

基于 Java 的软件再工程支持工具研究

彭 鑫 赵文耘 夏宽理 沈 铨

(复旦大学计算机科学与工程系软件工程实验室,上海 200433)

E-mail: xspengxin@163.com

摘 要 Internet 的发展和构件化的软件开发为再工程指明了新的方向。面向对象软件再工程将以构件库为中心,并关注于将遗产系统改造为 B/S 结构的 Internet 应用。论文提出了一个面向对象软件再工程模型,分析了面向对象软件再工程支持工具的特性,然后介绍了基于 Java 的再工程支持工具原型。

关键词 软件再工程 支持工具 逆向工程 再工程

文章编号 1002-8331-(2003)18-0063-04 文献标识码 A 中图分类号 TP311

Research on Support Tools Based on Java for Software Reengineering

Peng Xin Zhao Wenyun Xia Kuanli Shen Cheng

(Computer Science and Engineering Department of Fudan University, Shanghai 200433)

Abstract: Internet and component-based software development have presented a new orientation for reengineering. Object-oriented software reengineering should be based on component library and focus on transforming legacy systems into B/S Applications. This paper presents a model of object-oriented software reengineering, discusses support tools and introduces the tool prototype.

Keywords: Software Reengineering Support Tools Reverse Engineering Reengineering

1 引言

面向对象方法已经成为软件开发的主流。然而仅仅运用面向对象技术并不足以提供具有良好适应性的系统。面向对象设计经验的不足、面向对象技术发展过程中的不成熟都会导致原有系统缺乏可演化性,无法很好地通过维护来满足用户需求的变动。而再工程提供了一条把遗产系统转化为可演化系统的切实可行的途径^[1]。

再工程所面临的遗产系统往往都很复杂,完全由实施者手工完成重构过程比较困难。因此自动化的工具支持就显得尤为重要。国内外许多软件组织都在再工程方法以及支持工具上进行了一些研究,并且已经有了一些工具产品:

(1) 青鸟程序理解系统 JBPAS 包含一组分析工具,其中包括程序理解工具、设计文档逆向生成工具、面向对象度量工具、构件提取工具、C 到 C++ 转换工具、C++ 到 OLE 转换工具等具有再工程性质的支持工具。

(2) 瑞士伯尔尼大学的 FAMOOS 项目是欧共体资助的一个研究项目^[2],这项研究的成果是开发出了一套再工程方法和支持工具以及支持环境。其中包括软件质量评估、问题侦测、复写代码侦测、形象化度量结果描述等支持功能。

(3) McCabe IQ 是美国 McCabe & Association 公司的产品。其中的 McCabe Reengineering 模块支持对已有软件系统的维护、改变软件特性或移植到新的平台或结构中。支持的功能包括识别代码中的冗余代码以及进行冒险(risk)分析。

基于 Internet 的以构件库为核心的软件开发将成为商业应用的主流。新的再工程方向关注于将已有的面向对象软件系

统向构件化的 Internet 应用的方向转化。因此相应的支持工具也必须支持构件化的演化方向,并考虑与企业构件库的无缝连接。

以下的第 2 部分提出了一个面向对象软件再工程模型。第 3 部分对再工程支持工具的特性进行了讨论并介绍了作者的再工程支持工具原型 FDRengineer,其中的一些实现细节将在第 4 部分涉及。最后的第 5 部分进行了总结并对今后的工作方向进行了探讨。

2 面向对象软件再工程模型

再工程面临的遗产系统经常十分复杂,更糟糕的是还会面临缺少描述清晰、组织良好的相关文档的情况。这时候由程序源代码(可能通过反编译获得)出发重新理解遗产系统,然后通过决策选择做出下一步重构改造的实施策略几乎是唯一的选择。如果没有自动化工具的支持,这个过程将十分困难。

由于面向对象技术自身的特点,面向对象软件系统的再工程策略与结构化系统有很大不同,而构件技术对再工程策略也有很大影响。基于构件的系统是容易进行演化的系统,也就是更容易进行再工程的系统^[1]。因此,在面向对象软件的再工程过程中要充分考虑构件化的方向。

