

65-68

软件重用研究与应用

TP311.5

毛新军 齐治昌

(国防科技大学计算机系 长沙410073)

摘要 The Potential of software reuse is that it can effectively resolve the problem of software development at present. Research on software reuse, project development and application have induced more and more governments, organizations and software engineers' attention. The paper introduces software reuse development, research and application status, gives a prospect of future development.

关键词 Software reuse, Software component

一、技术问题研究

九十年代,软件重用的潜力日益为人们所接受。软件重用的研究方向也发生了根本性变化,人们不仅研究技术问题,而且开始研究非技术问题,如管理问题、资金问题等等。

软件重用的研究状况如图1所示。

技术问题是软件重用研究的基础和核心,它涉及领域分析和领域工程、软部件工程以及软部件可重用性等方面的研究。

1. 领域分析和领域工程(DA&DE)

领域分析(DA)是在软件开发过程中对应用领域信息进行收集、抽象和组织,识别和构造出各种可重用信息以使得这些信息在开发该领域新软件系统时能最大限度地加以重用。DA&DE是系统化、形式化以及有效地进行软件重用的关键。目前,这一领域的研究包括:

·DA研究。DA通过抽象和提取应用领域系统间的共性,生成领域模型,字典以及体系结构。领域模型描述领域各系统间的公共功能、数据和关系;字典定义该领域所使用的术语;体系结构描述有关封装、控制、界面情况。

·DA过程模型。研究DA过程中领域知识来源、种类和形式;各类人员(领域专家、分析人员和工程人员)的关系以及人员的组织和通讯;DA过程中的相关活动、方法和管理过程。

·形式化方法。包括提取、组织、描述、操作和理解可重用信息,形式化DA过程以及开发支持它的技术和工具。

·领域知识的表示。如实体-关系图、谓词逻辑方法、语义网络方法等等。

·其它研究,包括领域边界的定义、领域语言的形式和开发、引进DA的软件开发过程模型的研究等等。

DA是一个知识密集复杂活动,DA过程中有

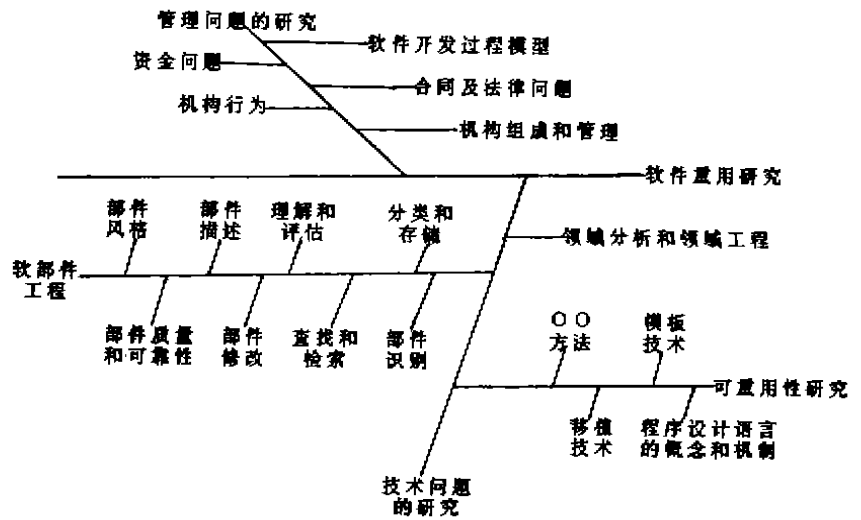


图1 软件重用研究现状

关人员和过程的管理、DA 方法以及领域专家过程知识重用等问题仍需进一步研究。

2. 软部件可重用性识别

为了确保软部件库中各可重用软部件的质量,必须对软部件的可重用性加以识别。软部件可重用性识别是软件重用技术的一大难题,原因是:(1)缺乏软部件可重用性统一的认识和标准;(2)目前没一种好的方法和技术识别软部件的可重用性。近期工作多数还停留在源代码级上,主要集中在三个方面^[1]:

- 提供一个软部件可重用的属性模型,反映可重用软部件的特性;

- 基于软件度量 and 测量技术对软部件属性模型进行度量;

- 建立一组标准,对软部件可重用性加以识别。

今后的工作包括建立软部件可重用性标准,研究和开发软部件识别技术和方法,研究生命周期其它阶段的软部件可重用性。

3. 部件描述

为了对软部件库中的软部件进行分类、存贮、检索和理解,必须对软部件的内部特性(如部件功能)和外部特性(如部件间关系)加以描述。目前描述软部件的方法有:

- 传统的库和信息科学方法^[2];

- 基于知识方法。用框架描述部件的内部特性,语义网络描述部件的外部特性,并通过一组规则对这些框架和语义网络进行操作。

- 超级文本方法。超级文本由一系列结点和边组成。结点对应于软部件实体,边则反映这些结点(软部件)间的关系。超级文本允许用户从一个结点通过连接该结点的边到达另一结点,从而对这些点进行查找、检索、浏览和重用。

4. 软部件的分类和存贮

软部件的分类直接决定了对软部件的检索和定位。目前软部件分类方法主要有:

