

遗产软件重构技术的研究

刘芳,瞿有甜,周波,胡顺扬

(浙江师范大学数理信息学院,浙江金华 321004)

摘要:随着计算机技术的不断发展,遗产软件在软件生产过程中起着越来越重要的作用。如何积极地利用好遗产软件已经成为目前软件工程中讨论的一个热门话题,国内外科研人员对重构遗产软件已经做了大量的研究并且取得了不少研究成果。文中是一篇关于遗产软件重构技术研究的综述性文章。介绍了一些遗产软件系统相关的概念,从整体上对遗产软件重构方法进行了分类,阐述并分析了近年来国内外科研人员就遗产软件重构提出的一些思想、方法及模型,对遗产软件重构技术做了一个总结,并提出了组合 Agent-Component(AC)来重构分布异构环境下遗产软件的思想,为解决遗产软件的重构尤其是大型复杂系统的重开发问题提供了一个新的思路和途径。

关键词:重构;遗产软件系统;软件工程

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2009)03-0118-05

Research of Reconstruction of Legacy Software System

LIU Fang, QU You-tian, ZHOU Bo, HU Shun-yang

(College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China)

Abstract: With the development of computer technology, the legacy software systems play a more and more important role in software production today. At the same time, how to take the advantages of legacy software systems effectively has become a hot discussion topic in current software engineering and a lot of researchers inside and outside the country have done a number of researches on reconstructing legacy software system. This paper is an overview on reconstructing the legacy software system. Introduces some relative concepts about legacy software system first, then describes the classification of current reconstruction approaches used in legacy software system and analyses these approaches in details. The end of the paper is conclusions on reconstruction of legacy software system and the concept of AC is come up to reconstruct distributed and heterogeneous legacy software systems. It provides a new concept and approach in reconstructing legacy software, especially for those large scale and complex systems.

Key words: reconstruction; legacy software system; software engineering

1 背景

遗产软件系统是一个较新的概念,还没有一个严格统一的定义,一个比较正式的定义是这样描述的:随着计算机技术的广泛使用,出现了一些难以维护和进化的大规模的复杂软件系统,称之为遗产软件系统^[1]。这些系统一般经历了几十年的应用积累,同时这些系统的规模越来越大、功能越来越复杂,它们积聚了大量的领域知识,包括需求、设计、业务规则、历史数据等等,因而这些系统有着巨大的重用价值^[2]。

软件重构是实现软件重用的一种基本方法,软件

重构或重用已成为计算机科学的普遍问题和基本目标。在软件项目的开发过程中,用于改善和维护系统的人力和时间所占的比例越来越大,利用已有的软件系统服务于新的业务需求,已经成为软件工程的另一个重要发展方向。在此基础上,遗产软件重构的概念开始出现,并且其在软件生产过程中充当越来越重要的角色。

新系统的开发可以完全抛弃遗产软件,在零的基础上进行重新开发。这样无论在费用、开发周期、开发风险上,都会面临着很大的挑战。如果利用遗产软件资源对软件进行重用,则可以明显地减少费用、缩短开发周期、降低开发风险。挖掘遗产软件中的可用价值,并把先进的信息技术融合进遗产软件中,是对待遗产软件系统的积极方式。通过此种方式可以使遗产软件系统具备更高效的操作方式、系统能力、新的功能,并

收稿日期:2008-06-12

基金项目:浙江省自然科学基金项目(Y106469)

作者简介:刘芳(1982-),男,硕士研究生,研究领域为软件工程、构件技术;瞿有甜,教授,研究领域为软件工程、构件技术和数据库技术。

且是以较低的成本、较短的时间、较低的用户风险完成的。

2 方法分类

目前对待遗产软件系统的方法基本可分为再开发(Redevelopment)、打包(Wrapping)和移植(Migration)三大类。

(1)再开发方法主要可以分为重新开发和反向工程两类。重新开发是解决遗产软件问题的一种基本方法,这种方法抛弃了原有系统的代码,从零开始重新开发整个系统。反向工程又可以分为黑盒方法和白盒方法^[3]。黑盒方法一般是通过打包遗产软件及相关环境为一个软件层来隐藏遗产软件复杂的具体实现过程,同时提供了一个现代化的接口。在白盒方法中,是通过对遗产系统的深入分析来获取业务逻辑。白盒方法一般包括基于数据库的反向工程和基于程序的反向工程两种方法。基于数据库的反向工程方法相对比较成熟,可以借助一些辅助工具来较好地实现。基于程序的反向工程方法要求深入分析并理解遗产系统的体系架构,使得新的体系架构能够满足新的需求并消除遗产软件的一些缺陷。

