

xUML を用いた実行可能モデリングによる情報システム設計手法の試作

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

現在、情報システムの開発において、分析・設計・実装の各工程は分断されているため、工程間でシステムに対する認識のギャップが生じていると考えられる。分析工程においては、いかに組織とシステムを結びつけて管理を行っていくのか、行いやすいシステムにするのかを、考えた上でシステム設計に着手しなければならない。また、開発全体を通して、開発期間は短縮している傾向がある。これらの課題を解決する目的で、本稿では、UML をベースとしてビジネスプロセスモデリングを分析に用い、設計および実装に xUML(eXecutable UML)を用いる情報システム設計手法を提案する。本手法では、分析・設計・実装の各工程に UML(Unified Modeling Language)技術を用いて工程間の分断状況を改善する。分析工程では、ビジネスプロセスモデリングを用いて組織とシステムを関連付けて分析する。設計工程では、UML の拡張技術である xUML を用いてシステムの動作検証を行い実装時の後戻りを未然に防ぐ。また、動作検証に用いたモデルのエンジン部分を実装時に活用することで、開発期間の短縮化を図る。このような本手法を試作し、プロトタイプへと適用し一部評価を行った。

An Information System Design Method with Executable Modeling by using xUML

Tsukasa Kaminishi[†] Syuichi 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

Recently, in the information system development, there are some gaps of recognition to the system between the development processes which are divided to the analysis, the design, and implementation. Moreover, in the analysis process, we have to consider the tie of the organization and the system before the design. Additionally, development period tends to be shortened. In this paper, we propose an information system design method based on UML(Unified Modeling Language) that uses BPM(Business Process Modeling) to analyze and uses xUML(eXecutable UML) to design and implement. In this method, each process of divided analysis, design, and implementation is connected in seamless by using the UML technology. Moreover, the organization and the system are related and analyzed by using BPM. Additionally, in the implementation process, going back to previous process is prevented by xUML simulation which can check current validity in each process. Besides, in the implementation process, we can use the code of the models which are used in xUML simulation to shorten the development period. In this paper, we describe the implementation of a part of the proposed development method and its application to evaluate by the prototype system.

1. はじめに

情報システム開発では、開発に着手する前のシステム分析において、組織が運用するシステムに対して、どのように人員を配置すれば良いのか、現状の組織構成で運用できるシステムなのか、など組織とシステム

の関係を検証する必要がある。しかしながら、現状の開発工程では、分析・設計・実装の各工程が分断されており、分析で検証した成果を次の工程である設計に最大限活用することは難しい。また、このことは設計・実装間でも存在する問題である。設計工程では、現在、広く普及している UML(Unified Modeling Language)を用いるが、

このような背景を受け、筆者らはこれまでに、BPM(Business Process Modeling)^[1]に関する研究を行い、下流工程に xUML による実行可能モデリングを用いる可能性を示してきた^{[2][3]}。本稿では、開発の分析工程に BPM を用いて、組織とシステムを可視化・検証し、設計・実装工程に xUML(executable UML)^{[4][5][6]}を用いて、実行可能なモデリングを行いシステムの動作検証を行う、要求定義から実装までの一貫したシステム開発手法を提案、試作し、評価を行う。

2.1 xUML 実行可能モデリングによる設計手法の流れ

- 組織とシステムの分析に BPM を用い、関係を可視化
- 組織の運用状況をシミュレーションより検証
- 分析工程と設計工程を UML を用いて連結し、シームレスな工程移動を実現

