第8卷第4期 软件学报 Vol. 8, No. 4
1997年4月 JOURNAL OF SOFTWARE Apr. 1997

类相似性的比较及其在类库检索中的应用*

吕枫华 王和珍 费翔林

(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘要 本文给出了一种比较类相似程度的近似度量方法,讨论了它在类库检索中的应用,并给出了一个基于规则的类库检索工具RBRT.

关键词 软构件,重用,类库,检索.

中图分类号 TP311

面向对象程序设计风范被认为是一种实现软件重用的较好途径.它的模块性和继承机制为软件重用提供了强有力的支持.一方面,通过多次创建某一个类的实例对象,即可实现类的重用;另一方面,继承性提供了一种更有效的重用方法——通过继承已有类得到新类,已有类的一切均可以为新类重用.目前,已经有多种面向对象程序设计范型,比较有名的流派有:Coad-Yourdon 的 OOA-D,Booch 的 OOD,ESA 的 HOOD 以及 Rumbaugh 等人的 OMT 等.这些范型采用的技术路线各有特点,但软件开发一般都包含 OOA,OOD,OOP 等阶段.[1~5]

在面向对象程序设计范型中,重用可出现在各个阶段,重用的对象也可不同(例如:在 OOA 阶段对已有领域分析的重用、在 OOD 阶段对已有应用框架的重用和对已有类的重用等),比较典型的是对已有类的重用.[4,5]假设有一类库,库中存放了许多以前开发的类,典型的类重用模式可以表示如下:

(1)给出类 X 的规格需求;

(2)在类库中查找

①如果找到的与 X 相似的候选类集合 CS 非空,则转(3);

②否则转(4).

(3)计算 CS 中每一个候选类与 X 的相似匹配值;

②选择最佳匹配类 Y;

③修改或采用重载等方法调整类 Y,使之符合 X 的规格需求,结束.

(4)设计开发一个新类,使之符合 X 的规格需求,结束.

如能利用 Y 来构造 X,而不用重新开发 X,就可以节省大量的时间和精力,提高正确

---

* 作者吕枫华,1970年生,讲师,主要研究领域为面向对象技术和软件工程.王和珍,女,1943年生,副教授,主要研究领域为面向对象技术和软件工程.费翔林,1941年生,教授,主要研究领域为面向对象软件工程和操作系统.

本文通讯联系人:吕枫华,南京 210093,南京大学计算机软件新技术国家重点实验室

本文1996-04-01收到修改稿

性，改进可靠性.

成功重用已有类的前提是要有一个好的类库. 目前，已有许多类库产品出现，它们大都具有以下特点：规模小，一般只包括几十个类；应用范围窄，用于构造用户界面和数学计算(如向量运算)等通用领域者居多，例如：与 Borland C++结合在一起的 OWL 以及适用于多种 C 语言产品的 Classix 等；缺乏有效的类库管理设施，尤其缺乏检索工具，一般只提供简单的浏览工具.

类库的规模一旦变大(比如包含几百乃至上千个类)，将会遇到许多困难. 困难首先来自类库的设计，许多著作都对此作过阐述，例如：Korsn 和 Mcgregor 曾经提出有关类库应具备的特性和设计规范[6]；其次，建立类库是为了实现类的重用，类库的规模变大以后，在上述类重用模式中，如何迅速寻找到符合用户需求的候选类？如何计算每一个候选类与 X 的相似匹配值？必须提供功能强大的检索工具，否则用户就可能需要花费大量的时间和精力去熟悉类库. 然而，目前尚无行之有效的检索工具，最主要的原因是缺乏简单有效的方法用以比较类的相似性.

最简单的方法是比较类名，即在类库中找出类名与用户给出的类名相同或同义的类，作为候选类. 这种方法的局限性显而易见. 一方面，找出的类可能与用户需要的类毫不相关；另一方面，尽管类库中含有符合用户需求的类，但由于用户没有给出合适的类名而找不到. 事实上，类所反映的概念应由它的语义决定，企图用类名来浓缩类中所有的信息是不可能的.

