

一种面向方面的软件资源特征分析方法

阳王东

(湖南城市学院 信息科学与工程学院, 湖南 益阳 413000)

摘要:针对互联网环境下存在丰富多样的软件资源,为了能够有效地重用已有的软件资源,探讨一种基于方面化特征对可用软件资源的软件分析方法。这种方法从所关注的程序切面来分析软件所能提供的功能特征,从而得到软件资源的规格描述,并通过切面结构分析建立使用软件资源的调用关系和织入点,为现有的软件资源提供一种逻辑结构描述及重用方法。通过该种软件分析方法,能够对现有异构的软件资源的重用提供一种逆向工程方法,从而能提高软件资源复用中的可用性。在对互联网中日志组件的分析中能够得到验证。

关键词:面向方面;特征化;软件分析;逻辑语言

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2012)11-0153-04

A Features Analysis Method of Software Resources Aspect-oriented

YANG Wang-dong

(Department of Information Science and Engineering, Hunan City University, Yiyang 413000, China)

Abstract: For a variety of software resources in the Internet environment, an analysis method based on aspects feature of the available software resources was explored in order to effectively reuse existing software resources. This approach from the concerns of the program section analyzes the functional characteristics the software can provide, which describe the specifications of software resources, and the call between using software resources and the weaver of software was built by section structure analysis to provide a logical schema and reuse existing software resources. A reverse engineering approach could be provided by the software analysis methods for the reuse of existing heterogeneous software resources, which can improve the availability of software resources reuse. The approach could be verified by analysis on log components in the Internet.

Key words: aspect-oriented; features; software analysis; logic language

0 引言

随着计算机应用的普及和信息化建设的深入,大量软件系统被开发出来,而且随着 Internet 的普及,逐渐改变了人们传统的获取软件的方式与途径,越来越多的软件被放置到 Internet 上,有些是为了销售、有些是提供各种软件服务,另外还有大量开源软件、免费软件。Internet 中放置的软件有些是有成熟应用的完整软件系统,有些是商业化软件产品,还有些是提供交流和学习的软件系统和组件。这些软件的形式,有些是可直接运行的执行程序,有些是编译好的软件组件,有些是提供接口的软件服务,还有些是软件源代码^[1]。现在几乎每一种应用系统,都可以在 Internet 上找到相

似的软件系统,例如 OA 系统在 Internet 上有不下几百种,而且其中有一部分还提供源代码。在 Internet 中像这样的软件资源浩如烟海,其实这些软件资源针对特定问题开发出来的,也是满足相应的用户需求,如何利用现有的软件资源,是软件开发人员都十分关注的问题^[2]。

在开发软件的时候,定义应用需求时一般采用非形式化的方式来描述,它易懂,容易描述,但是缺乏严格的可验证性。但根据软件需求规格来设计和实现软件时,则必须是严格的、可验证的,因此从需求到实现之间的过渡,就容易出现不一致性,尤其对于现有的软件资源如何描述在规格与实现之间关系,就变得更为困难^[3]。现在软件工程中强调迭代和敏捷的开发方法,如何更加高效地开发甚至复用已有的软件资源自动生成软件始终是软件工程追求的目标之一,软件产品线(SPL)被认为是一种可行的解决方案。方面化特征模块(Aspectual Feature Module, AFM)编程范式是一种最新提出的 SPL 编程范式,它解决了现有编程范

收稿日期:2012-04-03;修回日期:2012-07-08

基金项目:湖南省自然科学基金项目(11JJ6052);湖南省科技计划项目(2011GK3124);湖南省教育科研项目优秀青年项目(11B023)

