

EAI 技術が提供するネットワークソリューション

EAI Technologies and Network Solution

小林 雅 浩

要 約 昨今、多くの企業が、厳しい競争を生き抜くため、アプリケーション・システムを統合する必要性に注目し、企業は統合の検討と構築に着手しはじめている。

昨年から、統合に関する話題として、EAI (Enterprise Application Integration, 企業業務統合) が盛んに取り沙汰されている。EAI とは、統合対象に極力手を加えず、変更を最小限に抑えながらの統合を実現し、リスクの最小化と Time To Market の迅速化を図ることを主眼とする製品である。

本稿では、EAI の調査研究活動を通して得た知識をもとに、統合の問題点と、EAI を利用して統合を実施する場合に役立つ知識について解説する。

最初に、EAI 出現の背景となった従来の統合方法の問題点について説明する。

次に EAI 製品の機能を紹介し、最後に EAI の適用分野について記述する。

Abstract In these days, in order to win the severe competition among enterprises, many enterprises pay a close attention on the necessity of application systems integration, and begin to study and implement the integration of application systems.

Since last year, the word "EAI" (Enterprise Application Integration) has been used and cited briskly in the context of the systems integration. EAI is the product that realizes integration of application systems with minimum effort to rebuild and change existing application. Furthermore, it aims to minimize the risk of integration and shorten "time to market".

This paper describes the issues involved in applications integration and the methodology to resolve these issues using EAI solution based on the knowledge acquired through study and examination of EAI.

First, the background why EAI came on the scene is discussed, next, the functionalities of EAI products, and last, the field of application of EAI are discussed.

1. はじめに

昨今、多くの企業が、厳しい競争を生き抜くため、アプリケーション・システムを統合する必要性に注目し、統合の検討と構築に着手しはじめている。企業におけるアプリケーション・システムの統合の必要性に、以下の要因があげられる。

- ・ M&A や業務提携による複数のアプリケーション・システム統合
- ・ ERP (Enterprise Resource Planning) に代表される業務パッケージの導入によるレガシー・アプリケーション・システムとの統合
- ・ インターネットを使った新しいビジネスモデルに対応するため、Web アプリケーションとレガシー・システムとの統合

昨年から、統合に関する話題として、EAI (Enterprise Application Integration, 企業業務統合) が盛んに取り沙汰されている。EAI とは、統合対象に極力手を加えず、変更を最小限に抑えながらの統合を実現し、リスクの最小化と Time To Market

の迅速化を図ることを主眼とする製品である。

複数のアプリケーション・システムを統合する場合、理想的なのは、「完全な統合」である^[1]。「完全な統合」とは、ある論理的機能を実行する上で、ただ一つのデータ、1種類のコードしか使わないことを意味する。しかし、「完全な統合」を行うには、膨大な時間と工数が掛かり現実的ではない。「完全な統合」に対し、アプリケーション・システム間でデータやメッセージを連携して動作させる「見かけ上の統合」を行う方が現実的である。本稿では統合という場合、この「見かけ上の統合」を意味する。

本稿では、EAI 出現の背景となった従来の統合方法の問題点と EAI 製品の機能を紹介し、EAI の適用分野について記述する。

2. アプリケーション・システム統合における従来の問題点

本章では、最初に、アプリケーション・システムの統合手法について考察する。次に、従来のテクノロジーを用いた統合方法とその問題点を明らかにする。さらに、EAI を利用することにより、この問題点がいかに解消されるかについて説明する。

2.1 アプリケーション・システムの統合手法

複数アプリケーションを統合する手法には、データの再利用を目的とする「データ一貫性型統合」^[2]とアプリケーション間を機能レベルで統合する「複数ステップ型統合」、「複合アプリケーション型統合」があげられる。

次に、この3種類の統合手法と、既存のテクノロジーを用いた実現方法について考察する。

1) データ一貫性型統合

複数のアプリケーション・システムにおいて、同一エンティティについてのデータを冗長的に格納せざるを得ないことがある。例えば、顧客に関するデータは、会計、顧客サービス、ロジスティクスの各アプリケーション・システムに重複して格納される。

データ一貫性型統合は、アプリケーション・システム間で重複したデータの整合性をとることを目標とする統合手法である。つまり、顧客サービス・システムで「商品お届け先」の住所を変更したら、他のシステムのデータベースにも変更内容が反映される統合方法である。

データ一貫性型統合の特徴は、データベース・レベルでデータの整合性を保つことである。つまり、データの更新が行われた場合でも、アプリケーション・ロジックの実行を必要としない。

データ一貫性型統合を実現する方法として、データベースのレプリケーション機能を用いることが考えられる。しかし、レプリケーション機能は、統合対象となるデータベース間でデータ・モデルやデータ・セマンティックスが異なる場合には使用することができない。一般的に、別々のアプリケーション・システムは、別々の開発者が設計するため、データ・モデルやセマンティックスは異なる。従って、レプリケーション機能を使用することはめったにない。

データ一貫性型統合を実現するために広く活用されている方法には、ファイル転送を伴ったバッチ連携がある(図1)。この方法は、

- ① データベースから直接データを抽出
- ② 抽出したデータをファイル転送
- ③ そのデータを送り先アプリケーションのデータ・モデルとセマンティックスに合わせるためのデータ変換
- ④ 変換したデータをターゲットのデータベースに挿入

することで行われる。

この方法では、データの抽出や挿入処理、データ変換を行うプログラムを作成する必要がある。

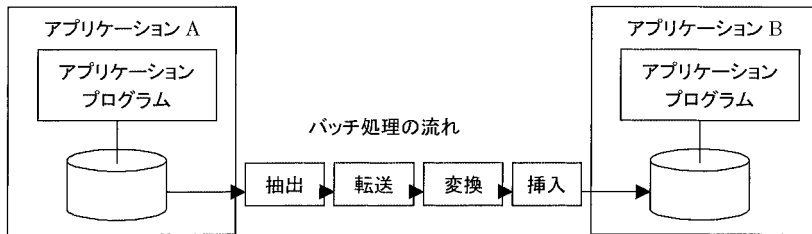


図 1 データー貫性統合

2) 複数ステップ型統合

複数の段階を経るビジネス・プロセスでは、一般的に、複数アプリケーション・システムの使用が必要となる。例えば、「見積り要求から現金支払いまで」の注文を処理するには、注文入力、与信、倉庫管理、ロジスティクス、顧客サービス、売掛金、といった各アプリケーション・システムが必要となる。これらのアプリケーション・システムが独立して存在し、統合されていない状態では、アプリケーション・システム間で帳票のやり取りと手入力作業を繰り返すことにより、一連のビジネス・プロセスが実行される。

複数ステップ型統合は、一連のビジネス・プロセスを自動的に制御することで、情報の再入力をなくし、それにより、入力エラーを削減すると共に、一連のビジネス・プロセス終了までの時間を短縮することを目的とする統合手法である。

