一种面向复用的增量软件开发过程框架*)

张广泉[1] 戎 玫[2] 陆 明[1]
(苏州大学计算机科学与技术学院 苏州 215006)[1]
(暨南大学深圳旅游学院计算中心 深圳 518053)[2]

摘 要 以复用思想指导软件开发是软件产业发展的趋势,而当前主要面向复用的软件开发方法没有考虑到我国软件企业以中小型为主的现状,缺乏针对性,使得复用在我国软件开发中应用不够广泛。在基于构件开发的基础上提出了一种面向复用的增量软件开发过程框架。该过程框架适于我国中小型软件企业,采用了以体系结构为指导的增量开发模式,涵盖了可复用资产的生产过程和利用可复用资产的软件开发过程。

关键词 软件复用,软件构件,领域,增量开发

A Process Framework of Reuse-Oriented Incremental Software Development

ZHANG Guang-Quan[1] RONG Mei[2] LU Ming[1]
(School of Computer Science and Technology, Soochow University, Suzhou 215006)[1]
(Shenzhen Tourism College, Jinan University, Shenzhen 518053)[2]

Abstract Software development approaches in software industry now are directed by reuse idea. But most present reuse-oriented software development approaches do not apply to medium-sized software companies that are common in our country. So software reuse is not extensively used. Based on Component-Based Development(CBD), a process framework of reuse-oriented incremental software development is proposed. It is designed for medium-sized software companies, and uses incremental development model with software architecture. It covers the whole reusable development processes, including development for reuse and development with reuse.

Keywords Software reuse, Software component, Domain, Incremental development

1 引言

软件复用,特别是系统化的软件复用,被认为是解决软件危机,实现软件生产工业化的有效途径之一,也是未来软件产业发展的必然趋势[1~4]。经过30多年的不断研究和发展,软件复用技术逐渐成熟,由探索阶段过渡到应用阶段,出现了一些以复用思想为指导的软件开发方法。本文将这样的方法称为面向复用的软件开发方法,它们运用软件复用原理和技术进行系统开发,以期望提高软件质量和生产率。区别于传统开发方法,它们在软件生命周期过程中明确定义了软件复用,是系统化的软件复用方法。

当前,我国软件企业规模以中小型为主,要实现软件产业的工业化,必须重视它们的发展;但是它们对软件复用的理解还大部分停留在基本概念阶段,缺乏实际的应用;而且现在主要的面向复用的软件开发方法还不够成熟,存在一定的缺点和局限性,没有考虑我国国情,不适用于我国中小型软件企业。针对这样的状况,本文对当前主要的面向复用方法进行分析,选取合适的方法进行本地化研究,提出一种适合于我国中小型软件企业的面向复用增量软件开发过程框架,对于在我国如何将软件复用理论运用到实际开发过程中具有一定的参考作用。

本文首先简介了面向复用的软件开发方法,分析和比较了现阶段主要的面向复用方法;然后在它们的基础上提出了适合于我国中小型软件企业的软件过程框架,并给出了研究实例;最后是结论和进一步工作。

2 面向复用的软件开发方法

软件危机的出现,使人们认识到软件生产既不是求解数学题,也不是创作小说,它是工程而不是艺术,要用工程学科的知识来生产软件[5]。复用是每一门工程学科的主要部分,任何一门工程学科都不会遵从"一切从零开始"的开发模式,而是尽可能最大限度的利用已有的资源。面向复用的方法利用了软件系统间的相似性,将多个系统相似的部分抽象成可复用资产,通过重复使用资产而不是重新开发系统,缩短了软件产品的开发时间并降低了开发成本,并且可复用资产是已经开发好的,经过完全的测试和验证,质量和可靠性在一定程度上能得到保证。因此,相对于传统的方法,面向复用的方法有缩短开发周期、降低开发成本、提高软件质量、提高软件生产率等优势。

根据系统建立的方式,面向复用的方法可分为两类:(1)基于合成的方法:将软件构件插装在一起,集成和组装出新系统,是对构件的复用;(2)基于生成的方法:利用应用程序生成

