

Part 3: 実践編

ウィザードによるCMP EntityBeanの作成方法を詳しく解説 SymfowareとInterstageによる EJBの活用

富士通株式会社 ソフトウェア事業本部 朽木 忍 KUTSUKI, Shinobu

Symfowareは、メインフレームの分野で培った堅牢性/高信頼性を有しており、さらにInterstage Application Serverを用いることで、最新のJava技術を活用したミッションクリティカルシステムの構築を可能としている。ここでは、Symfowareを利用したEJBアプリケーションの構築を解説し、さらにInterstageとの組み合わせで実現する高度な機能を紹介する。

富士通では、Webシステム構築のためのアプリケーションサーバーとしてInterstage Application Serverを用意している。Interstage Application ServerはJ2EEに対応し、EJBを含む最新のサーバーサイドJava技術を数多くサポートしている。

ここでは、まずJ2EEに関する一般的な技術を解説し、次にEJBアプリケーション開発の実際を紹介する。最後に、Interstage Application Serverの特徴、特にSymfowareなどとのデータベース連携におけるEJB利用のメリットを説明する。

J2EEによるWebシステム

現在のWebシステムは、サーバーサイドJavaによる、いわゆるJ2EEベースの3階層システムの本格的普及期に入ってきている。Webシステムの初期段階では、Perlなどで開発されたプログラムをCGI(Common Gateway Interface)で呼び

出す方法が使われていた。しかし、Webアプリケーションがオンラインバンキングなどのミッションクリティカルな領域まで浸透しはじめると、サーバーのリソース消費の問題やトランザクション処理、セキュリティなどの要求にGGIモデルでは応えきれなくなった。そこで登場したのがJ2EE対応のアプリケーションサーバーである。

J2EEとは米サン・マイクロシステムズが提唱するJavaによる分散アプリケーション向けのコンポーネントアーキテクチャおよび規約で、Javaでのコンポーネント開発の標準仕様をまとめたものである。ここではJ2EE技術を用いたWebシステムの構築において、データベースとの親和性に優れたCMPベースのEntity Beanについて述べる(図1)。

Enterprise JavaBeans

Enterprise JavaBeans(以下、EJB)とは、分散オブジェクト指向のサーバーコンポーネント技術

である。データベースの管理や、コンポーネントのライフサイクル管理、トランザクション処理などの低レベルなインターフェイスは、EJBコンテナと呼ばれるアプリケーションサーバーが提供する実行環境により隠蔽され、開発者はロジック部分に専念できる。

このため、サーバーアプリケーションの開発においてEJB技術を用いる場合、以下のような利点がある。

生産性の向上

ミドルウェア、プラットフォームに依存しないアプリケーションの開発

Entity BeanとCMP

EJBには、大きく分けるとSession Bean(クライアントとの対話処理を担う)とEntity Beanの2種類がある。このうちEntity Beanとは、データベース処理を担い、データベースのレコードをオブジェクトとして操作可能にするコンポーネントである。一般的にEntity Beanはデータベースのテーブルに1対1で対応し、Entity Bean内のインスタンスが、テーブル内のレコードに1対1で対応する(図2)。

このEntity Beanは、さらにBMP(Bean-Managed Persistence)とCMP(Container-Managed Persistence)の2種類に分類される。

BMPの特徴は、EJBアプリケーションにデータベース操作文を発行する処理を記述し、EJBアプリケーションでデータの永続化を行なう。つまり、JDBCを使用してBean自身でデータベースにア

