

## ハイブリッドEAIモデルの技術的検討

既存のシステム資産を活用するためのEAIモデルには、端末系EAI基盤とサーバー系EAI基盤を分けた上でこれを連携させる「ハイブリッドEAIモデル」が有効である。本稿では、このモデルを構築する上での技術的なポイントと、NRI(野村総合研究所)が提供する本モデルのソリューションについて解説する。

### ハイブリッドEAIモデル

本誌9月号でも述べたように、ハイブリッドEAI(Enterprise Application Integration)モデルは、端末系を統合する「フロントEAI」とサーバー系を統合する「バックエンドEAI」という2つのEAI基盤を用意し、これを使い分ける点に特徴がある。また、「システム拡張性」「ノード種別」「通信処理形態」「Web 3

階層対応」という4つの観点から基盤構成が考えられている。これにより、性能面や運用面において有用性の高いEAIシステム構築が可能となる(図1参照)。

具体的にハイブリッドEAIモデルを実現するためには、フロントEAIとバックエンドEAIの2つのEAIソフトウェア製品(以下、EAI製品)を組み合わせて適用する。バック

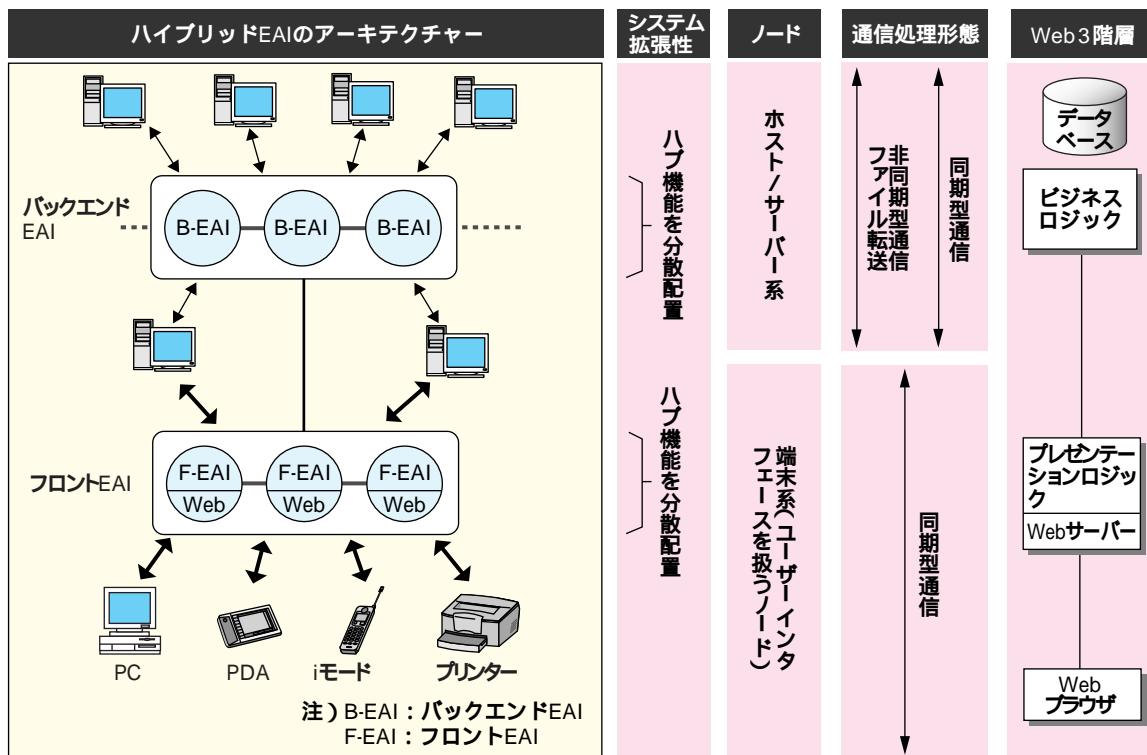


図1 NRIの提唱するEAIアーキテクチャ(ハイブリッドEAI)モデル

エンドEAでは、米国IBM社のMQSI、米国New Era of Networks社のe-Biz Integratorなどさまざまな製品がある。また、フロントEA製品としては、NRIから「オブジェクトワークス」が提供されている。

