

6S-06 Java,XML を用いた企業間連携のためのフレームワークの実装

石川格, 小川博史, 宇和田弘美
(株) 情報技術コンソーシアム 研究開発部

1. はじめに

現在,インターネットを利用した企業間連携の実現に向けての活発な取組みが行われている.企業間電子商取引や SCM(Supply Chain Management)の構築は究極的に企業間でのアプリケーション統合を目指したものと考えられる.ここでは企業間連携をインターネット環境でのアプリケーション統合と捉えている.さらに統合のための連携フレームワークは企業内でのアプリケーション統合(EAI)へも適用可能である.従来,この統合問題に対し多対多のアプリケーションを個別 API の利用により手作りでつなぐ方法が主流であった.しかし,ヘテロジーニヤスな環境における多数の企業間でのプロセス連携およびデータ交換のためには共通の基盤が必要である.本稿ではインターネット標準を採用した XML ベースの企業間連携フレームワークを Java で実装した成果について報告する.

2. 企業間連携フレームワークのアーキテクチャ

企業間連携を実現する基盤にはコンポーネント技術とインターネット技術が融合したプラットフォームが最も有望と考えられる.このため,データ表現にはプラットフォーム独立な XML を採用した.XML は基本的にテキストベースであり,インターネット標準の通信プロトコルで送受信が可能である.またメタデータであるため,メッセージ以外のプロセスフロー定義,ログ情報などにも利用している.

XML ベースの階層アーキテクチャを図1に示す.

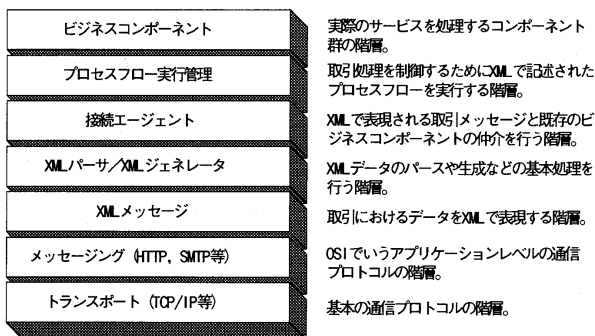


図1.連携フレームワークの階層アーキテクチャ

3. 企業間連携フレームワークの実装環境

企業間連携フレームワークは EJB(Enterprise JavaBeans)を中核とした J2EE をサポートするアプリケーションサーバ上に実装した.EJB は Java のサーバ側コンポーネントモデルである.多数のベンダーから実装環境が提供されており,企業間接続用のハブとして最適と考えた.

この実装環境における連携フレームワークの構成コンポーネントの概要を図2に示す.

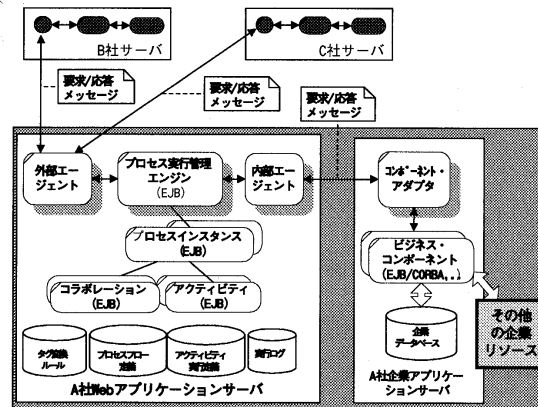


図2. J2EE による連携フレームワークの構成概要

4. プロセスフローの定義および実行

以下に旅行業者が予約申込みを受け付け航空会社に対し空席確認を行うプロセスフローを示す.(XML宣言は省略してある)

```

<CfWorkflow Name="TR" Start="ReserveFlights">
  <Activity Name="SaveFlightResult">
    <Split>
      <To Condition="VACANT"> SendVacantMail </To>
      <To Condition="FULL">SendFullMail</To>
    </Split>
  </Activity>
  <Activity Name="SendVacantMail"/>
  <Activity Name="SendFullMail"/>
  <Transition Name="WaitFlightResult">
    <From>ReserveFlights</From>
    <To>SaveFlightResult</To>
  </Transition>
</CfWorkflow Name="ReserveFlights" Type="Res">
  
```

```

<CName>SearchFlights</CName>
<To>WaitFlightResult</To>
</Collaboration>
</CfWorkflow>

