

ビジネスプロセスから IT 実装への連携

Linking Business Processes to IT Implementation

篠原 圭

要約 多くの国内企業が抱える IT 課題の一つとして「ビジネスと IT の乖離」がある。代表的な要因として「システムが業務要求を満たしきれていない」ことや「システムが複雑化し、ビジネスの変化に迅速に対応できない」ことが挙げられる。

Unisys 3D Blueprinting では、URUP (Unisys Rational Unified Process) という RUP (Rational Unified Process) を拡張したソフトウェア開発方法論を採用している。URUP では業務領域からシステム領域までを段階的にモデリングし、さらにツールを用いて業務領域とシステム領域のモデルを連動させることができるため、業務要求とシステム間の乖離の防止が期待できる。また、モデリングの際に共通部分を抽出することで、業務ロジックのコンポーネント化を促進し、ビジネスの変化に対応しやすい柔軟なシステムの構築を可能としている。

本稿では、URUP におけるモデリング手法に焦点をあて、その特徴、期待される効果を述べる。

Abstract One of the IT problems faced by many enterprises in Japan is the gap between business and IT. Typical contributing factors for this are that the system does not meet the business requirements, and that the system has become too complex to quickly adapt to changes in the business, etc.

Unisys 3D Blueprinting has adopted a software development methodology called URUP (Unisys Rational Unified Process), an extended version of RUP (Rational Unified Process). Using URUP, the business area and the system area are modeled in phases. Tools are used to link the both models, enabling the prevention of the gap between business requirements and the system. Also, common sections can be extracted during the modeling process, which helps to promote the component-based creation of business logic, and enables the development of an agile system that can easily adapt to business changes.

This paper focuses on the URUP modeling methodology, and discusses its features and expected results.

1. はじめに

多くの国内企業が抱える IT 課題の一つとして「ビジネスと IT の乖離」がある。代表的な要因として「システムが業務要求を満たしきれていない」ことや「システムが複雑化し、ビジネスの変化に迅速に対応できない」ことが挙げられる。業務要求を満たしていないとシステムは使われず、結果的にシステム投資が無駄になってしまう。また、システムが複雑で変更や拡張をしにくい場合、ビジネスの変化に対応させる改修に時間とコストがかかり、ビジネスチャンスの喪失につながる。

Unisys 3D Blueprinting では、業務要求をソフトウェア開発につなげるために、RUP^{*1} のモデリング手法を採用し、それを支援するツールを提供することにより、モデル駆動型開発^{*2}を

実現している。

本稿では、最初に Unisys 3D Blueprinting におけるソフトウェア開発の位置づけとその構成要素を解説する。その後、モデリング手法の詳細を解説し、最後に期待される効果を述べる。

2. Unisys 3D Blueprinting におけるソフトウェア開発

Unisys 3D Blueprinting とは、企業のビジネス活動構造全体を四つの層（ビジネス戦略、業務プロセス、情報機能、情報基盤）に区分けし、立体的にあらわすことで企業の可視化を支援する知財活用型のシステム化アプローチである（図1）。Unisys 3D Blueprinting の各層は方法論（開発プロセス）、ブループリント、ツールの三つで構成される。

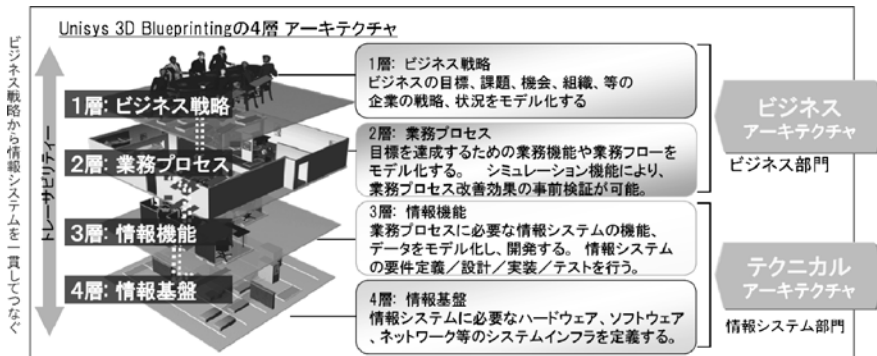


図1 Unisys 3D Blueprinting の4層構造

上記4層の中でソフトウェア開発は3層に該当する。開発プロセスにはRUPをベースとした米国Unisys社の開発方法論であるURUPを用いる。ブループリントとはUnisys 3D Blueprintingの成果物一式であり、参照モデルとして業種・業界領域毎に蓄積、再利用される業務知財である。ツールとしてはIBM社のRationalツールを採用しており、これを補足する形でいくつかの米国Unisys社製のツールを提供している。

