

# 基于构件的软件配置管理技术研究

张 路,谢 冰,梅 宏,邵维忠,杨芙清

(北京大学计算机科学技术系,北京 100871)

**摘 要:** 本文分析了基于构件的软件开发对配置管理技术的需求,提出了基于构件的软件配置管理系统模型,并介绍了一个基于该模型的系统,讨论了系统中的管理策略和关键技术。

**关键词:** 软件工程;软件配置管理;软件复用;软件构件;版本控制

**中图分类号:** TP311 **文献标识码:** A **文章编号:** 0372-2112 (2001) 02-0266-03

## Study of Component-Based Software Configuration Management Technologies

ZHANG Lu, XIE Bing, MEI Hong, SHAO Wei-zhong, YANG Fu-qing

(Dept. of Computer Science and Technology, Peking University, Beijing 100871, China)

**Abstract:** The requirements for software configuration management (SCM) technologies in component-based software development are studied, and the component-based SCM system model has been proposed, which can satisfy the requirements. In this paper, an SCM system based on this model is introduced, and the management strategies and key techniques of the system are discussed.

**Key words:** software engineering; software configuration management; software reuse; software component; version control

### 1 引言

软件配置管理(简称配置管理)是指一套按规则管理软件开发和软件维护以及其中各种中间软件产品的方法<sup>[1]</sup>,它研究怎样在不同时刻标识软件系统的配置,以便系统化地控制配置的改变,以及在整个软件生命周期内维护配置的完整性和可追踪性<sup>[2]</sup>.软件配置管理主要包括配置识别、变化控制、状态记录报告以及审计四种活动.配置管理系统可以提供版本控制、配置支持、构造支持、审计控制、统计报告、变化控制、过程支持和团队支持八方面的功能<sup>[1]</sup>.

软件复用是解决软件危机,提高软件生产效率和质量的现实可行的途径<sup>[3]</sup>.软件复用是指重复使用“为了复用目的而设计的软件”的过程<sup>[4]</sup>.基于构件的软件开发是软件复用的主要形式.

基于构件的软件开发对配置管理提出了许多新的需求<sup>[5]</sup>:配置管理中应有与基于构件的软件开发一致的构件概念;配置管理应能够有效地维护构件的演化历史,即保存构件的所有版本;配置管理应对构件组装提供支持.

本文提出了基于构件的配置管理系统模型,解决了构件的版本控制、并发控制以及配置管理对构件组装的支持等问题.

### 2 基于构件的软件开发对配置管理技术的需求

基于构件的软件开发体现了软件的构造性和演化性:新系统是通过复用已有构件构造出来的;构件在使用之前需要

根据需求进行修改(演化),已有系统也要根据需求的变化而不断演化成新的系统.

这种开发方式要求:配置管理应支持构件的概念,能够对构件进行管理.已有构件不一定能完全符合用户的要求,需要经过适应性修改后才能使用;有些构件需要从头开发.配置管理应该能够维护构件修改和开发的历史.新系统通过构件集成(组装)得到,配置管理应该能够反映这种系统构造方式,维护构件组装的历史.系统和构件的开发需要多人参与,配置管理应提供并发控制机制,支持多人并行开发.系统和构件要随需求的变化而不断进行演化,有时还会有多个演化方向并存.配置管理应该能够对系统和构件的演化进行变化控制,并维护多个演化方向.

### 3 基于构件的配置管理系统模型

在配置管理技术中引入构件的概念,以构件为版本控制的基本单位,在此基础上定义自包含的配置,支持构件组装,得到了基于构件的配置管理系统模型.

#### 3.1 基本概念

构件:构件是可以被多个软件系统所复用的具有独立功能的系统构成成分<sup>[6]</sup>,这是构件的逻辑概念.在实际的物理存储中,构件表现为一组与一个逻辑概念密切相关的文件.构件的组成文件需要按一定的目录结构进行组织,因此配置管理系统中的构件可以定义为通过目录结构组织起来的一组密切相关文件的集合.这个构件概念支持各种形态的构件.

