基于代理的程序挖掘系统的设计与实现

夏东林,张尧学,方存好
(清华大学计算机系,北京 100084)

摘 要: 随着 Internet 和软件复用技术的发展,用户根据自己的计算需求,从网上构件库中搜索构件,并由构件动态生成程序成为可能.针对这种按需计算的需求,本文提出一种在网络环境下进行程序挖掘的方法:采用智能代理分析理解用户的计算需求,从网上构件库中识别、搜索可用构件,并组装链接,形成满足用户要求的可执行程序.文中论述了程序挖掘的基本概念、基于代理的程序挖掘原型系统,并讨论了进一步的研究方向.

关键词: 程序挖掘; 代理; 软件复用; 构件; 构件检索; 领域工程
中图分类号: TP393 **文献标识码**: A **文章编号**: 0372-2112 (2003) 05-0793-04

Design and Implementation of an Agent-Based Program Mining System

XIA Dong-lin, ZHANG Yao-xue, FANG Cun-hao
(Computer Science Department, Tsinghua University, Beijing 100084, China)

Abstract: With the development of Internet and software reuse, it will be possible that end user who wants to customize some functional feature can get such program by finding some components from component libraries and compiling them according to the user's demand. A program mining methodology under networked environments is presented: with the aid of intelligent agents, analyzing and understanding user's requests for computing, identifying and searching component candidates from online component libraries, composing and reassembling them to form programs that perform the expected computing. The basic concepts, system framework, overall process and agent platform of program mining are investigated. The research issues are also discussed to present an open research area.

Key words: program mining; intelligent agent; software reuse; component; component retrieval; domain engineering

1 引言

随着 Internet 从信息发布与媒体共享到分布式计算平台的演变,越来越多的用户要求在网上定制计算服务.一方面,用户需要根据不同的网络环境和不同的资源条件灵活选择服务功能;另一方面,主动网的研究使网络成为一种可编程网络[1],要求其根据需要动态生成网络结构.针对这种按需计算的需求,我们提出了一种新的计算模式——程序挖掘.它的基本思想是利用多个智能代理,分析用户的计算需求,从 Internet 上大量的构件资源中检索获取所需构件,并把这些构件组装成满足用户要求的程序[2].与程序挖掘思想相似的概念有 Sun 的 Sun ONE,微软的.NET,Compaq 的按需计算等.

程序挖掘的实现有三个前提:Internet 上有大量的可复用构件资源;构件之间可以容易的组装,即能够复用;存在一个有效的程序挖掘方法.随着构件技术的发展,Internet 上已经出现了大量的构件库,如 Alphaworks, ComponetSource, ComponentPlanet 等.虽然目前关于通用的标准构件技术还正在发展,但构件标准如 JavaBean/ EJB, COM/ DCOM, CORBA 为构件的可复用打下了基础.本文将介绍程序挖掘的一般过程,讨论其中的关键技术,并给出原型系统证明这种计算模式的可行性.

2 程序挖掘介绍

程序挖掘是一种根据用户需求,为用户量身定制应用程序的过程,一般可以顺序的分为以下 5 个步骤:①用户提交计算需求;②查找相应的组装方案;③根据组装方案搜索所需构件;④依照组装方案将构件组装成程序;⑤返回用户所需的程序.根据这个步骤流程,程序挖掘系统可分为以下几个功能模块,如图(1)所示:智能接口子系统,需求分析与构件获取子系统,知识库,构件目录与构件资源子系统,和构件组装与代码生产子系统.下面分别介绍:

智能接口子系统通过提供友好的输入界面帮助用户更准确地提交请求.按照程序的应用领域和功能提供相应的主题词表和图文导航目录,在应用领域和程序功能两个层次对用户的计算请求进行限定,避免模糊和歧义性,同时为在相应范围内选取可用构件创造条件,提高构件搜索和获取的针对性.