因此，研究类相似性的比较问题对于解决大型类库的检索，开发高效率的类库产品具有重大意义. 本文给出了一种比较类相似程度的近似度量方法，讨论了它在类库检索中的应用，并给出了一个基于规则的类库检索工具 RBRT.

1　类相似性的近似度量

类刻画了一组具有共同特性的对象，从它自身的内容来看，应由 3 部分组成：属性、操作以及接口. 它们可能是在该类本身中声明的，也可能是从父类继承而得到的. 其中属性用以描述对象的状态. 操作用来刻画对象接收到某个消息后所作的动作，该动作可能是改变对象的某些属性，即改变对象的状态；可能是创建一个新对象；也可能是给其它对象发送某一个消息. 在程序实现中，操作常表现为一段程序体. 接口用以描述对象的外部可见的特性，一般可以分成 2 种接口：消息服务接口和消息发送接口，前者用以刻画对象接收到某个消息后提供的服务的名称、格式等；后者用以刻画对象发出消息的接收对象的名称、消息的名称和格式等. 类的组成及相互关系如图 1 所示.

要全面完整地比较 2 个类的相似程度，很自然，必须对所有方面分别进行比较，包括它们的相互关系. 其中操作的比较有其特殊性，原因有 3 点：

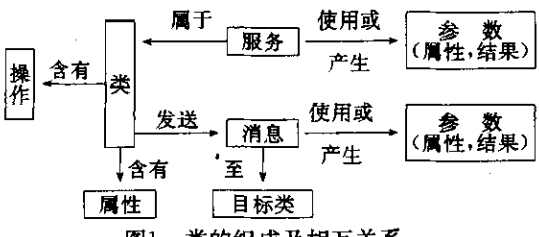

图1　类的组成及相互关系

第 1，操作的定义涉及具体的程序体，对 2 个分别属于 2 个类的操作进行比较，意味着要对 2 段程序体进行比较，这是难以实

现的;

第2,某些场合下2个操作的程序体虽不同,它们实现的功能却一致.例如,2个数组类A和B,各含有一个操作$Sort$实现对数组的排序,在类A中$Sort$采用冒泡排序算法,而在类B中$Sort$采用快速排序算法,算法不同,程序体肯定不同,但它们实现的功能相同;

第3,用户进行类库检索,选择可重用的类时,尚未实现他所需要的类,一般不可能也不必要提供操作的具体实现,否则重用就变得没有意义了,但是可以给出所需类的属性和接口的描述.

于是,在比较2个类的相似程度时,就可以不考虑操作的具体实现,而从其他几个方面(属性、服务和消息)进行比较,以得出比较类相似性的近似方法.此外,还基于以下前提:

第1,比较类相似性的目的是找出检索类库的有效方法.一般而言,某个类库总是覆盖了某个应用领域.在一个具体的应用领域中,某些概念都有比较统一的名字.程序员在构造类库时,必须充分考虑到这一点,类名、类的属性名和服务名的选择必须比较恰当.用户给出检索要求时,也要尽量使用该领域中具有代表性的名字.通过建立一个应用领域的同义词字典可以减少由于使用不同名字造成的检索困难.

第2,具有相同属性名的2个类,它们可能是毫不相干的,但是如果局限在某个应用领域中,一般认为它们是密切相关的;同样,如果2个类提供一致的服务(服务名,服务使用到的属性名和产生的结果值都相同)或者具有一致的消息(消息名,接收消息的目标对象所在类的类名、消息使用到的属性名和产生的结果值都相同),那么这2个类相似的可能性将大大提高.

1.1 类的近似描述

类X的近似描述用一个三元组表示:$X(At,Se,Me)$.该三元组中的元素定义如下:

(1)At(属性集)At为一集合,形式是$\{A_1,A_2,A_3,\cdots,A_u\}$,$A_i(1\leqslant i\leqslant u)$表示一个属性的名字,用一个标识符来描述.$At$也可为空集$\varnothing$.