作者简介:阳王东(1974-),男,湖南衡南人,硕士,副教授,研究方向为软件工程。

式一面向特征编程范式(Fecture Oriented Programming, FOP)存在的问题^[4,5]。传统编程范式(例如面向对象编程范式)中,程序的组成模块(例如 class)对应系统的参与者(也称为角色);FOP 编程范式下,程序的组成模块对应系统的特征(即可以为用户感知的功能),在许多领域,将系统基于特征而非角色划分能够获得重用性更强的设计。但 FOP 在表达同构横切时非常繁琐,并且无力表达复杂的动态横切,这限制了它全面表达各类特征的能力。AFM 范式引入了 AOP 编程范式(Aspect Oriented Programming, AOP)的相关设施解决了上述问题^[6-8]。AFM 范式要求系统的开发者首先开发一个只实现系统基本功能的最简单的程序,这个程序通常被称为基程序;然后开发一组 AFM 模块实现系统的其他功能。这样,针对不同用户的需求,开发人员挑选实现了相应功能的 AFP 模块,将它们与基程序组合,从而得到满足用户需求的程序。但是如何有效快速从现有软件资源获取所需的 AFM 对于软件资源的利用非常重要,LMP(Logic meta programming, LMP)逻辑元程序描述语言恰好为规格与实现提供一条连接的纽带,它不但能够对程序规格提供描述,还能对设计模式、程序行为等进行描述,是分析现有软件资源从而得到可用的 AFM 描述的一种有力的工具^[9-11]。

文中探讨一种利用 LMP 基于 AFM 对软件资源的一种分析方法,为已存在的软件资源提供一种规格描述,从而提高软件资源复用中的可用性。

1 软件功能特征的方面化

1.1 AFM 的定义

传统的以对象为核心的软件模块是基于软件的内部构成结构来描述的。而使用者关注的是外部体现的服务,AFM 体现的是这种服务,服务包括服务契约、服务规范和服务发现。因此从服务的角度来描述 AFM,应该包含三个方面的构成元素:

AFM = (Concerns, Pointcuts, Facts)

其中 Concerns 为关注点,关注点是识别、区分和使用 AFM 的重要标识,类似于服务发现。

Pointcuts 为切入口,它相当于服务契约,是导入使用 AFM 的接口。

Facts 为业务规则,相当于服务规范,是用来描述 AFM 的功能规格的。

AFM 作为一种编程范式,并不是一种具体的组件模型,而是程序的一种范式描述。对于要重用已存在的软件资源(软件资源可能包括源代码,可运行程序以及各种组件),首先要先认识它,了解它的功能特征,AFM 能够提供一种功能特征的描述方式。以 AFM

范式来描述软件资源,要满足以下范式要求:

(1)完备性。AFM 所包含的三个构成元素缺一不可。

(2)独立性。AFM 所描述的软件资源具有独立存在的实体单元。

(3)可用性。AFM 描述的软件资源具有可导入性,可组合性。

1.2 软件功能特征的分割

AFM 是面向方面来描述软件资源的功能特征,对于实际的一个软件资源,其特征是多角度的,如何从方面的视角来看待软件的功能特征,就需要对软件功能特征从某些特定方面进行分割,得到满足上述范式要求的 AFM。由于软件体系结构的不同,面向服务对象的不同,其分割的视角也会不同^[12-14]。

下面针对几种典型的软件模型探讨分割的方法。

1.2.1 多层模型的分割

对于复杂的软件系统,对其进行分层是降低复杂性的一种典型的方法。采用多层模型的软件资源在复用当中粒度较大,需要进行分割。对于多层的软件资源,可采用一种横向切面分割的方法。

根据软件的分层来分割软件资源的特征,由于软件资源的功能实现是通过各层协同工作来完成,所以其中一层难以体现完整的功能实现,在分割特征时,应采用产品流水线中不同的工序环节来分割,从分层服务来定义每一层的功能特征。层数的数量与规模以每一层特征既具有实现的单一性有体现了服务的完备性为原则。这些特征一般为软件在宏观上的方面,属于一种体系结构方面。

