

青鸟构件库的构件度量*

梅宏 谢涛 袁望洪 杨美清

(北京大学计算机科学技术系 北京 100871)

E-mail: meih@pku.edu.cn/xietao@cs.pku.edu.cn

摘要 对软件产业发展所需要的软件生产率和软件质量的重视引起了对软件复用技术的研究,同时,有关复用度量的研究和应用也引起了广泛的重视.青鸟构件库系统可以对可复用构件进行描述、管理、存储和检索,用以满足基于“构件-构架”复用的软件开发过程的需要.该文介绍了青鸟构件库中的构件度量模型,并阐述了利用青鸟程序分析系统中的面向对象度量工具和青鸟构件库后期度量系统(包括反馈信息的收集、处理和和分析工具)来实现模型的方法.

关键词 软件复用,构件库系统,软件度量,面向对象度量,可复用性度量,复用度量.

中图法分类号 TP311

最近几年来,人们开始认识到,要真正实现软件的工业化生产方式,达到软件产业发展所需要的软件生产率和质量,软件复用是一条现实可行的途径^[1].对软件复用技术的研究已成为软件工程学科的主攻方向之一.

青鸟工程是在国家支持下的重点科技攻关课题,历经国家“六五”、“七五”和“八五”科技攻关计划,已有十余年的发展,开发成功了集成化软件开发环境 JBI 和 JBII.在国家“九五”计划中,青鸟工程的任务是在前期成果的基础上,针对软件工业化生产的需求,实现软件生产线的思想,制定软件工业化生产标准,强化采用面向对象技术,支持以软件复用为基线的,基于“构件-构架”模式的软件工业化生产技术,开发基于异构平台、可访问多信息源的应用系统集成(组装)环境青鸟 III 型(JB3)系统.

JB3 作为一个支持复用的软件开发环境,构件的有效管理和查询是关键,其核心是一个构件库系统(JBC1. Jade Bird component library).青鸟构件库系统用于对可复用构件进行描述、管理、存储和检索,以满足基于“构件-构架”复用的软件开发过程的需要.它以青鸟构件模型为基础,建立青鸟构件库数据模型,并与其他 CASE 工具相结合,支持构件的生产、描述(使用青鸟构件描述语言 JB-CDL)、分类、存储、查询和复合^[2].

在软件开发过程中,进行软件度量可以帮助管理人员控制、安排软件开发,并利用反馈信息对软件进行改善,从而提高软件质量.软件度量的必要性和重要性已为软件界所认同.构件库中的构件度量是构件库系统不可分割的一部分,通过度量有助于测度复用的潜力和预期的收益.

本文第 1 节介绍度量理论以及传统的、面向对象的以及有关复用的度量方法,第 2 节简要介绍青鸟构件库的构件管理流程,第 3 节提出构件库度量的内容和研究领域,第 4 节介绍青鸟构件库中的构件度量,首先介绍了它所基于的要素-准则-度量模型,并提出了青鸟构件度量模型以及对它的实现,第 5 节讨论相关的工作,第 6 节总结全文,并提出了进一步研究的课题.

* 本文研究得到国家“九五”科技攻关项目基金(No. 96 729)、国家 863 高科技项目基金(No. 863-306-02-02 01)和日本理光公司资金资助.作者梅宏,1963 年生,教授,博士生导师,主要研究领域为软件工程,软件工程环境,新型程序设计语言,软件复用,软件构件技术.谢涛,1975 年生,硕士,主要研究领域为程序分析,软件复用,软件度量,软件再工程.袁望洪,1972 年生,硕士,主要研究领域为程序理解,软件复用,软件度量.杨美清,女,1932 年生,教授,博士生导师,中国科学院院士,主要研究领域为系统软件,软件工程和软件工程环境,软件复用和软件构件技术,软件工业化生产技术.

