

产品线工程理论应用与研究

王超轶, 葛斌, 茅洪波

(大连大学 辽宁省智能信息处理重点实验室, 辽宁 大连 116622)

摘要:产品线工程(Product Line Engineering)是运用软件重用思想的一种软件开发方法,它属于一种大粒度的软件复用方式,用于开发相同领域的类似产品过程。由于PLE强调针对特定产品集合的领域开发和面向重用技术,所以能有效地提高软件产品的开发效率和质量,降低软件开发成本。文中介绍了PLE理论,着重研究产品线模型、开发模式、生命周期和体系结构,介绍领域工程和应用工程在生命周期内部的交互作用。通过产品线开发周期分析和比较出PLE在系统开发过程中的特点,最后介绍PLE在HIS系统中的开发应用。

关键词:产品线工程;软件复用;软件工程

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2008)08-0206-05

Product Line Engineering Theoretical Research and Application

WANG Chao-yi, GE Bin, MAO Hong-bo

(Liaoning Key Lab. of Intelligent Information Processing, Dalian University, Dalian 116622, China)

Abstract:Product Line Engineering (PLE) is a specialized form of software reuse that promises to improve software productivity and quality, reduce software development costs and short time to market. It is comprehensively presentation product line engineering theory in this thesis. The study on some key techniques to support software reuse in product line architecture introduced in this thesis simplifies the application of product line and improves the efficiency, expendability and flexibility of domain-specific software reuse. At the end of this thesis used PLE to develop home integration system.

Key words:product line engineering; software reuse; software engineering

0 引言

由于软件的复杂性和软件规模的不断增大,伴随而来的是软件开发难度的增加、周期和成本都在不断提高,如何能在最短的时间内开发出高质量的产品,成为软件工程研究的焦点。目前国内外专家们纷纷提出软件重用的开发思想,提高软件生产效率,降低开发成本。产品线工程(Product Line Engineering)则是运用软件重用思想的一种软件开发方法,由于PLE强调针对特定产品集合的领域开发和面向重用技术,可以有效地提高生产率和质量,缩短产品上市时间,所以被越来越多的开发组织所接受,并应用到实际的软件开发中去^[1]。

文中全面介绍了产品线工程理论,着重研究了产品线模型、体系结构和基于复用开发过程,最后用HIS系统举例说明。

1 软件产品线概述

1.1 软件产品线基本概念

软件产品线(Software Product Line, SPL)是指一系列处于特定领域、功能相似的软件产品集合^[2],这些产品共享一个公共的、可管理的特征集,这个特征集能满足选定的市场或任务领域的特定需求。软件产品线是多个软件密集系统组成的集合,这些系统共享一个公共的、可管理的特征的集,这些系统遵循一个预描述的方式,是在公共的核心资源(core asset)基础上开发的^[3]。软件产品线是专业软件开发组织的软件开发方法,能有效地提高软件生产率和质量、缩短开发时间、降低总开发成本。对于软件开发,可以把“软件产品”定义为:“指定交付给顾客或最终用户的一个完整的计算机程序、规程以及相关文档和数据的集合,或该集合中的任何个别项”。比软件产品颗粒度更小的软件形态可以是:软件模块(Software Modules,例如子产品、中间件、构件、服务等)、软件功能(Software Functions)和软件特性(Software Features)等,它们有机地整合在一起,构成了软件产品。

收稿日期:2007-11-01

基金项目:国家自然科学基金(200507)

作者简介:王超轶(1982-),男,安徽芜湖人,硕士研究生,研究方向为软件产品线工程;葛斌,教授,研究方向为自动化过程控制。

1.2 产品线模型

最初的产品线开发过程采用双生命周期的模型(如图1所示),分为领域工程和应用工程两个重叠周期,两个周期内部相互配合。两个周期内部都分为分析、设计和实现三个阶段。领域工程周期的主要任务包括:

(1) 领域分析:构建领域模型,分析产品线中的家族产品中的共性与个性。

(2) 领域设计:通过领域模型的分析,设计产品线的体系结构。

(3) 领域实现:遵照产品线体系结构,开发领域可重用资源。

应用工程在领域工程的基础上,结合应用的需求构造新产品。应用工程经过以下三个阶段:

●需求分析:将系统需求与领域需求进行比较,划分领域共同需求和个性需求两部分,得出系统说明。

●系统设计:基于领域体系结构结合系统独特需求,设计应用的软件体系结构。

●系统实现:遵照应用体系结构,用领域可重用资源实现领域公共需求,用定制开发的组件满足系统独特需求,构建新的系统。

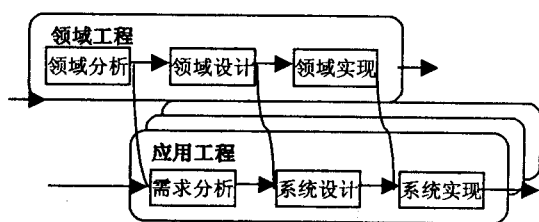


图1 产品线双生命周期

2 PLE 与软件复用

2.1 关于软件复用

通常情况下软件开发都包括:需求分析、设计、编码、测试、维护这几个阶段,无论是采用传统软件工程的瀑布模型,还是现在使用比较多的原型方法和螺旋模型,主要阶段都是遵循上述过程。然而随着软件规模的不断增大,软件开发的复杂性也在提高,如果按照传统软件开发步骤,开发每一个系统都从头开始,必将存在大量的重复劳动,如:重复的需求分析,重复编码,测试等等。软件复用技术就是避免开发过程中的重复劳动。充分利用已有应用系统开发的经验和资源,在已有系统的基础上,通过软件复用可以充分地利用已有的开发成果,消除了包括分析、设计、编码、测试等在内的许多重复劳动,从而提高了软件生产率,同时,通过复用高质量的已有成果,避免了重新开发可能引入的错误和不当,从而提高了软件的质量^[4]。

2.2 PLE 中的复用

软件开发方法随着软件工程的不断深入研究而得到长足改进。很早以前,软件开发工程师把每个新项目看作是一个孤立系统,从头开始进行开发,很少与其它系统进行技术互访,抛弃以前类似系统的开发经验和开发类似软件系统所产生的资源集(例如:模块、构件等等)。与此不同的是,PLE 开发过程则注重先前每一次相关系统开发的环节,充分利用积累的经验知识和资源集。对多个系统之间提取它们所包含的共同特征集,对相异的地方单独设计开发,此外,已经存在的资源集,必须再工程(Reengineered)才能进行复用,而且,新组件也必须采用可复用的思想进行开发。

产品线工程(PLE)是一种特殊形式的复用,用于在开发相同领域类似产品的过程中,由于产品线工程强调针对特定产品族集合的领域分析和面向对象重用的开发。它是针对产品族的规模化开发,因此产品线工程具有集成化的开发生命周期,其中应用工程和领域工程交互在特定领域的一组产品牵引下进行。PLE 的目标,就是通过合成公共领域产品生产相同领域中的类似产品。产品线精简整合领域工程和应用工程阶段实现软件的大粒度复用。

假如某一个系统产生变化,将要投入巨大的精力去理解新的需求,去重新设计、编码和重新测试。PLE 的架构起初设计就充分考虑到这一点,它支持系统中发生的改变,所以,花费在理解产品族的重新需求,重新设计、编码、测试的代价将控制在最小程度。人们知道,某个组件优化后可复用在其它特定产品或多个产品中,但几乎不可能复用在不同的应用环境下。因此在 PLE 的家族产品开发中应确定复用应用领域的范围。PLE 在领域范围内生产可复用的组件并应用到具体的产品中。采用 PLE 技术,可以降低软件开发成本与周期。