*)基金项目:中国科学院计算机科学国家重点实验室开放课题(编号 SYSKF0303);江苏省高校自然科学研究项目(批准号 05KJB520119);重庆市科学技术研究项目(编号 040803)。**张广泉** 博士、教授,主要研究方向:形式化方法、软件工程。**戎 玫** 副教授,研究方向为软件工程、电子商务等。**陆 明** 硕士研究生,主要研究方向为软件复用。

器将输入规约转换成应用系统，是对模式的复用。基于合成的方法出现较早，它类似于将集成电路芯片插装成硬件系统，其思想容易被理解，技术要求相对不高。构件技术的成熟使得这类方法的研究进展迅速，现在这类方法中具有代表性的是基于构件的方法，并在其基础上发展出了基于 COTS (Commercial Off-The-Shelf)构件的方法和软件产品线方法。基于生成的方法把软件开发抽象成一组参数化规则，输入参数输出系统，涉及到数学和逻辑原理，难度较大，只在特殊应用领域获得局部成功，距离广泛应用还相差较远，当前这类方法中主要的有基于应用程序生成器的方法。

我国中小型软件企业的人力物力资源有限，因此，相对成熟的基于构件的方法适用于它们。基于构件方法的开发过程如图 1 所示，它的开发过程类似于传统的瀑布式模型，主要区别在于虚线所示的系统组装过程，它分为构件审核、构件适配和构件合成[6]。构件审核即选取候选构件，验证它们对需求的满足程度、对体系结构的匹配程度以及它们的质量特性，筛选出最佳构件。构件适配即选择正确的参数对构件进行配置或适当的修改来适应新的系统。构件合成即在构件模型的基础上，通过构件框架、体系结构描述语言、胶水代码、脚本语言和协同语言等技术，将适配的或新开发的构件组装成一个完整的应用系统。基于构件的方法适合开发大型软件，并且前期需要一定量的资金投入；基于 COTS 构件的方法通过购买商业构件避免了前期的大量资金投入，但它需要有成熟规范的市场支持并有一定的风险[7,8]；软件产品线方法是现阶段最高效的面向复用方法，它利用核心资产生产领域内的一系列产品，但对人力物力资源要求很高[9,10]。因此，它们都不太符合我国中小型企业以开发小规模常用软件为主、注重短期效应进行快速开发和资金量不多等特点，需要对它们进行本地化研究。

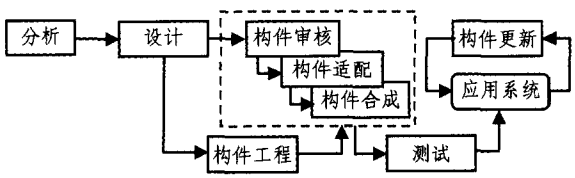

图 1　基于构件的软件开发过程

3　适合于我国中小型软件企业的软件开发过程框架

我国中小型企业应该将产品定位在某一领域范围内，以形成自己的核心竞争力，因为在领域内构件在相似系统间的复用率高，能形成范围经济，即一个厂商同时生产多种产品的支出小于多个厂商分别生产这些产品的支出，那么这个厂商就获得了范围经济。同时，基于构件的合成方法与增量式软件开发的思想不谋而合，使得在基于构件的方法中能充分利用增量开发的增量资金投入特点，符合我国中小型软件企业的人力物力资源有限不能进行大规模资金投入的状况。因此，本文对其进行面向领域、增量开发和降低开发成本等方面研究，提出了适合于我国中小型企业的开发过程框架。

如图 2 所示，整个过程框架立足于领域，将软件开发分成两个主要阶段：领域分析阶段和系统实现阶段。领域分析阶段是针对领域进行的，由创建领域需求模型、创建领域体系结构参考模型和创建领域构件列表 3 个子过程组成；在系统实现阶段进行领域内系统的开发，包含创建系统体系结构、计划构件、设计构件和实现构件 4 个子过程，其中计划构件、设计构件和实现构件是一个增量迭代过程，通过与构件库的交互最终完成系统的开发。