本文通讯联系人:梅宏,北京 100871,北京大学计算机科学技术系

本文 1999-03-22 收到原稿,1999-06-01 收到修改稿

1 度量的基本理论和方法

1.1 度量理论

现代度量理论是一个数学分支,它是研究广泛适用于自然科学、社会科学以及工程研究等各种度量的逻辑基础和基本原理的理论。目前对软件度量的批评之一就是度量缺乏坚实的理论基础,Kearncy等人批评软件度量没有坚实的理论基础并且缺乏合适的性质^[3]。

度量就是把数值或符号赋予所研究的对象,任意一个度量都必须具备4个因素:被度量的对象、将被度量的属性、赋予对象的数值或符号以及使对象与数值或符号相联系的映射。由此可以对度量作以下形式化定义:一个度量由3个部分构成:(1)经验性关系系统 $Q=(O,R)$,其中 O 为被度量的对象集合, $R=\{R_1,R_2,R_3,\dots,R_n\}$ 为 O 上的一系列关系;(2)数值关系系统 $N=(C,P)$,其中 C 为数值或符号的一个集合, $P=\{P_1,P_2,P_3,\dots,P_n\}$ 为 C 上的一系列关系;(3)映射 $M:Q \rightarrow N$, $M(x)$ 为对象 x 在被度量属性上的度量值。度量对象的属性还可以通过度量此对象的其他一些属性进行计算而得,这样的度量称为间接度量。

软件的属性可分为内部属性和外部属性两种。内部属性通常描述软件结构上的复杂性,如软件大小、控制流、耦合度等。内部属性一般有清晰的定义并能进行客观的度量。外部属性通常涉及到人和环境等外部因素,如复杂性、可维护性、可读性等。通常只有外部属性的度量才能真正给出人们所需的可靠数据,并且预示软件开发行为。但是,外部属性常常缺乏清晰的定义,也无法进行直接的客观度量,所以一般只能通过内部属性进行推测性的间接度量来得出外部属性值。度量研究的任务之一就是建立外部属性和内部属性之间的合理联系。

随着软件度量纳入更严密的理论体系,有必要将度量评估纳入形式化范畴。Weyuker提出了一组形式化评估软件度量性质的定理^[4]。虽然关于这9条性质的定理并不见得很完美,但是,这组性质定理仍不失为度量评估的良好出发点。

1.2 度量的方法

度量方法的研究可以追溯到70年代,开始的研究大部分是针对面向过程和结构化的软件复杂性度量。其中主要的度量方法有代码行数方法、以程序控制流为基础的McCabe环计数方法(cyclomatic number)^[5]、Halstead的软件科学方法^[6]和Albrecht的功能点方法^[7]等。70年代后半期,McCall和Boehm分别提出了两个关于软件质量因素的层次结构模型,其所基于的要素-准则-度量模型至今仍为人们所采用。

面向对象软件开发在软件生命周期、系统结构、复用以及项目管理等方面与传统的软件开发有着显著的区别,这些区别很大程度上影响了软件度量,使得传统的度量不再适用于面向对象的某些概念,而需要进行专门的适合面向对象的度量。面向对象的度量方法主要有Chidamber和Kemerer提出的6条适于面向对象设计的度量准则^[8]以及Lorenz等人提出的一些适于面向对象的度量准则^[9]。

在软件开发过程中,软件复用是提高软件生产率和软件产品质量上有着巨大的潜力。软件开发者在利用软件复用方法进行开发时,需要去度量开发进度,找出最有效的复用策略,并测度出复用的潜力和预期的收益。另外,还要能够识别出现存系统中的可复用构件等。

有关复用的度量主要包括可复用性度量(reusability metrics)和复用度量(reuse metrics)。可复用性度量主要用来判定一个构件的可复用性和质量,它的主要作用有:识别出遗产系统中的领域知识和有用的产品^[10],为构件的提取提供基础;对构件库中的可复用构件进行客观的控制,保证构件库中保存的是高可复用性和高质量的构件。另外,在复用库中如果能同时包含构件的复用和质量度量的信息,可以给使用构件库中构件的复用用户提供有价值的帮助^[11]。

复用度量主要用于判定复用对生产率、质量和开发时间的作用,它可以在不同级别上进行度量,包括构件级、产品簇级、项目级和机构级。复用度量的主要作用有:监控在过去时间内已复用的数量;为判定复用对软件生产力和质量的作用提供基础;加深对开发可复用软件的理解力;判定特定行为对复用数量的作用。

当前已存在不少复用度量的模型和准则,William Frakes和Carol Terry在文献[12]中总结了这些模型和准则,大致可分为以下4大类:经济模型类(economic models)、成熟度模型(maturity models)、复用比率模型(reuse

ratio models)以及复用潜力度量模型(reuse potential metrics and models).

