

顧客参加促進を目的とした 画面遷移主導による要求抽出支援ツール

寺中 慎介, 上田 賀一

茨城大学 工学部 情報工学科

システム開発の問題点として、重要な要求が出されていない状況や要件変更・仕様変更が頻発しプロジェクトの失敗を誘発してしまうというものが挙げられる。本研究では顧客と開発者間のコミュニケーション不足が原因であると判断し、この原因を解決するために、顧客が積極的に参加でき、抽出した要求情報を共有できる状況を作り出すことを目的とした支援ツールを開発した。顧客が積極的に開発に参加できる環境として画面遷移主導で顧客の要求を抽出することと、要求情報を共有できる環境として画面遷移から導出されるUML出力をサポートする。本研究によりUMLをもとに開発者側は顧客から新たな要求を引き出すといったものや画面遷移主導で顧客に画面や画面遷移を作成することで普段発掘できなかった要求が導けるようになることが期待できる。

Demand extraction support tool driven by screen transition for customer participation promotion

Shinsuke TERANAKA and Yoshikazu UEDA

Ibaraki University

The failure of the system development project is induced due to the conditions that the important demands aren't shown, and the frequent change of the requirement and the specifications. In this research, we thought that the insufficiency of communication between client and developer was a cause. The support tool, which aimed at creating the conditions to share the extracted requirement information for a client to participate actively, has been developed to solve this cause. This tool supports a customer's demand extraction under the screen transition drive so that a client can participate in the development actively. And this supports the UML output derived from screen transitions to share the requirement information. By this research, it can be expected that a customer shows more and adequate requirement, and that a developer draws a requirement appropriately from the customer.

1 はじめに

近年、要求されるシステムが大規模化・複雑化してきているにもかかわらず工期の短縮化は著しくなるばかりであり、現状のシステム開発は失敗に終わっている場合が多い。システム開発を行う際、システム開発に必要な、あるいは重要な部分であるにも関わらず、顧客から要求として出されてい

ないという状況はよく発生する。この原因として最も考えられるのは、顧客と開発者のコミュニケーションがうまく取れていない場合である。このような状況でありがちな問題として開発途中に要件変更・仕様変更が頻発することが挙げられ、手戻りによる開発負荷の増大が発生し、顧客と開発者間に軋轢を生じさせる原因となってしまう。これ

を解決する方法として、開発者が顧客に目に見える成果物をいったん提示し、その状況を作った上で開発者が顧客に対しインタビューをして新たな要求を引き出すという手法が得策ではないかと考えられる。そのためのアプローチとして、プロトタイピングという開発スタイルが挙げられる [1].

また、獲得した要求に対しては顧客と開発者間で共有できるのが望ましい。そこで現在要求分析で主流であるユースケース分析を背景に UML [2] で情報共有できるのが得策であると考えられる。

本研究では、プロトタイピング手法と UML 出力が同時にできる環境を構築することを目的とした。その実現方法として、画面遷移主導による要求抽出が有効であることを示し、顧客が画面設計と画面遷移を編集すると同時に UML も出力できるようなツールの構築を試みた。顧客の画面イメージの要求や画面遷移を UML のユースケース図や状態チャート図として対応付けることによって UML 出力を可能にし、本ツールでは XMI [3] 形式出力を採用した。

2 画面遷移主導による要求抽出

2.1 画面遷移主導に着目する理由

- 顧客の目的達成

顧客にとっては UI が製品そのものとなる場合が多い。また UI デザインの前に画面にどの機能を持たせるかを明らかにするのが重要であると考えられる。顧客がシステムのイメージを理解するためには、システムの見かけ (GUI)、システムの振舞い (システム応答) を見せなければならない。顧客はシステムの動的な側面に関心を持ちやすく、画面や画面遷移といった目に見える要求は開発者側にイメージが伝わりやすい。

- 要求に対する顧客の意識

画面仕様に対しては、実際の顧客との打ち合わせでは、UI デザインは話のきっかけに過ぎない場合が多い。それは顧客自ら画面イメージを作成しようとして初めて画面でやりたい機能を考えることが多いからである。顧客が意図した要求を引き出すためには実際に顧客に画面イメージと画面遷移を作成させる。そこで初めて「その画面で何がしたいか」と