具体的には、一連のビジネス・プロセスに対応したアプリケーション・プログラムを自動起動し、そのプログラムの完了した結果を受けて、後続アプリケーション・プログラムを起動する。さらに、先行アプリケーション・プログラムの出力データ（帳票データ）を後続アプリケーション・プログラムの入力データとして渡す処理も行う。また、データの受け渡しでは、出力データと入力データのシNTAXとセマンティックスを一致させる変換処理を行う必要がある。

複数ステップ型統合の特徴は、データ更新がアプリケーション・システムのプログラム・ロジックを介して行われる点である。データー貫性型統合と異なるところは、新規顧客データが追加された場合に顧客にメールを出すというようなアプリケーションの固有ロジックを考慮する必要がない点である。

複数ステップ型統合を実現する方法には、

- ・ファイル転送を伴うバッチ連携を利用する方法

- ・メッセージ・キューイングなどの非同期通信ミドルウェアを利用する方法がある。
どちらの方法でも、連携プログラムの作成が必要となる（図2）。
次に、これらの方法の適用例を示す。
- ・ファイル転送を伴うバッチ連携を利用する方法は、EDI（電子データ交換）などで利用されている。しかし、最新の情報を求めるユーザーの需要には対応していない。
- ・通信ミドルウェアを利用する方法の適用例としては、金融業界における、STP（Straight Through Processing）がある。この方法を適用することで、リアルタイムに近い速度で、データが伝播される。

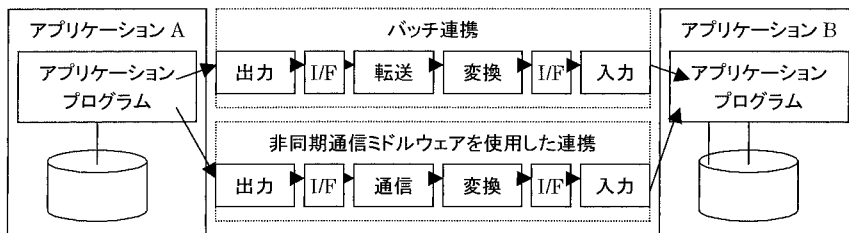


図 2 複数ステップによる統合形態

3) 複合アプリケーション型統合

アプリケーション・システムの開発では、既存アプリケーション・ロジックとの連携が必要となることがある。例えば、新たに Web ベースの出荷状況管理アプリケーションを開発する場合、顧客データベースへのアクセス機能のためにわざわざ新たなコードを作成しなくても、既存の顧客情報システムで使われている顧客ルックアップを行うアプリケーション・ロジックを呼び出すことで対応できる。

複合アプリケーション型統合は、既存アプリケーション・ロジックを再利用して、新しいアプリケーションに組み込むことで、開発生産性を高め、アプリケーション・システム間の機能重複をなくすことを目的とする統合手法である。

複合アプリケーション型統合の特徴は、三つの統合パターンのなかで最も緊密に構成され、必ずリアルタイムで動作することである。

複合アプリケーション型統合を実現する方法には、各種通信ミドルウェアや、TCP/IP、RPC（Remote Procedure Call）などの基本通信を利用する方法がある（図3）。これらの方法を利用した場合でも、アプリケーション間で、受け渡しするデータの文字コードなどを変換する必要がある。

2.2 従来の統合方法の問題点

前節では、アプリケーション・システムを統合するための3種類の統合パターンと、それを実現する従来の方法について記述した。本節では、従来の統合方法の問題点について説明する。

企業は、前節で説明した3種類の統合パターンを組み合わせることで、独立してい

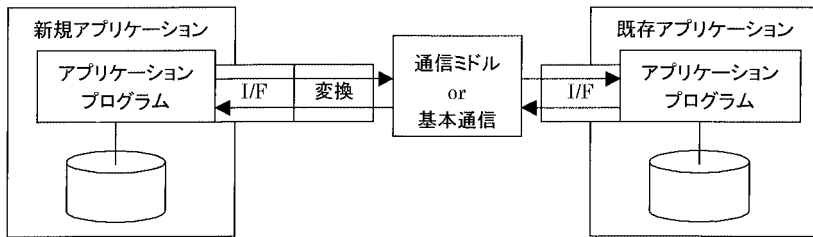


図 3 複合アプリケーション型統合

た多くのアプリケーション・システムを統合することができる。しかし、通信ミドルウェアやファイル転送を使用する従来の統合方法では、以下の機能を持つ連携プログラムを開発する必要がある。

- ・アプリケーション・システムのデータベースやプログラムの入出力データをアクセスするためのインタフェース機能
- ・取得したデータ形式を連携先アプリケーション・システムのデータ・モデル、セマンティックスやシンタックスに合わせるための変換機能
- ・データを連携先アプリケーション・システムに送信するための接続機能、もしくはファイル転送機能のインタフェース機能

さらに、連携プログラムが取得したデータを複数のアプリケーション・システムに輸送する必要がある場合、連携プログラムにはデータの輸送経路を特定するルーティング機能も必要となる。

次に、この連携プログラムを利用して、アプリケーション・システムを統合する場合の問題点について説明する。この問題点が、EAI 出現の背景となった重要なポイントである。

複数アプリケーション・システムの統合では、連携プログラムを使用することで、アプリケーション・システム間の接続が複雑になる。この状態をアプリケーションのスパゲティ状態という(図4)。

この状態では、情報システム部門や企業の経営上、さまざまな問題が発生する。情報システム部門における問題点として、

- ・N個のアプリケーション・システムを統合するには、 $N * (N - 1) / 2$ 個の連携プログラムの開発が必要である。
- ・新たにアプリケーション・システムを開発して、他のN個のアプリケーション・システムと統合するには、N個の連携プログラムの開発が必要である。
- ・あるアプリケーション・システムで、データ・フォーマットやインタフェースが変更になった場合、関連する連携プログラムの変更が必要である。

など、生産面や保守面での影響があげられる。また、企業の経営上の問題点としては、新しいビジネスモデルの実現、新しいサービスや商品の提供に素早く対応できないことがあげられる。