需求分析与构件获取子系统根据用户的查询信息可以得

收稿日期:2002-08-24;修回日期:2003-01-28
基金项目:国家 973 计划(No. G1998030409)

到用户所需程序的初步描述. 然后通过知识库中的领域信息确定相应领域. 再根据用户所需的计算功能,在该领域中选定包含这些计算功能的问题解决方案,其具体体现为解决这个问题的一个软件体系结构. 对它按照用户需求进行裁剪定制就形成了组装方案. 然后根据组装方案向构件目录与构件资源子系统搜索所需构件. 最后向构件组装与代码生成子系统提交组装方案和构件. 可见需求分析与构件获取子系统是整个系统的核心,它连接了各个子系统.

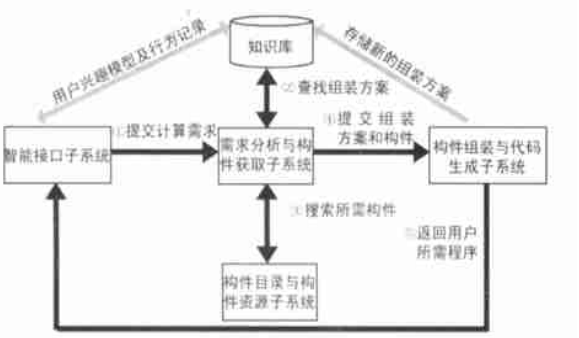

图 1 程序挖掘系统功能模块图

知识库存放着领域知识、用户记录、组装方案. 与知识库交互的有智能接口子系统,需求分析与构件获取子系统,构件组装与代码生成子系统. 智能接口子系统根据知识库中的领域信息向用户进行领域导航,同时还将用户兴趣模型及行为记录在知识库中,以便下次向用户提供更为方便的查询界面. 在领域知识中问题解决方案的软件体系结构则为用户的计算需求提供了原始方案. 所有的程序挖掘组装方案都是由这些原始方案生成的. 在经过需求分析以后产生的新组装方案,如果通过构件组装验证可行,则会存储到知识库以备以后复用.

构件目录与构件资源子系统储备标准构件,并提供统一的构件检索工具. 用于组装的构件通常由不同的供应商提供,有各自构件描述和检索方法. 因此,需要一个统一描述的构件库,存放各个异构构件库的信息,并提供统一的构件检索,我们称之为构件目录库. 构件目录库可以只存放构件描述信息和构件资源信息,而没有构件实体. 众多构件库在网上分布存在,需要有数据更新机制保证构件目录库和构件库的信息一致,并能自动把各种不同的构件描述转换为统一的构件描述.

构件组装与代码生成子系统按照构件组装方案定义的构件依赖和调用关系,对选定的构件进行连线和组装. 在组装过程中,需要根据用户的具体需求调整和设置各个构件的可定制属性,通过事件触发和方法调用将构件连接成一个统一的整体,并自动生成相关的程序代码,完成用户所要求的计算功能. 其中,自动生成的代码主要包括事件发出和事件响应方法之间的适配代码,构件之间的动态参数传递,以及支持整个程序执行的静态入口方法和变量设置等. 构件之间的调用关系可以用某种构件配置和连线语言描述,例如 Caltech 的 KDL (Kind Description Language)语言,IBM 的 BML (Bean Markup Language)语言等.

以上五个子系统相互联系,相互影响,形成一个按需计算的程序挖掘系统.

3 程序挖掘原型系统

根据上文介绍的程序挖掘过程,本实验室开发了一个基于代理的程序挖掘原型系统,初步验证了这种计算模式的可行性. 基于智能代理的程序挖掘有以下几个优点. 系统结构清晰:各个模块之间的关系是通过代理系统的交互协议实现的,易于分而治之. 程序挖掘无论是对 Internet 上的资源进行检索,还是实现按需计算时的分析与计算都需要基于知识的推理,而智能代理本身在实现结构,交互语言等各方面都有利于引入机器推理. 而且代理的可移动性,使它可以在远端执行,这样就没必要立即传送原始数据,只要传送结果就可以了,大大节约了网络带宽,避免网络延迟.

