

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

大谷 真†

湘南工科大学†

1. はじめに

モデル動的協調[1]をベースとした自律対等型 Web サービスミドルウェアについて、課題と現在研究中のアーキテクチャ概略を述べる。

2. 背景と問題点

Web Services は今や Web 上のサイト内のアプリケーション間接続の基盤技術となり、サイト間で自由な電子取引やコミュニケーションが可能になることが期待されている。しかし、ステークホルダであるサイト群の全体的管理/制御または事前取決め事項なしに、自発的に作られたサイト間で自由に自動商取引などができるためには、現在の RPC を基調にした Web Services 技術だけでは不十分である。Web 環境において次の3点に関し技術的解決を図る必要がある。

- ・自律対等型メッセージング技術
- ・長期トランザクション技術
- ・ビジネスプロトコルの動的協調

Web 上でのメッセージングプロトコルについては ebXML Messaging [2]や WS-Reliability [3]で技術確立されている。しかし本格的な対等型ミドルウェアは登場していない。更にメッセージング実行のためのポリシー(タイムアウトや再送方法など)を事前取決めなくかつ一方だけが主導権を持つことなく動的決定する方式を含め、十分に解決されていない課題が残っている。

長期トランザクションについては、BPEL や BPMN によって、ACID でないが比較的短期なビジネスランザクションを記述/制御する技術は確立しているが、寿命が数時間ではなく、数日、数週間、場合によっては数カ月・1年かかる長いトランザクションに関しては、ステークホルダー間の密な管理を伴うワークフロー技術以外には十分な技術確立がなされていない。

ビジネスプロトコルモデルの動的協調とは、インターネット内に点在するビジネスシステム間で動的にビジネスプロトコルモデルを整合させることである。その基本方式は大谷[1]により提案されている。異なる企業・組織・個人間での取引の多くでは、共通のプロトコルモデルを事前合意するのは、企業/組織グループ内など限定された応用分野を除いて、一般に困難である。

提案アルゴリズムはビジネスプロトコルが事前定義不能であることを前提とした上で、2つのシステムが遭遇した時点でビジネスプロトコルの動的な整合を行うことを特徴としている。一方、本アルゴリズムは外部化されたビジネスプロトコルモデルの整合化であり、各システム内のミドルウェアとの連携方法は未解決である。本論文ではモデル動的協調を基軸としてこれら3点の解決を図ったミドルウェアの構想を述べる。

3. ビジネスプロセスの動的協調

[1]のアプローチは以下のとおりである。各システムはそれが期待するビジネスプロセスモデル BPM を外部に expose する。 $BPM = (O, B)$ と表現され、 O はオペレーション op の集合で、 $op = (p, t)$ である。 p はメッセージパターンと呼ばれそのオペレーションが受信か送信かなどを示し、 t は入力または出力するメッセージの型を示す。なお O は WSDL[4]の interface に相当する。 B は O 内のオペレーション実行の順序を有限状態機械として表現したものである。[1]では特に非決定性オートマトンの範囲に限定させている。図1に示すように、①Web 内の2つのシステムが遭遇した時点でそれぞれの BPM を交換する、②アルゴリズムを実行して両システムがビジネスプロトコルを結合可能か判定し、可能なら可能な範囲に BPM を縮退する。このとき双方の O 要素がどう対応するかも決定される。③その後一連のメッセージ交換を開始する。

