

## ビジネス・エンティティー抽出アルゴリズムの評価と提言

— インフォメーション・セントリック・ビジネス・プロセスへのアプローチ —

服部 洋一 太田 健一郎 広瀬 健志郎

## Business Entities Discovery Algorithm: Evaluation and Recommendations

- An Approach to Information-centric Business Processes -

Yohichi Hattori Kenichiroh Ohta and Kenshiroh Hirose

一連の業務処理単位に着目したアクティビティー・セントリックなビジネス・プロセスに対して、それらアクティビティー間で渡されるデータ（ビジネス・エンティティー）に着目した新しいアプローチのビジネス・プロセス・モデリング手法（インフォメーション・セントリック・ビジネス・プロセス）が IBM 基礎研究部門により提唱されている。その手法の中心となるビジネス・エンティティー（BE）をアクティビティー・セントリック・ビジネス・プロセスより自動抽出するアルゴリズムが IBM バンキング・インダストリー・フレームワークの Core Banking Transformation（CBT）に採用されている。本論文では、CBT を海外仕向送金シナリオに適用した結果を紹介し、BE 抽出アルゴリズムを事例で評価し、提言を行う。

The activity-centric business process approach focuses on the series of activities in a process. IBM Research proposes a new approach to business modeling (an information-centric business process approach), which focuses on the data (business entities) that are passed through the series of activities. The algorithm to discover the business entities (BE)—the core of this new approach—from the activity-centric business process is used by the Core Banking Transformation (CBT) of the IBM Banking Industry Framework. The authors applied CBT to the overseas remittance scenario and evaluated it. This paper first presents the result of this evaluation of the BE discovery algorithm, then points out some of the issues raised, and finally provides some possible solutions for these issues.

**Key Words & Phrases** : Core Banking Transformation (CBT), ビジネス・エンティティー・ライフサイクル分析 (BELA), インフォメーション・セントリック・ビジネス・プロセス, ビジネス・ステートマシン (BSM)  
Core Banking Transformation (CBT), Business Entity Life Cycle Analysis (BELA), Information-centric Business Process, Business State Machine (BSM)

## 1. はじめに

ビジネス・プロセスを自動化し、管理する手法の一つに、ビジネス・プロセス管理がある。これは、ビジネス・プロセスの一連の業務処理単位（以下、アクティビティー）を、その処理される順に従って接続し、業務全体のワークフローを作成し、ビジネス・プロセス・エンジン上で実行することにより業務の自動化を実現する手法である。このワークフローを、記述言語によるモデル化によって作成する手法をビジネス・プロセス・モデリング（BPM）と呼ぶ。ビジネス・プロセス管理と同じ略語を使用するが、本論文では、ビジネス・プロセス・モデリング、もしくはその成果物であるビジネス・プロセス・モデルの意味

で BPM を使用する。BPM の記述言語には、ビジネス・アナリストにとって記述しやすい、グラフィカルな記述法である Business Process Modeling Notation (BPMN) [1]、ランタイム制御のための適切な機能が豊富で、実装局面での使用に適した Business Process Execution Language (BPEL) [2] などがある。さて、このような BPM は、アクティビティーの一連の記述からなるので、アクティビティー・セントリック BPM とも呼ぶ。アクティビティー・セントリック BPM は、世界中ですでに多くの導入例があるが、幾つかの課題点も指摘されている。例えば、イベント・ドリブンな記述や実行済みのアクティビティーに差し戻す処理の記述に不向きであるという点、さらに、目まぐるしいビジネスの変更への即応性が弱い、ビジネス変更要求を取り入れながら複雑化しやすいなどの点である [3] [4]。これに対し、ワークフローのアク

提出日:2010年9月3日 再提出日:2011年3月18日

ティビティー間で渡されるデータ、すなわち、ビジネス・エンティティーに着目した新しいビジネス・プロセスの手法が IBM 基礎研究部門を中心に提唱された。この手法によるビジネス・プロセスをインフォメーション・セントリック・ビジネス・プロセスと呼ぶ。インフォメーション・セントリック・ビジネス・プロセスの中心となる手法は、IBM 基礎研究部門による Business Entity Life Cycle Analysis (BELA) [5] である。BELA は、ビジネス・エンティティー（以下、BEとも略記する）を対象のビジネス・プロセスの中心に据え、BEの状態（ステート）とその遷移をモデル化したライフサイクル（有限ステートマシン）を分析し、ビジネス・プロセスを実現するための分析手法である。複雑度が増すにつれ、アクティビティー・セントリック BPM は、全体の理解が困難となりやすい傾向がある。そこで、複雑なビジネス・プロセスを幾つかの BE のライフサイクル・モデルに分割することにより、ビジネスの意図の理解を容易にすることが可能となる。

また IBM 基礎研究部門は、Model Driven Business Transformation (MDBT) と呼ばれる手法を考案し、BPM から SOA ランタイム実装モデルを自動変換により生成することを可能とした [5]。さらに、アクティビティー・セントリック BPM を起点に、BE を自動抽出し、そのライフサイクル・モデルを構築するアルゴリズム (BE 抽出アルゴリズム) を提唱した [3]。