2.3 EAI を用いた統合

前節で、連携プログラムを作成する従来の統合方法では、アプリケーションのスパゲティ状態となり、生産性および保守性が悪くなることを説明した。本節では、EAI

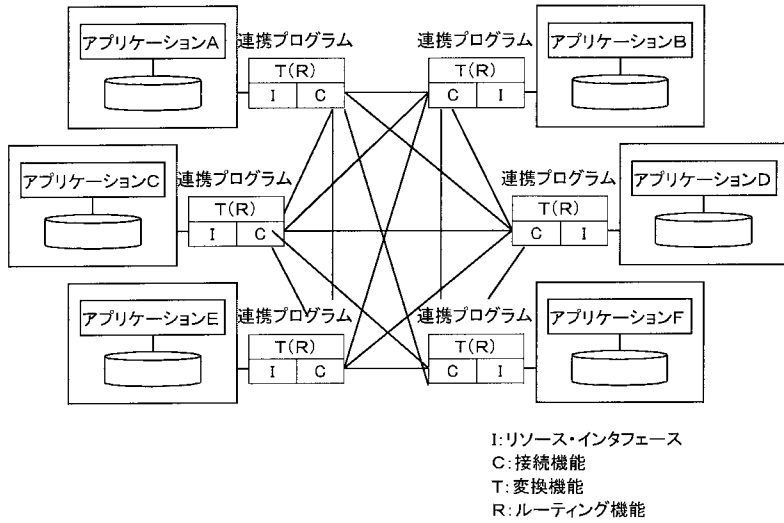


図 4 アプリケーションのスパゲティ状態

を利用することで、スパゲティ状態が解消されることを示す。

EAI は、論理的にはハブ・アンド・スポークの形態をとる (図 5)。この形態の優れた点は、アプリケーション・システムごとの連携処理部分を集中化することで、アプリケーション・システム間の接続インタフェースを最小限に抑えることができることである。従って、EAI は、アプリケーションのスパゲティ状態を解消する。

さらに、EAI は、連携時にコーディングを最小限にする機能を提供することで、統合における生産性と保守性を大きく向上させる。EAI の機能詳細は、第 3 章にて説明する。

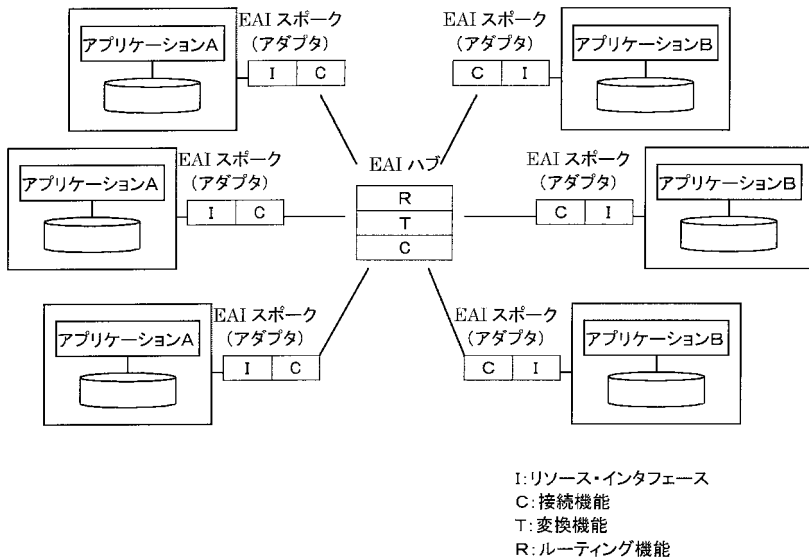


図 5 ハブ・アンド・スポークの形態

3. EAI の機能詳細

EAI の製品体系としては、「インテグレーション・ブローカ^[4]」と「ビジネス・プロセス・マネージャ」に大きく分かれる^[3]。ビジネス・プロセス・マネージャは、インテグレーション・ブローカが発展したものである。

この二つの製品体系における機能の違いは、

- ・インテグレーション・ブローカは、データの輸送経路を一つずつ指示するルーティング機能を実装している。つまり、一つの接続部分の処理の流れを定義する。
- ・ビジネス・プロセス・マネージャは、ビジネス・プロセス全体の処理の流れとデータの輸送経路を定義する、ワークフロー機能を実装している。

点である。つまり、ビジネス・プロセス・マネージャの方が、ビジネス・プロセス全体を設計できる点が優れている。図6にEAI製品の定義範囲を示す。なお、図6で示すルーティングはデータ変換とルーティングを含んでいる。

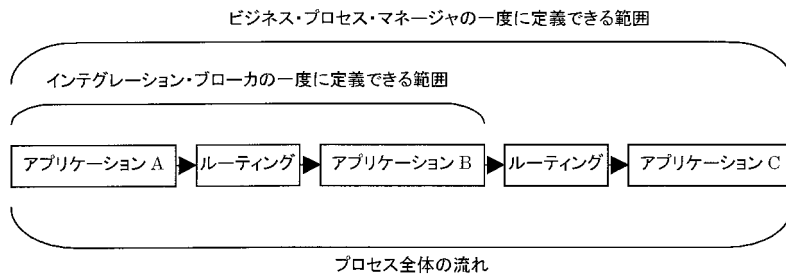


図 6 EAI 製品の定義範囲

3.1 節ではインテグレーション・ブローカの機能を、3.2 節ではビジネス・プロセス・マネージャの機能を中心に説明する。但し、インテグレーション・ブローカのほとんどの機能はビジネス・プロセス・マネージャと同じであるため、3.2 節ではインテグレーション・ブローカの機能と異なる部分を中心に説明する。

なお、製品ごとに機能差があり、全ての機能を満たす製品があるわけではなく、製品に共通する機能の説明であることに注意していただきたい。

3.1 インテグレーション・ブローカの機能

本節では、インテグレーション・ブローカの機能について説明する。図7にインテグレーション・ブローカの機能配置を示す。それぞれの機能詳細については後述する。

一般的なインテグレーション・ブローカの各機能のノード配置は、次の通りである。変換サービス機能とルーティング機能は、インテグレーション・ブローカのサーバ機能としてサーバ・ノード上に配置される。アダプタは、連携するアプリケーション・プログラムと同一ノードに配置される。接続サービス機能は、インテグレーション・ブローカ・サーバとアダプタ間の通信を行うため、アダプタとサーバ側に配置される。

3.1.1 アダプタ (リソース・インタフェース機能)

統合対象となるアプリケーション・システムのリソースは、さまざまなプロトコルで実装されている。例えば、

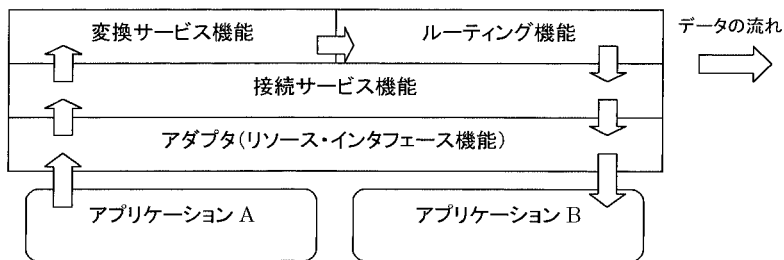


図 7 インテグレーション・ブローカの機能配置

- ・ TPM (Transaction Processing Monitor) やメッセージ・キューイングなどのミドルウェア製品を使用したプログラム
- ・ RDBMS , ISAM や CODASYL 型データ・モデルをサポートするさまざまなデータベース製品
- ・ 独自 API (Application Program Interface) を使用する ERP に代表される業務パッケージ製品

