

基于复用的软件生产技术*

杨芙清 王千祥 梅 宏 陈兆良

(北京大学计算机科学技术系, 北京 100871)

摘要 软件复用是提高软件产品质量与软件生产效率的关键技术,在此,重点论述与软件复用及软件构件技术相关的系列问题:描述构件本质特征的构件模型、以领域工程为主要途径的构件获取技术、以构件库为核心的构件管理技术、以应用工程为途径的系统集成组装技术,以及以软件再工程为途径的软件演化技术等.最后,介绍将上述各软件生产技术进行有效集成的软件开发环境:青鸟软件生产线系统.

关键词 软件复用 软件构件 领域工程 构件库 应用工程 再工程

信息技术中,微电子是基础,计算机硬件及通讯设施是载体,而计算机软件是核心.软件是固化的知识,人们抽象的经验、知识正越来越多地以软件的形式予以精确地体现,随着更廉价、更高速微处理器的出现,软件将承担更多的功能,这使得软件变得越来越实在,成为信息时代的新型“物理设施”¹⁾.

尽管当前社会的信息化过程对软件需求的增长非常迅速,但目前软件的开发与生产能力却相对不足,这不仅造成许多急需的软件迟迟不能被开发出来,而且形成了软件脱节现象.自 20 世纪 60 年代人们认识到软件危机、并提出软件工程以来,已经对软件开发问题进行了不懈的研究.近年来人们认识到,要提高软件开发效率,提高软件产品质量,必须采用工程化的开发方法与工业化的生产技术^[1,2].这包括技术与管理两方面的问题:在技术上,应该采用基于复用的软件生产技术;在管理上,应该采用多维的工程管理模式.

本文重点讨论软件工业化生产中的技术问题.文章的第 1 部分对软件构件等概念进行介绍,并提出了青鸟构件模型;第 2 部分以领域工程为核心介绍构件的获取;第 3 部分介绍构件的管理:构件库管理系统;第 4 部分以应用工程为核心介绍构件的复用;第 5 部分以再工程为核心介绍软件演化;最后,文章介绍支持这种工业化生产技术的青鸟工程,主要阐述青鸟软件生产线系统的概念、活动、体系结构以及青鸟软件生产线对软件生产的支撑作用.

1 软件复用与软件构件

近年来人们认识到,要真正解决软件危机,实现软件的工业化生产是惟一可行的途径.分析传统工业及计算机硬件产业成功的模式可以发现,这些工业的发展模式均是符合标准的零部件(构件)生产以及基于标准构件的产品生产(组装),其中,构件是核心和基础,“复用”是必

2000-09-20 收稿

* 国家“九五”科技攻关项目与国家“八六三”高科技资助项目

1) President's information technology advisory committee interim report to the president. PITAC, August 1998. <http://www.itrd.gor/ac>

需的手段. 实践表明, 这种模式是产业工程化、工业化的成功之路, 也将是软件产业发展的必由之路^[1].

软件复用概念提出已久, 并得到了广泛的应用, 例如对子程序的复用、对一般类的复用、对编译器的复用等等. 软件构件概念的提出为软件复用提供了技术基础, 从而使软件复用得到了更为普遍的关注^[3]. 软件产业要发展并形成规模经济, 构件的生产和复用是关键. 本文首先对构件、构件模型、构件实现等一系列重要概念进行简要介绍.

1.1 构件

从一般意义上讲, 构件是系统中可以明确辨识的构成成分, 而软件构件则是软件系统中具有一定意义的相对独立的构成成分. 由于目前讨论的软件构件主要着眼于其复用意义, 因此软件构件主要是指可复用软件构件, 即: 可以被多个软件系统所复用的、具有相对独立功能的系统构成成分. 在后面的讨论中, 除非特别指出, 我们论及的构件皆指可复用软件构件. 本文中, 我们将构件(component)定义为: 构件是可以被复用的软件实体, 由构件规约与构件实现两部分组成. 其中, 构件规约主要由构件模型进行描述.