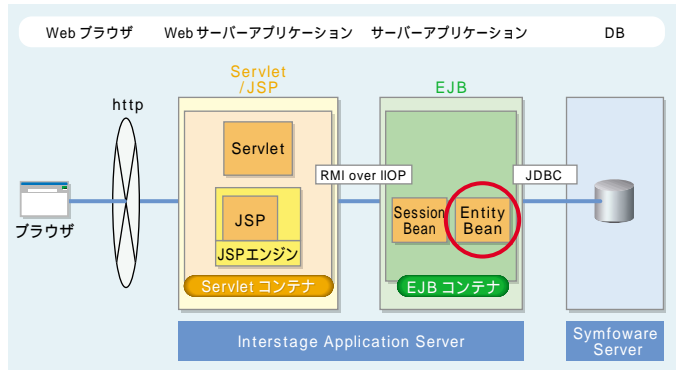


図1: Webシステム形態

クセスするということである。その分、コーディング量が増えスキルも必要であるが、CMPと比べ自由度が高く、細かい処理の設定が可能である。

これに対して、CMPはDBアクセスに関する情報をBeanの外に定義し、それをもとにEJBコンテナがデータの永続化を行なうのがデータの永続化を担うEJBコンテナなので、Entity Bean自身にデータベース操作文を記述する必要がない。したがって、CMPのEntity Beanを用いれば、開発者にデータベースの知識がなくても、ポータビリティ性の高いアプリケーションを開発できる。

なお、Interstage Application ServerはEJB 2.0仕様に準拠し、BMPとCMPの両方をサポートしている。

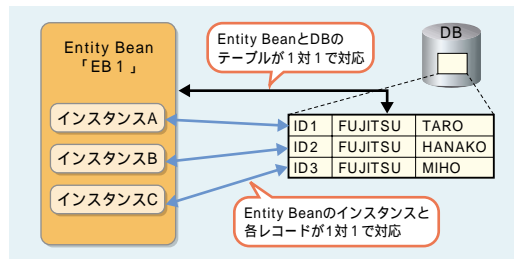


図2: Entity Beanの働き



画面1: プロジェクト作成ウィザード(種別、名前の設定画面)

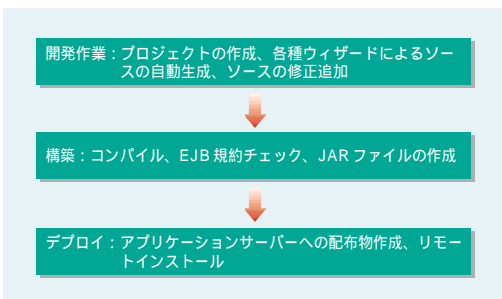


画面2: プロジェクト作成ウィザード(その他の設定画面)

EJB コンポーネントの開発の実際

Interstage は、統合開発環境 Interstage Apworks(以下、Apworks)を提供している。ここでは、Apworksを用いた開発の実際を紹介する。

ApworksでEJB コンポーネントを開発し、サーバーに配置する流れは以下ようになる。



開発作業

Apworksでは、アプリケーションを開発する際に、まず「プロジェクト」を作成する。

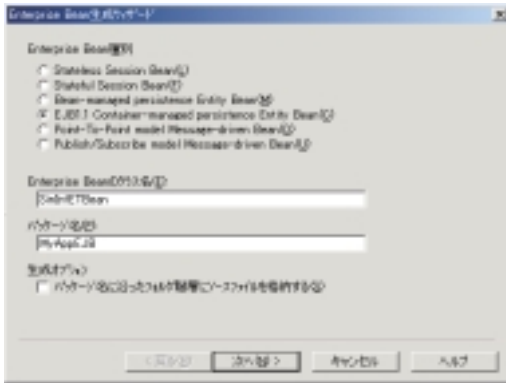
アプリケーションを1つの「プロジェクト」として作成し、このプロジェクト単位で開発資産を管

理する(画面1)。Enterprise JavaBeansの場合、複数のEnterprise JavaBeansを同プロジェクトに混在させ、1つのJARファイルを作成できるし、1つのプロジェクトに1つのEnterprise Beanを作成し、Enterprise BeanごとにJARファイルを作成することもできる(画面2)。

プロジェクトの作成ウィザードでEJBやServlet/JSPなどを選択すれば、目的のソースを自動生成するためのウィザードが表示され、そのウィザードとの対話を通して、アプリケーションの雛型が生成される。この後、必要に応じてソースの修正/追加を行なう。

EJBを開発する場合、ウィザードでは主に以下のような設定を行なう。

- 1.EJBの規約バージョンの選択
- 2.Enterprise Beanの種別 (Session Bean [Stateful/Stateless]、Entity Bean [BMP/CMP]、Message Driven Bean)、Beanクラス名、パッケージ名など
- 3.Deployment Descriptorの生成の可否、デー