BELA の分析アプローチ、MDBT のモデル変換

手法および BE 抽出アルゴリズムを銀行業へ応用したソフトウェア・グループのアセットが、Core Banking Transformation (CBT) である。CBT はバンキング・インダストリー・フレームワーク [6] としても位置付けられている。筆者らは、この CBT を国内のお客様に提案する前に、Proof of Concept (PoC) として評価する機会を得た。PoC では海外仕向送金シナリオを例に取り、BPM を作成し、CBT の提供するワークベンチ機能 (Banking Transformation Workbench : BTW) を適用し、モデルの変換、BE の自動抽出、ランタイム・コードの実装などを実施した。

本論文では、CBT を適用した結果、抽出された BE を評価・考察し、不要な BE が抽出されるなどの問題点を指摘し、BE 抽出アルゴリズムの改善案を提言する。

## 2. インフォメーション・セントリック BPM

インフォメーション・セントリック BPM の概要を図 1 に示す。その図の上部に、8 個のアクティビティー (Activity A ~ H) で定義された簡略化したアクティビティー・セントリック BPM の例が示されている。各アクティビティーには、その処理を行うサービス定義が関連付けられている。それらアクティビティーへの入出力となるデータ・エンティティーとして「契約」と「顧客」があるとすると、前述の BE 抽出アルゴリズムを適用すると、「契約 BE」と「顧

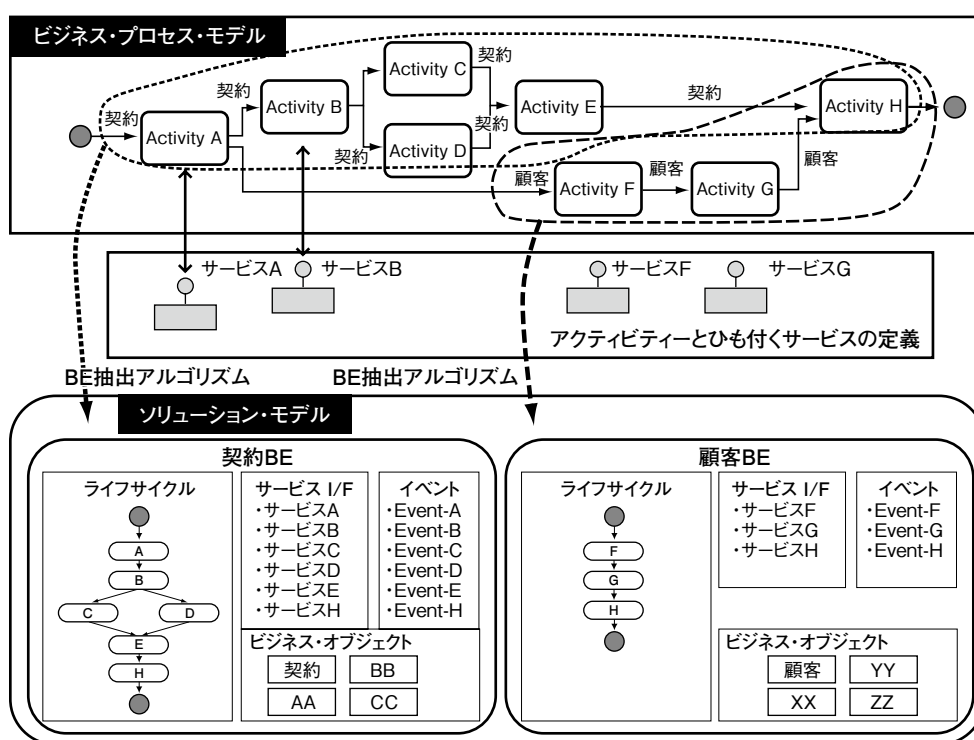


図1. インフォメーション・セントリック BPM

客 BE」の BE が抽出される。それぞれの BE には、そのエンティティのライフサイクルや、アクティビティにひも付き、ライフサイクルのステートを遷移させるサービスのインターフェース、遷移をトリガーするイベント、BE 内で使用されるビジネス・オブジェクトが含まれる。

このように、アクティビティ・セントリックなビジネス・プロセス・フロー内のデータ・エンティティに着目し、主要なエンティティを抽出し、そのライフサイクル、サービス、イベント、およびビジネス・オブジェクトを内包する BE へと再構築し、BE 内部のモデルを分析・設計により洗練させていく BPM の新しいアプローチがインフォメーション・セントリック BPM であり、その中心となる手法が BELA である。なお、抽出された BE の集合体をソリューション・モデルとも呼ぶ。

### 3. CBTの海外仕向送金への適用

#### 3.1 ビジネス・プロセスの概要

PoC で取り上げた海外仕向送金のビジネス・プロセスは、以下の 2 種類である。

- 受付・対顧処理(プロセス#1)
- 外為事務センター処理(プロセス#2)