3 PLE 生命周期

3.1 领域工程与应用工程

为了识别、开发存在特定领域中的软件产品的公共资产,产品线工程涉及到选取、精化和构建等技术实践。PLE 从生命周期上可分为应用工程和领域工程两个部分(如图2所示)。

应用工程包括:应用需求分析,参考体系结构,领域构件定制,基于构件开发。

领域工程包括:领域分析,领域定义,构建领域体系结构,构建领域组件。

●领域分析:这个阶段的主要目标是获得领域模型(domain model)。领域模型描述领域中系统之间的

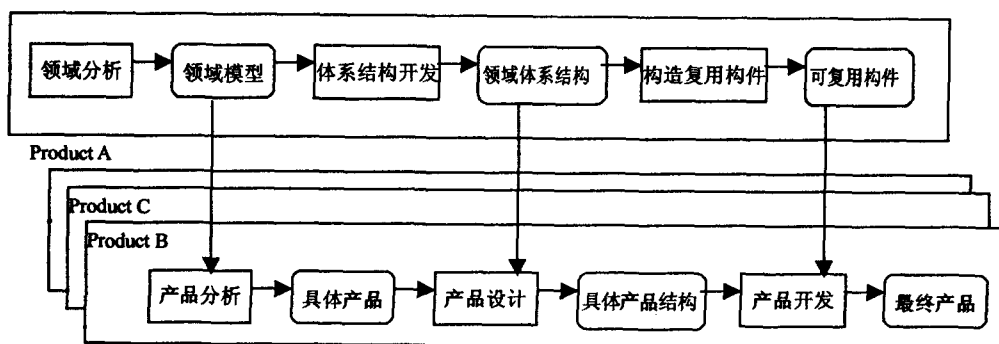


图 2 PLE 生命周期

共同需求。这个阶段的主要活动包括确定领域边界,识别信息源,分析领域中系统的需求,确定哪些需求是被领域中的系统广泛共享的,哪些是可变的。

● **领域设计:**这个阶段主要获得领域构架(Domain-specific software architecture,简称 DSSA)。DSSA 描述在领域模型中表示的需求解决方案,它不是单个系统的表示,而是能够适应领域中多个系统的需求的一个高层次的设计。建立了领域模型之后,就可以派生出满足这些被建模的领域需求的 DSSA。由于领域模型中的领域需求具有一定的变化性,DSSA 也要相应地具有变化性。

● **领域实现:**这个阶段主要行为是定义将需求翻译到由可复用构件创建的系统的机制。根据所采用的复用策略和领域的成熟及稳定程度,这种机制可能是一组与领域模型和 DSSA 相联系的复用构件,也可能是应用系统的生成器。

领域工程关注领域分析、领域定义、领域框架的开发和核心组件资产的开发,其交付物为领域模型、抽象的低级框架和最后演进出的具体的高级框架;应用工程关注具体应用需求的分析、产品线框架实例化、核心组件资产的复用等,其交付物为产品规约、特定应用的框架、复用记录和产品。领域工程指导应用工程开发具体应用。在产品线早期阶段,应用工程主要以抽象的低级框架为基础,按照框架的描述和约束,选取或新建组件,构造特定应用的框架。随着应用工程的多次实施,框架的复用记录越来越多。根据这些复用记录,抽象框架逐渐演进成为高级框架。然后,在新的产品线实例开发当中,这个高级框架将取代低级框架,从而提高开发效率。

体系结构在产品线生命周期处于重要地位,领域工程师通过对领域模型的分析来确定相应的参考体系结构。领域模型是对不同领域分析后得到的结果。在应用工程,根据参考体系结构开发其它产品体系结构。体系结构在 PLE 的早期阶段就被确定,它对产品族的质量有着重要的影响。如果软件体系结构开发不够细

致,很可能不能满足系统的需求和在产品族中缺乏灵活的能力。因此,体系结构分析与评价对 PLE 至关重要。

3.2 PLE 开发特点

由 PLE 的开发周期,可以概括产品线最大的特点是:领域工程与应用工程的相互作用,尽可能多地复用领域内的资源和组件。可以看出它与一般的软件开发过程有着明显的区别,主要体现在以下几个方面:

● **基于体系结构:**体系结构在产品线开发过程的初期阶段就被确定,领域工程师致力于在多个产品中构建参考体系结构(Reference Architecture)。

● **复用驱动:**PLE 的关键过程在于复用领域工程组件。这些组件在产品族中具有相同特性。因此复用技术在产品线中是一项常见的操作而不是偶然现象^[5]。

● **过程驱动:**特定的产品需求驱动产品线的开发过程,开发过程由工具集提供分析和开发。

● **成本驱动:**由于 PLE 开发某些特定领域内具有相似体系结构的产品族,因此投入在开发的成本被产品家族所分摊,大大降低单个产品所投入的成本。PLE 还根据市场的需求,尽可能缩短产品的开发周期和减少开发成本。

● **具有特定领域:**产品族是在某一特定领域范围内进行开发。PLE 的领域不是属于单一产品,而是应用于整个产品家族。因此,高度相关的产品集合推动领域工程的发展。

● **生产者-消费者关系:**PLE 开发周期包括两个阶段:首先,在领域工程阶段开发软件产品族所需的公有资源、组件,这一阶段可以被认为是生产者。然后,在应用工程阶段使用已有资源、组件去开发相应产品,可以被看作是消费者。这种相互紧密地关系最大化地提高了产品线的开发效率。

4 软件体系结构与 PLE 体系结构

软件体系结构(又称软件构架)是对系统整体结构

设计的刻画,包括全局组织与控制结构,构件间通信,同步和数据访问的协议,设计元素间的功能分配,物理分布,设计元素集成,伸缩性和性能,设计选择等。软件构架为 CBD 提供了构件组装的基础和上下文。一个典型的软件构架是由系统中的构件、连接器和约束构成的配置格局。

基于体系结构的软件开发注重体系结构的内部元素,例如构件、连接器被看作为高层结构。连接器能像过程一样被调用。同时,它也可以被复杂 C-S 协议用于连接分布式数据库和中间件。

在 PLE 中软件体系结构占有举足轻重的地位,规范不同系统的高层共性,保证灵活和正确的系统设计,对系统的整体结构和全局属性进行规约、分析、验证和管理。将构架作为系统构造和演化的基础,可以实现大规模、系统化的软件复用。然而,一般软件体系结构与产品线体系结构(Product Line Architecture, PLA)的区别在于:PLE 的领域构架用于产品线中的所有产品族,所以这种体系结构必须健壮,灵活,高度可定制化服务于大量产品所需的核心体系结构^[6]。PLA 提供产品开发所需的框架和根据开发环境重用和开发可复用组件。通常 PLA 建立难度远远大于一般软件体系结构的开发。这是由于在 PLA 中,不仅要满足产品族所有产品的共性需求而且还要符合个性化需求。因此,在 PLA 中需要有客户的积极参与进行需求配置和个性化管理。图 3 为 PLE 体系结构。

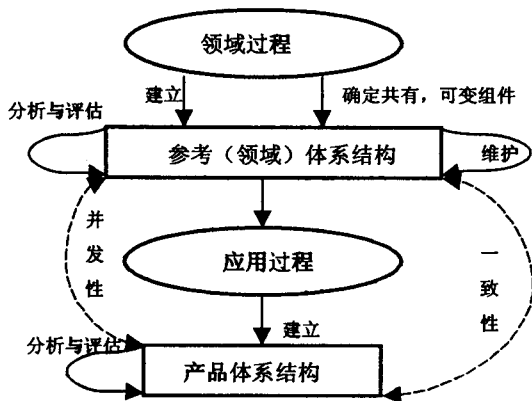


图 3 PLE 体系结构

研究体系结构对于进行高效的软件工程具有非常重要的意义:通过体系结构的研究,有利于发现不同系统在较高级别上的共同特性;获得正确的构架对于进行正确的系统设计非常关键;对各种软件构架的深入了解,使得软件工程师可以根据一些原则在不同的软件构架间作出选择;从构架的层次上表述系统,有利于进行系统较高级别性质的描述和分析。特别重要的是,在基于复用的软件开发中,为复用而开发的软件构架可以作为一种大粒度的、抽象级别较高的软件构件