### ハイブリッドEAモデル構築のポイント

#### (1) アプリケーション開発フレームワークの導入

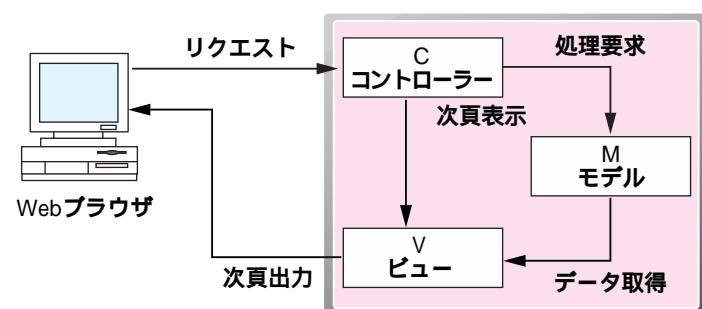
昨今のWeb技術の急速な普及とともに、システム開発はWeb 3階層システムによるものが標準となっている。ところが、Webシステム開発を行う際には、プレゼンテーションロジック（画面表示）とビジネスロジック（業務処理）が1つのプログラム上に混在したシステムとなりやすいという典型的な課題がある。一般に、このようなシステムは開発面や保守面においてコスト高となる。

この課題を解決するための方法として、MVC（Model View Controller）モデルと呼ばれるシステムアーキテクチャがある（図2参照）。ここでは、モデル（ビジネスロジック）とビュー（プレゼンテーションロジック）を明確に分離し、その連携制御をコントローラーが行うという設計思想に基づいている。実装には、J2EE（Java2 Enterprise Edition）技術が用いられており、モデルはJavaBeans（Javaプログラム部品

化の規格）またはEJB（JavaBeans仕様をサーバーソフト向けに拡張したコンポーネントモデル）で、ビューはJSP（JavaServer Pages：サーバー側のスクリプト処理技術）で、コントローラはServlet（Webサーバー上のJavaプログラム。主にブラウザからのリクエストに対する処理を司る）で構築するのが一般的である。

MVCモデルに基づいたシステム構築を行う際、Webアプリケーションサーバー製品単体で開発することも可能である。しかし、大規模なプロジェクト開発を管理するためには、設計の手順を含めた開発基準が必要となる。また、Webアプリケーションサーバー製品単体での開発では、JavaやServlet APIの使用が前提となり、熟練技術者の確保が困難となることも少なくない。

そこで有用となるのが、Web 3階層型システムにおける、MVCモデルに準拠したアプリケーション開発フレームワークの導入である。これにより開発者は、一定の開発基準・



M : モデル(ビジネスロジック)  
V : ビュー(プレゼンテーションロジック)  
C : コントローラー(ビューとモデルの連携制御)

図2 MVCモデル

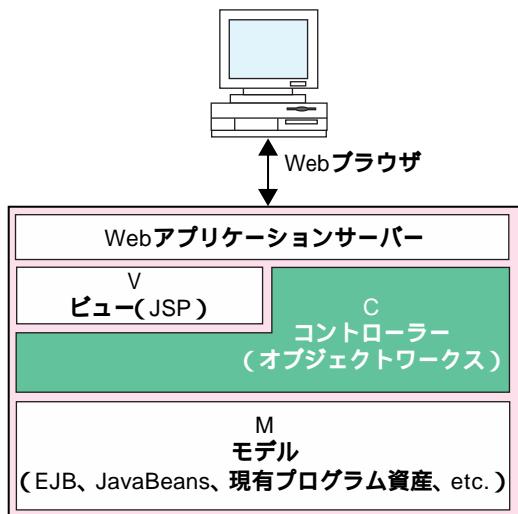


図3 フレームワークとしてのオブジェクトワークス

体系に基づいたGUI（グラフィカルユーザーインターフェース）ベースの開発環境を手にすることことができ、前述の課題を解決する。

一般にこのフレームワークはミドルウェアとして実現されるが、ハイブリッドEAIモデルでは、前述の「オブジェクトワークス」がフレームワーク部分を提供している（図3参照）。ここでは、実行系コンポーネント「オブジェクトワークス / CORE」と開発ツール「オブジェクトワークス / STUDIO」を活用することにより、高い開発生産性・保守性を備えたシステム構築が可能となっている。

