

# 一种基于关注点的用例模型的逆向恢复方法

崔伟勇, 邬丽红, 曹 翀

(西安电子科技大学 软件工程研究所, 陕西 西安 710071)

**摘 要:**用例作为展现程序系统级行为的有效手段,可以辅助理解程序的功能特征。针对面向对象系统提出了一种基于关注点的用例模型逆向生成方法。该方法分析目标系统运行时的动态信息,把其中方法调用序列的起始方法看作基本用例,使用一定的规则识别出基本用例间的关系来合并生成用例,依据从用户的关注点中获取到的有用信息,对生成用例进行改进,从而实现了用例模型的逆向生成。通过实验分析,表明该方法在恢复用例模型并提高其准确性方面是有效的。

**关键词:**逆向工程;用例模型;关注点;动态分析

**中图分类号:**TP311

**文献标识码:**A

**文章编号:**1673-629X(2008)09-0034-03

## An Approach of Recovering Use Case Models Based on Concerns

CUI Wei-yong, WU Li-hong, CAO Chong

(Software Engineering Institute, Xidian University, Xi'an 710071, China)

**Abstract:** As an effective means of representing the systemic behavior of program, use case is helpful for users to comprehend the functionality of program. In this paper, an approach of recovering use case models based on concerns is proposed for object-oriented systems. By analyzing the dynamic information of target system, in which the beginning methods of method calling sequences are considered as primitive use cases, the approach employs some rules to identify the relations of the primitive use cases to build use cases and improves on the built use cases according to the useful information obtained from the user's concerns, thereby the recovery of use case models is implemented. By a case study, the effectiveness of the approach in recovering use case models and improving their veracity is verified.

**Key words:** reverse engineering; use case models; concerns; dynamic analysis

## 0 引言

文中针对面向对象系统,提出了一种基于关注点的用例模型<sup>[1]</sup>逆向生成方法,该方法已经在逆向工程工具 XDRE(XiDian Reverse Engineering)中实现。它首先分析面向对象系统运行时的动态信息,抽取出其中蕴涵的用例<sup>[2,3]</sup>,然后利用用户提供的关注点获取相关信息,对抽取出的用例进行改进,使其能够更准确地体现系统的行为特点。

## 1 用例模型的逆向生成

### 1.1 基于动态信息的用例模型抽取

对于面向对象系统,程序运行时的动态信息对理解它的行为特征是十分重要的。动态信息是程序一次动态运行时对象间方法调用的信息,它从运行时的角

度反映了目标系统内静态的方法调用的特点,从而在某种程度上体现了系统的行为特征,而用例是系统或系统一部分的行为描述<sup>[4]</sup>,因此,动态信息是抽取用例模型很好的信息源。

用例模型抽取的主要流程如图1所示。

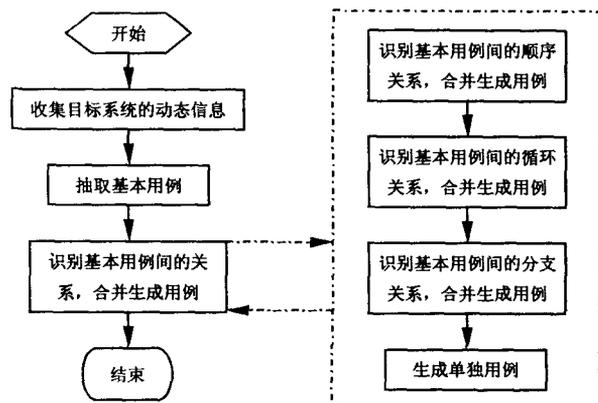


图1 用例模型抽取的主要流程

图1中灰色模块是整个流程中最主要的模块,具体实现为虚线框图中的四个步骤,其中又以前三个步骤最为重要。要注意的是,这三个步骤是有先后顺序

收稿日期:2007-12-13

基金项目:国家自然科学基金资助项目(60473063);“十五”国防预研项目(41306060106)

