ビジネスプロセスモデリングと実行可能なモデルを基盤とした情報システム設計手法とその評価

上西 司† 平林秀一† 小泉寿男† 大川 勉††
†東京電機大学 理工学研究科 情報システム工学専攻
††三菱電機株式会社情報システム技術センター

BPM(Business Process Management/Modeling)が業務プロセスの管理・分析・改善の手段として用いられるようになってきた．また，実装前にシステムの動作検証を行うことが可能なxUML(eXecutable UML)が注目されている．BPMでは，業務プロセスの現状をモデル化(As-Isモデル)し，モデル分析によってその問題点を解明し，それを基に，最適モデル(To-Beモデル)を創出することを目的としている．xUMLでは，生成したUMLがシステムが動作するのに妥当であるモデルか否かを判断することを目的としている．われわれはこれまでに，BPM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・評価からモデルの最適化を行い，そのモデルをxUMLへ変換し，xUMLでのシステム動作確認を行うシステム設計方式を提案してきた．本稿では，ビジネスプロセスモデリングの最適モデルから生成したxUMLについて，xUML設計法とシミュレーションが可能な開発手法の一部構築を行い，プロトタイプへ適用し，評価を行った．

An Information System Design Method Based on Business Process Modeling & Executable Models and Its Evaluation

Tsukasa Kaminishi† Shuichi Hirabayashi† Hisao Koizumi† Tsutomu Ookawa††
†Department of Computers and Systems Engineering
Graduate School of Science and Engineering
Tokyo Denki University
††Information Technology Center
Mitsubishi Electric Corporation

Business process modeling is gaining wide recognition as a measure of management, analysis and improvement of the business process. Also the executable modeling language, xUML(executable UML), which enables execution and test of the modules before implementation stage is gaining attention. Business process modeling aims to generate an optimized model(To-Be model)based on an analysis model(As-Is model) of the current business process. xUML aims to judge whether it is the UML that model is appropriate to the movement of the system. We have suggested an information system development method, that the generation of an optimized To-Be model after analyzing and evaluating bottlenecks through business process simulation based on an As-Is model, then the transformation of the model into xUML and execution & test of the system on the model. In this paper, we describe that construction and application of a part of the development method which enables xUML design and simulation about the generated UML based on business process modeling, and we evaluate a part of the result.

1. はじめに

情報システム開発におけるモデリングを用いた設計の普及は目覚しく，上流・下流工程問わず用いられている[4]．その背景には，UML(Unified Model Language)の認知度の高まりがある．UMLはシステムを視覚化できるという利点によって広く浸透してきた．さらに，近年，業務プロセスをモデル化しフローの検証・業務管理などを行うBPM(Business Process Management/Modeling)[1]，あらゆる工程でモデルを中心とした開発を行うMDA(Model Driven Architecture)[2]，実行可能なUMLであるxUML(executable UML)[3]といった技術が登場し注目され，モデリング技術浸透の追い風となっている．こうした動向にもかかわらず，実際にビジネスプロセスモデルから実装へと結びつけた研究が殆ど行われていない．われわれはこれまでに開発の上流工程にBPMを，下流工程にxUMLを用いた実行可能なモデリングを行う，要求定義からのシームレスなシステム開発手法(以下，BPM/実行可能モデリング連携手法)を提案し，主にBPMの有効性を示してきた[7][8]．

しかしながら，これまでBPMから生成・拡張したxUMLによる実行可能なモデリング手法の有効性については検証していなかった．さらに，xUMLの設計に関しては現在存在するxUML設計方法ではモデルごとに小規模プログラミングが発生するという難点も存在する．