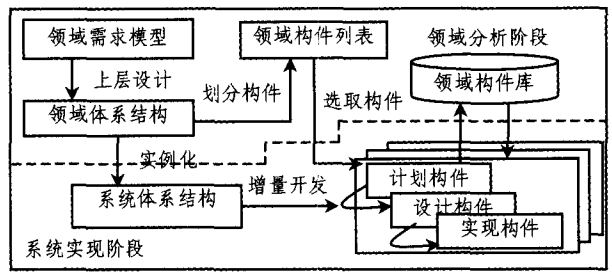

图 2　领域内软件开发过程框架

3.1　领域分析阶段

现有领域内面向复用的方法，如软件产品线等，虽然有较高的复用率和丰厚的投资回报，但是构件库等可复用资产的开发需要在实际系统项目开发前投入大量人力物力，使得开发成本部分集中开发初期。针对我国中小型企业现状，在此采用构件库的增量开发模式，即在领域分析阶段只进行领域构件列表的创建，将构件的实现推迟到实际系统的开发中，这样将早期的成本分散到多次开发中，降低领域内软件复用的门槛；同时为了保证构件兼容性，降低增量开发的风险，引入合适的软件体系结构支持基于构件的开发。因此，在领域分析阶段主要有创建领域需求模型、创建领域体系结构参考模型和创建领域构件列表三个过程。

(1)创建领域需求模型

领域是指一组具有相似和相近软件需求的应用系统所覆盖的功能区域，不同于单个系统，领域内存在共性和变化性需求，即领域内单个应用系统间的共同部分和区别部分。因此，在创建领域需求模型时需要对领域所包含的多个系统进行需求分析，并且选择合适的模型将它们的共性需求和变化性需求表示出来。传统的需求建模方法不支持对变化性需求的建模，不能用来建立领域需求模型。现阶段可以使用支持变化性的用例模型、特征模型等模型来建模领域需求，由于特征模型具有可复用性、结构良好、易于交流和图形化建模等优点，在此选用它来建模领域需求。特征是可复用、可配置的需求单元，并且每个特征必须与其它特征相区别，文[11]中首次运用特征在领域内进行需求建模，经过后来不断的研究和发展[12~14]，现在特征模型已成为识别和捕捉变化性的关键技术。

(2)创建领域体系结构参考模型

在领域需求模型建立后，就可以根据需求模型创建领域体系结构参考模型。为了便于理解和交流，该参考模型用开发小组认可的适当方式表示出来，可以是文档、图形等多种方式。它是领域的高层抽象描述，应当由经验丰富的领域专家来设计并对其进行验证。不同于单个系统的体系结构模型，领域体系结构模型是覆盖领域内多个系统的高层设计模型，需要支持领域内的共性和变化性。在进行领域内系统开发过程中，需要将这个模型进行实例化，对变化性进行裁剪，建立单个系统的体系结构，因此领域体系结构参考模型在领域内具有举足轻重的地位。

(3)创建领域构件列表

完成领域体系结构参考模型后，就可以进入领域构件列表的创建阶段。根据领域体系结构参考模型由经验丰富的开发专家将领域内的构件划分出来，建立领域构件列表。系统构件列表由不同粒度的构件组成，可以是大粒度的由多个构

件组成的复合构件，也可以是功能较单一的原子构件。构件可以从领域构件库中搜索得到或在项目中开发，还可以从第三方购买。由于不同构件之间需要用到不同的构件组装技术，如连接子、胶水代码和框架等，所以每个复合构件所用的组装技术要在列表中表示出来。构件的设计和开发都遵循领域体系结构参考模型的标准和规范，以减少构件的兼容性问题，提高合成系统的性能。

3.2 系统实现阶段

当需要在领域内开发单个应用系统时，就进入了系统实现阶段。通过利用构件合成应用系统，使得整个领域内系统实现是一个增量开发过程，包括系统的增量完成、领域构件的增量积累和资金的增量投入。这样在应用系统开发完成后将实现领域构件列表中的部分领域构件，并作为可复用资产保存入库，可以应用于领域内其它系统的开发中。

