

次世代Webアプリケーションの基盤技術

e-ビジネスを成功に導くカギはシステムにある。e-ビジネスに求められるのは、短期間で構築でき、24時間365日の稼働に耐え、かつ頻繁な仕様変更に対応できるシステムである。本稿では、コンポーネント技術を取り入れた次世代Webアプリケーションサーバーの技術動向と、それが拓く新たな可能性について述べる。

隆盛の兆しを見せるサーバー側コンポーネント

Webサーバーには当初から、Webブラウザから与えられた変数に応じて動的に表示内容を生成するCGI（コモンゲートウェイインタフェース）が備わっていた。しかし、従来のクライアントサーバーシステムをリプレースするほどには、その機能は強力ではなかった。

そこで、登場したのがWebアプリケーションサーバーである。Webアプリケーションサーバーは、複数のページにまたがって状態変数の値を保持するセッション機能、レスポンスと処理能力を向上させるプロセス常駐化、拡張性と可用性を向上するクラスタリング（冗長化）機能、レガシーシステムなど他システムとの連携機能——の4要素を備え、Webアプリケーション実現の中核として欠かせないシステムとなった。

Kiva、NetDynamicsなど初期のWebアプリケーションサーバーは、上記の特徴を実現するための基盤として、CORBA（Common Object Request Broker Architecture）を採用し、言語非依存、他システムとの連携を実現していた。ただし、この場合、アプリケーション開発者はビジネスロジックだけではなく、トランザクション管理、セキュリティ管理など高度な

システムプログラミングまで行わなければならなかった。そのため、一部の金融、通信業界向けシステムに適用された程度で、広く普及するには至らなかった。

1999年12月、サン・マイクロシステムズ社がJavaによるWebアプリケーションサーバーの規格群である、J2EE（Java 2 Platform, Enterprise Edition）を発表したことで状況は大きく転換した。使用言語をJavaに絞ることで言語非依存にともなう煩雑さが解消されるとともに、サーバー側コンポーネント技術である、EJB（サーバー版に拡張されたジャバ部品規格）の導入により、システムプログラミングはEJBサーバーの担当、ビジネスロジックプログラミングはアプリケーション開発者の担当、と責任が明確に分離されたのである。

また、ビジネスロジックを包み込むコンポーネントは、J2EE準拠アプリケーションサーバー間での互換性を確保する設計になっているため、システム投資のリスクを抑制する効果も生まれた。結果としてBEA社、IBM社、iPlanet（Sun/Netscape 連合）、オラクル社など、マイクロソフト社を除く主要ベンダーのほとんどが、自社のWebアプリケー

ションサーバーをEJB準拠としたため、EJBがサーバーサイドコンポーネントの業界標準規格として確立しつつある。

EJBの仕様自体は発展途上にあるとはいえ、EJBを用いたソリューションは実用レベルに達している。すでにブリティッシュ・エアウェイズ社、アマゾンドットコム社といった、信頼性や拡張性が求められるサイトに、その使用例が見られる。

コンポーネント技術とは何か

なぜEJBが注目されるのか。それは、「イージーオーダー」かつ「アダプタブル（融通の利く）」なアプリケーションを実現することが可能になるからである。イージーオーダーとは、半完成品であるフレームワークを基に、短期間で作成する形態、アダプタブルとは、完成後も短時間で、柔軟に仕様を変更できる特徴のことである。

コンポーネント技術は、電子ブロックに例えることができる。電子ブロックとは、弁当箱大のケースにブロックを支える枠と電源、スイッチ、スピーカーが備わり、ブロックに包まれたトランジスタ、抵抗、コンデンサなどの電子部品を枠に差し込んでゆくことにより、電子回路を構築するものである。組み合わせによって、ラジオなどの機能が実現できる。コンポーネント技術は、この電子ブロックに類似した次のような特色をもつ。

- ・単体では動作せず、コンテナを必要とする。

- ・インタフェースが標準化されており、差し替え可能である。
- ・コンテナが発生させたイベントを引き金として、コンポーネントは入力値に対して定められた処理を行い、結果を出力する。
- ・コンポーネントの振る舞いは変数で制御できる。

EJBサーバーは、永続性、スレッド管理などを行うEJBコンテナに、セキュリティ、トランザクションなどの基本的な機能を付加したものである。上記の例でいえば、EJBコンテナは、ブロックを支える枠に、EJBサーバーは、枠に加え、スピーカーの配線や電源の変圧など、どの処理にも必要な基盤機能をあらかじめ備えたケースに相当すると言えよう。

従来のソフトウェアにも、サブルーチン、あるいはライブラリというコンポーネントに類似した形式が存在するが、コンポーネント技術とは次にあげる点で決定的に異なる。

従来のソフトウェアでは、メインプログラムはアプリケーション開発者が各自で記述し、共通ライブラリをメインプログラムから呼び出していた。これに対してコンポーネント技術では、メインプログラムの役割をコンテナが担い、アプリケーション開発者はコンポーネントを記述するだけでよい。つまりプログラムの主導権は、開発者からコンテナに委譲されているのが、この技術のポイントである。