(2)根据打包的内容,打包方法一般可以分为基于用户界面打包、基于数据打包和基于功能打包三种方法^[4]。基于用户界面的打包方法是根据遗产系统中的用户界面映射为新系统的用户界面。基于数据的打包方法是在遗产系统的数据结构的基础上来开发的,从而使得原有的数据可以应用到新的系统当中来,基于数据的打包又可分为数据库网关方法、XML集成方法和数据复制方法等等。基于功能的打包方法在利用原有数据的同时还考虑了遗产系统中业务逻辑。基于功能的打包还可细分为组件打包方法、面向对象打包方法和公共网关接口方法等等。

(3)移植方法基本上可以分成基于组件的移植方法和基于系统的移植方法。前者是先把大型的遗产系统分解成独立的组件然后对组件进行个别地移植。在整个过程中必然有遗产系统与新的系统并存的一个过渡时期,对待这个问题一般有两种策略,即协同操作和并行运行。当然这两种策略都需要数据的共享,可以通过数据库网关、复制或者将独立的域划分成小的片段来逐渐移植等方式来实现。而后者则是把整个遗产系统及数据的移植到新的系统作为一个步骤。基于系统的移植可以分为无增值的移植和有增值的移植两类,两者的本质差别在于是否存在用户界面、数据库、程序代码等方面的改进。

计算机技术的发展为解决遗产软件系统的重构问

题提供了各种不同的方式以及大量的方法,遗产系统重构的现状可以归纳如下:

①再开发方法通常被大多数组织机构认为是风险最大的。

②大型遗产系统的基于过程组件的反向工程仍未得到解决。

③基于数据库的反向工程方法已经相当成熟,并且已有不少成功的实践应用。

④打包方法仅是暂时解决了当前问题,并且加重了系统维护和管理难度。

⑤成功的遗产系统重构方法还很少,很多重构工程都不是很成功。

3 研究现状

近年来,人们越来越认识到遗产软件重构在实际软件开发过程中的重要作用,国内外很多专家和研究人员对遗产软件的重构进行了大量的研究,取得了不少研究成果,提出了很多遗产软件重构方法及模型。

3.1 再开发(Redevelopment)

CORUM^[5]方法的基本思想是用统一标准化的方式来实现再工程信息的交互并最终创建一个齐全的并且能够共同使用的再工程工具集。然而这个方法主要关注的是在代码层的再工程,注重的是抽象语法树、控制流程图、数据流程图的创建及处理,而忽略了软件体系结构以及支持体系结构的工具集等方面的考虑,导致了体系结构标准化程度低、开放性差等。

卡耐基梅隆大学软件工程研究所在CORUM方法的基础上提出了CORUM II^[6]模型,这是一个基于体系结构的再工程过程和前向工程过程的集成模型,弥补了CORUM方法在体系结构及其相关方面的考虑不足,他们将这个模型主要分为代码和体系结构的恢复以及一致性评估、体系结构的转换、基于新体系结构的扩充,包括高层的详细设计等三个步骤,但是该方法并没有提供详细的工作过程及过程执行的细节。

MaRMI-RE方法^[7,8]是韩国电子与通信研究院在CORUM II模型的基础上建立的,重点考虑了在分布式环境下遗产软件体系结构的标准化及开放性的重要性,他们提出通过规划、逆向工程、构件化及部署四个阶段来完成其遗产系统重构的工作。使其能够在当前的技术环境下,更好地适应层出不穷的业务要求和需求的不断变化,同时实现体系结构的清晰明了及系统较好的可重用性。

