

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成24年6月25日現在

機関番号：32706

研究種目：基盤研究（C）

研究期間：2009～2011

課題番号：21500110

研究課題名（和文） インターネットにおける自律型 Web サービスの研究

研究課題名（英文） Study on the Autonomous Web Services in the Internet

研究代表者

大谷 真 (OYA MAKOTO)

湘南工科大学・工学部・教授

研究者番号：20360935

研究成果の概要（和文）：

自律型 Web サービス（AWS）は、関連サイトにまたがる全体的なビジネスプロセスモデル（BPM）を前提とせず、各サイトが独自に持つ BPM を動的に協調させることで、自由に作られたサイト間でもビジネスプロセスが実行できることを狙いとしている。本研究では、AWS を実現するための基盤ソフトウェアである AWS ミドルウェアについて、その実装方式に関する各種の課題を解決し、全体プロトタイプを実装・評価した。これを通して AWS ミドルウェアの基本技術が確立できたとともに、AWS の実現可能性、妥当性、有用性が確認できた。

研究成果の概要（英文）：

The AWS (Autonomous Web Services) aims to enable business processes between freely built sites by dynamically harmonizing their BPMs (business process models), without assuming a predefined entire BPM across the related sites. This research is on the AWS middleware, the infrastructure software realizing the AWS. The identified issues on its implementation were solved and the total prototype was developed and evaluated. Through these studies and experiments, the basic technology for the AWS middleware was established, and the feasibility, appropriateness and usefulness of the AWS were confirmed.

交付決定額

（金額単位：円）

	直接経費	間接経費	合計
2009年度	1,000,000	300,000	1,300,000
2010年度	1,200,000	360,000	1,560,000
2011年度	1,100,000	330,000	1,430,000
総計	3,300,000	990,000	4,290,000

研究分野：総合領域

科研費の分科・細目：情報学、メディア情報学・データベース

キーワード：インターネット、Web サービス、自律システム、電子商取引、
ビジネスプロセスモデル

1. 研究開始当初の背景

(1)自律型 Web サービス(AWS)の狙い

インターネットでの電子商取引等のビジネスメッセージ交換において、従来の Web サービスでは（図 1(a)）、関連サイトにまたがる全体的なビジネスプロセスモデル（BPM）が事前定義されていることが必要である。各サイトはそれにそって動作するよう

に作られていなければならない。これに対して自律型 Web サービス（AWS; Autonomous Web Services）は（図 1(b)）、全体的な BPM を前提とせず個々のサイトが独立に BPM を持つものとし、各サイトの BPM を動的に協調させることで、自由に作られたサイト間でも柔軟にビジネスプロセスが実行できることを狙いとしている。

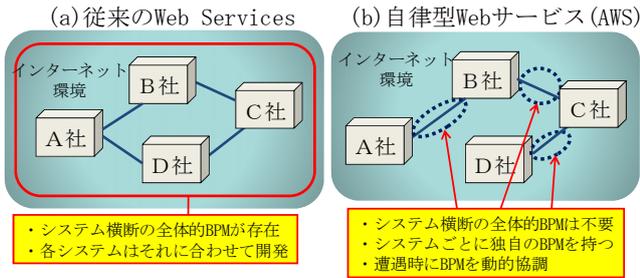


図 1：従来の Web サービスと AWS

(2) 研究開始当初までの研究状況

AWS の核技術である動的モデル協調 (DMH) については、その方式およびアルゴリズムが考案・評価済であった([1]等)。また AWS ミドルウェア技術については 2007~2008 年度の科研費研究(19500095)をとおして、適切なソフトウェア階層構造(図 2)、各層の基本機能、および主な基本方式が明らかになっていた([2]等)。



図 2：AWS ミドルウェアの構成

しかし、AWS ミドルウェアの実装方式については、各層ごとに解決すべき多くの課題があった。また、理論面(DMH)についても BPM 表現やアルゴリズム拡張の課題があった。さらに、実応用を想定した実験の必要性があった。

- [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]Oya, M.: "Autonomous Web Services Based on Dynamic Harmonization", IFIP I3E, Toward Sustainable Society on Ubiquitous Networks, Springer, pp.139-150, September, 2008.