2.1 開発プロセス

URUPとは、RUPをベースにカスタマイズを加えたソフトウェア開発方法論である。反復型開発やモデル駆動型開発といったRUPの特徴を持ちつつ、運用・保守フェーズの追加やCMMI^{*3}レベル3への対応などを行っている点が特徴である。URUPの詳細については、ユニシス技報88号の掲載論文「国際競争時代のコスト構造改革と需要拡大を支える航空基幹システムの世代交代「AirCore」^[1]を参照のこと。

2.2 ブループリント

3層のブループリントは23種類の成果物で構成される。成果物にはユースケースシナリオなど、WordやExcelで文書として記述する成果物も含まれるが、URUPはモデル駆動型開発を採用しているため、UML^{*4}で記述するモデルが主要な成果物である。代表的なモデル成果物の例を図2に示す。

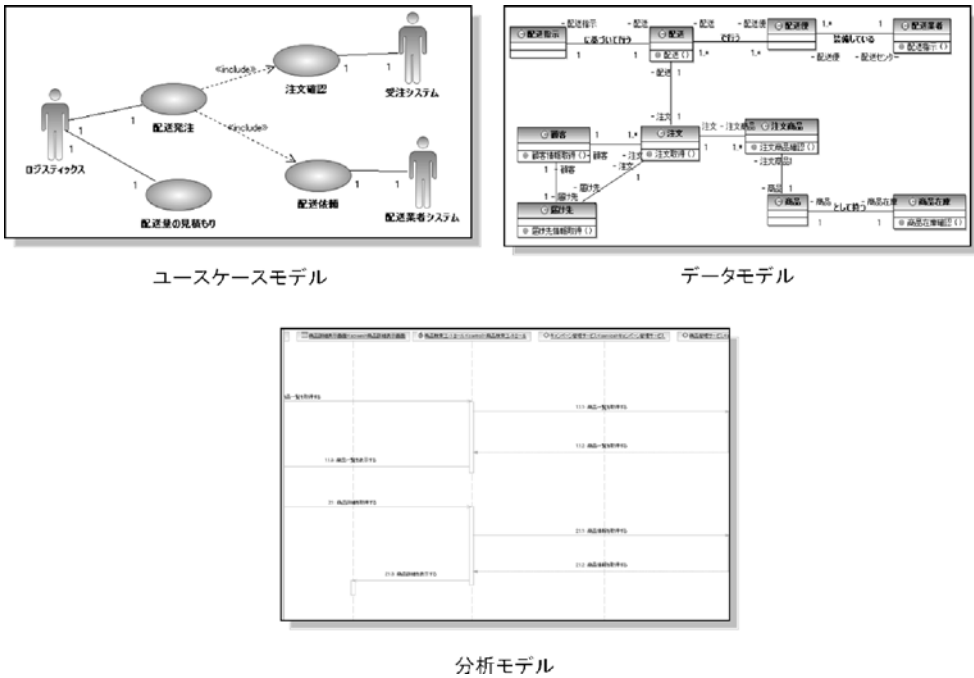


図2 3層の代表的なUML成果物の例

2.3 ツールセット

URUPではRationalツールを中心に、様々なツールを使用して開発を進める。URUPの主要なツールの構成を図3に示す。

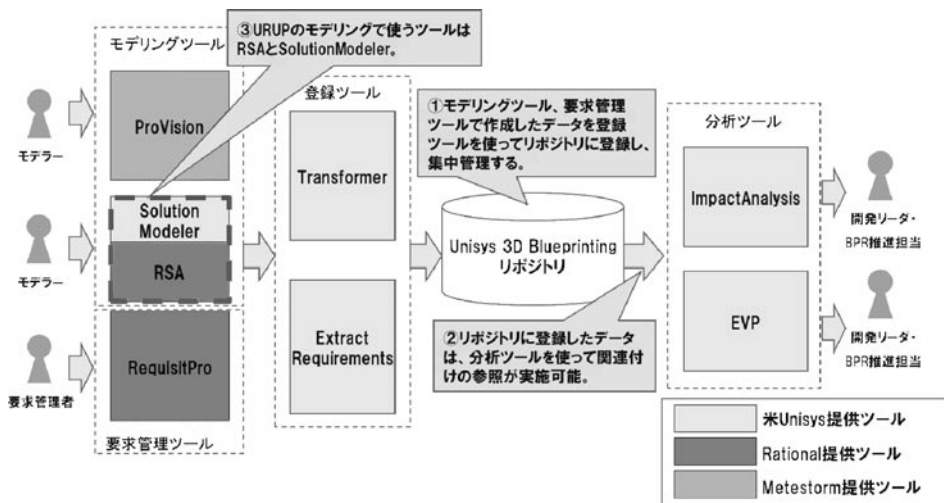


図3 URUPにおける主要なツールの構成

中心にある Unisis 3D Blueprinting リポジトリとは、Rational ツールや ProVision で作成したモデルデータを集中管理するデータベースである (図3-①)。成果物の情報を一箇所で管理することで、異なるツールで作成したモデル同士の関連付けやその関連付けの表示を行うこ