进行复用,而且软件构架还为构件的组装提供了基础和上下文,对于成功的复用具有非常重要的意义。

5 应用举例

下面以家居集成系统(Home Integration Systems, 简称 HIS)为例介绍 PLE 理论在实践中开发的意义。首先介绍家居集成系统:就是能把住户家庭里的电器设备集成在一个系统下统一管理、使用和控制。例如: HIS 系统可以根据室内外温度来自动控制室内的温度,当发现不明外来人员入侵,发生火灾或水灾时会自动报警。能控制家庭里的娱乐电器:电视机,收音机,家庭影院,计算机,家庭主要电器等等^[7]。

按照 PLE 的双生命周期模型,在领域工程阶段,得出 HIS 系统必须具有如下典型特征:

- 系统可升级:能够根据主人的需要增减电器设备,系统具有可扩展性,可以应用到办公室、写字楼等其它环境。
- 稳定性:系统应具有较高的稳定性能:当终端设备增加或减少,某一设备发生故障,新设备接入等情况下, HIS 系统都应能稳定工作,不受外界干扰。
- 兼容性:不同类型、型号的电器设备接入后,系统应能仍然保持正常工作。
- 安全性:系统提供多用户本地、远程控制,能够防止黑客非法入侵系统。
- 易操作性:系统应具有简单操作界面,即用户不需要接受特别培训,即能熟练操作使用。

对 HIS 分析,得到系统的体系结构需要通过网络或无线网络来连接每个设备,集成在统一管理控制下,这就需要对不同类型的设备进行访问控制的协议标准。开放系统网关控制协议(Open Systems Gateway Initiative, 简称 OSGI)符合 HIS 系统的需要,支持对不同设备进行控制,OSGI 协议的提出解决了 HIS 系统控制的核心问题,为用户与终端设备架起服务的桥梁。在 OSGI 标准提供了一个最全面的解决方案,整合沟通,从用户需求到服务的提供。如图 4 所示:在 OSGI

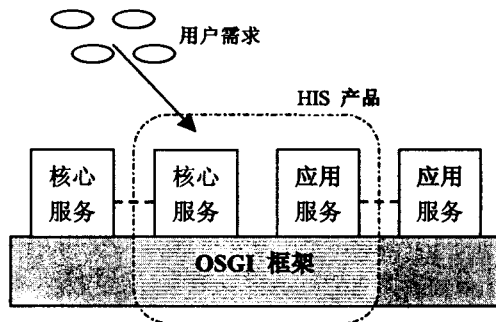


图 4 HIS 原型系统模型

框架下的概念模型,提供规范的标准,以及如何模块化一个 OSGI 的兼容系统,让每一个个性化的服务满足每个用户需求。

应用工程在领域分析的基础上进行实例化,在实例化的过程中有些组件从现有组件库中选取,有些则重新构建,这些组件都是 COM 组件,根据每个系统的具体需求而变更组件,最终实现大粒度的复用。

6 结束语

自从软件危机的提出,人们一直都在寻找高效的软件开发方法,软件复用技术被大家认为是降低成本、缩短开发周期的有效方法。现在的复用技术有基于组件复用技术和产品线开发复用技术等。文中从产品线理论进行研究,对体系结构、开发过程等内容进行了深入的探讨。最后用 HIS 系统举例,提出在 OSGI 框架结构下的应用开发。完善的产品线是一个长期的工作,国外已经有较成熟的产品线,所以我们有责任把产品线理论在国内的软件产业中推广起来。