目前比较系统化的可复用性度量指标主要有3种:基于要素-准则-度量模型的REBOOT方法^[11]、提出对构件质量和可复用性的评价标准的NATO标准^[13]以及US Army Reuse Center对提交给Defense Software Repository System(DSRs)的软件所采用的可复用性度量^[14].

目前的可复用性度量方法大都是根据构件的内部属性来得出度量结果的,当构件所处的上下文(context)也和其内部属性一样影响构件的可复用性时,就必须在度量构件可复用性时的同时考虑构件所处的领域属性和环境属性^[10].

2 青岛构件库(JBCL)的构件管理流程

青岛构件库的管理流程如图1所示.

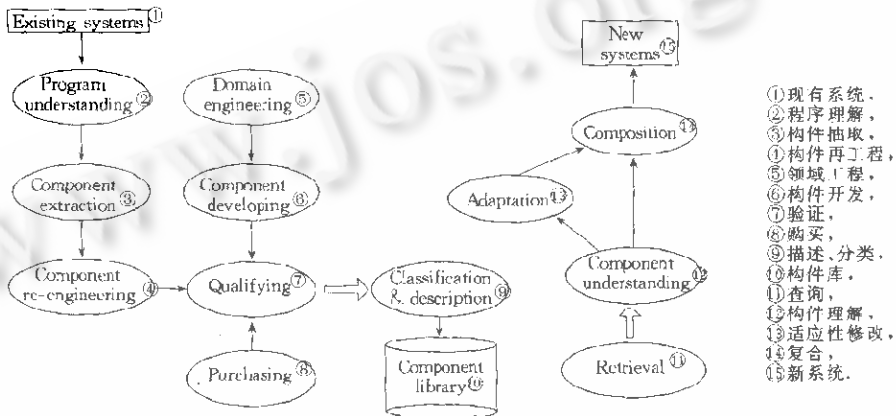


Fig. 1 Management process of JBCL
 图1 青岛构件库的管理流程

青岛构件库中的可复用构件可以在领域工程的基础上通过有目的的开发而获得,或者从现有系统中提取并进行再工程获得,也可以直接从某个商业性的软件机构购买,构件库需对构件进行验证,以确保构件满足一定的质量标准,然后进行分类和描述,将其存入构件库中.用户通过查询工具在构件库中查找满足自己需求的构件,如果找到,则将其复合组装到新的应用程序中,此过程中可能要进行适应性的修改^[3].

3 青岛构件库度量内容

度量主要有3个作用:度量构件的质量和可复用性、评估复用项目并估测复用的预期收益、为构件库提供所需的信息^[13].与此相对应,青岛构件库的度量也相应地有3个内容:青岛构件的质量和可复用性度量(quality and reusability metrics)、一套对采用复用技术开发的项目级和机构级复用效果进行评价的复用度量(reuse metrics)以及对构件库所需的相关信息的度量.例如,对可复用构件的使用、问题报告、失败的查询和其他构件配置管理所需的数据的度量,为构件进行配置管理和维护提供依据.

青岛构件的质量和可复用性度量包括前期度量和后期度量.前期度量在构件的开发期进行,主要基于静态程序度量,考察构件的内部属性;而后期度量则在构件已被复用后进行,主要基于复用用户的反馈信息,考察构件的外部属性.青岛构件度量模型是建立外部属性和内部属性之间合理联系的一个间接度量方法模型.前期度量所得的内部属性值可以根据这个模型推导出外部属性值,这些数值和后期度量得到的外部属性值可以进行比较.通过计算反馈信息和估算信息的差值来评价模型的准确度,而且还可以进一步指导调整度量模型的权值以提高模型的准确性.

针对如图1所示的青岛构件库管理流程,构件的前期度量包括两个过程,首先是在构件抽取前进行构件的质量和可复用性度量,度量结果可以为构件抽取提供基础,从而能够在现有系统中识别出可复用性高的构件来

加以提取.另外,在构件加入到构件库之前的验证过程中进行质量和可复用性度量,构件的度量结果加入到需要提交给构件库的构件的特殊文档中.构件的后期度量是基于复用户反馈信息的度量,用户在构件库中查找到满足需求的构件,然后对构件进行复用活动,在复用过程中得到的反馈信息由青鸟构件库的反馈信息收集工具进行收集,然后处理工具对收集到的数据进行检查、筛选等处理,最后由分析工具根据度量模型得出度量结果.

