

# UML・プロトタイプを組み合わせた要求仕様の妥当性確認

小形 真平<sup>†</sup> 松浦 佐江子<sup>‡</sup>

芝浦工業大学 システム工学部 電子情報システム学科<sup>†‡</sup>

## 1. はじめに

システム開発では、開発者は顧客の要求を分析・仕様化した要求仕様を作成する。しかし、作成初期の要求仕様は顧客にとって不満足である場合が多いため、顧客自身が妥当性確認を行う必要がある。この確認を行うことを目標としてプロトタイプを用いる方法が提案されている[1]。

しかし、プロトタイプを用いる場合、表1のような開発コストや設計物との整合性について問題がある。

本稿では、Web アプリケーションを対象とし、プロトタイプ、そして、UML(Unified Modeling Language)のモデルであるアクティビティ図、ユースケース記述を用いて、表1に示す方法で解決する要求分析方法を提案する。

表1 問題点と解決策

問題点	解決策
プロトタイプと設計物との整合性を図ることが難しい。	プロトタイプと設計物を、業務手順と入力項目の点で等価に対応付けを行う。
プロトタイプ作成により、開発コストが増加する。	プロトタイプと設計物間で、自動変換を可能にし、作業量を抑える。
要求仕様の妥当性確認で生じる修正を、開発者独自の方法で修正することにより修正品質の低下する。	プロトタイプと設計物間の自動変換により、修正方法を形式化し、一定の修正品質を保证する。

## 2. 要求仕様の定義

開発者は、要求分析段階でいくつかの設計物を作成し、それらを元に要求仕様を作成する。本稿では、要求分析段階の設計物をアクティビティ図、ユースケース記述とし、開発するシステムの全体像や機能をまとめて、顧客に提示するために作成するプロトタイプを要求仕様とする。なお、「設計物」と「プロトタイプ」は明確に区別するものとする。

### 2.1 要求分析段階での設計物

UMLのアクティビティ図は、業務の流れをモデル化するために処理や作業を順序づけて表す。本稿におけるアクティビティ図は、加えて業務手順中に必要な入力項目を表現し、ユースケース記述、プロトタイプとの相互変換を行うために記述制約を加えたものを指す。

記述制約として、アクティビティ名はユースケース単位で記述されることを明示するためユースケース名と同名にする。アクションは「~を~する」という簡潔な記述を行う。アクションの行為者を明示するために、その行為者をパーティション名としたパーティションを必ず記述する。なお、開発対象のシステムのパーティション名は「システム」とする。システムがアクターの入力項目の正否を判定することにより起こる分岐条件では、「(入力項目)が正しい」、「(入力項目)が正しくない」という記述にする。

ユースケースとはシステムの機能の一部を指す。ユースケース記述とは、システムとユーザとのやりとりを自然言語で記述した文書である。この記述では、ユースケース名、アクター、事前条件、基本フロー、代替フロー、例外フローにより構成される。各フローはステップと呼ばれる順序だてた処理を1単位とし、ステップ番号と呼ばれる番号を処理の順番として構成される。また、アクティビティ図との相互変換のため記述制約がある。

1つのステップでは、アクティビティ図の要素で定義すると「(パーティション名)は、(アクション)」という記述になり、条件分岐がある場合は、「(ガード条件)場合、(パーティション名)は、(アクション)」となる。また、並列動作において、ステップ番号を表現する場合、「(フォークノードの順番<1,2,...>)(並列動作のフローの番号<a,b,...>)(ステップ番号<a,b,...>)」で表す。例えば「3ab」となる。

アクティビティ図は開発者が業務手順を直感的に理解する上で適した形式であり、ユースケース記述は、図が膨大なサイズになると読みにくくなってしまふのに対し、文章として整理された形式で記述できる。また、2種の設計物を比べることによって、上記に述べた各設計物の特性から内容についての誤りを早期発見できることを意図している。以上が2種の設計物を選んだ理由である。