构架是一类特殊的构件, 它刻画了组成系统的构件和构件之间相互作用的关系.

1.2 构件模型

构件模型是对构件本质特征的抽象描述. 目前, 国际上已经形成了许多构件模型, 这些模型的目标和作用各不相同, 其中, 部分模型属于参考模型, 例如 3C 模型^[4]; 部分模型属于描述模型, 例如: RESOLVE 模型^[5], REBOOT 模型^[6]等; 还有一部分属于实现模型, 典型的如: CORBA (common object request broker architecture) 模型¹⁾, DCOM (distributed component object model) 模型^[7], EJB (enterprise java bean) 模型²⁾, 这些实现模型将构件的接口与实现进行有效的分离, 提供了构件交互的能力, 从而增加了复用的机会, 并适应了目前网络环境下大型软件系统的需要.

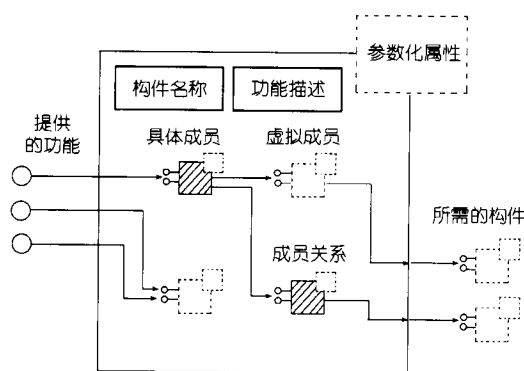


图 1 青鸟构件模型

青鸟构件模型充分吸收上述模型的优点, 并与它们相容. 青鸟构件模型由外部接口与内部结构两部分组成, 见图 1.

(i) 外部接口

构件的外部接口是指构件向其复用者提供的基本信息, 包括: 构件名称、功能描述、对外功能接口、所需的构件、参数化属性.

(ii) 内部结构

构件的内部结构包括两方面内容: 内部成员以及成员之间的关系. 其中, 内部成员包括具体成员与虚拟成员, 而成员关系包括内部成员之间的

互联, 以及内部成员与外部接口之间的互联.

1.3 构件实现

构件实现是指具体实现构件功能的逻辑系统, 通常也称为代码构件. 构件实现由构件生

1) OMG 99, Object Management Group, The Common Object Request Broker: Architecture and Specification, Revision 2.3, Oct, 1999

2) Thomas A. Enterprise JavaBean, Dec. www.psgroup.com, 1997

产者完成,构件复用者则不必关心构件的实现细节. 复用者在复用构件时,可以对其定制,也可以对其特例化.

2 领域工程:构件获取

存在大量可复用的构件是有效地使用复用技术的前提. 通过对可复用信息与领域的分析,可以得到^[8]:

(i) 可复用信息具有领域特定性,即可复用性不是信息的一种孤立的属性,它依赖于特定的问题和特定的问题解决方法. 为此,在识别、获取和表示可复用信息时,应采用面向领域的策略.

(ii) 领域具有内聚性和稳定性,即关于领域的解决方法是充分内聚和充分稳定的. 一个领域的规约和实现知识的内聚性,使得可以通过一组有限的、相对较少的可复用信息来解决大量问题. 领域的稳定性使得获取的信息可以在较长的时间内多次复用.

从以上两方面的认识可以得到这样的结论:通过领域工程得到的面向领域的构件数目大,复用程度高. 不仅如此,通过领域工程得到的面向领域的软件构架比一般意义上的软件构架具体、易操作. 因此,领域工程是获取构件、构架的主要途径.

领域是一组具有相似或相近软件需求的应用系统所覆盖的功能区域,领域工程是为一组相似或相近系统的应用工程建立基本能力和必备基础的过程. 青鸟领域工程方法将领域工程过程划分为领域分析、领域设计、领域实现等多个活动. 其中的活动与结果见图 2.