- 本方式の一連の流れを図 1 に示す。

- ① 開発対象となるシステムの仕様を用意する。
- ② 運用する組織構成とシステム化するビジネスプロセスの分析を行うため、組織構成およびビジネスプロセスをモデル化する。作成したモデルは、現状組織構成・プロセスを織り込むため、As-Is（現状）モデルである。
- ③ 作成した As-Is モデルについて、組織が運用可能なシステムかどうか、人員配置は適切かどうかなどを検証するため、As-Is モデルに対してシミュレーションを行い、ボトルネックとなる箇所を割り出す。
- ④ 人員の再配置やビジネスプロセスを修正し、検出されたボトルネックの改善を図る。検出内容によっては、要求仕様にフィードバックを行う。
- ⑤ シミュレーションにおいて、ボトルネックの改善が確認されれば、対象プロセスモデルが To-Be（改善）モデルとなる。
- ⑥ To-Be モデルに対して、どの UML モデルに変換する



か、など条件を加え、ビジネスプロセスを記述したモデルから変換し、UML モデル要素を生成する。

- ⑦ 生成された UML モデル要素をもとに、UML 分析モデルの作成を行う。分析モデルの作成においては、UML モデル要素生成時には、検出されなかったオブジェクトの振る舞いなどについても、追加する。
- ⑧ UML 分析モデルから設計モデルへと詳細化を行う。詳細化の際に、新たにモデル要素を追加することは無い。
- ⑨ 作成した設計レベルの UML を xUML に拡張するため、xUML 用のアクションを設計する。アクションとは、オブジェクトの振る舞いを表したものである。本方式において、xUML 化する対象となる UML モデルはステートチャート図である。
- ⑩ UML に設計したアクションを付加し、xUML が作成される。
- ⑪ xUML シミュレータに xUML を投入し、シミュレーションを行う。シミュレーション結果によっては、UML 設計モデルの再作成、アクションの再設計、または、ビジネスプロセス記述の変更、へと結果をフィードバックする。
- ⑫ xUML シミュレーションによって、実現の妥当性が確認された機能については、xUML モデルの内包するコードを流用し、実装においては、このコードを用いる。

2.2 本設計手法におけるツールの活用方式

図 1 に示した方式は、要求①、BPM 工程部分②～⑤、連携部分⑦⑧、xUML 工程部分⑨～⑪、実装⑫と、大きく 5 つに分けることができる。このうち、要求を除く 4 つの工程においては、ツールを用いた開発を行う。まず、BPM 工程部分については、ビジネスプロセス記述、作成したプロセスのシミュレーション、および UML 要素変換を行う機能を持つ BPM ツール ARIS^{[7][8]}を活用している。次に、連携部分については、UML モデルの詳細記述と XMI(XML Metadata Interchange) 形式による保存が可能なツール PoseidonForUML^[9]を活用している。最後に、xUML 工程部分および実装については、筆者らが開発しているツール(以下、本 xUML ツール)を活用する。以下に、本方式を実現するためにどのようにツールを活用していくのかを、図 1 の流れに沿って述べる。

2.2.1 ARIS を活用した BPM 工程

(1) モデリングガイドライン

ARIS には、BPM 工程を行う機能がいくつかあるが、それらの機能をただ使うだけでは、本方式の特徴とするシームレスな工程移動が難しいために、モデリングの指針となる次のようなガイドラインを設定した。

- モデル化は、要求仕様より、「組織構成」「各組織の持つシステム」「各システムのドメイン」「各ドメインごとのビジネスプロセス」の順に行う。

モデルの段階的詳細化を行うことが目的である。徐々に粒度を小さくすることで、再利用時の利便性の向上や、ドメインの独立性の向上を期待できる。

- ドメイン分割は細かく行い、サブドメインは階層構造で表現する。

ドメインの粒度が粗ければ、複数のドメインに跨るビジネスプロセスが出来てしまい、設計時の負担が大きくなる。そのため、ドメインの粒度を細かくしておき、ビジネスプロセスを 1 つのドメイン、またはサブドメインに収める。