1.2.2 插件模型的分割

插件模型是利用一个共有的程序框架,可以导入其他的软件单元,这些软件单元称之为插件。插件模型是一种树形分割。

插件是扩展软件功能特征的一种形式,一个插件所实现的是一项功能,对于同一个插入接口的插件有着相同的服务契约,也就是有相同的织入方式。这种特征分割是一种斜面分割方法。插件特征一般是具体微观上的方面,属于一种实现上的方面。

1.2.3 管道模型的分割

管道模型一般由四种类型的管道构成:导管、源、槽和联结。每一种管道有着相似的特征。按照管道类型进行分割,可以得到导管、源、槽盒连接四种类型的 AFM 描述。

1.2.4 交互模型的分割

交互模型其实是一个消息响应模型。请求交互方给响应交互方发送请求消息,响应交互方进行交互响应并返回响应结果消息。

交互模型以发生的消息来分割,每一种消息体现了一种行为特征,而一次请求和响应的交互过程是一种较为完整和独立的过程,比较符合 AFM 范式要求。以这种行为特征来进行方面划分属于一种行为方面。

2 基于 AFM 对软件资源的分析与描述

2.1 AFM 的逻辑描述语言

在 LMP 当中,使用一种逻辑程序语言来阐述已存在软件资源的基程序,这种关于基语言程序的元描述由逻辑事实和规则构成。在 AOP 的上下文当中,这种逻辑语言也作为一种基础环境服务嵌入到 AFM 当中。它提供了 AFM 的描述与实现的通用框架,这样为 AFM 的利用提供一种高层抽象。

2.1.1 逻辑描述语言的语法

一种描述方面的语言一系列逻辑谓词的集合,对于 AFM 而言,则需要一种通用的能体现特征状态的方面语言。在这种语言中,一种方面特征是这些谓词的逻辑声明集合,在 AFM 模型中,则包含一个逻辑模块。

这种逻辑描述语言的语法包含以下几个方面:

(1)在逻辑描述的基程序文本中嵌入特定的逻辑术语(用大括号分割“{}”),这个术语甚至可以包含逻辑变量以类似的方式复合逻辑上可以嵌套包含他们。

(2)逻辑变量以一个问号开始,例如? 变量名。

(3)AFM 由逻辑声明封装,一个 AFM 由逻辑描述语言对其构成的三元素进行声明描述,从而形成 AFM 的逻辑定义。

(4)AFM 能够由其他的 AFM 的逻辑声明组合而成,也可设置自身的声明能够给其他 AFM 可见,这样其他 AFM 也可以引用该 AFM 的声明。

2.1.2 AFM 逻辑描述语言的样例

对于一段软件代码 c,它具有一个功能特征 f,基程序中的一个方法 m 中需要使用代码 c 的功能特征。代码 c 可以作为一个 AFM 来进行声明 c(f,m,logics),利用逻辑描述语言对该 AFM 的特征状态的描述如表 1 所示:

表 1 AFM 的特征状态描述

逻辑谓词	描述
adviceBefore(? m,? c->f)	在执行方法 m 之前导入 AFM c 的功能特征 f
adviceAfter(? m,? c->f)	在执行方法 m 之后导入 AFM c 的功能特征 f

2.2 AFM 的关注点

AFM 的关注点是从使用者的角度看待的功能特征。而对于 AFM 自身而言,则是一组服务的集合。服

务集合的描述对象与 AFM 的基程序类型不同而不同。AFM 关注点的描述形式为:object->f。其中 object 为关注对象体,关注对象体可以使代码段、组件、运行程序等多种软件资源基程序;f 是具体的关注点,具体的关注点可以是功能函数、服务接口、消息事件等。对于一个写日志的组件,例如 Java 中的 log4J,对于使用者而言,关注点主要有两个:

(1)写日志。Log4J 存在一个 Log 类,类中有一个写日志的 log 方法,该关注点可以描述为 methodInClass (? log,? Log)=Log->log。

(2)日志所写入存储的位置。Log4J 有一个配置文件 log4j. properties,配置文件中有一个配置项 log4j. appender. file,该关注点可以描述 statementsOfMethod(? log4j. properties ,? log4j. appender. file)= log4j. properties-> log4j. appender. file。

