

## ビジネスプロセスモデリングと実行可能 UML を連結させた 情報システム構築法の試作と評価

北島 聡史<sup>†</sup> 中村 義幸<sup>†</sup> 上西 司<sup>†</sup> 小泉 寿男<sup>†</sup> 坂 和磨<sup>††</sup>

<sup>†</sup> 東京電機大学大学院理工学研究科 <sup>††</sup> (株)ビジネスプロセスウェア

現在、システム分析の最上位に用いられるビジネスプロセスモデリング(BPM)の有用性が認められてきている。BPM では業務プロセスの現状分析モデル(As-Is モデル)を作成し、それを基に、構築するシステムを含めて最適化されたモデル(To-Be モデル)を創出することを目的とする。また、実装前の段階でシステムの可動性検証を行うことができる実行可能モデル言語 ExecutablXUML(xUML)が注目されている。本稿では、ビジネスプロセスモデリングをもとに、業務フローのシミュレーションによるボトルネックの分析と評価を行い、モデルの最適化を行う。そのあと、そのモデルを xUML へ変換し、実行可能なモデルでのシステムの動作確認を行う。本方式を部品調達システムのプロトタイプを試作に適用し、評価を行った。

### An Information System Development Method connecting Business Process Modeling with executablXUML and its experimental evaluation

Satoshi Kitajima<sup>†</sup> Yoshiyuki Nakamura<sup>†</sup> Tsukasa Kaminishi<sup>†</sup>  
Hisao Koizumi<sup>†</sup> Kazuma Ban<sup>††</sup>

<sup>†</sup> Graduate School of Science and Engineering, Tokyo Denki University

<sup>††</sup> Business Processware co.

Recently, the effectiveness of business process modeling (BPM) which gives the top-level view in the system analysis stage is gaining wide recognition. BPM aims to generate an optimized model (To-Be model) based on an analysis model (As-Is model) of the current business process. Also the executable modeling language, Executable UML (xUML), which enables execution and test of the modules before implementation stage is gaining attention. In this paper, we describe the generation of an optimized To-Be model after analyzing and evaluating bottlenecks through business process simulation based on an As-Is model. Then, we describe the transformation of the model into xUML and execution & test of the system on the model. We applied the proposed method to prototyping a parts procurement system, and we describe a part of the result of the evaluation.

#### 1. はじめに

情報システムの構築において、構築対象システムの分析・設計を行うモデリングの重要性が高まっている。業務プロセスの分析・改善への方策として、ビジネスプロセスモデリング (BPM) <sup>[1]</sup>が注目されている。BPM は、現状の人材やシステム、情報などの資産を把握することにより業務プロセスをビジネスプロセスとしてモデル化し可視化・効率化を図る手法である。しかしながら、BPMによって得られた業務改善プロセスを、実際にどのようにして情報システムの構築に結び付けていくかについての研究は殆ど行われていない。BPMによって確認されたプロセスモデルをUMLに引継ぎ、あとはUMLの手法によって実装し

て行くことができれば、情報システム構築の効率向上が期待される。一方、OMG(Object Management Group)が提唱し、仕様化を進めているモデル駆動型アーキテクチャ MDA <sup>[2][3][4][5]</sup>は、開発するシステムのモデルをUMLモデルを活かし、プラットフォーム非依存で作成して、そのモデルを変換して行くことで実装へと至るシステム構築技法である。また、ExecutablXUML(以下 xUML) <sup>[6]</sup>はモデルとコードのギャップを解消して、MDAを実現する手法として注目されている。

筆者らは、BPMによって業務分析を行い、その結果をUMLのモデルに変換し、OMGが提唱しているMDA <sup>[3][4]</sup>による実行確認を経て実装に至るまでの一貫した構築法

の研究を行ってきている<sup>[7][8]</sup>。

本稿ではビジネスプロセスモデリングと実行可能 UML を連結させた情報システム構築法の試作と評価を述べる。本方式では、BPM によって業務プロセスを改善し、UML に変換して実行可能な Executable UML<sup>[8]</sup> (以下 xUML)へ拡張・動作検証を行う。BPM においては、回までの研究内容に加え、作成したビジネスプロセスモデルに対してシミュレーションを行い、業務プロセスのボトルネックとなる部分に対して改善を行う具体的な方法を示す。改善部分の xUML による実行確認についても述べる。また、xUML による実行確認が BPM に与えるフィードバック効果についての検証を行う。