があげられる。

インテグレーション・ブローカは、異なるプロトコルを吸収するために、アダプタと呼ばれるリソース・インタフェースを標準で提供する。

次に、EAI 製品が提供する一般的なアダプタの種類 (図 8) と機能について説明する。

1) 業務パッケージ用アダプタ

業務パッケージが提供する API を実装し、業務パッケージが管理するデータの取得や業務パッケージの各処理プログラムを実行することができる。

2) ミドルウェア用アダプタ

ミドルウェアが提供するクライアント関数を実装しており、ミドルウェアを使用して作られたサーバ・アプリケーション・プログラムと連携することができる。

3) データベース用アダプタ

データベースを操作する言語や ODBC などのプロトコルを実装し、直接データベースをアクセスすることができる。

4) 企業間データ交換用アダプタ

企業間取引 (EDI : 電子データ交換) , 金融業界 (S.W.I.F.T) や医療業界 (H 17) などの、業界標準の通信プロトコルを使用して、企業間の情報交換をサポートする。最近では、ロゼッタネット (RosettaNet) で使用される XML (eXtensible Markup Language) データをアクセスできるアダプタも提供する。

5) その他のアダプタ

FTP , HTTP やメールなどのプロトコルを経由して、データ交換することができる。

3.1.2 接続サービス機能

接続サービス機能は、アダプタとインテグレーション・ブローカ間を接続する。こ

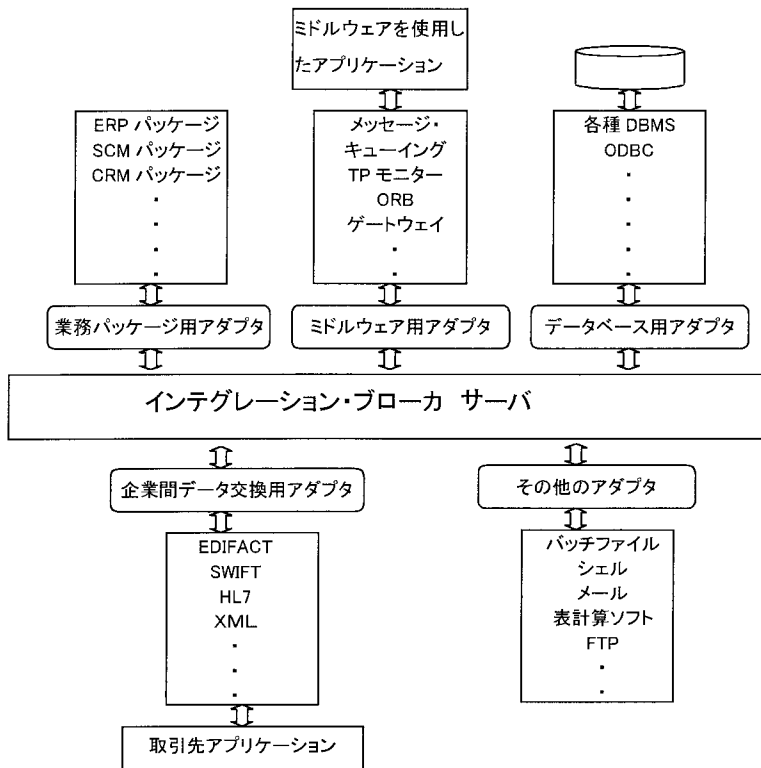


図 8 アダプタの種類

のサービス機能は、「コミュニケーション・サービス」、「アドレス付け配布サービス」と「セキュリティ・サービス」の各サービスを提供する。

次に、サービスの詳細を説明する。

1) コミュニケーション・サービス

アダプタとアプリケーション・システム間のデータを転送することに責任をもつサービスである。

2) アドレス付け配布サービス

アプリケーション・システム間で情報を転送するために、次の三つのアドレス付けスキームをサポートする。

- ① ポイント to ポイント/ブローカー
- ② ブロードキャスト/マルチキャスト
- ③ パブリッシュ/サブスクライブ

①は、同期型の要求応答通信処理で利用される。②と③は、非同期型の通信処理やイベント通知で利用される。

3) セキュリティ・サービス

インテグレーション・ブローカは、安全なアプリケーション・システム統合を実現するために、次のセキュリティ・サービスを提供する。

・認証

ネットワーク上の通信相手が本物であり、途中で通信文が改ざんされていない

ことを保証する。

- ・保証付きメッセージの保全
暗号理論を用い通信を介した取引や文書のやり取りを安全に行う。
- ・権限付与/アクセス制御
デジタル情報に署名情報を付加して、文書の正当性を保証する。

3.1.3 変換サービス機能

変換サービス機能は、アダプタから転送されてきたデータのフォーマットを識別し、連携先アプリケーションの入力データ・フォーマットに変換する。

変換サービス機能は、「同期サービス」「識別/正当性検査サービス」「データ・マッピング」「データフィールドの基本変換サービス」「意味論的データ変換サービス」を提供する。次に、これらの機能について説明する。

1) 同期サービス

複数のアプリケーション・システムからデータを受け取る場合、予約されたデータが全て到着するのを待つ。

2) 識別/正当性検査サービス

入出力データ構造を識別し、ユーザが指定したデータ・フィールドの有効値や変換前後の条件（例えば、あるデータ・フィールド値のエラー判定）など、データの正当性を検証する。

3) データ・マッピング・サービス

出力データ・フィールドと入力データ・フィールドの1対1,1対多,多対1,多対多の関連付けをサポートする。つまり、このサービスでは、入出力データ・フィールド間で分割やマージもサポートする。また、文字の位置合わせ（左詰め to 右詰め,文字列シフト操作）やフィールドの長さ調整もサポートする。

4) データフィールドの基本変換機能

マッピング機能にて対応付けられたデータ・フィールドごとに、

- ・文字セット変換（例えば、EBCDIC to ASCII, EUC to SJIS）
- ・フォーマット変換（10進数 to 2進数, 整数型 to 文字型）

をサポートする。

5) 意味論的データ変換機能

マッピング機能にて対応付けられたデータ・フィールドごとに、

- ・論理演算子（OR, AND, etc）を使用したデータ変換
- ・テーブル・ルックアップを使用したデータ変換

をサポートする

変換サービス機能では、変換パイプライン（図9）操作もサポートする。変換パイプライン操作とは、入力データを中間データにマッピング/変換し（以降、マッピング/変換の組合せを「変換」と省略する）、さらに中間データを出力データに合わせて変換する変換のネスト操作のことである。また、図9で示す通り、1回目の変換では2回目の変換の入力データ（中間データ）の他に、ログ等の出力データを作成（出力データ1）したり、1回目の変換の出力データ（中間データ）とは別の入力データ（入力データ2,2回目の変換に参照するデータ）を参照しながら2回目の変換を行うこと