とができる (図3-②)。このような仕組みにより、ある業務プロセスのモデルに変更が加えられた際、関連付けられているシステムのモデルを確認することで、変更による影響範囲を分析することが可能となる。

3層におけるモデル作成は、IBM社製のツールであるRSA (Rational Software Architect) に米国 Unisys社製のツール Solution Modeler をプラグインとして組み込んだものを利用する (図3-③)。RSAはEclipseベースの統合開発環境であり、モデリングや実装、モデル/ソースコード間の自動変換機能、アーキテクチャの構造分析機能など、多様な機能を備えている。Solution Modelerは、モデル駆動型開発を実現するRSAのプラグインである。モデルを次の工程のモデルへ変換する機能を備えており、2層で作成したProVisionのモデルから3層で作成したRSAのモデルへ変換する機能も提供している。この機能により、モデル作成の生産性向上、品質の向上が期待できる。

3. URUPにおけるモデリング手法

URUPにおけるモデリングの流れを図4に示す。

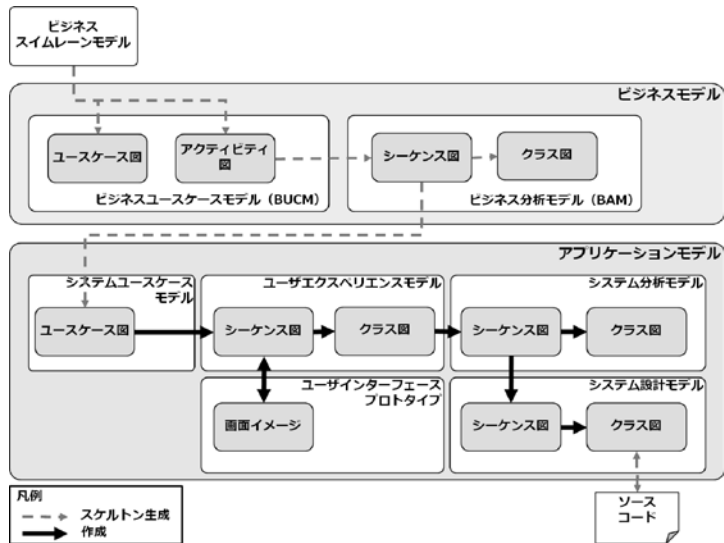


図4 URUPにおけるモデリングの流れ

URUPで作成する主要なモデルは、ビジネスモデルとアプリケーションモデルの大きく二種類に分けられる。ビジネスモデルは現状の業務を分析し、システム化対象範囲を明確にするために作成する。ビジネスモデルはさらに、ビジネスユースケースモデル (業務機能の特定)、ビジネス分析モデル (業務機能の詳細化、システム化候補の抽出) の二種類に分けられる。アプリケーションモデルは、システムの機能や構造を明確にするために作成する。アプリケーションモデルは、システムユースケースモデル (システム機能の特定)、ユーザエクスペリエンスモデル (画面遷移の定義)、ユーザインターフェースプロトタイプ (画面イメージの定義)、システム分析モデル (プラットフォームに依存しないシステム内部構造の定義)、システム設計モデル (プラットフォームに依存したシステム内部構造の定義) の五種類に分けられる。URUPでは上記のモデルを順を追って作成していく。具体的には、ビジネスユースケースモ

デルを基にビジネス分析モデルを作成し、次に、作成されたビジネス分析モデルからシステムユースケースモデルを作成する。このように、作成したモデルを基に次のモデルを作成していく。

モデルの作成には Solution Modeler の機能を利用し、元のモデルから次のモデルのスケルトンを自動生成することができる。ポイントは、Unisys 3D Blueprinting の 2 層の成果物であるビジネススイムレーンモデルを利用して 3 層のビジネスユースケースモデルを自動生成できる点、ビジネス分析モデルを利用してシステムユースケースモデルを自動生成できる点にある。機械的な変換であるため、上位モデルの情報を漏れなく正確にビジネスユースケースモデルの業務機能に、システムユースケースモデルのシステム機能に反映することができる。

このように、ツールによるモデル自動生成機能を使いながらモデルを上流から下流へと段階的に作成し、それぞれのモデル間の整合性を逐次確認していくことで、業務要件を漏らさずシステムに反映することができる。

本章では、各モデルの作成手法の詳細を述べる。

3.1 ビジネスユースケースモデルの作成

ビジネスユースケースモデルとは、業務の目的や内容、業務の流れを表現したモデルである。業務内/外の役割分担と提供するサービスを明確化するために作成する。このモデルはユースケース図とアクティビティ図で構成され、ある業務が提供するサービスをビジネスユースケースとしてユースケース図に記述し、ユースケース図に記載された業務の流れをワークフローとしてアクティビティ図で記述する。ビジネスユースケースモデルの作成手順を図 5 に示す。