OSET^[9]是从遗产系统接口的角度提出了一种对象结构提取的方法。这个方法可以通过接口使用情况的分析、接口对象划分、对象结构建模、对象模型集成

四个步骤来完成,这种方法能够高效地分析和理解遗产系统信息并将其应用到新的目标系统当中。

3.2 打包(Wrapping)

在基于组件的遗产系统重构过程中,打包和重用一直都比较费时并且效率低下,EJB 模型^[10]主要研究了组件打包的自动化问题,他们还提出用功能限制和打包系统架构的方法来了解业务规则。这种方法在增加系统的可重用性的同时降低了整个重构过程的风险。

面向服务计算的遗产系统中的表格移植是一项极具挑战性的任务,这种移植要求系统接口能够和 Web 服务进行交互。在文献[11]中介绍了一种打包方法,用来实现在遗产软件移植过程中把系统中的交互功能转换成某种 Web 服务,这种包就相当于一个有穷自动机的某个解释程序。文中还提出了一个实现用户与系统交互的智能模型,这个模型是用反向工程中的黑盒技术来实现的,在移植过程和软件体系架构中,遗产系统的功能是以 Web 服务的形式作为输出的。

把一些面向服务的体系结构应用到某些特定的领域,是遗产系统打包中的一个普遍问题。针对这一问题,文献[12]提出了最小公分母接口、最受欢迎接口和协商接口三种设计模式,这三种模式都是先为遗产系统集成一些特定的服务,然后把把这些服务规范化为一些数据模型和术语的形式。该文还用了一个电子商务中的实例介绍了这三种模式的综合应用。

3.3 移植(Migration)

Brodie 和 Stonebraker 提出了 Chicken Little^[13]方法,在这个方法中突出强调了一个通过利用复合关口来实现移植的策略,这种方法虽然能够很好地实现遗产信息与目标系统信息的操作一致性,但是这种方法只能适应于小型遗产系统信息的移植,因为随着系统规模的不断增大,在移植的过程中就越有可能会造成某些信息的丢失。在 Chicken Little 方法的基础上,文献[14]中提出了一个移植遗产软件信息的 Butterfly 方法,这一方法提出用自由关口取代 Chicken Little 方法中的复合关口,以提高移植过程中系统信息的独立性。

Tilley^[15]在综合考虑工程、系统、软件、管理、进化、维护等多方面的基础上提出了在遗产软件重构过程中使用框架的思想,从而使得在目标系统开发的同时能够保证遗产软件的正常运行,但是由于使用的框架是在一个非常高的层次上构造出来的,因而不能很好地应用于实际当中,同时这种方法还忽略了遗产软件中数据的移植。

文献[16]提出了一种能减小遗产系统移植的复杂性的新方法。他们认为综合应用过程动态分析、软件

可视化、知识恢复和分治方法,可以较好地处理遗产系统移植过程中的复杂性问题。SGF 方法^[17]选择了基于组件的开发和聚类技术来进行面向栅格的移植,遗产系统中的程序理解则是通过反向工程中的程序转换方法来实现的。这个方法侧重考虑了通过代码分析来实现重用和在栅格平台上进行遗产系统的移植。

中国科学院计算技术研究所智能信息处理重点实验室史忠植等人提出了 AGrIP 方法^[18],主体网格平台 AGrIP 架构在信息丰富的互联网上,是由底层集成平台 MAGE、中间软件层和应用层共同组成。AGrIP 是一个高度开放的软件环境且其结构可以动态地变化,它是一个规模和复杂性不断扩大的松耦合的计算机网络,同时也可以被看成是一个大型的分布式的信息资源。在这个方法中他们提出了一个通过 AGrIP 这个主体网格智能平台来实现分布系统集成的新途径,创建了网络信息时代软件系统发展的新模式。