复用度量(reuse metrics)的指标包括:为软件增加可复用性带来的附加成本、可复用构件的复用次数、各种可复用构件(分析、设计、代码、测试可复用构件)的构成比例、可复用构件被修改后复用的次数、复用对于项目进度的影响等.这些度量有助于使软件复用的成本合理化,并为估计下一个项目的进度和成本提供经验^[12].

4 青鸟构件库构件度量方法

4.1 要素-准则-度量模型(Factor-Criteria-Metrics)简介

要素-准则-度量模型通常被作为软件评价的基础,和其他一些模型一起被 IEEE 和 ISO 所采用.早在 70 年代后半期,McCall 和 Boehm 就基于这个模型提出了两个软件质量层次模型.为了和国际标准相接近,青鸟构件可复用性和质量度量也以这个模型为基础.这个模型的基本原则是,每个属性可以被分解成一系列要素,而每个要素又可以分解成一系列准则,而准则值可以采用软件度量方法的评定而得到.如图 2 所示,描述了属性、要素、准则和度量的关系.

要素、准则和度量组成如图 2 所示的一个评价层次图.

(1) 在用户和管理层次上使用要素概念.所有非功能的软件需求在这个层次上的说明,如“软件应该有高的可维护性”等.

(2) 在软件设计人员和项目管理层上使用准则概念.在这个层次上应该实现每一个要素的需求.如“为了使软件有高可维护性,我们必须使软件有一致性、自描述性”等.

(3) 在软件和文档层次上使用度量概念.如“为了使软件有自描述性,我们必须有一个描述功能、参数的头”等.

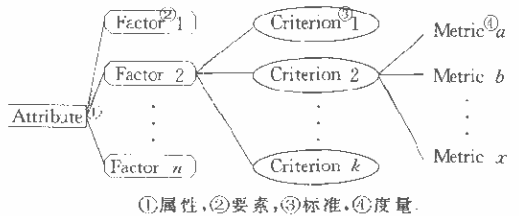


Fig. 2 Factor-Criteria-Metrics model
图2 要素-准则-度量模型

4.2 青鸟构件度量模型

青鸟构件度量模型基于以上的要素-准则-度量模型,主要参照了 REBOOT 实现模型^[11],从构件的质量和可复用性两个角度对青鸟构件进行评价.图 3 和图 4 分别是青鸟构件质量模型和青鸟构件可复用性模型.这两个模型主要针对的是青鸟构件库中的源代码构件,对于分析件、设计件、目标码、测试件等其他类型的构件,可以通过对以上模型进行简化、修改而得到.

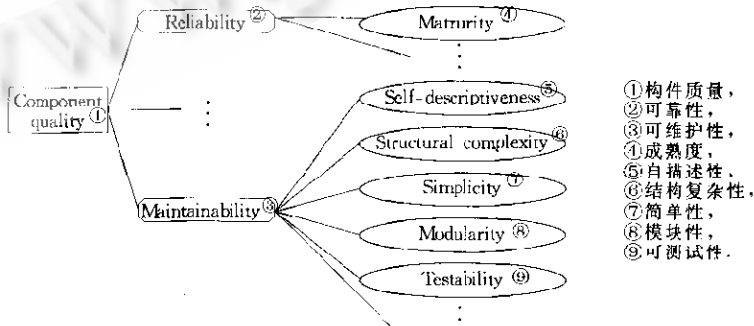


Fig. 3 Jade Brid component quality model (Adapted from REBOOT general model)
图3 青鸟构件质量模型(参考自REBOOT通用模型)