### 2.2 プロトタイプで確認すべき内容

本稿ではWebアプリケーションを対象としているため、プロトタイプはユーザインタフェースのプロトタイプでありHTMLで構成される。プロトタイプで顧客に確認してもらう内容は、業務手順とその手順を行う際に必要となる入力項目についてである。

業務手順については、どのように画面が遷移していくのか、入力に対してどのような出力がなされるのか、1画面中に出力される情報はどの程度の量と内容を持つのか、を確認する。これにより、顧客がある入力を行うときに、どのような情報を見ながら入力できるのか、または、入力に対する出力結果は期待されるものかどうかを確認できる。

入力項目については、どの画面でどのような入力項目があり、その入力形式はなにかを確認する。これにより、顧客は入力を行う際に入力項目が不足していないかの確認と、テキスト形式で入力するのか、それとも、リストから選択形式で選ぶのか、といった開発者の考えている具体的な入力形式を確認できる。

上記のプロトタイプはアクティビティ図より自動変換されるため、画面構成はアクティビティ図のフローの遷移状態に対応し、入力項目はアクティビティ図のシステム外のパーティションに記述されるアクションに対応する。

より具体的な対応付けは3.3節に後述する。

## 3. 要求仕様の開発・妥当性確認の手順と自動変換規則

### 3.1 提案方法の作業手順

1. 既存の業務手順、機能要求が含まれる顧客の要求からユースケースを導出する。
2. ユースケース毎に、システム導入後の業務手順を開発者が考案する。

Validation of Requirements Specification using UML and Prototypes .

<sup>†</sup> Shinpei Ogata      <sup>‡</sup> Saeko Matsuura

<sup>†‡</sup> Shibaura Institute of Technology Department of Electronic Information Systems

- システム導入後の業務手順に沿ったアクティビティ図、ユースケース記述を作成する。  
(一方を手動で作成し、それを元にもう一方を自動変換で生成する)
  - 3.の設計物に対して、主に業務手順と入力項目に注目して比較し、手順の誤りや必要な入力項目の不足がないかを開発者が確認する。
  - 開発者が設計物の内容に満足した場合、アクティビティ図を元にプロトタイプを自動変換し生成する。満足しない場合は、再度アクティビティ図とユースケース記述を修正・作成し手順4に戻る。
  - 顧客により、プロトタイプを確認してもらい、修正箇所を指摘してもらう。
  - 開発者はプロトタイプを構成するHTMLを直接編集することによって修正を行う。
  - 修正されたプロトタイプを元にアクティビティ図を自動変換により生成する。
  - 8.で生成されたアクティビティ図を元にユースケース記述を自動変換により生成し、手順4に戻る。
- 以下、6.のプロトタイプ確認で顧客が満足するまで4~9のプロセスを繰り返す。なお、本稿では、5.までのプロセスを目的とした自動変換ツールを作成する。

### 3.2 ユースケース記述とアクティビティ図の変換規則

表2にユースケース記述の要素とアクティビティ図の要素の対応付けを示す。

表2 ユースケース記述とアクティビティ図の対応付け

ユースケース記述	アクティビティ図
ユースケース名	アクティビティ名
アクター名	パーティション名
ステップ	アクション
例外フローへの分岐	分岐条件の判定がアクターの入力項目の正否
代替フローへの分岐	分岐条件が例外フローへの分岐条件以外
並列動作のステップ番号の記述	フォークノードによる並列動作する複数フローの定義

例えば、図2に記述される「ユーザ認証する」のアクティビティ図の[システム]パーティションの[ユーザ認証情報の入力]アクションは、ユースケース記述のステップ「2.[システム]が[ユーザ認証情報の入力]を要求する」と対応付けられる。

