

## 要求定義段階のモデルからのシナリオ生成

後藤 健太<sup>†</sup> 小形 真平<sup>‡</sup> 白銀 純子<sup>††</sup> 中谷 多哉子<sup>††</sup> 深澤 良彰<sup>†</sup>

要求定義段階でソフトウェアの構造の概略を概念モデルとして、操作フローをシナリオとして表すことが多く、これらはその後の開発過程での様々な成果物を作成していく上でのベースとなることも多い。本研究では、概念モデルの妥当性を確認するために概念モデルからシナリオを生成する手法を提案する。

## Scenario Generation from Models in Requirements Definition Phase

KENTA GOTO<sup>†</sup> SHINPEI OGATA<sup>‡</sup> JUNKO SHIROGANE<sup>††</sup>  
TAKAKO NAKATANI<sup>††</sup> YOSHIAKI FUKAZAWA<sup>†</sup>

In requirements definition phase, outlines of structures of software is often expressed as conceptual models, and the operation flow is often described as scenarios. These are often used for the base on creating various documents in the later development process. In this paper, we propose a method of generating scenarios from conceptual models to confirm the validity of conceptual models.

### 1. はじめに

要求定義の段階でソフトウェアの構造の概略を概念モデルとして表すことが多い。概念モデルは要求定義の段階でも特に初期に作られることが多く、その後の開発過程での様々な成果物を作成していく上でのベースとなることも多い。したがって、この概念モデルの妥当性を確認することは非常に重要である。

一方で、要求定義段階でソフトウェアの操作フローはシナリオで記述されることが多い。シナリオはユーザとソフトウェアのインタラクションのフローを表したものである。主シナリオ、代替シナリオ、例外シナリオという種類があるが、最も一般的な成功フローである主シナリオを定義することがシナリオ作成の最初の段階として重要である。

本研究ではシナリオのフローは概念モデル上で辿ることができると考えている。シナリオのフローを概念モデル上で辿ったとき、概念モデル上での構造の順序と一致する必要がある。つまり、概念モデルを辿ってシナリオのフローを表現したとき、そのシナリオのフローが妥当なものになっていれば、その概念モデルの妥当性を確認できると考えている。

そこで本研究では、概念モデルからシナリオを生成する手法を提案する。これにより、生成されたシナリオを開発者やユーザが確認することで概念モデルの妥当性を確認できる。また、シナリオのフローは実際のソフトウェアと

ユーザとのインタラクションであるため、UI の実装に結び付いていく。将来的には UI プロトタイプを生成し、さらに厳密に概念モデルの妥当性等を確認できるようにすることを目指す。

### 2. 本手法の特徴

本研究では、概念モデル[I]を入力として、シナリオを生成することを目指す。

ソフトウェアの概略を表現するためにしばしば概念モデルが利用される。この概念モデルはクラス図であるため静的な構造を表現したものである。また、シナリオ等のフローは概念モデルのクラス間の関連を辿ったフローになっていると考えられる。したがって、概念モデルのクラス間の関連を辿ったフローがシナリオとして妥当なフローになっている必要がある。また、シナリオには主シナリオ、代替シナリオ、例外シナリオという種類があるが、代替シナリオと例外シナリオは主シナリオを基に作成される。したがって、主シナリオを抜けや間違いが無いように作成することが大変重要である。

本手法の利点は以下の通りである。

- 概念モデルからソフトウェアのフローを抽出することで概念モデルの妥当性を確認できる。
- 概念モデルを利用することでソフトウェアの構造と一致したシナリオを生成できる。
- 抜けや間違いの無いような主シナリオを生成することで、代替シナリオや例外シナリオも質の高いものを作成する支援となる。

<sup>†</sup>早稲田大学基幹理工学部情報理工学科

<sup>‡</sup>信州大学工学部情報工学科

<sup>††</sup>東京女子大学現代教養学部人間科学科

<sup>†††</sup>筑波大学大学院ビジネス科学研究科

### 3. 本手法の手順

本手法では UML によって記述した概念モデルを入力としてシナリオを自動で生成する。

本手法は入力とするモデルを作成してからシナリオを自動生成するまで、大きく分けて 4 つのステップで構成される。

1. 概念モデルの入力
2. 入出力項目の抽出
3. フローの抽出
4. シナリオ生成

まずステップ 1 で概念モデルを入力する。ステップ 2 でシナリオ中のイベントの基となる入出力項目等の情報を抽出し、ステップ 3 でシナリオのフローの基となる情報を抽出する。そしてステップ 4 でシナリオを生成する。

