

基于构件的软件开发方法

陈建新，李革新

(孝感学院 计算机科学系, 湖北 孝感 432100)

摘要: 本文介绍了基于构件的软件开发方法的有关概念, 可重用构件的描述、分类、存储和检索; 分析了它的特征, 阐明了利用可重用构件构造软件系统的过程; 探讨了目前主要的研究重点, 展望了它的应用前景。

关键词: 构件; 软件重用; 基于构件的软件开发

中图分类号: TP311.52 文献标识码: A 文章编号: 1671—2544(2001)06—0014—04

0 引言

在软件开发方面, 多数软件的开发都是针对具体的应用。软件的开发一般都是从头开始, 即从需求分析开始, 经过设计, 编写代码, 测试, 最后交付使用。近三十年来, 软件开发商一般都是以这种手工方式生产软件, 从而出现了大量同类软件的重复开发, 软件生产的质量也不高, 软件的发展严重滞后于硬件。根据硬件设计、生产的经验, 人们希望有一些软件工厂专门生产软件构件(Component)。软件人员在开发软件时可根据需要到软构件市场购买所需的构件组装生产软件。显然这样生产的软件可降低软件开发、维护的费用, 提高软件生产率, 促进软件开发过程的标准化。由于软件工厂生产的这些构件往往已经经过了严格的测试和广泛的使用, 其可靠性通常比较高, 从而也提高了新软件的质量。

关于软件重用问题, 软件研究人员从各自不同的角度对软件重用这个概念进行了说明。认为软件重用是在新的开发项目中使用以前已获得的概念和对象; 在建立新的软件系统时, 重用某些以前开发的对系统有用的信息来产生新系统。

软构件的重用(组合式重用)是一个系统工程,

从设计软构件到使用软构件组装生成软件都需要认真研究。怎样描述、生产软构件, 如何从构件库检索所需的构件, 构件的组装集成问题, 异质构件间如何协同计算等都是需要研究与解决的问题。

1 软件构件的概念

软件开发方法已经经历了符号语言、高级语言、结构化和模块化编程以及面向对象的方法。近三十年来, 软件开发方法以及组织结构和管理方式都没有发生太大的变化。而基于构件的软件开发方法(Component Based Software Development 简称CBSD)是在吸取了已有软件开发方法的经验基础上的一种全新的、高效的开发方法。关于软件构件, 目前还没有一个统一的定义。认为构件是分布对象技术的一个重要概念; 一个可运行的构件是一个或多个程序的动态捆绑包, 它可作为一个单元进行管理, 通过接口进行访问; 一个构件指任何可被分离出来的标准化的、可重用的公开接口的软件(子)系统; 一个构件是明确规定了接口和环境附属物所构成的组合体; 构件是完成通用或特定功能的可重用软件模块, 一般遵循二进制标准, 其实现不依赖于某种高级语言, 外界仅能通过接口访问构

收稿日期: 2001—10—10

作者简介: 陈建新(1953—), 男, 湖北应城人, 孝感学院计算机科学系副教授。

李革新(1969—), 男, 湖北孝昌人, 孝感学院计算机科学系副教授, 理学硕士。

件; 构件支持对象意义下的封装性、继承性以及多态性, 支持互操作的对象; 可以在跨越地址空间、网络、语言、操作系统的异质环境下被调用或与其他构件协同工作。构件支持灵活的“即插即用”, 可方便地插入到软件构架中。

基于构件的软件开发(CBSD), 主要思想与搭积木的过程类似。一般先构筑系统的总体软件构架, 然后从构件库中检索出各构件, 并依次把它们安装到系统中去。软件构架是对系统结构设计的刻画, 可以作为一种大粒度的抽象级别较高的软件构件进行重用, 而且它还为软件的组装提供了基础和上下文。

CBSD 可以提高软件的可重用性, 使软件开发摆脱了小作坊式的工作方式, 按照大规模的工业化方式进行。CBSD 是软件开发方法发展的必然结果, 现在 CBSD 方法已成为软件重用研究的重点。

2 CBSD 的基本活动

CBSD 与传统的软件开发方法有着明显的不同, 其基本活动包括构件的描述、构件的获取、构件的存取、构件的检索、构件的组装以及测试和运行等过程。

2.1 构件的描述

对于一个可重用的构件, 应该是完整的, 一个构件重点解决一个问题。根据应用领域的特征及相似性, 进行领域分析, 并预测可重用性, 对具有重要价值的构件要进行一般化处理, 以便能适应新的应用领域。

构件的描述, 已经出现了多种构件模型, 其中 3C 模型是学术界普遍认同的一个具有指导性作用的构件模型。该模型从概念(Concept)、内容(Content)和语境(Context)三个方面来描述构件。“概念”是关于“构件做什么”的抽象描述, “内容”是概念的具体实现, “语境”是描述构件和外围环境在概念级和内容级的关系, 语境刻画构件的应用环境。