プロセス #1 は、さらに以下のようなサブプロセスからなる。

1. 海外仕向送金依頼書をOCR(光学式文字読取装置)でスキャンし、受付登録

2. 代り金口座振替の場合、印鑑/署名の照合
3. 違法性および疑わしい取引のチェック(図2)
4. 送金金額が指定額を超えている場合、ポジション報告を実施
5. 資金連絡が必要な場合、資金連絡の実施
6. 対顧客の勘定処理を実施し、外為事務センターの処理をトリガー

プロセス #2 は、以下のサブプロセスからなる。

1. PO(ペイメント・オーダー)を作成し、再鑑
2. POを外部ペイメント・システム(SWIFTなど)に発信

これらの処理フローを WebSphere® Business Modeler (WBM) 上にモデリングした。WBM はビジネス・ユーザー向けに BPMN 構文をサポートし、グラフィカルにビジネス・プロセス・フローを作成可能なツールである。図 2 に簡略化した BPM の一部を示す。図中の長方形(“System”と記載)はシステム・タスクを、二重線長方形はヒューマン・タスクを、ひし形は条件分岐を示す。データ・フローはここでは省略した。

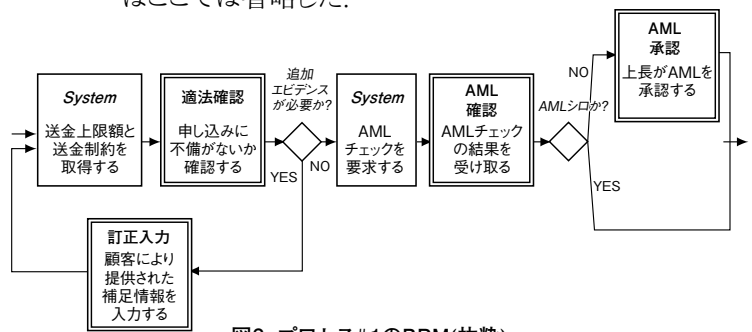


図2. プロセス#1のBPM(抜粋)

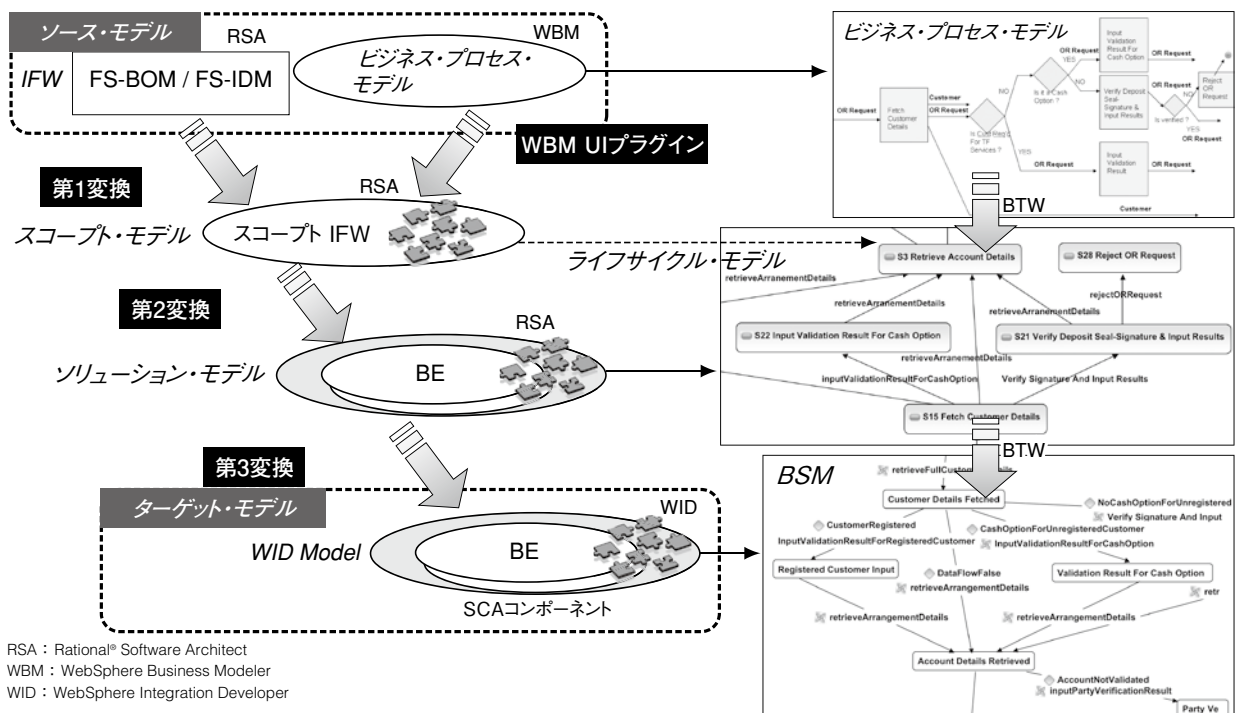


図3. CBTワークベンチ(BTW)の概要

RSA : Rational® Software Architect  
 WBM : WebSphere Business Modeler  
 WID : WebSphere Integration Developer