いったものが顧客と開発者の双方が理解できるようになる。

- 開発途中での画面仕様変更によるデメリット

画面はクラスやデータベーステーブルと密接な関係があるため、開発途中での画面仕様変更はクラス設計やデータベース設計を見直す可能性が考えられ、後戻りが発生するため、コストがかかる。よって早期に画面仕様と画面に対する機能を決定していなければならない。

以上から画面遷移主導は顧客の要求の抽出に有効であると期待でき、顧客の積極的な参加を促すことができると考えられる。

2.2 UML 出力に着目する理由

- 現在の要求分析

現在のシステム開発において要求分析を行う場合、UML を使ったオブジェクト指向による概念モデリングや、ユースケース分析が主流である。したがって、開発者側は顧客の要求をユースケースを用いて分析して、要求の理解を図る。

- UML による情報共有

開発者は主流の UML を利用したいという要望と顧客でも理解可能な図式を使いたいという要望を満たすために一般に広く利用され出したユースケース分析が適当であるとも考えられる。ユースケース図といった視覚的にわかりやすいものは、顧客は理解できるので情報共有には適しているといえる。

- 仕様書作成の負担軽減

分析した成果物を仕様書にまとめるときにも UML は利用される。UML は様々な制約があり、ある程度熟練している必要がある。したがって、UML 出力ができれば開発者側の手間が省けるメリットが生じる。本研究ではユースケース図と状態チャート図の出力によって仕様書作成の負担軽減につながると考えている。

以上から UML 出力は顧客と開発者間の情報共有に有効であると期待でき、コミュニケーション手段として適切であると考えられる。

2.3 提案手法

本節では、まず提案手法の概要を示し、次にUML生成手順について述べる。

● 概要

画面遷移主導による要求抽出を支援する環境として顧客がビジュアルプログラミング環境で機能要求、画面遷移図、UI画面を作成し、これらからUMLを導出することで、開発者側に画面仕様出力やUMLモデルを提供する。提供するUMLモデルはユースケース図と状態チャート図である。開発者側はUML記述をもとに要求分析を行い、開発者側が顧客にインタビューを行うことで顧客から新たな要求を引き出したり、支援ツールにより出力されたユースケース図から足りない要求を引き出すことを目的としている。また前節で画面遷移主導が有効であることを述べたので、機能要求作成時のシナリオには画面遷移を適用する。図1に提案手法の概要を示す。

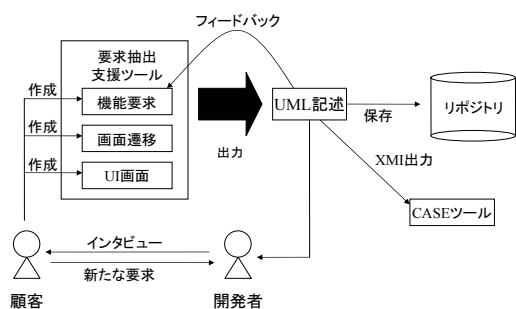


図1: 提案手法の概要

● UML 生成手順

本手法では以下のステップにしたがってUML図を生成する。

step1 顧客が本研究で開発するツールを用いてユースケースを作成する

step2 顧客が本研究で開発するツールを用いてUI画面を作成する

step3 顧客が本研究で開発するツールを用いて画面遷移図を作成する

step4 本研究で提案するUI画面とUMLの対応関係に従って、UI画面と画面遷移を対応付ける

step5 本研究で提案するUI画面とUMLの対応関係に従って、ユースケースと画面遷移を対応付ける

step6 UML図を生成する

これら6つのステップに基づいてUML図出力をサポートするツールを構築する。

3 UMLとUI画面との対応付け

本章では画面遷移からUMLの導出方法としてUMLとUI画面の対応付けについて述べる。

3.1 ユースケースと画面の対応関係

ユースケースと画面との対応関係は主に1対1の関係、1対多の関係、多対1の関係が考えられる。これらの場合について説明する。

● 1対1の対応関係

システムの機能と画面が一致しているときの関係。この場合は再利用しやすいユースケースである場合が多い。図2に1対1の対応関係の例を示す。



図2: 1対1の対応関係

● 1対多の対応関係

システムの1機能が1つの画面遷移と一致しているときの関係。図3に1対多の対応関係の例を示す。

● 多対1の対応関係

1つの画面に複数の機能が現れているときの関係。図4に多対1の対応関係の例を示す。

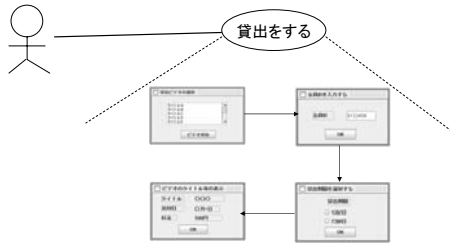


図 3: 1 対多の対応関係



図 4: 多対 1 の対応関係

3.2 GUI と状態チャート図の対応付け

本研究では、各画面に配置されている GUI コンポーネントに基づいて各画面が持つ振舞いを決定する。画面遷移での GUI コンポーネントが持つ振舞いを次の 4 つの項目として定義する。また「分岐」の起きる場合を 4 つの条件として定義する。

