

AWS(自律型 Web サービス)ミドルウェアの研究・アプローチと構成

伊東 正起† 塚本 修也† 高木 良輔† 木村 泰輔† 大谷 真†

湘南工科大学†

1. はじめに

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、事前に定められたビジネスプロセスモデル(BPM)に従ってそのサイトが作成されていることが前提である。独自の BPM を持ち勝手に作られたサイトはサービスに参加できない。この制約を廃し、自由に作られたサイト間でも、柔軟にビジネスメッセージを交換できることが、次世代の Web サービスでは望ましい。この技術を自律型 Web サービス(AWS; Autonomous Web Services)と呼び我々は研究を行っている。この論文では、AWS 実現の中心技術に触れた後、現在開発中の AWS ミドルウェアについて構成と主な開発方針を述べる。

2. AWS を実現する主要技術

(1) 動的モデル協調(DMH)

DMH は図 1 に示すように、インターネット内でシステムが遭遇した時点で BPM を交換し、互いに整合させた上で一連のメッセージ交換を行うことを基本原理としている。整合のためのアルゴリズム(DMH アルゴリズム)が既に提案されている[1][2]。DMH アルゴリズムによって BPM は変形される。変形に成功した場合、変形後の BPM(協調済 BPM という)に従ってアプリケーションを実行し一連のメッセージ交換を行う。

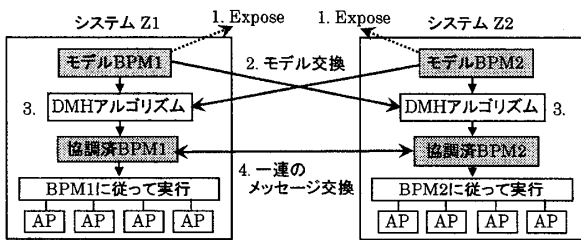


図 1 : DMH の原理

(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が完了しミドルウェア実装製品も登場しつつある。AWS ではこれら既存技術がそのまま適用可能である。一方でプロトタイピング[3][4]により次の考慮の必要性が判明している。

- ・非常に長期にわたるセッションの維持

AWS ではメッセージの送信・受信に極めて長

Study on AWS middleware -Approach and Structure-

† Masaki ito, shuuya Tsukamoto, Ryousuke Takagi, Taisuke Kimura, Makoto Oya, Shonan Institute of Technology

時間を要することがある。非常に長期にわたるセッション (VL セッション) の維持が必要である。

- ・メッセージキューの耐久性

DBMS などを用いた耐久性高いキューの実装が必要である。

(3) DMH に基づくアプリケーション実行

AWS では動的に生成される協調済 BPM 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。この解決方法として、図 2 に示すように、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提案されている[2]。メッセージ送受信は AP 内では行わず、AP セグメント実行後または実行前にミドルウェア側で実行される。