まず、ProVision で作成したビジネススイムレーンモデルの内容を Solution Modeler の機能を使い RSA に取り込む。すると、ビジネスユースケースモデルを構成するユースケース図とアクティビティ図が、ビジネススイムレーンモデル単位で自動的に生成される (図 5-①)。ユースケース図は、ビジネススイムレーンモデル内の組織/ロールなどのレーン情報がアクタとして、ビジネススイムレーンモデルがビジネスユースケースとして作成される。アクティビティ図については、ビジネススイムレーンモデルのワークフローがそのままアクティビティ図となって出力される。

次に、この業務を要求する外部のユーザと内部で業務を遂行するロールを区別する。ビジネスユースケースモデルで記述されるアクタは、ビジネスワーカーとビジネスアクタの二種類となる。ビジネスワーカーは対象業務を遂行する業務内の人物または役割を示し、ビジネスアクタは反対に業務外の人物または役割を示す。

図 5 の例では、「顧客」は Web サイトで商品を注文する一般顧客であり、対象業務外部の人物であるため「ビジネスアクタ」として表現される。一方、「販売管理担当者」は Web サイトを運営する業務内部の人物であるため「ビジネスワーカー」として表現される。Solution Modeler の機能を使って自動生成されたビジネスユースケースモデルにおけるアクタは、全てビジネスアクタとして生成されるため、Solution Modeler の機能を利用して、適宜ビジネスワーカーへの変換を行う必要がある (図 5-②)。