2. 研究の目的

本研究では、AWS ミドルウェアの実装方式についての課題を解決し、AWS ミドルウェアの本格的なプロトタイプ実装を行い、これを通じて AWS ミドルウェアの基本技術を確立させることを主目的とした。

これに加えて、実装方式を検討する中で DMH の改良を提案すること、および、プロトタイプ実装を用いて実応用を想定した評価実験を行うことも目的とした。

3. 研究の方法

ミドルウェア実装方式に関して各層ごとに、具体的な問題点を抽出し、解決方式を考案し、それを AWS ミドルウェアのプロトタイプ

イブとして実装し評価するアプローチをとった。プロトタイプは Java/Tomcat を使って実装したが、部分的なものではなく、AWS ミドルウェアとして必要な全機能を具備し、AWS の実装可能性や妥当性をトータルに評価できるようにした。

抽出された主な問題点とそれぞれに対する解決アプローチを以下に記す。

(1) 動的モデル協調層

① BPM の外部表現の拡張

従来の状態遷移表型に加え、正規表現型の外部表現を追加した。

② BPM 受渡し方式

動的モデル協調層内で自動的に BPM を受け渡す方式を考案した。

③ 動的モデル層全体の実装

BPM の内部表現クラスを定め、動的モデル層全体の制御を 1 つのクラス(DMH クラス)に集中させるアプローチをとり、動的モデル層全体の実装を完了させた。

(2) フレームワーク層

① delegation パターンの適用

従来はアプリケーションが AWS フレームワーククラスを継承していたため、他のスーパークラスを持っていないとの問題があった。これを、動的モデル層の制御クラスがアプリケーションにオペレーション対応の処理を委譲(delegate)する形に変更した。実装にはメタプログラミングを適用した。

② スレッド制御

ビジネスプロセスインスタンス(BP インスタンス)との新たな概念を導入し、BP インスタンスごとにスレッドを割りつける方式にした。これに伴いスレッドプール制御の方式も考案した。またライブアーカイブについても考慮した。

(3) メッセージング層

① Push 型と Pull 型の混在

従来の Push 型メッセージングに加え、中小規模を想定した Pull 型メッセージングも実装した。形態に応じて切り替えて使用できるように工夫した。

② メッセージング層 API

フレームワーク層でライブアーカイブにも対応した BP インスタンスに対応したスレッド制御を行うため、受信待ち時間を限定するなどの新たな API を考案して実装した。

(4) DMH 方式およびその関連

① 3 個以上のシステム間での DMH

従来の DMH 方式および DMH アルゴリズムは 2 個のシステム間での BPM 協調しかできなかったが、新たに 3 個以上のシステム間での DMH 方式を提案し、その間で BPM を動的に協調させるためのアルゴリズムを提案した。

② メッセージ format マッチング

DMH アルゴリズム実行に必要なになる

format マッチングについて初期的方式を提案した。

(5) 実応用を想定した評価実験

DMH の効果は理論的にはすでに検証されていたが、実際の商取引での有効性が実証されていなかった。本研究では、実際の商取引を模擬したアプリケーションを作成し、開発した AWS ミドルウェアを用いて、実際のサーバネットワークに近い環境で実証試験することで、AWS 全体の有効性などを評価した。

4. 研究成果

(1) AWS ミドルウェアの目的と概観

AWS ミドルウェアの目的は、BPM の変形、アプリケーションフローの動的な変更、あるいは HTTP と SOAP を使ったメッセージングプロトコルなどの実装詳細をアプリケーションプログラム (AP) から隠蔽することである。AWS ミドルウェアを使えば、AP 開発者はこれらの詳細を知ることなく、自システムの BPM を記述してビジネスロジックをプログラミングするだけで、AWS を適用したシステムを容易に作成できる。図 3 に AWS ミドルウェアの概観を示す。

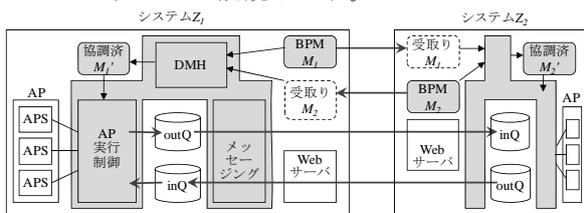


図 3 : AWS ミドルウェアの概観

(2) AWS ミドルウェアの実装

開始当初に考案済であった DMH やモデル駆動型 AP 実行を、実装を踏まえて整理した後、各層の仕様や実装方式を決定し、実際のソフトウェアを開発した。各層の機能と実装の概略を以下に示す。

① 動的モデル協調層

動的モデル協調層は図 4 に示すようにクラス DMH として実装した。

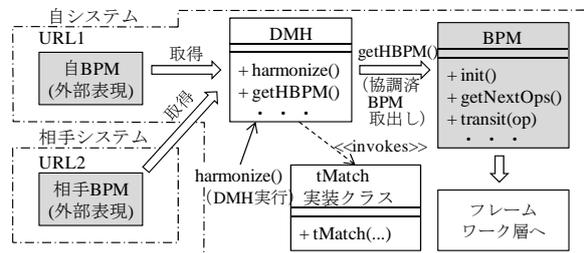


図 4 : 動的モデル協調層の構成

AP(アプリケーションプログラム)は DMH オブジェクトを生成した後、自システムの BPM の URL (図では URL1) と相手システムの URL (図では URL2) を、DMH オブジェクトに連絡しておく。これにより BPM がインターネット上にある場合にもローカルファイルにある場合にも BPM が自動取得さ

