基于 Mobile Agent 技术的遗留系统再工程方法*

詹剑锋, 程 虎

(中国科学院 软件研究所,北京 100080)
E-mail: jfzhan@ncic.ac.cn; chenghu@163.com
http://www.iscas.ac.cn

摘要: 针对因特网环境下遗留系统需要不断引入新的需求和技术的需要,提出了一种基于 Mobile Agent 的遗留系统再工程方法.在该方法中,对遗留系统采用 Agent 思想重新建模,将频繁的需要与其他部分交互的组件实现为移动 Agent,并且允许以定制的 Agent 的方式添加新的需求,使得目标系统能够更加适应因特网环境.通过将一个单机环境下的单用户计算软件迁移到网络环境下,允许多个远程用户并发访问的再工程尝试,为遗留系统引入新的需求和技术提供了一条可以选择的途径.

关 键 词: 遗留系统;再工程;agent 技术;mobile agent;迁移;软件体系结构
中图法分类号: TP393 文献标识码: A

在石油、机械、化工、邮电等领域广泛使用着一类软件,它们作为知识积累的载体,都包含有效的商业事务处理模型和商业规则,在算法上无可挑剔.但在其他形态上,却或多或少具有一些遗留系统的弱点(legacy system),如运行在缓慢而难以维护的硬件上;软件维护费用昂贵,文档的缺乏和系统细节的困难导致纠错成本高且耗费时间;系统缺乏清晰的界面,与其他系统的集成非常困难;软件体系结构趋于单一,不具有可扩展性;运行在单机环境下,缺乏网络获取特性和网络发布界面,不易实现资源共享.

对于这些情况,长期以来人们已经开始形成了一套软件再工程的经验和方法[1~4],如在逆向工程(reverse engineering)的基础上,理解软件本身,形成软件的智力模型(mental model),然后再予以重构造(restructuring)或引入新技术、新需求和新规范予以重实现,或者利用新的技术将系统从旧的环境中迁移到新的平台.而软件组件技术也给软件再工程注入了新的内容[5,6],如在系统理解的基础上,有效地识别程序的组件,遵循有效的软件界面规范,如 CORBA 或 COM/DCOM,开发对应于新的需求的组件,再装配新的系统.组件技术显然对软件再工程起了很大的促进作用[5],再工程师们开始"有章可循",以往无序、混乱而繁琐的工作开始有了清晰的路标:先识别有效的系统模型,再按照模型划分合理的组件,高效地构造包装器,添加旧系统到新组件规范的变换代码,在这些工作的基础上人们就可以开始装配符合市场需求,并能根据市场需求动态变化新的系统.

尽管在这一类系统的集成的工程化上,人们已经做了较多的工作,但在本文中,我们将讨论上述提及的方法在灵活性、动态性、可靠性上无法适应把遗留系统迁移到网络平台上的需求变化.本文第 1 节和第 2 节首先给出再工程的需求分析,然后在满足这些需求的情况下提出了基于 Mobile Agent 的系统迁移方法.第 3 节分析了一个具体案例,通过将单机环境下的单任务计算软件再工程为能够适应网络平台并能满足多个远程用户并发访问的分布式系统,验证了方法的可行性.第 4 节是结论,分析了本文提出的方法的特点,并指出了以往基于 CORBA 的再工程方法的一些弱点.

* 收稿日期: 2001-01-09; 修改日期: 2001-07-13

作者简介: 詹剑锋(1976－),男,安徽太湖人,博士,助理研究员,主要研究领域为软件体系结构,多 agent 系统,分布式构件;程虎(1938－),男,江苏常州人,研究员,博士生导师,主要研究领域为语言编译,软件工程,人工智能,神经网络.

1 再工程的需求分析

对于以遗留系统形式存在的专业领域软件,倘若在相当长的时间内保持其经济/成本率在较高的水平,必须满足以下条件:

(1) 能够适应新组织操作的因特网环境,系统应该具有健壮性、智能性、灵活性、动态性以及资源的易访问性.

(2) 在应用程序之间鼓励核心功能的复用,封装商业组织最好的实践,这样在商业条件改变时能够更快地适应.

