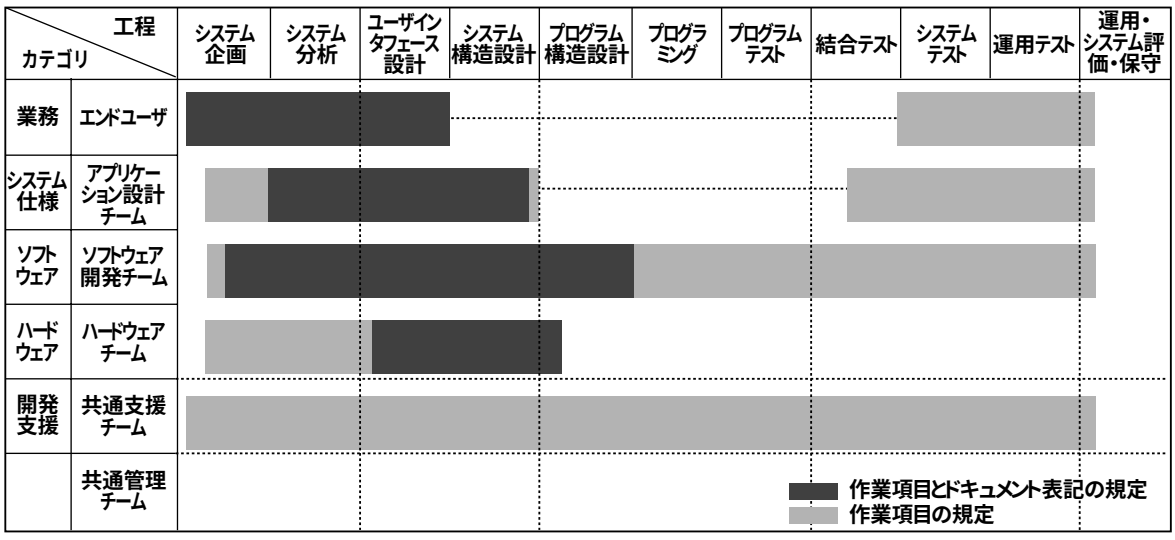

図-3 コンポーネント指向システム開発作業手順
Fig.3-Component-based application development methodology-WBS coverage on SDEM90.

表-1　ドキュメント体系とその表記法

工程	側　面	ドキュメントタイプ数	代表的なドキュメントタイプ名	表記法
分析	機能	4種	業務機能構造，システム化業務フロー	業務フロー，ユースケース図
	相互作用	3種	イベントシーケンス，コアビジネスモデル・ビジネストランザクションマトリクス	イベントシーケンス，マトリクス，シーケンス図
	エンティティ	3種	管理対象定義，エンティティ関連図	エンティティ関連図，クラス図
設計	クラス・メソッド	9種	コアビジネスモデル定義，ドメイン定義，メソッド定義，プロセス実装クラス図	クラス定義，インタフェース定義，メソッド定義，クラス図
	制御構造	4種	プロセスアーキテクチャ，ビジネストランザクション構造	アーキテクチャ図，シーケンス図
	補助	2種	コード定義，業務用語定義	—
ニーズ分析		7種	問題点ネットワーク図，手段体系図	—
規約，テスト仕様書など		26種	DB設計書，テスト仕様書，テスト報告書，画面レイアウト	—

ストマイズ仕様を固める。

● ドキュメント体系

CAA/BRMODELLINGで規定するドキュメントタイプ(58種)とその表記法を表-1に示す。表記法は，従来のAA/BRMODELLINGで規定したドキュメントに加えて，UML表記法を採用している。UML表記法は，オブジェクト指向開発の実質標準として採用されているが，各開発局面でどのような粒度や詳細度で記述するかの規定がないのが現状である。CAA/BRMODELLINGでは，その点を改善するため，実用的な記述レベルまで含めて，ドキュメンテーションをガイドしている。

ビジネスオブジェクトの標準化への取組み

● コアビジネスモデルの標準化

ビジネスアプリケーションは，これまでそれぞれの業種で歴史的にある種の「型」が形成されてきた。銀行の勘定系，契約データベースを中心とする保険業システム，部品管理データベースを中心とする生産計画システム等々である。これらはそれぞれ業種のアーキテクチャと呼べるものである。

筆者らは現在，これらのアーキテクチャの再整理に取り組んでいる。上記の業種アーキテクチャは，中核となるデータベースの構造やデータの制約に従って，アプリケーションの振舞いが決まっている。このデータベースの基本構造と制約が同種であれば，アプリケーションの振舞いも併せてコンポーネント化が考えやすい。例えば，航空機の座席予約システムと在庫管理システムは，座席と商品という違いはあれ，有限の資源を取り合う，という意味で共通の振舞いをしている。これは，座席や商品を管理するデータベースの構造と制約が類似していることによる。この考え方を押し進め，前述のコアビジネスモデルという概念を導入した。コアビジネスモデルにより，業種を超えた標準化が可能になると考えている。

む　す　び

本論文では，コンポーネント指向によるビジネスアプリケーション開発の基礎となるアーキテクチャと，それに基づいた作業手順・ドキュメント体系の整備状況を紹介した。また，コアビジネスモデルをビジネスオブジェクトの中核と考え，ビジネスオブジェクトの標準化を進めるシナリオの一端を紹介した。

ビジネスオブジェクトの普及のためには，CORBAやEJBの基盤技術以上に，今後ますますアプリケーションのアーキテクチャが重要な要素になることは間違いない。このためには，ビジネスルールを分析・モデル化し，パターン化していく努力が必要である。富士通には，10年来の業務モデリング技術の蓄積がある。これを生かして，アーキテクチャの精密化と適用範囲の拡大，さらにビジネスオブジェクトの開発環境の充実に努力したい。

参考文献

(1) 藤井：サーバコンポーネントのインフラ技術．*FUJITSU*, 50，2，pp.84-87(1999).
(2) 濱野ほか：コンポーネントをベースとしたアプリケーション開発環境．*FUJITSU*, 50，2，pp.88-93(1999).
(3) システム分析・設計技法AA/BRMODELLING解説書．富士通，1993.