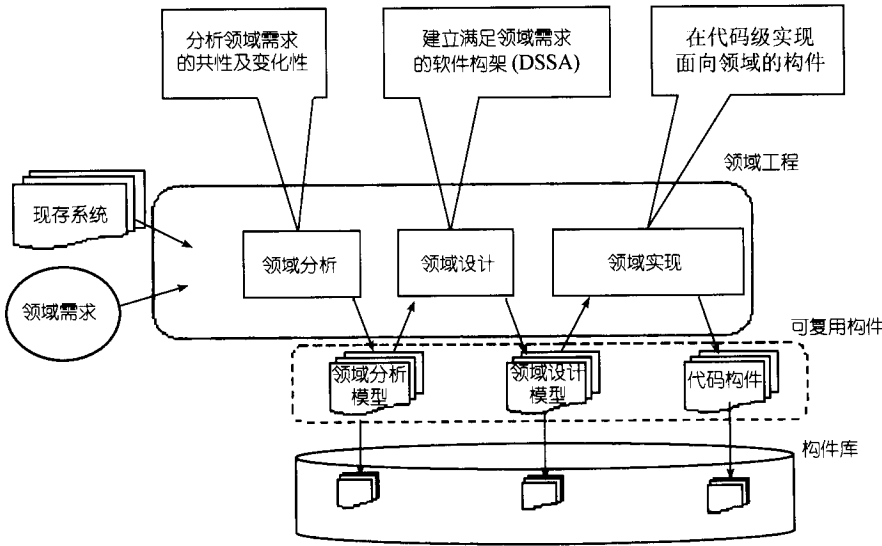


图 2 领域工程中的活动与结果

2.1 领域分析

这个阶段的主要活动是建立领域分析模型,它包含三项活动:建立领域需求定义,建立领域面向对象分析模型和建立领域术语字典. 其中建立领域需求定义又可以划分为确定领域业务模型,确定领域业务过程和确定领域需求三项活动. 建立领域需求定义与建立面向对象分析模型构成领域分析的主线,在这两项活动中,可以根据需要随时在领域术语字典中定义术语.

领域分析过程中必须根据领域的特点,发现并表示出不同的需求,这些需求可以分为如下几类:

必须需求,即领域中所有系统皆必须包含的需求,体现了该领域中系统的本质需求.

可选需求,即领域中部分系统可以包含的需求,体现了领域中系统间的差异.

多选一需求,一组互相之间存在着特定关系的需求,体现了领域中系统间的变化性.

领域分析模型描述领域中系统之间的共同的需求. 领域分析模型主要包括:领域术语字典、领域需求定义、面向对象分析(OOA)模型等.

2.2 领域设计

这个阶段的主要活动是建立领域设计模型,即特定于领域的软件构架(DSSA: domain-specific software architecture)^[9]. 与一般设计类似,领域设计需要考虑若干实现问题,例如操作系统、采用的编程工具、软件分布方式、数据存取方式等等. 除此之外,还需要选取体系结构风格(例如两层的 C/S 方式、B/S 结构、三层结构等),选取构件实现模型(例如 DCOM, CORBA 等).

青鸟领域工程方法中的 DSSA 以面向对象设计模型为主要考虑对象,其中的基本元素是属性、服务、类、对象及其相互关系. 类、类的属性、方法、各种关系都可能是可选的或多选一的,这些元素之间也存在着依赖、互斥等关系. 在 DSSA 与领域分析模型间存在着可追踪性,领域分析模型中的必须需求要追踪到 DSSA 中的必须元素,可选需求要追踪到 DSSA 中可选元素,多选一需求追踪到 DSSA 中多选一元素.

