AWS(自律型Webサービス)とそのミドルウェア

大谷 真† 伊東 正起† 塚本 修也† 高木 良輔† 木村 泰輔†
湘南工科大学†

1. はじめに

従来の Web サービスで商取引を行う場合、関連サイトは商取引サービスごとに事前に定められているビジネスプロセスモデル(BPM)に従う必要がある。しかし自由に作られた自律的サイト間でも、自由にビジネスメッセージの自動交換ができることが、次世代の Web サービスでは望ましい。我々はこの基盤技術を自律型 Web サービス(AWS; Autonomous Web Services)と呼び研究を行っている。この論文では、AWS 実現の核である3つの技術について概略と研究状況を述べる。また現在開発中の AWS ミドルウェアについてその構成と主な開発方針に触れる。

2. 自律型Webサービス(AWS)

今や Web サービス(Web Services)は Web サイト間でのアプリケーション接続の基盤技術となり、Web での電子商取引などのビジネスメッセージ交換の重要な手段となりつつある。しかし、BPEL に代表される従来の Web サービスでは、あるサイトがあるビジネスサービスに参加するには、サービスごとに事前に詳細に定められた BPM に従ってそのサイトが作成されていなければならない。独自の BPM を持ち自由に作られたサイトはサービスに参加できない。これでは多くの一般の取引には適用できずまたインターネットの目標である自由なネットワーク世界の実現にも符合しない。AWS は次の点でこの制約を廃すことを狙いとしている。

・各サイトは独自な BPM を持って良い。
・各サイトは独立に運用されて良い。
・集中管理サーバなどの中央機構を持たない。

そのうえで、インターネット内でサイトが遭遇した時点で双方が BPM を相手に合わせて変形させ、それに従って一連のビジネスメッセージ交換を行う。これが AWS の目的である。

3. AWSを実現する技術

AWS を実現するための主要技術は、BPM の変形方法、メッセージング基盤の実現方法、アプリケーションフローの制御方法である。これらに関して概略と研究状況を述べる。

3.1 動的モデル協調(DMH)

DMH の原理[3]を図1に示す。各システムは外部に対する BPM を expose しておく。BPM は (O, B)として定義され、O はそのシステムが持つオペレーションの集合、B は振舞いであり一般に O 上の有限状態機械として定義される。BPM は XML で記述可能であり既存の Web サービスの WSDL の自然な拡張とすることも可能である。

2つのシステムが遭遇した時点で両システムは BPM を交換する。各システムでは、モデル変形アルゴリズム（DMH アルゴリズムと呼ぶ）を実行し、相手 BPM と整合するように自 BPM を変形する（変形した BPM を協調済 BPM と呼ぶ）。変形に成功後は、協調済 BPM に従ってアプリケーションを駆動し一連のメッセージ交換を行う。

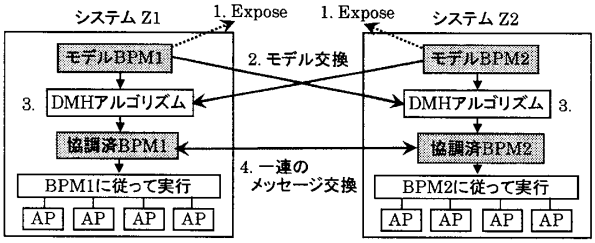

図1：DMH の原理

DMH アルゴリズムについては、BPM を有限オートマトンに限定した範囲で、基本アルゴリズム[1][2]と若干拡張したアルゴリズム[3]が既に示されている。システム間でオペレーションのマッチングを取りながら 2 つの有限状態機械を結合・縮退させる方式をとっている。

3.2 メッセージング基盤

Web 上での非同期メッセージングについては標準化(WS-Messaging[7])が完了しており、ミドルウェア実装製品も登場しつつある。AWS でのメッセージ交換は非同期メッセージングであり、これら既存技術がそのまま適用可能である。一方で先行プロトタイピング[5]を通して以下の考慮が必要であることが判明している[3]。

・非常に長期にわたるセッションの維持

AWS ではシステム間で運用が統一されていないことから、メッセージの送信・受信に極めて長い時間がかかることがあり得る。このため非常に長期にわたるセッション（VL セッションと呼んでいる）の維持が必要である。

・メッセージキューの耐久性

VL セッションの中でメッセージが長期間キューにとどまることが考えられる。DBMS などを用いた耐久性高いキューの実装が必要である。

AWS (Autonomous Web Services) and its Middleware
† Makoto Oya, Masaki Ito, Shuya Tsukamoto, Ryosuke Takagi, Shonan Institute of Technology

3.3 DMH に基づくアプリケーション実行

DMH の結果 BPM は協調済モデルに変形される。これに合わせてアプリケーション(AP)の動作が変化しないといけない。この解決方法として、AP を各オペレーションに対応したプログラム部分（AP セグメントという）の集合として記述し、協調済 BPM をメッセージ交換の進行に合わせて状態遷移させ、それに応じて AP セグメントをイベント駆動型で呼び出すフレームワーク機能が[3]で提案された。図 2 に示すように、メッセージの送受信はアプリケーション内では行わず、AP セグメント実行後（送信の場合）または AP セグメント実行前（受信の場合）にフレームワークが実行する方式である。