## 2. 本方式による構築の流れ

本方式は、ビジネスプロセスモデリングを行うことで業務プロセスのボトルネックの解消を行いつつ、ビジネスプロセスモデルを MDA のモデル変換によるシステム構築を行い、設計段階において実行可能なモデルで動作の検証をすることでシステムの可動性を確認することを狙いとす。開発の流れを以下の図 1 に示す。

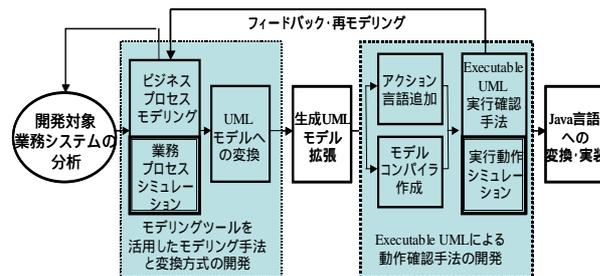


図 1 本方式の開発の流れ

まず、作成の対象となるシステムに関し、複数の業務にまたがっているビジネスプロセスを分析して、現状(As-Is)プロセスを織り込んだビジネスプロセスモデルを作成する。作成した As-Is モデルをシミュレーションにより分析・ボトルネックとなる箇所を割り出し、分析結果をもとに改善(To-Be)プロセスのモデルを作成する。ツールを用いて UML モデルへ変換する。UML モデルの段階では、アクション言語を作成追加し、xUML へと拡張を行う。モデルコンパイラによってコードへ変換してモデルを実行させる。実行評価により、必要なアクション言語追加の再設計、さらにビジネスプロセスモデルの再モデリングが必要な場合には BPM へのフィードバックを行う。

## 3. ビジネスプロセスモデリングの方式

企業の事業活動を分析し、効果的な業務内容を実現すると共に、事業環境の変化に迅速に対応しうる情報システムを構築するために、ビジネスプロセスを可視化して記述す

ることが重要である。モデリングは、単なるお絵かきツールではなく、作成したビジネスプロセスの有意性を検証できると共に、作成したプロセスモデルへの修正、追加、削除などの更新が容易に出来なければならない。

本研究では、ビジネスプロセスの可視化記述、モデルの階層化による段階的詳細モデル化、およびモデル更新が可能な手法とツールを備えた IDS Scheer 社の ARIS (Architecture of Integrated Information Systems)<sup>[9][10]</sup> <sup>[11]</sup> を活用する。ARIS は、ビジネスプロセスをプロセスビュー、ファンクションビュー、データビュー、組織ビュー及びアウトプット/サービスビューの 5 つのビューから記述し、プロセスビューにて 5 つのビューを統合化する。

### 3.1 業務プロセスの分析と改善

#### (1) 業務プロセスの現状分析 (As-Is モデル作成)

まず、業務プロセスの現状分析を行う。現状の業務プロセスをビジネスプロセスモデルとして表記し、表記したモデルを現状のモデル(As-Is モデル)として分析に用いる。業務プロセスはプロセスビューに含まれる図 eEPC 図(拡張イベントプロセス連鎖図)で表すことができる。

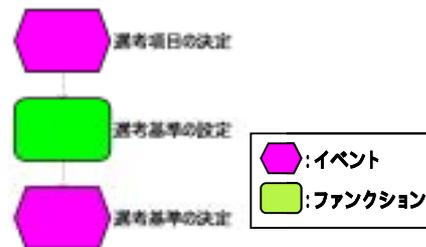


図 2 ビジネスプロセスを表す eEPC 図の例

eEPC 図では「イベント」及び「ファンクション」の 2 つの要素を主に用いる。イベントはプロセスにおける情報オブジェクト(モデリングの対象になる情報)の状態やファンクションの起動や結果を表す。ファンクションは、企業目的を達成するために情報オブジェクトに対して遂行される行為や作業をさす。

eEPC 図は機能の単位で処理の流れを描く。eEPC 図内にある 1 つのファンクションの内容をさらに eEPC 図を用いて詳細に書き、階層化されたモデルとして表すことができる。

eEPC 図を書くためには他の各ビューで記述されたオブジェクトを用いる。

#### (2) 業務プロセスの改善・強化 (To-Be モデル作成)

作成した As-Is モデルの問題点に対して改善・強化を行い、To-Be モデルを作成する。これから作成するシステムを業務内に組み込み、それが動作している状態を想定してモデル化し、それを最適化する。修正した To-Be モデルが、As-Is モデルから改善されていることを判断するためには、