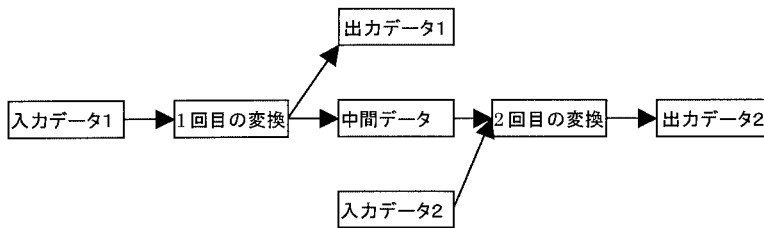


図 9 変換パイプライン

もできる。

3.1.4 ルーティング・サービス機能

ルーティング・サービス機能は、変換された出力データを複数のアプリケーション・システムに渡すための輸送経路選択サービスを提供する。さらに、変換パイプライン内で、生成される複数の出力データに対して、柔軟に輸送経路を選択することができる。

3.1.5 開発機能

本項では、最初にインテグレーション・ブローカによる開発手順について説明し、次により簡単にアプリケーション・システム統合を行うための開発サポート機能について説明する。

インテグレーション・ブローカを使用した場合の開発手順は、

- ① 統合対象のアプリケーション・システムの入出力データ構造を調べる。
- ② 入出力データ構造を、インテグレーション・ブローカが提供する GUI を使い定義する。
- ③ GUI を使用して、定義した入出力構造に対して、マッピング/変換/ルーティングの各指示を行う。
- ④ 通常、マッピング/変換/ルーティングの一連の流れは、アダプターを介して入力を受け取ることで起動される。また、時刻指定やイベント通知などのトリガによって、この一連の動作を実行することもでき、トリガの指定も GUI で行う。

次に、開発サポート機能について説明する。

1) リポジトリ

入出力情報、マッピング/変換情報などの定義情報を格納する機能。定義情報を検索して、再利用できる。

2) データベース・スキーマのインポータ

データベースのスキーマを参照し、データ構造を自動取得する機能。データベースのデータ構造の入力負荷を軽減する。

3) テンプレート・ファイル

業務パッケージの各 API の入出力データ構造を定義したファイル。API の入出力データ構造の入力負荷を軽減する。

4) アダプタ開発キット

提供されていないアダプタを開発するためのライブラリ・セット。新しく作成

するアプリケーションからインテグレーション・ブローカを介して、別のアプリケーションと連携するための C,C++ , Java 用の API を提供する。

3.1.6 運用機能

本項では、インテグレーション・ブローカの運用に関する機能を紹介する。

1) 監視機能

インテグレーション・ブローカの稼働状況を確認するモニター画面を提供する。また、運用管理ツールに対し、障害情報や性能情報を通知する機能も提供する。

2) トポロジ

インテグレーション・ブローカは、ハブ型とバス型のトポロジをサポートする。トポロジにより、信頼性、有効性、拡張性が異なる。図 10 にインテグレーション・ブローカのトポロジの機能配置を示す。

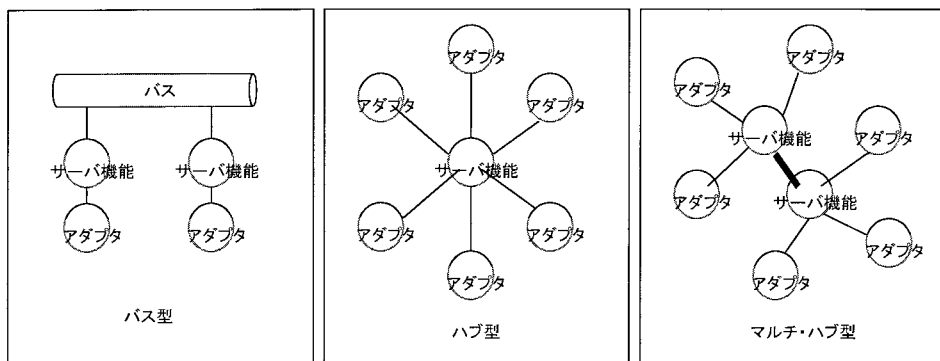


図 10 インテグレーション・ブローカのトポロジ

ハブ型は、アダプタを追加するだけで統合対象アプリケーションを増やすことができるため、最も拡張性が高い。

バス型、マルチ・ハブ型はサーバ機能を分散し、障害時に障害を局所化することができる。障害が発生してもサーバ機能が分散されているため、処理を継続することができ信頼性を高めている。

処理効率は、バス型におけるメッセージの変換がシステム全体に分散されるために効率が良いと言われている。一方、ハブ型の場合もマルチ・ハブ型構成を取ることで、サーバ間で連携することで負荷分散に対応できる仕組みを提供している。

3) メッセージ・ウェアハウス

メッセージ・ウェアハウスは、インテグレーション・ブローカを介してなされたデータの送信記録と変換前後のデータを管理する。メッセージ・ウェアハウスは、受信記録の検索機能と、障害発生時のメッセージの再送機能を提供する。

