

99,26(5)

软件复用

软件开发

软件构件

计算机

计算机科学 1999 Vol. 26 No. 5

1-4

软件复用及相关技术

Software Reuse and Its Correlated Techniques

杨芙清

TP311.52

(北京大学计算机科学技术系 北京 100871)

Abstract In order to promote research and application to software reuse in domestic academic and industrial circles, we organize and write a batch of articles which present the important results of oversea research and practice in software reuse. These articles discuss mainly software reuse and its correlated techniques. The paper is a general survey of these articles.

Keywords Software reuse, Development for reuse, Development with reuse, Survey

一、关于软件复用

通常情况下,应用软件系统的开发过程包含以下几个阶段:需求分析、设计、编码、测试、维护等。当每个应用系统的开发都从头开始时,其中必然存在大量的重复劳动,如:用户需求获取、需求分析和设计、编码、测试和文档工作等方面都会有重复。

应用系统的本质通常包含三类成分:①通用基本构件,是特定于计算机系统的构成成分,如基本的数据结构、用户界面元素等,可以存在于各种应用系统中;②领域共性构件,是应用系统所属领域的共性构成成分,存在于该领域的各个应用系统中;③应用专用构件,是每个应用系统的特有构成成分。

其中的重复劳动主要存在于前两类构成成分的重复开发。

软件复用是在软件开发中避免重复劳动的解决方案,出发点是应用系统的开发不再采用一切“从零开始”的模式,而是以已有的工作为基础,充分利用过去应用系统开发中积累的知识和经验,如:需求分析结果、设计方案、源代码、测试计划及测试案例等,从而将开发的重点集中于应用的特有构成成分。

通过软件复用,在应用系统开发中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。

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,相应地,可复用软件是指为了复用

目的而设计的软件。与软件复用的概念相关,重复使用软件的行为还可能是重复使用“并非为了复用目的而设计的软件”的过程,或在一个应用系统的不同版本间重复使用代码的过程,这两类行为都不属于严格意义上的软件复用。

以下的类比有助于进一步说明软件复用的概念,在软件演化的过程中,重复使用的行为可能发生在三个维上:

1. 时间维:使用以前的软件版本作为新版本的基础,加入新功能,适应新需求,即软件维护(适应性维护)。

2. 平台维:以某平台上的软件为基础,修改其和运行平台相关的部分,使其运行于新平台,即软件移植。

3. 应用维:将某软件(或其中构件)用于其他应用系统中,新系统具有不同功能和用途,即真正的软件复用。

这三种行为中都重复使用了现有的软件,但是,真正的复用是为了支持软件在应用维的演化,使用“为复用而开发的软件(构件)”来更快、更好地开发新的应用系统。

复用概念的第一次引入是在1968年NATO(北大西洋公约组织)软件工程会议上,McIlroy的论文“大量生产的软件构件”中。之前,子程序的概念也体现了复用的思想,但其目的是为了节省当时昂贵的机器内存资源,并不是为了节省开发软件所需的人力资源。然而子程序的概念可以用于节省人力资源的目的,从而出现了通用子程序库,供程序员在编程时使用。例如,数学程序库就是非常成功的子程

杨芙清 科学院院士。

序复用的例子。

在其后的发展过程中,有许多复用技术的研究成果和成功的复用实践活动,但是,复用技术在整体上对软件产业的影响却并不尽如人意。这是由于技术方面和非技术方面的种种因素造成的,而技术上的不成熟是主要原因。近十几年来,面向对象技术出现并逐步成为主流技术,为软件复用提供了基本的技术支持。软件复用研究重新成为热点,被视为解决软件危机、提高软件生产率和质量现实可行的途径。

分析传统产业的发展,其基本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”是必需的手段。实践表明,这种模式是产业工程化、工业化的必由之路。标准零部件生产业的独立存在和发展是产业形成规模经济的前提。机械、建筑等传统行业以及年轻的计算机硬件产业的成功发展均是基于这种模式并充分证明了这种模式的可行性和正确性,这种模式是软件产业发展的良好借鉴,软件产业要发展并形成规模经济,标准构件的生产和构件的复用是关键因素。这正是软件复用受到高度重视的根本原因。

软件复用可以从多个角度进行考察,依据复用的对象,可以将软件复用分为产品复用和过程复用。产品复用指复用已有的软件构件,通过构件集成(组装)得到新系统。过程复用指复用已有的软件开发过程,使用可复用的应用生成器来自动或半自动地生成所需系统。过程复用依赖于软件自动化技术的发展,目前只适用于一些特殊的应用领域。产品复用是目前现实的、主流的途径。

依据对可复用信息进行复用的方式,可以将软件复用区分为黑盒复用和白盒复用。黑盒复用指对已有构件不需作任何修改,直接复用。这是理想的方式。白盒复用指已有构件并不能完全符合用户需求,要根据用户需求进行适应性修改后才可使用。而在大多数应用的组装过程中,构件的适应性修改是必需的。

