

# AWS ミドルウェア:動的モデル協調の改良

大友 浩照<sup>†</sup> 安齋 太一郎<sup>†</sup> 木村 泰輔<sup>†</sup> 大谷 真<sup>†</sup>  
 湘南工科大学<sup>†</sup>

## 1. はじめに

自律型 Web サービス(AWS)では、動的モデル協調(DMH)を用いて自身のビジネスプロセスモデル(BPM)を相手の BPM に合わせ取引可能な形に協調変換し、それに従い取引を実行する。これにより異なる BPM を持つシステム間での取引を可能とする。AWS の根幹技術である DMH アルゴリズムの改良は、AWS の性能改善において最も重要なファクターの一つである。本論文では、より自由で柔軟な Web サービスの構築を実現するために、複数の BPM を用いた DMH アルゴリズムを提案し DMH アルゴリズムの改良を行った。

## 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、各システムが独自の BPM を持ち、システムを開発・運用できる。取引を行う際、互いの BPM を交換し協調変換することで、協調済 BPM を生成し、それに従いアプリケーションを実行し一連の取引を行う(図 1)[1]。

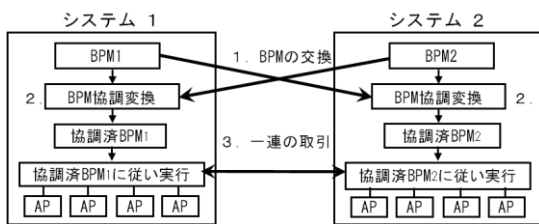


図 1 AWS 動作

## 3. ビジネスプロセスモデル (BPM)

BPM=(O,B)と定義され、O はオペレーション op の集合。op=(format,pattern)として表わされ、format はメッセージフォーマット、pattern は送受信パターンを表す。B は取引の振舞を表したもので、 $B=(S, s_0, F, \Phi)$ で定義され、S は状態の集合、s0 は開始状態、F は終了状態、 $\Phi$ は op による状態遷移を表す。図 2 に見積を依頼し、見積結果を受け、その後発注または取引を中止する手順を表す BPM の例を示す。

果を受け、その後発注または取引を中止する手順を表す BPM の例を示す。

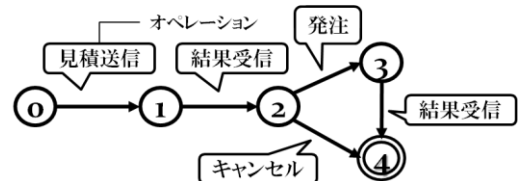


図 2 発注を行う BPM 例

BPM は現在状態遷移表の形で定義する方法と正規表現による定義方法との 2 種類が扱える仕様になっており [3]、決定性オートマトンの範囲に限定して BPM を定義することができる。

## 4. 従来の DMH アルゴリズム

DMH アルゴリズムを以下に示す(図 3) [1][2]。図の丸は状態、アルファベットは遷移 op を表し、「A と a」のように対になるものは整合性のある op であるものとする。

- i. 自身の BPM<sub>1</sub> の振舞の状態 S<sub>1</sub> と相手 BPM<sub>2</sub> の振舞の状態 S<sub>2</sub> の直積を取る。
- ii. i の結果に対し各パスの format 及び pattern の整合性を検査し、整合性のあるパスを作る。(整合性検査に関しては[4]参照。)
- iii. 両者共に終了状態である状態を新しい終了状態とし、到達不能なパス、状態の削除を行う。

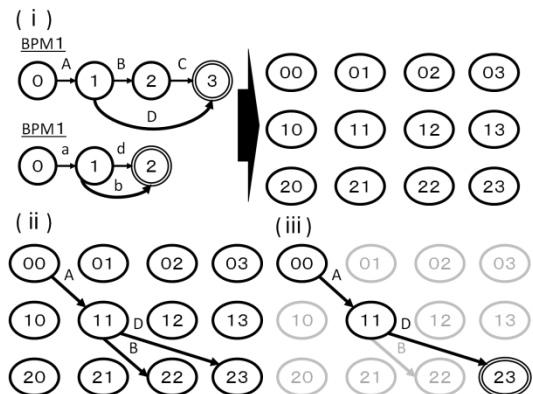


図 3 DMH アルゴリズム

## 5. 複数の BPM を用いた取引

実際の取引を行う際複数のシステムが連携しあい取引を行うことが考えられる。本論文での連携した取引とは、図 4 のようにシステム同士が連携しあい動作する形態の取引である。現在の DMH

Improvement of Dynamic Model Harmonization

<sup>†</sup>Hiroaki Otomo, <sup>†</sup>Taichiro Anzai, <sup>†</sup>Taisuke Kimura,