(3) 能迅速地在正在演变的系统中引入新的需求,系统具有较好的可演变性,如果遗留系统本身僵化,无法迅速地适应系统需求的改变和调整,则系统的价值会迅速下降.

(4) 在软件系统发布和应用方面,重视新技术提供的机遇.能迅速利用新的主流技术,如果不能采用,则系统的实现技术本身会迅速边缘化,从而需要专门的研究和培训,会增加成本.

(5) 系统体系结构的动态性和灵活性.只有实现系统组成运行时(run-time)的动态和灵活,才有可能提高资源的利用率以及系统的稳健性,如引入网络位置的可变性,使组件可以在不同的位置移动,一方面可以实现资源访问的局部性,另外一方面又可以增强系统的稳健性,如组件从负载较高的主机迁移到负载较低的主机,或从即将崩溃的主机迁移,都会提高系统的可靠性.

(6) 迁移后的遗留系统必须具有资源的易访问性,远程用户可以通过某种软件的形式实现资源的访问.这种软件既像浏览器软件那样具有简单、无处不在的特性,还应该具有智能特性和交互特性,能够实现与用户、软件环境、Agent 或非 Agent 之间的交互和合作.

2 基于 Mobile Agent 的遗留系统迁移方法

正是根据以上需求,我们提出了基于 Mobile Agent 的动态集成的再工程方法.

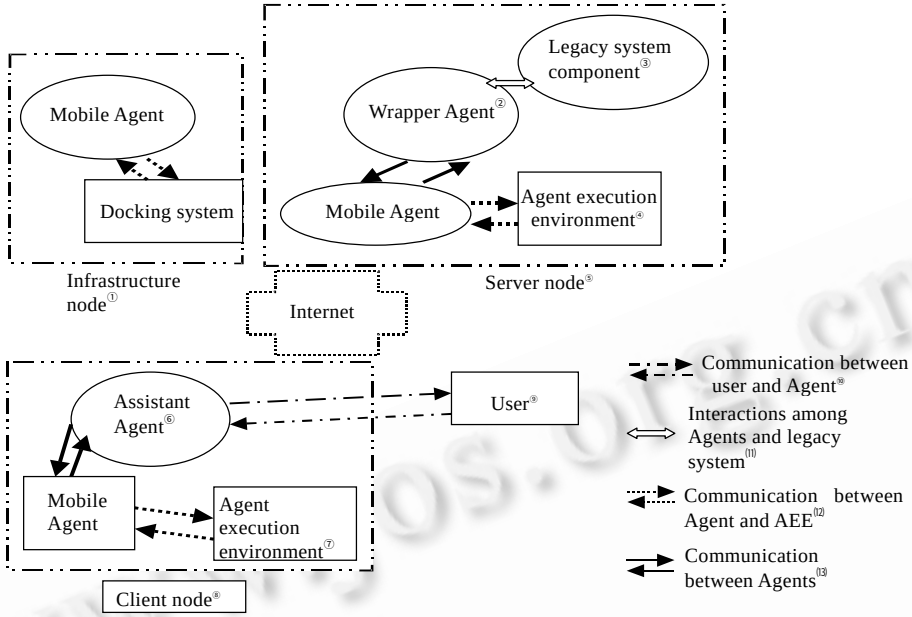
Mobile Agent 可以定义为在计算机环境里代表用户的计算实体,它能为了满足某个目标自治地从一个结点移动到另一个结点,并代表用户完成某种类型计算.Mobile Agent 具有自治性、交互性(与用户、计算机环境、非 Agent 程序实体、其他 Agent 交互)、移动性以及理性(满足某一特定目标)[7].

根据文献[8]的划分, Agent 可以根据 Agent 之间以及 Agent 与环境的主要交互模式分为 3 类:自治 Agent、多 Agent 和助理 Agent(assistant Agent).自治 Agent 主要强调 Agent 与环境的交互,而多 Agent 的环境不仅包括非 Agent,还包括其他 Agent.助理 Agent 主要强调与特别类型的 Agent 交互或代表它们,如用户.

在基于 Mobile Agent 的动态集成的再工程方案中,将在目标系统中采用 Agent 思想建模.这是因为在因特网环境下,要求遗留系统以灵活、弹性、智能、动态的方式实现与新的系统合并与集成.而 Agent 允许遗留系统(非 Agent)以比较直观的方式合并.只要有新的需求,定制的 Agent 就可以开发并且添加.这种特性允许复杂系统以演变的方式增长.需要指出的是,本文只涉及到单机环境下体系结构单一的系统(monolithic system)的迁移问题.