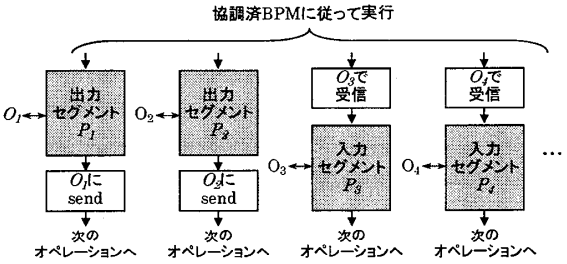

図 2：BPM にもとづくアプリケーション実行

図 3 は AP の例である。メソッド sendEst(), receiveEst(), order()は、BPM 内のオペレーションに対応する AP セグメントであり、Estform, EstResult, OrderForm はそれぞれの入出力メッセージを受け渡すためのユーザクラスである。

```
class SampApp extends AWSFramework {
  public sendEst(EstForm: ef) {
    /* efに送信内容をセットする */
    return;
  }
  public receiveEst(EstResult: er) {
    /* erに入っている受信内容を処理する */
    return;
  }
  public order(OrderForm: or) {
    /* orに送信内容をセットする */
    return;
  }
}
```

図 3：アプリケーションの例

4. AWS ミドルウェアの開発

これまでの研究成果を基に AWS ミドルウェアの研究開発を現在行っている[6]。ここではそのソフトウェア構造と開発の主な方針を述べる。

(1) ソフトウェア構造

図 4 に AWS ミドルウェアの層構造を示す。

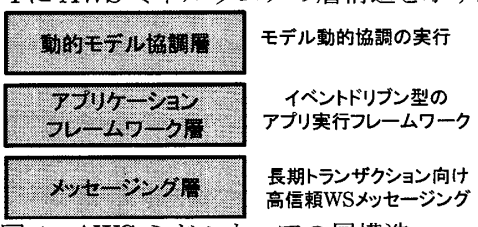

図 4：AWS ミドルウェアの層構造

動的モデル協調層(MH 層)は、DMH アルゴリズムを実行して協調済 BPM を生成し、アプリケーションフレームワーク層(AF 層)に渡す。AF 層は、協調済 BPM の状態遷移に従って、AP セグメントを起動するとともに、適宜メッセージング層(MS 層)を使ってメッセージの送信(send)と受信を行う。MS 層はメッセージングプロトコルエンジン基盤部であり、低レベル API(send, receive)を受付け、Web 環境で store and forward 型の非同期メッセージングを行う。MS 層は VL セッションの維持機能を持つ。

(2) 主な開発方針

★モデル協調層（MH 層）
- モデルはオートマトンの範囲に限定する。
- モデルは XML で記述する。正規表現[1][2]ではなく、実用面から状態遷移関数表現とする。
- DMH は、オートマトンの直積と各状態遷移に対して再帰的に整合性検査を行う方式とする。

★アプリケーションフレームワーク層（AF 層）
- VL セッション全体を AP オブジェクトとする。
- AP セグメントは Java メソッドに対応させる。
- モデルとプログラムの独立性を確保する。

★メッセージング層（MS 層）
- 低レベル API をサポートする。
- DB を用いて VL セッション維持を実装する。
- キューは DB を用いて実装し耐久性を高める。
- 効率良い非同期処理構造とする。

5. まとめ

AWS の究極の目標は独立開発されたサイト間でもインターネット上で自由に商取引が行えることである。その中核技術は、DMH アルゴリズム、BPM による AP 駆動、長期セッションを扱える非同期メッセージングである。AWS は挑戦的なテーマであり ontology など他にも多くの課題があり今後更なる研究が必要である。本研究は科研費(19500095)の助成を受けたものである。

参考文献

1. 大谷, 木下, 嘉数: *自律的 Web サービスにおけるビジネスプロトコルの動的生成について*, 電子情報通信学論文誌, vol.J87-D-I, no.8, pp.824-832, 2004
2. M. Oya and M. Ito, *Dynamic Model Harmonization between Unknown eBusiness Systems*, IFIP I3E2005, Springer ISBN:0-387-28753-1, pp. 389-403, 2005
3. M. Oya, *Autonomous Web Services Based on Dynamic Harmonization*, IFIP I3E2008, Springer, ISBN:978-0-387-8590-2, pp.139-150, 2008
4. 大谷, *モデル動的協調による自律対等型 Web サービスのアーキテクチャ*, 情報処理学会第 70 回全国大会, pp.1-457-458, 2008
5. 伊東, 澤口, 松原, 大谷, *自律型 Web サービス向けの非同期 P2P ミドルウェア*, 情報処理学会第 70 回全国大会, pp.1-565-568, 2008
6. 伊東,塚本,高木,木村,大谷, *AWS ミドルウェアの研究*, 情報処理学会第 71 回全国大会, 2009
7. OASIS, ebXML Messaging Services Ver. 3.0: Part 1. Core Features, OASIS Standard, 2007.