构件版本和构件版本树:构件在开发或修改过程中可能出现多个版本,这些版本记录了构件开发或修改的历史。构件可能同时存在多个开发流,构件的一个版本分支代表一个开发流,每个分支包含多个版本。因此,构件的所有版本不再表现为版本的序列,而是构成一个树型结构,称为构件版本树。构件的版本控制通过构件的检出和检入操作完成。

构件版本树记录了构件的演化情况。构件的演化包括三个方面:构成构件的文件集合变化、结构变化以及其中某些文件的内容变化。

配置——配置是指一组配置项的集合,其中每个配置项可以是一个构件,也可以是一个配置(作为配置项的配置也称为子配置)。配置具有自包含性。配置可以表示基于构件的软件开发中的组合构件,也可以表示组装出来的系统。

配置的基线(baseline)——为配置及其所有子配置中的构件都选定一个特定版本,就得到了配置的一个基线,该操作称为基线操作。配置的基线表示组合构件或系统的一个版本。

### 3.2 模型概述

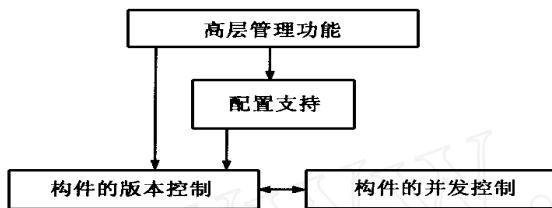


图1 基于构件的配置管理系统模型

如图1所示,基于构件的配置管理系统模型分为四个部分,其中:

构件是版本控制的基础,是检出和检入的基本单位。使用者按照“检出-修改-检入”模式修改构件,并在该构件版本树上增加一个新版本。对于需要多个开发流的构件,利用分支操作可以从构件的任何版本开始,创建一个新的开发流。不同的开发流可以合并。

构件的并发控制解决多人协同开发同一个构件的问题,协调、解决可能出现的并行冲突。

配置支持建立在构件基础上:在构件的基础上定义配置,在已有的构件和配置的基础上定义更大的配置,直至定义出表达整个系统的配置。不断定义配置的过程体现了构件组装的过程。通过基线操作,使用者就可以得到各个组合构件和系统的版本。

高层管理功能建立在构件的版本控制和配置支持的基础上,能够进一步满足基于构件的软件开发中的其它管理需求。

### 3.3 高层管理功能

模型中的高层管理功能包括构造支持、审计控制、统计报告、变化控制、过程控制以及团队支持等。我们研究了基于构件的软件开发对相应高层管理功能的新需求,提出了相应的解决策略。

构造支持是指构件和系统的构造(build)和部署(deploy-ment)。对于构件,只要保留进行构造所需的结构,就可以利用

make文件完成构造;组合构件和系统是用配置表示的,在配置中记录与构造相关的信息,可以通过构件组装工具完成系统构造;在分布式系统中,对于使用各种分布对象技术的软件构件,配置管理系统可以辅助正确地进行构件的分布,称为系统部署。

审计控制分为两个层次:构件层和构件的组成文件层(简称文件层)。在进行审计时,审计人员既可以在构件层进行初步的审计,也可以在文件层进行深入的审计。

统计报告也分为两个层次:构件层和文件层。构件层的报告概括性高,文件层的报告深入细致。

变化控制分为两个层次:构件的变化控制、配置的变化控制。构件的变化控制是和构件的版本控制一起完成的。配置的变化控制要涉及多个构件的检出和检入。配置的变化包含多个构件的变化和多个子配置的变化。

过程控制可用于两个阶段:构件的开发(修改)阶段和构件组装阶段。这两个阶段的过程控制可以用统一的过程模型来描述和实施。

团队支持包括:工作区管理、并行开发管理和远程开发管理。工作区管理既可以保证多个开发人员开发同一构件时不相互干扰,并行开发可分为两个层次:构件间的并行开发和构件内的并行开发。对于远程开发,涉及文件传输的操作只有:构件检出、构件检入和构件更新三种,这三种操作中的文件传输都是不可避免的。

## 4 基于构件的配置管理系统设计与实现

基于以上的模型和策略,我们设计了一个基于构件的配置管理系统——青岛软件配置管理系统JBCM,用以支持基于构件/构架模式的软件开发方法。

### 4.1 系统总体结构

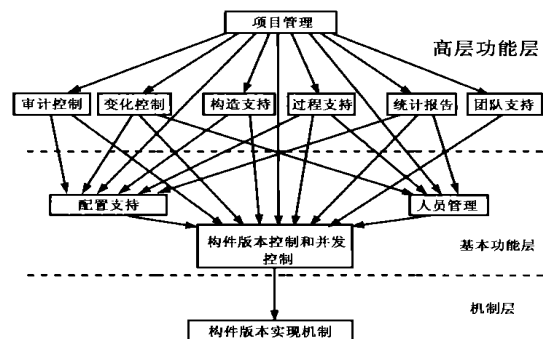


图2 基于构件的配置管理系统JBCM

如图2所示,整个系统可以分为三个层次:机制层、基本功能层和高层功能层。

机制层对用户是透明的,它通过文件版本和控制信息的存储,为构件的版本控制和并发控制提供实现基础。

基本功能层包括:构件版本控制和并发控制、配置支持和人员管理,可以满足软件开发项目组级别的配置管理需求。

高层功能层包括七个部分:项目管理、构造支持、审计控制、统计报告、变化控制、过程支持和团队支持,可以满足软件企业级别的配置管理需求。

## 4.2 管理策略

应用基于构件的配置管理系统,还需要一些策略指导配置管理的实施.主要包括:一致性维护策略和构件组装策略.这些策略通过配置管理计划进行实施.