## 3.2 CBTの適用

次に、CBTの提供するワークベンチ BTW を使用し、3.1で作成した BPM からのモデルの変換を実施した。ここでは、BTWの概要(図3)およびCBT適用の手順について示す。

BTWは前述MDBTのモデル変換手法を受け継ぎ、ソース・モデルから3段階のモデル変換を経てターゲット・モデルを自動生成する。その間、2種類の間接モデルも生成される。以下、これらのモデルとそれぞれの変換について示す。

### (1) ソース・モデル

ソース・モデルとなるのが、金融機関向けフレームワーク IFW (Information Framework) [7] である。IFWは、金融機関向けに開発された、金融機関のさまざまなビジネス情報を全社規模で構築するためのフレームワークで、15年以上にわたり世界中のお客様に導入いただき、拡張を続けてきたモデル群が提供される。BTWでは、この中の以下の IFW モデル群が使用される。

- ビジネス・プロセス・モデル
- 分析モデル (FS-BOM: Financial Services Business Object Model)
- 設計モデル (FS-IDM: Financial Services Interface Design Model)

FS-BOMは業務要件分析、ドメイン分析に活用されるオブジェクト指向による分析モデルであり、FS-IDMはFS-BOMをベースとしてコンポーネント指向設計を行うための設計モデルである。

IFWはお客様ごとに銀行固有の業務要件に合うようにカスタマイズを行うことが可能だが、今回のPoCの実施にあたっては、IFWを大規模にカスタマイズした場合を想定し、そのように大規模カスタマイズしたIFWに対するCBTの稼働検証を行うため、独自のカスタマイズに比重を置く方針とした。従って、ビジネス・プロセス・モデルに関しては、ビジネス・プロセス・フローを基に3.1で述べたように独自に構築した。FS-BOMのデータ・モデルに関しては、約半分を新規に定義し、ビジネス・プロセスのアクティビティに関連付けられるFS-IDMのサービスをすべて新規に定義した。

### (2) 第1変換: ソース・モデル(IFW) → 第1中間モデル (スコープ・モデル)

IFWのFS-BOM/FS-IDMは豊富なオブジェクト・モデルを提供する反面、作成する業務によっては使用されないモデルも多数含まれる。そこで、第1変換として、IFW

の中からビジネス・プロセスで使用されるエンティティおよび関連するクラスのみを抽出(スコーピング)し、モデルの絞込みを行う。事前に、ビジネス・プロセスのアクティビティを、その処理を実現するビジネス・サービスと関連付ける必要がある。このサービスは(1)でFS-IDMに追加したものである。関連付けは、前述のビジネス・プロセス・モデリング・ツールであるWBM上のBTWによる拡張機能を使用して行い、アクティビティごとにFS-IDMサービスとの関連を定義した。その後、第1変換を実施し、中間モデル(スコープ・モデル)へ変換した。

### (3) 第2変換: 第1中間モデル → 第2中間モデル (ソリューション・モデル)

この変換において、BE抽出アルゴリズムが使用されており、BPMよりBEが抽出され、ソリューション・モデルが自動生成される。通常はこの後、ソリューション・モデルでの洗練を行うが、PoCでは割愛した。

### (4) 第3変換: 第2中間モデル → ターゲット・モデル (WIDモデル)

最終変換では、ソリューション・モデルから最終ターゲット・モデルを自動生成する。これまで、分析・設計のためのモデルを扱っていたが、この段階で初めてビジネス・プロセス・エンジン上で実行可能なランタイム・モデルとなる。BTWでは、SCA (Service Component Architecture) に準拠し、WebSphere Integration Developer (WID) 上にSCAコンポーネントが作成されることにより、最終的に生成されたターゲット・モデルをWIDモデルと呼ぶ。最終変換は、ストレート・フォワードな変換であり、ライフサイクル・モデルからBSM (Business State Machine) が、サービス・インターフェースからWSDL (Web Services Description Language) 文書が生成される。

生成されたWIDモデルはあくまでランタイム実装のためのひな形に過ぎず、SCAプログラミング・モデルにとり、WIDモデル上の実装を行う必要がある[8]。アクティビティ・セントリックBPMがBPELを実装手法に据える場合が多かったのに対し、インフォメーション・セントリックBPMではBEのライフサイクル・モデルに対応するBSMが実装の中心となる。

## 4. BEの抽出

### 4.1 BELAにおけるBE抽出アルゴリズム

前述BE抽出アルゴリズム[3]の概要を示す。まず、

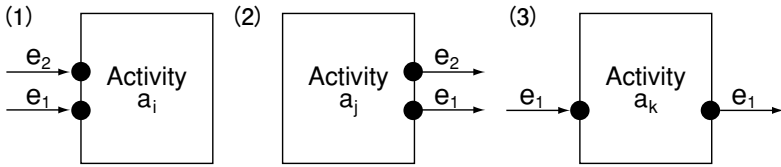


図4. データ間の支配関係

BPM の各アクティビティの入力と出力データ項目に着目し、データ間の支配関係を決定する。プロセス・スコープ  $s$  において、データ項目  $e_1$  がデータ項目  $e_2$  を支配することを  $e_1 \alpha e_2$  と記す。これは、以下の3つの判定条件がすべて成立する場合と定義する。