本稿では，xUMLの抱える問題を解決する新たなxUML設計方法を提案し，その設計方式を実現できる（xUMLの生成から検証

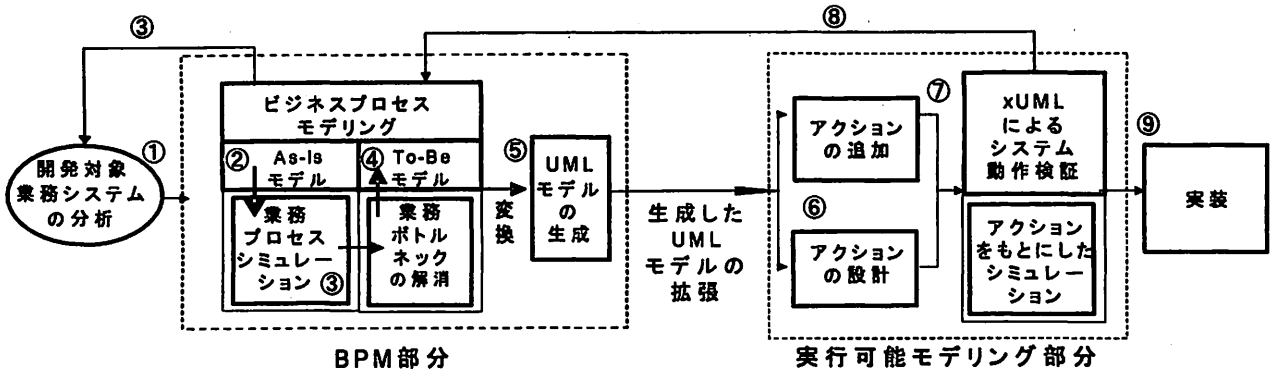

図1　BPM/XM連携手法の流れ

までをサポートする）開発手法を一部構築し，xUMLの一部有効性の評価を行う．また，実装工程においては，これまでソースコードの生成，およびコーディングが実装手段であったが，今回，これまでとは異なる実装方式を提案する．

2. BPM/実行可能モデリング連携手法

われわれのこれまで提案してきた設計方式では，

- BPMを行うことで業務プロセスの管理，およびシミュレーションによるボトルネックの検知が可能
- ビジネスプロセスモデルからMDAのモデル変換によってUMLを生成し，シームレスな工程移動が可能
- 設計段階において実行可能なモデル(xUML)で動作の検証を行うことでシステムの可動性確認を狙いとしている．

今回は新たに，

- シミュレーションエンジンを実装に用いることによるコーディングの軽減

を本方式に組み込む．本方式の流れを図1に示す．図1の流れは次のようになる．

① 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の分析を行う．
② 現状プロセスを織り込んだビジネスプロセスモデル(以下，As-Isモデル)を作成する．
③ 作成したAs-Isモデルをシミュレーションにより分析しボトルネックとなる箇所を割り出す．場合によっては要求仕様へフィードバックする．
④ 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．
⑤ ツールを用いてUMLモデルへ変換する．
⑥ アクションを作成追加し，xUMLへと拡張を行う．
⑦ モデルコンパイラによってモデルを解釈・実行させる．
⑧ 実行評価により，または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．
⑨ シミュレーションエンジンを利用して実装を行う．

2.1 BPM部分

BPMは，ビジネスプロセスの有意性を検証すると共に，作成したプロセスモデルへの修正，追加，削除などを容易に実現することを目的としている．本方式では，図1で示す①～⑤のBPM部分に，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機能を備えたIDS Scheer社のARIS (Architecture of Integrated Information Systems)[6]を活用する．

(1) ARISによるビジネスプロセスモデリング

ARISは，ビジネスプロセスを次の4つのビューから記述する．

- 組織ビュー：社内の組織構造をモデル化したもの．
- プロセスビュー：業務プロセスを表すモデル．イベントとファンクションという2つのモデル要素によって表す．
- ファンクションビュー：システムの機能を分割し，階層表示したモデル．
- データビュー：業務処理遂行に関わるデータとその相関関係を記述．

これら4つのビューは互いに関連付けられており，一部の図の修正が即座に反映されるようになっている．図の外形と関連の様子を図2に示す．以下に，モデリングの手順を述べる．