根据用户对现有软件改进要求的不同,再工程活动一般可分为系统级、数据级和源程序级三个层次^[2]。多数情况下的再工程过程都会涉及到系统级的再工程活动。数据级和程序级的再工程既然已经超出常规维护所能解决的范围,那么就很难局限于数据和程序的层次开展。只有从系统的底层形态(代码、文档

基金项目:国家 863 高科技研究发展计划项目(基于 Internet 以构件库为核心的软件平台,编号:2001AA1100241)资助

作者简介:彭鑫(1979-),男,硕士研究生,主要研究方向:软件工程。赵文耘,男,教授,主要研究方向:软件工程。夏宽理,男,教授,主要研究方向:软件工程。沈铨,男,硕士研究生,主要研究方向:软件工程。

等)经过多次抽象提升回到概念世界,并以此为起点结合再工程改进要求,逐步实现为新的底层形态,遗产系统到新系统的转换才会平稳地进行。

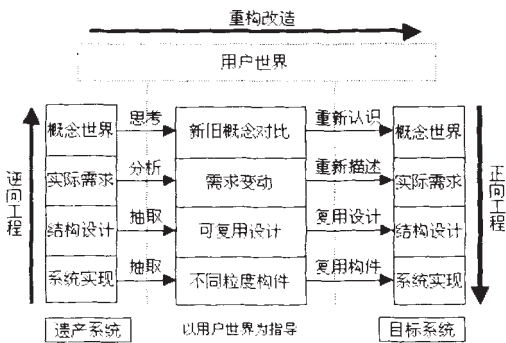


图1 面向对象软件再工程模型

再工程必须充分发掘遗产系统中可加以很好利用的各方面,否则再过程将蜕化为一般的二次开发。特别是在面向对象环境下,软件复用技术将促使再工程在各抽象层次上通过充分复用的形式完成向目标系统的转化。图1是面向对象系统再工程的一般模型。从图中可以清楚地看出再工程是一个从遗产系统的具体实现出发到新系统的具体实现结束的倒U字形过程。整个抽象(逆向部分)再具体(正向部分)的过程都以人(再工程实施者)为主体进行,其中的工具支持也很重要。逆向阶段的各个抽象层次在一定阶段还要为正向过程中对应层次生产可复用资源,其中的抽象、选取等工作都要在更高抽象层指导下以用户实际需求为准则进行。正向过程将充分消费这些可复用资源,产生各层次上的相应系统实现。

3 面向对象软件再工程支持工具的设计

3.1 面向对象软件再工程支持工具的一般特性

再工程的出发点是运行中的软件系统。“软件”的范围是广泛的,它包括源代码、设计记录以及其它文档资源等。这些资源中,设计记录等文档资源由于其自然语言的特性,基本上很难由自动化工具加以利用。源代码由于遵循某种程序设计语言特定的词法和文法规则,在某种程度上可以使用自动化工具进行自主理解。

同样,支持工具在图1中结构设计以上的抽象层次中发挥作用的余地较小。适合于工具支持的部分是结构设计和系统实现这两个层次。支持工具可以较好地地从程序代码中抽象出多个不同抽象层次上的结构设计视图。再工程实施者在此基础上结

合文档及设计经验等可获得较完全的遗产系统结构设计视图。从遗产系统代码和结构设计中提取可复用构件、设计的过程同样需要人工干预。支持工具在这方面只能以某一客观尺度进行工作,但某一个具体的再工程环境下的相关过程都必须体现实施者的主观选择。因此必须考虑工具支持与主观选择结合的一个较好的结合点。

按照构件化的演化方向,正向部分通常应该结合企业构件库和组装工具平台完成。因此再工程支持工具应该侧重于扮演构件生产者的角色,按照具体的再工程需求生产构件,并通过相关的接口与构件库、组装平台无缝连接。