3.4 小结

国内外研究人员针对遗产软件重构进行了大量的研究,但这些研究工作往往针对某一特定领域,其应用范围也局限于特定的领域,各种方法体系之间难以统一,难以通用化,也难以形成适合企业各业务领域的一般性重构模型,更难适合分布异构环境下的遗产系统重构。现有的重构工具都是半自动的,分析的程序实体主要限于程序中的类和方法。现有的工作针对设计变化的重构不多,主要的工作是面向设计模式的重构,而这类重构都是通过增加类,并在类中增加方法和数据来实现。由于缺乏一种全局的观点、缺乏与现行的新技术和新方法的结合,所以,现有方法都没能提供一种以实现系统化的转换和集成分布异构遗产软件成为一个基于构件的系统的过程、技术和方法,同时多数方法都不能支持体系结构的演化。

4 Agent - Component

随着 Internet 平台的快速发展,人们迫切需要新的软件技术来支撑这个开放、动态的环境中的软件开发。人们认为 Agent 技术具有发展为未来一种面向 Internet 平台的主流技术的潜力。Agent 是一封装的软件实体,它可在开放、动态的环境下代表用户或其他实体完成指定的任务。Agent 的基本特性有:自主性、反应性、主动性、社会性、适应性、移动性等。与传统开发方法相比,将软件系统构造为一组相互合作的自主的 Agent 这种方法虽尚不成熟,却已在几个方面显示出其优势^[19]:对某些问题的刻画更自然;支持数据和控制的分布;支持遗产软件的利用;适应开放系统的需要。

20世纪90年代软件开发技术的一个重要进展就是软件构件化^[20]。这是由于基于软件总线的软构件技术的研究现在的系统规模越来越大,一个系统要求完成的功能很多,因此软件复用和集成具有非同寻常的意义。当今软件开发技术的主流是软件构件技术,软件构件是指软件系统设计中能够重复使用的模块。构件包括一系列互相关联的操作和服务。软件构件与其他可复用软件模块的区别在于,它既能在设计和使用上进行修改,也能够二进制模块中使用和修改。一个以二进制形式实现的软件构件能够有效地嵌入其他开发商开发的构件中。软构件给软件开发带来的优点如同硬件元件工业中的集成电路所带来的是类似的、集成电路的出现,把硬件工程师们从琐碎的工作中解脱出来,使他能够集中精力于更复杂的设计上。这就造成了前人未曾想到过的如微波炉、笔记本电脑、宇宙飞船等设备的出现。软构件技术的提出也使软件工程师们从琐碎的工作中能够解脱出来,可以集中精力于更复杂和更方便实用的软件工程设计上。Agent技术和构件技术是当前软件开发的主流技术,它们能够提高软件开发的质量和效率,缩短软件开发周期,节约软件开发成本。

Martin L. Griss 在 2001 年撰写的两篇综述性论文^[21,22]中首先提出了“Agent-Component”的概念,并且他认为 AC 能成为柔性、智能、基于 Web 的企业应用系统的基础,利用 AC 能加速软件开发并将成为下一代软件构件。近几年的研究表明,利用构件来开发 Agent 系统或 MAS 确实能提高 AOSE 的性能、质量和效率^[23]。

AC 是松散耦合的,且具有构件和 Agent 的双重特征;其组合过程中无需对原有构件做任何更改;AC 是能同时提供服务描述及服务的构件;AC 是可被重用和定制的;AC 是语言中立、独立实现的软件服务包,提交在一个封装的可替换的容器内,经由一个或多个发布接口来存取;AC 不受平台的约束和应用的限制,AC 具有自治性、反应性、主动性和社会能力。

5 结束语

遗产软件常表现为脆弱、非柔性、孤立、不可扩展、缺乏开放性等。因此,建立一个柔性、分布、可扩展、开放的软件体系结构和服务框架是遗产软件重构的关键因素。国内外研究人员针对遗产软件重构进行了大量的研究,但这些研究工作往往针对某一特定领域,其应用范围也局限于特定的领域,各种方法体系之间难以统一,难以通用化,也难以形成适合企业各业务领域的一般性重构模型,更难适合分布异构环境下的遗产系