シミュレーションを用いて統計結果を比較する。改善が見られない場合は再度 To-Be モデルに手を加えていく。

### 3.2 業務プロセスのシミュレーション

業務プロセスのシミュレーションはビジネスプロセスの分析・検証であり、ARIS ツールの持つ機能により行う。作成した eEPC 図に対して、処理の発生回数や各ファンクションの処理時間、分岐の確率など必要な情報を設定し、任意に指定した一定の期間に対しシミュレーションは行われる。一般的なプロセス分析と異なり処理時間の合計のみならず、プロセスが処理を開始するまでの待ち時間を示す動的待機時間や、各プロセスの実行状況など、詳細な統計結果が得られる。その統計結果より、一定期間内に実行可能なプロセス数の把握、プロセスに含まれる弱点やボトルネックの把握、リソース利用の最適化やコストの抑制などについても分析することができる。

シミュレーションは As-Is モデル、To-Be モデルの両方で行われる。As-Is モデル作成後の統計結果を基準値として、To-Be モデル時との比較を行いつつ、To-Be モデルの作成を行う行程となる。シミュレーションを含むモデルの作成行程を以下に示す。

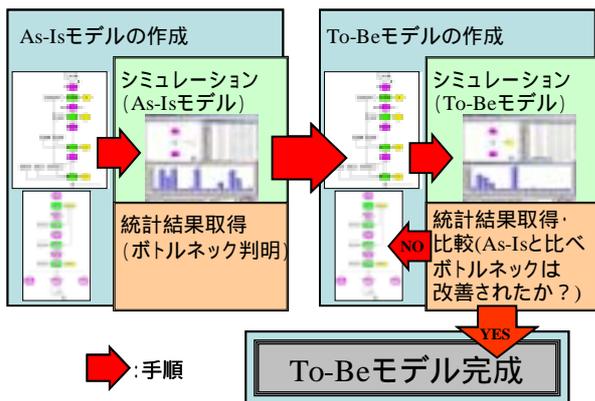


図4 シミュレーションを含むモデルの作成行程

To-Be モデルの作成はシミュレーションを繰り返し、As-Is モデルの統計結果と比較してプロセスの改良によるボトルネックの改善やシステムが行う機能の割り振りなどを行う。プロセスの改良が出来たと作成者が判断できるまでプロセスの改良を行い、To-Be モデルは完成となる。

## 4. 実行可能 UML への変換・動作検証

### (1) UML への変換

図1に示したように、まず、作成したビジネスプロセスモデルから各種 UML モデル図への変換を行う。

### (2) 実行 UML への拡張

ビジネスプロセスモデル図より変換された UML モデル図に対して拡張を行い、実行可能な xUML のモデルとする。実行確認には UML ステートチャート図を用いる。

拡張されるモデルから生成するメタモデル、アクション言語、セマンティクスおよびモデルコンパイラの関係を図5に示す。

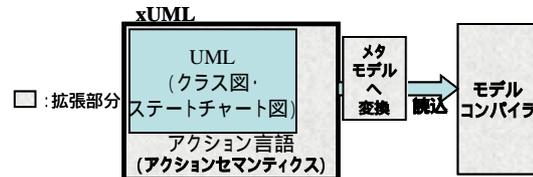


図5 xUML への拡張と実行

実行シミュレーションを行うためにはモデルに動作が詳細に記述されていなければならない。アクションセマンティクスは、モデルの動作の標準仕様を記述し、それに沿ってモデルに書き込む動作の記述としてアクション言語を規定する。モデルに記述される動作はシステムのデータフローによって生成されるため、実行シミュレーションはシステムのデータフローを追うことを意味する。