3.2 ビジネス・プロセス・マネージャの機能

本節では、ビジネス・プロセス・マネージャの機能を説明する(図 11)。

ビジネス・プロセス・マネージャは、一連のビジネス・プロセスの流れをワークフ

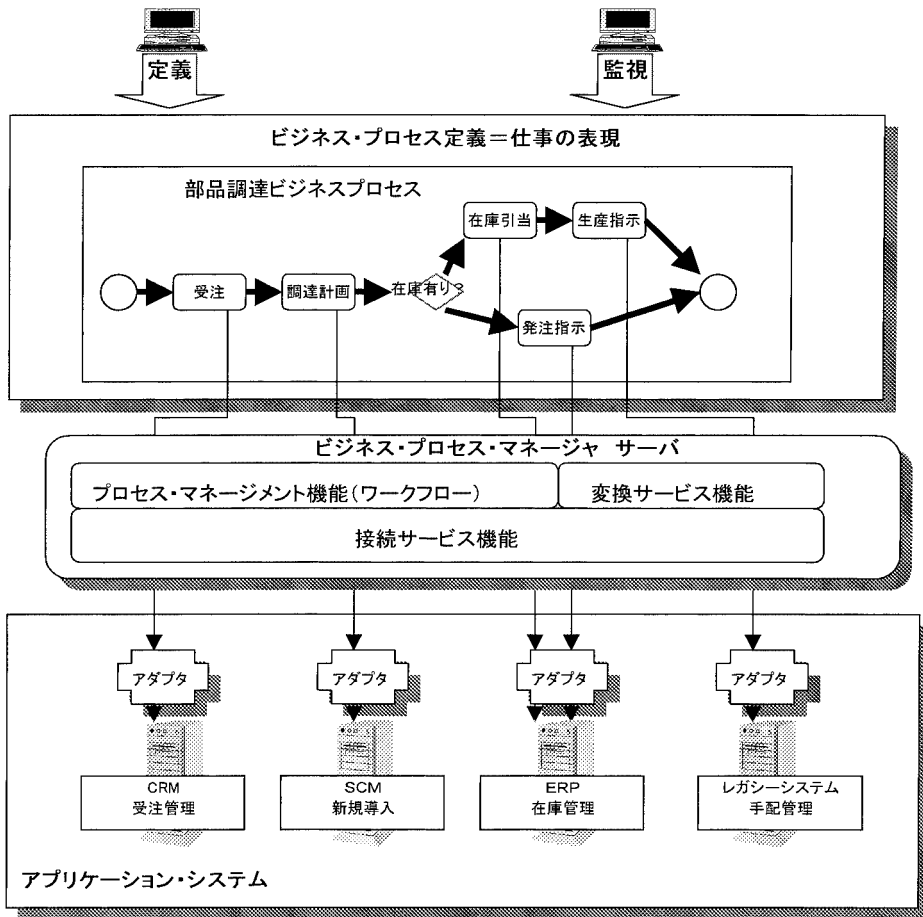


図 11 ビジネス・プロセス・マネージャの機能

ローとして管理することができる。また、プロセス間でやり取りするデータフローも定義することができる。これらの定義に従って、プロセスの起動制御と進捗管理およびデータ変換を行う。

ビジネス・プロセス・マネージャとして必要なワークフローの機能を以下に示す。

- ・プログラミングの必要なしに、複雑なビジネス・プロセスを GUI で定義、管理することができる
- ・プロセスの自動化を統合された環境下で管理することができる
- ・ビジネス・プロセスの変化にも迅速に対応できる
- ・ビジネス・プロセスの再利用ができる

以上のような機能を搭載したプロダクトであれば、ビジネスの変化にも速やかに対応することができる。

ビジネス・プロセス・マネージャにおける、開発手順を以下に示す。

- ① 一連のビジネス・プロセスをワークフローとして、GUI で定義する
- ② 各プロセスを、実際に起動するアプリケーションと対応付ける
- ③ 先行、後続のアプリケーションでやり取りする入出力データ構造を調べる

- ④ 入出力データ構造を GUI で定義する
- ⑤ GUI を使用して、定義した入出力構造に対して、マッピング/変換指示を行う
- ⑥ 起動トリガを定義する

3.3 統合パターンへの対応

ここまで3種類の統合パターンがあることを示し、EAIの機能について記述した。EAIが、3種類の統合パターンにおいて、どのような機能で対応するかを表1にまとめた。

表 1 EAI の統合パターンごとの対応機能

統合パターン	使用するアダプタ	その他の関連機能
データ一貫性型統合	データベース用アダプタ	
複数ステップ型統合	ミドルウェア用アダプタ 業務パッケージ用アダプタ 企業間データ交換用アダプタ	ビジネス・プロセス・マネージャを使用することで、一連のビジネス・プロセスを定義できる。
複数アプリケーション型統合	ミドルウェア用アダプタ 業務パッケージ用アダプタ 企業間データ交換用アダプタ	EAIが提供するAPIを使用することで、容易に連携することができる。

4. EAI の適用分野

本章では、EAIの四つの適用分野について考察する。

- ・意思決定支援システムの構築
- ・企業間電子商取引
- ・業務パッケージとレガシー・システムの連携
- ・Web アプリケーションの構築

4.1 意思決定支援システムの構築

意思決定支援システムは、データウェアハウスを構築して、OLAP(Online Analytical Processing) ツールを利用して売上の傾向や顧客特性などの情報分析を行うために構築される。

一般的には、連携プログラムを作成することで、意思決定支援システムを構築する(図12)。構築の流れは、次の通りである。

① データウェアハウスの構築

基幹系システムや情報系システムのデータベースとERPに代表される業務パッケージから抽出したデータを直接データウェアハウスに挿入する。通常、バッチ処理にて行われる。

② データマート用データベースの構築

データウェアハウスに対して直接検索/分析を行うことができるが、最近では分析/検索の目的に合わせて、マイニング用データベース/ MDB(多次元データベース)/マイニング機能をもつRDBMSを構築する傾向がある。

これらデータベースの構築は、RDBMSが提供するレプリケーション機能が使われる。

③ データ検索/分析

様々なデータ分析機能を使用して、意思決定の支援を行う。

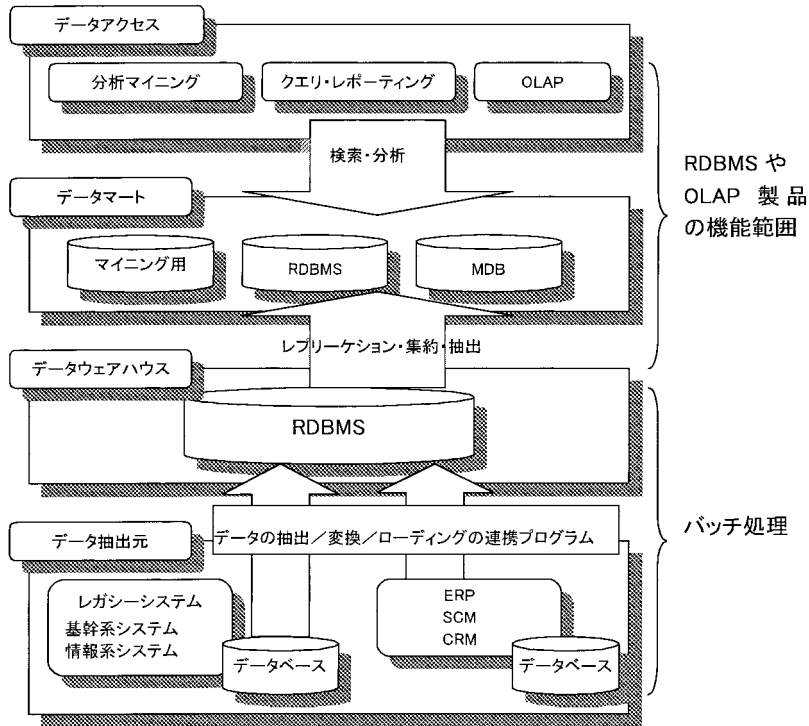


図 12 意思決定支援システムの一般的な構築例

EAI を利用すると、以下の操作を行うだけで、プログラムを作成せずに意思決定支援システム（データウェアハウス）の構築ができる（図 13）。

- ① 使用するアダプタを決める。この場合は、データベース用アダプタと業務パッケージ用アダプタを使用する。
- ② 入出力データ構造を、データベース・インポータと業務パッケージのテンプレート・ファイルを使用して、GUI で定義する。
- ③ マッピング・ルールと変換ルールを GUI で定義する。
- ④ 起動トリガを決める。バッチ起動であれば、起動時刻を設定する。