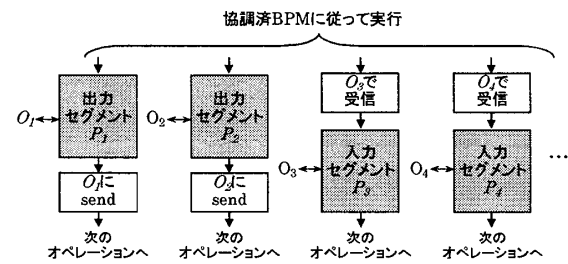


図 2 : BPM にもとづくアプリケーション実行

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]。

4.1 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。

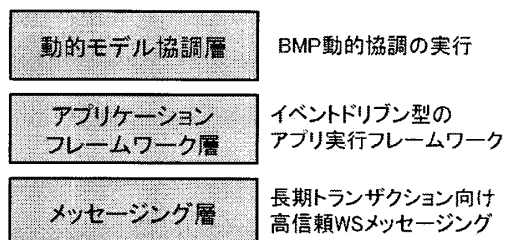


図4: AWS ミドルウェアの層構造

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。

4.2 処理方式概観

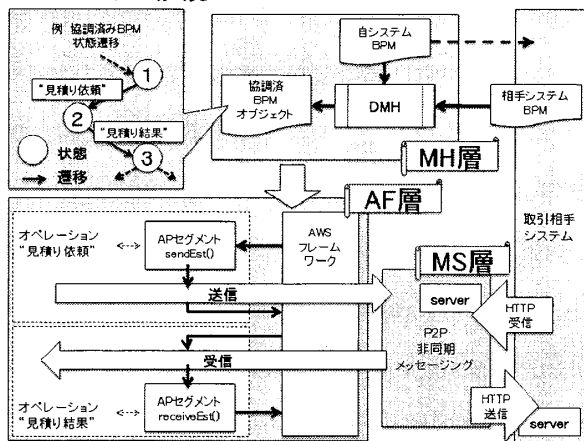


図5 AWS 各層の連携

図5に各層が連携した処理の流れの例を示す。AWSの動作はMH層でBPMの動的協調(DMH)を行うことから開始する。MH層は相手システムとBPMの交換し協調済BPMを生成しそのオブジェクトをAF層に渡す。図5の例では、3つの状態①②③と2つのオペレーション"見積り依頼"と"見積り結果"を含む協調済BPMオブジェクトが渡されている。"見積り依頼"と"見積り結果"に対応するAPセグメントはsendEst()とreceiveEst()であるものとしている。AF層では、このBPMオブジェクトの状態をメッセージ交換の状況に合わせ順次変化させる。その結果、次に実行すべきオペレーションの候補が決定される。これに基づいてオペレーションに対応するAPセグメントを起動する。図5では、状態①から状態②に遷移した時点で"見積り依頼"に対応す

るAPセグメント sendEst()が実行され、状態②から状態③に遷移する時点でAPセグメント receiveEst()が実行される。sendEst()が実行されるとMS層は当該メッセージテキストを含む電文がHTTPを使って送信する。receiveEst()が実行されるとMS層は当該オペレーションに対する電文が受信されるまで待ち状態になる。メッセージはデータベースシステムを使って実装された耐久性が高いキューに蓄積される。APセグメントはJavaのメソッドとして実装されている。1つのVLセッションの中で行われる一連のメッセージ交換は、1つのAPオブジェクトの中で実行され、APセグメント間ではセッションコンテンツを共有できる。

4.3 主な開発方針

★モデル協調層 (MH層)

- ・BPMはオートマトンの範囲に限定する。
- ・BPMはXMLで記述する。[1][2]では正規表現だったが、実用性を考慮して状態遷移関数表現とする。
- ・BPM変換方式は、オートマトンの直積をとり、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)

- ・1つのVLセッション全体を1つのアプリケーションオブジェクトとする。
- ・1つのアプリケーションセグメントは1つのJavaメソッドに対応させる。

★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ミドルウェアの開発言語にはJavaを使用。MS層は本格的な実装を狙い、MH層とAF層はプロトタイプ実装とした。各層それぞれの動作と3層を連携した状態でテストを行い動作することを確認した。しかし、VLセッションのアーカイブが未実装などの課題がまだ残っており更に研究を進めていく予定である。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. M. Oya and M. Ito, *Dynamic Model Harmonization between Unknown eBusiness Systems*, IFIP 13E2005, Springer ISBN:0-387-28753-1, pp. 389-403, 2005
2. M. Oya, *Autonomous Web Services Based on Dynamic Harmonization*, IFIP 13E2008, Springer, ISBN:978-0-387-8590-2, pp.139-150, 2008
3. 伊東, 澤口他, *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
4. 伊東, 澤口他, *自律型 Web サービス向けの非同期 P2P ミドルウェア*, FIT2008
5. 大谷, 伊東, 塚本, 高木, 木村, *AWS とそのミドルウェア*, 情報処理学会第 71 回全国大会, 2009