在客户端需要建立与用户交互的客户端,在目标系统上以助理 Agent 形式存在.在遗留系统中,经常与用户交互的部分极为简单或不存在,这就需要创建新的组件或增强原来的功能.由于 Agent 之间的合作与通信是通过 Agent 通信语言(Agent communication language)实现的,而对遗留系统的主要部分仍采用包装的方案,这样就需要添加一个支持 ACL 并能把 Agent 通信语言映射为遗留系统消息的包装器 Agent.而在因特网环境下,为了减少对网络带宽性能瓶颈的依赖,将系统中需要与其他部分频繁交互的那部分实现为 Mobile Agent,这样 Mobile Agent 理性地移动到需要与之通信的组件的本地系统上(通过引入资源的局部访问特性,能够减少对网络带宽的浪费),这里的理性移动要求 Mobile Agent 具有自治性和智能性.而为了增强系统的可靠性,也可以将系统的关键组件实现为 Mobile Agent,这样就可以通过提高系统在灾难条件下的生存能力而提高系统的可靠性.对于 Mobile Agent,不仅需要在客户端和服务端提供创建、迁移 Mobile Agent 以及调用系统资源的 Agent 执行环境(AEE),还需要在一些公共结点为 Mobile Agent 提供临时的驻点:Docking System.这样,在需要调整结点负

载平衡的时候,在网络阻塞或系统崩溃时,都可以提高系统的健壮性和可靠性,如图 1 所示.



Note: in this paper, all dotted line boxes represent the same node⑭.

①公共结点,②包装器 Agent,③遗留系统组件,④Agent 执行环境(AEE),⑤服务器结点,⑥助理 Agent,⑦Agent 执行环境,⑧客户机结点,⑨用户,⑩用户与 Agent 之间的通信,⑪Agent 与遗留系统之间的交互,⑫Agent 与 AEE 之间的通信,⑬Agent 之间的通信,⑭注:在本文中所有的虚线框表示同一结点环境.

Fig.1 Software architecture of the migrated system

图 1 迁移后的目标系统的体系结构

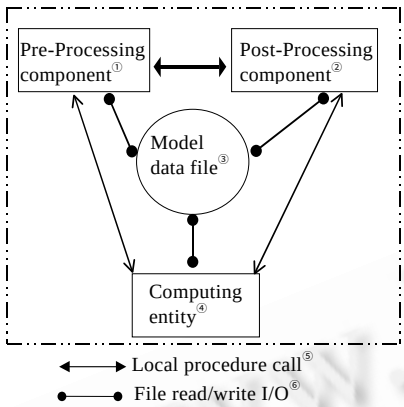
在本文叙述的方法中,系统的配置不再固定,Mobile Agent 会根据组件之间通信的需要,以及网络的动态环境和主机的结点负载情况作出理性的决策.而系统组件的构成和数量也会发生动态变化,如系统某个任务已经完成,被分配该任务的 Mobile Agent 可以被销毁,而当某个任务需要涉及到多个遗留系统资源时,通过向 AEE 执行环境请求克隆(clone)多个 Mobile Agent,可以并发地完成相应的任务,从而提高系统的效率,这些 Agent 之间本身也可以合作和协调完成某项任务.通过在目标系统中采纳 Agent 的计算模型,对遗留系统采用以界面理解为中心的包装方案,通过引入包装器 Agent 实现遗留系统与目标系统的集成与交互,并引入 Mobile Agent 的可移动性和自治性,单一整体的遗留系统能够在新的环境下满足第 1 节中的 6 个条件.

3 案例分析

在本文中,以一个单机环境下单任务有限元软件为例,应用上节中描述的基本方法,将其迁移到因特网环境下,在实现远程访问的同时,还支持多个用户的并发访问.首先,从软件体系结构角度阐述.软件体系结构可以定义为组件(component)与组件之间的相互关系.从软件体系结构着手研究遗留系统的再工程,可以从宏观上进行系统结构形式的改变,从而使系统更能够适应不断演变的环境.