画面6：Enterprise Bean 生成ウィザード

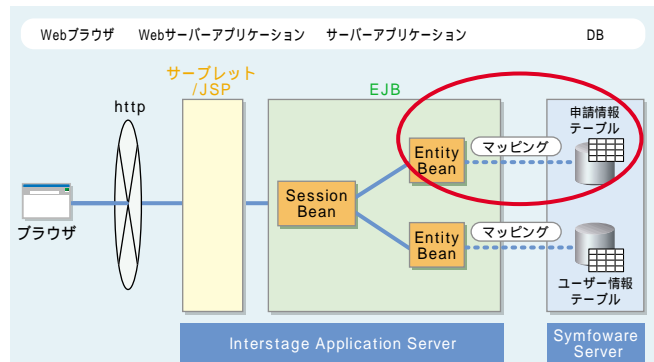


図3：年次休暇システム構成(EJB)

を通じて雛型のファイルやソースが自動生成し、サーバーへの配備までを行なう。こうしたウィザードを数多く提供することで、開発者のコーディング量を軽減し、高生産性を実現している。

年次休暇システムの開発

次に、実際にいくつかのウィザードを紹介しながら、Apworksを用いたEnterprise Bear(CMP)の開発を紹介する。ここでは、J2EEを用いたWebシステムとして、簡単な「年次休暇システム」を例にする。

このシステムには、年次休暇の申請機能とログイン機能が必要である。データベースには年次の申請情報を格納するテーブルと、認証時に必要なログイン名/パスワード名などの情報が格納されたテーブルがある。EJB コンポーネントは、2つのテーブルをマッピングしたEntity Beanが2つ、クライアントからのログイン情報のリクエストや

申請情報の登録などを行なうSession Beanが1つで構成されている(図3)。

ここでは、申請情報が格納されたテーブル(テーブル名: SINSEI)にマッピングしたEnterprise Beanを開発する。

プロジェクトの作成ウィザードでEnterprise Beanを選択し、「Enterprise Bean生成ウィザード」が表示されたら種別を決定する。ここでは、「EJB1.1 Container-managed persistence Entity Bean」を選択する(画面6)。

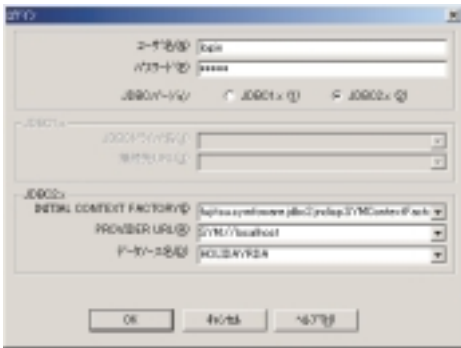
そして、画面7のウィザードでデータベースへの接続のログイン名、パスワード、データソース名など、接続する際に必要な事項を入力する。

これで開発環境であるApworksからデータベースに接続し^{注1}、永続化したいフィールドをGUIでソースに反映することができる(画面8)。

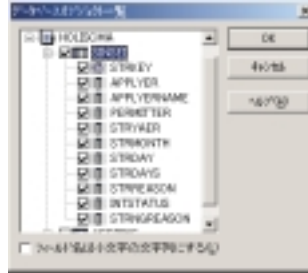
メソッド定義時に使用するクラスとejbCreateメソッドの定義はデフォルト設定のままにして、次のウィザードへ進む。EJBの規約上必要なメソッドは、Apworksのウィザードでデフォルトとして設定されているので、追加/変更がなければ特に手を加える必要はない。

規約上必要なメソッドに加え、新たにメソッド

注1：データベースに接続するには、あらかじめプロジェクトにデータベースへ接続するためのクラスパスを設定し、Symfoware RDA/RDB サービス、JDBC ネーミングサービスを起動しておく。