① 図2に示すように，まず，組織構造をモデル化し，組織ビューを作成する．通常，階層のトップに会社自体，もしくは社長や役員会を置く．
② サブ組織ごとに扱うシステムやサービスを構成する機能を分割してモデルし，ファンクションビューを作成する．
③ ファンクションビューの各機能について，プロセスをフローチャート形式でモデル化し，プロセスビューを作成する．
④ ファンクションビューの作成時に，必要なデータリソースを関連付ける．データリソースは別途，データビューとして階層構造でモデル化したものを用いる．

このような手順で現状の組織・システム形態をモデル化したものをAs-Isモデルといい，改善を施したものをTo-Beモデルという．

(2) プロセスのシミュレーション分析とUML変換

作成したプロセスビューに対して，処理の発生回数や処理時間，

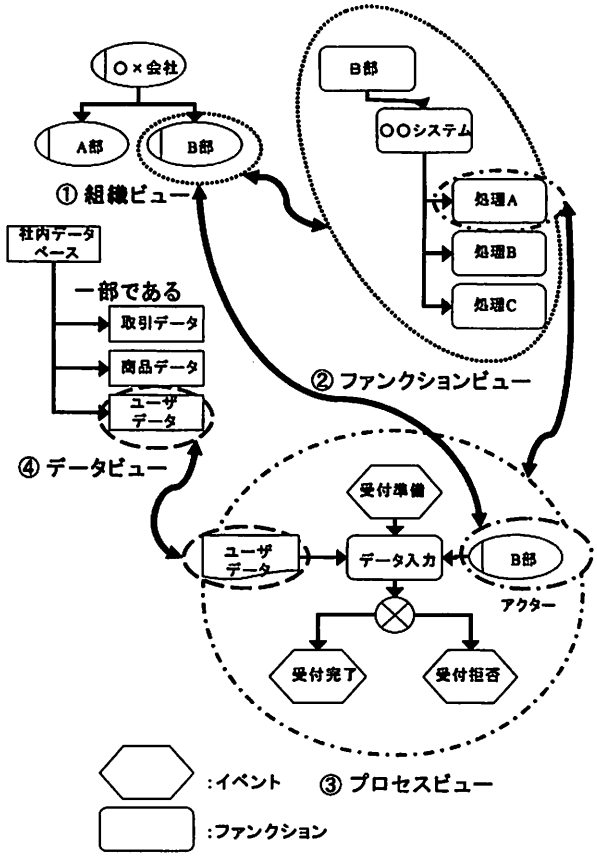

図2 ARISによるビジネスプロセスモデリング

分岐の発生する確率や必要人員などのパラメタを設定し，任意に指定した一定期間に対し，業務シミュレーションを行う．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，詳細な結果が出力される．これらの結果を比較することで，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．分析結果をモデルに反映させ，To-Beモデルを作成する．

To-Beモデルが完成すれば，これにツールのモデル変換機能を用いてプロセスビューをUMLの各モデルに変換していく．

2.2 実行可能モデリング

実行可能モデリングの目的は早期にシステムの検証を行い，アジャイルな開発を行うことである．しかしながら，xUMLの作成手法や，作成したxUMLを解釈・実行する機構の明確な作成方法などは標準化されておらず，またxUMLに付加する振る舞い(以下，アクション)のセマンティクス以外は開発者に委ねられているのが現状である[3]．さらに，xUMLのアクション設計においては小規模プログラミングが発生する．本方式では，図1で示す⑥～⑧の部分が実行可能モデリング部分であり，実行可能モデリングを実現するための開発手法は独自に作成している．

以下に，本方式における実行可能モデリングの流れを述べる．

① まず，アクションの設計を行う．本方式におけるxUML設計では予めいくつかのアクションを提供している．提供しているアクションを本方式が提案している方式により設計する．この設計方式により，アクション設計の小規模プログラミングを回避する．