对构件的描述可用系统化和标准化的用语表示, 也可用自然语言进行描述。对构件的描述必须清晰, 无二义性, 且容易理解。构件的描述一般由两部分组成:

- 1) 构件的类型信息说明, 包括构件的功能类型、目标对象、源对象、中间对象、系统类型、活动类型、应用类型等。

- 2) 构件的实现说明及配置特征, 包括接口信息及使用方法说明、核心算法说明、实现语言、开发方法、运行环境、构件版本号、制作时间、关联构件、参考构件等。

2.2 构件的获取

CBSD 技术要求具备大量可供选择的可重用构件, 这些构件都是在充分进行领域分析的基础上, 构造并生成的基本构件。它们必须满足下列条件: 构件的设计应具有较大的抽象程度; 构件应易于调整; 易于组装; 必须具有可检索性; 构件必须经过充分的测试。构件的获取可以商业采购, 也可以利用项目承包商和合作伙伴开发, 或者在领域工程和再工程的基础上从已有的应用系统中发掘和提炼以及组装可重用构件, 或者针对新需求重新自主开发新构件。对于外来构件要考虑以后的维护与升级。

构件的接口是在获取构件时必须认真考虑的问题。接口是一组函数和变量的集合, 程序的两个不同部分可通过它进行通信。为实现程序框架的重用, 有两点必须注意。其一, 构件只是接口的实现, 所有应用程序对构件的调用都必须通过接口来完成。应用程序并不直接同构件打交道, 与构件直接联系的只能是接口。其二, 接口的不变性。因为只有这样的接口才可能被一致使用。构件的变化是被限制在自身范围内的, 它并不涉及到具体的的应用程序。

2.3 构件的存储

为了在软件开发过程中能重用现有的软件构件, 必须要经过积累, 组织成构件库。按照领域特征以及分类方法对构件进行分类并置于构件库的适当位置。目前使用的方法是将构件分类存储。对可重用构件进行分类主要是为以后的检索提供支持。构件的分类方法及相应的库结构对构件的检索和理解有着极为深刻的影响。构件库的组织方法有: 1) 基于关键字的分类法, 它是一种简单的构件库的组织方法, 按照从抽象到具体的顺序逐次分解为树型或有向无回路结构。在浏览关键字分类结构的同时, 将构件置于最合适的原子级构件之下。若无法找到相应的属主关键字, 则可以引进新关键字, 扩充现有的关键字分类结构。2) 刻面分类法, 在刻面分类机制中, 定义若干用于刻画构件特征的刻面(facet)。刻面分类模式根据一组刻面对

构件进行分类，不同侧面从不同的角度描述构件，并根据重要性设置构件的优先级。每个构件的侧面集合被赋予关键词（值）。并从反映构件本质特征的视角（侧面）对构件进行分类，意义相同（近）的若干词汇构成同义词库，所有词汇按照隶属于“面”的概念分组，每个概念用组内的某一同义词汇作为表示载体。3)超文本组织法，它对所有构件都必须辅以详细的功能或行为说明文档，说明中出现的重要概念或构件以网状链接方式相互连接。

对构件库的管理而言，需要一个专门的构件库管理组，负责创建并维护构件库资源。因为构件库的体系结构的变化会影响所有使用库内构件的系统的维护过程。以可重用构件库为基础的工作平台不仅要提供对构件的查询、浏览、选择功能，而且还要提供对构件增加、修改、删除等功能。维护构件库必须确保构件的修改同使用构件的所有应用都是兼容的。此外，对构件库的维护还包括对构件的认证、插入、提取、评价和适应等工作。构件库管理组还在生产开发各种形式的可重用构件方面起着很重要的作用。

2.4 构件的检索

根据构件需求规格说明（如构件应用领域、功能说明、约束条件等），向构件库提交检索请求。从众多的候选者中选出最适合在该软件环境中使用的构件。检索者可在阅读文档的过程中按照联想思维的方式采用相应的检索方法。对构件的检索，从构件的表示出发有人工智能方法、超文本方法和信息科学检索的方法。根据复杂度和检索效果的不同则可分为基于文本的和基于规约的编码检索。信息科学的方法有枚举、侧面、属性值、关键字和正文检索等。在使用自动检索机制的系统中，通过检索规则库和词典（存有同义词，包含关系词、领域关系词等），在推理机制的作用下，根据查询要求进行推理，查出满足条件的目标。检索方式可采用交互式语义网络联想查询、条件查询、专家选择查询等。

检索到的候选构件并不能完全满足需求规格说明，特别是在刚开始用户的需求还不够精确的时候，它可能是一种近似的匹配。因此对检索到的构件还有一个评估的过程，评估该构件在多大程度上符合用户的要求。