作者简介:崔伟勇(1975-),男,硕士研究生,讲师,研究领域为软件体系结构、逆向工程、面向对象技术。

的,因为根据基本用例间关系的识别规则和具体的实现算法,只有先识别了所有的顺序和循环关系后,才能使分支关系的识别更彻底。

## 1.2 基于关注点的用例模型的改进

### 1.2.1 用例模型改进的必要性

由于动态信息本身的不确定性和代码的复杂性,使得从动态信息中抽取到的基本用例序列会呈现各种可能的情况,而基本用例序列本身所能提供的用于关系识别的信息有限,所以不可能完全准确地恢复出所有基本用例间应有的关系,这就需要进一步获取相关信息以改进用例模型。

### 1.2.2 关注点方法间关系的获取

由于关注点方法间的关系信息蕴涵在动态信息中,而动态信息中的关注粒度是每个方法调用序列的起始方法,所以可以对由动态信息获取的基本用例序列进行过滤,抽取其中与关注点信息有关的基本用例,构成关注点基本用例序列,再利用基本用例间关系的识别规则对其间的关系进行识别,这样就可以获取到关注点方法间的关系信息。由于用户关注点方法之间本身存在某种确定的关系,且关注点基本用例序列中只有与关注点方法有关的基本用例,因此,可以通过这种方法获取到的关系信息比较准确。

### 1.2.3 基于关注点的用例模型的改进规则

利用获取到的关注点方法间的关系信息,可以对抽取出的用例模型进行改进。不妨用向量  $C = (c_1, c_2, \dots, c_r)$  表示关注点基本用例序列,  $R$  表示获取到的关注点方法间的关系,  $U = (U_1, U_2, \dots, U_m)$  表示抽取出的用例集合,则改进规则描述如下:

规则 1:若  $R = \text{SEQUENCE}$ (将  $c_1, c_2, \dots, c_r$  作为一个用例  $U_c, U_c. \text{INC} = (c_1, c_2, \dots, c_r)$ ,对于  $U_i (i \in [1 \dots m])$ ):

1)  $U_i. R = \text{SEQUENCE}$ ,若  $U_i. \text{INC} \cap U_c. \text{INC} \neq \emptyset$ ,则摒弃  $U_i$ ;

2)  $U_i. R = \text{LOOP}$ ,则  $U_i. \text{INC} \cap U_c. \text{INC} = \emptyset$ ,不做任何操作;

3)  $U_i. R = \text{BRANCH}$ ,若  $U_i. \text{EXT} \cap U_c. \text{INC} = (c_1)$ ,则用  $U_c$  替换  $U_i. \text{EXT}$  中的  $c_1$ ;否则,若  $U_i. \text{EXT} \cap U_c. \text{INC} \neq \emptyset$ ,则摒弃  $U_i$ 。

规则 2:若  $R = \text{LOOP}$ (将  $C$  中具有循环关系的子序列  $c_1, c_2, \dots, c_r (r' < r)$  作为一个用例  $U_c, U_c. \text{INC} = (c_1, c_2, \dots, c_r')$ ,对于  $U_i (i \in [1 \dots m])$ ):

1)  $U_i. R = \text{SEQUENCE}$ ,则  $U_i. \text{INC} \cap U_c. \text{INC} = \emptyset$ ,不做任何操作;

2)  $U_i. R = \text{LOOP}$ ,若  $U_i. \text{INC} \cap U_c. \text{INC} \neq \emptyset$ ,则摒弃  $U_i$ ;

3)  $U_i. R = \text{BRANCH}$ ,若  $U_i. \text{EXT} \cap U_c. \text{INC} = (c_1)$ ,则用  $U_c$  替换  $U_i. \text{EXT}$  中的  $c_1$ ;否则,若  $U_i. \text{EXT} \cap U_c. \text{INC} \neq \emptyset$  或者  $U_i. \text{Mai} \in U_c. \text{INC}$ ,则摒弃  $U_i$ 。