アクション言語はアクションセマンティクスに基づいて作成されるが、実際のアクション言語の構文には関係せず、セマンティクスを満たせば開発者が自由に作成できる。

モデルに記述したアクション言語を解析する機構をモデルコンパイラといい、これによりアクション言語を含むメタモデルが解釈され、実行シミュレーションが行われる。このとき、モデル自体の持つ情報を一意に読み込める形式へ変換しなければならない。ここで、モデルをメタモデルに変換する。本稿では意味情報を記述した UML メタモデルの記述に関する意味論や文法に MOF (Meta Object Facility) [11]を用いる。MOF はメタモデルの構成要素を定義し、またメタモデルを管理するための標準技術仕様である。本方式では、MOF を基盤としてモデルの情報を XML 形式で管理するために、UML のメタデータとしての標準技術仕様である XMI (XML Metadata Interchange)を用いる。

### (3) モデルの作成

UML モデルから実行可能 UML モデル作成の流れを以下の図6に示す。

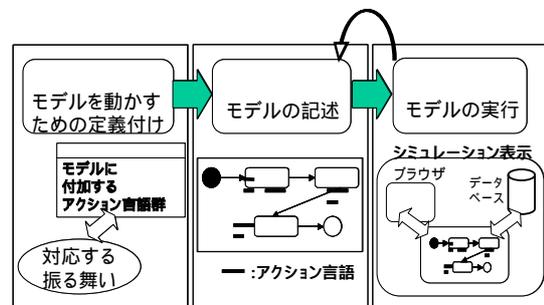


図6 モデル実行までの流れ

モデルを動かすための定義付け (アクション言語の作成): モデルと動作を表すアクション言語を定義付ける操

作を行う。構築すべき実行環境は、モデルの動作に対応したアクション言語の作成である。実際の設計においては、新たなアクションを追加するときのみ手動で行う。

**モデルの記述 (UML 作成ツールの使用):** 第1のフェーズで定義したアクション言語を付加したモデルを作成する。本研究では、現在、ステートチャート図上での実行を検討している。アクション言語はこの図上に適宜付加する。

**シミュレーションの実行 (モデルコンパイラの作成):** 前のフェーズで記述したモデルによるシミュレーションを行う。モデルの解析はXMI ファイルのタグをもとに行う。解析結果にはモデル情報とアクション言語が含まれる。結果をもとにデータフローをオブジェクトごとに作成する。

#### (4) 実行可能UMLからBPMへのフィードバック

次に、xUMLでの実行シミュレーションからビジネスプロセスモデリングへのフィードバックが考えられる。具体的には、業務プロセスからビジネスプロセスモデルを作成してからxUMLを作成し実行した場合に、業務プロセスが統合できる部分を発見した場合である。

### 5. 部品調達システムへの適用・評価

#### 5.1 部品調達システムの概要と業務の流れ

部品調達システムは、業務システムの評価のために研究室にて作成したプロトタイプシステム<sup>[12]</sup>である。このシステムは、e-マーケットプレイス上で逆オークション形式を用いて買い手側企業(バイヤー)が調達したい部品の購入希望条件を提示し、複数の売り手側企業(サプライヤー)から条件に合った企業を選択する支援を行う。本調達システムでは一回目の見積り選考のあと、選考に通ったサプライヤーに再見積りを依頼し、これらのプロセスを繰り返すことにより適切な業者が選択されていく。

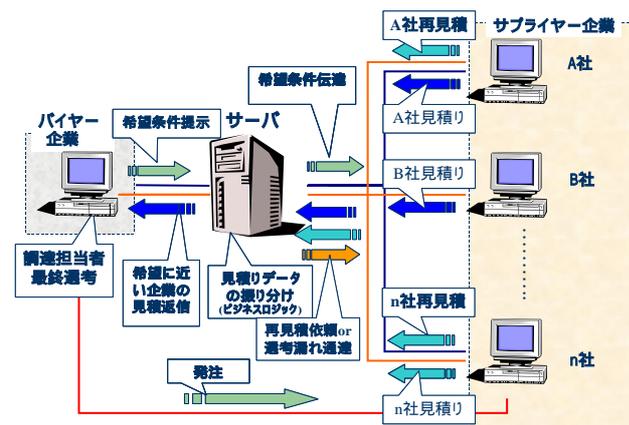


図7 部品調達システムの流れ

部品調達システムの一連の流れを図4に示す。流れは、調達条件をWebで提示する。サプライヤーは調達条件を参照する。サプライヤーはサーバに見積りを送信する。

バイヤーは選考条件をサーバに送信し、システムは見積書の中の単価・品質・取引実績の3つを参照し、見積りの選考を行う。バイヤーは選考結果から、よい見積りを出したサプライヤーに再度見積りを出してもらうか、各サプライヤーに直接交渉するかを決定する。選考漏れ企業には落選通知を、選考に残った企業には再見積り依頼を電子メールで通知する。サプライヤーは再見積りを送信する。システムは再見積りの選考を行う。選考に残った見積りをバイヤーに送信する。バイヤーの担当者による選考を行う。最後に残ったサプライヤーを発注対象とする。となる。