下文将首先介绍程序挖掘系统中代理的设计,然后讨论系统实现中的关键问题,如构件资源的组织,构件组装与代码生成,最后给出综合领域知识的程序挖掘算法.

3.1 程序挖掘中代理系统的设计

为了实现程序挖掘这个分布式的应用系统,我们在代理平台 Ki[4]的基础上开发了一个移动代理平台 ARE(Agent Running Environment),通过引入 JAS[5]包,使其具有符合代理标准 FIPA(Foundation of Intelligent Physical Agents)[6]的代理管理机制,目录服务机制和信息传递机制. 同时,为了更方便地处理 XML 格式的用户需求与构件描述,ARE 中代理采用的通信语言是基于 XML 的 ACML(Agent Communication Markup Language)[7]. 基于移动代理平台 ARE,我们为程序挖掘设计了代理系统. 如图(2)所示,程序挖掘系统中涉及的实体对象有:客户端,程序挖掘服务器,构件库和构件目录库. 智能接口子系统在客户端中为用户提供查询导航. 程序挖掘服务器是核心,包括了需求分析与构件获取子系统,知识库和构件组装与代码生成子系统. 目录库和构件库组成了构件目录与构件资源子系统.

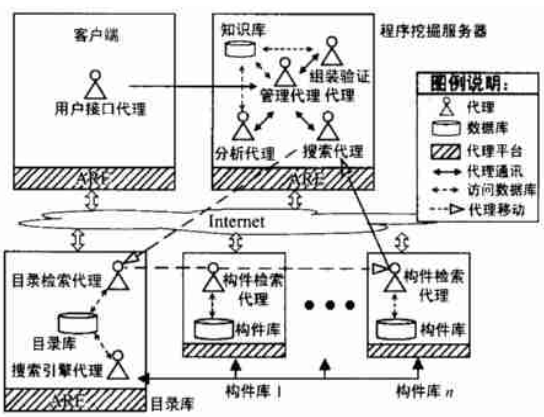

图 2 程序挖掘中代理交互图

根据每个实体的功能可以设计相应的代理,如图(2)程序挖掘中的代理交互图所示:

对于客户端,有一个接口代理. 它实现与用户的友好交互,并向程序挖掘服务器提交用户请求,接收反馈.

程序挖掘服务器中的代理呈星形拓扑结构. 中心是一个任务管理代理. 它和客户端交互,然后把用户的需求交由分析

代理.分析代理根据知识库理解了用户的需求以后,得到组装方案.接着任务管理代理把这个方案交给搜索代理处理.搜索代理根据组装方案一个个获取构件实体.然后任务管理代理就把组装方案和相应构件交由组装验证代理进行组装并验证.如果成功,就得到用户需要的程序,并把它反馈给用户.其中搜索代理可以在目录库、构件库之间移动,通过与它们的库代理交互完成对所需构件的搜索,并返回服务器.

构件目录库有目录检索代理,提供对目录库的查询服务;构件库有构件检索代理,提供对构件库的查询服务.搜索代理与它们交互以获取构件信息.为了使构件目录库的信息及时更新,和构件库保持一致,在目录库中有搜索引擎代理.它专门负责搜索网上的构件库,并获取新的构件信息,将其转换为统一的构件描述存放到目录库中.

3.2　构件资源的组织

对于构件资源,我们认为往往各种不同结构的构件库在 Internet 上广泛分布.要对他们统一检索,首先要对他们进行统一描述.我们提出了一种特殊的构件库 - 构件目录库,针对分布式异构构件库提供构件信息的统一描述和索引.文献[8]介绍了描述构件的各种方法.构件目录库是一种复合型的构件库应该采用多种构件描述.但是不是每个构件都能用各种方法来描述的;通常相同领域的构件往往适用于同一种描述方法.结合两者,我们用目录树结构对构件库实现领域的划分;对每个领域根据其特性采用了相应的描述方法,如数学函数领域用形式化描述,标准化构件用刻面描述.当然每个领域的描述方法也可以多种并存.