## （2）フロントEAIIに実装される機能

次にポイントとなるのが、フロントEAIIへ実装する機能である。実践の観点から実装が欠かせないのは、以下の4つの機能である（表1参照）。

### 統合認証

企業システムにおいては、データ資産保護の目的で、ユーザー認証とアクセス制御の仕組みを各々のアプリケーションシステムに導入している。これらのセキュリティ制御をフロントEAIIの枠組みの中で実現することで、セキュリティ情報の一元管理とシングルサインオンなどの高可用性を両立するシステム構築が可能となる。

### プリント統合

企業システムにおいては、法定帳票出力などのプリント処理が重要となるが、一般にWebシステムの環境では、信頼性の高い印刷環境の構築が難しいとされている。そこでフロントEAIIでは、Web系の画面情報統合のほかに、プリント情報の統合・制御も重要な機能として求められる。

### バックエンド接続

企業システム構築では、コスト低減の観点から、現有のシステム資産の有効活用が求められる。そのためフロントEAIIでは、現有のバックエンドシステムへの接続を実現するコネクター機能が重要なものとなる。

### セッション管理

フロント部分にWebブラウザを用いたWebシステムでは、端末とWebサーバー間のセッションは保持されない仕様となっている。したがって、1つの業務処理が複数画面を遷移して実行される場合など、アプリケーションにとってセッション管理の機能は必須

となる。そのためフロントEAIによりセッション管理機能を提供することが求められる。

### (3) バックエンドEAIの実装

最後に、バックエンドEAIの実装のポイントについて述べる

#### BPM機能の導入

最近のバックエンドEAI製品には、BPM (Business Process Management) 機能の搭載をセールスポイントとしているものを見られる。BPMとは、業務フローを定義し、この定義内容に基づいて複数システムの実行やデータ連携を制御する考え方を指す。この考え方自体は新しいものではなく、一昔前に注目されたワークフローシステムのEA版と言える。EAの目的は複数システム間の連携にあるので、その連携ルールに業務フローを用いるというBPMの発想は自然なものである。

ただし、この場合、バックエンドEAIのハブの中にビジネスロジック (=業務フロー) をもつことになり、システムのメンテナンスポイントを増やすことになる。また、本来シンプルにデータ連携をするはずのハブ上の処理が複雑になるため、高性能のサーバーが必要になるなど、コスト面での課題が発生する。したがって、BPM機能の導入に際しては、目標を明確にした上で、費用対効果などを多面的に検討することが不可欠である。

#### 現有ゲートウェイサーバーの有効活用

#### バックエンドサーバー（ホストコンピュータ

表1 オブジェクトワークスのコンポーネントと機能

機能	コンポーネント
統合認証	オブジェクトワークス / OSM
プリント統合	オブジェクトワークス / CSP
バックエンド接続	オブジェクトワークス / BC
セッション管理	オブジェクトワークス / CORE

タなど）へのオンライン接続においては、バックエンドEAIを通さず、現有のゲートウェイサーバーを活用する方式も有用である。

バックエンドEAIを通してオンライン接続を行うと、ホスト用の通信プロトコルに対応する仕掛けを別途に用意する必要があったり、メッセージキューの枠組みでオンライン処理を指向したため、追加構築が必要になったなど、コスト高につながったという話も耳にする。現有のゲートウェイサーバーの活用も視野に入れて、どのような方式がベストであるか検討することが肝要である。

#### 最後に

EAはシステム連携の万能薬であるかのように謳っているパンフレットなどを目にすることがある。しかし、実際のEAシステム構築にあたっては、以上に述べたようなシステム基盤の方式設計を適切に行ることが重要となる。そのひとつの結論が、NRIがEAの実適用のなかで開発したハイブリッドEAモデル、およびフロントEA製品「オブジェクトワークス」なのである。

（野村総合研究所 八木晃二、藤岡賢治）