2.3 领域实现

该阶段的主要目标是依据领域分析模型和 DSSA 开发领域特定的构件与构架,它们可以从现有系统中提取得到,也可以通过重新开发而得到. 在可复用构件与 DSSA 之间需要建立可追踪性,以将可复用构件与其规约联系起来. 从方便使用角度,应尽量提供黑盒式复用方式.

领域实现的另一个重要内容是构件测试. 构件的质量关系到未来多个应用系统的性能,因而显得格外重要.

最后需要指出的是,在上述三个阶段之间以及各个阶段内部,存在大量的迭代工作.

3 构件管理

对大量的构件进行有效的管理,以方便构件的存储、检索和提取,是成功复用构件的必要保证¹⁾. 构件管理的研究内容包括:构件描述、构件分类、构件库组织、人员及权限管理和用户意见反馈等.

3.1 构件描述

构件模型是对构件本质的抽象描述,主要是为构件的制作与构件的复用提供依据;从管理角度出发,也需要对构件进行描述,例如:实现方式、实现体、注释、生产者、生产日期、大小、价格、版本和关联构件等信息,它们与构件模型共同组成了对构件的完整描述.

3.2 构件分类

青鸟构件库采用刻画方法对构件进行分类,这些刻画包括:

1) NATO. NATO Standard for Management of a Reusable Software Component Library, Vol 2. NATO contact number: CO-5957-ADA, 1991

- (i) 使用环境,使用(包括理解/组装/修改)该构件时必须提供的硬件和软件平台。
- (ii) 应用领域,构件原来或可能被使用到的应用领域(及其子领域)的名称。
- (iii) 功能,在原有或可能的软件系统所提供的软件功能集。
- (iv) 层次,构件相对于软件开发过程阶段的抽象层次,如分析、设计、编码等。
- (v) 表示方法,用来描述构件内容的语言形式或媒体,如源代码构件所用的编程语言等。

3.3 构件库的组织

为了给使用者在查询构件时提供方便,同时也为了更好地复用构件,构件库为构件实体间建立了多种关系,这些关系包括:

- (i) 精化关系,主要是指处于软件生命周期相邻阶段(相邻的抽象层次)的构件间的对应关系。
- (ii) 版本关系,主要是指处于同一个构件演化系列中的构件间的关系。给定一个构件,通过版本关系可以找到它的所有版本。
- (iii) 协作关系,与一个构件具有协作关系的构件主要是指与该构件相互合作、共同完成一个特定任务的构件。
- (iv) 包含关系,不同形态的构件间的包含关系。例如,一个类树构件包含了一个或多个类构件,一个框架构件也包含了一个或多个类构件。
- (v) 继承关系,这是一个一对一的关系。在构件库中,构件之间允许有继承关系,但不鼓励使用复杂的继承层次和多继承。

3.4 人员及权限管理

构件库系统是一个开放的公共构件共享机制,任何使用者都可以通过网络访问构件库,这在为使用者带来便利的同时,也给系统的安全性带来了一定的风险,因此有必要对不同使用者的访问权限作出适当的限制,以保证数据安全。

构件库系统包括5类用户,即注册用户、公共用户、构件提交者、一般系统管理员和超级系统管理员。他们对构件库分别有不同的职责和权限,这些人员相互协作,共同维护着构件库系统的正常运作。同时,系统为每一种操作定义一个权限,包括提交构件、管理构件、查询构件及下载构件。每一用户可被赋予一项或多项操作权限,这些操作权限组合形成该人员的权限,从而支持对操作的分工,为权限分配提供了灵活性。

3.5 用户意见反馈

构件库系统的用户意见反馈部分为复用者理解和选取构件以及构件库各个层次的管理人员管理和改进构件库系统提供辅助决策支持。其主要特点是基于数据仓库技术,采用多维数据模型,支持面向主题的、集成的、带有时间属性、包含历史数据的数据组织方式。主题中包含多个事实表、维表以及不同层次的汇总表,为多维分析提供数据存储和组织的基础。构件库系统的反馈库对构件使用反馈信息进行多维的分析和度量,分别从构件文档、构件组装、构件测试和构件应用4个角度综合度量出构件的可复用性,以协助用户更好地识别和选取构件。青鸟构件库管理系统的体系结构见图3。