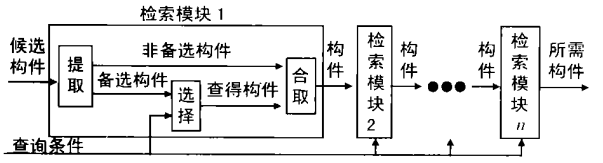

图 3　综合构件检索模型

文献[9]介绍了多种构件检索方法.而我们的构件目录库则采用了一种综合的检索模型.如图(3)是由 n 个构件检索模块串联组成的检索系统.其中检索模块 1 给出了其内部结构:首先对构件库中的候选构件进行提取,对适合本检索模块算法的备选构件提取有用信息,在选择子模块中根据查询条件进行筛选,获得"查得构件",并和非备选构件合并作为下一个检索模块的候选构件.这样根据 n 个检索模块的筛选得到了所需的构件.有多少,哪几个检索模块组合可以由用户指定.而且每个检索模块的合取子模块默认是将"非备选构件"和"查得构件"合并,也可以设置为只取其中一个.

3.3　构件组装与代码生成

程序挖掘原型系统中,我们对 JavaBean 构件实现了自动组装和代码生成.构件组装代理在构件容器中将搜索得到的 JavaBean 构件根据解决方案组装成需要的 Java 程序.构件容器采用 Sun BDK 中的 Beanbox,扩充了 XML 文件处理、构件分类显示和代码生成等功能.在获取需要的 JavaBean 后,构件组装代理读取解决方案中的构件组装文件,分析文件中构件间的事件触发、方法调用以及属性绑定等连接关系,自动生成事件适配、参数传递、静态入口等代码,最终形成可执行的 Java 程序.对于可视化构件,还需要根据构件的位置、大小等信息,生成版面设置代码.所有代码生成完毕后,构件组装代理调用系统的 Java 编译器将源文件编译成目标码,并生成可调用 Java 程序的批处理文件.

3.4　综合领域知识的程序挖掘算法

程序挖掘的目标是要实现按需计算,即根据用户需求自动组装构件,生成满足要求的程序.软件复用的前提是领域知识的完备.我们假设领域知识完备且存放在知识库中,而且每个领域所需的可复用构件也已存放在构件库中.

在原型系统中,我们采用 FODA(Feature-Oriented Domain Analysis)[10]对各个领域进行分析建模.用 FODA 对领域进行分析一般有三个步骤:上下文分析,领域建模,体系结构建模.上下文分析要确定这个领域所要解决问题的范围,以及和其他领域的关系.领域建模则要阐明领域中各个实体之间的关系、实现的功能、术语空间.体系结构建模的内容包括模块结构图,进程交互模型.在原型系统中,这些领域分析后得到的结果,称之为领域知识,存放在知识库中.这样有了各个领域所解决问题的范围,用户就可以确定自己需要的计算问题属于哪个领域;进而根据领域建模的结果,可以选择所需的计算功能集合;相应的就能找到包含这些功能集合的软件体系结构,然后进行裁剪定制就得到满足用户计算需求的组装方案.如图(4)所示是综合领域知识的程序挖掘算法流程.

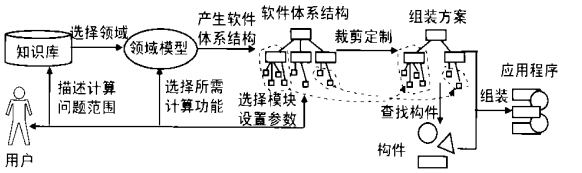

图 4　综合领域知识的程序挖掘算法流程

(1)用户描述所需计算的问题范围,和知识库交互,选定领域模型;

(2)在领域模型的导航下,用户选择所需的功能集合,确定相应的软件体系结构.软件体系结构是作为领域知识的一部分预先存放于知识库的;

(3)对该软件体系结构,用户可以按照需要进行裁剪,选择模块,设置参数,形成满足用户要求的组装方案;

(4)程序挖掘服务器根据组装方案从构件库中查找获取所需构件;

(5)最后根据组装方案,将构件组装成满足用户计算要求的应用程序.