#### 5.2 ビジネスプロセスモデリングの作成

まず、As-Isモデルの作成を行った。部品調達システムの要求仕様を検討し、必要な部品の確認からサプライヤーへの見積り依頼の検討までを資材部の業務、見積り依頼の依頼から選考結果の報告までを営業部の業務と規定した。資材部と営業部の業務をそれぞれ3つのフェーズに分割し、その業務全体の大まかな流れ(上位モデル)を1枚のeEPC図として作成し、一つ一つのフェーズを詳細に記述したeEPC図6枚(下位モデル)を作成して階層化した。

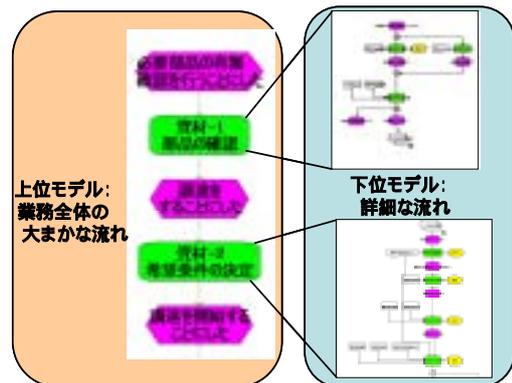


図8 作成したeEPC図の階層

モデルを作成後、シミュレーションを行った。まず、シミュレーションを行うために必要なパラメータを設定した。必要なパラメータはeEPC図内のオブジェクトに直接設定を行うものと、シミュレーションのためにパラメータモデルを作成することで準備する。eEPC図内には図内で用いられるオブジェクトであるイベントやファンクションに関するパラメータを設定した。シミュレーションのために作成するモデルは2種類で、処理の流れる頻度を規定したプロセスインスタンス生成モデル、業務を行う人間の就業期間を規定したシフトカレンダーの2つである。2モデルを作成した後、シミュレーションを行った。

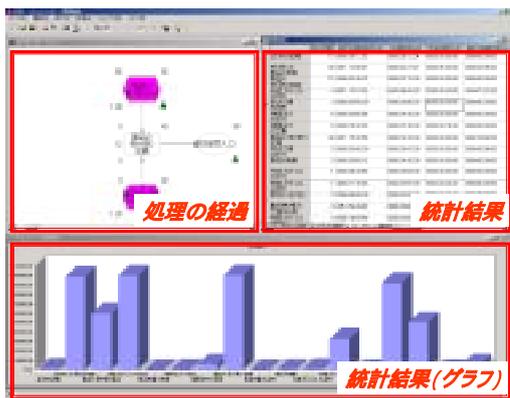


図9 シミュレーション画面

上が実際に行ったシミュレーションの画面である。シミュレーションはリアルタイムに動作し、指定した期間の中、処理が図中のどこまで到達したかが表示される(図中左上)。また、どのファンクションで待ち時間が発生したか、統計表(図中右上)とグラフ(図中下)出表示させることが出来、ボトルネックとなっている箇所と時間が分かる。またシミュレーション終了後は、流れた処理ごとの処理時間・待ち時間の詳細な統計が確認できる。

結果、見積要求送信・部品数の確認・見積選考処理などでボトルネックが発生していた。

To-Be モデルの作成においては、システム化できる部分をシステム化し、業務プロセス中でボトルネックとなる箇所についてもシステム化の検討を行った。ボトルネックとなっていた見積要求送信・部品数の確認を手で行っていた箇所をシステム化し、見積選考処理の一部をシステム化してフローを改善したところ、ボトルネックが改善された。