れる。AP が harmonize() で DMH の実行を指示すると、DMH アルゴリズムが実行される。動的協調に成功すれば協調済 BPM オブジェクトが作成され、フレームワーク層に渡される。

② フレームワーク層

図 5 にフレームワーク層の構成を示す。利用者が開発するのは Main クラスとアプリケーション本体のクラス (図中の App) だけである。他は AWS ミドルウェアの一部である。これによりアプリケーション開発者はビジネスロジックの開発に専念できるように工夫した。この層の中心は AP の実行を制御する APSController クラスである。App は APSController の委譲先 (delegation) として登録され、これによって APP 内のオメソッドが APSController から実行できる。

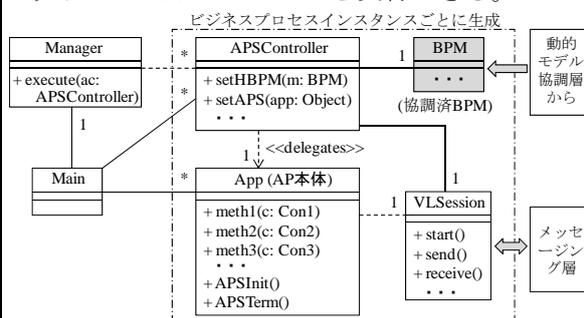


図 5 : フレームワーク層の構成

ビジネスプロセスに従ったアプリケーション間での一連のメッセージ交換処理をビジネスプロセスインスタンス (BP インスタンス) と呼ぶ。おおざっぱに言えば、ビジネスプロセスが取引手順を意味するのに対して、BP インスタンスは毎回の取引を意味する。図 5 の一点鎖線内は、BP インスタンスごとにインスタンス化され、実行スレッドが割当てられ並行実行される。ただし、スレッド数が過多にならないように特別なスレッドプール制御を行うようにした。

③ メッセージング層

メッセージング層のメカニズムは上位層から完全に隠蔽される形で実装した。アプリケーションの端点間での対等型、非同期、store-and-forward かつリライアブルなメッセージ交換機能を提供する。システムの自律性を確保するために、従来のメッセージング基盤 (MQ や JVM など) で一般的な事前定義されたメッセージングチャンネルをシステム間で共有する方式ではなく、ビジネスプロセスを開始する時点で VLSession と呼ばれる長期間持続可能なセッションを動的生成する方式とした。

下位システム構成としては、両システムに Web サーバが存在する構成 (対称型構成) (図 6) と一方にしか Web サーバが存在しない構成 (非対称型構成) (図 7) の 2 つをサポートした。前者は Push 型プロトコルを使用し、

一般の企業間の構成を想定している。後者は Pull 型プロトコルを使用し、一方が小規模オフィスやモバイルなどの構成を想定している。

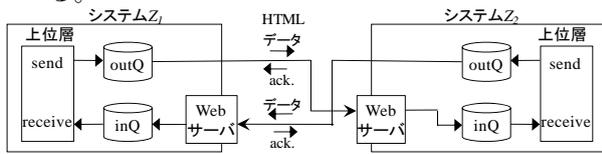


図 6：対称型構成の下位システム

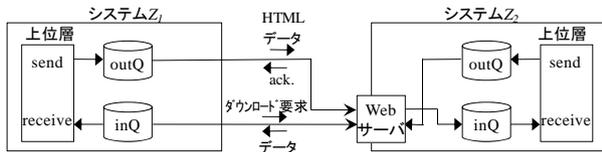


図 7：対称型構成の下位システム

(3)DMH 改良

動的モデル協調について理論面から考察を行い、次の 2 つの提案を行った。

①3 個以上のシステム間での DMH

従来の 2 システム間での DMH を自然に拡張して n システム間での DMH 方式および初期的な DMH アルゴリズムを考案した。2 システムのときと同様に、 n 個の BPM に対して直積をとってパス削除を行ったのち射影をとるアルゴリズムとした。この方式では計算量が状態数の n 乗のオーダーとなる欠点があるので、2 システム間でマッチしたオペレーションにだけ状態遷移を限定し性能の改善を図った。