- プロセス記述はあまり長く書かない(1 つの画面で収まるように描く)。

長いプロセス記述は、見辛い上に、設計時に混乱する。例えば、UML 要素はビジネスプロセスから変換し生成するが、変換する部分を選定する際に、ビジネスプロセスのどこからどこまでを 1 つの UML モデルとするのか、考えなければならない。短いプロセス記述であれば、ドメインの特徴から判断し、1 つのプロセスを丸ごと UML 変換することが可能である。

- ビジネスプロセス記述に用いたデータは必ずデータビューで別途記述し関連付けるか、ビジネスプロセス記述内に直接明記する。

システムで扱うデータに関しても、BPM において分析するが、ビジネスプロセス記述に関連付けておくことで、後工程である UML 生成部分において、どのデータを UML 要素とすれば良いのか、設計者は、プロセス記述から選択するだけで済み、例えば、UML クラス図の属性などを 1 から考える必要がなくなる。

(2) ビジネスプロセス記述

ビジネスプロセス記述を表すビジネスプロセスビューは、組織構成を表す組織ビュー、データ構造を表すデータビュー、システムのドメイン構造を表すファンクションツリーの各図を図 2 に示すように関連付けて表現する。

(3) プロセスのシミュレーション

作成したビジネスプロセス記述に対して、仮想的なシミュレーション時間を設定し、プロセスを発生させ、処理時間の合計、次のプロセスが処理を開始するまでの待ち時間、各プロセスの実行状況などの結果が得られる。シミュレーションの実行のためには、予め、各プロセスが処理に要する時間をパラメータとして割り振る必要がある。これらのパラメータは、システムの要求仕様などからも入手するが、これまでの業務処理にかかった時間など、蓄積されたデータが好ましい。

業務ボトルネックの判断については、シミュレーション結果の中から突出したデータを設計者自らがボトルネックであると判断する必要がある。

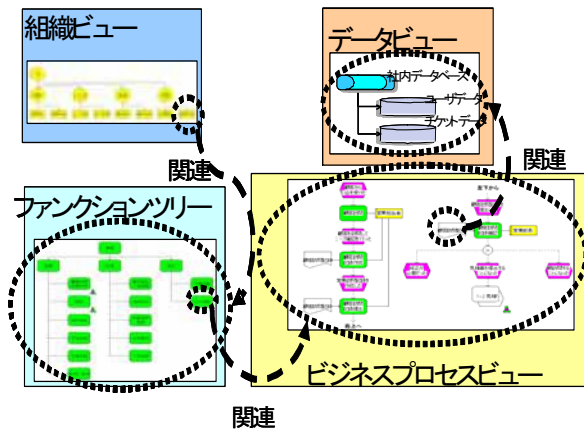


図2 BPMを実現する各図の関連

(4) UML 要素変換

BPM 工程でシミュレーションが終了し、To-Be モデルが完成すれば、次は UML による設計工程である。シームレスな工程移動と効率向上のために、ARIS による UML 要素変換を用いる。変換メカニズムについて述べる。ARIS では、ビジネスプロセス図のモデル要素をもとに UML の各図のモデル要素を一対一対応で生成する。例えば、クラス図であれば、ビジネスプロセス図の処理を表す要素がクラス図のメソッドに対応する、という具合である。

2.2.2 PoseidonForUML を活用した UML モデルの作成

UML 分析モデルの作成においては、ARIS が生成したモデル要素をもとに、UML モデリングツールを用いて、不足部分を補い、UML 分析モデルを完成させる。また、開発中の xUML ツールがモデルを XMI 形式で扱うため、モデルは、XMI 形式を用いる。

2.2.3 本 xUML ツールによる xUML 工程

(1) 本 xUML ツールの機能と特徴

現在も開発中である本 xUML ツールの機能と特徴は次の通りである。

- XMI 形式のデータを読み込むことによる他ツールとの連携

業界標準である XMI 形式を取り込むことによって、XMI 対応のツールとの連携の可能性がある。他のツールと連携することができれば、他の開発手法においても本 xUML ツールを利用することができる。現在は、先に挙げた PoseidonForUML との連携を確認している。