結果として、プログラミングの単純化（スレッド、セキュリティなどの基盤プログラムが省け、ビジネスロジックを記述するのみ）、差し替えによる機能変更の実現、仕様の標準化による互換性の獲得——などの特徴が生まれている。

イージーオーダー・アプリケーションの実現

従来、大規模システムにおいては、一からデザイン、実装を行うカスタム開発が主流であった。これはいわば「オーダーメイド」システムである。一方、会計、生産管理など、特定分野においてはERP（統合業務パッケージ）の導入も行われている。こちらは「レディメイド」に相当する。

しかし、今日注目されているe-ビジネスシステムには、極めて短期間での開発、24時間365日の稼働、頻繁かつ柔軟な仕様変更への対応、などが要求される。ところが、カスタム開発ではすべての業務にわたって分析、設計、実装、テストを行う必要があるため、を満たすことができない。

一方、パッケージソフトでは想定以上のカスタマイズを行うと、大幅に導入コスト、工期が増加することが知られており、の実現が困難である。またどちらも、の条件を満たしつつ、ロジックを変更する柔軟性を装備するには高度な技術が必要とされる。つまり、今、e-ビジネスのシステムにふさわしいアプリケーション形態の登場が求められているの

である。EJBは、まさにこのニーズに応えるために登場した技術である。以下に、EJBの柔軟性と汎用性を示すポイントをあげる。

幅広い稼働環境

Javaでは、仮想マシンがプラットフォーム間の差異を吸収し、同一のバイトコードが複数のプラットフォーム上で動作する。さらに、J2EE規格により、コンポーネントの互換性も向上している。

データとロジックの分離

従来のライブラリでは、データ形式とロジックの分離が困難であった。結果的に汎用化する対象は日付計算、変換処理など、自らがデータをもたない、フィルタ形式の処理が多かった。EJBでは、ビジネスロジックが操作するデータをオブジェクトで表現し、実際のデータが格納されているRDBMS（リレーショナルデータベース管理システム）のスキーマと分離することで、汎用性を獲得している。

ドキュメントとソースコードの一致

従来の開発では、文書作成を特定のワープロに依存しており、時間経過とともに、保守が困難になる場合があった。この問題に対する解答として、Javaの場合は、JDK（Java開発キット）に含まれるJavadocというツールがソース構造とソースコード内に記述されたコメントを解析し、HTML形式のドキュメントを自動生成する。この機能を利用すると、文書を最新に保つ負荷を低減することができる。

アダプタブル・アプリケーションの実現

Java言語自体が備える、多態（抽象クラスで定義したメソッドに対し、個々の具象クラスで異なる振る舞いを実装することで、拡張性を確保する手法）と遅延バインディング機能（コンパイル時に呼び出されるオブジェクトを必要とせず、実行時に解決する方式）に加え、EJBによるインタフェースの統一化により、ソースコードを変更することなく、導入時に振る舞いを変更したり、本番稼働をさせたままロジックを差し替えたりすることが可能になる。24時間365日の稼働が前提のe-ビジネスにあって、この特徴は今までになく重要視されている。

また、さらに柔軟性を究めるためには、ルールエンジンの導入が有効である。変更が予想されるロジックの条件分岐部分を、Javaではなく、英語に類似した文法のルールスクリプトで記述し、エンドユーザーに変更を任せることによって、極めて敏速に対応させることが可能である。商用のルールエンジンには、Blaze Software社のBlazeAdvisor、ILOG社のJRulesなどがある。

コンポーネント技術による21世紀のシステム開発

コンポーネント技術は、これらの効用にとどまらず、システム開発方法を大きく変えるポテンシャルを秘めている。

■分散開発の実現

EJBによるインタフェースの統一は、遠隔

地での分散開発にも効果がある。EJBサーバーがトランザクション、セキュリティなどの基本機能を提供しているため、アプリケーション開発者はビジネスロジックの構築に専念することができる。EJBフレームワークという共通認識があるため、開発者間のコミュニケーションもスムーズになる。条件を整え、ビジネスロジック仕様さえ明確に伝えることができれば、一定品質のコンポーネントを遠隔で作成することが可能になる。

成功の条件としては、UML（Unified Modeling Language）による設計書の標準化、設計者によるビジネスロジック・インタフェースの定義、開発に必要な設計書、クラスライブラリ、ソースコードのリポジトリ化、チャット、メールを活用した開発情報の共有——などがあげられる。

■コンポーネント市場の形成

規格が標準化され、遠隔開発が可能になれば、設計者、実装者、利用者の三者をつなぎ、コンポーネントの供給を行う市場の形成も視野に入ってくる。

エンドユーザーはコンポーネントを購入するばかりではなく、自社がもつビジネスナレッジをコンポーネント化し、有料で販売、あるいは従量制のサービスを実現することもできるだろう。さらに、コンポーネントの運用サービスも付加されれば、ASPサービスの一形態として成立することも考えられる。

（NRIパシフィック 畠山紳一郎）