状態 : 画面遷移における画面のデータ単位

属性 : 画面データの属性。文字列表示や画像表示に該当

遷移 : アクションとして遷移を伴うもの

分岐 : 分岐条件として設定できるもの

- 入力
- 項目選択
 - － 排他的選択
 - － 複数選択可

以上の位置づけを GUI コンポーネントに対応させたものを表 1 にまとめる。この対応付けは画面遷移の編集で用いる。対応付けさせる GUI は様々な GUI ツールキットにも存在するような基本ウィジェットを採用した。次に GUI コンポーネントの

表 1: GUI 部品の画面遷移での振舞い

コンポーネント名	振舞い
フレーム	状態
ダイアログ	状態
ラベル	属性
テキストフィールド	属性
パスワードフィールド	属性
テキストエリア	属性
スライダー	属性
スクロールバー	属性
キャンバス	属性
ボタン	遷移
チェックボックス	分岐
コンボボックス	分岐
ラジオボタン	分岐
トグルボタン	分岐
リスト	分岐
ツリー	分岐

振舞いと状態チャート図との対応付けについて説明する。

「状態」の場合は、GUI コンポーネントが画面そのものである。状態チャート図の主要な構成要素の一つである。状態チャート図での状態の表示名は画面の機能名を記述する。

「属性」の場合は、画面に配置されている GUI コンポーネントを指す。状態チャート図においては状態が持っている属性であるので本研究では構成要素として扱わない。

「遷移」の場合は、画面が遷移先を持つときである。状態チャート図の主要な構成要素の一つである。状態チャート図での遷移のラベルは GUI コンポーネントの **text** で記述する。

「分岐」の場合は、画面が遷移するときに分岐情報に従って、それぞれの状態に遷移する。状態チャート図では選択点を使用する。選択点は白丸で図示される。

選択点に至るときの遷移のラベルは次の通りになる。

type [name]

また選択点に達した後は次のラベルを使って表現する。

[value = text]

[else]

遷移のラベルで定義した変数の説明を表2に示す。

表 2: 遷移のラベルで定義した変数

変数名	説明
type	GUI コンポーネント名
name	ボタングループ名
value	遷移先を決定する評価用変数
text	GUI コンポーネントのテキスト名

この選択点を用いた場合を考えると、「排他的選択」の場合は、グループの要素数を n 個とするならば最大 n 通りの遷移を考えられるが、「複数選択可」の場合は、グループの要素数を n 個とするならば最大 2^n 通りの遷移が考えられる。

図5にボタングループ名 **group1** に属するラジオボタンを3つ配置したときの「分岐」の場合のステートチャート図を示す。

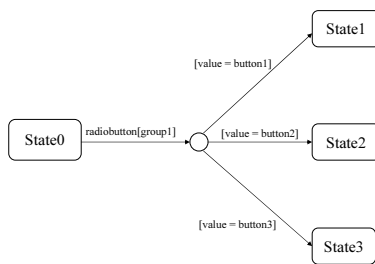


図 5: 分岐を伴うときのステートチャート図

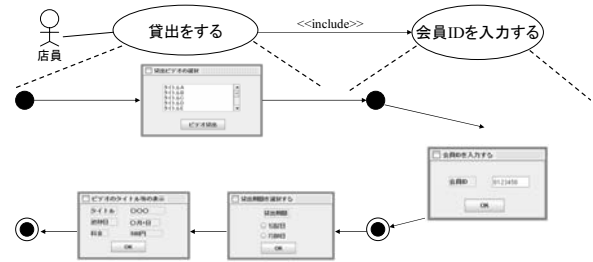


図 6: 包含関係

ユースケースが他のユースケースを拡張する。ユースケースの代替フローに該当するのが拡張されるユースケースである。画面遷移の中で条件により他のユースケースの画面遷移を含むとき拡張関係が成り立つ。この図式表現の例を図7に示す。拡張関係は選択的な要素を含んでおり、拡張するユースケースは再利用できるユースケースである必要がある。