3.2 基于Java的再工程支持工具FDReengineer

基于java的再工程支持工具FDReengineer采用以数据库为核心的整体架构。分析获得的各种基本信息均在数据库相关数据表中存储,并以存储信息为进一步工作的起点。考虑到Jbulider工程文件的良好组织性,作者选择Jbulider工程作为分析对象。

FDReengineer是再工程支持工具的第一阶段版本,其中主要体现了逆向阶段的相关支持功能。FDReengineer的功能模型如图2所示。支持工作流程从源代码整理入库开始。代码入库后即进行类结构分析、类间引用关系分析及数据库访问分析。分析结果入库后待应用分析结果进行进一步工作。其中的核心工作是在用户具体需求的指导下完成构件提取、筛选和包装入库的工作。正向部分的工作需要与企业构件库以及组装平台联系起来。目前已经有了一些企业级的构件库管理系统,只要取得相关的接口信息就可以与构件库无缝连接。

3.2.1 生成UML类图

类图展现了系统中类的静态结构,即类与类之间的相互联系。类之间有多种联系方式,这里主要体现泛化、聚集、依赖这三种关系,分别对应于面向对象系统中继承、成员对象、引用的关系。面向对象系统中,类是经过良好封装的逻辑实体,也是最基本的抽象单位。因此类图通常是系统最低抽象层次上的表示。但原有的类结构也有可能需要修改,例如某些设计不合理或者有缺陷的类。

FDReengineer生成两种类图:全局类图和局部类图。全局类图体现整个系统中各类之间的相互关系。通过这种类似于树形的组织结构可以较好地获得各类在系统中所处地位的感性认识。局部类图以一个类为观察对象,具体体现该类内部结构以及与其他类的关系。该类内部结构(属性和方法)的访问控制特性通过相应图标体现。局部类图还支持当前类方法代码查看。全局类图和局部类图均支持类间导航。在类间导航的过程中,相应类的各项度量指标会同时提供,以帮助实施者获得对

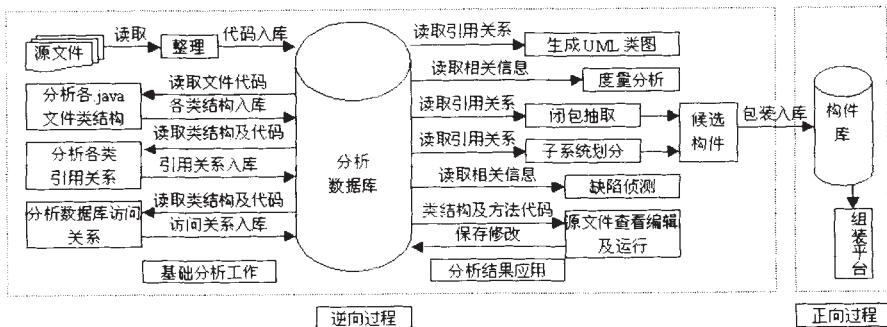


图2 FDReengineer功能模型

该类的充分认识。

3.2.2 度量分析

度量是一种以量化的方式获取软件属性的方法。作者认为度量能从大量的软件代码中提取出某些具有丰富涵义的评价信息,可以帮助再工程实施者获得一部分系统理解。同时可根据度量结果对相应指标做出评价,帮助实施者对再工程实施细节进行决策。

FDRengineer 提供了一些常用的度量指标,如代码长度、继承深度等。这里介绍一下类包内聚度。设类包中包含 N 个类,其中共有 R 个从引用类指向被引用类的(只考虑类包局部)引用关系,那么此类包的内聚度计算方法为: $\frac{R}{N*(N-1)}$ 。这