4.2 企業間電子商取引

一般的な、EDI（電子データ交換）による、企業間の受注/発注データのやり取りは、以下の通りである（図 14）。

- ① レガシー・システムのデータベース、ERP パッケージもしくはデータウェアハウスから取引データの元になる業務データを抽出するバッチプログラムを起動する。
- ② 抽出したデータは、EDI サーバに送られ、そこで EDI メッセージ（シンタックスルールに基づいた取引情報メッセージ）に変換する。この処理は、専用のソフトウェアを用いて行われる。
- ③ EDI サーバは、専用線や VAN などの専用ネットワークを使い、業界ごとの専用プロトコルを使用し企業間でデータのやり取りを行う。

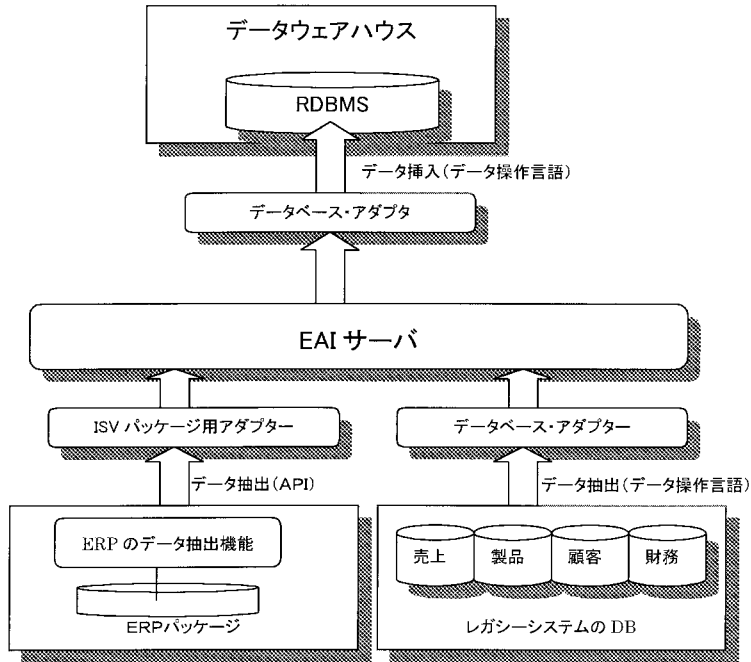


図 13 EAI の適用 意思決定支援システム

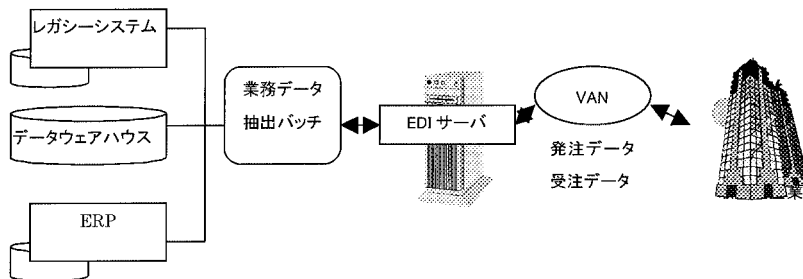


図 14 EDI を利用した企業データ交換

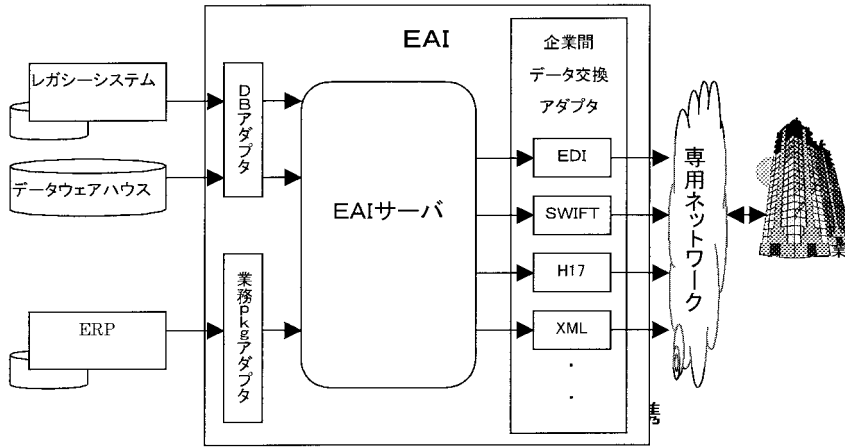
専用プロトコル/専用シンタックスメッセージを交換する業界の標準は ,EDIFACT, FIX , SWIFT , HI7 , ロゼッタネットの XML 利用などがある .

EAI を利用すると , 前節と同様に , プログラムを開発しないで企業間データ交換を行うことができる (図 15) . 異なる点は , 企業間データ交換用アダプタを採用することと , データ交換のテンプレート・ファイルを使用することである .

4.3 業務パッケージとレガシー・システムの連携

業務パッケージとレガシー・システムとのシステム統合要因を , 以下の表 2 に示す .

EAI を使用した業務パッケージとレガシー・システムの統合では , 前節のようにバッチ連携することが一般的に行われる . しかし , 本節ではビジネス・プロセス・マネージャ製品を利用して , 複合ステップ型統合のパターンにより , 業務パッケージとレガシー・システムを統合する例を紹介する (図 16) .



(注) Pkg=パッケージ

図 15 EAI の提供 企業間電子商取引

表 2 業務パッケージの統合要因

パッケージ	概要	統合要因
ERP	企業の財務／人事／製造／流通を統合したスイート製品。	企業のデータ統合を狙った製品であるが、単独で使用される例はほとんど無く、レガシーシステムと情報の統合が必要である。 最近では、「Best Of Bread」といわれるように、複数の ERP 製品のモジュールを企業に合うように組み合わせさせて使用する。この場合、ERP 製品同士の統合も必要となる
SCM	企業の調達・生産・輸送・保管・販売という一連のビジネス・プロセスの最適化を図る製品。	在庫情報、生産情報、設計情報などの情報を、企業内外のデータウェアハウス、ERPとレガシーシステムで統合して使用することが必要。
CRM	顧客の囲い込みを目的とする製品。「One to One マーケティング手法」/SFA(営業支援)/ヘルプデスクなどの機能を提供する。この分野では、データウェアハウスやナレッジ・マネージメントなどの技術も活用される。	顧客情報、製品情報を他のアプリケーションと統合して使用することが必要。
レガシーシステム	トランザクション処理などのミッションクリティカルな分野では、未だ比類なき存在である。業務パッケージとの関連を深めるため、オープンネットワークに対応しつつある。	企業で活用できるデータを多く持ち、業務パッケージと統合する場合、必ず考慮しなければならない。

商品の受発注システムは、以下の複数のアプリケーション・システムで構成される。

- ・受注アプリケーション：CRM パッケージ
- ・在庫管理アプリケーション：ERP パッケージ
- ・商品発送アプリケーション：TP モニターを使用したレガシー・システム

