



GS LetterNeo Vol.4

2008年11月

ユースケースモデリング

オブジェクトモデリングスペシャリスト
土屋 正人

TSUCHIYA Masato
m-tsuchi@sra.co.jp

◆ユースケースの悩みどころ

UML の中で、最も単純でありながら最も作成するのが難しいものがユースケースです。ユースケースは要求を整理するときに使われますが、ユースケース図で表現するのはユースケースとアクタ、そしてそれらの関係のみで、個々のユースケースの内容は**ユースケース記述**と呼ばれる成果物で別途定義することになります。この書式については何も規定されていないため、プロジェクトや組織で決める必要があります。

また、**ユースケースの粒度**をどうするかという点も悩みどころですが、これは**ユースケースの定義**がわかりにくいことが影響していると思います。

私の手元にあるものだけでも、ユースケース専門に扱っている本が8冊もありますが(表1)、ユースケースの定義がそれぞれ微妙に異なっています。また、同じ著者でも少しずつ変わってきたりしています。

◆ユースケースの定義と粒度

例えばユースケースの生みの親である **Jacobson** は、「分析に先行する要求記述」
「システム内の一連のトランザクション。システムはタスクの実行結果として特定のアクタに対して測定可能な価値を生み出す」
「システムの振る舞いをモデル化するもの。システムで

実行される一連のアクションであり、システムに関する1つ以上のアクタや利害関係者にとってたいは有益な目に見える結果を生む」

という具合に次第に定義が細かくなって来ています。

表 1 ユースケース関連書籍

著者／邦訳タイトル	原著出版
I.Jacobson	1992
オブジェクト指向ソフトウェア工学 OOSE-use case によるアプローチ	
I.Jacobson, M.Ericsson, A.Jacobson	1994
ビジネスオブジェクトユースケースによる企業変革	
D.Rosenberg, K.Scott	1993
ユースケース入門－ユーザマニュアルからプログラムを作る	
G.Schneider, J.P. Winters	1998
ユースケースの適用－実践ガイド	
A.Cockburn	2000
ユースケース実践ガイド－効果的なユースケースの書き方	
D.Kulak, E.Guiney	2000
ユースケース導入ガイド	
I.Jacobson, Pan-Wei Ng	2005
ユースケースによるアスペクト指向ソフトウェア開発	
D.Rosenberg, M.Stephens	2007
ユースケース駆動開発実践ガイド－オブジェクト指向分析から Spring による実装まで	

ちなみに UML2.0 仕様書では、

「システムにより実行される動作集合の仕様。一般的に1つ以上のアクタまたはシステムの他の利害関係者のために、価値ある観測可能な結果を生み出す」

となっていて、Jacobson の最後の定義とほぼ同じです。

ロバストネス図を中心とした ICONIX 手法を提唱した **Rosenberg** や、見積もり手法の一つであるユースケースポイント法を提唱した **Schneider** は、それぞれ、

「アクタが目的を達成するためのシステム内における振る舞いのシーケンス」(R)

「外部から見てわかるシステム要求を記述するために使うもの」(S)

と、大雑把です。

Cockburn は、ユースケースの粒度を「概略レベル」「要約レベル」「ユーザ目的レベル」「サブ機能レベル」「詳細レベル」の5段階に分類していますが、「システムの振る舞いに関する利害関係者の契約の表現。または主アクタの要求に対してシステムが反応するときの様々な状況におけるシステムの振る舞いの記述」と、契約関係を持ち出して説明しています。

粒度については、私も **Cockburn** を参考に「概念レベル」「目的レベル」「手段レベル」に分類しています。

概念レベルは、ユースケースパッケージに相当します。サブユースケースに相当する多数のバリエーションを包含します。

目的レベルは、ひとつ以上のアクタと相互作用して目的を達成するものです。ひとつの基本フローと0個以上の代替フローを持ちます。

手段レベルは、ひとつのアクタとしか相互作用しない単位です。データアクセスの基本単位であるCRUDひとつひとつは、このレベルになります。

◆ユースケース記述

ユースケースは、もともと、Jacobson の母国語であるスウェーデン語の「使用の状況」「使用事例」を英訳する際に使われた言葉だそうです。現在では「シナリオ」と呼ばれているものが、オリジナルの目的に一番近いといえるかも知れません。

多くの場合、ユースケース記述にはテキストや表形式が使われますが、MindMap を使うとユースケース全体を俯瞰しやすく、便利かと思います。

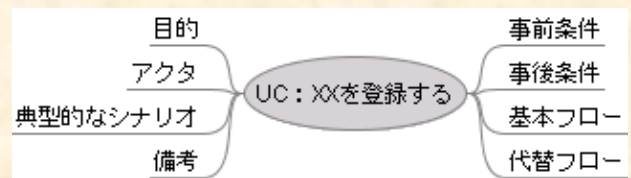


図 1 MindMap によるユースケース記述

事前条件と**事後条件**はオプション扱いされることがありますが、必須項目です。これらは、ユースケースを実行する前後の「システムの状態—ユースケース実行により影響を受けるエンティティの状態」です。

事前条件・事後条件でユースケースをブラックボックスとしてとらえることができるようにすることで、テスト設計を誘導することができます。

◆◆◆◆ コラム ものの伝え方 ◆◆◆◆

産業第三事業部チーフコンサルタント 中島亜澄

「どうして無理な要求ばかり言うのだろう」「この方がよいやり方なのに、なぜ意見を聞き入れてもらえないのだろう」
このような声が周りからよく聞こえてきます。そして私自身も、このように悩んでいたことがありました。

私の場合、相手の言うことをよく「聞く」そして「想像してみる」ことで、楽に自分の意見を通したり、相手のことを理解することが出来るようになりました。

同じ人間でも、「対お客さま」「対上司」「対チームメンバ」というように、相手によって立場が変わります。
今から自分が意見を伝えたい相手は、どういう立場なのか。どういうミッションを持っているのか。どういう性格なのか。
会議での様子や世間話から、それらをうまく「聞き出す」努力をします。意外と難しいのですが、思い込みやフィルターをかけずに素直に聞き取ることを大切にしています。

データがそろったら、相手の立場を想像して頭の中で相手になりきってみる。そうすることで、どのように伝えると相手の心に響くのか、ということが分かってきます。

論理的な説明がよいのか。易しい例え話が効果的なのか。あるいは持ち上げて気分をよくしてもらった上で伝えるのがよいのか。

『伝えたいこと』はひとつでもそれを『どうやって』伝えるか。

伝える道は何通りもあります。相手に合った道を選ぶことで、無用な角を立てず、お互いの心に優しいコミュニケーションとなり、今まで聞き入れてもらえなかった意見も易しく受け入れてもらえるようになる。

自分の体験を通して確信しています。

GSLetterNeo Vol. 4

2008年11月10日発行

発行者 ●株式会社 SRA 産業第三事業部

編集者 ●土屋正人、板沢智雄、柳田雅子、小嶋勉、野島勇

ご感想・お問い合わせはこちらへお願いします ●gsneo@sra.co.jp

株式会社SRA

〒171-8513 東京都豊島区南池袋2-32-8