画面7: データベースへの接続設定



画面8: 永続化するフィールドをGUIで指定する



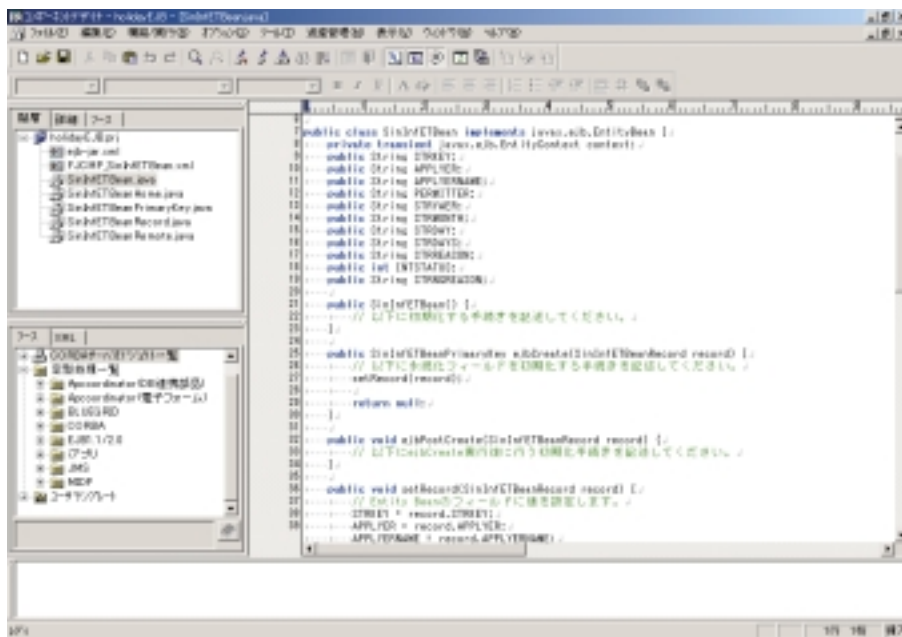
画面9: メソッドの追加

を追加したいときも、Apworksではウィザードを利用する。ここでは、プライマリーキーで検索を行なうfinderメソッドに加え、カラム「APPLYER」に対して申請情報を検索するfinderメソッドをGUIで追加している(画面9)。

このようなウィザードとの対話による設定が終了すると、Entity Beanアプリケーションを構成するファイルがプロジェクトに自動生成される

(画面10)。

EJBコンテナが、データベースにアクセスするために必要な永続化フィールドとデータベースのカラムとの対応付け情報(CMFマッピング定義と呼ぶ)や、finderメソッドの検索条件などの情報は、XML形式のファイル(CMP拡張定義ファイル)



画面10: EJBのソースコードがプロジェクトに追加される



画面 11 : deployment descriptor ファイル編集

ルと呼ぶ)に外付けで格納されている。

Apworksでは、Entity Bean生成後「deployment descriptor ファイル編集」と呼ばれるGUIベースの編集機能で、CMFマッピング定義や、finderメソッド定義などが編集できる(画面11)。

これでEnterprise Bean(CMP)コンポーネントの開発が完了した。あとは、Apworksでプロジェクトを構築し、生成されたJARファイルをアプリケーションサーバーにデプロイすれば良い。

以上のように、Apworksを用いて開発すると、EJBアプリケーションの作成に関するJavaの特別なスキルがなくても、ウィザードとの対話を通じて容易にCMPのEntity Beanが開発できる。

Interstage Application Server の先進機能

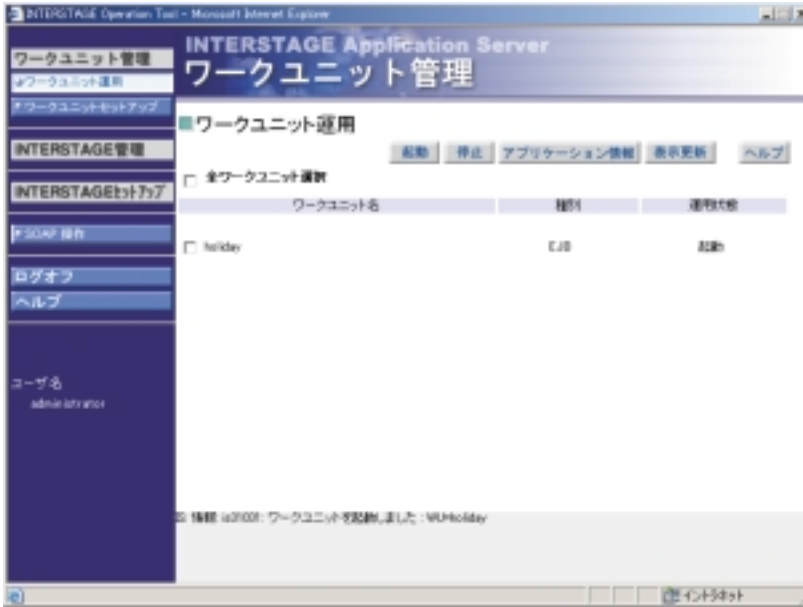
最後に、Interstage Application ServerのEJB技術の特徴とデータベース連携、特にSymfowareとの組み合わせにおけるメリットを解説する。

