

AWS ミドルウェアにおける動的モデル協調層の開発

大友 浩照[†] 二宮 良太[†] 大谷 真[†]
 湘南工科大学[†]

1. はじめに

自律型 Web サービス (Autonomous Web Services: AWS) は、Web サービス構築時の選択肢の一つとして研究された全く新しい Web サービス基盤技術である [1]。我々は AWS の機能を実現するミドルウェアの開発を行っている。本論文では AWS の根幹技術である DMH の実装方法を示し、その実装を行いその結果を示す。

2. 自律型 Web サービス (AWS)

通常 Web サービスでは、サービス全体に対し BPM が事前定義され、サービスに参加するシステムはこの BPM に従い構築されている。

AWS では、この事前取り決めを不要とし、Web 上でシステムが遭遇した際、個々のシステムの BPM を動的に協調させる。変形したモデルに従いアプリケーションを駆動させ実際の取引を可能とする (図 1)。こうして自由に開発され独自の BPM を持つシステム間でもより柔軟に商取引を行えるようにすることが AWS 最大の目的である。

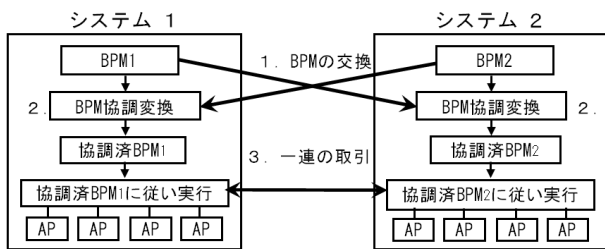


図 1 AWS 動作

3. ビジネスプロセスモデル (BPM)

AWS における BPM は、取引における処理 (オペレーション) の実行順序を定義したものである。BPM = (O, B) と定義され、O は operation (o) の有限集合、 $o = (\text{format}, \text{pattern})$, format: メッセージのフォーマット, pattern: 送受信パターン

(“input” or “output”)。B は O を入力とした非決定性オートマトン。B = (S, I, L, δ)、S は状態の有限集合、I は初期状態 ($\in S$)、L は終了状態 ($\in S$)、 δ は遷移関数 ($O \times S \rightarrow P(S)$ P(x) は x の冪集合)。BPM は XML 文書として記述する。状態遷

移表、正規表現による記述方法と 2 種類の記法が利用できるものとしている。

4. AWS ミドルウェア

AWS ミドルウェアは java により開発が行われており、DMH の機能を提供する DMH 層、BPM に従い APS クラスを実行するフレームワーク層、実際の通信を行うメッセージング層の 3 層から構成されている [3]。APS クラスは BPM の o に対応した java メソッドの実装クラスである。このメソッドを BPM に従い実行し実際の取引を行う。AWS ではこの APS クラスが実際にビジネスプロセスを実行するアプリケーションに相当する。

5. DMH 層

DMH 層が満たすべき要求仕様は以下である。

- A) 取引実行時に必要な情報の取得
- B) DMH アルゴリズムによる BPM の協調変形
- C) 協調済 BPM の提供

5-1. (A) APS 実行時に必要な情報の取得

AWS ミドルウェアを利用して取引を行う際に必要な情報はメッセージング URL、自身の BPM 取引相手の BPM、協調済 BPM、アプリケーション ID (AppId) である。AppId はグローバルユニークな ID であり、メッセージング URL、BPM、APS クラス、Config を一意に特定する (図 2)。AWSD は AppId と関連付けられたメッセージング URL、BPM を定義した XML 文書であり、外部に公開されている。ユーザは APS クラス実行前に、取引相手の AWSD を取得し必要な情報を得る。

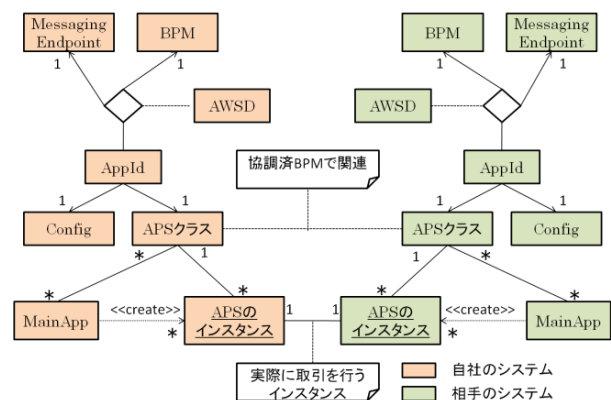


図 2 アプリケーション ID 関係図

Development of a Dynamic Model Harmonization Layer in the AWS Middleware.