项度量可以在构件提取过程中用来评价类包内部结构的紧密程度,从而为实施者提供一个衡量其作为构件的适合度的参考指标。

在支持工具中,度量指标既可以作为分析中的某个类的相关指标提供,也可以作为分析尺度从系统所有类中寻找符合相关度量值范围的目标。

3.2.3 缺陷侦测及系统划分

遗产系统数量巨大的代码之中往往隐藏着许多设计缺陷,侦测并修改这些缺陷以保证获得高质量的候选构件也是再工程的一项重要内容。

而构件的提取必须通过对各类间引用关系的分析,结合系统划分进行。闭包是一种比较粗糙原始的划分方法,必须使用更精细的方法加以改进。

第4部分将详细介绍相关内容。

3.2.4 代码查看编辑及运行

FDRengineer 提供对类图中显示的当前类的源文件进行修改、编译和重新分析入库的功能。对文件进行了修改之后可以通过刷新重新生成类间关系图等。工具还提供了对当前项目的执行功能,可以在本工具环境下执行当前分析的项目。

4 相关问题的讨论

4.1 缺陷侦测

面向对象设计有一些基本的设计原则。发现并改造遗产系统中的不良设计也是再工程的一项重要内容。目前 FDRengineer 可以检查诸如“基类对子类存在依赖”、“聚合成分类对聚合体类存在依赖”这样的潜在不良设计。另外,通过基本的度量指标也能发现一些不良设计。

现实的软件开发中经常存在的一个现象是大量复制代码。许多相似的代码经复制或稍加修改后在另一个地方运行。这样做的一个问题是差错会随着复制传播。而且这种复制是隐性存在于系统中的,当需要进行修改时,即使是最初的程序编写者也无法准确定位所有的复写代码位置。作者的支持工具提供复写代码侦测功能。代码比较通常针对两个方法进行。通过一一比较每一条语句获得这两个方法的语句相似矩阵,然后从矩阵中平行于对角线的各方向发现复写代码。为了减少无关成分的干扰,语句在进行比较前都经过了去掉注释、多余的空格和字符串常量内容的过程。发现的复写代码一方面可以帮助跟踪错误传播情况,另一方面这些复写代码也是潜在的改造对象,例如提取出来成为一个功能性方法等。

4.2 系统划分

类图体现了遗产系统底层的设计结构。为了更好地体现原有系统的设计结构并为正向阶段的设计提供可复用资源,支持工具必须支持抽象层次的提升以提供更高层次(例如包、模块或者子系统级等)上的设计抽象。

4.2.1 闭包抽取

闭包是一个封闭的功能单元,包含了完成入口类(可以是一个或多个)功能所直接或间接需要的全部相关类。这样的一个单元作为构件往往会显得很臃肿,许多引用层次较高的类的闭包几乎就是整个系统,而许多底层单元频繁重复出现在很多类的闭包中。闭包方法是一种比较原始的构件抽取方法,但这种方法可以严格保证功能的完整性。

4.2.2 子系统划分

闭包方法中引用关系是唯一的判断标准。这样划分出来的类包过于臃肿,并且各类包都包含大量重复出现的类,特别是一些提供公共服务功能的类。改进的子系统划分方法基于对引用关系的进一步分析,因此能取得更好的效果。具体的划分方法如下:

初始类包为全部入口类(一个或多个)。划分过程依次考察当前类包中每一个类所直接引用的类,以一定的判定算法评价该类是否应该加入当前类包。如果该类达到标准则加入当前类包否则暂不加入,然后继续考察。当考察过程中已经不再有新的类加入时,考察过程可以到达当前类包的末尾,这时候划分过程结束,得到从初始入口开始的类包。划分过程中的停止点(一些考察后没有被加入类)可以在综合评估后作为新的入口开始系统其他部分的划分。划分过程中的评价标准主要反映一个类与当前类包的关系紧密程度。对于某类 A ,目前的评价指标由两部分组成:对 A 的引用中来自当前类包的比例和对该类有引用关系的类占系统全部类的比例。其中,第一部分表达了类 A 对当前类包的专属程度,第二部分表达了这个类作为整个系统共用服务类的程度。通过这种自动划分方法,遗产系统被分为多个部分。其中的一些是联系紧密从而被划分在一起的子系统,还有一些提供系统公用服务功能的类被单独划分出来。划分出来的子系统可以作为复用的候选,粒度较小的部分还可以作为以后相关领域开发的候选构件。