- 基于库的面分类方法。每个面从不同的侧面描述软部件特性;

- 基于实体-关系模型的软部件分类和检索方法^[3];

- 基于自然语言处理的分类方法。该方法通过填充界面,形成软部件描述符框架,从而对软部件进行分类^[4]。

软部件的存贮机制应支持软部件的有效检索。软部件库不仅存贮软部件实体,还应存贮有关软部

件关系信息。人们已经研究,如何更好地利用数据库系统、知识及信息检索系统来支持软部件库的创建和管理,以及采用面向对象数据库模型、语义数据模型来创建和管理软部件库。

5. 软部件的查找和检索

对面分类法,软部件的查找和检索依赖于对软部件各个面的说明。Grag 开发的 REX(Resource Extractor)把属性和/或关系用作修饰符形成查询请求;Wood 为他的库系统提供了一个检索界面,该界面通过提示用户一个描述软部件动作的动词以及一个描述软部件操作对象的名词,形成软部件描述符框架,查找软部件库;Tarumi 基于用户对对象名、属性、关系和操作的输入,开发了一个基于规则的检索机制,该机制组合了形式化和非形式化方法;Frakes 等建议使用信息检索(IR)系统检索和查找软部件,同时他们还利用成熟的 IR 技术进行术语匹配。

技术问题的研究还包括:

- 程序设计语言的概念和机制对部件可重用性的支持;

- 面向对象方法同软件重用的关系;

- 软部件的移植性同软件重用的关系,软部件的理解和评估,质量和可靠性问题,代码部件的组合问题等。

二、管理问题的研究

八十年代后期,人们开始认识到管理问题的解决是提高重用效率,发挥重用潜力的关键。管理问题的解决将比技术问题更重要,更困难。

1. 机构组成和机构管理

确立机构组成,包括机构层次、各层次的职能、资源和人员配置等等,以促进和加强在机构范围内的软件重用以及对软件重用项目的管理,如建立重用目标和策略,制定重用规划和决策,包括重用范围、机构行为、合同和资金问题以及人员培训问题;建立设计、重用和程序设计机构标准,并在机构范围内推广使用;为实现上述目标建立控制和变量,确保管理人员和技术人员在项目开发中执行重用/可重用性活动。

2. 机构行为、合同和法律问题

制定有关机构行为,支持和鼓励机构人员积极参与重用活动。通过合同的方式鼓励合同商开发可重用软件以及重用已有的系统,通过合同和法律的方式解决有关合同商潜在的责任以及可重用软部件的所有权问题。

3. 资金问题

研究和开发软件重用费用评估模型,支持管理人员的软件重用决策。在机构范围内建立一种在多个项目开发中共享可重用软部件费用的机制。积累和组织软件开发、维护,包括软件重用的历史资金信息,作为管理人员决策的依据。

4. 软件开发过程模型

选择支持软件重用的软件开发过程模型对于成功地进行软件重用是非常重要的,它不仅可以提高重用效率,发挥重用潜力,而且支持标准化工具和方法的开发。当前支持软件重用活动的软件开发过程模型主要有:

- 通用重用/可重用性模型。该模型以 DOD-STD-2167A 生命周期模型为基础,支持软件开发各个阶段重用/可重用性活动^[1];
- 操作模型,操作模型从可执行的面向问题的说明出发,经一系列变换,生成更为有效的系统实现;
- 螺旋模型。该模型的一个重要特点是在每个螺旋阶段都引进风险分析;
- 面向领域的生命周期模型^[2];
- 支持软件重用的原型生命周期范例^[3]。

三、软件重用的应用分类

软件重用的应用分类状况见表1。

按重用内容,软件重用分为人员重用、产品重用和过程重用。人员重用是指选用经验丰富的人员参与软件开发;产品重用是重用各种软部件和思想,它是当前软件工厂在重用内容上采用的主要形式;过程重用是重用完成各种活动的计划性知识。就重用范围而言,分为垂直重用和水平重用。垂直重用指在一个确定的专用领域内的重用,水平重用则在多个应用领域间进行。从重用方式的角度上看,软件重用可分为规划重用和即席重用。规划重用的重用活动是在项目层次上以一种标准化的方式进行,是系统化的软件重用;即席重用支持若干人的重用活动,重用活动的发生是随机的。在重用技术方面,组合重用是目前人们采用的主要形式,它把已有的软部件视为积木块,用它来构造新的软件系统,生成重用则通过各种应用程序生成器和代码生成器,在说明层次

上进行重用,它一般用在比较小的专用领域内。按重用意向分,黑箱重用指不加修改地对各软部件进行重用,白箱重用则允许对可重用软部件进行修改。黑箱重用机率小,白箱重用应用广泛。产品重用包括思想和软部件重用。思想重用指重用各种形式化的概念,如一组问题的解;软部件重用是重用软件开发阶段生成的软部件,包括正文文档、体系结构、面向对象程序设计里的对象等等。