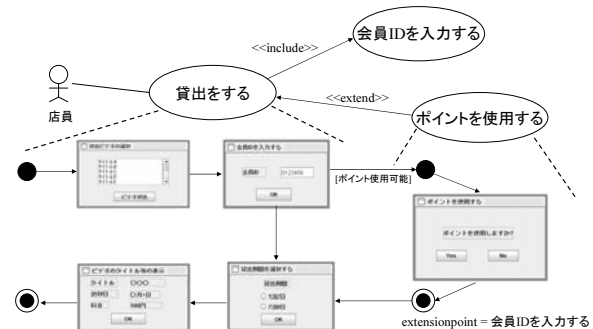


図 7: 拡張関係

3.3 ユースケース同士の対応関係

ユースケース同士の対応関係には包含関係 (include), 拡張関係 (extend), 汎化関係 (generalization) が存在する。

● 包含関係

あるユースケースが他のユースケースを含むときの関係。あるユースケースの画面遷移の中で他のユースケースを含むとき、包含関係が成り立つ。この図式表現の例を図6に示す。

● 拡張関係

ユースケース同士の継承関係であり、ある

● 汎化関係

あるユースケースAがあり、ユースケースBはAと同じ一連の流れを持ちながら、B独自の機能も持っているという場合に汎化関係が成り立つ。画面遷移においてはユースケースと画面の対応関係が多対1の場合に成り立つものとする。これは1つの画面が1つのユースケースとしてみなすと、そのユースケースは1画面の中に存在する複数のユースケースの集合体として存在する形となる。集約という捉え方もできるがユースケースには集約の概念がないため、汎化という形で多対1の対応関係を表現する。この図式表現の例

を図 8 に示す。

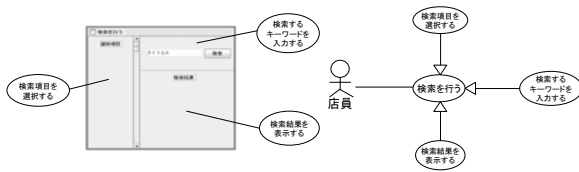


図 8: 汎化関係

3.4 ユースケース図とステートチャート図の対応付け

ユースケース図上で包含関係や拡張関係が存在する場合、ステートチャート図では包含ユースケース内の画面遷移や拡張ユースケース内の画面遷移は下位状態で表す。ユースケース図に対応したステートチャート図の例を図 9 に示す。

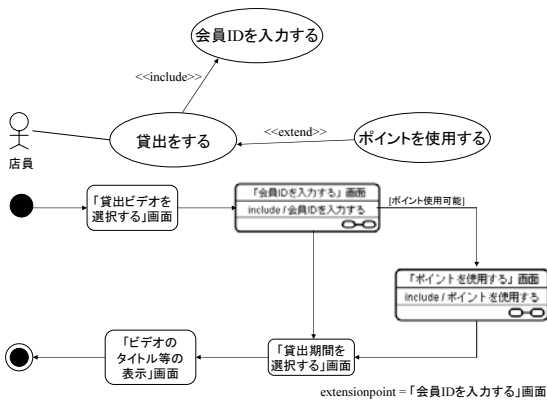


図 9: ユースケース図に対応したステートチャート図

4 ツールの提案

本研究で提案するツールは 2 章で提案した手法の中で、ユーザが簡単に画面イメージおよび画面遷移を作成できるツールである。また、3 章での対応関係にしたがって、ユースケースとステートチャート図を作成し、出力する UML エディタである。本ツールでは 3 つの機能をユーザに提供する。1 つめは機能要求エディタ。2 つめは画面遷移を編集できる画面遷移図エディタ。3 つめは画面設計をビジュアル的に行える画面エディタである。

4.1 ツールの利用者

過去に GUI 要件作成を経験したことがある顧客を対象ユーザとしている。

4.2 機能

機能要求エディタでは機能要求の設定を行う。主な設定項目として、機能要求に対応するユーザの設定、包含機能や拡張機能の設定、事前条件や事後条件の設定が行える。画面遷移エディタ部では画面遷移図の作成を行う。画面エディタ部では画面遷移エディタ部分で配置した画面に対し、表 1 の GUI を配置することで画面を作成し、画面エディタで作成した画面は画面遷移エディタの画面の概観に反映される。また UML 出力機能として本ツールではユースケース図と各ユースケースに対応したステートチャート図を出力する。出力フォーマットは他の MDA ツールとの連携を考慮し、XMI 出力をサポートした。ツールの概要を図 10 に示す。