### 5.3 xUML への変換

#### (1) UML へのモデル変換

作成した To-Be モデルから UML への変換を行った。各種の UML モデル図への変換は eEPC 図の中に変換後の図に対応したオブジェクトがない場合などは変換の前後に操作が必要となる。例えば、ユースケース図は eEPC 図内に操作対象の人間を表すオブジェクトが存在するために操作の必要はないが、クラス図の場合は、ファンクションを実行するのに必要なクラス・属性を追加し、変換を行う。このとき、eEPC 図は必要に応じて詳細化し、必要な機能・属性を明確にしておく必要がある。変換後にファンクションは機能となる。また、アクティビティ図の場合は、ファンクションに付随する状態を追加し、変換を行う。変換後は状態を表すアクティビティに加えて、部署ごとに区切られたレーン(区切り線)が抽出されるので、開始・終了状態を追加し、これらに対し接続線を結んで完成させる。

今回はユースケース図・クラス図・状態チャート図

の作成を行った。生成されたクラス図を以下に示す。

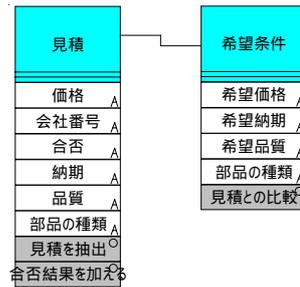


図10 生成されたクラス図

#### (2) 実行可能 UML への拡張

実行シミュレーションに用いる UML 図は状態チャート図である。シミュレーションに付加するアクション言語は独自のコンパイラにより解釈されるが、表記法は基本的に UML1.4 ベースである。ゆえに、シミュレーション用の図の描き方と、従来用いてきた描き方に大きな違いはない。振る舞いを表現するアクションの表記に若干の制約(あらかじめ用意されているアクションをベースに拡張を行うなど)が設けられる程度である。

本シミュレーションで用いるアクション言語は独自に作成したが、一部は OMG で提唱されているアクションセマンティクスに準拠している。作成したモデルをメタデータである XMI 形式へ変換するために MDA ツールである Poseidon for UML を用いた。

#### 5.4 実行可能 UML での動作確認

ボトルネックが改善されたと思われる部分についてアクション記述を内包した状態チャート図を XMI の形に変換して動作シミュレーションを行った。実行シミュレーションの行程とデータの流れを図 11 に示す。

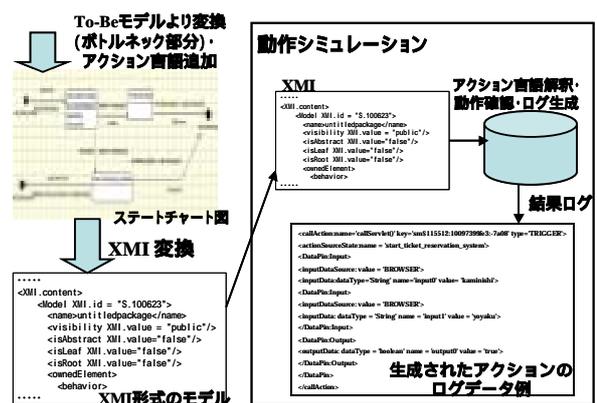


図11 実行シミュレーションの行程

作成した xUML の XMI ファイルからシミュレーションに必要なステート、変数、振る舞いなどの情報を抽出する。抽出されたデータを基に、クラスごと(状態チャートごと)にオブジェクトの変化をシミュレートするが、クラ

ス間でのオブジェクトの遷移をシミュレートすることも可能である。また、シミュレーションログは随時更新され、エラーが生じた場合のバックトレースも可能である。

シミュレーション結果はログデータに格納される。ログデータとして保存されるのは、生成されたアクション、実行したステート情報、用いられる変数、データタイプなどである。動作確認結果については現在評価中である。

## 6. 評価と考察

### (1) BPM・シミュレーション

シミュレーションを行うことでボトルネックとなる処理の待ち時間を減らすことが出来た。待ち時間のなかでも過負荷による待ち時間をシミュレーションでき、シミュレーションする期間の長さや休日をはさんだプロセスなども考慮できるために、業務内に新たなシステムを導入する際にボトルネックとなる箇所を特定しつつ、問題の改善を考えたシステムを提案することができた。後はコストとスピードとのトレードオフについては、設計者の配慮が必要になると思われる。

