

领域工程概述

An Outline of Domain Engineering

李克勤 陈兆良 梅 宏 杨美清

(北京大学计算机科学技术系 北京 100871)

Abstract Domain engineering is the process establishing fundamental capability and necessary basis for the application engineering of a group of similar systems. It covers all the activities to build reusable software components.

Domain engineering helps to solve the identification, organization and utilization of reusable information, so it provides powerful support for software reuse. Domain engineering promises to produce component with higher reusability, and defines chances and contexts of reuse by defining architecture. All of these eventually support developers to reuse these components, and make reuse formal, systematic and efficient.

This article introduces domain engineering briefly, discusses its supporting to software reuse, and introduces systematic software reuse as the basis of domain engineering. After that, this article discusses basic technological problems of domain engineering in detail, and presents two domain engineering methods.

Keywords Software engineering, Software reuse, Domain engineering

一、引言

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,它覆盖了建立可复用的软件构件的所有活动。其中“领域”是指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。

领域工程对领域中的系统进行分析,识别这些应用的共同特征和可变特征,对刻画这些特征的对象和操作进行选择 and 抽象,形成领域模型,依据领域模型产生出领域中应用共同具有的体系结构(即特定领域的软件构架,缩写为 DSSA)或生成过程,并以此为基础识别、开发和组织可复用构件^[1]。这样,当开发同一领域中的新应用时,可以根据领域模型,确定新应用的需求规约,根据特定领域的软件构架形成新应用的设计,并以此为基础选择可复用构件进行组装,从而形成新系统。

领域工程有助于产生具有较高可复用性的构件。领域工程将关于领域的知识转化为领域中系统共同的规约、设计和构架,使得可以被复用的信息的范围,扩大到了抽象级别较高的分析和设计阶段。由于通过领域工程产生的可复用构件来源于领域中现

有的系统,体现了领域中系统的本质需求,因此这些构件具有较高的可复用性^[1]。

同时,领域工程产生了领域模型和特定领域的软件构架(DSSA)或应用系统的生成过程,这对于基于复用的开发很有帮助。可复用构件是根据领域模型和 DSSA 组织的,方便了构件的检索。开发以领域模型和 DSSA 为线索进行,可以帮助开发者识别复用机会,判断可复用构件是否符合当前需要, DSSA 为构件组装提供了上下文,使得利用可复用构件组装或生成新的系统较为容易。

下面先介绍系统化的软件复用,然后详细讨论本文主题,最后简介二种方法。

二、系统化的软件复用

2.1 个别复用与系统化复用

对复用进行分类的方式之一是依据复用的组织方式,将复用区分为系统化的(或有计划的)复用(Systematic, planned reuse)和个别的(或“机会主义的”)复用(Ad-hoc, opportunistic reuse)。

在个别的软件复用中,存在一组可复用构件,应用开发者对它们进行选择和复用,大致过程是,识别可能复用的机会,选择满足需要的构件或经过修改

得以满足的构件,并利用它们组装成新的应用系统^[1]。在这种复用中,复用是在个人的,而不是在项目的级别上进行的,也没有定义复用的过程^[9]。

在系统化的软件复用中,不但存在一组可复用构件,而且定义了在新的应用系统的开发过程中复用哪些构件以及如何进行适应性修改^[10]。由于一般性地识别、表示和组织可复用信息是非常困难的,因此系统化的复用将注意力集中于特定的领域,而且在系统化的复用中非常重视软件生存周期中抽象级别较高的产品的复用^[4]。其中,复用是在项目级别进行的,定义了复用的指南和过程,还定义了度量标准以衡量复用的效率^[1]。

与个别的软件复用相比,系统化的软件复用对于提高软件的质量和生产率具有更大的作用。由于将注意力集中于特定的领域,使得软件开发组织可以获得对该领域的深入了解,对于可复用信息的识别和表示会比较容易和准确,根据复用指南对复用过程会有帮助。完整定义的复用过程和对复用的度量,使得复用可以比较规范和系统化,从而有助于实现软件开发组织实施复用的预期目标。

同时实施系统化的软件复用也有较大的风险。实施系统化的软件复用的软件开发组织需要解决一系列技术和非技术的问题,例如,分析本组织的需要,定义适合这些需要的复用过程,调整人员的组织方式,建立度量标准以衡量复用的效率,并据此调整复用过程,估计投资和收益,建立特定领域的可复用构件,等等,这些行为需要较大的前期投资和整个软件开发过程和原则的变化,而预言这些投资的回报却是困难的。