② 設計したアクションをBPMから生成したUMLに追加する．

③ UMLモデルのメタデータ(XMI形式)からシミュレーションに必要なモデルのフローやアクション情報を読み取る．

④ モデルより抽出した情報を基に，モデルコンパイラがアクションを逐次実行していく．

⑤ アクションの実行結果は逐次，設計者に通知される．また，累積ログはログファイルに出力される．現状ではリアルタイムログはコンソール上に示す形となっている．

⑥ ログファイルにはシミュレーション途中での警告やエラー内容が書き込まれる．設計者はこれらのデータを吟味して，改善点を見つけていく．

⑦ 改善されたモデルとシミュレーションエンジンを用いて，システムを動かす実環境と結びつけ，実装する．

3. 実行可能モデリング手法の構築

3.1 モデリングに必要な機能と各機能の関連

本方式において実行可能モデリングを行う際に必要な機能とその関係を図3に示す．

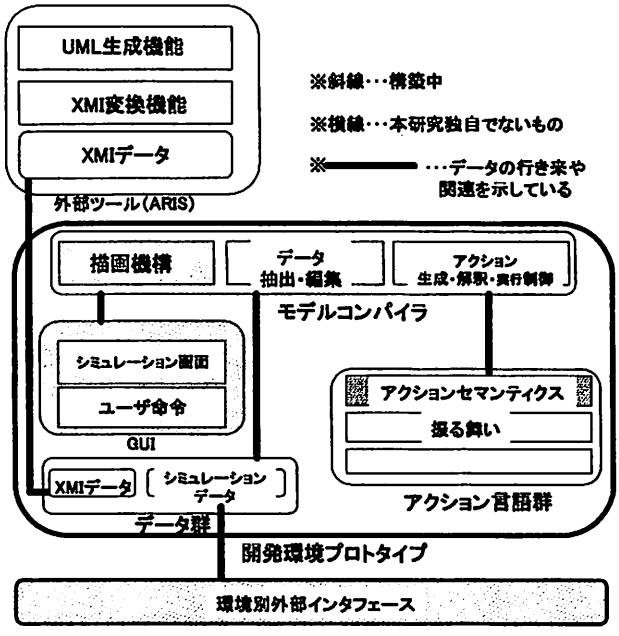

図3 実行可能モデリングの機能とその関係図

図中の斜線部分は現在構築中である．さらにGUI部分とアニメーション表示機能もまだ設計途中である．しかしながらモデル情報の抽出とアクション言語の生成・解釈・実行などは可能であるため，GUI部分をコンソールとのやりとりで代行してのモデルのシミュレーションは可能である．

図3に示す各機能は，本方式における実行可能モデリングで行う次の3つの工程を実現する．

- 独自アクション設計によるxUML作成
- xUMLシミュレーションによるシステムの動作検証
- シミュレーションエンジンを利用した実装

3.2 独自アクション設計工程

xUMLに付加するアクション設計においては小規模プログラミングが発生する場合が多い．本方式ではその点を改善するため，予めアクションをいくつか用意しておき，それらを組み合わせることでより複雑なアクションを表現できるようにしている．

(1) 本方式によるアクション設計の特徴

本方式のアクション言語には以下のような特徴がある．

- メソッド呼び出し形式の記述：「アクション名()」によりアクションを記述できる．
- シミュレーション用スクリプト：動的検証を発生させるスイッチとしての役割がある．
- ステートチャート図表記法の活用：設計者は標準ルールであるUMLルールをベースとして設計が可能である．
- 属性付加によるアクションの変化：ひとつのアクションによる振る舞いにバリエーションを持たせる．
- 選択と組み合わせによる設計：アクション設計における小規模プログラミングを回避する．

(2) 提供するアクションの構成

提供するアクションはプリミティブなものばかりである．

設計者は提供されたプリミティブなアクションからいくつか選択し，または組み合わせて振る舞いを記述する．これらプリミティブなアクションの機能分割はアクションセマンティクス[3]を参考にしている．

3.3 xUMLシミュレーション工程

(1) シミュレーションの目的と検証可能な範囲

先にも述べたがシミュレーションの目的は設計図となるUMLが構築するシステムの用件に見合ったものであるかどうかを確認するためである．

目的を達成するためにはシステム全体に対してシミュレーションすることが好ましい．しかし，現状で本開発手法の適用対象は1つのドメインに限られている．適用可能なモデルも現状ではUML図のステートマシン図に限られている．