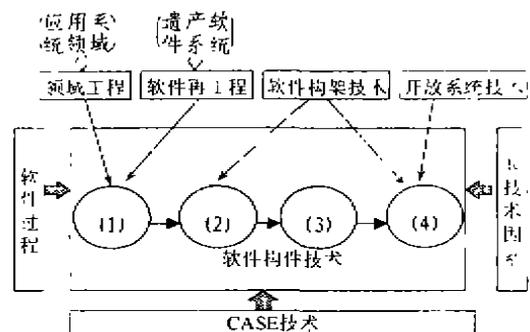
软件复用有三个基本问题,一是必须有可以复用的对象,二是所复用的对象必须是有用的,三是复用者需要知道如何去使用被复用的对象,软件复用包括两个相关的过程:可复用软件(构件)的开发(Development for Reuse)和基于可复用软件(构件)的应用系统构造(集成和组装)(Development with Reuse),解决好这几个方面的问题才能实现真正成功的软件复用。

二、相关技术因素

实现软件复用的关键因素(技术和非技术因素)

• 2 •

主要包括:软件构件技术(Software Component Technology)、领域工程(Domain Engineering)、软件构架(Software Architecture)、软件再工程(Software Reengineering)、开放系统(Open System)、软件过程(Software Process)、CASE技术等以及各种非技术因素,且各种因素是互相联系、互相影响的,如图1所示,它们结合在一起,共同影响软件复用的实现。



(1) 构件、构架获取; (2) 构件标准化与描述;

(3) 构件分类、存储与检索; (4) 构件组装。

图1 实现软件复用的关键因素

2.1 软件构件技术

构件是指应用系统中可以明确辨识的构成成分,而可复用构件是指具有相对独立的功能和可复用价值的构件。

可复用构件应具备以下属性:①有用性:必须提供有用的功能;②可用性:必须易于理解和使用;③质量:自身及其变形必须能正确工作;④适应性:应该易于通过参数化等方式在不同语境中进行配置;⑤可移植性:应能在不同的硬件平台和软件环境中工作。

随着对软件复用理解的深入,构件的概念已不再局限于源代码构件,而是延伸到需求、系统和软件的需求规约、系统和软件的构架、文档、测试计划、测试案例和数据以及其他对开发活动有用的信息。这些信息都可以称为可复用软件构件。

软件构件技术是支持软件复用的核心技术,是近几年来迅速发展并受到高度重视的一个学科分支,其主要研究内容包括:

1. 构件获取:有目的的构件生产和从已有系统中挖掘提取构件;

2. 构件模型:研究构件的本质特征及构件间的关系;

3. 构件描述语言:以构件模型为基础,解决构件的精确描述、理解及组装问题;

4. 构件分类与检索:研究构件分类策略、组织模式及检索策略,建立构件库系统,支持构件的有效管理;

5. 构件复合组装:在构件模型的基础上研究构件组装机制,包括源代码级的组装和基于构件对象互操作性的运行级组装;

6. 标准化:构件模型的标准化和构件库系统的标准化。

2.2 软件构架

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,构件间通讯、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。

研究软件构架对于进行高效的软件工程具有非常重要的意义:通过这一研究,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之间作出选择,从构架的层次上表示系统,有利于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。

软件构架研究如何快速、可靠地从可复用构件构造系统的方式,着重于软件系统自身的整体结构和构件间的互联,其中主要包括:软件构架原理和风格,软件构架的描述和规约,特定领域软件构架,构件向软件构架的集成机制等。

2.3 领域工程

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软件构件的所有活动。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。领域工程包括三个主要的阶段:

1. 领域分析:主要目标是获得领域模型。该模型描述领域中系统之间共同的需求。本阶段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,从而建立领域模型。

2. 领域设计:目标是获得领域构架(Domain-Specific Software Architecture,缩写为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。

3. 领域实现: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与稳定程度,这种机制可能是一组与领域模型和DSSA相联系的可复用构件,也可能是应用系统的生成器。

这些活动的产品(可复用的软件构件)包括:领域模型,领域构架,领域特定的语言、代码生成器和代码构件等。

2.4 软件再工程

软件复用中的一些问题与现有系统密切相关,如:现有软件系统如何适应当前技术的发展及需求的变化,采用更易于理解的、适应变化的、可复用的系统软件构架并提炼出可复用的软件构件?现存大量的遗产软件系统(Legacy Software)由于技术的发展,正逐渐退出使用,如何对这些系统进行挖掘、整理,得到有用的软件构件?已有的软件构件随着时间的流逝会逐渐变得不可使用,如何对它们进行维护,以延长生存期,充分利用这些可复用构件?等等。软件再工程正是解决这些问题的主要技术手段。

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。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。

2.5 开放系统技术

开放系统技术的基本原则是在系统的开发中使用接口标准,同时使用符合接口标准的实现。这些为系统开发中的设计决策,特别是对于系统的演化,提供了一个稳定的基础,同时,也为系统(子系统)间的互操作提供了保证。开放系统技术具有在保持(甚至是提高)系统效率的前提下降低开发成本、缩短开发周期的可能。对于稳定的接口标准的依赖,使得开放系统更容易适应技术的进步。当前,以解决异构环境中的互操作为目标的分布对象技术是开放系统技术中新的主流技术。