2.3 AFM 的切入点

在方面语言中,连接点(joinpoints)是描述在一个方面中何时何地激发该方面的功能特征。切入点(Pointcuts)则是连接点的集合。在 AFM 中,连接点是基程序中的一个特定接入口。如何才能把 AFM 导入或组合到基程序中,切入点非常关键。

切入点的切入方式有两种:

(1)同步切入。同步切入是把 AFM 导入到基程序进程中,在同一进程执行 AFM。例如上述样例中 adviceAfter(? m,? c->f,? onBlock)为同步切入,在基程序中的方法 m 之后同步执行 AFM c 的功能特征 f。

(2)异步切入。异步切入方式中,AFM 与基程序之间为异步通信关联,并非运行于同一进程之中。上述样例中 adviceBefore(? m,? c->f,? onStart)为异步切入,在基程序中的方法 m 之前同时开始执行 AFM c 的功能特征 f,方法 m 也同时执行并不阻塞等待 f。

定义 AFM 的切入点,要注意 AFM 与切入的基程序之间的纠缠状态。为了 AFM 的独立性,切入点的定义要注意三个原则:

(1)透明性。基程序与 AFM 的切入在源代码级上是透明的,不要存在代码纠缠现象。

(2)互不干涉性。基程序与切入的 AFM 在运行状态之间尽量互不干涉,不要存在运行态纠缠现象。

(3)可替换性。AFM 切入基程序的切入点,应该允许满足类似功能特征的 AFM 切入进来。

3 案例研究:日志 AFM 程序的分析

3.1 日志 AFM 的功能特征分析

日志是一种非常普遍通用的功能特征,使用日志的主要关注点是如何写日志,写日志一般是在一个操作进行的前后或一个事件发生的时刻,但是并不是所

有操作和所有事件发生时都要写日志,日志存在不同的级别,达到某一级别或满足某一条件才会触发写日志,因此在定义日志 AFM 的功能特征时,需要定义写日志的条件参数。

3.2 日志 AFM 的切入点

写日志的切入点是在某一通知发生的前后进行,这里增加了对通知触发的参数有条件切入点的定义。

```
@Pointcut("((adviceBefore(? m,? c->log) || adviceAfter(? m,? c->log)) || ((adviceBefore(? m,? c->log,? c->log equal ? para-name) || (adviceAfter(? m,? c->log,? c->log equal ? para-name)))

@Before[? m,? c->log]
Call before(JoinPoint joinPoint, Loggable loggable);

@Before[? m,? c->log,? c->log equal ? para-name]
Call before(JoinPoint joinPoint, Loggable loggable); //切入的织入点

@After[? m,? c->log,? c->log equal ? para-name]
Call after(JoinPoint joinPoint, Loggable loggable, Object returnValue); //切入的织入点

@After[? m,? c->log]
Call after(JoinPoint joinPoint, Loggable loggable, Object returnValue);
```

3.3 日志 AFM 织入

在宿主程序当中,如何织入日志 AFM 的功能特征,是通过连接点来触发的。

下面定义了一个在通知前后的切入点织入日志 AFM 的功能特征织入程序。

```
@Before(? value = "@ annotation(loggable)", ? argNames = "joinPoint, loggable")
public void before(JoinPoint joinPoint, Loggable loggable)
{
    Class clazz = joinPoint.getTarget().getClass(); //反射类
    String name = joinPoint.getSignature().getName();
    if (ArrayUtils.isEmpty(joinPoint.getArgs()))
    {
        logger.log(loggable.value(), clazz, null, BEFORE_STRING, name, constructArgumentsString(clazz, joinPoint.getArgs()));
    }
    else { logger.log(loggable.value(), clazz, null, BEFORE_WITH_PARAMS_STRING, name, constructArgumentsString(clazz, joinPoint.getArgs())); }

    @After(? value = "@ annotation(trace)", ? returning = "returnValue", ? argNames = "joinPoint, trace, returnValue")
    public void after(JoinPoint joinPoint, Loggable loggable, Object returnValue)
    {
        Class clazz = joinPoint.getTarget().getClass(); //发射类
        String name = joinPoint.getSignature().getName();
        if (joinPoint.getSignature().instanceof MethodSignature)
        {
            MethodSignature signature = (MethodSignature) joinPoint.getSignature();
            Class returnType = signature.getReturnType();
            if (returnType.getName().compareTo("void") == 0)
```

```
{ logger.log(loggable.value(), clazz, null, AFTER_RETURNING_VOID, name, constructArgumentsString(clazz, returnValue)); }
return; } |

logger.log(loggable.value(), clazz, null, AFTER_RETURNING, name, constructArgumentsString(clazz, returnValue)); }
```