系统实现阶段的第一步是根据待开发的单个应用系统需求，定制领域内的变化性，将领域体系结构实例化成系统体系结构。然后以系统体系结构为指导进入迭代开发过程。如图3所示，这是一个从领域构件组装列表选取相应构件进行开发的迭代过程，同时也是实际开发对领域构件列表的反馈过程，可以根据实际开发经验增加未列举的重要构件或删除多余构件。一次迭代包括计划构件、设计构件和实现构件三个子过程，完成组成系统的部分构件。在计划构件中确定这次迭代开发需要完成的构件和构件的来源，即安排开发进度。因为许多构件互相不影响，可以独立开发，增加并发度，所以既要根据关键路径对整体进度作好计划也要对每次迭代的任务作好安排，才能保证项目的不间断进行，并提高开发效率，节约时间和成本。计划构件结束后，可以开始对构件进行设计和实现。在设计阶段对需要自己开发的构件进行设计、设计验证，在实现阶段对其进行编码、测试，完成后可以作为可复用资产保存到构件库中。接着根据构件组装列表中提供的组装技术进行相应构件的组装并配置相应的变化性，这些构件可以是自己项目小组开发的，也可以是可复用构件库里的资产，或是来自第三方的。组装完成后进行模块测试，确认没有问题后这次迭代就结束了。在新的一次迭代中，可以根据上次的完成情况，计划新的任务，增加开发的灵活性。经过若干次迭代后最终完成整个系统的开发，它一个高效的并发过程并且经过一系列测试保证了质量。其中，构件库的管理也是一个重要环节，要根据实际情况进行构件的更新。

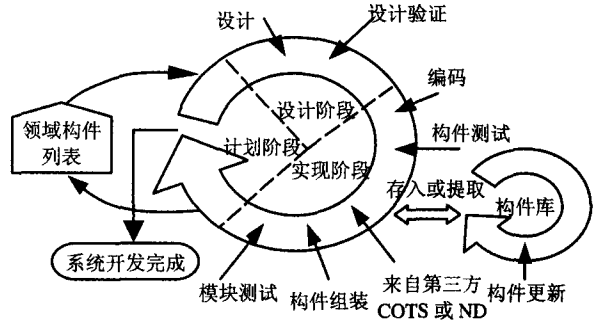

图3 系统实现的增量迭代过程

4 实例研究

货运管理领域属于信息管理范畴，是一个成熟稳定的领域，适合软件复用的实例研究。货运管理领域主要任务是根据委托人的要求将货物用适当的方式运输到目的地并收取相应的费用。根据上述领域内开发过程框架，首先建立该领域的领域需求模型。如图4所示，为了支持变化性需求，使用特征模型对该领域建模。由于篇幅的原因，这里只列举了主要的特征。