开放系统技术为软件复用提供了良好的支持,特别是分布对象技术使得符合接口标准的构件可以方便地以“即插即用”的方式组装到系统中,实现黑盒复用。这样,在符合接口标准的前提下,构件就可以独立地进行开发,从而形成独立的构件制造业。

2.6 软件过程

软件过程又称软件生存周期过程,是软件生存周期内为达到一定目标而必须实施的一系列相关过程的集合。一个良好定义的软件过程对软件开发的

质量和效率有着重要影响,当前,软件过程研究以及企业的软件过程改善已成为软件工程界的热点,并已出现了一些实用的过程模型标准,如 CMM、ISO9001/TkkIT 等。

然而,基于构件复用的软件开发过程和传统的一切从头开始的软件开发过程有着实质性的不同,探讨适应于软件复用的软件过程自然就成为一个迫切的问题。

2.7 CASE 技术

随着软件工程思想的日益深入人心,以计算机辅助开发软件为目标的 CASE (Computer Aided Software Engmeernug) 技术越来越为众多的软件开发人员所接受,CASE 工具和 CASE 环境得到越来越广泛的应用,CASE 技术对软件工程的很多方面,例如分析、设计、代码生成、测试、版本控制和配置管理、再工程、软件过程、项目管理等等,都可以提供有力的自动或半自动支持。CASE 技术的应用,可以帮助软件开发人员控制软件开发中的复杂性,有利于提高软件开发的效率和质量。

软件复用同样需要 CASE 技术的支持。CASE 技术中与软件复用相关的主要研究内容包括:在面向复用的软件开发中,可复用构件的抽取、描述、分类和存储;在基于复用的软件开发中,可复用构件的检索、提取和组装;可复用构件的度量等等。

2.8 非技术因素

除了上述的技术因素以外,软件复用还涉及众多的非技术因素,如:开发人员如何组织;开发工作如何管理;开发人员知识的更新;创造性和工程化的关系;开发人员的心理障碍;知识产权问题;保守商业秘密的问题;复用前期投入的经济考虑;标准化问题等等。

三、关于青鸟工程

青鸟工程是国家重点支持的科技攻关课题,已有十余年的发展历程。“七五”、“八五”期间,青鸟工程面向我国软件产业基础建设的需求,以实用的软件工程技术为依托,研究开发具有自主知识产权的软件工程环境,为软件产业提供基础设施—软件工具、平台和环境,建立工业化生产的基本手段,促进我国软件开发由手工作坊式转向用计算机辅助开发,以提高软件开发效率,改善软件产品质量。大型软件开发环境青鸟系统便是这一阶段攻关工作的成果。

“九五”期间,青鸟工程的任务是在前期攻关工作的基础上,为形成我国软件产业规模提供技术支

持。重点是研究软件的工业化生产技术,开发软件工业化生产系统——青鸟软件生产线系统,即基于构件-构架模式的软件开发技术及系统,为软件开发提供整体解决方案,推行软件工业化生产模式,促进软件产业规模的形成。

作为研究成果之一,青鸟工程开发了基于异构平台、具有多信息源接口的应用系统集成(组装)环境青鸟Ⅱ型(JB3)系统。青鸟Ⅱ型系统研制的目标是针对软件工业化生产的需求,完善并初步实现青鸟软件生产线的思想,制定软件工业化生产标准和规范,研究基于“构件-构架”模式的软件工业化生产技术,研制支持面向对象技术,支持软件复用的,基于异构平台、具有多信息源接口的应用系统集成(组装)环境。

四、关于复用的研究与实践

软件复用经过近 30 年的发展,已有许多成功的研究和实践成果,正逐步成为实现软件工程化、工业化生产的首选途径。结合青鸟工程攻关,我们一直致力于软件复用及相关技术的研究工作,并取得了一定成果。我们的研究思路是:保持和国际研究主流的不同步,借鉴国际上最新研究成果,结合自身实践,探索创新之路。

为了促进我国学术界和产业界对软件复用技术的研究和应用,我们特组织了一批文章,介绍国际上在软件复用技术研究及实践方面的重要成果。连本文在内,共有 14 篇文章,内容涉及软件复用及相关技术的主要方面,包括:领域工程、再工程和逆向工程、构件及构件库标准化工作、分布对象技术和数据仓库技术在复用中的应用、以及其他成功的研究和实践活动(包括青鸟工程)等。希望这批文章能对同行有一定借鉴和帮助。

软件复用技术将促进软件产业的变革,使软件产业真正走上工程化、工业化的发展轨道,软件复用将造成软件产业的合理分工,专业化的构件生产将成为独立的产业而存在,软件系统的开发将由软件系统集成商通过购买商用构件,集成组装而成。软件复用所引起的产业变革将会带来更多的商业契机,形成新的增长点。这对我国软件产业的发展是一个良好的机遇,要抓住机遇,需要我国产业界和学术界的共同努力。

本组文章受到国家“九五”重点科技攻关项目的资助,在此一并表示感谢。