

平成21年6月19日現在

研究種目：基盤研究（C）

研究期間：2007～2008

課題番号：19500095

研究課題名（和文） インターネットにおけるビジネスプロトコルモデルの動的協調

研究課題名（英文） Dynamic Harmonization of Business Protocol Models 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：

従来の Web Services を用いたインターネット電子取引では、取引のビジネスプロセスモデル（BPM）を事前に定めておき、取引参加システムすべてがこれに完全に従う必要がある。このため、自由に作られたシステムは取引に参加できないとの大きな問題がある。これに対し、システムが遭遇した時点で BPM を動的に協調させることでこの問題を解決できる。本研究では、このビジネスプロトコル動的協調に関して、ミドルウェア制御方式やメッセージング方式などの実現上の基本課題を解決し、実験実装開発をとおして妥当性を確認した。

交付額

（金額単位：円）

	直接経費	間接経費	合計
2007年度	1,200,000	360,000	1,560,000
2008年度	1,600,000	0	1,600,000
年度			
年度			
年度			
総計	2,800,000	360,000	3,160,000

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、モデル化、Web Services、自律システム、電子商取引

1. 研究開始当初の背景

Web Services はインターネットでの電子取引などのビジネスメッセージ交換の基盤技術となった。しかし従来の Web services では、取引に参加するすべてのシステムを横断したビジネスプロセスモデル（BPM）を前もって定め、BPEL などを用いて詳しく定義してお

く必要がある。そのうえで、取引に参加するシステムは、取引全体の BPM に厳密に従って動作するように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システムは取引に参加できない。

これでは、企業/組織内での取引、企業/組織グループ内での取引、eMarket プレースのよ

うな特定契約に基づくビジネスシステム間での取引、強く連携した業界団体内での取引などは良いとしても、中小企業・個人を含む柔軟性を求められる一般取引には適用できない。システムを横断した取引全体の BPM の事前定義を前提とせず、個々のシステムが独自に持つ BPM を取引実行時に整合化させ、未知のシステム間でも best effort で取引を可能とする技術が将来的に望まれる。

この解決を目的にしているのが、研究開発当初時に既に代表研究者が発表していた動的モデル協調 (DMH) [1][2]である。動的モデル協調では、図 1 に示すように、(1)各システムは各々のビジネスプロセスモデル (BPM) を外部に expose しておく、(2)システムが遭遇した時点で互いの BPM を交換する。(3)相手の BPM に整合するように自 BPM を変形する (協調させる) (DMH アルゴリズム)、(4)変形した BPM (協調済 BPM) に従って一連のメッセージ交換を実行する。

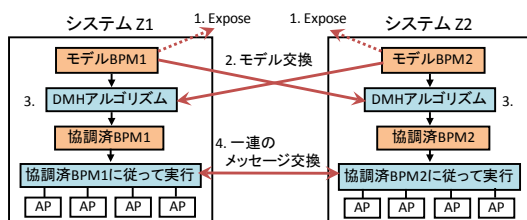


図 1 : 動的モデル協調

[1][2]は上記の動的モデル協調の理論的なモデル化とアルゴリズムを提示している。しかし、研究開始当初の時点では、これを実際のインターネットでどのように実装するか、またインターネットミドルウェア技術にどのように適用できるかは研究されていなかった。

2. 研究の目的

本研究の目的は、先行研究結果である動的モデル協調およびそのアルゴリズムのインターネットミドルウェアへの適用方式を明らかにすることである。特に、有限状態機械として表現される BPM の振舞いに対応して、

アプリケーションの実行フローをどのように動的に変化させるか、および、システムの自律性を考慮に入れてメッセージングプロトコルエンジン部分をどう実装するかを中心に研究し、プロトタイプ実装を行い方式の妥当性を検証することを目的とした。また、これをとおして、ミドルウェア実装上の各種課題の抽出、動的モデル協調方式の改良など将来課題の抽出も目的とした。

[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] 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. 研究の方法

本研究での研究開発対象である動的モデル協調を用いたミドルウェア (本研究の中で「AWS ミドルウェア」と名付けた; 注: AWS: Autonomous Web Services、自律型 Web サービス) は、

- ① Web Services を用いてシステム間で非同同期対等型でメッセージングを行う部分と、
- ② その上位で、動的モデル協調を実行しその結果に従ってアプリケーションを制御する部分

に分かれる。①は既存のメッセージング技術をこの分野に適用することが主な研究内容であり、②は①が前提だが、新たな概念に基づく実装研究が必要である。

この点から、初年度 (2007 年) は、まず①についての PHP を用いた初期のプロトタイプ実装を作成し、実現可能性を評価し基本課題抽出を抽出した。②については基本方式の検討を行った。2 年目 (2008 年) は、①については、2007 年度の結果をもとに Java を用いた本格的なプロトタイプを開発し、メッセージングに関する実験と検討を行うと

ともに、②開発のプラットフォームとした。②は前年度に考案した方式をもとに実際に Java プロトタイプを開発し、①の上でその動作を確認し方式の妥当性を検証した。またモデル協調アルゴリズムもこれに即して新たに実装した。

2年目に関しては、他に、AWS ミドルウェアのソフトウェア構造の検討、および、今後の課題抽出も行った。

4. 研究成果

本研究の主要成果は以下とおりである。

- ①AWS ミドルウェアのソフトウェア階層構造を決定した。
- ②各層（メッセージング層、フレームワーク層、モデル協調層）について、基本的な機能仕様、インタフェース仕様、制御など基本方式を明らかにした。
- ③上記に従って、各層のプロトタイプを開発し、実験評価をとおして方式の妥当性を確認した。
- ④各層および AWS ミドルウェア全体に関して、派生課題を含め今後の課題を抽出した。