2.2 复用基础设施

在系统化的软件复用中,充分的可复用信息的存在是非常重要的,这些信息需要被显式地表示,以便在开发过程中被复用。这些可复用信息,和为方便地定位和操作它们的一些辅助信息一起构成了复用基础设施^[1]。复用基础设施也是基于特定领域的。

复用基础设施必须适应软件开发组织的需要。它根据具体的需要回答下列问题:可复用的信息包括哪些类型,可复用信息如何描述,可复用信息如何进行分类以利于检索,等等。

在复用实施的过程中,复用基础设施是不断演化的。在开发新系统时,开发者从中获得可复用信息,将它们集成到新系统中去。在此过程中可能产生对于复用基础设施的关于信息描述、分类方式等方面的反馈意见,复用基础设施将根据这些意见进行调整。新系统可能具有自己的特征,它的开发就可能需要复用基础设施中没有包含的信息,这就需要进

行新的开发。当新系统开发完成,成为现有系统时,复用基础设施就需要根据当前所有的现有系统进行演化,以尽可能地把握本领域中系统的共同特征。复用基础设施的演化,是可以并且需要随着领域中系统的开发持续进行的^[11]。

2.3 领域工程是其成功因素之一

系统化复用的成功依赖于很多因素,其中领域工程是其关键^[11]。

通过领域工程,将关于一个领域的知识转化成为一组规约、构架和相应的可复用构件。由于这些信息来自于同一领域中现有的系统,因此它们具有较高的可复用性。这些可复用信息构成了复用基础设施的重要组成部分,因此,复用基础设施的形成是通过领域工程实现的。

复用基础设施的演化也是通过领域工程进行。当一个领域增加了应用系统时,通过领域工程,可以对这些系统进行分析,将新系统的特征也包含在规约、构架和可复用构件中,从而使本领域中系统开发的知识和经验尽可能地反映在复用基础设施中,以促进新系统的开发。

领域工程对于系统化的软件复用的意义还在于,领域工程不仅产生了可复用性较高的构件,而且通过产生构架定义了复用的时机和复用的上下文。这样就对开发者复用这些构件提供了有力的支持,使得复用变得规范、系统和高效。

三、领域工程

3.1 基本假定

进行领域工程的研究和实施领域工程是基于两个基本的认识^[12]:

(1)可复用信息的领域特定性。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。即,我们说某信息具有可复用性,是指当使用特定的方法解决特定的问题时,它是可复用的。基于这一基本认识,在识别、获取和表示可复用信息时,应采用面向领域的策略。

(2)问题领域的内聚性和稳定性。关于现实世界问题领域的解决方法的知识是充分内聚和充分稳定的,这才使得获取和表示这些知识的努力是有意義的。这一基本认识是实际观察的结果。一个问题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把握这些可以解决大量问题的知识,领域的稳定性,使得获取和表示这些信息所付出的代价,可以通过在一段较长的时间内多次复用它们来得到补偿。

3.2 基本行为

实施领域工程的过程中包含了一些基本的行为。虽然具体的领域工程方法可能定义不同的概念、步骤、产品等,但这些基本行为是大体上一致的。以下将分三个阶段介绍这些行为:

(1)领域分析:这个阶段的主要目标是获得领域模型。领域模型描述领域中系统之间的共同的需求^[6],称领域模型所描述的需求为“领域需求”,在这个阶段中首先要进行一些准备性的行为,这包括定义领域的边界,从而明确分析的对象;识别信息源,即领域分析和整个领域工程过程中信息的来源,可能的信息源包括现存系统、技术文献、问题域和系统开发的专家、用户调查和 market 分析、领域演化的历史记录等^[7]。在此基础上,就可以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,从而建立领域模型,当领域中存在大量系统时,需要选择它们的一个子集作为样本系统,对样本系统需求的考察将显示领域需求的一个变化范围。一些需求对所有被考察的系统是共同的。一些需求是单个系统所独有的,很多需求位于这两个极端之间,即被部分系统共享。

(2)领域设计:这个阶段的目标是获得 DSSA。DSSA 描述在领域模型中表示的需求的解决方案,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^[6]。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。它可以通过表示多选的(alternative)、可选的(optional)解决方案等来作到这一点。由于复用基础设施是依据领域模型和 DSSA 来组织的,因此在这个阶段通过获得 DSSA,也就同时形成了复用基础设施的规约^[8]。