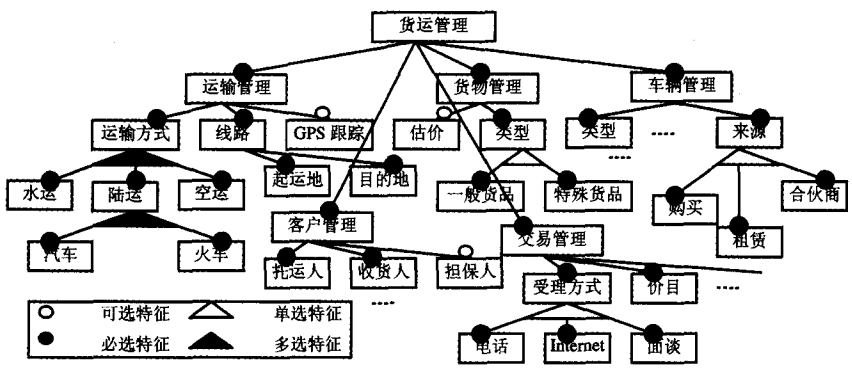

图4 货运管理领域主要特征模型

接下来，在领域需求模型的基础上，建立领域体系结构参考模型，通过对特征模型的进行一步分析，为该领域的体系结构参考模型选择分层结构。将该领域由底向上分为系统软件层、系统构件层、领域构件层、应用系统层四个层次，每一层具有大致相同的抽象程度，上层和特定应用有关，下层更具一般性。每一层都建立在另一个更一般的层上，向其上层提供服务，并利用下层的服务，并且同一层间可以相互交互。其中，系统软件层包括计算和联网基础实施的软件，该领域可以选用 UNIX、LINUX、WINDOWS2000 SERVER 等工作站，WINDOWS98、2000、XP 等客户端，TCP/IP 协议，硬件接口等；系统构件层包括系统的开发环境及支持环境中的各类构件及中间件，在标准数据库访问接口上根据不同的数据库选用合适的中间件如 ODBC、Oracle Net8、Sybase Netlib 等，分布对象计算中间件可以选用 CORBA，EJB，DCOM 等；领域构件层封装领域业务相关的领域知识，分为领域通用构件和应用专用构件，该领域内通用构件如单据管理构件、查询统计构件、报表管理构件、调度管理构件等等，而适合不同应用系统的构件有 Web 端账单下载构件、租赁管理构件、合伙商管理构件、水运、陆运耗损构件、普通物品管理构件、贵重物品管理构件等等；领域业务层由一组提供具体逻辑功能的软件系统组成，这些软件系统通过接口可以直接互操作或者通过底层提供的某些服务或对象规范间接口实现互操作，如运单生成、对账单结算、回单管理等等。

领域分析阶段的最后一步就是根据领域体系结构参考模型抽取领域构件，建立领域构件列表。该领域构件列表中的构件有：单据管理构件、查询统计构件、数据维护构件、报表管

理构件、审批构件、调度算法构件等等百余种领域通用构件和应用专用构件。在建立了这些领域资产后，可以根据实际要开发的系统，在系统实现阶段实例化分层体系结构，从领域列表选择合适的构件进行增量迭代开发直至系统的完成。

结束语 本文提出的领域内软件开发过程框架将增量开发与复用技术有效地结合起来，为我国的中小型软件企业提供一个进行复用活动的有效途径。相对于传统开发方法，具有以下优势：(1)基于构件方式，具有软件复用的优点；(2)增量开发，分散了资金的投入，降低了开发成本；(3)以体系结构为指导，将自上到下的需求分解与自底向上的软件开发相结合，保证了系统的质量和可靠性，降低开发风险；(4)定位领域，形成范围经济，提高企业竞争力；(5)在迭代中完成系统，测试充分、开发迅速等。

由于软件复用技术还在不断发展，面向复用的方法应用不够广泛，它们都处于进一步完善阶段，所以该过程框架还存在许多不足和需要改进的地方。比如，遇到需求变化时该如何处理、可复用资产库如何有效管理和从第三分获取的构件的兼容性问题等。

下一步的研究工作包括通过更多的实践和应用研究来完善该开发框架；构件库的组织形式及如何有效查找匹配合适的候选构件；在开发过程中如何有效地应对需求的变化；研究开发小组的组织结构对开发效率的影响等。

参 考 文 献

1 Krueger CW. Software Reuse. ACM Computing Surveys, 1992, 24(2): 131～183

2 Mili H, Mili F, Mili A. Reusing Software: Issues and Research Directions. IEEE Transactions on Software Engineering, 1995, 21(6): 528～562

3 Frakes WB, Kang K. Software Reuse Research: Status and Future. IEEE Transactions on Software Engineering, 2005, 31(7): 529～536

4 Jacobson I, Griss M, Jonsson P. Software Reuse: Architecture, Process and Organization for Business Success. 1st edition. New York: Addison Wesley, 1997

5 Cox BJ. Planning the Software Revolution. IEEE Software, 1990, 7(6): 25～35

6 贾育. 基于演化构件的软件复用方法:[博士学位论文]. 北京：中科院软件研究所，2002

7 Torchiano M, Morisio M. Overlooked Facts on COTS-based Development. IEEE Software, 2004, 21(2): 88～93

8 Brownsword L, Oberndorf T, Sledge C. Developing New Processes for COTS-Based Systems. IEEE Software, 2000, 17(4): 48～55

9 Clements P, Northrop P. A Framework for Software Product Line Practice, Version 4.2. http://www.sei.cmu.edu/productlines/framework.html, 2005

10 王广昌. 软件产品线关键方法与技术研究:[博士学位论文]. 浙江：浙江大学，2001

11 Kang KC, Cohen SG, Hess JA, et al. Future-Oriented Domain Analysis (FODA) Feasibility Study: [技术报告]. Pittsburgh: Camegie Mellon University, Software Engineering Institute, 1990

