

動的協調ミドルウェアの実装方法の研究

伊東正起[†] 平本真道[†] 大友浩照[†] 大谷真[†]

湘南工科大学[†]

1 概要

インターネットの普及に伴い取引の形態が電子商取引に代表される電子的なやり取りに置き換わりつつある。ほとんどの電子商取引システムは専用のシステムではなく、実装の環境を選ばないことから Web サービスという形で実装される。

しかし、Web サービスで商取引を行う場合、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)や商取引ルールに従う必要がある。そのため現状では、同じルールを適用した Web サービスを利用して取引を行うグループ単位で運用されるにとどまっている。

自由に作られた自律的なサイト間で P2P かつ動的にビジネスメッセージの交換ができることが次世代の Web サービスでは望ましい。

本論文では、この「自由に作られた自律的なサイト間でも、動的にビジネスメッセージの交換ができる」ことを目標とした AWS ミドルウェアの実装方法について言及する。

2 AWS ミドルウェア

AWS ミドルウェアは、独自に定義された BPM を持つシステム間で取引を行う際に、双方の BPM を交換し動的に協調することで、双方のシステムで利用可能な BPM を生成し、その BPM に従って一連の取引を動的に実行するものである。

BPM の違いは AWS ミドルウェアが吸収するため、単一の実装で複数の異なった BPM との取引が可能になる

3 動的モデル協調

この研究の基礎として、BPM の動的協調がある。BPM の動的協調は、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[1]にて、その理論と実装の一部について、定義されている。

BPM 協調の流れは以下の通りである。

- ① 各システムで独自に定義・作成された BPM は、外部に公開される。
- ② 取引を行う際に公開された取引先の BPM を検索し、BPM を交換する。

③ DMH アルゴリズム (BPM を協調するためのアルゴリズム) で BPM の協調を行う。

④ 強調された BPM (強調済み BPM) の内容に従い動作をおこなう。

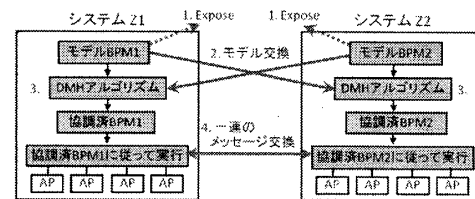


図1 BPMの動的協調

4 要求仕様

AWS ミドルウェアを実装する上で絶対的に必要な仕様として以下の3点があげられる。

1. 動的モデル協調

上記の DHM アルゴリズムを実装が必要。昨年度研究において、実験的に実装を行い、動作を確認した。[2]

2. BPM の内容に従ったイベントドリブンの実行

動的協調により変化する BPM の内容に従って動作を制御することができる必要がある。昨年度研究により、基本的手法としてリフレクションを利用、動作を確認した。[3]

3. 非同期 P2P メッセージング

Reliable Messaging の要件を満たし、なおかつ超長期的なセッションの維持が必要である。2008年度研究において、実験的実装を終了、昨年度研究において、改良、本格的実装を行った。[4]

5 構成・仕様

構成は、大きくモデル協調層、アプリケーションフレームワーク層、メッセージング層の3層に分かれている。ここでは特にアプリケーションフレームワーク層の構成を記す(図2)。

図2の要素の色で塗りつぶされた部分をユーザが作成する。

BPM は、ユーザが作成するセッションアプリケーション(以下 SAP)が提供する機能のインタフェースとなる Operation(以下 OP)の集合、また OP の実行順序(Behavior)を表現する有限オートマトンからなる。OP は、それぞれパターン(Input or Output)と対応するフォーマットの情報を持っている。

Study on implementation of dynamic Harmonized middleware
[†]masaki ito [†]masamiti hiramoto [†]hiroaki otomo [†]makoto oya
[†]Shonan Institute of Technology

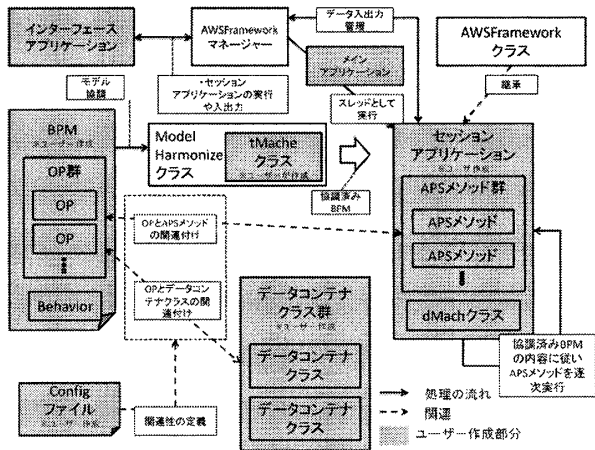


図 2 AWS の構成

Config ファイルは、OP と APS メソッド加えて、データコンテナクラスの関連付けの設定を行うためのファイルである。AWS では、BPM の動的協調を行うことを前提とするため OP 動作順序制御と各 OP の機能実装は祖結合であるために必要なファイルである。