(3)领域实现:这个阶段的主要目标是依据领域模型和 DSSA 开发和组织可复用信息^[6]。这些可复用信息可能是从现有系统中提取得到,也可能需要通过新的开发得到。它们依据领域模型和 DSSA 进行组织,也就是领域模型和 DSSA 定义了这些可复用信息的复用时机,从而支持了系统化的软件复用。这个阶段也可以看作复用基础设施的实现阶段^[8]。

领域工程是一个反复的、逐渐精化的过程^[8]。在实施领域工程的每个阶段中,都可能返回到以前的步骤,对以前的步骤得到的结果进行修改和完善,再回到当前步骤,在新的基础上进行本阶段的行为。

3.3 与应用工程的关系

在以下的讨论中,将开发单个应用系统的软件工程过程称为应用工程。领域工程与应用工程是既有区别又互相联系的。

在应用工程中,软件开发人员的任务是在特定的条件下,针对一组特定的需求产生一组特定的设计和实现。其中的行为及其产生的结果基本上是针对当前开发的特定系统的。与此相对,在领域工程中,领域工程人员的基本任务是对一个领域中的所有系统进行抽象,而不再局限于个别的系统。因此,与应用工程相比,领域工程处于一个较高的抽象级别上^[7]。其中,对领域中相似系统的共同特征进行了抽象,并通过领域模型和 DSSA 表示了这些共同特征之间的关系。

领域工程和应用工程又是互相联系的,一方面,通过应用工程得到的现有系统(包括需求规约、设计、实现等)是领域工程的主要的信息来源,领域工程的各个阶段主要是对应用工程中相应阶段的产品进行抽象。领域工程的产品—领域模型、DSSA、可复用构件等,又对本领域中新系统的应用工程提供了支持,另一方面,领域工程和应用工程需要解决一些相似的问题,如:如何从多种信息源中获取用户的需求,如何表示需求规约,如何进行设计,如何表示设计模型,如何进行构件开发,如何在需求规约、设计和实现间保持逻辑联系,如何对需求规约、设计和实现进行演化等。因此,领域工程的步骤、行为、产品等很多方面都可以和应用工程进行类比^[7]。

3.4 相关技术

领域工程不是一个孤立的学科,它与应用工程、人工智能、信息管理等学科有着密切的关系,这些学科中的一些相关技术和原则可以用来解决领域工程的问题^[9]。

如上所述,领域工程与应用工程要解决一些相似的问题,因此,在应用工程中已被广泛应用的许多方法、技术和原则都可以在领域工程中进行应用,如,结构化的分析设计方法、面向对象的分析设计方法、实体-关系图、数据流图等等,都可以在领域工程中得到广泛的应用。同时,这些原本用于开发特定系统的方法、技术和原则应用于领域工程时,需要进行补充和裁剪,以适应新的应用环境,如需要增加表示多选的、可选的等变化性的能力。

领域工程,特别是其中的领域分析阶段,在一定程度上,可以看作一种知识获取的过程。而人工智能学科已经提供了进行知识获取、知识表示、知识维护的方法、技术和工具,例如机器学习和专家系统的研究都可以对领域工程中的知识获取提供帮助,而谓词逻辑、语义网络、产生式规则等知识表示方法,也可以用于领域工程中。超文本等信息管理科学的技术可以对可复用信息的可视化和理解提供帮助。在这方面,领域工程的研究需要根据这些方法、技术和

工具各自的优点和限制,对它们进行裁剪和组合,以满足自己的需要。

3.5 参与人员

与对应用工程的研究类似,参与领域工程的人员可以划分为四种角色:领域专家、领域分析员、领域设计员和领域实现员。以下将对这四种角色分别通过回答三个问题进行介绍^[4]:这种角色由什么人员来充当?他们在领域工程中承担什么任务?他们需要具有哪些技能?