(2) シミュレーションの動作

シミュレーションの実行はモデルコンパイラにより，ステートマシン図のステートごとに逐次実行されていく．

モデルコンパイラは，ステートの先頭から逐次モデルとアクション情報を解析し，実行していく．実行結果はその都度，ログファイルやコンソール上に反映される．ステートが並列フローとなったり，ステート内のアクションが並列アクションである場合(時間制限のあるアクションなど)はそれぞれ並列に処理される．

3.4 シミュレーションエンジンを利用した実装工程

一般的に，実行可能モデリングの実装工程ではソースコードの自動生成が最終成果物であり，従来のわれわれの方式もそれに倣っていたが，今回，システムの実行環境の相似点を利用し，ソースコードの生成を行わない実装工程を本開発手法では提案する．

提案する方式は，シミュレーションエンジンを実行エンジンとして用いるという手法である．シミュレーションエンジンと実行環境(データベース，サーバーなど)との間にはデータのやり取りを可能にするインタフェースを用いる．本提案方式は現在設計中である．

4. プロトタイプへの適用と評価

4.1 適用対象システムと現在可能な適用範囲

(1) チケット予約システムの概要

このシステムはユーザがブラウザを用いてチケット予約サーバにアクセスし，オンラインでコンサートチケットの予約を行える会員制のシステムである．ユーザは事前に会員登録を行う必要があり，パスワードは会員ごとに発行されている．システムは1つの予約サーバと2つのデータベース（チケットデータ用，会員データ用）を持つ．また，このサービスを行うのは「カミニシシステム」という会社のIT事業部とする．

(2) 本方式の適用

本方式における開発手法は未完成であり，適用範囲が1つのドメインに限られる．また，シミュレーションエンジンを利用した実装機能についても現在構築中である．従って，適用範囲はシステムの分析からxUMLによる1つのドメインに関するシミュレーションの適用までとする．適用対象ドメインはチケット予約システムの認証ログイン部分とした．

4.2 適用結果

(1) BPMの結果

本方式によるBPMの結果(組織ビュー，ファンクションビュー，プロセスビュー)を図4に示す．

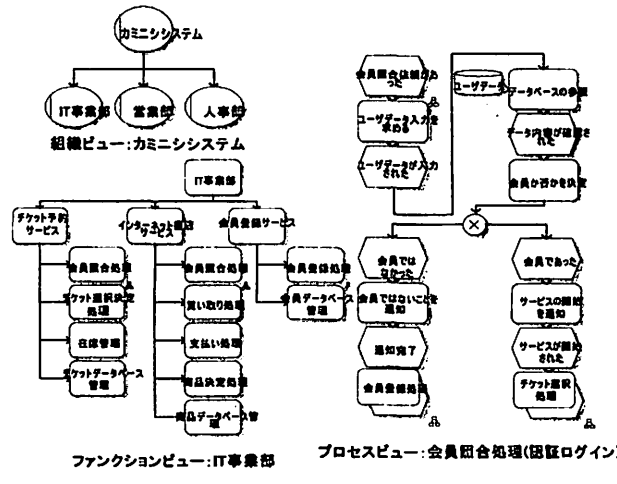

図4 チケット予約システムのBPM結果

(2) UML変換，およびxUMLへの拡張

1日あたりのアクセスを10万件として，1週間分のBPMシミュレーションを行ったが，分析結果からはボトルネックは検出されなかったため，認証ログイン部分である会員照合処理のプロセスビューをUMLのステートチャート図へと変換し，各ステートに本開発手法で提供するアクションを設計し，追加した．図5に作成したxUMLを示す．