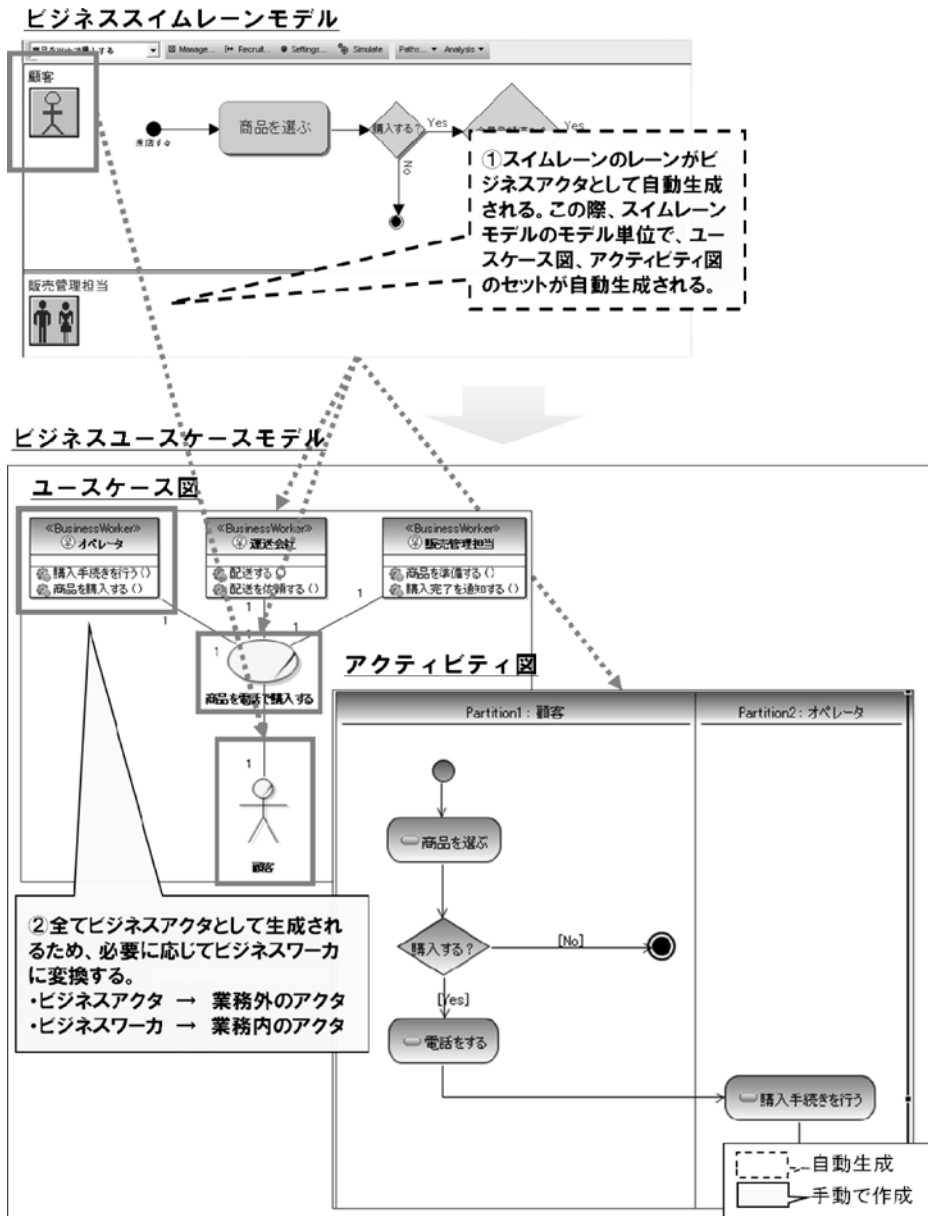


図5 ビジネススイムレーンモデルからビジネスユースケースモデルを作成

3.2 ビジネス分析モデルの作成

ビジネス分析モデルは、ビジネスユースケースがどのように実現されるかをシーケンス図とクラス図で示したモデルである。システム化候補を抽出し、業務とシステムの境界を明確にすることがこのモデルを作成する目的である。ビジネス分析モデルの作成手順を図6に示す。

ビジネス分析モデルの作成では、先に作成したビジネスユースケースモデルを利用する。最初に Solution Modeler の機能を使い、ビジネスユースケースモデルのアクティビティ図からビジネス分析モデルのシーケンス図のスケルトンを生成する。アクティビティ図のレーンがシーケンス図のライフラインに相当し、またアクティビティはシーケンス図の同期メッセージと

して生成される (図6-①). スケルトンが生成されたら, 対象業務がどのように実現されるか分析し, シーケンス図内にメッセージを追加していく. またシステム化対象候補を追加し, そのシステムへの業務指示をメッセージとして追加する (図6-②). 図6の例で言うと, システム化対象候補であるショッピングサイトシステムをライフラインとしてシーケンス図に追加し, ショッピングサイトシステムが担う業務 (ここでは「商品の購入手続きをする」) をショッピングサイトシステムへのメッセージとして追加する. シーケンス図が完成したら Solution Modeler の機能を使ってシーケンス図からクラス図を自動生成する (図6-③).