SAP は、AWS が提供するフレームワーククラス (AWSFramework クラス) を継承することで、AWS の前提となる BPM の内容に従った動作を可能にする。SAP のメソッドとして、OP の機能を実装する。加えて APS メソッドでは、BPM では判別不可能な状態遷移の指定を行うことができる。判別不可能な状態とは、送信型のすべての OP、プログラム内のステータスを分岐条件とするものである。

メインアプリケーションは、マネージャからプロセスとして実行される。SAP をスレッドとして実行する。ここでのプログラム次第では、複数の SAP の同期をとり連携させることが可能である。

データコンテナクラスは、APS メソッド内で利用しているデータの形式からメッセージデータ形式への変換またはその逆を行う。フォーマットごとに、別のクラスを実装する必要がある。この処理は、AWS が提供するものではなく、APS メソッドの実装と OP で指定された format を変換するための機能をユーザが実装する。

dMatch クラスは、メッセージのフォーマットの種類を判別するための同名のメソッドをもつ。引数として与えられた、メッセージデータとフォーマットのデータが一致するまたは、整合性があるか判定する。フォーマットの種類から、どの OP に対して送られたメッセージなのかを、フレームワークが自動判別するために利用される。

tMatch クラスは、与えられた 2 つのフォーマットが整合するかを判定する同盟のメソッドを持つ。BPM を協調する際の指針の一つとして DMH アルゴリズムで利用される。

インターフェースアプリケーションは、ユーザが任意に作成する。AWS ミドルウェアからソケットのような使用感の API が提供される。この API は、ハンドラとセレクタを備える。

6 実験と評価

今回の実験では、AWS が BPM の内容に従って動作することが可能であるか、仕様の問題ないかをテストするため、BPM が表現可能な範囲、すなわちオートマトンで表現可能な遷移、分岐、ループなどの要素に加え、送受信パターン、フォーマットなどの要素を含んだ BPM モデルを準備し、その BPM モデルの初期状態から終了状態までに至るすべてのパターンにおいて正常に動作させることが可能であるかをテストした。

結果、すべての課題に対する対応を現在の仕様ですることができた。

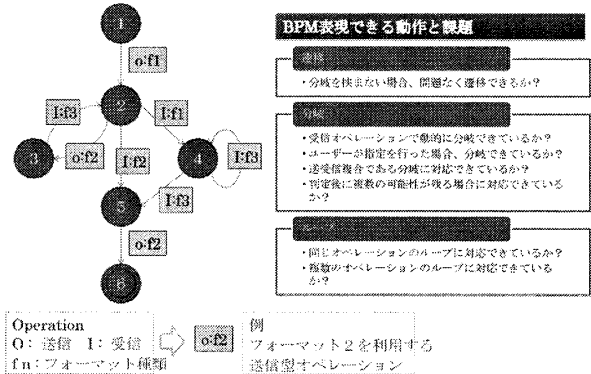


図 3 実験モデル

7 まとめ

BPM に従い動作を正常に行うことが可能であることが確認された。フレームワークとしては、一先ずの完成といえる。今後、マネージャによる管理機能の充実が主な課題になるだろう。本研究は科研費 (21500110) の助成を受けたものである。

8 参考資料

- 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
- 塚本, 高木, 木村, 大谷, AWS ミドルウェアの研究 - 動的モデル協調層-Web サービスのアーキテクチャ, 情報処理学会第 71 回全国大会, pp. 1-511-512, 2009
- 高木, 塚本, 木村, 大谷, AWS ミドルウェアの研究 - アプリケーションフレームワーク層-, 情報処理学会第 71 回全国大会, pp. 1-513-514, 2009
- 木村, 高木, 伊東, 大谷, AWS ミドルウェアの研究 - 自律型メッセージング層-, 情報処理学会第 71 回全国大会, pp. 1-515-516 2009