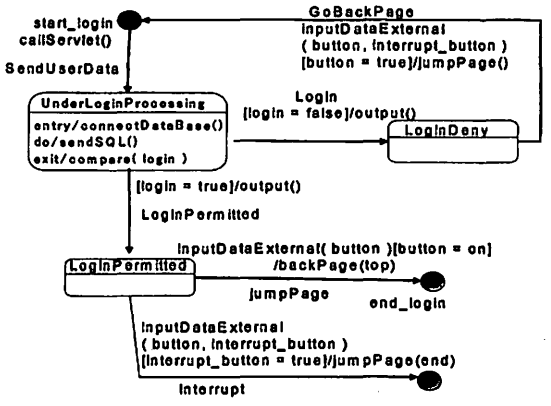

図5　認証ログインステートチャート

(3) 実行可能モデルの検証(xUMLシミュレーション)結果

図5のxUMLに対して，認証される場合，されない場合，途中で終了する場合と3つのケースについて，うまく実行されるか，テストデータを用意してシミュレーションを行った．結果を図6に示す．

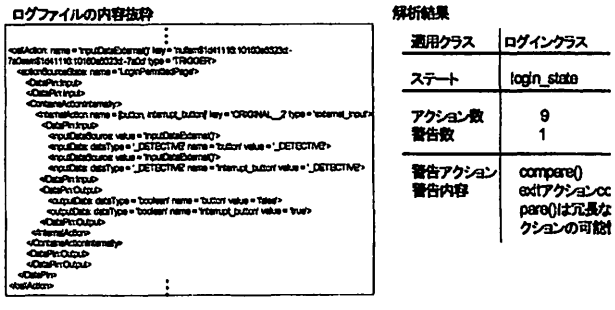

図6　xUMLシミュレーション結果

4.3 評価と考察

(1) BPM

モデル化したシステムのプロセスにシミュレーションを施すことで，システムのプロセスの整合性を確認できた．BPMシミュレーションに関しては，待ち時間や休日をはさんだプロセスなども細かに考慮できるが，ランダムなプロセスの中断(社員が風邪で休むなど)を考慮はできなかった．実際の業務を考えるとランダムな要素を考慮したシミュレーションが必要であると考えられる．

(2) xUMLの設計

ステートチャート図に付加してあるアクションを本方式で提供しているアクションに対応付けるだけなので，ほぼ機械的に作業でき，小規模プログラミングも発生しなかった．しかし，アクションの組み合わせによって振る舞いを表現する本方式においては組み合わせ方や属性の付加の仕方によって目的のアクションができるまでは同じ作業の繰り返しとなる．

(3) xUMLによるシミュレーション

実行シミュレーション機能は作成途中であるが，アクション言語の規定までを行い，プロトタイプでは一部のステートチャートを読み込み，検証することができた．今回の検証はある変数について，ステートチャート図作成前に設定した事項が守られていないことを指摘するものであった．xUMLによる動作確認には変数1つ1つにまで振る舞いの制約を施さなければならないこともわかった．

(4) シミュレーション駆動な開発

本方式においてはBPM，実行可能モデリングともにシミュレーションをシステム開発の中心に据えている．現在は未実装であるが，本開発手法で設計中であるコード生成によらない実装機能は，シミュレーションエンジンを対象システムの実環境で直接活用するものであり，完成すれば，ここまで述べたBPM/実行可能モデリングを連携した本方式においては，コーディングをともなわない，シミュレーションを中心とした開発が確立すると考える．

5. まとめ

xUML設計に小規模プログラミングを発生させない機能と，選択と組み合わせによるアクション設計方式を行う開発手法のプロトタイプを一部構築し，BPM/実行可能モデリング連携手法において，BPMからのシームレスな開発の中で，その効果の一部を確かめた．また，コード生成によらない実装方式を提案した．今後は，本開発手法の早期完成と，本実装方式の組み込みを目指す．

なお、本研究は東京電機大学総合研究所研究Q05-J-03として行ったものである．

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] "Object Management Group", http://www.omg.org/mda

[3] "Action Semantics for the UML", http://www.kabira.com/as/download/ActionSemantics.zip

[4] 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，Vol.45 No.1，pp.3-9 2004

[5] スティーブJメラー・マークJ.バルサー：「ExecutableUML 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，2003

[6] "ARIS", IDS Scheer, http://www.ids-scheer.co.jp

[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・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