统重构。

Agent 技术和构件技术有许多相似之处,组合 Agent 技术和构件技术各自的优势,创建未来更具柔性、可重用性和可定制性的 AC,将为有效地解决遗产软件的重构尤其是大型复杂系统的开发问题提供新的思路和途径。如何利用 AC 来探索一种真正有效且高效、通用且易于实现的分布异构遗产重构的技术和方法,增加软件开发的柔性、动态性、适应性,将是今后研究和工作的重点。

参考文献:

- [1] Bennett K. Legacy Systems: Coping with Success[J]. IEEE Software, 1995, 12(1): 19-23.
- [2] 陆波, 赵合计. 遗产软件的重构过程[J]. 计算机工程与科学, 2004, 26(6): 89-91.
- [3] Weiderman N H, Bergy J K, Smith D B, et al. Approaches to Legacy System Evolution[R]. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997.
- [4] Comella-Dorda S, Wallnau K, Seacord R C, et al. A Survey of Legacy System Modernization Approaches[R]. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.
- [5] Woods S, O'Brien L, Lin T, et al. An Architecture For Interoperable Program Understanding Tools[C]//In Proceedings of the 6th International Workshop on Program Comprehension. [s.l.]: IEEE, 1998: 54-63.
- [6] Kazman R, Woods S G, Carriere S J. Requirements for Integrating Software, Architecture and Reengineering Models: CORUM II[C]//Fifth Working Conference on Reverse Engineering. Honolulu, Hawaii: [s.n.], 1998: 154-163.
- [7] Cha Jung-Eun, Kim Chul-Hong, MaRMI-RE: Systematic Componentization Process for Reengineering Legacy System [C]//ICCSA 2005. Lecture Notes in Computer Science. Heidelberg, Berlin: Springer, 3482/2006: 896-905.
- [8] Cho Eun Sook, Cha Jung Eun, Yang Young Jong. MARM-RE: A Method and Tools for Legacy System Modernization [C]//SERA 2004. Lecture Notes in Computer Science. Heidelberg, Berlin: Springer, 2005: 42-57.
- [9] Lee Chang-Mog, Yoo Cheol-Jung, Chang Ok-Bae. An Object Structure Extraction Technique for Object Reusability Improvement Based on Legacy System Interface[C]//ICCSA 2005. Lecture Notes in Computer Science. Heidelberg, Berlin: Springer, 2006: 758-768.
- [10] Lee M S, Shin S G, Yang Y J. The Design and Implementation of Enterprise JavaBean (EJB) Wrapper for Legacy System[C]//International Conference on Systems, Man and Cybernetics. [s.l.]: IEEE Computer Society, 2001.
- [11] Canfora G, Fasolino A R, Frattolillo G, et al. Migrating Interactive Legacy Systems To Web Services[C]//CSMR 2006.

- [s.l.]: IEEE Computer Society, 2006: 24 - 36.
- [12] Millard D E, Howard Y, Chennupati S, et al. Design Patterns for Wrapping Similar Legacy Systems with Common Service Interfaces[C]//The 4th European Conference on Web Services (ECOWS). [s.l.]: IEEE Computer Society, 2006: 191 - 200.
- [13] Brodie M, Stonebraker M. Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach[M]. [s.l.]: Morgan Kaufmann Publishers Inc, 1995.
- [14] Lawless B W, Bisbal D, Richardson J, et al. The Butterfly Methodology: A Gateway - free Approach for Migrating Legacy Information Systems[C]//Proceedings of the 3rd IEEE Conference on Engineering of Complex Computer Systems (ICECCS97). Villa Olmo, Como, Italy: [s.n.], 1997: 200 - 205.
- [15] Tilley S R, Smith D B. Perspectives on Legacy System Reengineering[EB/OL]. 1996 - 11. <http://www.sei.cmu.edu/reengineering/lisree>.
- [16] Wu Lei, Sahraoui H, Valtchev P. Coping with Legacy System Migration Complexity[C]//Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems 2005. [s.l.]: IEEE Computer Society, 2005: 600 - 609.
- [17] Li Jianzhi, Yang Hongji. Incorporating legacy system into semantic grid framework[C]//International Conference on Information Reuse and Integration, 2006. [s.l.]: IEEE Computer Society, 2006: 319 - 324.
- [18] Luo Jiewen, Shi Zhongzhi, Wang Maoguang, et al. AGriP: An Agent Grid Intelligent Platform for Distributed System Integration[C]//APWeb Workshops 2006, Lecture Notes in Computer Science. Heidelberg, Berlin: Springer, 2006: 590 - 594.
- [19] 马晓星. Internet 软件协同技术研究[D]. 南京: 南京大学, 2003.
- [20] 杨美清, 梅宏, 李克勤. 软件复用与软件构件技术[J]. 电子学报, 1999, 27(2): 68 - 75.
- [21] Griss M. Software Agents as Next - Generation Software Components[M]//Component - Based Software Engineering. Putting the Pieces Together. In: Heineman G, Council W. Reading, Mass: Addison Wesley Longman, 2001.
- [22] Griss M L, Pour G. Accelerating Development with Agent Components[J]. COMPUTER, 2001, 34(5): 37 - 43.
- [23] Amor M, Fernández L F, Mandow L, et al. Building Software Agents from Software Components[C]//In: Conejo R et al. CAEPIA - TTIA 2003. Lecture Notes in Artificial Intelligence. Heidelberg, Berlin: Springer, 2004: 221 - 230.