②メッセージ format マッチングの方式

RDF/OWL により帳票様式間の互換性関係を記述しておき、システムを横断して推論実行する方式を提案した。

(4)実応用を想定した評価実験

実際の商取引を想定して BPM およびアプリケーションプログラムを作成し、今回開発した AWS ミドルウェアを用い、インターネット環境と同等のネットワークで接続したサーバ機 2 台を用いて実験を行った。具体的には、典型的な見積・発注・受注の業務を想定し、購入側と販売側それぞれに 9 種類の BPM を準備し、それぞれのアプリケーションプログラムを用いて、9 個×9 個=81 通りの網羅的テストを行った。この実験をとおして実応用を想定したアプリケーションでも、DMH が妥当かつ効果的に動作することが確認できたとともに、AWS ミドルウェアの基本方式と実装方式が妥当であり、AWS が実用的なソフトウェアとして実装可能であることも検証できた。

(5)成果のまとめ

AWS ミドルウェア実装上の各種の方式を提案し、それを適用して AWS ミドルウェアを開発した。実際的な実験も行った。これを通して AWS ミドルウェアの基本技術が確立できたとともに、AWS の実現可能性、妥当

性、有用性が確認できた。

今後の課題は、本研究でも検討したが、ユビキタスやスマートフォン環境などインターネット以外の環境への応用や従来 Web サービス (SOA) とのハイブリッド化などが挙げられる。

5. 主な発表論文等

〔雑誌論文〕 (計 2 件)

①大友浩照、大谷真：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、第 10 回情報科学技術フォーラム (FIT2011)、査読あり、RO-012、pp.165-168、2011

②Oya, M., Ito, M. and Kimura, T.: "Middleware for the Autonomous Web Services (AWS)", IFIP I3E, Software Services for e-World, Springer, 査読あり, pp.5-16, November, 2010.

〔学会発表〕 (計 12 件)

①大谷真：自律型 Web サービス (AWS) の原理と実装、情報処理学会第 74 回全国大会、pp.1-37-38、2012 年 3 月 7 日、名古屋工業大学

②大友浩照、二宮良太、大谷真：AWS ミドルウェアにおける動的モデル協調層の開発、情報処理学会第 74 回全国大会、pp.1-583-584、2012 年 3 月 6 日、名古屋工業大学

③二宮良太、大友浩照、大谷真：委譲型 AWS フレームワークの実現方法、情報処理学会第 74 回全国大会、pp.1-83-84、2012 年 3 月 7 日、名古屋工業大学

④平本真道、安齋太一郎、大谷真：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、pp.4-775-776、2012 年 3 月 6 日、名古屋工業大学

⑤安齋太一郎、平本真道、大谷真：AWS における電子帳票の互換性決定の実装検討、情報処理学会第 74 回全国大会、pp.2-441-442、2012 年 3 月 6 日、名古屋工業大学

⑥二宮良太、平本真道、安齋太一郎、大谷真：AWS (自律型 WEB サービス) ミドルウェアフレームワーク制御、情報処理学会第 73 回全国大会、pp.1-707-708、2011 年 3 月 3 日、東京工業大学

⑦木村泰輔、二宮良太、平本真道、大谷真：自律型 Web サービス メッセージング基盤の開発、情報処理学会第 73 回全国大会、pp.1-709-710、2011 年 3 月 3 日、東京工業大学

⑧大谷真：自律型 Web サービスの研究状況と課題、情報処理学会第 72 回全国大会、pp.3-101-102、2010 年 3 月 11 日、東京大

学

- ⑨伊東正起,平本真道,大友浩照,大谷真: 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 年 3 月 11 日、東京大学
- ⑩大友浩照,伊東正起,吉川恭平,大谷真: AWS におけるビジネスプロセスモデル、情報処理学会第 72 回全国大会、pp.1-789-790、2010 年 3 月 11 日、東京大学
- ⑪吉川恭平,木村泰輔,平本真道,大谷真: AWS における非対称構成型メッセージング機能の実現、情報処理学会第 72 回全国大会、pp.1-795-796、2010 年 3 月 11 日、東京大学
- ⑫木村泰輔,吉川恭平,伊東正起,大谷真: AWS メッセージング基盤改良の検討、情報処理学会第 72 回全国大会、pp.1-793-794、2010 年 3 月 11 日、東京大学

6. 研究組織

(1) 研究代表者

大谷 真 (OYA MAKOTO)
湘南工科大学・工学部・教授
研究者番号: 20360935

(2) 研究分担者

なし

(3) 連携研究者

坂下 善彦 (SAKASHITA YOSHIHIKO)
湘南工科大学・工学部・教授
研究者番号: 50339793

奥野 拓 (OKUNO TAKU)
公立はこだて未来大学
・システム情報科学科・准教授
研究者番号: 30360936