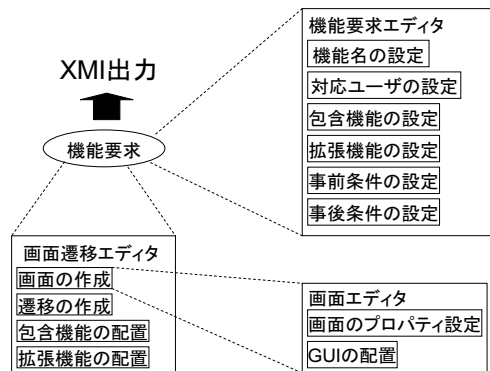


図 10: ツールの概要

4.3 利用手順

本ツールのおおまかな利用手順を以下に示す。

- step1 機能要求エディタでユースケースやアクタの名前の設定や関係付けを指定。
- step2 画面遷移図エディタで画面を作成。
- step3 画面エディタでは step2 で作成した画面に対し、GUI コンポーネントを配置。
- step4 画面遷移図エディタで遷移と GUI の遷移を関係付け。

以上のステップに基づいて機能要求を作成していく。

5 ツールの実行

本章では4章で提案したツールがCASEツール Rational XDE Developer Plus for JAVA[4]と連携が取れることを示すために、ツール上で画面遷移を作成し、ツールで出力したXMIファイルをもとに Rational XDE に読み込ませた。

5.1 テストモデル

本ツールでは表3のレンタルビデオシステムを例として作成した。ツールの実行例を図11に示す。

まず、レンタルビデオシステムというプロジェクトを作成し、その中に「店員」ユーザ、「貸出をする」機能、「会員IDを入力する」機能、「ポイントを入力する」機能を作成し、「会員IDを入力する」機能を編集して、この機能要求に「店員」ユーザと包含機能として「会員IDを入力する」機能、拡張ユースケースとして「ポイントを使用する」機能を登録した。

作成する画面遷移は図7で例示したものを使用する。遷移のステップは以下のようにになっている。

step1 貸出ビデオを選択する

step2 会員IDを登録する

step3 ポイントを使用する

step4 貸出期間を選択する

step5 ビデオのタイトル等を表示する

以上のステップの中で **step3** は包含機能「会員IDを入力する」、**step4** は拡張機能「ポイントを使用する」であるので、ツール上ではユースケースの図式表現である楕円で表示されている。

5.2 CASE ツールへの適用

本ツールで作成したテストモデルをXMIファイルとして Rational XDE Developer Plus for JAVA に読み込ませた。出力した結果を図12、図13に示す。

表 3: レンタルビデオシステムのシナリオ

ユースケース	貸出をする
アクタ	店員
事前条件	
事後条件	
シナリオ	<ol style="list-style-type: none">1. 「ビデオ貸出」を押す2. 貸出ビデオを選択する3. 会員IDを入力する (include)4. 利用履歴を調べる5. 貸出期間を選択する6. 「OK」ボタンを押す7. 料金を計算する8. ビデオのタイトルを表示する9. 返却日を表示する10. 料金を表示する11. 「了解」を押す12. 会員データベースを更新する13. ビデオデータベースを更新する

6 関連研究

本研究の関連研究として森田ら [5] によるシステムの動的側面に着目した要求獲得支援の研究が挙げられ、ユースケース図、拡張ユースケース図、アクティビティ図を顧客に提示することで要求を獲得しようとする試みが行われている。本研究では、顧客が作成したプロトタイプからユースケース図と状態チャート図を生成し、それをもとにフィードバックすることで新たな要求を獲得しようという点で異なっている。また、白金 [6] によるシナリオやユースケースから GUI プロトタイプを導出する研究では、ユースケースシナリオのシナリオ併合グラフから入力・出力情報といったインタラクション項目を加えることで GUI プロトタイプを生成する。本研究では画面イメージや画面遷移から UML への導出をするアプローチをとっており、顧客がとっつきやすい画面や画面遷移を要求のシナリオとして捉えていることがこの研究と異なっている。