- xUML のアクション設計に工夫

他ツールのように、アクション言語を用いた小規模プログラミングを行うことなく、モデルの振る舞いを定義できる。また、各アクションに属性という概念を設け、同様のアクションでも、属性の違いによって異なる振る舞いをさせ、振る舞いにバリエーションを持たせている。アクション設計については、本項(3)にて詳細説明を行う。

- 3 段階のシミュレーション

シミュレーションの範囲を、1 つの xUML モデル、1 つのドメイン、システム全体と 3 段階に分けることによって、シミュレーションの粒度を 1 つの機能レベルから全機能まで、変更できる。

- シミュレーションで検証した機能についてはコード化し、実装時に活用

実装工程での負担軽減や、モデルに従った機能の実装のために、検証した機能についてはコード化し、活用する。コード化の詳細については、本項(5)にて詳細説明を行う。

(2) 本 xUML ツールの構成と概要

本項(1)で挙げた機能と特徴を実現する本 xUML ツールは、図3の構成を持つ。

アクション言語群において作成したアクションを、外部ツールにおいて UML に付加し、xUML に拡張したものを入力部に読み込む。次に、制御部においてユーザより入力された命令に従って、アクションの解釈実行が行われ、シミュレーションが実施される。シミュレーションに必要なモデルのアクション情報などは、抽出部で読み取る。シミュレーションを行うシミュレーションエンジンでは、リアルタイムで結果を出力部に送り、ユーザに知らせる。また、シミュレーション終了後、累積結果もユーザに返す。シミュレーションの仕組みについては、

