

基于相似性度量的软构件的选择

The Selection of the Software Components Based on Similarity Evaluation

刘 升¹ 游晓明¹ 石 玮¹ 陈传波²

(湖北师范学院计算机系 黄石435002)¹ (华中科技大学计算机系 武汉430074)²

Abstract In this paper, the conception model of software component has been analyzed and discussed. By utilizing ideas of OO and extending them, the author proposes an improved descriptive method based on OWE(OBJECT WITH EVENT) for software component. Finally, from the view of software reuse, the methods of identify and select reusable components from software components repository are presented, by which the software productivity can be increased rapidly.

Keywords Reusable software component, Object-oriented design, Similarity evaluation

1 问题的提出

近几年来,随着软件规模和复杂性的不断增长,传统“手工作坊”式的软件开发模型,难于适应和满足现代软件工业发展的需要,基于构件的软件开发方法已成为提高软件生产效率和产品质量,缩短产品交付时间的现实有效的途径之一,但在长期的实践中,软件重用的研究主要集中在技术方面的研究,如:构件化的编程方法,可重用构件的分析与设计,可重用构件库的设计等,而对于一些非技术问题没有得到妥善的解决,其中包括:为了对软构件库中的构件进行分类、存储、检索和理解,如何有效地对软构件的内部特性和外部特性加以描述,如何从构件库中检索出所需要的可复用构件,其主要依据和标准是什么?本文针对这些问题,提出了一种描述和选择软构件的方法。

2 软构件的形式描述

软构件的有效组织和管理是软件复用的关键,软构件描述是软构件组织和管理的基础和核心,一方面,为了开发和使用可复用软构件,软构件必须是易于理解的,需要对软构件的属性及软构件间的关系加以描述,另一方面软构件描述不仅能帮助用户理解、书写、修改和复用软构件,而且能促进软件复用工具对软构件进行有效的组织和管理^[1]。

结合面向对象的思想,同时为了使其实用,增强对象的表达能力,我们对通常的对象结构作了扩充,我们

把这样的对象称为结合事件的对象,这样我们便可以

将软构件表示成如下三元组:

软构件=(属性,方法,事件)

其中属性描述软构件外在的或内在的一些特性,软构件属性指出一类可以赋给它的特定的值,属性可以区分为状态主导属性和状态相从属性,状态主导属性决定构件的内部状态,而状态相从属性的引入则多是出于某种使用方便的考虑如私有属性、导出属性等。状态相从属性依赖于构成对象状态的主导属性,对象状态的变化亦可能导致这些相从属性值的改变。

方法是指处理构件类或类实例的函数和过程的总称,任何一个对象(类)内在地拥有属于它自己的一些函数和过程,并能够被它所调用,方法反映了一个对象所拥有的“能力”,描述这些函数和过程的代码则被封装于对象内部,根据需要对外提供一些必要的参数,调用这些方法是发挥软构件功能的基本手段。要使应用

软件中的构件去执行某一特定的任务最常用的方法就是在程序中灵活运用为每个构件所内建的方法。

事件是软件开发中一个比较重要的概念,Windows应用程序普遍采用事件驱动方法管理程序和应用之间的交互行为,事件能够触发相应程序的执行,即程序能够对不同的事件作出不同的反应,这种反应需要开发者为之编写代码,事件与代码之间通过事件句柄互相联系。由事件触发的过程称之为事件句柄,每一类的构件都有响应特定事件的能力。可以这么说应用程序的执行过程就是系统不断地检测事件执行。

刘 升 硕士,讲师,研究方向:计算机软件和分布式处理。游晓明 硕士,副教授,研究方向:计算机软件和分布式处理。陈传波 教授,研究方向:计算机网络与信息管理工程。

我们以对象管理组技术框架作为指南,并以 Rumbaugh 的研究为基础给出软构件类的具体描述,这样做的原因主要有两点,第一,Rumbaugh 的方法似乎最容易被软件工程师们所理解,它或多或少建立在已经被软件工程从事者使用多年的图形概念基础之上;第二个最主要的原因是有一个能在 PC 或 UNIX 平台下的建模工具是基于 Rumbaugh 研究,其输出代码很容易被提取和编译成所要的形式^[3-5]。

```
@Component "Comp-name"
$ ATTRIBUTE
* 1: "attribute-name"
  [, TYPE="type-name"] [= "default-value"]
  [, CONSTRAINT="constraint-expression"]
  [, DERIVED FROM="derivation-expression"]
* 2:
  :
$ METHOD
* 1: "Function-name" [, type = "type-name"] ([ type
  "param1" [, type "param2"] ... )
* 2: "Procedure-name" ([ type "param1" [, type "param2"] ... )
  [, INHERITED FROM="super-comp"]
  :
$ EVENT
* 1: "event-name"
  [, ATTRIBUTE="attr-name"] [= "default-value"]
  [, INHERITED FROM="super-comp"]
  :
$ IMPLEMENTATION
& VAR-NAME [, TYPE="type-name"]
& METHOD
* 1: "method1-code"
  :
* n: "methodn-code"
& EVENT
* 1: "event1-code"
  :
* n: "eventn-code"
@ End-Component
```