- (1): 「 $s$ において、 $e_2$ がアクティビティの入力にある場合、常に $e_1$ も同じアクティビティの入力にある」
- (2): 「 $s$ において、 $e_2$ がアクティビティの出力にある場合、常に $e_1$ も同じアクティビティの出力にある」
- (3): 「 $e_2$ が入出力になく、 $e_1$ が入力もしくは出力にあるアクティビティが $s$ において存在する」

図4にこの支配関係について補足する。矢印と●は、入出力データ・フローを示す。

BE抽出アルゴリズムは、アクティビティへの入出力データ項目間での支配・被支配の判定を上記条件に基づき行い、それをプロセス・スコープ内の全アクティビティに対して行いながら、一度も被支配と判定されなかったデータ項目（支配的エンティティと呼ぶ）をBEとして抽出する。図5に例示する。

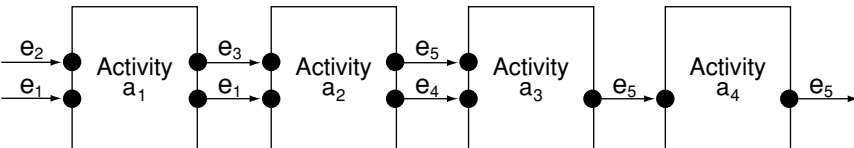


図5. 支配関係の例

この例では  $a_1 \sim a_4$  の4つのアクティビティ、 $e_1 \sim e_5$  の5つのデータ項目がある。まず、アクティビティ  $a_1$  に対しては、2つの入力  $e_1$  と  $e_2$  がある。この時点では、 $e_1$  と  $e_2$  は対等である。次のアクティビティ  $a_2$  に対しては、2つの入力  $e_1$  と  $e_3$  がある。この時点で、 $e_2$  が存在しないため判定条件 (3) が成立する。アクティビティ  $a_3$ 、 $a_4$  には  $e_1$  と  $e_2$  は存在せず、 $a_1$  において (1) が成立、(2) はそもそも  $e_2$  が出力にないため成立、以上により、 $e_1 \alpha e_2$  が成立する。また、同様に  $e_3$  に対してはアクティビティ  $a_1$  において入力に存在せず、判定条件(3)が成立し、 $e_1 \alpha e_3$  が成立する。アクティビティ

$a_3$  に対しては、新たなデータ項目  $e_4$  と  $e_5$  が入力に存在し、アクティビティ  $a_4$  に対しては、入力は  $e_5$  しか存在しない。従って同様に  $e_5 \alpha e_4$  が成立する。ここまで支配的エンティティとして残ったのが  $e_1$  と  $e_5$  であり、この2つの間に関しては支配関係がないため、 $e_1$  と  $e_5$  がBEとなる。残りのデータ項目

の内、 $e_2$  と  $e_3$  は  $e_1$  に支配され、 $e_4$  は  $e_5$  に支配される。すなわち、上記例で示したように、支配されるデータがアクティビティに入出力する場合、常にその支配するデータ (BE) が共に入出力する。

このように、BE抽出アルゴリズムは、BPM内の全アクティビティを手繰りながら、全データ項目間の支配関係を導き出し、最終的に自分以外のどのデータ項目にも支配されないデータ項目、すなわち支配的エンティティをプロセス・スコープ内のBEとして抽出する。さらに、BEのライフサイクル・モデルを構築する。

#### 4.2 BE抽出結果とその理由

PoCでは、CBTの適用により以下の4つのBEが抽出され、ライフサイクル・モデルが構築された。

- PaymentInstruction BE (支払い指示)
- PaymentOrder BE (ペイメント・オーダー)
- Customer BE (顧客)
- ORRule BE (海外仕向送金ルール)

以下、それぞれの抽出結果の詳細を考察する。

##### (a) PaymentInstruction BE

当オブジェクトはプロセス #1 の海外仕向送金依頼書を受け付けた時点で生成され、プロセス #1 の各アクティビティで一貫して入出力として使用される。従って、プロセス #1 の支配的エンティティであり、BEとしての抽出は妥当なものである。

##### (b) PaymentOrder BE

当オブジェクトも同様にプロセス #2 の各アクティビティで使用される支配的エンティティであり、BEとしての抽出は妥当である。

##### (c) Customer BE

当オブジェクトの抽出について以下に考察する。図6にBPMでの定義を示す。ここで、●と矢印はデータ・フローを○と矢印はコントロール・フローを示す。また、PIはPaymentInstructionを示す。