4 应用工程:构件复用

利用领域工程结果进行软件系统开发的过程通常称为应用工程。与一般的软件开发过程

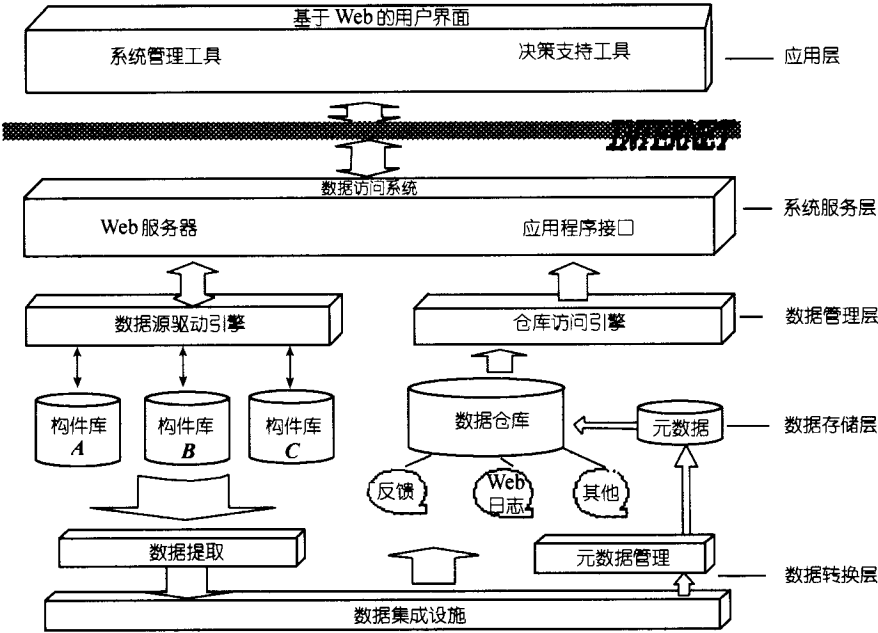


图 3 青鸟构件库管理系统的体系结构

类似,应用工程也可以划分为分析(需求获取)、设计、实现等阶段.不同的是,本过程中的每一阶段皆可以从构件库中获得可复用的领域工程结果,并将其作为本阶段集成与开发的基础^[10].

青鸟应用工程中的活动与结果见图 4.

(1) 系统分析(捕获需求)