规则 3:若  $R = \text{BRANCH}$ (将分支关系中的主用例  $c_1$  作为一个用例  $U_c, U_c. \text{Mai} = c_1, U_c. \text{EXT} = \text{SUC}(c_1)$ ,对于  $U_i (i \in [1 \dots m])$ ):

1)  $U_i. R = \text{SEQUENCE}$ ,若  $U_i. \text{INC} \cap U_c. \text{EXT} \neq \emptyset$ ,则摒弃  $U_i$ ;

2)  $U_i. R = \text{LOOP}$ ,若  $U_i. \text{INC} \cap U_c. \text{EXT} \neq \emptyset$ ,则摒弃  $U_i$ ;

3)  $U_i. R = \text{BRANCH}$ ,若  $U_i. \text{Mai} = U_c. \text{Mai}$  或者  $U_i. \text{EXT} \cap U_c. \text{EXT} \neq \emptyset$ ,则摒弃  $U_i$ 。

规则 4:对于单独用例,若其与关注点基本用例相同,则摒弃该用例。

### 1.2.4 基于关注点的用例模型改进流程

用例模型改进的主要流程如图 2 所示。

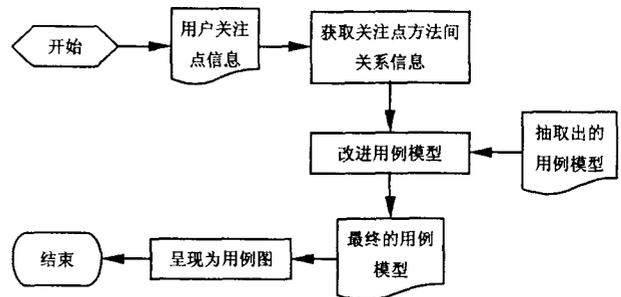


图 2 用例模型改进的主要流程

在图 2 所示的流程中,灰色模块所示的是其中最主要的一步,利用获取到的关注点方法间的关系信息,对抽取出的用例模型进行改进。

## 2 实验研究

基于关注点的用例模型逆向生成方法已经应用在逆向工程工具 XDRE 中,为了验证该方法的有效性,文中采用自主开发的 Windows 平台下的高速公路客运活动仿真程序作为测试用例。该测试用例是用 C++ 语言实现的面向对象系统,有 31 个类,16325 行代码,主要功能是对两个城市间的高速公路上客车运行情况 and 乘客乘车情况进行仿真。

由 XDRE 采用基于反射的植入方法可以收集到程序运行时的动态信息<sup>[5]</sup>,实验采用对目标系统进行植入收集到的以 XML 文件形式记录的部分动态信息。XML 文件的层次关系对应着方法间的调用关系,遍历该文件抽取方法调用序列的起始方法作为基本用例,可以形成目标系统的基本用例序列。图 3 显示了 XDRE 使用 Rational Rose 呈现的用例模型。由于篇幅所限,图中只展示了其中的两个用例。

利用用户指定的关注点信息过滤基本用例序列得到关注点基本用例序列,再使用关系识别规则就可以获取到关注点方法间的关系信息。根据这个信息对抽取出的用例模型进行改进,最终得到的用例模型如图 4 所示。受篇幅所限,图中只展现了与图 3 中的用例相关的部分改进后的结果。