本文研究的系统是一个中等规模大约 2 万行源代码的用 C 语言编写的名为 Cppfem 的有限元计算包[6].从软件体系结构来说,系统主要可以划分为前处理组件、后处理组件以及计算实体和存储计算模型数据的文件等 4 个组件构成,连接体(connector)则主要为本地过程调用和文件读写 I/O,如图 2 所示.为了在网络环境下实现资源共享,在文献[6]中,通过使用 Java 技术和 CORBA 技术,已经将系统迁移为一个可以实现远程访问的系统,如图 3 所示.在这种体系结构中,为了改善资源的易访问性,在实现上将前处理和后处理组件合并,并用 Java 重新实现,从而获得平台无关性.而在遗留系统的主要组件部分,使用 CORBA IDL 进行包装.为了增强遗留系统服务端的能力,添加了一个用 Java 实现的 Wrapper 组件,负责与客户端通信,并把客户端的消息格式映射为遗留系统的消

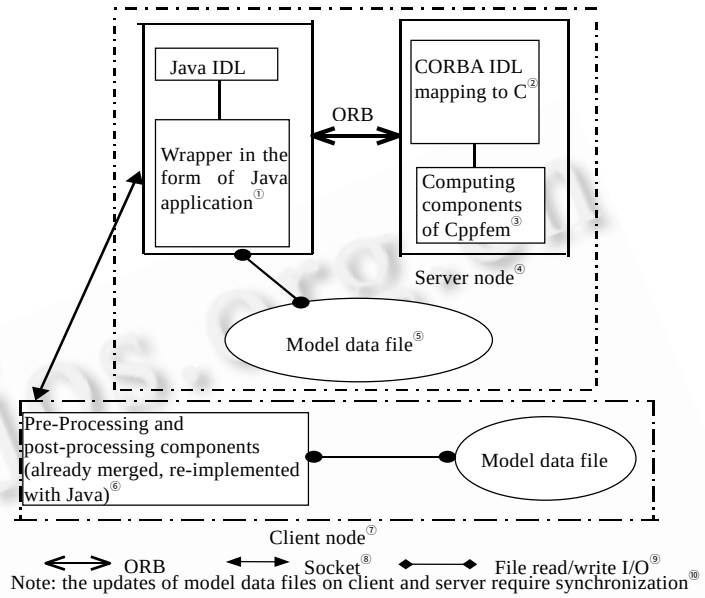
息格式.需要指出的是,在该系统的迁移中,采用的是一种混合的连接体构造.ORB 通信仅在服务器本地环境下使用,而客户机和服务器的通信通过原始的 Socket 构造.为了便于满足对用户的快速响应,在客户机端和服务器端存在需要同步更新的两个模型数据文件,这样当用户选择某种后处理时,不必重新从服务器结点获取模型数据.



①前处理组件,②后处理组件,③模型数据文件,④计算实体,⑤本地过程调用,⑥文件读写 I/O.

Fig.2 Software architecture of Cppfem

图 2 Cppfem 的体系结构



①以 Java application 形式存在的 Wrapper,②CORBA IDL 向 C 的映射,③Cppfem 的计算实体组件,④服务结点,⑤模型数据文件,⑥前处理组件和后处理组件(已经合并,用 Java 重新实现),⑦客户机结点,⑧套接字,⑨文件读写 I/O,⑩注:客户机和服务器结点上的模型数据文件需要同步更新.