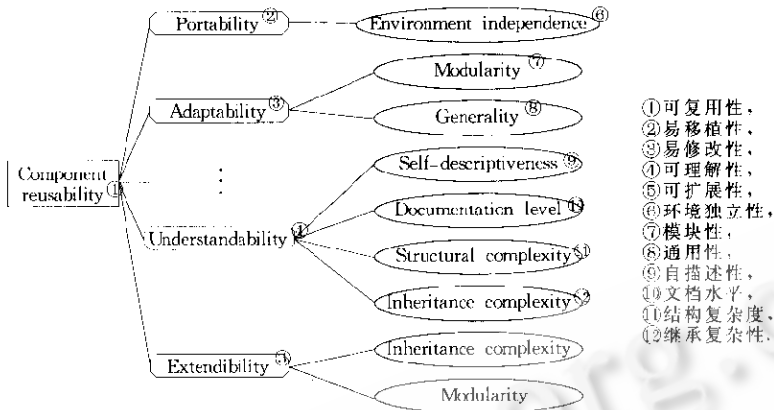


Fig. 4 Jade Bird component reusability model (Adapted from REBOOT general model)
 图4 青鸟构件可复用性模型(参考自REBOOT通用模型)

4.3 青鸟构件度量的实现

前期度量主要依据上述模型,由准则再分解成若干度量元去实现,例如,结构复杂度可以分解为成员函数扇入/扇出度量方法及McCabe的环计数度量方法等。这些度量方法都可以利用青鸟程序分析系统中的面向对象度量工具来实现,如图5所示。

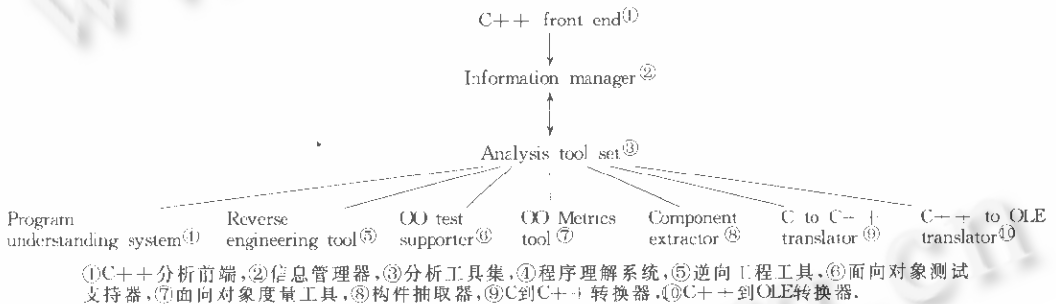


Fig. 5 Jade Bird program analysis system
 图5 青鸟程序分析系统

青鸟程序分析系统中的分析器前端分析C++的源代码,抽取程序信息并通过数据库服务器保存到程序信息库中。以上的所有分析工具集都通过数据库服务器来使用程序信息库中的信息。面向对象度量工具由程序信息库的信息依据度量模型生成度量结果集。青鸟构件度量可以使用面向对象度量工具来实现代码相关的度量值。

另外一些由准则分解出的度量方法,例如,可测试性准则分解成的可追溯性度量和需求质量度量等,则不能从源代码信息抽取而得,只能由度量者考察度量对象后,按类似于问卷调查的方法填写规定的检查表(check-list)来获取度量结果。后期度量主要是基于用户对构件复用的反馈信息,所以后期度量模型主要在要素和准则的级别上进行定义和度量。我们使用青鸟构件库的后期度量系统对后期度量提供自动化或半自动化的支持。

在后期度量系统中(如图6所示),通过收集工具捕获用户提交的反馈信息并存入临时库中,用户可以直接向收集工具提交反馈信息,当构件库连入INTERNET以后,收集工具也可以基于Web进行收集;处理工具提供半自动化功能,辅助管理员处理反馈信息,包括清理、聚集、多维化等,合格的反馈信息通过转换、集成子工具装载到用户反馈库中;分析工具为前端用户工具,通过访问分析工具可以查询、分析用户反馈库中的数据,它支持用户分析、评估构件并灵活定制度量权值。系统中的用户反馈库的数据模式采用数据仓库中通常使用的多维数据模式——星形模式(star schema)。

在获取前期或后期度量级别的初始数据后,对数据进行标准化处理,使数据统一落在0和1之间,接近0的

值代表可能会引起问题的度量值,接近 1 的数值代表符合特定标准的度量值。对标准化后的度量值进行加权综合评价可以得到准则层次的评价值。最后可以对后期度量和前期度量的结果进行比较,并可调整权值使得两者尽量保持一致。