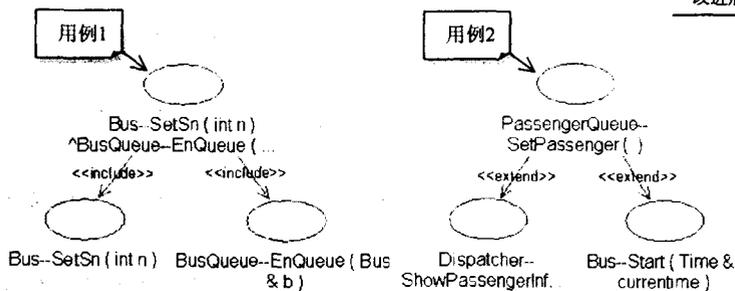


图 3 基于动态信息抽取出的用例模型

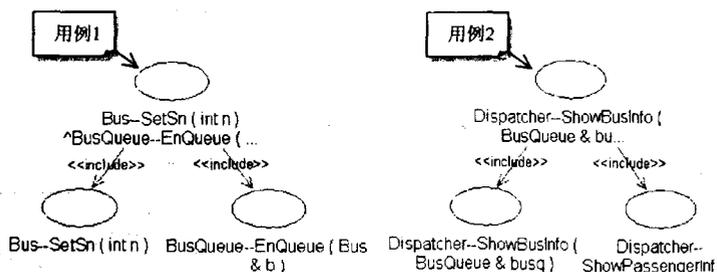


图 4 基于用户关注点改进的用例模型

在图 4 中,PassengerQueue & pasq 这个用例有两个包含用例 Dispatcher - ShowBusInfo (BusQueue & busq) 和 Dispatcher - ShowPassengerInfo (PassengerQueue & pasq),这两个包含用例所表示的两个方法正是用户指定的关注点,可见该用例是对用户关注点信息用例恢复的结果。由用例抽取过程可知两个方法之间有顺序或循环关系,通过分析目标系统发现它们之间是循环关系,因此,该用例准确反映了关注点的行为特点。与图 3 中的用例 2 相比,该用例准确涵盖了基本用例 Dispatcher - ShowPassengerInfo (PassengerQueue & pasq) 所实现的功能。其实,某种意义上基于用户关注点的用例模型改进策略是对人工修改用例模型的自动化。

对比改进前后的用例模型发现,原来可以准确反映目标系统行为特点的用例被保留了下来(如图 3 中的用例 1),而一些没有准确反映出系统行为特点的用例则被摒弃(如图 3 中的用例 2),最后增加了反映用户关注点信息的用例(如图 4 中的用例 2),可见通过

改进,提高了原有用例模型的准确性。表 1 对用例模型改进前后的情况作了具体对比。

表 1 用例模型改进前后的情况对比

	总的用例个数	准确反映系统行为特点的用例个数	准确率
改进前	15	8	53.3%
改进后	14	9	64.2%

从表 1 可以看出,改进后的准确率提高了 10.9%,如果用户增加关注点信息,可以获取到更多的方法间关系信息,从而可以更准确地恢复出目标系统的用例模型。

### 3 结束语

文中提出的基于关注点的用例模型逆向生成方法,是一种动态分析的方法,它利用动态信息抽取用例模型,然后结合用户的关注点信息对其进行改进,从而实现了用例模型的逆向生成。不过,这种方法所得到的用例模型是源于代码的,属于低层次的、反映系统一定行为特征的用例模型,它与传统意义上正向工程中的用例模型有一定的差距,但这种方法所恢复的用例模型对用户理解目标系统提供了有效的帮助,这也正是逆向工程的意义所在。

在下一步的工作中,针对动态信息的特点,可以对用例进行迭代累加式的逆向生成,通过目标程序的多次运行来获取动态信息的并集,从而使得逆向的结果更完善、更全面。

#### 参考文献:

- [1] Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language User Guid[M]. [s.l.]: Addison - Wesley, 1999.
- [2] 邬丽红,陈平.一种多进程系统用例模型的逆向生成方法[J].微电子学与计算机,2006,23(11):191-194.
- [3] 齐微,李青山,陈平.多机并发系统动态信息的逆向抽取和过滤策略[J].系统工程与电子技术,2006,28(9):1423-1430.
- [4] Jacobson I. Basic Use - Case Modeling[J]. Report on Object Analysis and Design, 1994,1(2):15-19.
- [5] 李青山,陈平,王伟,等.逆向工程中反射植入的研究[J].计算机学报,2004,27(4):535-542.

中国计算机学会会刊、中国科技核心期刊  
《计算机技术与发展》欢迎订阅,邮发代号:52-127。