Fig.3 Software architecture of migrated Cppfem

图 3 迁移后的 Cppfem 的体系结构

在该系统使用后,发现存在严重的问题.首先,系统的稳健运行需要长时间的可靠和实时的网络连接,而在因特网的环境下,无法提供这种可靠、安全和高效的网络服务,通过重新实现专门的通信协议不足以根本解决该问题.而且,系统严重依赖于数据模型文件,而数据模型文件的同步更新将依赖于网络的通信,这更加剧了对网络带宽的需求.而正如前面所指出的,将数据模型文件以及相关的用户指令封装为 Mobile Agent 将会根本地解决问题.如图 4(a)和图 4(b)所示.以 Agent 思想重新为系统建模.前处理和后处理组件因为其与用户交互的固有属性,自然应该被实现为 Assistant Agent,而数据模型文件以及相关指令被封装为 Mobile Agent.遗留系统计算实体组件采取图 3 中同样的处理方法.Wrapper Agent 作为计算实体组件与系统其他组件交互的界面,将 Agent 通信语言转换为遗留系统消息格式.调度 Agent 负责调度多个用户以 Mobile Agent 形式加载的计算任务.在服务结点和计算结点都提供了 Mobile Agent 赖以存在的 Agent 执行环境(Agent execution environment),它同时为 Mobile Agent 的创建和迁移以及定位等提供基本服务.在公共结点上,提供了 Docking System,当服务器繁忙时,或请求计算的客户结点出现故障或网络堵塞时,Mobile Agent 就都可以作暂时驻留.

这样,一次计算请求的服务过程可以描述如下:

(1) 用户通过与 Assistant Agent 交互,完成计算模型.

(2) Assistant Agent 通过与 AEE 通信,由 AEE 创建一个封装了模型数据文件以及用户计算要求,如大致完成时间、计算精度、计算内容的 Mobile Agent.

(3) AEE 将代表该用户计算任务的 Mobile Agent 迁移至服务器结点.

(4) 调度(dispatch) Agent 与 Mobile Agent 通信,根据用户计算要求和系统资源使用状况调度用户任务队列.

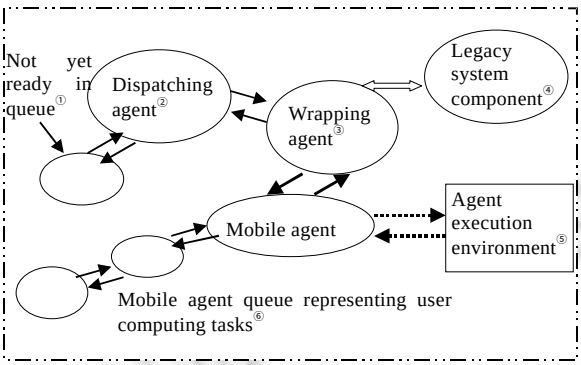
(5) 代表用户计算任务的 Mobile Agent 根据调度的优先次序以及服务器结点的负载情况和用户计算需求作出相应决定,如是否迁移至相近的公共结点 Docking System.

(6) Mobile Agent 与 Wrapper Agent 通信,完成计算后,封装计算结果.

(7) Mobile Agent 返回客户结点,与 Assistant Agent 通信,以完成用户所需要的后处理任务.

(8) Mobile Agent 被销毁.

采用 IBM 的 AGLET[9]在局域网环境下实现了系统原型,初步验证了方法的可行性.在已实现的 Mobile Agent 系统中,仅具有基本的迁移、命名和查找以及定位功能.在 Agent 通信语言方面,选择了 XML[10]作为结构化的可扩展标记语言,XML 具有简单、易读性、平台无关性以及无处不在的优点.迁移后的系统可基本满足多个用户远程并发访问的需求.

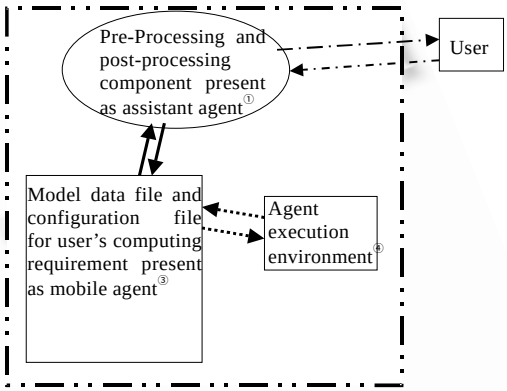


Note: arrows in this figure has same meanings as figure 1⑦

①尚未在队列就绪,②调度 Agent,③包装器 Agent,④遗留系统组件,⑤Agent 执行环境,⑥代表用户计算任务的 Mobile Agent 队列,⑦注:箭头表示的意义与图 1 相同.

(a) Software architecture of server node

(a) 服务器结点上的组件的体系结构



Note: arrows in this figure has same meanings as figure 1⑤

①以 Assistant Agent 形式存在的前处理和后处理组件,②用户,③以 Mobile Agent 形式存在的模型数据文件和用户计算需求配置文件,④Agent 执行环境⑤注:图中的箭头表示的意义与图 1 相同.