2.5 构件的组装

构件的组装必须以构架为蓝图，因此需首先确

定系统的总体构架。实际上可看作是用构件的实例将软件构架具体化的过程。构件实例必须符合系统中其他部分的要求。当从构件库检索到候选构件后，获取候选构件的功能，初步判定候选构件与需求的匹配程度，筛选候选构件集。对初选构件还要在满足约束条件的上下文的环境中测试构件语法/语义的正确性以及一致性。对已匹配（或近似匹配）的构件进行再配置，使其符合当前的应用环境。将符合应用环境的构件组装到系统的构架中，装配成系统的功能模块。一般地要将检索出的原子构件尽量绑定成较大的复合构件，再根据构件的接口进行装配。根据不同软件的开发模式，构件的装配可能是基于手工的，自动的或半自动化的实现。具体装配时，一般要求构件符合 COM（Common Object Model）标准，若是普通构件也要转换成符合 COM 标准的构件，从而使构件在许多编程环境下重用。

组装构件时，有时必须编写粘接代码。这些代码可以通过数据转换等手段来消除构件间接口的不兼容问题，使底层构件的功能按需激发，从而提供把不同的构件结合在一起的功能。粘接代码的应用，还能为系统提供统一的异常处理机制。

构件组装目前仍然缺乏通用工具与方法。实用的构件组装工具应该具有构件的浏览和定制、构件的图形化表示和操纵以及目标系统（半）自动生成等功能。真正意义上的 CBSD 应该在分析、设计、实现和系统部署等不同阶段，在函数、对象、模块、框架、服务进程和程序等不同粒度上，从构件构架的开发、描述、浏览、组装、定制等不同方面对构件组装提供全方位的支持。

2.6 测试与运行

用构件构造系统，要进行系统的测试，以验证构件的装配是否符合系统的设计标准。先排除特定上下文及构件搭配可能出现的错误。一般来讲，构件的开发者努力挖掘构件的可能应用，并开发了许多测试环境。但开发者不可能穷尽所有可能的潜在应用场合。因此构件的用户将它集成到应用系统中时，还要根据应用环境进行大量的测试。经构件组装获得的软件系统至少要经过三种测试：单元测试，测试单个构件；集成测试，测试由构件构成的子系统；系统测试，测试由子系统构成的整个系统。为某个特定系统而开发的构件，当系统的环境

发生变化时,需要重新进行测试。因为新的系统对构件使用的侧重点可能发生了变化,而原来的测试者可能没有重视新的侧重点,没有进行严格的测试。在测试过程中要侧重于分析、定位并排除可能出现的异常。组装成的系统在运行时必须要有有效的控制与协调机制,监测构件的状态与性能,做好构件的故障定位/恢复及通知机制,并作好相应的运行记录。

要注意到系统及其构件的演化将对系统的维护造成的影响。对于一个由构件组成的系统,每个构件都有其自身的升级安排,构件的使用者无法控制构件的升级。若把构件的每个新版本都加入到系统中,其构件的行为、界面、前提、性能及错误的消除都可能发生改变。

3 结论

基于构件的软件开发方法为软件开发开辟了

新的途径,它是解决软件危机的一种很好的方法。近几年对CBSD的研究与实践已经取得了一些成果。但目前软件(构件)重用还没有在软件设计领域得到全面推广,真正的软件工厂还未形成,还存在一些问题,如大的构件库的建设,构件的分类和检索,构架的标准化以及构件接口的标准化,构件的装配与集成,相应的各项标准和工具软件的研制与使用等,使得构件的重用仍处于很低的水平。开发者很少从不同的厂商采购软构件组成系统。现在基于构件的软件开发技术逐渐成熟,上述存在的问题也是目前正在研究和解决的问题。目前CBSD的有关标准也已经出台。如北大西洋公约组织制定的一组软件重用标准,包括“可重用构件开发标准”、“可重用构件库管理标准”、“软件重用过程标准”等。此外,还有对象管理组织(OMG)的CORBA和微软的OLE/COM等。这些新标准和新方法在CBSD中的使用,将会使软件的开发方式出现革命性的变化。

[参 考 文 献]

- [1] Mark R Vidger, John Cdean. An Architecture Approach to Building System from COTS Software Components [EB/OL]. <http://www.sel.lit.nrc.ca/projects/cots>.
- [2] 应时等. 基于构件库及构件组合的软件重用[J]. 计算机工程, 1998, (11).
- [3] 耿刚勇, 仲萃豪. 采用软件构件技术开发领域应用软件[J]. 计算机科学, 1997, 24(1).

The Method of Component—based Software Development

CHEN Jian-xin , LI Ge-xin

(Department of Computer Science, Xiaogan University, Xiaogan, Hubei 432100 China)

Abstract: This paper introduces the related concepts of software development method based on component and the description, classification, storage and retrieve of reusable component. Then their characteristics are analyzed and the procedure of building software system by using reusable component is illuminated. At the same time, this paper discusses the main research emphasis at present and predicts its application prospect.

Key Words: component; software reuse; software development based on component