基于过程的软件配置管理模型的研究

陈颂梅　夏宽理　刘燕秋
（复旦大学计算机科学与工程系软件工程实验室，上海 200433）
E-mail：chensongmei@sina.com

摘　要　软件配置管理是软件开发过程中的重要支持活动，应随着软件开发方式的发展而变化，并随着软件过程的成熟而不断进步。该文分析了软件过程工程的发展对支持工具提出的新要求，以及现有软件配置管理系统对过程支持的不足，在此基础上提出了基于过程的软件配置管理模型——PBCM，并介绍了该模型的一个实现：WINGCM 软件配置管理系统。

关键词　软件配置管理（SCM）　软件过程　PBCM
文章编号 1002-8331-（2003）15-0091-04　文献标识码 A　中图分类号 TP311

Research on Process-Based Software Configuration Management Model

Chen Songmei　Xia Kuanli　Liu Yanqiu
（Department of Computer Science and Engineering，Fudan University，Shanghai 200433）

Abstract：Software configuration management（SCM）is one of the most important supportive activities in the software development.It should advance with the progress of software development modes and the software process maturity.In this paper，a process-based software configuration management model-PBCM is presented.Based on the model，authors have realized a configuration management system-WINGCM.

Keywords：Software Configuration Management，Software Process，PBCM

1　前言

软件配置管理是贯穿于整个软件过程的软件质量保证活动。它是一种对软件系统的修改进行标识、组织和控制的技术。其作用是协调软件开发，减少由变更而引起的混乱，从而提高软件生产效率，确保软件质量。随着软件开发规模的日益扩大，软件配置管理日益成为软件开发过程中的关键要素。它是 CMM-2 级（可重复级）的一个关键过程域，而 ISO/IEC 12207 国际标准也把它作为软件生命周期过程中基本过程的重要支持过程。

当前，软件过程工程的发展对软件配置管理提出了新的要求。为提高软件配置管理对过程的控制和管理能力，增强其对各种不同开发过程的可适应性，应该将软件过程工程活动作为管理对象引入其中。

1.1　软件过程工程

软件过程工程是为建立软件过程所必须实施的一系列工程化的活动。过程建模、过程例化和过程运作是其中最基本的三项，定义如下：

过程建模：是指为满足某一目标（即指一类特定的软件项目）而对软件过程的结构及其属性进行形式化或半形式化描述的一系列活动。活动得到的通用的、总体的和抽象的软件过程，即为过程模型。

过程例化：是指在过程建模的基础上，针对某一个特定的软件项目对软件过程的结构及其属性进行更充实和具体化的描述，从而生成可运作的软件过程的一系列活动。活动对过程模型进行剪裁，得到一个已被例化的、可执行的软件过程实例。

过程运行：是指实际执行某一个过程实例所进行的一系列活动。

如今，软件企业都以获得 CMM 认证作为提高软件过程能力、软件开发质量的阶梯。而 CMM 度量一个软件组织所达到的软件过程成熟度的依据就是软件过程被明确有效地定义、管理、度量和控制的程度。要达到 CMM-2 级（可重复级），软件组织必须建立基本的稳定的项目开发和维护的过程。要达到 CMM-3 级（已定义级），软件组织必须定义完整的过程模型，由特定的过程小组利用过程模型，进行过程例化活动，从而获得一个针对某一个特定的软件项目的过程实例，并投入过程运作进而开展有效的软件产品工程实践。可见，软件过程工程活动在软件组织中的作用和地位日益重要。

然而就目前而言，软件过程活动在很大程度上依赖于管理的手段，过程建模、过程例化、过程运作等活动仅仅通过制定规约、执行规约的方式来推行，过程模型、过程实例与开发工具不能很好地融合在一起。因此，迫切需要一种实用的支持工具，确保这些过程活动规范地执行，提高软件过程工程的自动化程度。

1.2　软件配置管理的定义

根据 GB/T 11457：1995 软件工程术语给出的定义，软件配置管理是标识和确定系统中配置项的过程，它在系统整个生存周期内控制这些配置项的投放和变更，记录并报告配置的状态和变更要求，验证配置项的完整性和正确性。这里所说的配置项是软件产品在软件生存周期各个阶段所产生的各种形式

基金项目：国家 863 高技术研究发展课题“基于 Internet 以构件库为核心的软件开发平台”（编号：2001AA110241）
作者简介：陈颂梅，硕士研究生，研究方向：软件工程，软件过程，软件配置管理。夏宽理，教授，研究方向：软件工程，程序开发环境，软件重用技术。刘燕秋，硕士研究生，研究方向：软件工程，软件过程，软件配置管理。