(2)Se(服务集)Se为一集合,可以为空集$\varnothing$.一般形式是$\{S_1,S_2,S_3,\cdots,S_v\}$,$S_i(1\leqslant i\leqslant v)$表示一个服务,由三元组$S_i(Se_Name,Used_At,Result)$描述,其中$Se_Name$表示该服务的名字,用一个标识符来描述;$Used_At$表示该服务使用到的属性的集合,形式为$\{A_{i_1},A_{i_2},A_{i_3},\cdots,A_{i_p}\}$,$A_{i_j}(1\leqslant j\leqslant p)\in At$,用一个标识符来描述;$Result$表示该服务的结果或返回值,用一个标识符表示.

(3)Me(消息集)Me为一集合,可以是空集$\varnothing$.一般形式是$\{M_1,M_2,M_3,\cdots,M_w\}$,$M_j(1\leqslant j\leqslant w)$表示一个消息发送,由四元组$M_j(Me_Name,Object_Class,Used_At,Result)$描述,其中$Me_Name$表示该消息的名字,用一个标识符来描述;$Object_Class$表示该消息的接收对象所在类的名字,用一个标识符来描述;$Used_At$表示该消息使用到的属性的集合,形式为$\{A_{j_1},A_{j_2},A_{j_3},\cdots,A_{j_z}\}$,$A_{j_i}(1\leqslant i\leqslant z)\in At$,用一个标识符来描述;$Result$表示该消息的结果或返回值,用一个标识符表示.

1.2 计算2类的相似值

设有2个类C_1和C_2,它们的类的近似描述分别是$C_1(At_1,Se_1,Me_1)$和$C_2(At_2,Se_2,Me_2)$,$Card(Set)$是计算集合Set中元素个数的函数,则

2类C_1和C_2的相似确认值$I_c(C_1,C_2)$定义如下:

$$I_c(C_1,C_2)=\Delta_a\times I_a(C_1,C_2)+\Delta_s\times I_s(C_1,C_2)+\Delta_m\times I_m(C_1,C_2)$$

其中 $\Delta_a,\Delta_s,\Delta_m$ 分别是属性相似值、服务相似值和消息相似值的权重系数,满足 $\Delta_a\geqslant 0,\Delta_s\geqslant 0,\Delta_m\geqslant 0$ 并且 $\Delta_a+\Delta_s+\Delta_m=1$,

属性相似值 $I_a(C_1,C_2)=\dfrac{Card(At_1\cap At_2)}{Card(At_1\cup At_2)}$;

服务相似值 $I_s(C_1,C_2)=\dfrac{Card(Se_1\cap Se_2)}{Card(Se_1\cup Se_2)}$;

消息相似值 $I_m(C_1,C_2)=\dfrac{Card(Me_1\cap Me_2)}{Card(Me_1\cup Me_2)}$.

注意,当 $At_1=At_2=\varnothing$ 时,令 $I_a(C_1,C_2)=1$;当 $Se_1=Se_2=\varnothing$ 时,令 $I_s(C_1,C_2)=1$;当 $Me_1=Me_2=\varnothing$ 时,令 $I_m(C_1,C_2)=1$.

易见,2 个类的属性相似值反映了它们属性的相似程度,值越大,表示它们可能含有的相同属性越多,当属性相似值为 1 时,表示它们的属性完全一致. 同样服务相似值和消息相似值分别反映了它们的服务相似程度和消息相似程度. 而 2 个类的相似确认值,是 2 类相似性的近似度量,它包含了 2 个类的属性相似程度、服务相似程度和消息相似程度的信息,因而,很好地反映了它们的相似程度. 2 个类的相似确认值越接近 1,表示它们的相似程度越好. 极端情况下,当 2 个类的相似确认值 I_c 为 1 时,它们的属性相似值 I_a,服务相似值 I_s 和消息相似值 I_m 均是 1,此时,$At_1=At_2,Se_1=Se_2,Me_1=Me_2$,表示 2 个类的属性、服务和消息都是一致的.