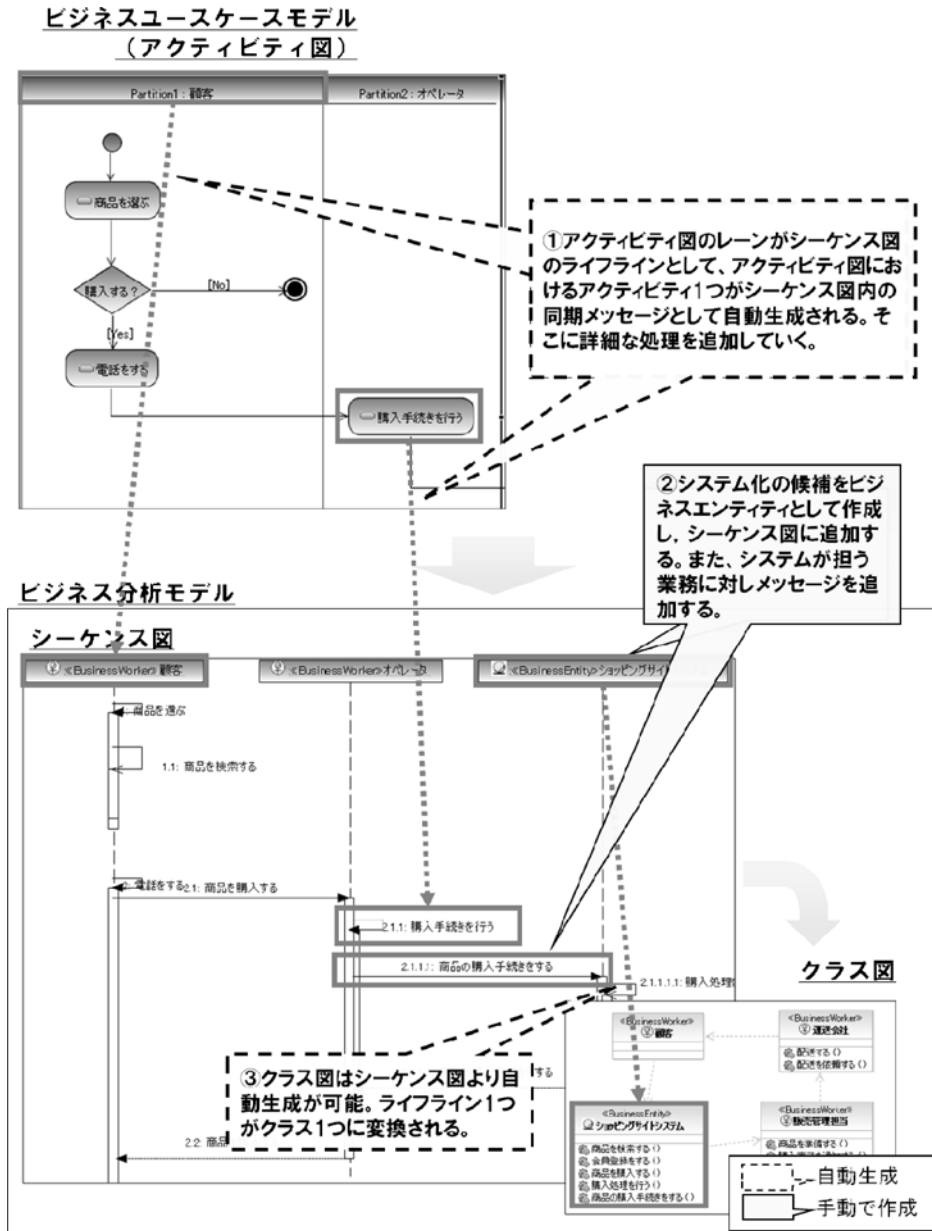


図6 ビジネスユースケースモデルからビジネス分析モデルを作成

3.3 システムユースケースモデルの作成

システムユースケースモデルは、開発対象となるシステムの機能をユースケース図で表現したモデルである。システムの機能とその利用者を明確にするのが主な目的である。システムユースケースモデルの作成手順を図7に示す。

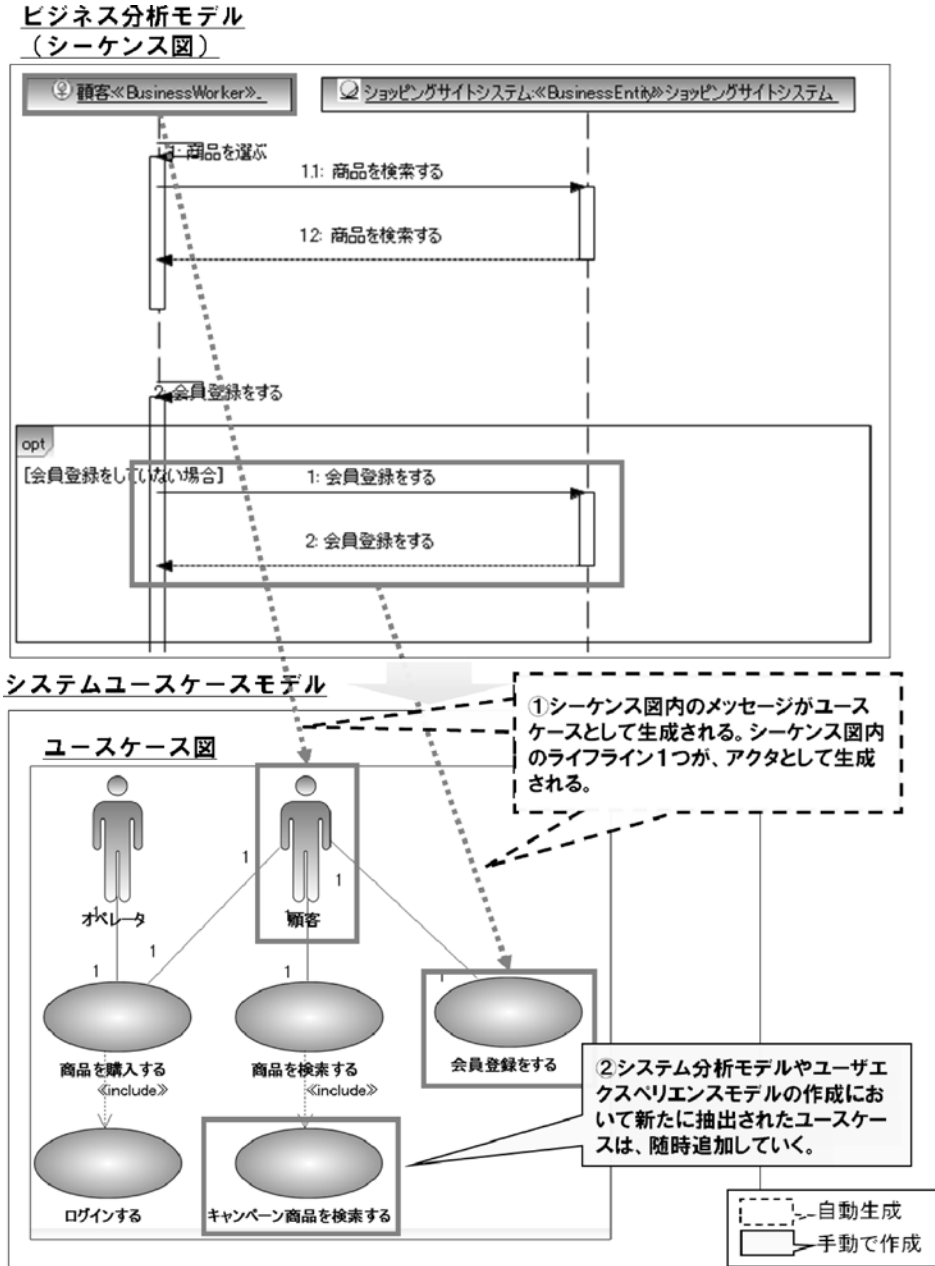


図7 ビジネス分析モデルからシステムユースケースモデルを作成

システムユースケースモデルの作成には、ビジネス分析モデルを利用する。Solution Modeler は、システム化候補に対する業務指示であるメッセージをユースケースとして、またシステム