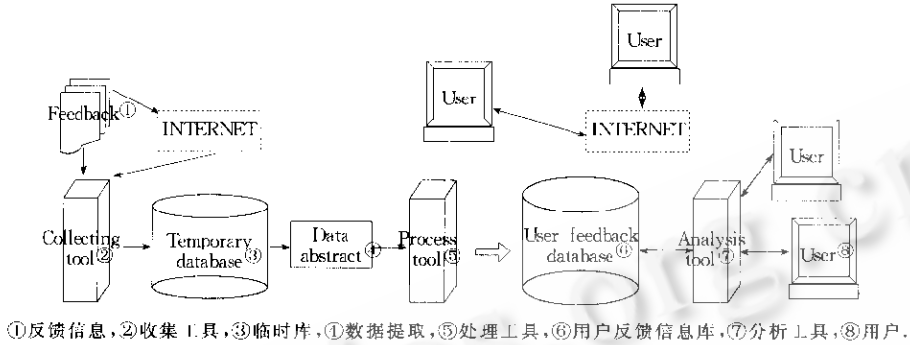


Fig. 6 JBCL posteriori metrics system
图6 青鸟后期度量系统

显示度量分析结果采用多维图形化的方式,并支持用户的多维分析。模型各个层次的评价值都可供用户浏览,使用户对度量的结果有一个直观的映像,结果的图形显示可以选择采用直方图的方法,也可以采用 Kiviat 图^[11]的方法等。另外,界面还支持用户灵活地调整各层次的权值。

在评价构件的可复用性时,构件所依赖的领域和环境特性在大多数情况下影响着用户对构件可复用性的评价,即同一构件在不同领域和环境中的可复用性是不同的,所以我们在后期度量的过程中加入构件在特定领域的使用反馈信息等,使得领域和环境的统计数据结合到对构件可复用性的度量中,作为构件评价时的参考值。在青鸟后期度量系统的反馈库中,包括时间维、用户维、构件提供者维,可支持多种分析。例如,不同供应商的构件使用情况;不同领域的用户对构件的不同评估;随着时间推移构件不同版本间的演化情况;不同产品族间的情况比较等。所以,青鸟后期度量系统能够有效地把领域和环境信息结合到度量体系之中。

5 相关工作讨论

青鸟构件库的构件度量在建立度量模型时参考了要素-准则-度量模型标准,着重参照了 REBOOT 实现模型,主要使用了青鸟程序分析系统中的面向对象度量工具和青鸟构件库后期度量系统来对青鸟构件进行度量,使得青鸟构件库中对构件等实体进行描述的信息包含了可复用性和质量度量的评价信息,这给使用构件库中构件的复用用户提供了有价值的帮助。同时,在对构件度量的过程中考虑了对构件所依赖的领域和环境特性对构件可复用性的影响,这给复用用户提供了有用的评价辅助信息。后期度量系统中的用户反馈库的数据模式采用数据仓库技术,为复用者理解和选取构件以及构件库各个层次的管理人员管理和改进构件库系统提供了一定程度的辅助决策支持。

REBOOT 包括一个存储可复用构件的构件库和一组产生、认证、插入、提取、评价和适配可复用构件的工具。它开发了一个基于要素-准则-度量模型标准的度量体系^[11],其中的度量方法包括依赖于诸如检查表等的主观性方法以及直接从代码中分析的方法。目前,REBOOT 中对构件的描述仅针对单个构件,且都是文本类型,当用户检索到一组候选构件时,如何提供量化的比较信息,使用户快速选取最能满足需求的构件,REBOOT 中尚未提供有效的技术支持。如何有效地组织用户反馈信息以改进构件度量,正在进一步的研究当中。

NATO 制定了一整套软件复用的指导性标准,它推荐将以下 4 个度量作为对构件质量和可复用性的评价标准:复用者考虑复用该构件的次数、复用者实际复用该构件的次数、复杂性、问题报告次数即构件的显著缺陷个数^[12]。

US Army Reuse Center 对提交给 DSRS(defense software repository system)的所有软件都进行了检查,其中一系列的可复用性评价也作为检查的一部分,采用基于 Ada 源代码分析工具进行两阶段的可复用性度量,第

1 阶段的度量采用了由分析工具支持的 31 种度量方法,第 2 阶段采用了 150 种度量方法. 这个系统采用了规模较大的可复用性度量,但是对后期度量没有提供足够的支持^[14].