アクティビティ "顧客の詳細情報を取得する" において、顧客情報が取得され、出力にCustomerを設定

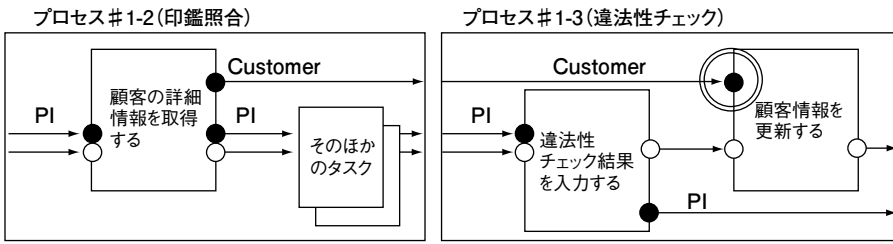


図6. Customerの抽出理由

した。しかし、後続のアクティビティでは Customer を必要とせず、次に必要となるアクティビティ "顧客情報を更新する" の入力へ直接データ・フローを接続した。一方、その間のアクティビティでは PI のみが必要であったため、PI をデータとして渡しながら、コントロール・フローも接続し、"顧客情報を更新する" の直前のアクティビティ "違法性チェック結果を入力する" から、"顧客情報を更新する" へはコントロール・フローのみを接続し、PI は "顧客情報を更新する" をスキップしながら次のアクティビティへ接続した。これは "顧客情報を更新する" が顧客情報のみを入力とするためである。これにより、二重丸で示した部分において、PI に支配されない Customer 単独での入力が存在するため、Customer が BE として抽出されたと考えられる。

(d) ORRule

次に ORRule の抽出について考察する。まず、図 7 にビジネス・プロセス・モデリングでの定義を示す。PO は PaymentOrder の略である。

ORRule は、図 7 に示した 4 アクティビティで使用される。その内、3 つがプロセス #1 で、残り 1 つがプロセス #2 で使用され、いずれも出力項目として定義されている。プロセス #1 の部分のみに着目すると、いずれも PI に支配される。すなわち、ORRule が定義されるすべ

でのアクティビティに PI が出力項目として存在する。一方、プロセス #2 に目を転ずると PO に支配されていることが分かる。しかし、ソリューション・モデルの生成は、プロセス #1 とプロセス #2 をプロセス・スコープとしたため、プロセス #2 において PI に支配されず、プロセス #1 において PO に支配されない、対等なエンティティとして ORRule が抽出されたと考えられる。

### 5. BE 抽出の評価

前節で述べたとおり、PI および PO の BE 抽出は極めて妥当なものであった。しかし、ほかの BE 抽出 (ORRule/Customer) は幾つかの課題を提起する。ここでは、以下の 3 点を指摘する。

- 1) BE抽出アルゴリズムは、エンティティの内容までを考慮しないため、マスター系データのように単にDB照会のためだけのエンティティであってもBEとして抽出される
- 2) 更新を伴うBEが抽出された場合でも、BE抽出アルゴリズムは、アクティビティの内容までを考慮しないため、単にDB照会を行うだけの有用でないステートも作成される
- 3) コントロール・フローとデータ・フローがペアとなっていた場合においてはBEとして抽出されなかったエンティティであっても、それが分離されることによって、BEに成りうる

ORRule は海外仕向送金の各種ルールに関するエンティティであり、参照のみに使用されるが、BEとして抽出された (課題 1)。

Customer はコントロール・フローとデータ・フローの分離の結果、抽出され (課題 3)、ライフサイクル・モデルには DB 照会のための不要なステートが生成されている (課題 2)。

BE 抽出アルゴリズムが、アクティビティ・セントリックなビジネス・プロセスより、支配的なエンティティを自動抽出し、ライフサイクル・モデルを構築可能であることは十分評価できた。一方、現アルゴリズムでは不必要なエンティティも抽出され、また、ライフサイクル・モデルのステートにも不必要なステートが入り込む可能性があるという点を指摘しておく。

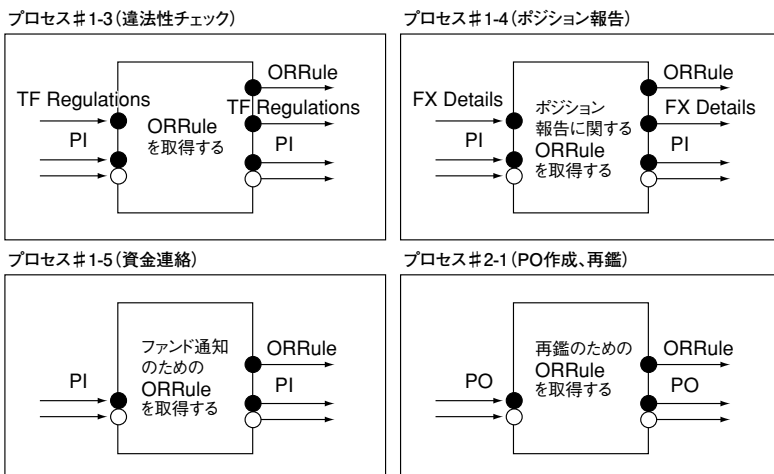


図7. ORRuleの抽出理由

## 6. BE抽出アルゴリズムの改善案

前節で示した課題への改善案を提示する。

A) データベースなどの外部データストアから参照されるだけで更新の発生しないことがビジネス・プロセス・モデリング時点で自明なデータ項目(例えば、マスター系データ)に対して、BE候補から落とすためのアノテーション(BE非抽出アノテーション)を設定し、当アノテーションが設定されたデータ項目はBEの対象から外す。例えば、図8において $e_2$ を非抽出アノテーション(@NoDiscovery)が設定されたデータ項目とする。この場合、 $e_2$ は無条件にBE抽出候補から外される。