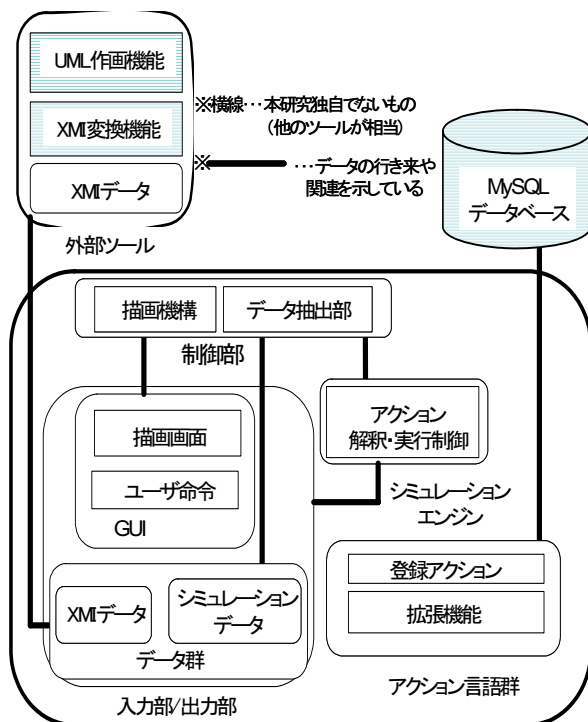


図3 本xUMLツールの構成

本項(4)にて詳細説明を行う。

(3) アクションの生成と連携による拡張

本 xUML ツールにおいて、アクション設計は、図 4 に示すように、既存アクションの連携による拡張や属性の付加によってモデルの複雑な振る舞いを定義される。

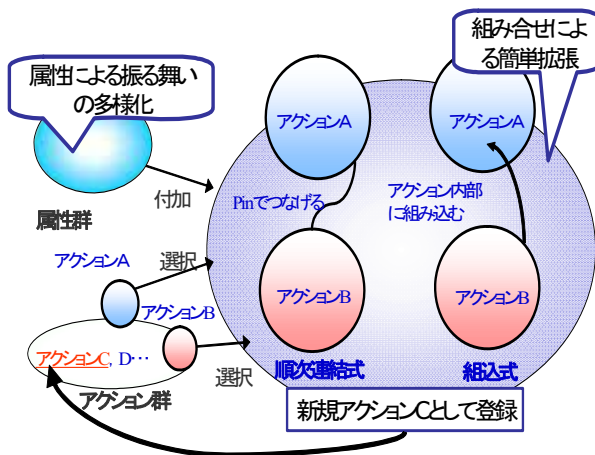


図4 連携によるアクション生成・拡張方式
アクションの連携方法は順次連結式と組込式の2通りがあり、どちらを選ぶかによっても振る舞いは変化する。また、既存アクションとしては図5のようなものを規定している。図5に示すアクションは、

OMG(Object Management Group)^[10]のアクションセマンティクス^[11]や UML1.4 仕様書、および本方式においてプリミティブと規定されているアクションである。

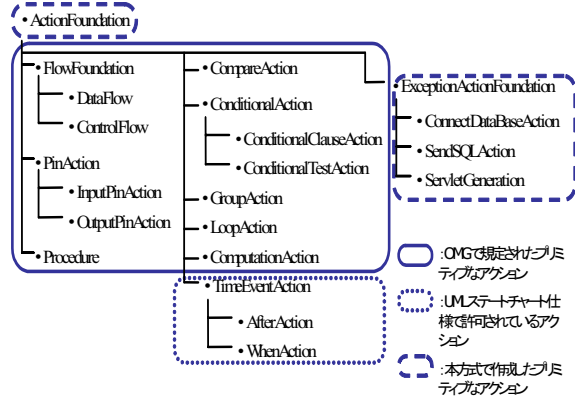


図5 既定アクション群

(4) xUML シミュレータの仕組み

本 xUML ツールの対象となる xUML はステートチャート図である。ステートチャート図は、フローが矢印によって明示されており、本 xUML ツールのシミュレータも読み込んだステートチャート図のフローを辿り、シミュレートしていく。動作の概要を図6に示す。

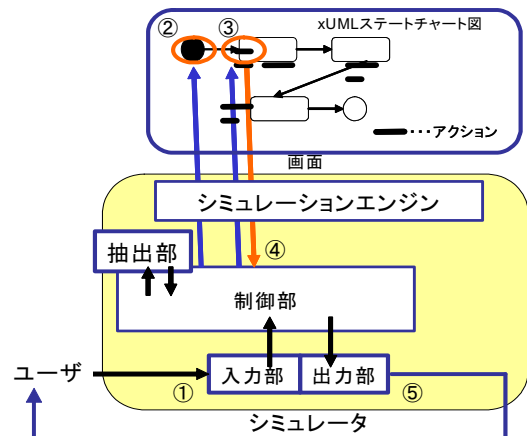


図6 xUML シミュレータの動作

図6の流れは、1つのxUMLモデルに対するシミュレータの動作である。動作の流れは次の通りである。

- ① ユーザより入力部にxUMLモデルとテストデータが入力される。入力されたxUMLモデルからは、制御部の抽出部によってモデルのフロー情報とアクション情報が抽出される。
- ② シミュレーションエンジンがxUMLモデル中より先頭状態を発見し、シミュレーションを開始する。
- ③ シミュレーションエンジンが、ステートチャート図

のフローを辿り、次のステートを確認していく。その中で、アクションが見つかる。

- ④ シミュレーションエンジンは、制御部にアクション発見を通知し、アクションを解釈・実行する。
- ⑤ 実行結果は、出力部からユーザへと出力される。ユーザは、システムの動作が仕様を満足しているか、シミュレータに出力されたテストデータに対する結果を検討する。

1つのxUMLモデルのシミュレーションでは、上記③、④を繰り返し、シミュレーションエンジンが終了ステートを見つけると終了する。

1つのドメインのシミュレーションは、ドメインを持つすべてのxUMLモデルのシミュレーションを行った上で終了する。複数のxUMLモデルの連携には、「リンク」という概念を用いて、リンク元からリンク先のモデルへシミュレーションの対象を変更させることで、可能にしている。