4　结论

本文针对分布式网络中的按需计算问题,提出了一种新的计算模式——程序挖掘,给出了基于代理的程序挖掘实现框架,并以代理运行环境 ARE 为基础,实现了对 JavaBean 构件动态组装网络应用的原型系统,证明了程序挖掘的可行性.

但是,要在 Internet 这样的大规模分布式网络中真正实现按需计算,仍然有很多问题需要研究:(1)如何有效的刻画构

件的动态行为?构件的动态行为描述直接影响到组装程序的功能和执行效率.因此,需要一种能全面描述构件的静态属性和动态行为的描述方法.文献[11]中的工作在这一方面作了有益的尝试:通过对 UML 的扩展建立了构件的动态生命周期模型;(2)构件组装问题.虽然 CORBA,COM/DCOM,JavaBean/EJB 等构件标准为构件组装提供了很好的技术支持,但构件组装问题现在还是一个开放的领域.要实现大规模的构件自动、半自动组装还有待进一步研究;(3)程序挖掘算法.在程序挖掘过程中,需要全面分析构件执行中的顺序控制和动态依赖关系,同时要分析和处理构件间的数据与控制流向.由于构件之间的调用关系在运行过程中是动态变化的,因此需要建立一个良好的挖掘模型,并设计专门的算法对所有可能的构件组合进行逻辑关系分析,在程序的功能规约和构件的接口规约之间建立联系.以机器学习理论为基础,设计并实现高效可用的程序挖掘算法是有待研究的重要课题.

参考文献:

[1] D Tennenhouse, D Wetherall. Towards an active network architecture [J]. Computer Communication Review, 1996, 26(2).

[2] ZZ Wei, YX Zhang, X Li. A new computing raradigm: Program mining [A]. FTC2001 [C]. Beijing, 2001.

[3] H Mili, F Mili, A Mili. Reusing software: issues and research directions [J]. IEEE Transactions on Software Engineering, 1995, 21(6): 528 - 562.

[4] Welcome to Ki for JavaWorld [DB/OL]. http://diana.cps.unizar.es/banares/IC/work/ki/, 1998 - 10 - 16.

[5] Java Agent Services-An Introduction [DB/OL]. http://www.java-agent.org/JAS_Intro.htm, 2000.

[6] FIPA Agent Management Specification [DB/OL]. http://www.fipa.org/specs/fipa00023/, 2000.

[7] B Grosof, Y Labrou. An approach to using XML and a rule-based content language with an agent communication language [A]. IJCAI-99 [C]. Stockholm, Sweden, 1999.

[8] R Mili, A Mili, R T Mittermeir. Storing and retrieving software components: A refinement based system [J]. IEEE Transactions on Software Engineering, 1997, 23(7): 445 - 460.

[9] S Atkinson. A formal model for integrated retrieval from software libraries [A]. TOOLS'96 [C]. Prentice Hall, 1996.

[10] Kyo Kang, Sholom Cohen, et al. Feature-Oriented Domain Analysis (FODA) Feasibility Study [R]. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, PA, 1990.

[11] A Wienberg, F Matthes, M Boger. Modeling dynamic software components in UML [A]. UML'99 [C]. Lecture Notes in Computer Science, Springer-Verlag, 1999, 1723: 204 - 219.

作者简介:

夏东林 男,1978 年 7 月生于浙江绍兴,清华大学计算机系硕士,主要研究方向为智能代理系统,软件复用和程序挖掘.

张尧学 男,1956 年 1 月生,工学博士,清华大学计算机系教授,博士生导师,主要研究方向为计算机网络,包括网络路由器、网络协议工程、服务质量控制方法与网络操作系统等,以及程序挖掘.