候補に業務指示を行うビジネスワークをアクタとしてユースケース図を自動生成する (図7-①)。その後、共通化可能なユースケースの抽出、ユースケースのマージ、システム分析モデル作成の過程で新たに発見されたユースケースの追加など、必要に応じて修正を行う (図7-②)。

3.4 ユーザエクスペリエンスモデルの作成

ユーザエクスペリエンスモデルとは、ユースケース内で使用する画面の遷移をシーケンス図で表現したモデルである。ユーザとウォークスルーを行うことで、ユーザが望む機能を漏れなくユースケースに反映するための手助けをすることが目的である。ユーザエクスペリエンスモデルの作成手順を図8に示す。

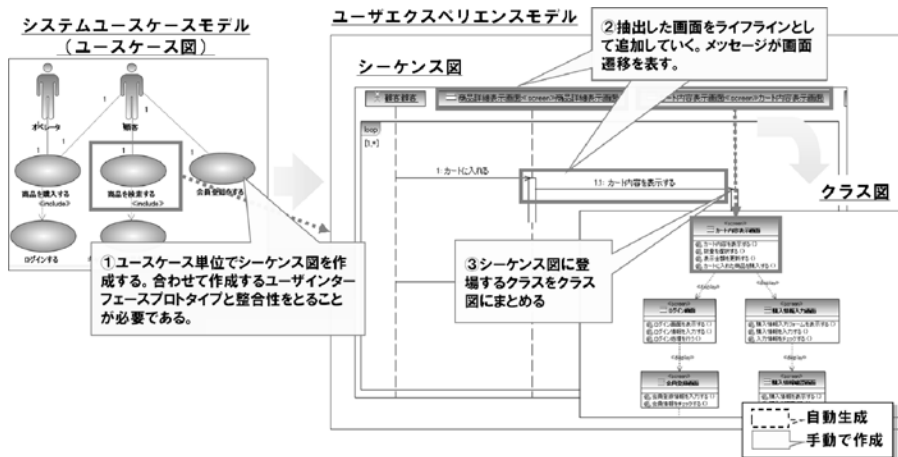


図8 ユーザエクスペリエンスモデルの作成

ユーザエクスペリエンスモデルのシーケンス図は、ユースケース単位で作成する (図8-①)。アクタと画面クラス (ユーザが操作する画面) をライフラインとして配置し、非同期メッセージで画面の遷移を表す (図8-②)。シーケンス図が完成したら、シーケンス図にライフラインとして追加した画面クラスをクラス図にまとめる (図8-③)。

3.5 ユーザインターフェースプロトタイプ作成

ユーザインターフェースプロトタイプとは、システムの画面イメージである。Web アプリケーションであれば、HTMLで作成されたモックアップがこれに該当する。この成果物の作成目的はユーザエクスペリエンスモデルと同様であり、アプリケーションの最終的な画面を決定することではない点に注意が必要である。

3.6 システム分析モデルの作成

システム分析モデルとは、システムユースケースモデルで明確にしたシステム機能の実現方法を定義したモデルである。シーケンス図でシステム内部の動作を定義し、クラス図でシステム内部の構造を定義する。システム分析モデルは、システムが稼働するプラットフォーム (OS や言語) に依存しない形で作成する (プラットフォームに依存したモデルはこの後作成するシ

システム設計モデルで表現される)。これにより、システムのプラットフォームが変更された場合にも活用することができ、再構築の際の開発コストの削減につなげることができる。

システム分析モデルを作成する際、各ユースケース共通に利用できる部分を振舞いとデータに着目して抽出し、共通コンポーネントとして切り出すことが重要である。これにより重複した機能の設計、実装を防ぐことができるため、開発コストの削減、保守性の向上が期待できる。この点もシステム分析モデルの大きな特徴の一つである。システム分析モデルの作成手順を図9に示す。