(1)领域专家:可能包括该领域中系统的有经验的用户、从事该领域中系统的需求分析、设计、实现以及项目管理的有经验的软件工程师等。主要任务包括提供关于领域中系统的需求规约和实现的知识,帮助组织规范的、一致的领域字典,帮助选择样本系统作为领域工程的依据,复审领域模型、DSSA等领域工程产品,等等。领域专家应该熟悉该领域中系统的软件设计和实现、硬件限制、未来的用户需求及技术走向等。

(2)领域分析员:应由具有知识工程背景的有经验的系统分析员来担任。主要任务包括控制整个领域分析过程,进行知识获取,将获取的知识组织到领域模型中,根据现有系统、标准规范等验证领域模型的准确性和一致性,维护领域模型。领域分析员应熟悉软件复用和领域分析方法;熟悉进行知识获取和知识表示所需的技术、语言和工具;应有一定的该领域的经验,以便于分析领域中的问题及与领域专家进行交互;应具有较高的进行抽象、关联和类比的能力;应具有较高的与他人交互和合作的能力。

(3)领域设计员:应由有经验的软件设计人员来担任。主要任务包括控制整个领域设计过程,根据领域模型和现有的系统开发出 DSSA,对 DSSA 的准确性和一致性进行验证,建立领域模型和 DSSA 之间的联系。领域设计员应熟悉软件复用和领域设计方法;熟悉软件设计方法;应有一定的该领域的经验,以便于分析领域中的问题及与领域专家进行交互。

(4)领域实现员:应由有经验的程序设计人员来担任。主要任务包括根据领域模型和 DSSA,或者从头开发可复用构件,或者利用再工程的技术从现有系统中提取出可复用构件,对可复用构件进行验证,建立 DSSA 与可复用构件间的联系。领域实现员应熟悉软件复用、领域实现及软件再工程技术;熟悉程序设计;具有一定的该领域的经验。

四、领域工程方法

领域工程的实施需要依据一定的领域工程方法

来进行,与应用工程类似,形成系统化的领域工程方法需要满足一定的条件^{[1][2]}。整个领域工程过程应划分为明确定义的阶段,每个阶段应有明确的目标和具体的行为,产生一定的中间产品,产品有明确定义的表示方法,产品的形成应遵循一定的操作指南,使用一定的工具进行开发,有一定的验证方法保证中间产品的正确性和一致性,整个过程有明确定义的里程碑,有明确定义的终止条件,对于整个过程的执行情况有明确定义的度量标准等等。

当前已经出现了很多领域工程的方法,它们都基于一些共同的观念,同时,它们之间存在一定的差异性,这主要表现在它们定义了各自不同的目标、产品和过程,有各自不同的适用情况。正如没有一种需求规约表示方法、设计方法、程序设计语言、CASE 工具能够适用于所有应用系统的开发一样,没有理由认为存在一种领域工程方法适用于所有需要进行领域工程的情况^[4]。因此,当软件开发组织选择领域工程方法时,需要根据本组织的具体复用情况、在具体的软件复用环境下对领域工程方法进行比较。这里的具体复用情况包括复用目标、复用过程、复用策略、复用度量、支持工具等等。选择领域工程方法时需要了解领域工程方法是如何与软件开发过程相结合的,需要衡量在本组织所采用的软件开发过程中结合领域工程方法可能带来的收益,特别需要考虑某种领域工程方法将带来的长期和短期的投资与收益,还需要考虑本组织当前的需要和预期的将来的收益。

以下将简要地介绍两种领域工程方法。

4.1 CMU/SEI 的 FODA 方法

FODA 方法(面向特征的领域分析法)是由卡内基·梅隆大学软件工程研究所提出的领域工程方法^[2]。FODA 的过程分为以下三个阶段。

(1)上下文分析:目标是定义领域的范围。在这个阶段要分析领域与外部元素间的关系,例如不同的操作环境,不同的数据需求等,还要对可变性进行评价。分析的结果是上下文模型。

(2)领域建模:领域的范围确定后,领域建模阶段提供了一些步骤来分析领域中的应用表现出的共同性和差异,并产生一些领域模型。本阶段主要包含三种行为:

- 特征分析。目的是要获得客户或最终用户对一类系统的一般能力的理解,即特征。特征描述了领域应用的上下文,需要的操作和属性,以及表示法的变化。

- 信息分析。目的是要定义和分析为实现领域中应用所需的领域知识和数据需求。目标是用领域

实体及其相互之间的关系表示领域知识,并使它们在操作分析和体系结构建模中可以用来派生对象和数据定义。

·操作分析 目的是要识别领域中应用的行为特性,例如数据流和控制流的共同性和差异、有限状态自动机模型。

(3)构架建模:这个阶段为领域中的应用提供软件解决方案。在这个阶段中开发出体系结构模型,即领域中应用的高层设计。这个阶段的焦点为识别并发进程和面向领域的共同模块。这个阶段中定义进程,并将定义在领域模型中的特征、功能和数据对象分配到进程和模块。