4 结束语

在互联网环境下,存在各种形态的软件资源,如何才能有效地利用这些软件资源,对其进行特征分析是利用的前提,文中基于方面化思想的基础上,结合特征化模块划分方法,利用 LMP 描述语言,不但为多种软件资源提供一个较为通用的描述模型,而且也提供一个抽象能力较高的逻辑描述方法。

这种方法为能够有效快速重用现有软件资源提供一种可信的途径。但是由于互联网环境下软件资源的多样性和异构性,如何提高对这些资源重用的能力和效率,在对这些资源的认识上更需要更广泛的适应性和可用性,因此如何提高对异构软件资源的描述能力及效率是下一步研究的重点。

参考文献:

- [1] 刘国梁,魏峻,冯玉琳. 基于组件模型分析的组件容器产品线体系结构[J]. 软件学报,2010,21(1):68-83.
- [2] 蔡斯博,邹艳珍,邵凌霄. 一种支持软件资源可信评估的框架[J]. 软件学报,2010,21(2):359-372.
- [3] 梅宏,王千祥,张路. 软件分析技术进展[J]. 计算机学报,2009,32(9):1698-1710.
- [4] 张伟,梅宏. 一种面向特征的领域模型及其建模过程[J]. 软件学报,2003,14(8):1345-1356.
- [5] 叶俊,谭庆平,李瞰,等. 基于层次状态机的方面化特征模块的增量式验证[J]. 计算机学报,2009,32(9):1773-1781.
- [6] 吴帅,张立臣. 基于信息物理系统的面向方面中间件特征模型[J]. 计算机技术与发展,2012,22(2):76-79.
- [7] 汤根生,王姚. 面向方面状态模型的 UML 扩展实现[J]. 计算机技术与发展,2011,21(1):116-119.
- [8] 李婷,刘建勋,尹雁青. 面向方面建模方法的研究及其应用[J]. 计算机技术与发展,2009,19(1):113-116.
- [9] Bravo F M. A Logic Meta-programming Framework for Supporting the Refactoring Process[D]. Brussel: Vrije Universiteit Brussel, 2003.
- [10] de Roover C. A Logic Meta Programming Foundation for Example-driven Pattern Detection in Object-oriented Programs[C]//IEEE 27th International Conference on Software Maintenance. [s. l.]:[s. n.], 2011:556-561.
- [11] Brichau J, Mens K, de Volder K. Building Composable Aspect-specific Languages with Logic Metaprogramming[J]. Lecture

(下转第160页)

The results are listed in Table 2. First, from the table 2, see that the detection rates of T-KMOD is the highest. Second, the detection rate and the false alarm rate are two constraining indicators, so a higher detection rate may result in higher false alarm rate. T-KMOD has the highest detection rate, however it has a higher false alarm rate than KMOD, it's lower than POLY, and still in the acceptable range. Third, as shown in Table 2, the training time of T-KMOD is the least, which indicates T-KMOD has good performance in reducing the training time.