本阶段的目标在于:根据领域工程获得的领域分析模型,对照用户需求,确认领域分析模型中的变化性需求,或者提出新需求,以获得具体系统的分析模型.其中包括确定具体的业务

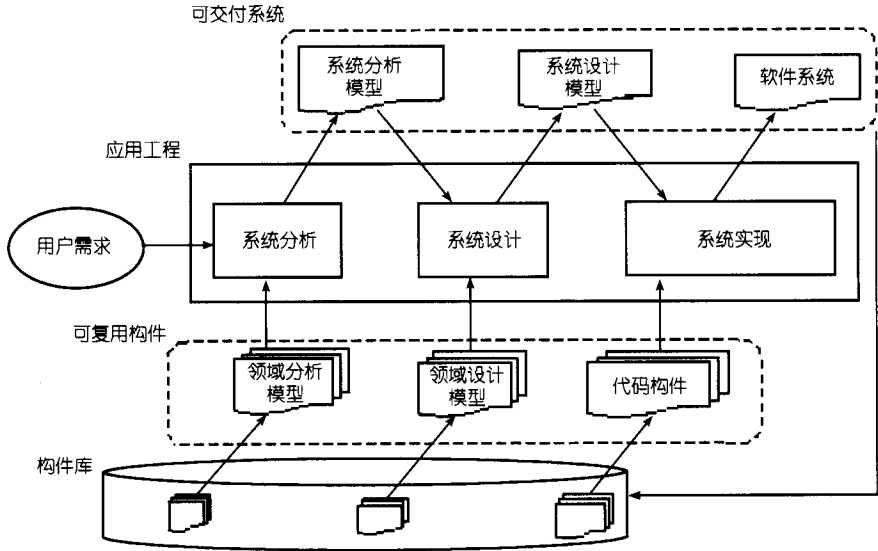


图 4 应用工程中的活动与结果

模型、固定领域分析模型中的变化性,调整领域需求模型等活动。最终用户的持续参与是获得良好模型的重要因素。

(ii) 系统设计

系统设计的目标在于:以领域工程获得的 DSSA 为基础,对照具体系统的分析模型,获取具体系统的设计模型。其核心环节是根据系统的需求模型,固定 DSSA 中相关的变化成分。针对用户提出的新需求,本阶段应当为其设计相应的模型。另外,如果对领域的知识有所增加,则可能需要对 DSSA 进行一定的调整。

(iii) 实现与测试

以领域构架/构件为基础,对照具体系统的设计模型,将构件与构架进行集成组装,并进行必要的代码编写工作,以实现并测试最终的应用系统。

应用工程在开发阶段将领域需求的变化性进行固定。对于一个十分成熟的领域,对需求变化性的固定更适于在后期实现,以满足不同用户的需求,这些后期的阶段包括安装、启动、运行等,其中安装阶段通过系统剪裁实现变化性,启动阶段通过参数控制实现,运行阶段通过动态配置实现。

5 再工程:软件演化

随着软件技术的发展,以及应用需求的变化,软件必须持续不断地演化,这是软件的本质特性之一。但当前采用的都是重新开发,既费时又不经济,而且把已有的软件看成包袱。实际上,任何事物均包含着再工程的过程。采用构件技术有效地适应了软件的演化特征,将“遗产软件”“变废为宝”,既加快和规范了开发过程,又积累了软件资产。

软件再工程是一个工程过程,它将逆向工程、重构和正向工程组合起来,将现存系统重新构造为新的形式^[11]。再工程的基础是系统理解,包括对运行系统、源代码、设计、分析、文档等的全面理解。但在很多情况下,由于各类文档的丢失,只能对源代码进行理解,即程序理解。

青鸟软件再工程中的主要活动如图 5 所示。

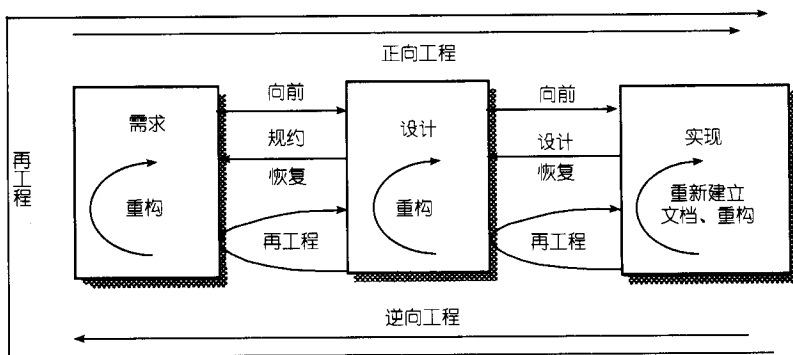


图 5 软件再工程

6 青鸟工程

青鸟工程^[2]是研究、解决软件开发与生产过程中所面临的生产手段问题的国家项目,始于“六五”末期,已经历“七五”、“八五”、“九五”一期,目前正在进行“九五”二期工程。青鸟工程的

目标是:以实用的软件工程技术为依托,推行软件工业化生产技术和模式,为软件企业提供必要的工业化生产手段和装备,建立软件产业基础,并形成规模经济所需的人才储备、技术储备和产品储备。