和各种版本的软件成分，包括计算机可执行的源代码、目标码、数据库和计算机不可执行的文档、源程序清单、测试案例等。

软件配置管理活动主要包括配置识别，变更控制，状态统计，审计和审查，以及加工，过程管理，小组协作等。这些活动必须依靠强有力的支持工具——软件配置管理系统才能有效地进行。Susan Dart 认为，理想的软件配置管理系统应提供版本控制、配置支持、构造支持、审计控制、统计报告、变更控制、过程支持、团队支持八大功能，尽可能地提高软件配置管理的自动化程度。

1.3 软件配置管理系统的发展现状及不足

软件配置管理提出的二十多年来，研究领域和商业领域的配置管理系统产品或原型不断涌现，它们都在一定程度上部分地实现了 Susan Dart 所提出的八大功能域。例如：W.Tichy 开发的 RCS 能够对存储在库中的源文件进行版本控制；Microsoft 公司的 VSS 是应用于 Windows 平台的基于文件的版本控制系统，能与 VC、VB 等开发环境紧密结合；INTERSOLV 公司的 PVCS 较好地实现了版本管理、建立管理、构造管理、问题追踪等功能；Rational 公司的 ClearCase 与该公司的其它一系列工具紧密集成，以 RUP 过程模型为基础，为软件开发过程提供了良好的支持。

然而，这些软件配置管理系统都是以版本控制和配置支持为核心的，对过程的支持较弱，主要表现在两个方面：

(1)缺乏自动追踪与驱动变更的能力

在软件开发的过程中，会产生大量相互关联的配置项，如需求用例、需求规约、设计文档、测试用例、源程序、目标代码、可执行代码等。这些配置项都是过程运行的产物，有明显的过程属性，它们之间具有前后承接的关联特性。例如：如果需求分析员修改了一个需求用例，那么设计人员应能感知这一更改，修改相应模块的设计方案。

现有的软件配置管理系统尽管以各种方式实现了变更控制的功能，但由于缺乏对配置项之间这种关联的定义，并不具有自动追踪和驱动变更的能力，变更控制没有深入到过程的每一阶段。

(2)缺乏对过程建模、过程例化活动的支持

现有的软件配置管理系统并没有真正地体现出对过程运作的控制和管理，更没有提供对过程建模、过程例化活动的支持。虽然有些配置管理工具（如 Softtools 的 CCC）实现了对生命周期模型的支持，但是它要求用户必须遵循它所制定的生命周期模型。这样，用户的开发过程受到了配置管理工具的制约，阻碍了软件组织对软件过程的改进，不符合软件过程工程的要求。

针对以上这些问题，该文提出了一个基于过程的软件配置管理模型——PBCM。它将软件配置管理建立在过程框架上，全面支持软件过程的各项活动。以这个模型为基础，笔者设计并实现了一个基于过程的软件配置管理系统——WINGCM。

2 基于过程的软件配置管理模型 PBCM

为提高对过程的支持能力，笔者从体系结构和功能两方面对现有的软件配置管理系统进行改进，建立了基于过程的软件配置管理模型——PBCM。该模型分为三层：企业级，项目级，开发者级，分别支持过程建模，过程例化，过程运行这三项基本的软件过程工程活动（图 1）。

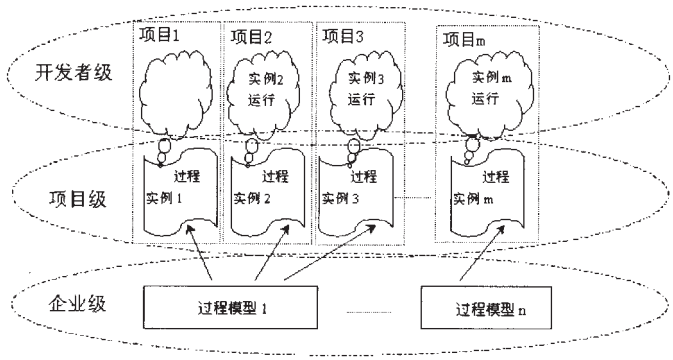

图 1 基于过程的软件配置管理三层模型

2.1 PBCM 模型中与过程相关的基本要素

(1)过程

PBCM 以过程的建立作为配置管理的前提，但它本身不设定任何一种特定的过程，而是由用户通过企业级的过程建模和项目级的过程例化来确定。在项目正式启动后，它对核心的过程活动进行管理和控制，所有的配置管理活动都建立在过程的基础之上，过程的迭代通过变更来实现。