ビジネス・プロセス・マネージャを利用する場合、以下の操作を行うことで、受注から商品発送までの一連のビジネス・プロセスを自動起動することができる。

- ① 一連のビジネス・プロセスを GUI で表現する。
- ② 各プロセスとアプリケーション・プログラムをマッピングする。この際、使用するアダプタも決定する。

- ③ 各アプリケーションの入出力データ構造を定義する．業務パッケージの入出力データ構造は，テンプレート・ファイルを利用することで入力負荷を軽減できる．
- ④ アプリケーション・プログラム間で受け渡すデータフローを定義し，必要な変換指示を行う．

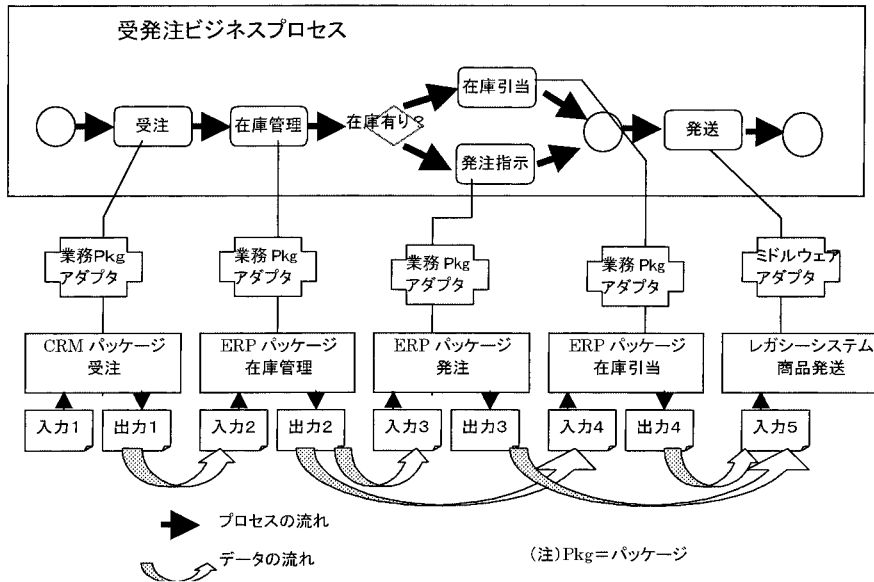


図 16 ビジネス・プロセス・マネージャを使用した受発注ビジネスプロセスの統合

4.4 Web アプリケーションの構築

本節では，Java を使用する Web アプリケーションと既存アプリケーション・システムとを連携する，複合型アプリケーション統合の例を紹介する（図 17）。

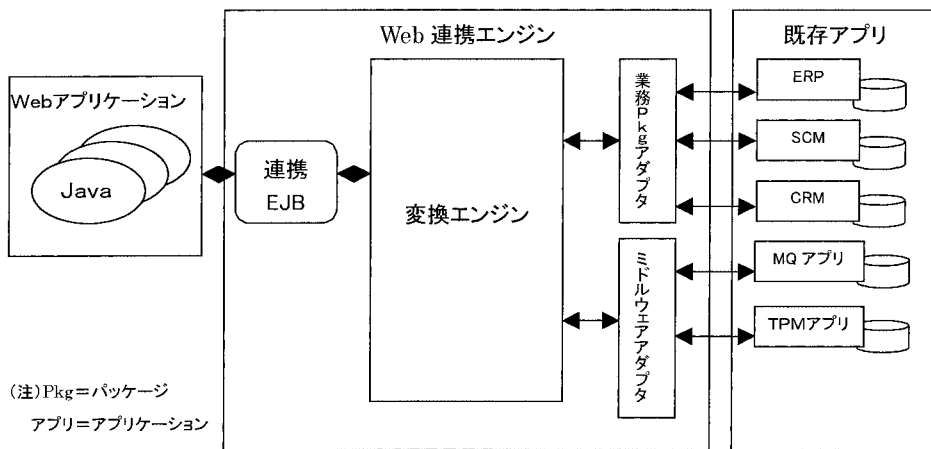


図 17 Web 連携エンジンを使用した Web アプリケーションと既存 AP の統合

EAI は、既存アプリケーション・システムと連携する EJB(Enterprise Java Beans) を生成する。つまり、Java で書かれたアプリケーションから、この EJB を呼び出すことで、さまざまな既存アプリケーションと統合できる。

この連携 EJB を生成する操作は、一つの連携 EJB で、複数のアプリケーションと連携することもできる。

- ① 連携するアプリケーションのアダプタを選択する。
- ② 連携するアプリケーションと連携する Java プログラムの入出力データ構造を定義する。
- ③ 変換およびマッピング・ルールを定義する。
- ④ 連携 EJB 生成指示行う。

5. おわりに

現状では、EAI は ERP パッケージ導入時に適用されるケースが多く、ERP+ と呼ばれることもある。しかし、EAI ベンダーの多くは、過去 EDI システム構築にかかわっていた経験を元に EAI 製品を提供している。このため、最近の EAI 製品は、B2B 対応の機能拡張が盛んで、企業間で同じ EAI 製品を適用している場合、すぐに企業間のビジネス・プロセスを連携できる仕組みを提供している。また、世界中で、アプリケーション統合の需要は大きいため、EAI 製品は、ますます機能拡張していくものと思われる。

当社で取り扱っている EAI 製品に NXDRB, EnConne, Mercator がある。Mercator は、ここまで記述した各種アダプタを持つインテグレーション・ブローカとビジネス・プロセス・マネージャを兼ね備えており、国内外で広く使われている。一方、EnConne は当社のメインフレームである HMP IX シリーズとオープン・システムとの連携、NXDRB は同じく当社のメインフレームである HMP NX シリーズとオープン・システムおよび多様なチャネルとの接続機能の特徴とした製品である。それぞれの製品には、各々の特徴があるため、ユーザの要求に最も適した製品を選定することが重要であると考えられる。

-
- 参考文献** [1] 「メッセージ・ブローカ：アプリケーション統合への集約的アプローチ」：ガートナーレポート JITS: SAR 96 11 1996年11月30日付
- [2] 「ビジネス問題に即した統合テクノロジー」：ガートナーレポート JITS: AIMS 99 96 1999年8月31日付
「従来の役割を拡充しつつ、新たな役割を採り入れる NEONImpact」：ガートナーレポート JECT: JECT 00 06 2000年4月10日付
- [3] “Enterprise Application Integration: Making the Right Connections,” Katy Ring, Neil Ward Dutton, Ovum Ltd. 1999
- [4] 「ミドルウェア：今日のアプリケーションの接着剤」：ガートナーレポート JITS: SAR 99 12 1999年11月30日付

執筆者紹介 小林 雅 浩 (Masahiro Kobayashi)

1986年東京電機大学卒業．同年日本ユニシス(株)入社．
HMP IXシリーズ(UNISYS製メインフレーム)のデータベースの開発/保守に従事．その後，運用管理システムの開発/保守に従事．現在，ネットワークサービス部ネットワークソフトウェア室に所属．