当改善により、課題1が解決され、ORRuleのような参照系みのデータがBEとして抽出されることは解消される。

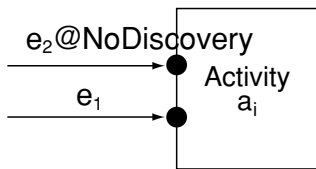


図8. BE非抽出アノテーション

B) 次の改善案(図9)は、BEのライフサイクル・モデル生成の改善で、課題2を解決する。あるデータ項目をBEとして抽出する場合であっても、BPM内のアクティビティを

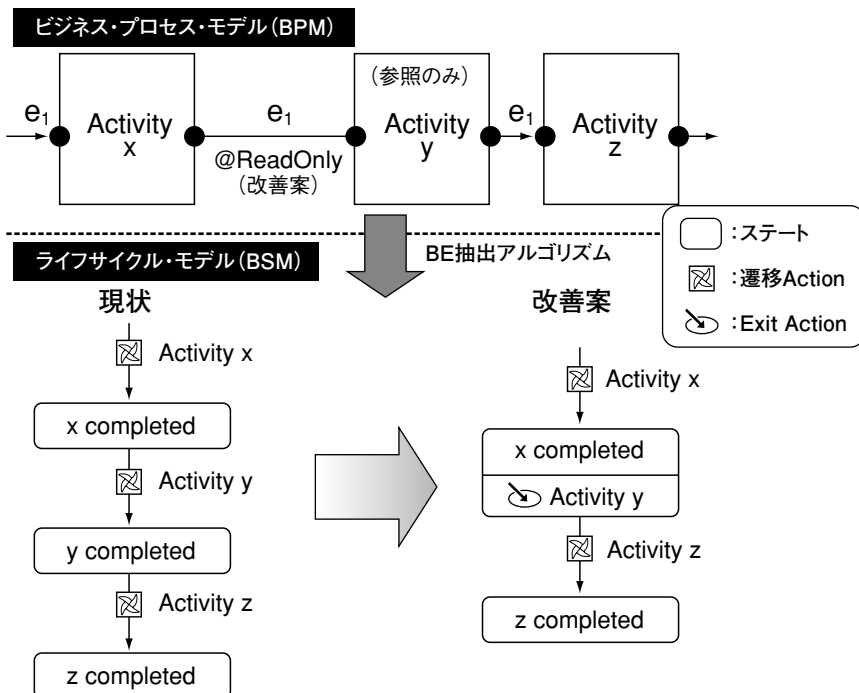


図9. 参照専用アノテーションによる改善案

無条件にステート・遷移に対応させるのではなく、例えば参照専用のアノテーションを設定することによって、アクティビティのステート・遷移への変換を抑制するものである。図9のBPMに示すように、3つのActivity(x,y,z)が定義されているとし、その内Activity yがDB参照のみを行うものであり、ステートへの変換を抑制したいものとする。現状では、図9下の左側に示すように、アクティビティに対し、完了を示すステートが無条件に作成され、アクティビティはそのステートへの遷移時に呼び出される遷移アクションに変換される。

改善案では、ステートの作成を抑制したいアクティビティに対して、その入力(もしくは出力)データ項目に参照専用アノテーション(@ReadOnly)を設定可能とし、設定された場合、図9下の右側で示すように、当該データ項目がBEとして抽出される場合であっても、ステートの作成を行わず、アクティビティを遷移前ステートのExit Actionに変換する。

これにより、ORRuleなどの参照系アクティビティをステートから除外可能で、生成されるライフサイクル・モデルの本質的でないステート数と遷移数が削減され、モデルがシンプルになることにより、生成モデルが質的に向上する。

C) 課題3に対しては、BPMにおいて、コントロール・フローとデータ・フローの分離を許さないという制約を課した上で、アクティビティ内で使用せず、後続のアクティビティに渡すためだけにデータ・フローを接続する場合に、データ・エンティティにアノテーション(@THRU)を付与することにより、その接続におけるアクティビティは生成されるライフサイクル・モデルから除外するという改善案が考えられる。(エンティティの支配・被支配判断は実施。) 図10に、改善案によるBPMの設定例を示す。これによりCustomerはPIに従属する。

