

# EJBを利用した開発における フレームワークの重要性

野村総合研究所 松田一彦

EJB (Enterprise JavaBeans) 技術を利用したビジネスロジック開発は、適切に導入すれば再利用性の向上など大きなメリットが得られるため、最近注目が集まっている。本稿では、EJB 技術を利用する場合の問題点や、業務レベルでの再利用性を高めるための方法について考察し、開発を効率的に行うためのフレームワークについて紹介する。

## EJBとは

EJBは、Java言語によるアプリケーション開発でビジネスロジックにあたる部分を構築するための標準規格である。EJBは、各ベンダーが提供するWebアプリケーションサーバーのEJBコンテナ上で実行される。EJBコンテナは、トランザクション管理やセキュリティ管理などの機能をアプリケーションに代わって担う。

## EJB利用時の問題点

EJB技術を利用することにより、コンポーネントのシステムレベルでの再利用性は高くなるが、業務レベルでの再利用性が高くなるわけではない。コンポーネントの機能範囲、コンポーネント間のインタフェースが標準化されない場合、業務レベルでの再利用性に関して以下のような問題が発生する。

### コンポーネントの機能範囲の違い

あるコンポーネントでは数値チェックなど項目チェックまで行うが、別のコンポーネントでは、クライアント側で実装することを想定して項目チェックを実装していないとする。これらのコンポーネントを組み合わせてある

機能を実現しようとするれば、機能の低いコンポーネントに合わせなければならないので、実質的に機能が低下する。

コンポーネント間のインタフェースの不統一  
コンポーネント間のインタフェースが統一されていないと、コンポーネントを呼び出すためにインタフェースの変換を行う必要が出てくる。これによりコンポーネントを再利用するための新たな開発が必要となり、再利用時の生産効率が低くなる。

## フレームワークの利用

コンポーネントの機能範囲とインタフェースの標準化を実現するためには、開発指針の枠組み、すなわちフレームワークを利用することが有効である。そしてフレームワークは、アプリケーション開発者にオブジェクト指向の高いスキルを要求せず、異なる業務でもコンポーネントの再利用性を向上できるものでなければならない。

そこでフレームワークには次のような点が求められる(図1参照)。

「継承指向」ではなく「組み立て指向」

コンポーネントを継承・修正していくためには、設計者にオブジェクト指向技術につい

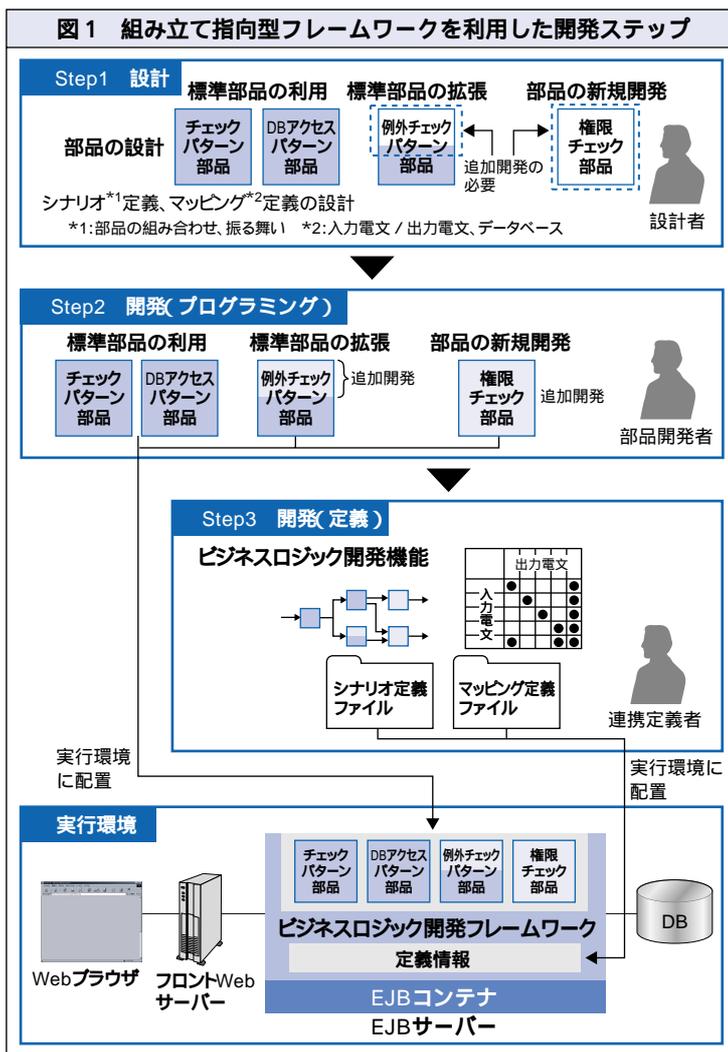
での高いスキルが必要となる。そこでフレームワークでは、コンポーネントを組み合わせてアプリケーションを構築する「組み立て指向」とし、部品の組み合わせやパラメータの変更だけで容易に修正できるようにすることが有効である。これによりクラスを修正する必要がある場合でも、個々のクラスは小さくなり、容易に修正が可能となる。

「業務部品」ではなく「汎用ソフトウェア部品」

異なる業務であっても同じような標準化手法を適用できれば、再利用性を向上させることができる。そのためには汎用のソフトウェア部品を充実させる必要がある。業務処理を含む部品は、汎用のソフトウェア部品を組み合わせることで作り、修正を容易にすることができる。

## NRIの「オブジェクトワークス」

EJBはシステムレベルでの再利用性を向上させるために有用な技術である。しかしながら、コンポーネントの機能範囲、コンポーネント間のインタフェースに何らかの規定を設けなければ、業務レベルでの再利用を実現できない。そこで、上に述べたような開発指針



の枠組みを提供するためのビジネスロジックのフレームワークが必要となる。

NRI（野村総合研究所）では、システム基盤ソリューションである「オブジェクトワークスR4.0」にEJBの開発フレームワークを追加機能として取り込んでいる。これを用いることで業務レベルでの再利用性を確実に高めることができる。