子句 ATTRIBUTE 用来定义构件的所有属性,其关键字 CONSTRAINT 指出该属性所应受的约束条件表达。

DERIVED FORM 指出该属性所属超类。

子句 METHOD 用来定义构件中所有的方法分为函数和过程,其中函数可采用归纳定义法描述如下:

```
FUNCTION<函数名>(t1:T1,t2:T2,...,tn:Tn):T
IND FOR t1:T1(PB(t1,t2,...,tn),PI(t1,t2,...,tn));
```

它是施归纳于 t_i,其中 PB 和 PI 是谓词。分别对应于基始和归纳。这时 T_i 应是良序类型,包括非负整数和非正整数和表。

过程是修改对象状态或完成某种功能的操作,它的描述形式如下:

```
PROCEDURE<过程名>(t1:T1,t2:T2,...,tn:Tn)
[,T][mod f1,f2,...,fm][compute t;T]P(t,t1,t2,...,tn);
```

其中 MOD f₁,f₂,...,f_m 是本过程的修改权限说明,表示本过程将修改 f₁,f₂,...,f_m,P 是函数或谓词名,修改权限说明中只能列出独立函数或谓词。

INHERITED FROM 的作用同属性说明中的 DRIVED FORM 一样。

子句 EVENT 被用来描述一个事件和它的属性。

子句 IMPLEMENTATION 中必须定义 METHOD 子句中说明的所有方法和 EVENT 子句中所有事件的实现代码。

3 软构件的相似性度量

从软件的开发过程,我们可以了解到所采用的复用技术无外乎以下四种:抽象技术、选择技术、例化技术和集成技术,但主要的也是最难的是抽象和选择。抽象技术我们已在有关文献中作了详尽的阐述,要对构件进行选择,首先要对构件的相似性进行度量然后才能进行选择,下面给出相似条件定义^[4]:

1. 名字相似性: N_i 和 N_j 具有相似性是指 N_i 和 N_j 完全匹配或具有同义关系或相似关系,表示为 N_i~N_j

2. 属性值域相似性:属性值域 D_i 和 D_j 具有相似性是指 D_i=D_j 或 D_i⊂D_j 或 D_i⊃D_j,表示为 D_i~D_j

3. 属性的相似性:属性 attr_i 和 attr_j 相似,即 attr_i~attr_j,是指属性名 N_{attr_i}~N_{attr_j} 和属性值 D_i~D_j 同时成立。

方法和事件的相似定义同上类似。

4. 相似性度量:构件 C_i 和 C_j 之间的相似度量 A(C_i,C_j) 定义为

$$A(C_i, C_j) = W_{ta} * TA(C_i, C_j) + W_{aa} * AA(C_i, C_j) + W_{ea} * EA(C_i, C_j) + W_{ma} * MA(C_i, C_j)$$

其中:TA(C_i,C_j)表示构件的名称即术语相似性度量;AA(C_i,C_j)表示构件的属性相似性度量;EA(C_i,C_j)表示构件的事件相似性度量;MA(C_i,C_j)表示构件的方法相似性度量;W_{ta},W_{aa},W_{ea},W_{ma} 代表权值,且 W_{ta}+W_{aa}+W_{ea}+W_{ma}=1。

·名称(即术语)相似的度量

$$TA(C_i, C_j) = \begin{cases} 1 & N_a = N_j \\ W_r & N_a <> N_j, I N_a \sim N_j \\ 0 & \text{其它} \end{cases}$$

W_r 是指 N_aN_j 之间的关系强度

·结构属性的相似性度量 Attr(C_i∩C_j)是指 C_i 和 C_j 中相似性的属性的集合

$$AA(C_i, C_j) = \frac{2|attr(C_i \cap C_j)|}{|attr(C_i)| + |attr(C_j)|}$$

|Attr(C_i∩C_j)|指 C_i 和 C_j 中具有相似性属性数目;|Attr(C_i)|和|Attr(C_j)|分别指 C_i 和 C_j 的属性数目。

·方法的相似性度量

$$MA(C_i, C_j) = \frac{2|fun(C_i \cap C_j)|}{|fun(C_i)| + |fun(C_j)|}$$

|fun(C_i)|和|fun(C_j)|分别指 C_i 和 C_j 中所定义的函数或过程的数目;|fun(C_i∩C_j)|指 C_i 和 C_j 中具有相似

(下转第101页)