アルゴリズムは非決定性オートマトンの変形、及びセマンティック Web 技術を適用したタイプマッチングによって構成されている。

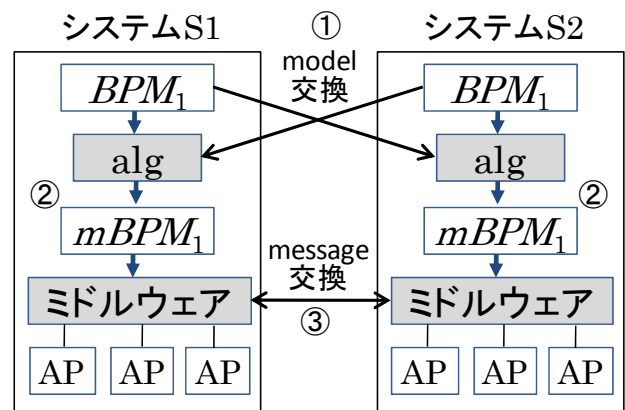


図1：ビジネスプロセスの動的協調

Autonomous P2P Web Services based on Dynamic Model Harmonization

† Makoto Oya, Shonan Institute of Technology

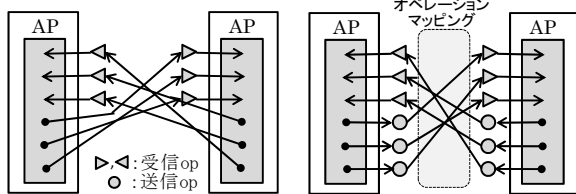
4. 自律対等型ミドルウェア

この実現のため現在研究を行っている自律対等型ミドルウェアに関して、解決すべき主要課題と基盤部分のアーキテクチャ概略を述べる。

4.1 課題

(1) AP とオペレーション接続の分離

クライアントサーバ型や従来のイベントチャネル型のメッセージ通信では一般に受信オペレーションだけが expose されており送信側の AP(アプリケーション)内から直接相手のオペレーションをアクセスする。双方向通信では図 2 (a)のように双方の AP が密に結合してしまう。これでは各システムの十分な自律性は確保できず、プロトコル動的協調での遭遇時動的 OP マッピングにも対応できない。このため図 2 (b)のように送信オペレーションも expose し、2システム間のオペレーションマッピングを、両システム内の AP 処理から独立化させる必要がある。AP は自システムのオペレーションだけをアクセスし、相手システムのオペレーションとの結合は AP の外側で決定できるようにする。



(a) APから直接受信opをアクセス (b) APとopを分離

図 2 : AP とオペレーションの分離

(2) 振舞 B の変更に従った AP 制御

BPM 内の振舞 B は相手システムによって変化する。AP での実際の処理の流れもこれに合わせて変化しなければならない。AP 内の処理を op ごとにセグメント化し流れを動的に変更する機能が要である。AP 内の処理セグメントをイベント駆動型で実行する方式が有力な解決策である。

(3) 長期トランザクション

一連のメッセージ交換を 1 つの長期トランザクションとして管理する必要がある。トランザクション実行中に一方または両方のシステムが停止し再開した場合であっても、このトランザクションは維持されねばならない。更に、第 3 の管理サーバを必須とせず、かつ 1 システムが一方的に主導権を握ることのない、対等型かつ疎結合の案件管理が必要である。長期にわたるセッションを動的に合意し、不揮発性コンテキストを維持するメカニズムが必要である。

(4) 自律対等型のメッセージング

プロトコル等は既に確立済み技術の延長で実現できるが、ポリシーの事前ネゴシエーション

と動的変更が必要である。

4.2 アーキテクチャ

図 3 にこの自律対等型ミドルウェアのオブジェクトのアーキテクチャを示す。構成要素は次のものである。ua (user agent) : メッセージ交換の主体、メールアドレスと同様にユニークな名前を持つ。lp (long process) : 一連のメッセージ交換を行うプロセス、通常のプロセスと異なり不揮発性でシステム停止/再開後も継続する。ls (long session) : lp 間の抽象的な通信路、双方の lp 間で会話を開始するときに生成され長期トランザクション完了まで維持される、lp と同様不揮発性である。BPM : 3 で述べたビジネスプロセスモデル、ls を作る前に動的に協調され lp ごとに変形される。mB : 変形後の BPM を示す。

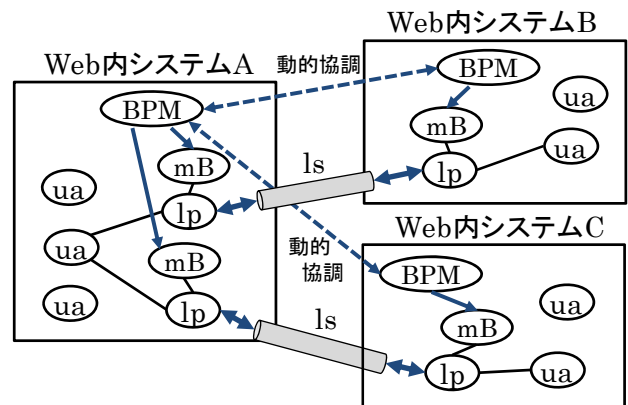


図 3 : オブジェクトモデル

ソフトウェア概略構成は図 4 のとおり。ネゴシエーション制御はモデル動的協調とポリシーのネゴシエーションを行い、対等型メッセージングは reliable なメッセージングプロトコル実行を司る。

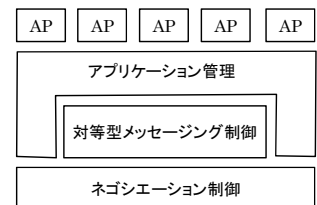


図 4 : ソフトウェア構成

5. まとめ

本論文では自律対等型 Web サービスミドルウェアの基本原則、課題、及びアーキテクチャ概略を述べた。これに従って現在対等型メッセージング制御のプロトタイプを開発中である [5]。今後全体のプロトタイプ開発を行う予定である。

参考文献

1. M.Oya and M.Ito, *Dynamic Model Harmonization between Unknown eBusiness Systems*, 5th IFIP conf. for I3E, Springer, pp.389-403, 2005
2. OASIS, *ebXML Messaging Services ver. 3*, OASIS Standard, 2007
3. OASIS, *WS-Reliability 1.1*, OASIS Standard, 2004
4. W3C, *WSDL ver. 2.0, W3C Recommendation*, 2007
5. 伊東, 澤口他, *非同期 P2P 通信ミドルウェア*, 情報処理学会第 70 回全国大会, 2008