<sup>†</sup>Makoto Oya, Shonan Institute of Technology

では1対1での取引しか想定していない。そのため複数のシステムを連携させた取引を行いたい場合、連携を想定したBPMを設計しておかなければならない。より柔軟で自律したWebサービスの構築を実現するためには現在のDMH手法だけでは不十分である。新たに複数のBPMを用いたDMHアルゴリズムを提案し、より柔軟なシステム構築を可能とする。

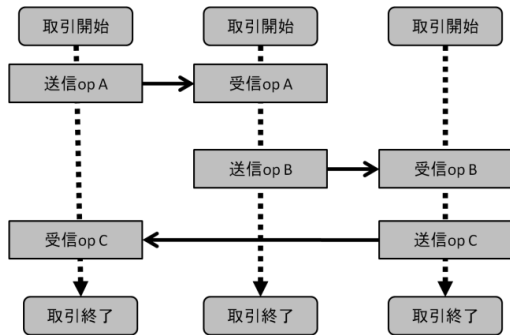


図4 複数のシステムが連携した取引例

### 6. 新たに提案したDMHアルゴリズム

各状態全てに  $\epsilon$  遷移を付加し自身と他一つの遷移だけが行われる遷移だけを残すことで複数のBPMを用いた協調変換を行う。以下に新しく提案したDMHアルゴリズムを示す(図5)。前例と同じくBPM1を自身のBPMとする。

- i. 状態全てに  $\epsilon$  遷移を付加する。
- ii. 利用するBPM全ての状態Sの直積を取る。
- iii. iの結果に対し、整合性のある一組のパス以外が  $\epsilon$  遷移であるパスをつくる。
- iv. 全BPMの状態が終了状態である状態、および終了状態が一組以上で他全てが開始状態である状態を新たな終了状態とし、到達不能なパス、状態の削除を行う。
- v. 自身の遷移を含まない状態をまとめる。

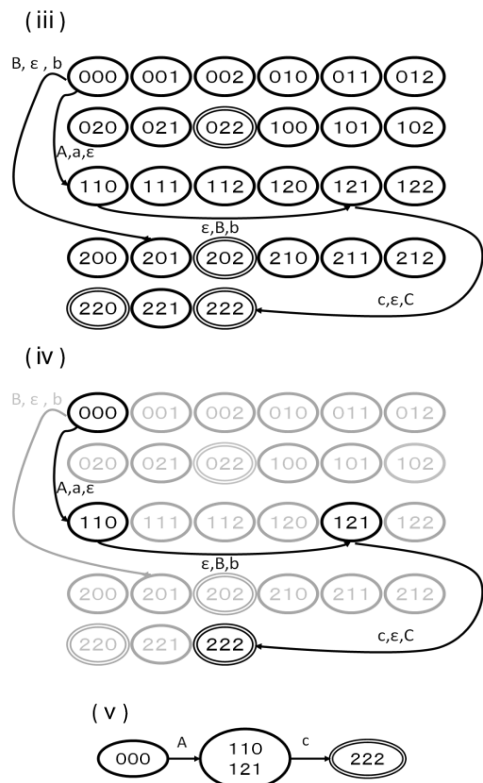
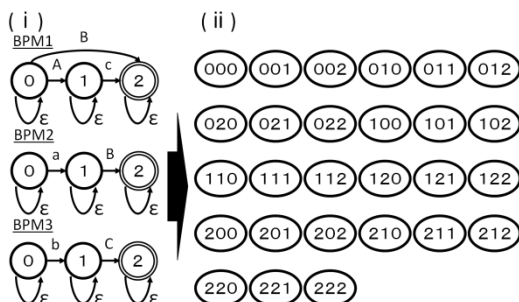


図5 複数BPMの協調アルゴリズム

問題点として終了状態を決める時、開始状態から開始状態へと遷移を行うようなモデルに対して正しく動作しない。よってこの解決方法として、BPMを解析する際に図6のように変換することとした。

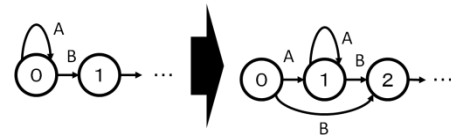


図6 BPMの変換

### 7. まとめ

従来のDMHアルゴリズムに加え新たに複数のBPMを用いた協調変換が可能なDMHアルゴリズムを提案した。今後の課題としてより効率の良いDMHアルゴリズムの検討が考えられる。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] 大谷,伊東,塚本,高木,木村,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回全国大会講演論文集, pp.1-503-504, 2009  
 [2] 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-,情報処理学会第71回全国大会, pp.1-511-512, 2009  
 [3] 大友,伊藤,他,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,2010,  
 [4] 安齋,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,2011