6 结束语

实际上,要对构件的可复用性作一个全面而令人满意的评价和度量总是很困难的,构件的可复用性受到外界很多因素的影响. 例如,复用用户在编程语言和应用领域等方面的经验影响着对构件的可复用性的评价,因为每个用户都是从他们自己的角度去评价一个构件相对于他的可复用性. 而我们的度量系统只是设法寻找出这些复用用户在复用构件过程中共同关心和涉及的构件特性的度量结果,以提供给用户作为参考.

今后需要对以下一些问题作进一步的研究:首先是如何更好地把整个度量结果以易理解的形式显示给用户,让用户对度量的结果有一个直观的印象. 其次,对构件库中的非源代码构件的度量需要建立更适当的度量模型区分而加以考虑. 再次,完整的青鸟构件库的度量体系还需要实现青鸟复用度量参考模型以及对构件库所需的相关信息的度量模型. 最后,还必须经过大量的实例研究来验证模型的正确性或可用性.

参考文献

- 1 Mili H, Mili F, Mili A. Reusing software: issues and research directions. *IEEE Transactions on Software Engineering*, 1995,21(6):528~532
- 2 Li Ke-qin, Guo Li-feng, Mei Hong *et al.* An overview of JB (Jade Bird) component library system JBCL. In: Chen Jian, Li Ming-shu, Christine Mingins *et al.* eds. *Proceedings of the 24th International Conference TOOLS Asia*. California: IEEE Computer Society Press, 1997. 261~267
- 3 Kearney J K, Sedlmeyer R L, Thompson W B *et al.* Software complexity measurement. *Communications of the ACM*, 1986,29(11):1044~1050
- 4 Weyuker E. Evaluating software complexity measures. *IEEE Transactions on Software Engineering*, 1988,14(9):1357~1365
- 5 McCabe T J. A complexity measure. *IEEE Transactions on Software Engineering*, 1976,2(4):308~320
- 6 Halstead M H. *Elements of Software Science*. New York, Elsevier North-Holland, 1977
- 7 Albrecht A J, Gaffrey S H. Software function, source lines of code and development effort: prediction: a software science validation. *IEEE Transactions on Software Engineering*, 1983,9(6):639~648
- 8 Chidamber S R, Kemerer C F. A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 1994,20(6):476~492
- 9 Lorenz M. *Object-Oriented Software Development: A Practical Guide*. New York: Prentice Hall, Inc., 1993
- 10 Poulin J S. Measuring software reusability. In: Frakes W B ed. *Proceedings of the 3rd International Conference on Software Reuse: Advances in Software Reusability*. Los Alamitos, CA: IEEE Computer Society Press, 1994. 126~138
- 11 Chichester Even-Andre K. *Software Reuse: A Holistic Approach*. New York, NY: John Wiley & Sons, Inc., 1995
- 12 Frakes W B, Christopher J F. *Software Reuse Survey Report*. Sterling, VA: Software Engineering Guild, 1993
- 13 NATO. *NATO Standard for Developing Reusable Software Components*. Vol. 1, NATO Contact Number CO-5957-ADA, 1991
- 14 RAPID. *RAPID Center Standards for Reusable Software*. U. S. Army Information Systems Engineering Command, 3451-4-912/6.4, October 1990

Component Metrics in Jade Bird Component Library System

MEI Hong XIE Tao YUAN Wang-hong YANG Fu-qing

(Department of Computer Science and Technology Beijing University Beijing 100871)

Abstract Focusing on software productivity and software quality demanded by the development of software industry has spurred the research on software reuse technology. Therefore, the metrics about software reuse technology are also attracting more attention. The goal of Jade Bird Component Library (JBCL) system is to

describe, manage, store and retrieve components in order to fulfil the requirement of component-reuse-based software development process. This article introduces the component metrics model in JBCL and discusses its implementation based on the Object-Oriented Metrics Tool of Jade Bird Program Analysis System (JBPAS) and JBCL. Posteriori Metrics System, which is made up of Feedback Collection, Analysis and Processing Tools.

Key words Software reuse, component library system, software metrics, object-oriented metrics, reusability metrics, reuse metrics.