FODA 方法定义明确,已被应用于商业和军事应用中。

4.2 Will Tracz 的 DSSA 方法

Will Tracz 提出了进行领域工程的 DSSA 方法^[10]。在最高的级别上,本过程有五个阶段。每个阶段可以进一步划分为一些步骤或子阶段。每个阶段包括一组需要回答的问题,一组需要的输入,一组将产生的输出和验证标准。本过程是并发的、递归的和反复的。或者说,它是螺旋型的。完成本过程可能需要对每个阶段经历几遍,每次增加更多的细节。

本领域工程过程的五个阶段是:

(1)定义领域范围:重点是确定什么在感兴趣的领域中以及本过程到何时结束。这个阶段的一个主要输出是领域中的应用需要满足的一系列用户的需求;

(2)定义领域特定的元素:目标是编译领域字典和领域术语的同义词词典。在领域工程过程的前一个阶段产生的高层块图将被增加更多的细节,特别是识别领域中应用间的共同性和差异性;

(3)定义领域特定的设计和实现需求约束:目标是描述解空间中有差别的特性。不仅要识别出约束,并且要记录约束对设计和实现决定造成的后果,还要记录对处理这些问题时产生的所有问题的讨论;

(4)定义领域模型和构架:目标是产生一般的构架,并说明构成它们的模块或构件的语法和语义;

(5)产生、搜集可复用产品:目标是为 DSSA 增加构件使得它可以被用来产生问题域中的新应用。

结束语:领域工程是系统化的软件复用的关键,支持了复用基础设施的形成和演化,有助于提高软件的生产率和质量。领域工程与应用工程有密切的联系,领域工程的研究可以从应用工程和其它相关学科中获得有益的借鉴。当前已经产生了一些领域

工程方法,它们有一些共同的基本观念,但由于各个方法目标、过程和产品的不同,这些方法又有一定的差别。软件开发组织在实施领域工程时,需要根据自身的复用情况作出选择和调整。

领域工程研究的重要内容是在实践中应用领域工程方法。通过在具体领域中的应用,对领域工程方法中的过程和产品进行调整,开发领域工程的辅助工具,并且通过实践中获得的数据,建立领域工程的度量 and 投资/收益模型,使得领域工程方法的选择和领域工程的实施逐步建立在量化的基础上。

参考文献

- 1 Arango Guillermo Domain Analysis-From Art Form To Engineering Discipline. In Proc. of the 5th Intl Workshop on Software Specifications and Design 1989. 152~159
- 2 Arango Guillermo, Prieto-Diaz Ruben Domain Analysis Concepts and Research Directions, in: R. Prieto-Diaz and G. Arango, eds. Domain Analysis and Software System Modeling Los Alamitos, California, IEEE Computer Society Press, 1991
- 3 Foreman John, et al. Software Technology Review CMU/SEI, June 1997
- 4 Frakes William B, Isoda Sadahiro Success Factors of Systematic Reuse IEEE Software, September 1994. 15~19
- 5 Mili Hamed, et al. Reusing Software: Issues and Research Directions. IEEE Trans on Software Engineering, 1995, 21(6)
- 6 Petro James, et al. Model-Base Reuse Repository-Concepts and Experience. In Hausi A. Muller & Ronald J. Norman eds. Proc. 7th Intl. Workshop on CASE. IEEE Computer Society Press, Los Alamitos, California, USA, 1995. 60~69
- 7 Prieto-Diaz Ruben. Domain Analysis for Reusability. In Proc. of COMPSAC'87 1987. 23~29
- 8 Prieto-Diaz Ruben Domain Analysis: An Introduction. Software Engineering Notes, 1990, 15(2). 47~54
- 9 Prieto-Diaz Ruben. Status Report: Software Reusability. IEEE Software, 1993, 10(3): 61~66
- 10 Tracz Will. Confessions of a Used Program Salesman-Institutionalizing Software Reuse. Addison-Wesley Publishing Co., New York, NY, April 1995
- 11 Wartik Steven, et al. Criteria for Comparing Reuse-Oriented Domain Analysis Approaches. International Journal of Software Engineering and Knowledge Engineering, 1992, 2(3). 403~431