一般的に、CMPはBMPに比べ開発生産性/再利用性が高い反面、データアクセスをコンテナに任せるためにパフォーマンスが劣ると言われている。しかし、Interstage Application ServerのCMP環境は性能向上のためのさまざまな仕組みが実装されている。また、Symfowareとの親和性が高く、高信頼性/高性能/安定性を実現している。これらの技術について以下に解説する。

ワークユニット運用による高信頼性/高性能

Interstage Application Serverの独自機能に「ワークユニット運用」がある。これは、富士通のメインフレームで培ったノウハウを継承し、さらに発展させた技術と言える(画面12)。

ワークユニットとは、ファイルやデータベースなどの資源を共有するアプリケーションや起動/停止のタイミングが同じアプリケーションを、運用単位にグループ化したものである。ワークユニットにデータベースや環境変数などの実行環境



画面 12: ワークユニットの運用画面

をあらかじめ登録することで、設定された実行環境下でアプリケーションを動作できる。ワークユニット運用によるメリットをまとめると以下のようになる。

- EJB アプリケーションの一括起動の実現**
- 多重度の制御**
- アプリケーションのタイムアウト監視**
- 異常時の自動再起動**

これらによって、業務ごとにEJBアプリケーションシステムを独立させ、異常発生時の問題の局所化や復旧、性能チューニングが可能となる。

運用性の向上、運用管理作業の自動化

Interstage Application Server は、富士通の統合運用管理ソフトウェア「Systemwalker」と連携させることで、運用性を大幅に高め、運用や

管理作業を自動化できる。

Interstage Application Server 上で運用されているアプリケーションは、Systemwalker / CentricMGRを利用することで、稼働管理 / 性能管理が可能となる。トラブル発生時もネットワーク経由で復旧できるため、システム運用性を大幅に向上でき、Systemwalker / OperationMGRを利用することで、ワークユニットの自動実行 / 監視 / 操作などが自動化される。

高速な JavaVM の採用

JavaVMは、WebシステムをはじめとするJavaアプリケーションの実行環境として重要だが、常に運用上の課題となる部分でもある。EJBなどのJavaプログラムは、JavaVMによってプラットフォームのネイティブコードに変換されるため、実行性能はJavaVMの性能に大きく左右される。

Interstage Application Server は、サン・マイ

クロスシステムズとのアライアンスにより、JDK 1.3.1に搭載されたHotSpot ServerVMをチューニングした、高速なJava実行環境を提供している。ベンチマークテストでオリジナルVMの2倍以上、実用モデルでは20%の性能向上を実現している。また、デバッグ機能を組み込み、迅速なトラブルシューティングも実現している。

Entity Beanの高速データベースアクセスの実現

CMPのEntity Beanを適用する場合の留意点として、データベースアクセスの性能がある。一般的にEntity Beanを適用すると、データベースの処理対象件数に比例してアクセス回数も多くなり、パフォーマンスに影響する。これに対しInterstage Application Serverでは、データベースアクセス回数を削減し、高速アクセスを実現する技術を実装している。この仕組みは以下のようなものだ。

finderメソッドの処理でデータベース検索をする場合、通常Entity Beanでは、まずプライマリーキーを検索するSQL文(SELECT文)が発行される。次に、ビジネスロジックを実行する際に、再度取得したプライマリーキーの値を条件にした

SQLで、レコード情報を取得する(図4)。よって、SELECT文は、

最初のプライマリーキーを検索するときの1回 +
ビジネスロジック実行時にヒットするレコード件数

の分だけ発行されることになる。そのため、データベースの検索結果のレコード数が多いほど、Entity Bean(CMP)を実装したときのSQL文発行数は多くなる。

これに対して、Interstage Application Serverでは、最初のプライマリーキー検索時にレコード情報を取得するようにチューニングされている。よってCMPのEntity Beanを適用しても、SQL文の発行数は格段に削減される(図5)。