闭包方法其实是这种划分方法将评价标准设为 0 的情况下的特例。在闭包方法下,由入口开始的所有被考察的类都将被加入类包。而改进的划分方法有两点好处:一是解决了非简单树形结构下一些多子系统重叠部分的归属,这是通过评价达到的;二是将一些公共服务类单独划分出来,从而避免了它们重复出现于多个类包中。

例如在图3所示的类树中,如果由类 B 出发划分一个子系统,那么根据闭包算法得到的将是 $\{B, E, F, G, H\}$ 。其中的类 F 还将出现在类 C 所在的子系统中,而类 H 由于为很多类提供服务,更将重复出现在多个子系统中。按照改进的划分算法,从 B 开始考察, E 将首先被加入这个类包,此时考察 F 由于引用 F 的 G, C 都不在当前类包中,因此 F 暂时达不到加入标准。然后考察到 G 达到加入标准,当前类包扩大到 $\{B, E, G\}$ 。这时开始考察 E 的直接引用, F 再次成为考察对象。此时引用 F 的四个类中 B, E, G 已经在当前类包中, F 的评价指数将符合标准,于是 F 加入当前类包,而从 C 开始的类包将不会包括 F 。观察 H , 引用 H 的类比较多,而且分布在 B, C, D 这几个分支之中,因此无论在哪个分支中考察 H ,两部分评价标准都将促使 H 单

独划分出来。于是改进的划分算法由类 B 出发将得到划分 {B、E、F、G} ,而且还能将 H 单独划分出来。由此可见这种方法可以取得比闭包方法更好的效果。

从遗产系统划分得到子系统后 ,可以用同样的方法处理每一个子系统。这时将只考虑子系统内部的依赖关系 ,从而得到更低抽象层次的系统划分。划分的抽象层次不同 ,所适合的划分评价标准也有所不同。因此划分算法的标准可以灵活控制。按照作者的经验 ,一般较高抽象层次上的划分标准较松 ,以划分出较大的子系统。随着抽象层次降低 ,划分标准将提高 ,以划分出粒度较小的模块。

考虑正向过程的需要 ,抽取大粒度构件将十分有利于节约成本提高再工程质量。如果站在更高角度上考虑本机构未来发展的需要 ,那么可以着眼于一些领域相关性较小或者符合机构内未来领域需要的通用构件 ,这样的构件一般粒度较小。后者的出发点已经超出了一般的再工程范畴 ,但却是一种很有价值的策略 ,特别是如果能和构件库无缝集成的话。

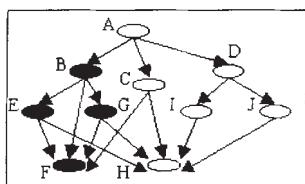


图3 子系统划分

(上接 62 页)

各不相同 ,彼此之间通过通信网络连接在一起 ,并以此为基础进行彼此协商、协调、协作 ,同时所有的 Agent 都与系统的黑板子系统连接在一起 ,通过黑板共享彼此之间的结果 ,且系统通过黑板子系统进行任务的分解、控制任务的解题方向和解题进度。

6.5 软件工程领域

当前软件工程领域主要的开发方法为面向对象开发方法 ,该方法以其特有的类、对象、继承、封装、消息传递等机制 ,实现了对客观事物的有力描述 ,实现了问题分析与处理的统一 ,大大提高了开发效率。然而 ,随着人们研究的不断深入 ,人们对软件的要求进一步提高 ,面向对象技术日益暴露出其固有的缺陷 :缺少自主性与主动性 ,在用对象描述自然界中有生命或有行为能力的组织、实体时便无能为力。采用面向 Agent 的软件开发方法可以构造一些复杂的、自我包含的组件 ,并且使它们能与其它一些独立开发的类似组件灵活交互。交互作用不再通过严格预定的接口进行 ,而是通过参与者之间的协商、承诺、同意等行为来实现。这意味着系统的整体性质和行为不能预先在程序中固定 ,而是在运行时通过各参与 Agent 的行为和交互作用来动态实现。因此 ,面向 Agent 的软件工程具有以下显著的优势 :数据、控制、资源的分布性、自然的表述方式且便于现有软件系统的集成。