青鸟工程提出的软件生产线的概念和思想,将软件的生产组织划分为 3 类不同的生产车间,即应用构架生产车间、构件生产车间、基于构件及构架复用的应用集成(组装)车间,在这三个车间之间存在着两个库,即应用构架库和构件库,从而形成软件生产组织内部的合理分工,实现软件的工业化生产。标准规范和质量保证对整个生产过程提供支持。青鸟软件生产线的概念图见图 6。与上述概念图相对应的青鸟软件生产线系统中的活动见图 7。

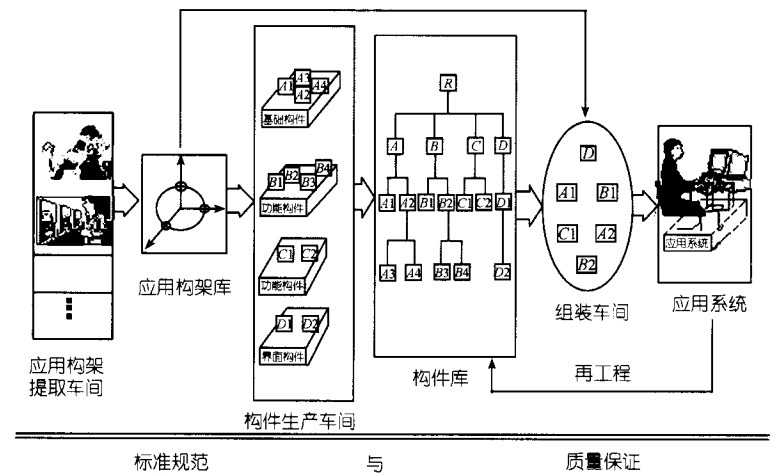


图 6 青鸟软件生产线系统的概念图

从图 6 和 7 中可以看出,青鸟软件生产线系统以软件构件、构架技术为核心,其中的主要活动体现在传统的领域工程与应用工程之中,但赋予了它们新的内容,并且通过构件管理、再工程等环节将它们有机地衔接起来。另外,青鸟软件生产线系统的每个活动皆有相应的方法和工具与之对应,并结合项目管理、组织管理、质量管理等管理问题,形成完整的软件生产流程。

软件构件技术的突破和发展,使软件产业和传统产业一样,划分为软件构件业、系统集成组装业和信息服务业 3 大类,为软件产业的规模化发展提供了合理的结构模式。青鸟软件生产线将促进软件产业基础设施的建设,并为软件企业的能力提升提供直接支持。

7 结束语

本文介绍了我们在基于复用的软件生产技术方面的研究,围绕构件这一概念,对构件的获取、管理、复用等问题进行了深入探讨。基于软件复用与软件构件技术的软件工程方法是目前软件领域的研究与实践热点,文献[3,10]等皆有论述,与这些方法相比,青鸟工程提出的青鸟软件生产线系统更关注其中的生产过程,以及生产线系统的完备性,其目的在于提高软件生产线系统的可操作性与实用性。