(1)ソフトウェア階層構造

図2に AWS ミドルウェアのソフトウェア階層構造を示す。

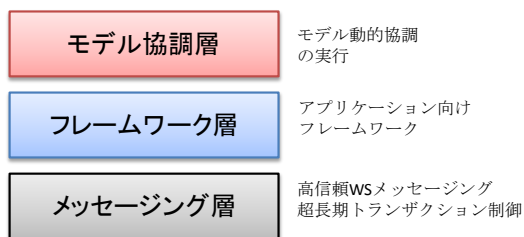


図2：AWS ミドルウェアの層構造

モデル協調層は、DMH アルゴリズムを実行して協調済 BPM を生成し、フレームワーク層に渡す。フレームワーク層は、協調済 BPM の状態遷移に従って、AP セグメント(後述) を起動するとともに、適宜メッセン

グ層を使ってメッセージの送信と受信を行う。メッセージング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。また、メッセージング層は VLセッション(後述) の維持機能を持つ。

(2)メッセージング層

Web 上での非同期メッセージングについては標準化(例えば ebXML Messaging)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、本研究の中で以下の考慮が必要であることが判明した。

- ・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。

本研究では、上記に基づき、以下を方針としてメッセージング層のプロトタイプ開発を行った。なお、他の層も含めて開発言語は Java とした。

- ・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 DBMS を使って実装し、長期セッション期間での耐久性を確保する。
- ・効率良い非同期処理構造とする。

(3)フレームワーク層

DMHの結果、BPMは協調済BPM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を考案した。図3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。

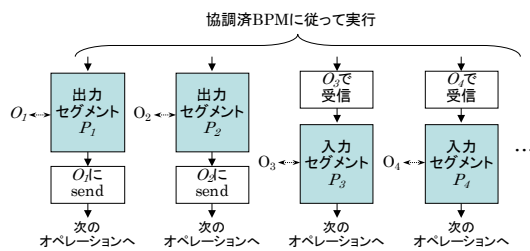


図3 : BPMにもとづくアプリケーション実行

図4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4 : アプリケーションの例

フレームワーク層のプロトタイプ開発では以下の方針とした。

- VLセッション全体を1つのAPオブジェク

トとする。

- 各APセグメントは1つの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。

(4)モデル協調層

モデル協調層では、BPMの表現方法、アルゴリズム実装方式、および協調済BPMを下位のフレームワーク層に渡す方法が、主要な課題であった。

BPMについては、オートマトン表現とし、実際の表記はXMLを用いたシンタックスを新たに定義した。アルゴリズムは既発表のアルゴリズムをオートマトン変形アルゴリズムに変更して実装した。同じ結果が出力されることはプロトタイプをとおして確認できた。フレームワーク層には、BPMのJavaオブジェクトを渡す方式とした。

そのうえで、以下の方針に基づいてモデル協調層のプロトタイプを開発した。

- 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。

(5)プロトタイプ結果と抽出された課題

プロトタイプ開発により本研究で考案した方式の実現可能性と基本的な妥当性が確認できた。一方、次の点について今後更に研究が必要であることが分かった。

①AWSミドルウェアの実装方式の研究

- 各層ごとに明らかになった課題の解決
- 超長期トランザクションの制御とライブアーカイブ機能の実現方式
- フレームワーク層でのメタプログラミング方式

- ・ 3層を横断したミドルウェア実装方式
- ②DMHの改良
 - ・ モデルの振舞い部分の記述能力の強化
(有限オートマトンを超える能力の追加)
 - ・ モデルの外部表現の改良
 - ・ オペレーションマッピング制御および
メッセージ format マッチング制御
- ③事前合意型とのハイブリッド化
- ④インターネット商取引以外の分野への
応用

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 1件)

①Makoto Oya, Autonomous Web Services Based on Dynamic Harmonization, IFIP I3E, Toward Sustainable Society on Ubiquitous Networks, Springer, ISBN:978-0-387-8590-2, pp. 139-150, 2008年、査読あり

[学会発表] (計 6件)

- ①大谷真、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年3月13日、筑波大学(大会優秀賞受賞)
- ②大谷真、伊東正起、塚本修也、高木良輔、木村泰輔、AWS(自律型Webサービス)とそのミドルウェア、情報処理学会第71回全国大会、pp.1-503-504、2009年3月11日、立命館大学
- ③伊東正起、塚本修也、高木良輔、木村泰輔、大谷真、AWSミドルウェアの研究 -アプローチと構成-、情報処理学会第71回全国大会、pp.1-509-510、2009年3月11日、立命館大学
- ④塚本修也、高木良輔、木村泰輔、大谷真、AWSミドルウェアの研究 -動的モデル協調層-、情報処理学会第71回全国大会、pp.1-511-512、2009年3月11日、立命館大学
- ⑤高木良輔、塚本修也、木村泰輔、大谷真、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年3月11日、立命館大学
- ⑥木村泰輔、高木良輔、塚本修也、大谷真、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、p

p.1-515-516、2009年3月11日、立命館大学

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

(2007年度のみ)

坂下 善彦 (SAKASHITA YOSHIHIKO)

湘南工科大学・工学部・教授

研究者番号：50339793

(2007年度のみ)

奥野 拓 (OKUNO TAKU)

公立ほこだて未来大学

・システム情報科学科・准教授

研究者番号：30360936

(3)連携研究者

(2008年度のみ)

坂下 善彦 (SAKASHITA YOSHIHIKO)

湘南工科大学・工学部・教授

研究者番号：50339793

(2008年度のみ)

奥野 拓 (OKUNO TAKU)

公立ほこだて未来大学

・システム情報科学科・准教授

研究者番号：30360936