⑧ 一致性维护策略基于构件的配置管理系统中,要保证构件的一致性和配置的一致性.构件的一致性是指构件概念的一致性;配置的一致性是指其中的构件和子配置可以组装在一起,成为一个构件或系统.

对于传统软件开发方法,使用者可把构件作为一个大粒度的版本控制单位使用,而不要求在概念上构成一个构件.

### ⑧ 构件组装策略

构件组装需要一组已有构件以及一套组装它们的模式<sup>[3]</sup>,其中构件组装模式描述如何将已有构件组装起来,通常可以用软件构架描述语言来描述.在JBCM系统中,配置只描述了一个组合构件由哪些构件组成,因而将构件组装模式包装成一个构件,并作为该配置的特定配置项.这种策略可以适应于用多种构件组装方式.

## 4.3 关键技术

### ⑧ 构件的版本控制

在JBCM系统中,用一个辅助文件表示构件的组成,通过对构件中的各个文件以及辅助文件的版本控制实现构件的版本控制.

### ⑧ 构件版本的增量存储

构件的不同版本间具有较大的内容相关性,构件版本的存储需要使用增量方式,以减少存储冗余.按照上面的实现方式,构件版本的增量存储就可以用文件版本的增量存储来实现.

### ⑧ 构件的并发控制

版本控制的单位提升为构件以后,如果将并发控制的单位也提升为构件,就会造成并发粒度增大,并发度下降.JBCM系统以构件为版本控制单位,以文件为并发控制的单位.

### ⑧ 构件版本的水平合并与垂直合并

水平合并——构件的不同分支代表不同的开发流,有时不同的开发流需要合在一起,称为水平合并.水平合并既包括构件中文件内容的合并,也包括构件结构的合并.

垂直合并——由于对其中任何文件的修改都被视为对构件的修改,构件版本的增长会非常快.为控制构件版本的增长,可以在构件的一个分支上将相邻的若干版本合并为一个版本,称为垂直合并.

## 5 结束语

本文从如何使配置管理更好地支持软件复用出发,在配置管理中引入了与基于构件的软件开发中一致的构件概念,提出了基于构件的配置管理系统模型,从而更好地满足了软

件复用对配置管理的需求.

基于构件的配置管理比传统配置管理具有优越性,主要表现为:构件首先是一个逻辑概念,它有明确的逻辑含义,记录和维护构件的版本更有实际意义;构件的粒度可以比较大,对大粒度的单位进行版本控制更适合现代大规模软件的开发;构件是有结构的,构件版本控制可以满足对文件的版本控制需求;构件组装是基于构件的软件开发的一个重要环节,基于构件的配置可直接支持构件组装.

本文设计并实现的JBCM系统已经在多个应用单位进行了使用.今后,我们将继续完善JBCM系统,并加强对分布协同的软件开发方式的支持.

## 参考文献:

- [1] C. Burrows, G. W. George, S. Dart. Ovum Evaluates: Configuration Management [M]. Ovum Ltd., 1 Mortimer Street, London W1N 7RH.
- [2] E. H. Bersoff. Elements of software configuration management [J]. IEEE Trans. on Software Engineering, 1984, 10(1).
- [3] H. Mili, F. Mili, A. Mili. Reusing software: issues and research directions [J]. IEEE Transaction on Software Engineering, June 1995, 21(6).
- [4] 杨芙清,梅宏,李克勤. 软件复用与软件构件技术 [J]. 电子学报, 1999, 2.
- [5] C. McClure. Software Reuse Techniques: Adding Reuse to the System Development Process [M]. Prentice Hall PTR, New Jersey, 1997.
- [6] 杨芙清. 青岛工程现状与发展·兼论我国软件产业发展途径 [A]. 第六次全国软件工程学术会议论文集 [C]. 杨芙清、何新贵主编,清华大学出版社, 1996. 5.

## 作者简介:



张 路 1973 年出生. 北京大学计算机科学技术系博士研究生. 主要从事软件工程和软件环境、软件复用和软件构件技术等方面的研究.



谢 冰 1970 年出生. 北京大学计算机科学技术系博士后. 1998 年毕业于国防科技大学计算机系获博士学位. 主要研究领域为软件工程、形式化方法、分布式系统等.