除本文介绍的技术内容外,近年来人们认识到:管理是软件工程中的重要问题,是发挥先

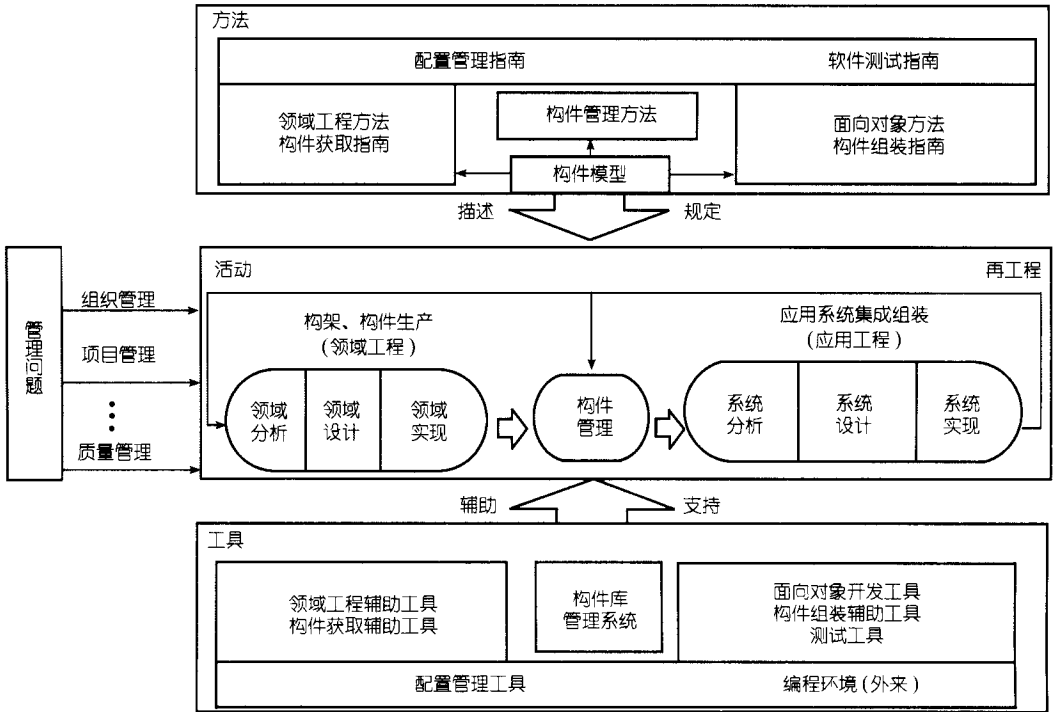


图 7 青鸟软件生产线系统中的活动

进技术的制约因素。因此,我们后续的工作除继续在技术方面进行深入研究外,还将结合企业资质认证体系等对管理问题进行深入研究,并继续与软件企业密切合作,结合我国软件企业的特点,探索符合中国软件企业特点的软件生产方法,并开发出相应的支持工具。

参 考 文 献

- 1 杨美清,梅 宏,李克勤. 软件复用与软件构件技术. 电子学报, 1999, 27(2): 68 ~ 75
- 2 杨美清,邵维忠,梅 宏. 面向对象的 CASE 环境 JBII 型系统的设计和实现. 中国科学, A 辑, 1995, 25(5): 533 ~ 542
- 3 Mili H, Mili F, Mili A. Reusing software: issues and research directions. IEEE Transactions on Software Engineering, 1995, 21(6): 528 ~ 562
- 4 Tracz W. Implementation working group summary. In: Baldo J, ed. Reuse in Practice Workshop, Pittsburgh, Pennsylvania, July 1989. 256 ~ 261
- 5 Bucci P, Edwards S H. Special feature: component-based software using RESOLVE, ACM SIGSOFT. Software Engineering Notes, 1994, 19(4): 21 ~ 67
- 6 Guttorm S. The REBOOT approach to software reuse. System Software, 1995, 30: 201 ~ 212
- 7 Rogerson D. Inside COM. Microsoft Press, 1997
- 8 Arango G, Prieto-Diaz R. Domain analysis concepts and research directions. In: Prieto-Diaz R, Arango G, eds. Domain Analysis and Software System Modeling. Los Alamitos: IEEE Computer Society Press, 1991. 9 ~ 32
- 9 Tracz W. Confessions of a Used Program Salesman-Institutionalizing Software Reuse. New York: Addison-Wesley Publishing Co, 1995
- 10 Jacobson I, Griss M, Jonsson P. Software Reuse: Architecture, Process, and Organization for Business Success. ACM Press, 1997
- 11 Feiler P H. Reengineering: An Engineering problem, CMU/SEI-93-SR-5, ADA267117. Pittsburgh: Software Engineering Center, CMU, 1993