同様に、複数のドメインを持つ全システムに対してのシミュレーションにもリンクを用いて可能としている。

(5) 検証したモデルのコード化および実装時の活用

xUMLシミュレーションの対象はモデルであり、検証した機能を実現するコードはモデルが保持している。モデルの内包する機能をコード化することができれば、図7に示すように、シミュレータと実機のエンジン部分の相似点を利用し、実機のエンジン部分に活用することができる。その結果、コード化したものと他のコードの接続点、および実行環境(データベース、サーバーなど)との間の接続点の実装のみによって実装工程を終えることができると考えられる。本機能については、現在設計中である。

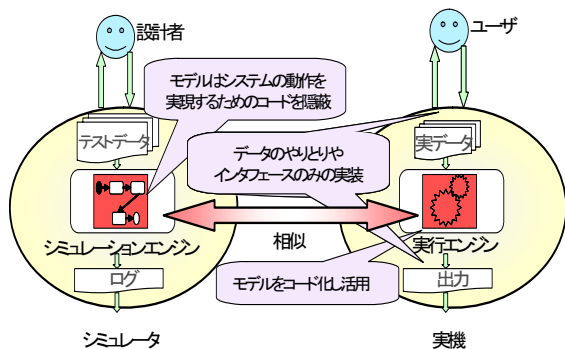


図7 シミュレーションエンジンと実機の比較

2.3 xUML 実行可能モデリングによる設計手法の構築

本設計手法を表1に示す各ツールを用いて構築した。

表1 本設計手法の構築に用いたツール

ツール	用途
WindowsXP Pro Ver2002 SP2	OS
ARIS Toolset	組織・プロセス設計に関するモデリング ビジネスプロセスシミュレーション
ARIS Business Server	ARIS用アプリケーションサーバ
ARIS Database Server	ARIS用データベースサーバ
PoseidonForUML Community Edition Ver2.5.1	UMLモデリング XMI変換
xUMLシミュレータ (開発:Java1.5.0_04)	xUMLシミュレーション アクション設計
MySQL Ver3.23.49	xUMLシミュレータ用データベースサーバ
Tomcat Ver5.0	サーブレットコンテナ

3. プロトタイプへの適用結果

3.1 対象システム

本方式を図8のような構成を持つ情報システムの開発に適用した。

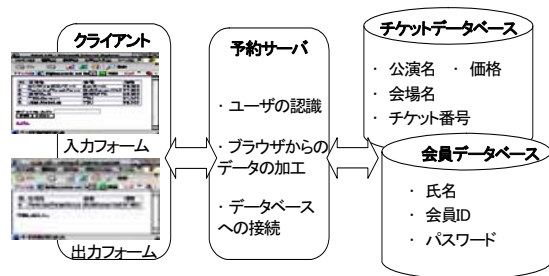


図8 チケット予約システムの構成

このシステムは、ある会社のIT事業部で運用予定のシステムである。システムは、応対専門スタッフによる電話による予約と、ブラウザを用いたインターネット予約が可能である。このシステムの利用には、ユーザによる事前の会員登録が必要である。会員パスワードは会員ごとに発行される。

3.2 適用範囲

本方式における開発手法は未完成であるため、適用範囲は、図1の①の要求仕様の用意からxUMLによる⑩のドメイン間シミュレートの実用までとした。

3.3 適用結果

(1) BPM 工程部分

本方式による BPM の結果, シミュレーションにて電話予約・インターネット予約の両方の入力受付処理にてボトルネックが生じたため, 入力受付処理を強化し, 電話予約による応対を自動音声ガイダンスに変更(スタッフ人数削減)し, ボトルネックを改善した. 改善後の To-Be モデルを図 9 に示す. なお, ビジネスプロセスビューは, 作成した 5 つのうちの 1 つ, 会員照合処理のビューである.