(b) Software architecture of client

(b) 客户机结点上的组件的体系结构

Fig.4

图 4

4 结 论

本文提出了一种基于 Mobile Agent 动态集成的再工程方法.它主要具有如下特点:

(1) 对遗留系统采用以界面理解为中心的包装方案,通过引入包装器 Agent 实现遗留系统与目标系统的集成与交互.

(2) 通过将频繁的需要与其他组件通信的部分实现为移动 Agent,并通过迁移 Mobile Agent 主动接近需要交互的组件来避免对网络带宽的依赖,从而使目标系统更加适应网络平台的需求.

(3) 以 Agent 形式存在的组件之间通过 Agent 通信语言交互,避免了运行之前的装配(assembly before runtime),从而提高了系统的弹性和动态性.

而在以往的工作中[5~7],基于 CORBA 的系统集成缺乏灵活性和动态性.系统的配置无法在运行时动态改变,也就无法适应系统资源、系统性能的动态变化.在某一组件负载过重或在主机出现故障的情况下,系统都将崩溃而无法挽回.另外,鉴于系统组件越来越向异构性、地域的分布性等方向发展,仅仅依赖 CORBA 的 ORB 实现组件之间的通信必然形成新的性能瓶颈(当两个组件依赖网络带宽通信时).最后,CORBA 的集成机制缺乏支持组件之间有效合作的机制,从而无法胜任涉及到的复杂协同工作的系统集成.而本文提出的方法较好地避免了这些问题.

由于仅仅在局域网环境下实现系统原型,并且被迁移的系统规模有限,因此本文提出的方法需要在进一步的工作中再加以完善.

References:

[1] Paul, P. Integrating legacy systems with modern corporate applications. Communications of the ACM, 1997,40(5):39~46.

[2] Noffsinger, W.B., Niedbalksi, R., Blanks, M. Legacy object modeling speeds software integration. Communications of the ACM, 1998,41(12):80~89.

[3] Harry, M.S. Planning the reengineering of legacy systems. IEEE Software, 1995,12(1):24~33.

[4] Frank, P.C. Legacy integration-changing perspective. IEEE Software, 2000,17(2):33~41.

[5] Nelson, H.W., Linda, N. Implication of distributed object technology for reengineering. Technical Report, CMU/SEI-97-TR-005, Software Engineering Institute, Carnegie Mellon University, 1998.

[6] Zhan, Jian-feng, Kong, Xiang-an. The integration and dissemination of domain-specific software on the internet. Journal of System Simulation, 2001,13(1):22~24 (in Chinese).

[7] Neeran, M.K., Anand, R.T. Design issues in mobile agent programming systems. IEEE Concurrency, 1998,6(3):52~61.

[8] Nicholas, R.J. On agent-based software engineering. Artificial Intelligence, 2000,117(2):277~296.

[9] Wong, D., Paciorek, N., Moore, D., *et al*. Java-Based mobile agents. Communications of the ACM, 1999,42(3):92~105.

[10] Roy, J., Ramanujan, A. XML: data's universal language. IT Professional, 2000,2(3):17~23.

附中文参考文献:

[6] 詹剑锋,孔祥安,专业领域软件在因特网上的集成与发布.系统仿真学报,2001,13(1):22~24.

A Solution to Reengineering the Legacy System Based on Mobile Agent Technology*

ZHAN Jian-feng, CHENG Hu

(*Institute of Software, The Chinese Academy of Sciences, Beijing* 100080, *China*)

E-mail: jfzhan@ncic.ac.cn; chenghu@163.com

http://www.iscas.ac.cn

Abstract: In this paper, a solution to reengineering the legacy system based on mobile agent technology is proposed to satisfy the demand of continuous introducing new requirements and techniques into legacy system. An agent viewpoint is adopted to model the legacy system, with the components frequently interacting with others implemented as mobile agent and new requirements added as customized agents. Through the practice of migrating a standalone single-user computing software onto networking platform with the support of several remote user's concurrent accesses, an alternative way of introducing new requirements and technologies into legacy system is proposed.

Key words: legacy system; reengineering; agent technique; mobile agent; migration; software architecture

* Received January 9, 2001; accepted July 13, 2001