(2)角色

角色是与某一活动相关的权力与责任的一个集合。它是一种对一个过程或一个活动的执行者（实施者）的抽象表示机制，在活动进行时由实施者（即开发人员）来扮演。角色与实施者之间存在着动态映射关系，一个实施者在不同的时间可以扮演不同的角色，而一个角色在不同的时间可由不同的实施者扮演。基于过程的配置管理主要有这样几种角色：系统管理员，项目经理，过程工程师，配置经理，软件工程师，质量保证经理（QA）等。其中，软件工程师是一个笼统的概念，可以细分为需求分析员，设计师，程序员，测试员等角色。PBCM 的角色也不是软件配置管理系统设定的，而是由用户结合过程的定义，通过企业级定义的角色模型和项目级的角色例化来确定的。

(3)权限

权限用于控制软件配置管理系统用户的操作。与过程相关的权限有：建立过程模型，建立角色模型，定义项目过程（过程例化），定义项目角色，为角色分配实施者，定义角色权限，建立配置，确立基线，审核基线，变更审批等。这些权限应授予相关的角色，代表了角色的职责和权力。权限设置也是在企业级的角色建模和项目级的角色例化过程中完成的，用户同样具有极大的自主权。

2.2 各层的主要配置管理活动

2.2.1 企业级

软件工程发展至今，在研究领域和商业领域诞生了许多过程模型。但在实际运用中，任何软件组织都不可能标准地采用某一种模型按部就班，而是根据软件组织的业务类型和范围，以及它自身的规模和组织结构，兼容并包。在 PBCM 中，企业级的配置管理就是支持软件组织以角色为中心，建立适用的过程模型。主要配置管理活动有：

过程建模——定义一个或多个过程模型，作为项目过程的基准。

角色建模——对每个过程模型，定义过程相关的基本角色，并设置他们所拥有的基本权限。

定义和维护企业项目开发者列表——企业项目开发者也

就是软件配置管理系统的用户,一个开发者可能同时参加多个项目,也可能暂时没有参加任何项目,在企业级定义企业项目开发者列表,是为项目的角色例化服务的。

2.2.2 项目级

在开发某个具体项目时,软件组织要根据项目的规模、交付时间、投入的资金,人员配置,软件开发平台、开发工具和技术,以及相关的配置资源等因素,调整并细化各个开发阶段的目标、任务和周期,制定项目特定的软件过程。PBCM 的项目级就是针对某个具体的项目,在项目定义时对过程模型进行例化,得到这个项目的过程实例。主要配置管理活动有:

过程例化——从企业级过程模型中选择一个,根据项目的实际情况做适当的调整和修改,作为项目开发实际遵循的过程。如果在项目进行过程中,发现该过程不适用,在不影响已开发产品的前提下,可以对过程进行动态的调整,软件配置管理系统按照更改后的项目过程继续管理和控制过程的运行,这种动态建模的方式将使过程支持更具灵活性。

角色例化——过程模型已定义了模型所需的基本角色,在项目级可对这些角色及角色的权限进行适当的修改和调整。

人员配置——为每个角色分配开发人员,他们拥有角色定义的权限,承担角色所需完成的任务。开发人员从企业级定义的企业项目开发者列表中选出,当人员发生变动时,要指明接替者,保证角色的完整性。

2.2.3 开发者级

当项目的过程、角色、人员及其权限都确定之后,项目正式启动,开发人员按照既定的项目过程顺序执行或迭代。基于过程的软件配置管理系统在对配置项进行版本管理的时候,确定它们的过程属性,建立配置项之间的联系,跟踪过程运行,确保产品的正确性、完整性和一致性。与过程相关的配置管理主要涉及:

(1)配置项关联

PBCM 将配置项之间的关联分为横向和纵向两大类。横向关联是具有相同过程属性的配置项之间的联系,如源代码的引用。纵向关联是具有不同过程属性的配置项之间的联系,如某个子模块的需求报告(用例)、设计文档、源代码、测试用例、测试报告之间的前后承接关系。

(2)基线

PBCM 将基线视为项目的一个过程进行到成熟阶段时的里程碑,标志着项目可以进入下一个过程。与基线相关的配置管理活动主要有:

确立基线——当一个过程的开发成熟时,确定在这一过程中产出的正确、有效、稳定、一致的配置项的版本,组成基线,作为进一步开发的基础。确立基线时还要指明与之关联的上一过程的基线,基于过程的软件配置管理系统根据基线之间的相互关联,保证产品的正确性、完整性、一致性。