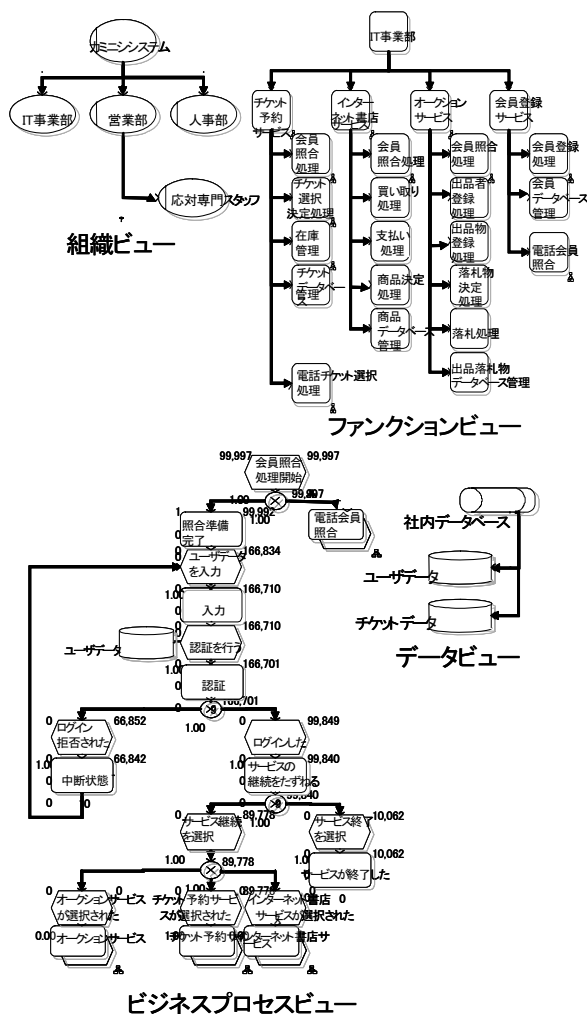


図 9 BPM 結果

(2)UML 連携部分

図 9 で示したビジネスプロセスビューに対して, ステートチャート図への UML 変換をし, 補足をしたものを図 10 に示す.

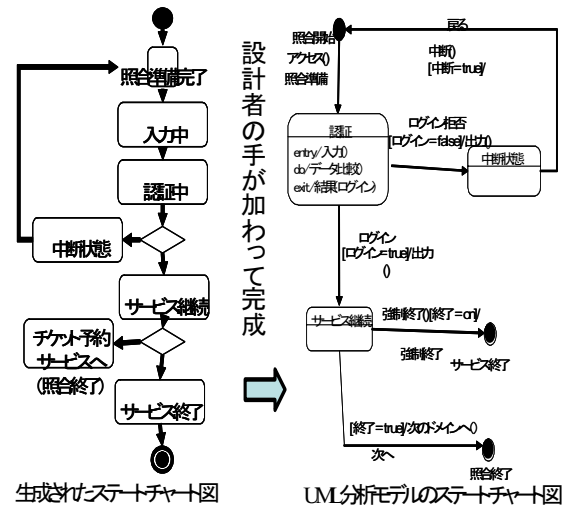


図 10 UML モデル

図 10 では, オブジェクトの振る舞いまでを作成している. この振る舞いが後工程での xUML アクション設計の指針となる.

(3) xUML 工程部分

作成した UML 分析モデルを設計モデルに詳細化し, アクションを設計・追加し, xUML への拡張を行った. 図 11 に作成した設計モデル, および xUML を示す.