12 Kang KC, Kim S, Lee J, et al. FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architecture. Annals of Software Engineering, 1998, 5: 143～168

13 Griss ML, Favaro J, d'Alessandro M. Integrating Feature Modeling with the RSEB. In: Proceedings of the Fifth International Conference on Software Reuse, Victoria, 1998

14 Chastek G, Donohoe P, Kang KC, et al. Product Line Analysis: A Practical Introduction:[技术报告]. Pittsburgh: Carnegie Mellon University, Software Engineering Institute, 2001

(上接第 220 页)

示为：$(M\chi)(x)\triangleq 2[2^{-n}\sum_{y}x(y)]-\chi(x)$，$\chi(x)$是取本征态 x 的概率幅。

(3)对输出进行测量，观察结果设为 S。若 $f(S)=1$，则得到结果，否则重新开始算法。

依这个算法，我们所需要的结果的概率不断增大，而其它本征态概率减小了。而且可以证明这个算法超越了经典的搜索算法，时间复杂度降低了。但要设计精巧的酉变换和算法，可以说是一件十分困难的事情，需要我们不断探索研究。

小结 由本文的讨论可以看出，量子力学中关于测量的研究，其基本观念还存在着很大的争议，不同学派观点的分歧往往在于对微观客体物理本质特征和规律的认识上，甚而上升到哲学意义的探讨。哥本哈根解释认为塌缩是随机的，而爱因斯坦不能接受非决定论的观点；有的学派认为测量发生了塌缩，而有的学派认为根本没有塌缩而是世界分裂了。这一切都没有定论，要深刻认识它，还需要我们不断探索研究。虽然量子力学还非尽善尽美，但它可以较好地解释大量实验中的量子现象。另一方面，量子力学对信息科学产生了革命性的影响，对量子计算等领域有很强的指导意义。

致谢 衷心感谢中国科技大学的张永德教授在我们深入学习量子力学过程中所给予的指导与帮助。感谢南京大学计算机科学与技术系量子信息与量子计算研讨班的徐家福教授和宋方敏教授，本文的工作正是在他们的精心指导下完成的，在论文的选题、开题直至定稿完成的整个过程中，两位教授都付出了极大的心力。感谢研讨班的同学们，在与他们讨论的过程中我得到了许多启发。

参 考 文 献

1 周光召．回顾与展望——纪念量子论诞生 100 周年．物理，2001，30：259～264

2 Einstein A, Podolsky B, Rosen N. Can quantum mechanical description of physical reality be considered complete? Physical Review, 1935, 47: 777～780

3 孙昌璞．量子理论若干基本问题研究新进展．物理学进展，2001，21(3)

4 张永德．量子力学．第一版．北京：科学出版社，2002

5 曾谨言．量子力学导论．第一版．北京大学出版社，1998

6 Leggett A J. The Quantum Measurement Problem. Science, 2005, 307: 871～872

7 Giancarlo G, Collapse Theories, Zalta E N, ed. The Stanford Encyclopedia of Philosophy. (Spring Edition), 2000

8 Faye J. Copenhagen Interpretation of Quantum Mechanics. Zalta E N, ed. The Stanford Encyclopedia of Philosophy. (Summer Edition), 2002

9 Everett H III. "Relative state" formulation of quantum mechanics. Reviews of Modern Physics, 1957, 29: 454～62

10 Albert D, Loewer B. Interpreting the Many Worlds Interpretation. Synthese, 1988, 77: 195～213

11 Nielsen M A, Chuang I L, ed. 量子计算和量子信息. 赵千川译. 清华大学出版社，2004

12 Selinger P. A Brief Survey of Quantum Programming Languages. In: Proceedings of the 7th International Symposium on Functional and Logic Programming, Nara, Japan. 2004. 1～6

13 Bettelli S, Calarco T, Serafini L. Toward an architecture for quantum programming. arXiv: cs. PL/0103009v2 23 Nov. 2001

14 Zuliani P. Quantum Programming: [PhD thesis]. University of Oxford, 2001

15 Grover LK. A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, 1996. 212～219