审核和发布基线——基线只有经过审核,确认其正确、有效、一致,才能正式发布。否则将不对其它过程产生任何作用。

基线变更——对到达基线的配置项的更改必须通过一系列严格的变更步骤在软件配置管理系统的变更控制下才能实现。

(3)变更

PBCM 将变更分为过程内变更和基线变更两类,采用不同的策略加以控制和管理。

对尚未到达基线的配置项,变更是自由的,称这类变更为过程内变更。软件配置管理系统根据配置项之间的横向关联,向属于同一过程的相关用户发布过程内变更通知,驱动相关配置项的变更。这些变化对过程外的用户来说是透明的。

对已经到达基线的配置项,变更是受控的,称这类变更为基线变更。变更提出者(客户或开发人员)必须提出变更申请,上报变更审批者,只有审批通过,变更才能正式进入开发过程。当一个过程完成变更后,将产生这一过程新的基线,基于过程的软件配置管理系统根据用户定义的过程,以及配置项之间的纵向关联,向下一过程的相关人员发送基线变更通知,驱动下一过程变更的启动。

3 PBCM 的应用

软件工程发展的这几十年来,各种软件开发模型相继提出。然而,不管是传统的瀑布模型、演化模型、螺旋模型,还是新生代的 PSP、TSP、RUP、XP 等,在实际运用时还是围绕着需求分析、设计、实现、测试这些核心活动展开,只不过在阶段的划分,每一阶段应该达到的目标,过程的周期,是否迭代,迭代的方式,及相关的管理策略上各有不同。

PBCM 正是基于软件过程的这种共性而建立的。它的三层体系结构可以支持各种开发过程,适应不同的开发需要。用户定义的过程、角色、权限不同,则配置管理的实现策略也不同。下面以一个虚拟的软件组织和两个虚拟的项目为例,说明这种模型的灵活性。

该软件组织定义的一个过程模型的核心过程是:需求分析→设计→实现→测试。基本角色有:系统管理员,项目经理,配置经理,需求分析员,设计师,程序员,测试员,QA。

项目一:开发一个管理数据库连接池的 JavaBean。

这是一个微型项目,需求确定,任务明确。项目定义时,对过程模型进行了简化。整个项目的开发只需三个过程:制定目标→实现→测试,三个角色:项目经理,程序员,测试员。其中,项目经理负责制定目标,分配一个程序员和一个测试员,变更审核,发布最终产品。由于每个过程都只有一个人,故基线的审核和发布都由基线确立者自行控制,项目经理对最终产品进行检验。该项目的配置管理执行流程见图 2。