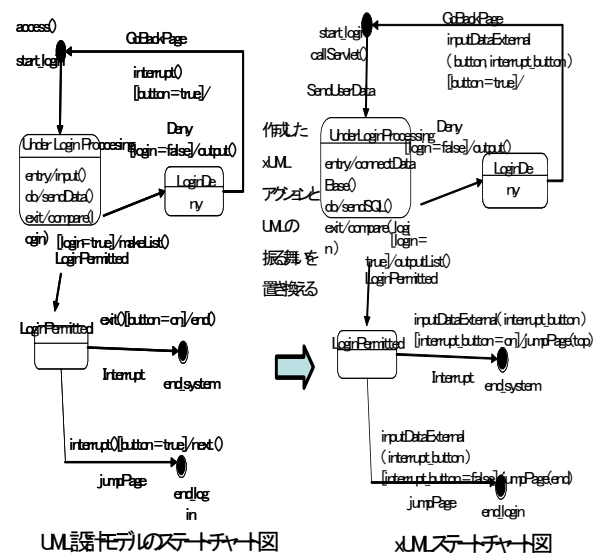


図 11 xUML モデルへの拡張

xUML シミュレーションのテストデータは, チケットデータ, およびユーザーデータとし, 実際にステートチャート図のフローに従って処理が実行されるかどうか, 確

かめた．図 12 に結果とログを示す．

実行ログの結果	
適用クラス	ログインクラス 予約クラス
実行ステート	login_state, reserve_state
アクション数	17
警告数	1
警告アクション	compare()
警告内容	exitアクションcompare()は冗長なアクションの可能性

複数のドメインでのシミュレーションを確認した

BPM→UML変換の際の設計者の冗長な処理を、実装前に発見することができた。ただし、シミュレーションは適用範囲をすべて実行できた

図 12 xUML シミュレーションの結果

5. 評価と考察

(1) BPM 工程部分

シミュレーションを行うことで，組織とシステムの人員配置や運用のバランスを確認できたが，ランダムなプロセスの中断(社員が風邪で休むなど)の考慮が必要である．ビジネスプロセスモデルの作成から UML の作成に関しては，モデリングガイドラインを用いることにより，UML 設計時に，1 からモデル要素を考えることはなく，工程移動はスムーズであったといえる．

(2) UML 連携部分

UML 分析モデルを完成させ，振る舞いを作成する工程でもあるため，もっとも設計者の誤りが混入しやすい工程であることが分かった．振る舞いの設定から適切でないと，UML ルールに準拠できなくなる．今回の適用では，冗長な振る舞いが混入した．

(3) xUML 工程部分

本方式により，アクション設計において，小規模プログラミングは発生しなかった．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ことも判明した．しかしながら，UML 連携部分において混入した冗長な振る舞いを実装前に検出することができたことから，一部妥当性を確認できたといえる．

xUML シミュレーションは構築途中であるが，プロトタイプではステートチャートを読み込み，複数ドメイン間シミュレーションを検証することができた．

6. まとめ

BPM により組織とシステムを可視化・検証し，xUML

によりシステムの動作検証を行う要求定義から実装までの一貫したシステム開発手法を提案し，BPM によるビジネスプロセスボトルネックの解消，実行可能モデリングによる一部モデルの妥当性確認を行った．また，モデルをコード化し活用する実装方式を提案した．

今後は，xUML シミュレータ，およびコード化機能の早期完成を行う．

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．

文 献

- [1] 戸田保一，"ビジネスプロセスモデリング"，飯島淳一（編），日科技連，2000
- [2] 小島 義幸、小泉 寿男，"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
- [3] 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
- [4] ステイブ 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
- [5] 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
- [6] 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．
- [7] "ARIS"，IDS Scheer，<http://www.ids-scheer.co.jp/>
- [8] ハインリヒ・ザイドルマイヤー，堀内正博訳，"ARIS によるビジネスプロセスモデリング"，ビー・エヌ・エヌ新社，2004
- [9] "PoseidonForUML"，Gentleware，<http://gentleware.com/>
- [10] "Object Management Group"，<http://www.omg.org/>
- [11] "Action Semantics for the UML"，<http://www.kabira.com/as/download/ActionSemantics.zip>