## 7. おわりに

CBTを海外仕向送金シナリオに適用し、BE抽出アルゴリズムを評価し、課題点を指摘した。また、課題に対して、ビジネス・プロセス・モデリングの際に、データ項目やアクティビティが参照専用などの判断が可能な場合、アノテーションを付加することによ

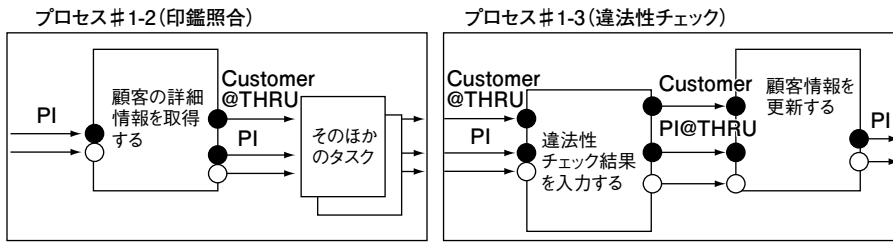


図10. 課題3の改善案

り、不要な BE の抽出やステートの生成を抑止する改善案を提言した。ビジネス・プロセス・モデリングの際に、判断がつかなかった場合でも、イテレーティブな分析アプローチを採用している場合に、ソリューション・モデルの洗練の過程での判断をフィードバックして、次の再変換に活用することも可能である。

今後は、機能拡張要求に応えながら長大化する BE のライフサイクルをどのように保守すべきかなど BE の保守方法についての検討を行いたい。

謝辞

CBT の PoC の実施にあたっては、[3] の著者でもある Santhosh Kumaran 氏、CBT 開発チームの V.R. Pramodh 氏、PoC オーナーの上野敏彦氏に多くの助言をいただきました。この場を借りて、あらためて深謝いたします。

参考文献

[1] Object Management Group Business Process Management Initiative: <http://www.bpmn.org/> (2011.2.22) .

[2] IBM: "Business Process Execution Language for Web Services version 1.1," developerWorks®, <http://www.ibm.com/developerworks/library/specification/ws-bpel/> (2011.02.22) .

[3] S. Kumaran, R. Liu, and F. Y. Wu: "On the Duality of Information-Centric and Activity-Centric Models of Business Processes," Proc. the 20th International Conf. on Advanced Information Systems Engineering (CaiSE '08), Lecture Notes in Computer Science 5074, Springer, New York, pp.32-47 (2008) .

[4] Zimmermann, O. et al: "Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned," Proc. of the 20th SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, pp. 301-312 (2005) .

[5] J. K. Strosnider et al.: "Model-driven synthesis of SOA solutions," IBM Systems Journal, Vol.47, No.3, pp.415-432 (2008) .

[6] IBM "IBM Banking Industry Framework," <http://www.ibm.com/software/industry/banking/> (2010.08.20) .

[7] N. Norris et al: "Building Service-Oriented Banking Solutions with IBM Banking Industry Models and Rational SDP," IBM Redbooks, <http://www.redbooks.ibm.com/redpapers/pdfs/redp4232.pdf> (2011.03.16) .

[8] G. Adams et al.: "IBM WebSphere Developer Technical Journal: A guided tour of WebSphere Integration Developer -- Part 1," developerWorks, [http://www.ibm.com/developerworks/websphere/techjournal/0602\\_gregory/0602\\_gregory.html](http://www.ibm.com/developerworks/websphere/techjournal/0602_gregory/0602_gregory.html) (2010.08.07) .



日本アイ・ビー・エム株式会社  
金融システム品質本部  
銀行ソリューション  
第一ソリューション  
シニア・アーキテクト

服部 洋一 Yohichi Hattori

[プロフィール]

1989年、日本IBM入社。大和研究所にて、主に金融インダストリー向け製品開発に従事。現在はサービス部門において、銀行基幹系システムのリノベーションを促進するアセット開発を担当する。  
hattoy@jp.ibm.com

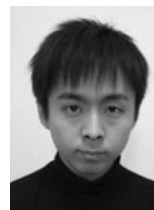


日本アイ・ビー・エム株式会社  
グローバル・ビジネス・サービス  
テストング・サービス  
テストマネジメント  
アドバイザー・ITスペシャリスト

太田 健一郎 Kenichiroh Ohta

[プロフィール]

2001年、日本IBM入社。オープンソースとRationalツールを使ったテスト自動化を中心に各種のプロジェクトに携わる。Rationalツールを活用した迅速なアジャイル開発を促進するものとして、開発とテストをクラウド上で行うことに興味を抱き始めている。  
ootaken@jp.ibm.com



日本アイ・ビー・エム株式会社  
グローバル・ビジネス・サービス  
EA&T  
インダストリーアプリケーション  
ITスペシャリスト

広瀬 健志郎 Kenshiroh Hirose

[プロフィール]

2006年、日本IBM入社。様々な業態のアプリケーション開発に従事。現在はIFW等特定の業界のアプリケーションに特化したアセットや製品の導入、推進に携わっている。  
kenshiro@jp.ibm.com