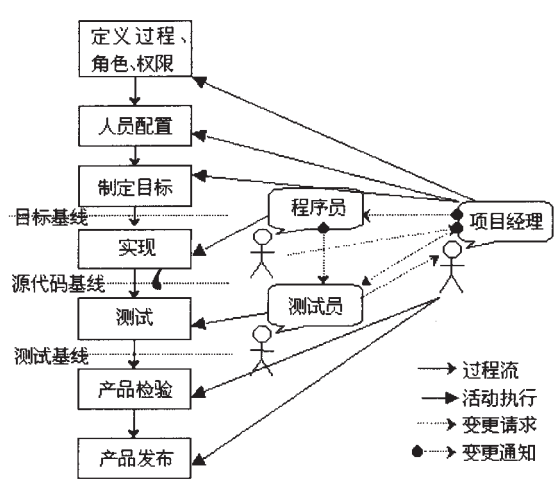

图 2 项目一配置管理执行流程

项目二:开发一个大型的 MIS 系统。

这个项目用户需求复杂、模糊而易变。项目定义时对过程

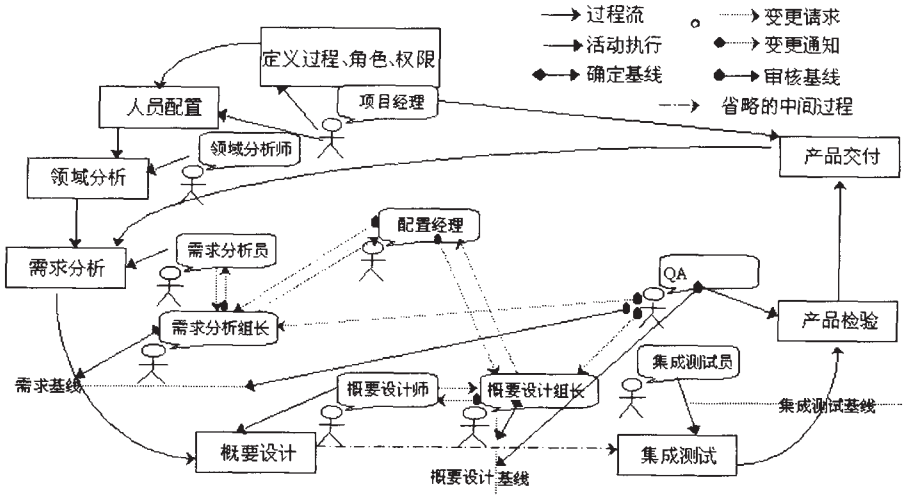

图 3　项目二配置管理执行流程

模型进行了细化。基本过程为：领域分析→需求分析→概要设计→详细设计→编程→功能测试→界面装潢→集成测试，在此基础上反复迭代。其中，界面装潢是在系统的功能实现的基础上，对产品的界面做统一的调整和美化。项目需要的角色有：项目经理，配置经理，领域分析师，需求分析员，设计师，程序员，功能测试员，界面装潢师，集成测试员，QA。由于开发队伍庞大，该项目为每个过程成立一个小组，并增添一个特殊的角色：组长，分别管理各个过程小组。以上这些角色的权限如下：项目经理负责定义项目、项目过程、角色、角色权限和人员，产品交付；配置经理负责管理开发者对配置项的操作权限，变更审批，以及整个系统的构建；每个过程小组的组长负责确立配置和过程基线；质量保证经理负责基线的审批和发布。该项目的配置管理执行流程见图 3。

4　结语

根据上述模型，笔者设计并实现了一个基于过程的可变粒度的软件配置管理系统——WINGCM。它将过程管理作为整个系统的支撑，版本管理，配置和基线的确立，变更控制，小组协作等功能都是建立在过程的基础上。图 4 说明了该系统的功能模型。与传统的软件配置管理系统相比，WINGCM 具有以下特色：

(1)允许用户自定义过程，提供对过程建模和过程例化的支持。

(2)将配置项与产生它们的过程活动相关联，并对它们之间的关系进行建模，在开发过程中确保关系的正确性、完整性和一致性。

(3)对基于构件的开发过程提供了有力的支持。

(4)提供了可变粒度的软件配置管理，提高了管理的灵活性。

(5)统计报告和日志能够提供有关过程运行状况的详尽信息，可作为过程度量的客观依据，进而有助于过程评价和过程改进。

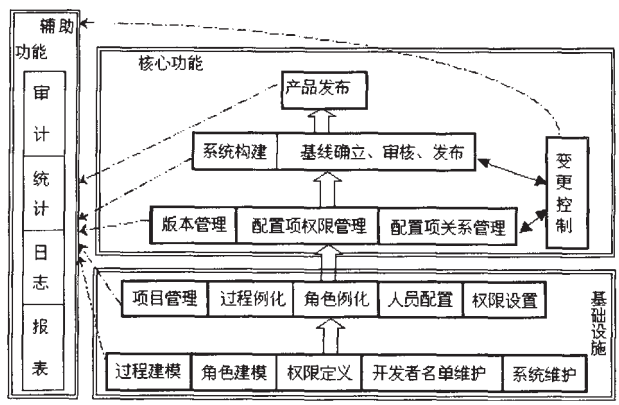

图 4　WINGCM 功能模型

该文提出的基于过程的软件配置管理模型 PBCM 是将软件配置管理向软件过程工程方向扩展的一种有益的尝试。它将各项过程工程活动融入软件配置管理中，能够灵活而有效地支持各种软件开发过程，同时也提高了软件过程管理的自动化程度，为过程改进奠定了良好的基础。笔者将以此为基础，进一步探索和研究面向任务的及支持分布式开发的软件配置管理。

（收稿日期：2002 年 4 月）

参考文献

1.André van der Hoek，Dennis Heimbigner，Alexander L Wolf.Does Configuration Management Research Have a Future?[C].In：proceedings of the 5th International Software Configuration Management Workshop，Seattle，USA，1995-04

2.Susan Dart.Concepts in Configuration Management Systems[C].In：proceedings of the Third International Workshop on Software Configuration Management，ACM SIGSOFT，1991：1~18

3.朱三元，钱乐秋，宿为民.软件工程技术概论[M].科学出版社，2002-01

4.Mark C Paulk，Bill Curtis，Mary Beth Chrissis et al.Capability Maturity Model，Version 1.1[J].IEEE Software，1993，10(4)：18~27

5.Walter Tichy.Software Configuration Management Overview.http：//www.ida.liu.se/~petfr/princprog/cm.pdf