```

プロセスフローは大きく Activity, Collaboration および Transition から構成されている。Activity はプロセスフローを実行している企業のサービス実行に対応し, Collaboration は別企業のサービス (プロセスフロー) 実行に対応する。Transition はプロセスの合流など制御点に対応する。

プロセスフローを XML で外部記述することにより, 処理の実装からフロー制御を分離した。また, この部分は独立したワークフローダイアグラムとして記述できるように考慮してあるため, 変更も容易である。さらに, Activity の実体であるコンポーネントは動的に起動される。

実行時の Collaboration におけるプロセスの連続性は, 委託側プロセスが生成したプロセス ID を受託側プロセスが引継ぐことにより認識する方式とし, プロセス ID によるステータス確認も可能とした。

要求と応答は非同期通信で行われるため, 各プロセスフロー実行エンジンの稼働サーバでは確認用の実行ログをとっている。Collaboration の場合にも受託側プロセスの開始/終了を依頼側サーバに通知する仕組みを実装しており, たとえ例外発生の場合でも実行状況を確認可能とした。

5. 接続エージェントによる XML データの変換処理

受発注などの基幹システムは各企業に既に存在しており, 用いられるデータ項目 (特に用語) は異なっているのが一般的である。共通の商取引用リポジトリが構築された場合でも企業内システムの間でのマッピング問題は残り, この部分を吸収する仕組みをフレームワークに持たせる必要がある。

今回の実装では図3に示す接続エージェント部分にデータ変換機能を持たせている。データ変換には名称変換, 構造変換がまず考えられる。さらに, 国際取引においてはロケールによる日付変換や為替レートによる金額変換等があり得るため, 変換ロジックの追加・変更が柔軟に行える必要がある。このためデータコンバータの Interface を用意し実装を差替え可能な設計とした。XSLT^{*1}を利用した XML データ変換の実装は一例である。XML-Schema^{*2}等のメタデータに基づけば, 多様なデータ処理系との間で意味や制約を考慮した相互変換も実現可能である。

また, 接続エージェント自体も http 以外のプロトコルでのメッセージ送受信も可能なように設計・実

装しており, 特に企業内で有効な MOM(Message-Oriented Middleware)の利用も可能である。

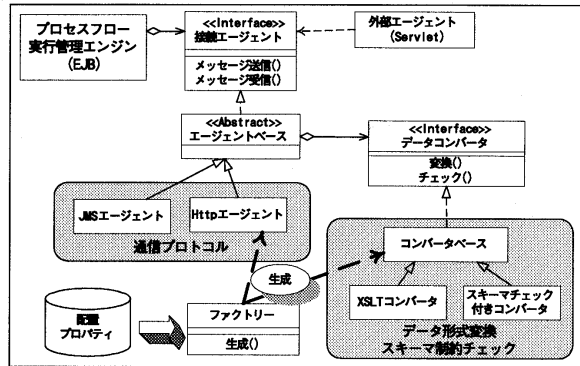


図3. 接続エージェントクラスの概要

差替え可能な実装クラスはプロパティファイルの設定を元にファクトリーにより動的に生成される。

さらに, 企業間商取引の運用に必須条件となるデジタル署名や暗号化などのセキュリティ機能もこの枠組みに追加可能である。

6. おわりに

今回, プロセスフローの管理を含めて企業間連携のためのフレームワークを実装した。J2EE 対応のアプリケーションサーバを利用した実装環境で, 複数サーバ間でのプロセス連携を確認した。また, 接続エージェント部分は各種通信プロトコルへの対応や XML データ変換ロジックの差し替えなどが可能な設計とした。

XSLT プロセッサ, XML-Schema など XML 関連の仕様やツールは急速に充実する方向にある。今回の実装は試行レベルであり, スケーラビリティに対する性能面での検証が必要であるが, XML, Java を利用した統合フレームワークは企業間連携に適した有望な基盤と考えられる。

謝辞

本研究開発は, (株) 情報技術コンソーシアムが情報処理振興事業協会より委託を受け実施したものである。実施に当たって, 新潟工科大学, 青山幹雄教授のご指導および (株) 東洋情報システム, (株) アイネス, (株) 野村総合研究所のご協力を得たことに感謝致します。

*1 <http://www.w3.org/TR/xslt>

*2 <http://www.w3.org/TR/xmlschema-1/>
<http://www.w3.org/TR/xmlschema-2/>