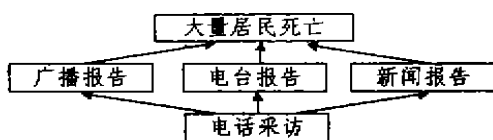


图2 规则不能局部地简单组合

外延系统的局域性太强,不能识别出证据的共同来源,因此,有时把并非是相互独立的证据当作相互独立的证据去更新假说的信度,从而进行一些不正确的推理。

3. 处理不确定性的内涵方法

内涵方法在语义方面非常清晰,因为它有全局一致的描述模型,不存在双向推理、相关证据处理方面的问题,但计算复杂性是其致命的弱点。在较复杂的环境下,内涵方法基本是不可用的。

举个例子形象说明外延方法和内涵方法这两种极端的情况。外延方法:一个人骑着一匹马穿过一片有地雷的区域。他可能选择一匹经过训练的、对地雷比较敏感的马,但是危险是实际存在的,有经验的人(知识工程师)会阻止马跑得太快以避免发生危险;

内涵方法:安全无虞的语义保证每一步前进都不会发生危险,但是前进非常困难,因为在每走一步之前都必须重新检测整个区域。

显然,在计算模块化和语义一致性之间存在矛盾,如何在计算的简便高效和语义的清晰之间选择一种较好的折衷方案是不确定性处理的重点研究内容。近年

来逐渐引起广大人工智能学者极大重视的贝叶斯网就是一种较好兼顾计算简单和语义清晰的方案。从本质上讲,贝叶斯网是一种内涵方法,是全概率分布的紧凑表示形式,由于它采用图形表示方式嵌入了大量的独立关系,使得计算复杂性大为下降,在很大程度上克服了内涵方法在计算方面的困难,使得贝叶斯网是可行的和实用的,逐渐成为不确定性处理的主流方法。

小结 本文比较不确定性处理的外延方法和内涵方法,分别指出了它们各自的优缺点,在实际处理不确定性的相关问题时,可以根据需要在计算简便性和语义清晰性间取舍,贝叶斯网是一种较好兼顾两者的解决方案。

参考文献

- 1 Pearl J. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 1986, 29: 241~288
- 2 Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA
- 3 Pearl J. Belief networks revisited. *Artificial Intelligence*, 1993, 59: 49~56
- 4 Pearl J. Causal diagrams for empirical research. *Biometrika*, 1995, 82: 669~710
- 5 Cooper G. Computational complexity of probabilistic inference using Bayesian belief networks (Research note). *Artificial Intelligence*, 1990, 42: 393~405
- 6 Friedman N, Yakhini M. On the sample complexity of learning bayesian networks. *UAI'96*
- 7 Charniak E. Bayesian Networks without Tears. In *AAAI'91*
- 8 Neufeld E. Direction in uncertainty reasoning. *The Knowledge Engineering Review*, 1997, 12(4): 413~415

(上接第122页)

性的函数或过程对数。其它相似性度量类似

基于相似性度量我们可以构造出相似候选构件次序表,用户就能够选出匹配度最高的可复用构件,下面给出选取可复用构件的算法

```

Begin
  查找复用成分库
  if 完全相同的匹配 Then 结束
else
  集中相似的构件
  for 每个构件
    计算匹配程序
  end
  排列和挑选最好的
  修改构件
fi
END

```

其中的匹配程度可用相似性度量 $A(C_i, C_j)$ 的值来衡量。

选择相似的构件是一个分类问题,相似性程度依赖于如何组织该集合,可以谨慎地选择相关的属性,将密切相关的构件分成组,并按意义组织它们。

结束语 随着应用系统的日益复杂,软件构件化作为软件系统发展的重要方向,引起了人们的普遍关注,软构件技术将会受到越来越多的重视,我们提出的基于 OWE(Object With Event)软件构件描述方法,以及基于相似性度量的软件选择方法为开发复杂的应用系统提供了一个很好的解决途径,对提高软件生产率、改善软件产品质量、缩短产品交付时间有着十分重要的意义。

参考文献

- 1 郑明春,张家重. 关于软件复用. *计算机科学*, 1994, 21(4)
- 2 Marovac N. Embedded Documentation For Semi-automatic Program Construction and Software Reuse. *ACM Software Engineering Notes*, 1998, 23(3): 70~74
- 3 Whittleben B. Models and Languages for Component Description and Reuse. *ACM SIGSOFT Software Engineering Notes*, 1992, 20(2)
- 4 Castano S, Antonellis V D. Engineering a library of reusable conceptual components. *Information and Software Technology*, 1997, 39: 65~76