今回 eEPC 図を 2 層に階層化して全体の業務プロセスを構築したが、システムが大きくなればさらに下位のモデルを作成する必要がある。UML への変換が複雑となるため、3 層で階層化を行った場合、あらかじめ階層ごとに扱うオブジェクトの単位を上位：パッケージ・中位：クラス・下位：オペレーション/属性と規定しておくが良い。3 層の場合の変換法確認については今後の課題である。

### (2) 実行可能 UML による動作確認

実行シミュレーションは現在作成段階であり、コンパイラの作成中であるが、アクション言語の規定までを行い、プロトタイプではステートチャート図内の一部のアクション言語を読み込むことが出来た。今後は読み込んだモデルの動作を表示すると共に、エラーが検出された場合に図中への表記、同時にオブジェクトの更新を行う機能を考えている。その後の実装への繋がり、また BPM へのフィードバックの役割を果たすものと考えられる。

eEPC 図から UML クラス図へ変換する方法としてファンクションをオペレーションに変換する方法を述べたが、オペレーション自体を eEPC 図へ書き込む方法もある。1 ファンクションに対して内包する複数のオペレーションを規定できるので、特に eEPC 図を階層化せずに変換が可能である。手間はかからないが、ファンクションに繋がるどのオペレーションが処理時間を必要とするのか分からず、ボトルネックの判別がしにくい問題点もあるので、階層化したほうがシミュレーションは行いやすい。

## 7. まとめ

本稿では、ビジネスプロセスモデリングと実行可能な

UML による開発を連結させた情報システム構築の方式を提案し、その一部の作成・検討を行った。部品調達システムのプロトタイプを対象として、現状となる As-Is モデルの作成と、シミュレーションの統計結果を基にボトルネックを解消した To-Be モデルの作成を行った。To-Be モデルはシミュレーションによりボトルネックの改善を確認した。また、To-Be モデルを UML へ変換し、その後の xUML への拡張の手順を示した。実行可能な UML から BPM へのフィードバックとして考えられるものを挙げた。

今後は、BPM から実装までの一環構築手順の評価を行う。評価対象は BPM 後、UML の実行確認後、実装後で、評価内容として、整合性とそれぞれの有効性についてを取り上げる。さらに、以前構築した UML のみを経て実装した場合のシステムと xUML による動作確認を経た実装した場合との比較と検討、xUML からシミュレーションへのフィードバックについての検討を行う。以上の評価から BPM から実装までの流れの確立を目指す。

## 参考文献

- [1] 戸田保一、飯島淳一編、ビジネスプロセスモデリング、日科技連、2000
- [2] <http://www.omg.org/mda/>
- [3] 山田正樹、モデリングとツールを駆使したこれからのソフトウェア開発技法 モデル駆動開発手法を中心として、情報処理学会学会誌、Vol.45 No.1、pp.3-9 2004.
- [4] ANNEKE KLEPPE 他著、長瀬嘉秀 訳、“MDA モデル駆動型アーキテクチャ導入ガイド”，インプレス、Dec. 2003
- [5] 峰岸巧、永田守男、神谷慎吾、山本修一郎、安東孝信、山城明宏、MDAに基づくソフトウェア開発の事例と開発プロセス、研究報告「ソフトウェア工学」No.140 - 003、2002.
- [6] スティーブJメラー、マークJ.パルサー、“Executable UML MDAモデル駆動型開発の基礎”，株式会社テクノロジックアート
- [7] 北島聡史、小泉寿男他、ビジネスプロセスモデリングによる情報システム構築の一手法、情報処理学会 DPS-119 研究報告、pp.15-20、2004
- [8] 小島義幸、北島聡史、小泉寿男他、ビジネスプロセスモデリングによる部品調達システムの構築とその評価、情報処理学会 IS-90 研究報告、pp.9-16、2004
- [9] ARIS, IDS Scheer, <http://www.ids-scheer.co.jp>
- [10] シェア A-W 著、坂和磨監訳、“ARIS ビジネスプロセスモデリング”，シュプリンガーフェアラーク社
- [11] “Action Semantics for thxUML”, Object Management Group
- [12] 石川俊之、小泉寿男他、自動選考機能を持つ部品調達ビジネスモデルと構築評価、情報処理学会 DPS ワークショップ、pp.133-138、2001.