また、一度取得したデータベースのレコード情報を、コンテナにキャッシュする機能も備えられている。

Symfoware Serverとの連携

Symfowareは、スケーラビリティの向上/連続運転性/運用負荷の軽減など、ミッションクリティカルなWebシステム構築の要件を満たした

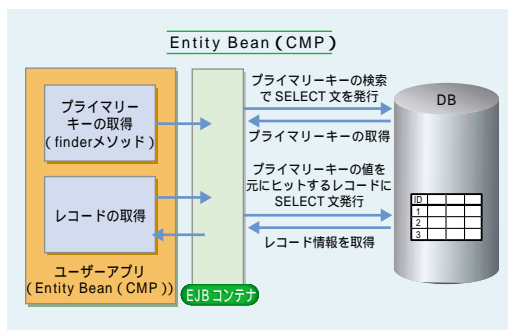


図4: 一般的なアプリケーションサーバーによるデータ検索

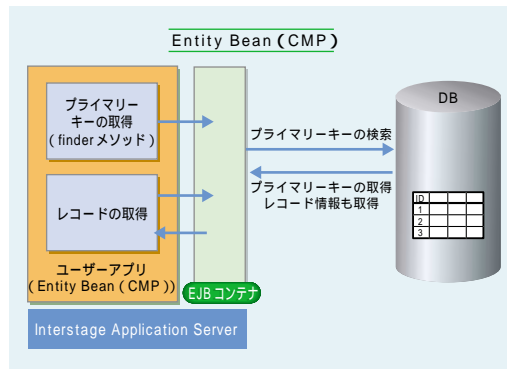


図5: InterstageのEntity Beanによるデータ検索

「Symfoware Server Connection Manager」を提供している。本製品とSymfowareおよびInterstage Application Serverを組み合わせることで、一層強固なシステムが構築できる。Symfoware Connection Managerのメリットには以下のようなものがある。

コネクション情報の自動回収

アプリケーションサーバーの異常時に、Symfowareのコネクション情報を自動回収する機能である。これにより、異常時の影響を最小限に抑え、業務の継続を保証する。

ロードバランサー機能

Symfowareのクラスタ運用時に、負荷状況を自動判断、各ノードに負荷分散を行なう機能である。ノード追加時もアプリケーションを意識する必要がなく、スケーラビリティが向上する

高速ノード切り換え

Symfowareのホットスタンバイ運用時に、待機系に事前コネクトする機能である。これにより、予期しないデータベースのサーバーダウンでもノード切り換えが高速に行なえる。また、アプリケーションの切り換え処理の考慮は不要となる^{注2}。

以上、本稿ではInterstage Application Server

の特徴、特にEJBを使用する際のメリットおよびSymfowareと組み合わせた場合のメリットを述べてきた。特にユニークなワークユニット運用や、高速JavaVMの採用、SQL文発行回数削減などの高速化技術により、EJB(CMP)の使用でも十分な実用性能を発揮できる。

さらに、Symfowareと連携した場合は、データベースサーバーのロードバランサー機能や、耐障害性を向上させる高速ノード切り換え、コネクションの回収機能などが用意されており、堅牢なWebシステム構築が実現できる。

また、Interstage Application Serverは数多くのツールベンダとアライアンスを結んでおり、負荷テストツールである「LoadRunner for Interstage」や、J2EEのパフォーマンスデータ測定ツールである「Precise/Indepth for J2EE」と組み合わせることで、システム構築時の負荷テストや検証も可能となっている。

これらによって、Interstage Application ServerはWebアプリケーションの構築に必要な機能をほぼ網羅した、現場の厳しい要件に十分耐えられるアプリケーションサーバーとなっている。



付録CD-ROMに、本文に掲載されている「年次休暇申請システム」のサンプルコードを収録しています(CD-ROMについては表紙裏をご覧ください)。また、Interstage Application Serverの試用版は下記URLからダウンロードが可能です。
<http://interstage.fujitsu.com/jp/>

^{注2}: ホットスタンバイ運用については、パート1で詳述している。