[†]Hiroaki Otomo, [†]Ryota Ninomiya, [†]Makoto Oya, Shonan Institute of Technology

5-2. (B) DMHアルゴリズムによるBPMの協調変形

DMHアルゴリズムは既に提案・実装が完了している[1][2][3]。自身のBPMと取引相手のBPMが協調して実行不可能な遷移を除去し、縮退変形するものである。本論文ではこれを協調変形と呼ぶ。協調して実行可能な遷移を求める為に整合性検査関数 `tMatch()` を使用する。`tMatch()` はユーザーが独自に実装するものとし、その実装方式は別途研究がおこなわれている[6]。

5-3. (C) 協調済BPMの提供

協調済BPMは非決定性オートマトンとして動作するBPMオブジェクト(BPMObj)としてアプリケーションに提供する。図4のように実装され、フレームワーク層で利用される。

表1 BPMクラスAPI

メソッド	詳細
<code>init</code>	初期化を行う。
<code>getNextOps</code>	次に実行可能なoの配列を返す。
<code>isFinal</code>	現在の状態が終了状態かどうかを返す。
<code>transit</code>	引数で指定されたoで状態遷移を行う。
<code>getOperations</code>	BPMに定義されたoperation配列を返す。
<code>getString()</code>	BPMのXML表現を返す。

6. 実装

以下にDMH層の動作の流れを示す。

- (1)DMHクラス(表2)のインスタンスPを作る。
- (2)`setMyAWS()`,`setYourAWS()` で互いのAWSをPにセットする。その際AWSの実態がWeb上に公開され`http Get()`で取得可能ならば引数にURLを指定しAWSを取得する。またはローカルに保存されたAWSをFileプロトコルで指定し取得する。DMHクラスはAWSを解析しAWSオブジェクトとしてPに保持する。
- (3)`tMatch()`実装クラスを`setFormatMatcher()`でPにセットする。
- (4)`harmonize()`が実行されたとき、セットしたAWSと`tMatch()`実装クラスを使用しDMHアルゴリズムを適用する。協調変形が成功した場合協調済BPMObjが生成されPに保持される。
- (5) `getHBPM()`が呼ばれたとき協調済BPMObjを返す。協調変形が未実行、又は失敗時にはnullを返す。同じBPMの組み合わせでの協調変形の結果が一定であることが保障される時、協調変形の結果を使い回せることが望ましい。その為BPMObjはコピー可能なように実装されている。またオリジナルのBPMObjに対しての操作を禁止するた

め、`getHBPM()`の戻値は協調済BPMObjのコピーである。

表2 DMHクラスAPI

メソッド	詳細
<code>setMyAWS()</code>	自身のAWSを設定する
<code>setYourAWS()</code>	取引相手のAWSを設定する
<code>setFormatMatcher</code>	tMatch実装クラスを設定する
<code>harmonize</code>	設定したAWSとtMatch実装クラスを使いBPMの協調変形を行う。
<code>getHBPM</code>	協調済BPMObjを取得する
<code>getMyAWS()</code>	自身のAWSオブジェクトを返す
<code>getYourAWS()</code>	相手のAWSオブジェクトを返す

*AWSオブジェクトはAWSの持つ情報を保持している。

7. テスト・評価

[4]のテストスクリプトを用いてDMH層の人工的テストを行った。結果、DMH層は正常に動作し、実装上の問題がないことが確認された。

また、実応用を想定したベンチマークによるAWSミドルウェア全体のテストを行っており[7]、AWSミドルウェアが正しく動作し、実装上の問題がないことも確認できている。

8. まとめ

本論文のとおりDMH層の開発を行いAWSミドルウェアを完成させた。今後は性能の向上のため、BPMの振舞い表現の拡張、DMH理論の改良が課題として考えられる。後者は既に改良点の提案と検討を行っている[5]。

本研究は科研費(21500110)の助成を受けたものである。

参考文献

- [1] M.Oya, and Ito: "Dynamic Model Harmonization between Unknown eBusiness Systems", IFIP I3E, Springer ISBN:0-387-28753-1, pp. 389-403, 2005
- [2] M.Oya.: "Autonomous Web Services Based on Dynamic Harmonization", IFIP I3E, Springer, ISBN:978-0-387-8590-2, pp.139-150, Sep, 2008
- [3] M.Oya et al.: "Middleware for the Autonomous Web Services (AWS)", IFIP I3E, Software Services for e-World, Springer, pp.5-16, Nov., 2010.
- [4]伊東, 大友,他, 動的協調ミドルウェアの実装方法の研究 情報処理学会第72回全国大会, pp. 1-787-788,2010
- [5]大友, 大谷, 自律型Webサービスにおける複数システム間での動的モデル協調, FIT2011 第10回情報科学技術フォーラム pp.165-168,2011.
- [6] 安斎,大友,他,AWSミドルウェアにおける取引様式の互換性決定方式の研究, 情報処理学会第73回全国大会, pp.1-711-712, 2011.
- [7]平本,他 統一的なビジネスプロトコルを前提としないAWS電子商取引の研究, 情報処理学会第74回全国大会