(上接第 201 页)

5 结束语

变电站电压无功控制(VQC)系统是保障电能质量提高的有效途径,采用优化的九区图控制策略来设计 VQC 系统,弥补了某些区域对控制结果产生振荡现象以及装置频繁动作的缺陷,明显减少了变压器分接头调节和并联补偿电容器组投切次数,增强了电压稳定性和无功补偿的有效性,提高了电能质量,降低了电网损耗,大大改善了电能的利用率。

参考文献:

- [1] 杨 益,方潜生.基于 FPGA 动态跟踪型功率因数补偿控

(上接第 205 页)

不再赘述。

3 结束语

文中探讨了通用的呼叫中心业务系统框架设计的可行性,并提出了一套设计方案。该设计方案采用 MVC 设计模式、基于 B/S 架构,与具体的业务无关,可应用于多种行业,呼叫中心业务系统的开发人员可以在本设计方案的业务系统框架的基础上扩展出与具体业务相关的功能模块。因为可以重复使用、不涉及业务,该设计方案可以减少开发成本,缩短开发周期。根据本设计方案实现的陕西移动 114 号码百事通项目正在实施中,验证了本方案的可行性。

参考文献:

- [1] Mili H. Reuse - Based Software Engineering[M]. 北京:电子工业出版社,2004.
- [2] Bass L, Clements P, Kazman R. Software Architecture in Practice[M]. [s. l.]: Addison Wesley, 1997.
- [3] Mili H, Mili A, Yacoub S, et al. 基于重用的软件工程:技术、组织和控制[M]. 韩 柯译. 北京:电子工业出版社, 2004.
- [4] 杨美清,朱 冰,梅 宏. 软件复用[J]. 软件学报,1995,6 (9):525 - 533.
- [5] Frakes W B, Kang K C. Software reuse research: Status and future[J]. IEEE Transactions on Software Engineering, 2005, 31(7):529 - 536.
- [6] Gornaa H. Designing Software Product Lines with UML: From Use Cases to Pattern - Based Software Architectures [M]. Boston: Addison - Wesley, 2004.
- [7] Chastek G, Donohoe P, McGregor J D. Product line production planning for the home integration system example[M]// Product Line Practice Initiative. [s. l.]: Carnegie Mellon University, 2002.

制器的设计[J]. 工业控制计算机, 2006, 11: 79 - 80.

- [2] 唐 艳,王万新,王洪博,等. 九区图法优化实时无功补偿及其应用[J]. 机械工程与自动化, 2006(4): 154 - 156.
- [3] 康明才. 基于遗传算法的变电站电压 - 无功综合控制[J]. 南京理工大学学报, 2002, 26(5): 490 - 493.
- [4] Celoxica. Handel - C Language Reference Manual[S]. [s. l.]: Celoxica, 2003.
- [5] Celoxica. DK Design Suite User Manual[S]. [s. l.]: Celoxica, 2003.
- [6] 杨 益,方潜生. 基于 Handel - C 的伪随机数发生器的设计与实现[J]. 计算机技术与发展, 2006, 16(12): 124 - 126.

参考文献:

- [1] 耿 刚,陈 平. XML 在呼叫中心业务系统中的应用与研究[J]. 现代电子技术, 2003, 26(24): 31 - 33.
- [2] 许 红. 分布式呼叫中心系统的设计[J]. 甘肃科技, 2006, 22(4): 36 - 38.
- [3] Bernett H, Jaramillo M L. Assessing Web - enabled call center technologies[J]. IT Professional, 2001, 3(3): 24 - 30.
- [4] Lam W, Shankararaman V. An enterprise integration methodology[J]. IT Professional, 2004, 6(2): 40 - 48.
- [5] Cheng I, Boyette N, Krishna V. Towards a Low - Cost High - Quality Service Call Architecture[C]// Services Computing, 2006. SCC '06. IEEE International Conference. [s. l.]: [s. n.], 2006: 261 - 264.