权重系数 Δ_a,Δ_s 和 Δ_m 表示了比较 2 类相似性时对属性、服务和消息的重视程度,一般可根据类的属性、服务和消息数目的多少以及检索的情况予以调整.

2　基于规则的类库检索工具——RBRT

基于规则的类库检索工具 RBRT 是我们设计的类库管理系统 CLMS 中的一个部分,采用 Borland C++ 3.0 和 PDC Prolog 3.2 开发. 它的体系结构主要由 1 个类库、1 个查询规则库、1 个字典和 1 个工具集组成. 它们相互配合,支持用户对类库的检索. 字典存放了领域单词的同义词,主要支持单词匹配工作,例如,window=view,circle=round=ring=tyre 等等. 查询规则库存放查询规则,一部分规则用于描述类库中各个类的情况,例如:类名、相关的应用领域、功能的简单描述、类的近似描述等;另一些规则描述了匹配计算候选类相似确认值的算法和策略. 查询规则库将使检索过程变得简单而高效,当添加新类时只需把新类信息组织成规则加入查询规则库即可.

用户通过填写类的查询卡片,输入类的查询要求. 类的查询卡片中主要包含以下信息:

- 类名,类的可能别名,用标识符描述;
- 类的可能应用领域,用几个关键词描述;
- 类的功能描述,用几个关键词描述;
- 类的近似描述,近似描述的定义如上一节所示.
- 检索参数,包括:属性、服务、消息的权重系数等.

用户提供欲查询的类的有关信息(越多越好)之后,检索工具将把查询卡片转化成查询目标,在一定的检索策略指导下搜索整个类库,通过计算相似确认值,找出相似确认值最大

的候选类，它可能就是满足用户需求的可重用的类，利用浏览工具可以观察检索到的候选类的全部信息.

3 结束语

本文给出了一个比较类相似程度的近似度量方法，着重讨论了它在类库检索中的应用，并给出了一个行之有效的基于规则的类库检索工具RBRT. 今后的工作有：①结合人工智能研究类查询规则的自动生成和查询规则库的自动建立；②对类相似性度量方法作进一步的研究；③探索类的相似性度量在新开发的类入库时的应用，如确定新类与类库中的已有类有无重复成分，以决定是否重新组织类库中类的层次等.

致谢 本文完成过程中得到徐家福先生的悉心指导，在此表示衷心感谢.

参考文献

1 Coad Peter, Yourdon Edward. Object-oriented analysis. 2nd ed., New Jersey: Prentice-Hall International, Inc. Englewood Cliffs, 1991.

2 Coad Peter, Yourdon Edward. Object-oriented design. 2nd ed., New Jersey: Prentice-Hall International, Inc. Englewood Cliffs, 1991.

3 Rumbaugh James *et al*. Object-oriented MODELING and design. New Jersey: Prentice Hall, Englewood Cliffs, 1991.

4 Jacobson Ivar. Object-oriented software engineering. New York: Addison-Wesley Company, 1992.

5 Meyer Bertrand. Object-oriented software construction. New Jersey: Prentice Hall, Englewood Cliffs, 1988.

6 Korson T, McGregor J D. Technical criteria for the specification and evaluation of object-oriented libraries. Software Engineering Journal, 1992,7(2):85～94.

CLASS SIMILARITY COMPARISON AND ITS APPLICATION IN CLASS LIBRARY RETRIEVAL

Lü Fenghua WANG Hezhen FEI Xianglin

(*State Key Laboratory for Novel Software Technology Nanjing University Nanjing* 210093)

Abstract This paper offers a method to determine the similarity between two classes and discusses how to apply it to class library retrieval. A rule-based tool called RBRT for retrieving class from a library is also presented.

Key words Software component, reuse, class library, retrieval.

Class number TP311