表1 软件重用应用分类状况

重用内容	重用范围	重用方式	重用技术	重用意向	产品
人员	垂直	规划	组合	黑箱	思想
产品	水平	即席	生成	白箱	软部件
过程					

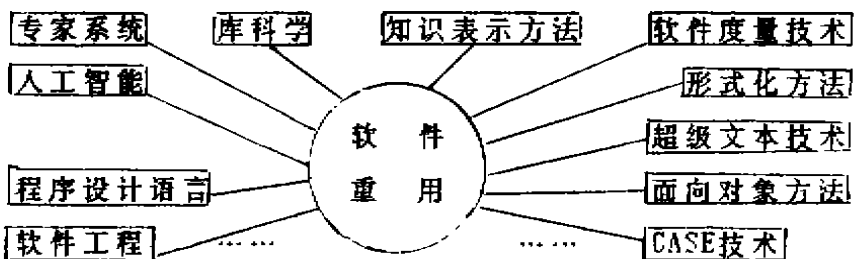


图2 软件重用同其它计算机学科和技术的关系

四、展望

软件重用概念虽然简单,但系统的软件重用并期望从中收到巨大的效益,仍存在不少困难和问题。九十年代,软件重用的研究和发展必将同其它计算机学科和技术紧密联系在一起(见图2);管理问题和标准化问题的解决将日益受到各国政府、机构和标准化组织的重视;在技术问题方面,过程重用将成为主要的研究对象,领域分析和领域工程将构成软件重用研究的一个重要分枝,由 Barry Bohem 等人提出的元程序设计(Metaprogramming)^[4]将得到进一步发展,它可能支持某些专用领域的重用,并具有建立新的、过程驱动的开发范例的潜力。

参考文献

[1] Gianluigi Caldiera et al., Identifying & Qualifying Reusable Software Components, IEEE, Compu-

软件开发

软件重用

16

68-71

关于软件重用*)

TP 311.52

郑明春 张家重 王岩冰

(山东师范大学计算机科学系, 济南250014)

摘要 In this paper, we discuss the basic concepts, reusable software components and essential factors of software reuse. We also analyse its main techniques by software schema reuse, and propose a method of acquiring reusable software schema automatically.

一、引言

1968年, D. Mellroy 在 NATO 软件工程会议上, 首次提出了软件复用的问题, 他倡导生产软件的标准构件用于软件系统的开发。二十多年过去了, 虽然软件复用在其某些方面发挥了较大潜力, 的确能够在一定程度上缩短软件开发周期, 提高软件可靠性, 减少投资风险, 但并不象人们所希望的那样, 能摆脱软件危机, 相反, 软件复用本身又受到了种种限制, 其意义却不尽人意。有鉴于此, 我们有必要对软件复用进行再认识, 弄清何为软件复用, 可复用成分有哪些, 切实可行的复用方式为何, 软件复用的难点在哪里等问题。基于上述目的, 本文首先讨论了软件复用的有关概念和基本技术, 阐述了软构件复用所要考虑的基本问题; 其次以自学习软件自动化系统为例介绍了机器学习用于复用成分的自动获取; 最后指出了软件复用的主要难点和进一步研究的内容。

二、有关概念

2.1 软件复用

所谓软件复用是指利用现有软件成分(资源)来构造新的软件系统。可利用的现有软件成分即复用成分是软件复用的核心, 复用成分的获取、管理和施用构成了软件复用的三个要素。

复用成分的获取有两层含义, 一是为了构造软件系统之便, 将现行软件成分设计(抽象)成可以复用的; 二是在复用成分库中选取用于某具体问题的可用成分。现有工作多集中在提供软件抽象的描述手段方面, 基本上无机器支撑。

在无机器支撑的情况下, 靠手工方式来管理、组织复用成分进行软件复用的一大障碍。Prieto-Diaz R. 在文[2]中提供了一种复用成分库的组织方法, 能够有效地组织和扩充复用成分。

复用成分的施用是获取和管理的目的。施用过程包括根据要求选择抽象的可复用成分, 将其进行适应性修改, 并把它集成到现行开发的软件系统等几个处理步骤。

2.2 复用成分

复用成分不应狭义地理解成软件的源程序代码, 而应广义地理解为一切实用来构造软件系统的成分, 包括软件开发方法、软件需求、设计规格说明、源程序代码与模块或其抽象结构、系统文档, 开发工具与支撑环境、测试和维护信息等。

上述复用成分的定义限定了软件复用的范围。D. Mellroy 提出的软件复用, 其复用成分属程序代码复用的范围, 而 R. Balzer 于1983年提出“代码是不可复用的”。可见, 目前对复用成分的认识尚不统一, 对复用成分不同的理解形成了不同的复用途径与技

*) 南京大学软件新技术国家实验室和山东自然科学基金资助。

- ter February 1991
- [2] W. B. Frakes et al., Representing Reusable Software, Information and Software Technology DEC 1990
- [3] James W. Hooper et al., Software Reuse-Guideline & Methods

- [4] Prieto-Diaz, Status Report, Software Reusability, IEEE Software 3, 1993
- [5] Prieto-Diaz, Domain Analysis For Reusability
- [6] Mark B. Ratcliffe et al., System Development through the Reuse of Existing Components, Software Engineering Journal November 1991