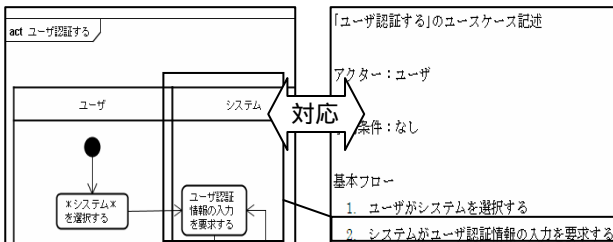


図2 ユースケース記述とアクティビティ図の対応例

### 3.3 アクティビティ図とプロトタイプの変換規則

表3にアクティビティ図の要素とプロトタイプの要素の対応付けを示す。先の例をプロトタイプへ変換した場合、画面遷移や入力項目は図3のように対応する。システム外のアクションからシステム内のアクションへ遷移する場合は3通りあり、計4画面が生成される。また、入力形式について、「~を入力する」はテキスト形式に、「~を選択する」およびガード条件はリンク形式に対応している。

表3 アクティビティ図とプロトタイプの対応付け

アクティビティ図	プロトタイプ
「~を~する」というシステム外のアクション、「~をする」について、現状は「確認する」「選択する」「設定する」「入力する」の4つ	「~を」の対象を項目に、「確認」を表示、「選択」をリンク形式、「設定」を択一選択形式、「入力」をテキスト形式に対応させた入力項目
システム外のアクションからシステム内のアクションへ遷移する制御フロー	画面遷移の起こるタイミング、具体的にはこのタイミングで画面を分割
アクションの「~を」に対し「*~*を」という特定表記	入力項目に対するキャプション

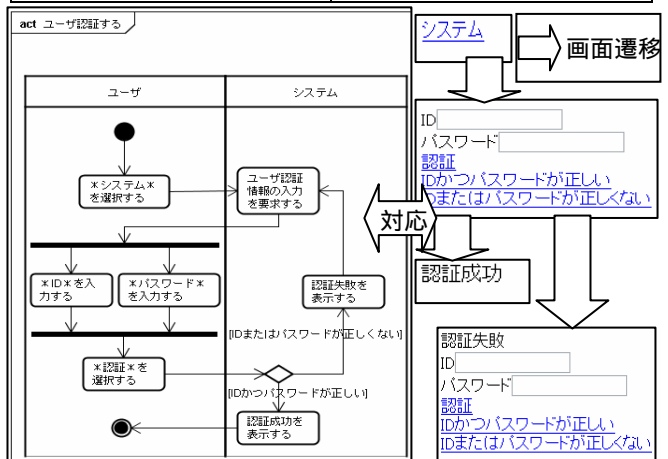


図3 アクティビティ図とプロトタイプの対応

## 4. 適用事例と考察

### 4.1 実稼働システムへの適用

このアプローチを本学で稼働中のシステムに適用した。これは、システム開発時におけるグループワーク支援目的のために開発された。その中には議事録機能がある。議事録機能では、話し合いの記録を行い、いつだれが内容を確認し、どの作業が終了したかを作業員任意で記録できる。

### 4.2 考察

本稿のプロトタイプは「業務手順と手順中の入力項目を理解」が目的である。この事例では、提案方法によりその目的の情報は一通り確認できた。しかし、自動変換が困難である部分や、表現が限定的になってしまう部分があった。

画面出力では、顧客の理解しやすい形式として具体的なデータが用いた出力が考えられる。しかし、抽象的なデータで表現されるアクティビティ図からの変換では、プロトタイプの出力部分の表現は非常に限定的になってしまう。

また、定数的なデータ(HTMLにおけるリストボックスに入っているような要素)をプロトタイプに反映するためには、アクティビティ図の表現要素では、ノートのような汎用的なコメント要素しかないため難しい。

### 5. まとめ

今後は、プロトタイプの出力表現についてアクティビティ図では表現の難しい具体的なデータ要素を持つモデル(オブジェクト図やクラス図)を自動変換に組み込み、プロトタイプの生成方法を改善していく。さらにプロトタイプからアクティビティ図への自動変換ツールを作成する。

### 6. 参考文献

- [1] 大西淳, 郷健太郎: 要求工学, 共立出版, 2002
- [2] R.Vonk: プロトタイピング CASE テクノロジーの有効利用, 共立出版, 1992