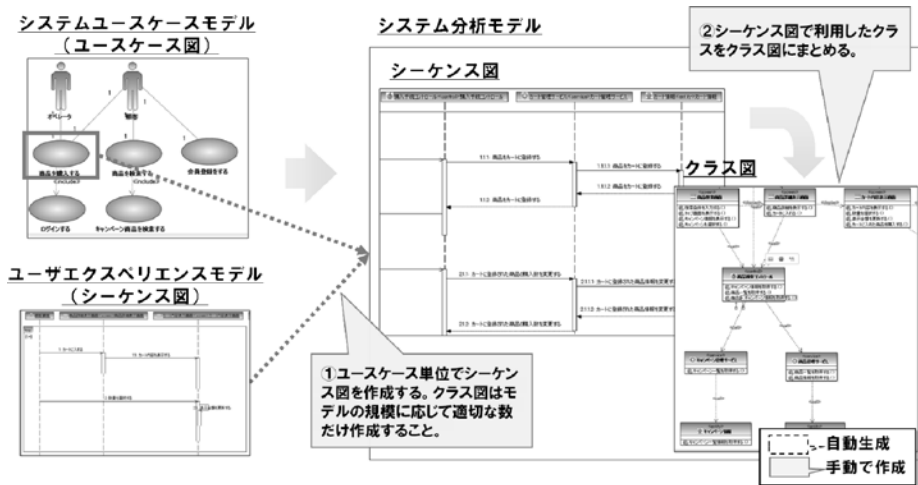


図9 システムユースケースからシステム分析モデルを作成

ユーザエクスペリエンスモデルと同様、ユースケース単位でシーケンス図を作成する (図9-①)。シーケンス図には、システムを利用するユーザをアクタとして追加し、画面、コントロール (画面遷移、サービスの呼び出しを行うクラス)、サービス (業務ロジックを持つクラス)、エンティティ (業務データを表すクラス) をシステムの内部構造を表すクラスとして追加する。クラス間の関連はメッセージで表す。クラス図は、シーケンス図にて利用したクラスをまとめることで作成する (図9-②)。

3.7 システム設計モデルの作成とソースコードの作成

システム設計モデルはシステム分析モデルをプラットフォームに依存した形で表現したモデルである。システム分析モデルと同様にシーケンス図とクラス図で表現される。

また、RSAの機能を用いて、システム設計モデルからソースコードのスケルトンを作成することができる (ただし、C#のソースコードを作成するためには、別製品が必要となる)。作成されたスケルトンに対しプログラマーが実装コードを追加していき、開発を進めていく。ソースコードで変更が加えられた場合、ソースコードからUMLへ変更を反映するリバースエンジニアリングを行うことで、ソースコードとシステム設計モデルの同期をとることが可能である。

4. 期待される効果

3章では URUP におけるモデリング手法を解説したが、この手法を使用することにより期待される効果について以下で述べる。

(1) 業務要求とシステム間の乖離の防止

本手法を用いることで、業務要求とシステム間の乖離の防止が期待できる。2層で作成したビジネススイムレーンモデルを基に3層のビジネスユースケースモデルを、ビジネス分析モデルからシステムユースケースモデルをツールの機能を使って自動的に、漏れなく作成する。その上で、ビジネス領域のモデルからシステム領域のモデルを段階的に作成し、ビジネスとシステムのモデルの整合性を抽象度の高いレベルで確認していくことで、業務要求がシステムの機能まで漏れることなく伝達できる。

(2) 変更に強い柔軟なシステムの実現

二つめの効果は、変更に強い柔軟なシステムの実現である。システム分析モデルを作成する際、共通部分を抽出しまとめていく作業を実施する。この段階で共通化の検討を行うことにより、業務ロジックのコンポーネント化が促進され、ロジックの変更を局所化することができる。

5. おわりに

本稿では、Unisys 3D Blueprinting でのソフトウェア開発における URUP のモデリング手法について紹介した。

日本ユニシスでは、業務ソリューションや社内システムに対して本手法の試行適用を始めたところである。今後の適用拡大により、ビジネス要求に合わせたシステム、変更に強く保守性が高いシステムを提供できると考えている。また、モデル成果物の知財化/再利用を促進し、システム開発の品質向上と顧客満足度の更なる向上に寄与したい。

最後に、本稿の執筆に協力頂いた関係各位に深く感謝する。

-
- * 1 RUP (Rational Unified Process) とは、IBM 社が提唱するオブジェクト指向型のソフトウェア開発プロセスである。ウォーターフォール型の開発プロセスと異なり、反復型開発を採用している点が特徴である。
 - * 2 モデル駆動型開発とは、ソフトウェア開発手法の一つである。抽象度の高いモデルから低いモデルまでを段階的に作成し、ソフトウェアを構築する手法である。
 - * 3 CMMI (Capability Maturity Model Integration) とは、ソフトウェア開発を実施する組織の成熟度を定量的に表す指標である。カーネギーメロン大学のソフトウェア工学研究所により開発された。
 - * 4 UML (Unified Modeling Language) とは、オブジェクト指向型ソフトウェア開発における、システムの構造や振舞いを記述するための記法である。OMG により策定された。

参考文献 [1] 佐藤覚, 小山田和人, 平松敦郎「国際競争時代のコスト構造改革と需要拡大を支える航空基幹システムの世代交代「AirCore」」, ユニシス技報, 日本ユニシス, Vol.25 No.4 通巻 88 号, 2006 年 2 月

執筆者紹介 篠原 圭 (Kei Shinohara)

2002年日本ユニシス(株)入社。Javaによるミドルウェア開発、ICタグを利用したシステム開発を経て、現在、知財活用センターに所属。