7 まとめ

今回の研究では、顧客参加を積極的に促せられ、開発者とのコミュニケーションを促進できる支援

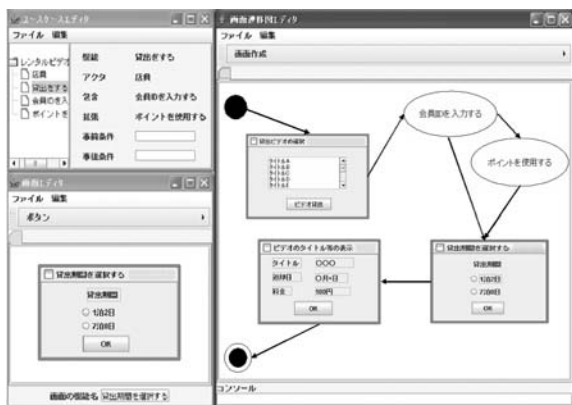


図 11: ツールの実行例

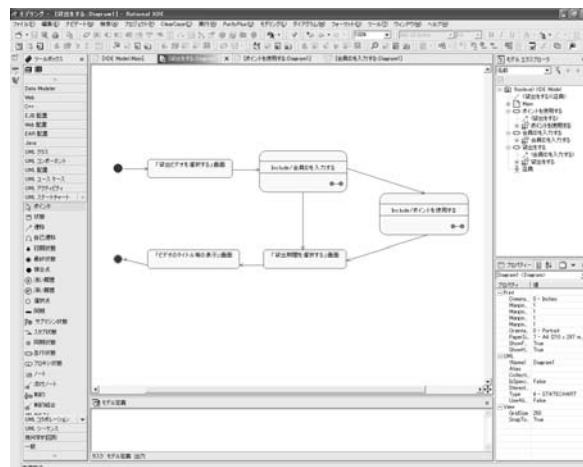


図 13: 「貸出をする」ユースケースのステートチャート図

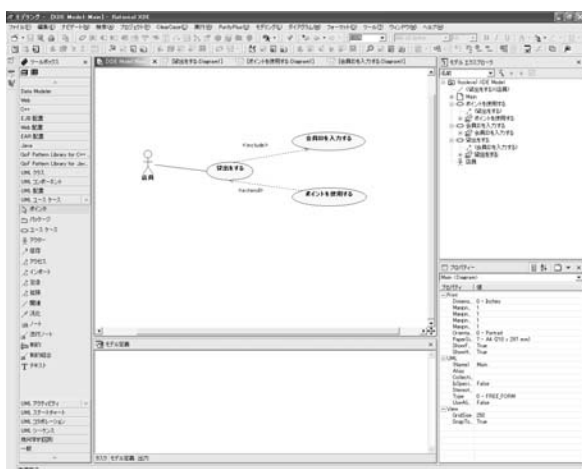


図 12: Rational XDE で出力したユースケース図

ツールを作成した。その際に画面遷移から UML を導出し、出力した XMI ファイルを IBM 社の MDA ツール、Rational Rose XDE for JAVA に読み込ませることによって UML 図を表示させ、他の CASE ツールとの連携が図れることを示したので、UML 出力が可能となったといえる。

今後の課題としては、ユーザに実際に利用してもらった実利用による評価、様々な CASE ツールとの連携、ツール内で UML 図を表示できる機能の追加、ツールのユーザインタフェースの改良、画面遷移のプレビュー機能を追加する必要がある。また、今回採用した GUI は様々なツールキットにある基本的なウィジェットを採用した。しかし、基本的なウィジェットでは汎用性はあるが、数が少ないと考えられる。よって今後は、対応付けさせる GUI の種類を豊富にすることも考える。

参考文献

- [1] 松本吉弘監訳：ソフトウェアエンジニアリング基礎知識体系 -SWEBOOK-, オーム社.
- [2] Object Management Group 著, OMG Japan SIG 翻訳委員会 UML 作業部会訳：UML 仕様書, ASCII (2001) 仕様書.
- [3] XMI : <http://www.omg.org/technology/documents/formal/xmi.htm>
- [4] IBM : Software - Rational Rose XDE Developer for Java - Product Overview <http://www-306.ibm.com/software/awdtools/developer/java/>
- [5] 森田俊夫 山田宏之：ソフトウェア開発対象業務の動的側面に注目した要求獲得支援, ソフトウェア工学 125-11, (2000. 1. 14).
- [6] 白金純子：GUI アプリケーションの開発に関する研究, 早稲田大学大学院理工学研究科, 博士論文審査報告書, (2002 年 6 月) .