(上接第 113 页)

- 平台的设计与实现[J]. 微电子学与计算机, 2006, 23(3): 180 - 181.
- [3] 肖志峰, 龚健雅, 王艳东, 等. 面向对象的软件设计模式在配电 GIS 中的应用[J]. 测绘信息与工程, 2005, 30(3): 3 - 5.
- [4] 李慧珍, 郑恩让, 张会生. 电信管理网络中观察者模式的应
- 用研究[J]. 微电子学与计算机, 2005, 22(9): 89 - 92.
- [5] 曾蔚, 陈维斌. 设计模式在新生报到系统中的应用与实现[J]. 计算机技术与发展, 2007, 17(7): 178 - 182.
- [6] 何杨欢. OPC 技术在 DCS 数据采集系统中的应用[J]. 化工发展, 2006, 25(12): 1496 - 1498.

(上接第 117 页)

- [9] Document type definition - DTD[EB/OL]. 2004 - 07. <http://www.w3schools.com/dtd/default.asp>.
- [10] Java DTD Parser[CP/OL]. 2004 - 07. <http://www.wutka.com/dtdparser.html>.
- [11] Sperberg - McQueen C M, Thompson H. XML Schema 1.1 specification[EB/OL]. 2000 - 04. <http://www.w3.org/XML/Schema>.
- [12] Davidson A, Fuchs M, Hedin M, et al. Simple OutLine XML - SOX[EB/OL]. 1999 - 07. <http://www.w3c.org/TR/NOTE-SOX/>.
- [13] RFC1014 - XDR External Data Representation standard: External data Representation standard[EB/OL]. 1987 - 07. <http://www.faqs.org/rfcs/rfc1014.html>.
- [14] Document Structure Description - The DSD 2.0 specification[EB/OL]. 2004 - 07. <http://www.brics.dk/TSD/>.
- [15] Wang G, Liu M. Extending XML Schema with Nonmonotonic Inheritance[C]// In Proceedings of 1st International Workshop on XML Schema and Data Management (ER Workshop XSDM'03). Chicago, Illinois, USA: [s.n.], 2003.
- [16] 张晓琳, 谭跃生, 周健. 用继承扩展 XML Schema[J]. 计算机工程与应用, 2006, 42(4): 179 - 182.
- [17] 张晓琳, 王国仁. 用继承扩展 XML - RL[J]. 小型微型计算机系统, 2005, 26: 2 - 4.
- [18] Document Object Model (DOM Level 3 Core)[EB/OL]. 2002 - 04. <http://www.w3.org/DOM/>.
- [19] The document of Javacc[EB/OL]. 2005 - 04 - 10[2006 - 01 - 02]. <http://javacc.dev.java.net/doc/CharStream.html>.
- [20] 张晓琳, 王国仁. 面向对象的 XML 数据的存储模式研究[J]. 小型微型计算机系统, 2004, 25: 11 - 13.