Table 2. Comparison of the performance of the four kernel functions

Set	C-SVM (kernels)	Detection rate(%)	False alarm rate(%)	Train time(s)	Test time(s)
DS1	POLY	84.134	15.866	15.350	7.842
	RBF	83.544	16.456	13.739	10.948
	KMOD	87.451	12.549	5.823	5.382
	T-KMOD	91.692	8.308	2.431	2.732
DS2	POLY	85.832	14.168	30.552	6.930
	RBF	86.358	13.642	58.398	25.932
	KMOD	90.683	9.317	20.743	4.539
	T-KMOD	93.802	6.198	15.590	2.473
DS3	POLY	82.483	7.517	48.395	7.892
	RBF	85.928	4.072	68.932	28.392
	KMOD	91.842	8.158	32.478	6.821
	T-KMOD	94.515	5.485	10.892	3.589

The above results show that the performances on network intrusion detection of T-KMOD is more better than that of other kernel functions.

3 Conclusion

In this paper, T-KMOD kernel function is described. From the discussion above, know that the T-KMOD kernel has two main advantages: one is that it is not necessary to make a selection out of the above mentioned kernel functions, which can simplify the modeling process and will save much computing time. The other is

that due to its flexibility to vary, it has a stronger mapping ability and can properly deal with kinds of mapping problems.

References:

- [1] Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods [M]. Cambridge, UK: Cambridge University Press, 2000.
- [2] Remaki L, Cheriet M. Kcs-new kernel family with compact support scale space [J]. IEEE Transactions on Image Processing, 2000, 9(6): 970-981.
- [3] 邓乃扬, 田英杰. 支持向量机-理论、算法与拓展 [M]. 北京: 科学出版社, 2009.
- [4] Ayat N E, Cheriet M, Remaki L, et al. KMOD-A New Support Vector Machine Kernel with Moderate Decreasing for Pattern Recognition, Application to Digit Image Recognition [C]//Proceedings of 6th International Conference on Document Analysis and Recognition. Seattle, USA: IEEE, 2001.
- [5] Schoenberg I J. Metric spaces and completely monotone functions [J]. Annals of Mathematics, 1938, 39(4): 811-841.
- [6] Scholkopf B, Burges C, Smola A. Advances in Kernel Methods: Support Vector Learning [M]. [s. l.]: MIT Press, 1999.
- [7] 张成伟, 郑 诚. 基于改进 SVM 的文本信息检索研究 [J]. 计算机技术与发展, 2009, 19(1): 71-73.
- [8] 王玉震, 李 雷. 基于 SVR 的图像增强方法 [J]. 计算机技术与发展, 2009, 19(1): 60-62.
- [9] 李 雷, 鲁延玲, 周蒙蒙. 基于核方法的一种新的模糊支持向量机 [J]. 计算机技术与发展, 2010, 20(2): 9-11.
- [10] 施其权, 李小明, 肖辞源. 一类新型快速模糊支持向量机 [J]. 计算机技术与发展, 2010, 20(2): 103-105.
- [11] 侯惠芳, 白莉媛, 刘素华. 基于支持向量机的图像边缘检测研究 [J]. 计算机工程与应用, 2007, 43(18): 32-33.
- [12] KDD99, KDD cup 1999 data [EB/OL]. 1999. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
- [13] Yi Yang, Wu Jiansheng, Xu Wei. Incremental SVM based on reserved set for network intrusion detection [J]. Expert Systems with Applications, 2011, 38(6): 7698-7707.

(上接第 156 页)

Notes in Computer Science, 2002, 2487: 110-127.

- [12] 张琳琳, 应 时, 倪 友. 一种软件体系结构关注点分析方法 [J]. 计算机学报, 2009, 32(9): 1782-1791.
- [13] 王 伟, 杨 庚, 张迎周, 等. 基于程序切片和服务构件的

语义 Web 服务组合 [J]. 计算机技术与发展, 2011, 21(11): 141-144.

- [14] 李 翔, 怀进鹏, 曾 晋, 等. 一种 Java 遗留系统服务化切分和封装方法 [J]. 计算机学报, 2009, 32(9): 1804-1815.