6.6 计算机仿真

目前 ,面向对象仿真方法是当前最广泛应用的仿真形式 ,并在工业控制、航天、军事、战场模拟等方面获得了广泛的应用。但面向对象的方法在模拟具有智能的对象中有其固有的不足 ,这在很大程度上限制了仿真的效果。随着 Agent 技术的出现 ,为人们进行真正意义上的智能仿真提供了有效手段。Agent 可能动地根据外界的条件调整自己的状态及反应的能力 ,大大

5 结束语

以构件库为核心的开发方式为面向对象软件再工程提出了新的框架 ,再工程支持工具将与构件库和组装平台融合起来构成完整的再工程支持体系。其中 ,再工程支持工具的出发点是帮助用户获得清晰的遗产系统理解并改进源代码质量 ,然后在用户的干预指导下生产出适应具体再工程需求并且经过良好封装的待用构件。

FDRengineer 目前的系统划分已经可以产生一些候选构件 ,下一步将继续研究更精细的提取方法。同时为了保证抽取出的构件能够比较可靠、容易地在重构中加以应用 ,下一步还要将它们封装为 JavaBean/EJB 构件 ,并开发相应的接口与构件库管理系统衔接。(收稿日期 2003 年 1 月)

参考文献

- 1.郭耀,袁望洪,陈向葵等.再工程——概念及框架[J].计算机科学, 1999 26 :78~83
- 2.谷耀.软件再工程及其应用[J].信息系统工程,1994 7 (1) :50~53
- 3.王超,陈力军,赵洪兵等.一个软件再工程的实例分析[J].计算机工程与应用, 2001 37 (15) :101~104
- 4.朱三元,钱乐秋,宿为民(编著).软件工程技术概论[M].北京:科学出版社, 2002-01
- 5.Holger B'ar, Markus Bauer, Oliver Ciupke 等.The FAMOOS Object-Oriented Reengineering Handbook.http://dis.sema.es/projects/FAMOOS

扩展了客观对象的空间 ,为模拟有生命的对象以及具有一定内在变化规律的对象实体的模拟提供了有效的工具 ,基于 MAS 的智能仿真必将成为今后主要的仿真形式。

7 结论及展望

MAS 技术是当前分布式人工智能及计算机科学领域的研究热点 ,它为分布式问题的求解提供了一种新颖的思想及高效可行的解决方案。应该指出 ,作为一个新的学术热点 ,关于 MAS 技术的研究多数尚处在理论研究及原型系统设计阶段 ,具备实用价值的研究成果还很欠缺。论文主要对 MAS 关键技术进行了阐述 ,并指出其在智能交通、多机器人系统及 Internet 网络管理等领域的巨大应用前景 ,以期起到抛砖引玉的作用。

(收稿日期 2003 年 1 月)

参考文献

- 1.Yannis I_labrov, Tim Finin, Yun Peng.Agent Communication Languages : The Current Landscape[J].IEEE, 1999 3
- 2.Nicholas R Jemfings, Katia Sycara, Michael Woohlridge.A roadmap of agent research and development.Autonomous Agents and Multi-Agent Systems, 1 7~38
- 3.Smith R, Davis R.Frameworks for Cooperation in Distributed Problem Solving[J].IEEE Transactions on Systems, MAN and Cybernetics, Bol. SMC-11, 1981 1
- 4.Muller H J.Negotiation principles[C].In Foundation of Distributed Artificial Intelligence 6th Generation Computer Technology Series, New York, John Wiley & Sons Inc, 1996 211~229
- 5.Katia P Sycara.Multiagent Systems[J].AI Magazine, 1998