

差出人: Naruaki TOMA <tnal@ie.u-ryukyu.ac.jp>
件名: [Slab34-ec:00329] UML(ユースケース図、ユースケース記述) 補足
日時: 2012年10月18日 18:10:51 JST
宛先: Evolutionary Computing <slab34-ec@ie.u-ryukyu.ac.jp>
返信先: Evolutionary Computing <slab34-ec@ie.u-ryukyu.ac.jp>

當間@情報工学科です。

単一の記述方法で全てを賅う事はできないのが一般的なので、
複数の記述方式を併用しながら抜け確認や共通機能洗い出し等を進める体験をしてみよう。

今回は設計の第一歩として外部設計+内部設計の途中ぐらいまでをやりました。
具体的にはマインスイーパーを例に、
ユースケース図を考え、
簡易的な外部設計として全体フローをフローチャートで捉え直し、
機能項目毎にユースケース記述を設計してみました。

ユースケース図 (写真)、フローチャート (写真)、ユースケース記述 (MindMap)
は、

```
hg clone ssh://info3@shark//home/info3/HG/2012/share2/  
cd MineSweeper
```

に置きました。

来週は今回のユースケース記述をベースに、実装 + α をやる予定です。
以下、補足情報です。

>ユースケース図
システムが持つユースケースを列挙し、それがどのアクターから利用され、
ユースケース同士の関係がどうなっているのかを一覧する図 [1,2]。
作ろうとしているシステムは、ユーザに対して何を提供するのか (=ユースケース)
を明確にするための図。

>ユースケース記述
アクターの目的を達成するために、システムが実行する (すべき) 振る舞い
を文章で記述したもの。ユースケーステキスト、シナリオという場合もある [3,4]。
UMLとして標準化はされておらず、どのような項目があれば十分にユースケース
を表すことができるかはケース・バイ・ケースであり、よく検討する必要がある。

今回の例:
-ユースケース名 (=概要)
-前提条件
-処理手続き
-事後条件

[1] ユースケース: 問題の分析: <http://think-on-object.blogspot.jp/2011/11/top-12-actor-3-31-3-top-top-123-top-2-1.html>
[2] ユースケース記述の例, <http://think-on-object.blogspot.jp/2011/11/uml2.html>

参考図書

- (a) O'Reilly Japan - 入門 UML 2.0 <http://bit.ly/txKUXC>
- (b) O'Reilly Japan - Head Firstソフトウェア開発 <http://bit.ly/uPcvft>
- (c) Amazon.co.jp: ずっと受けたかったソフトウェアエンジニアリングの授業1 増補改訂版: 鶴保 征城, 駒谷 昇一: 本 "増補改訂版" <http://amzn.to/spTGeO>
* (b,c)はどちらもUMLに関する解説は少なく、設計全般に関する話がメインです。
* 一応(b)は巻末で触れる程度の解説 (数ページ) はあります。

Naruaki Toma
E-mail: tnal@ie.u-ryukyu.ac.jp, Tel: 098-895-8830
<http://www.eva.ie.u-ryukyu.ac.jp/~tnal/>

Slab34-ec mailing list
Slab34-ec@ie.u-ryukyu.ac.jp
<https://ginowan.ie.u-ryukyu.ac.jp/mailman/listinfo/slab34-ec>