#### 3.1. 概念モデルの入力

本手法では、概念モデルとしてクラス図を入力とする。この段階で概念モデルから入出力項目等の静的な情報と必要な数のシナリオのフローを抽出する。本システムの入力として用いる概念モデルの例として図 1 に示す。この例は、銀行システムにおける概念モデルである。

#### 3.2. 入出力項目の抽出

入力とした概念モデルにおいて、各クラスの属性 1 つ 1 つをシナリオのイベント中の入出力項目として抽出する。このとき、入出力項目の項目名は「クラス名の属性名」という形式とする。例えば、図 1 に含まれる「引出」クラスであれば、「金額」という属性を「引出の金額」という形で入出力項目として抽出する。また、各クラスの属性に「<<input>>」または「<<output>>」というステレオタイプを追加することで、入力または出力項目のみとして抽出する。例えば、図 1 に含まれる「暗証」クラスであれば、「番号」という属性に「<<input>>」というステレオタイプが追加されているため入力項目のみとして抽出する。

#### 3.3. フローの抽出

##### 3.3.1. アクタの抽出

まず、概念モデルから「<<actor>>」というステレオタイプが追加されたクラスを抽出しアクタとして判断する。

##### 3.3.2. フローの組み立て

3.3.1 節にて抽出したアクタのクラス名と属性を抽出し、次にアクタの関連を辿りそのクラスのクラス名と属性を抽出する。さらにそのクラスの関連へと 1 つずつ順に辿っていきクラス名と属性を抽出し、すべてのクラスを辿り終えたら辿った順序に沿って抽出したクラス名と属性を基にシナリオのフローを抽出する。また継承によるサブクラスをもつクラスを辿った場合、それぞれのサブクラスを選択するシナリオのフローをサブクラスの数だけ抽出する。例えば、図 1 において継承によるサブクラスに注目すると、この場合の「銀行システム」は「振り込み」、「引出」、「残高参照」の 3 種類のシナリオが必要なが読み取れる。つまり、この場合入力とした概念モデルから 3 種類のシナリオのフロ

ーを抽出する。

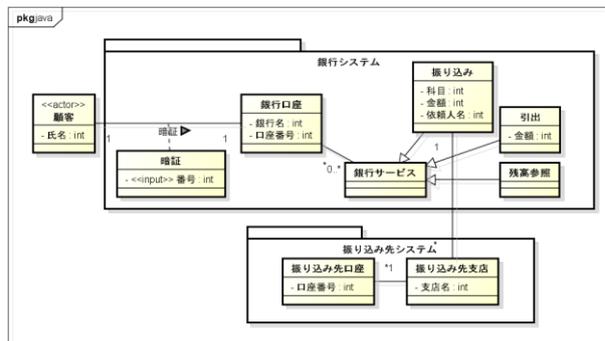


図 1 銀行システム概念モデル

#### 3.4. シナリオ生成

3.1 節にて抽出した入出力項目と必要な種類分のシナリオのフローを基にシナリオを生成する。ここで例として図 1 から抽出した情報を基に本システムが生成した「銀行システム」のシナリオのうち、「引出シナリオ」を図 2 に示す。生成するシナリオにおいて、ユーザの操作は現状では入力操作と選択操作から成り、各クラス毎の入力項目におけるユーザの入力操作に対しシステムが出力を行う。選択操作においては、各シナリオを選択する操作を入れ、そのシナリオに該当する入出力項目の処理を記述に必要なシナリオのフローを生成している。

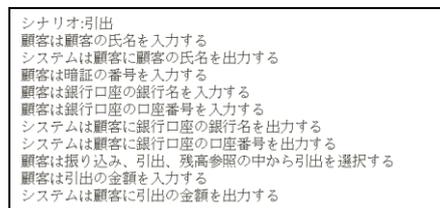


図 2 銀行システムにおける引出シナリオ

#### 4. おわりに

本研究では、概念モデルからシナリオを生成する手法を提案した。本手法により概念モデルの妥当性の確認や品質の高い主シナリオの生成を容易に行うことができるようになると思われる。

今後の課題は以下の通りである。

- ・シナリオのフローの抽出方法の確立
- ・UI 生成手法の確立
- ・システムの実装と評価

#### 参考文献

[1] Doug Rosenberg, Kendall Scott 著, 長瀬嘉秀, 今野睦 監訳, 株式会社テクノロジックアート 訳: ユースケース入門 ユーザマニュアルからプログラムを守る, 株式会社ピアソン・エデュケーション, 2001