

基于构件的指控软件复用

史浩辉,何 炜

(中国船舶重工集团公司第七一六研究所,江苏 连云港 222006)

摘 要:当前军用指控软件开发效率仍然较低,软件设计和代码开发重复性开发仍然普遍,由此导致软件系统的可靠性得不到保证,开发周期也较长。软件复用是指控软件开发中避免重复开发的重要解决方案,通过软件复用明显可以提高软件开发的效率和可靠性,改善军用软件的维护性和保障性。结合军工科研院所软件开发中流行的构件技术,介绍了软件复用的基本概念和关键技术,就基于构件的指控软件复用技术在指挥与控制软件系统做出应用研究,这些对指控软件复用的工程普及做出了实践贡献。

关键词:指控系统;软件构件技术;软件复用

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2011)02-0159-03

Component-Based Command Control Software Reused

SHI Hao-hui, HE Wei

(Jiangsu Automation Research Institute of CSIC, Lianyungang 222006, China)

Abstract: Current military command and control software development efficiency is still low, software design and development of code development is still widely repeated, resulting software system reliability can not be guaranteed, and longer development cycle. Software reuse is accused of software development to avoid duplication of important solution developed by software reuse can significantly improve the efficiency and reliability of software development, software maintenance to improve the military and security nature. This combination of military institute in the popular software component technology, introduce the basic concepts of software reuse and key technologies, component-based charges on software reuse in command and control software system to make application, these allegations software reuse practice of engineering has made contributions to universal.

Key words: command and control system; software component technology; software reuse

1 概 述

进入新世纪以来,指控系统和岸基指挥自动化系统得到飞速发展,现代指控系统对功能可变、地理分布、操控流程和实时性等特征提出的要求越来越高,同时由于战术需求、战场环境的日益复杂,使得指控软件系统开发、维护、升级、集成等环节越来越复杂,由此带来的开发效率和质量问题一直困扰着工程技术人员。绝大部分工程技术人员一直期望能够像硬件系统那样,将软件模块重新组合起来构建符合新需求的应用软件系统。

软件复用的概念是由 Mellroy 早在 1968 年的 NATO 软件工程会议上提出的^[1],然而软件复用技术对软件产业发展的支持效果不是如人所愿,这种状况是由技术方面、非技术方面中不成熟因素导致而成的。

近年来,随着面向对象技术的出现和成熟,软件复用关键技术瓶颈问题得到解决,在军用指控软件开发中,软件复用的研究也正成为焦点。软件复用的出发点采用充分利用已有成果的模式^[2],以已有的成果为基础,如软件需求分析、软件设计、源代码、单元及系统测试等,开发的关键点着重于各个构成成分上。软件复用使应用系统开发中可以大量避免分析、设计、编码等等过程中的重复劳动,极大提高开发的效率,同时复用已有开发成果可避免了重新开发可能带来的错误,间接提高软件质量。

2 软件复用的概念及分类

软件复用是指重复使用为避免重复开发同样或类似工作产品的过程^[3],可复用软件也仅指有复用目的的特殊软件,它与软件复用概念密切相关。重复使用软件范围很广,反复使用非复用目的而设计的软件,或在一个应用软件里重复使用同一段代码的过程,严格意义上这两类行为都不属于软件复用,进一步说明,针对上述复用类型从不同角度对它进行一个归类:

收稿日期:2010-06-02;修回日期:2010-09-14

基金项目:总装备部大型国防项目(装计字第 610 号)

作者简介:史浩辉(1979-),男,湖北罗田人,工程师,从事指控软件设计与开发、项目管理。

a) 时间的角度: 使用以往历史软件版本作为最新版本开发的基础, 加入新需求和功能, 适应新性能要求进行的少量改进, 这定义为软件维护;

b) 平台的角度: 在某个平台上运行的软件为基础, 适应性修改和平台运行相关的代码部分, 使新软件系统能重新运行, 这定义为软件移植;

c) 应用的角度: 某个软件系统或其中构件用于其他不同应用系统中且新系统具有不同功能和性能。

三种软件重复使用类型的目的都是为了重复使用现有的软件或软件模块, 真正意义上的软件复用是为了支持软件系统或其中构件用于其他不同的应用系统中, 这实现了“为复用而开发”这一原理概念, 使得开发人员能更快、更好地开发新的应用软件系统。

软件复用的类型可以从不同角度进行归类^[4]。根据复用的对象可将软件复用分为过程复用和产品复用。产品复用是指开发人员复用已有的软件构件, 通过构件组装、调试从而得到新系统。过程复用是指复用软件开发通用过程, 使用在一个可复用的应用框架内来自动或半自动地合成所需应用软件系统。根据对可复用信息以什么方式进行复用来区分, 可以将软件复用区分为白盒类 (White box) 复用和黑盒类 (Black box) 复用。黑盒类复用是指对已有软件构件不作任何类型的修改, 直接进行复用。白盒类复用指原有构件不能完全符合用户各种需求, 进行适应性修改后才可满足用户的使用。根据复用层次程度的高低, 可以划分为如下的复用级别:

1) 代码级别的复用。涵盖源代码和目标级代码的软件复用, 其中目标级代码的复用级别处在底层, 运用时间也最悠久。目前开发人员使用的绝大多数的编程语言的运行支撑系统均提供了绑定 (Binding)、连接 (Link) 等功能来支持代码级别复用。源代码的复用级别比目标代码的复用稍稍高些, 是指开发人员在编码时把一些可以复用的代码段通过各种方式如复制、动态库等方式直接复制到自己的应用系统中。

2) 软件设计级别的复用。软件设计比源程序的复用层次级别更高^[5], 这个级别的复用受运行环境的影响相对少些, 使得可复用构件被复用的范围更宽些, 并且适应性的修改极少。软件设计级别的复用有三种方式: 第一种方式是从已有软件系统的设计结果中根据模块特征提取一些具有可复用意义的软件构件, 将这些软件构件广泛应用于新软件系统设计过程中; 第二种方式是把一个已有软件系统的全部设计文档在新的运行或实现平台上进行再实现, 一个成熟的设计能运用于多个具体的过程实现; 第三种方法是软件设计与任何具体的应用没有关系, 为方便软件开发使用大量的可复用的设计构件来提高编程效率。

3) 软件分析级别的复用。这是比设计结果更高级别的复用。可复用的分析级别构件是针对特有领域的具体问题的复用应用^[6], 其受到设计技术及实现环境的影响更少, 所以这个级别的复用范围更大。该级别的软件复用方式是从现有应用系统的分析成果中获取能复用于新系统的软件分析, 如用一份完整的软件需求分析作为设计输入, 生成针对不同类型运行平台和支撑环境及其它实现条件的可复用软件设计。

3 软件复用的核心技术

软件构件是指软件系统中有独立功能或性能的构成成分, 复用构件则是指具有相对独立功能和可复用价值的构件。随着对软件复用实践的深入, 从需求分析源代码、体系架构、软件文档、软件测试数据以及其它对开发活动过程中产生的信息都可以当作复用构件。近几年来构件技术得到飞速发展^[7], 其已发展为支持软件复用的关键技术之一, 并且该学科分支广泛受到业界关注和高度重视。应用软件系统通过构件的整体替代或更换满足不同用户需求, 将新的构件进行构件组装调试从而实现新功能, 这样大大提高了开发生成效率和系统可维护性, 构件自身具有可封装性、可替换性和强扩展性, 软件开发人员在设计、编码等方面提供了很大灵活性。结合军用指控软件开发实践, 现提出工程实践运用的基于构件化开发模型, 见图 1。

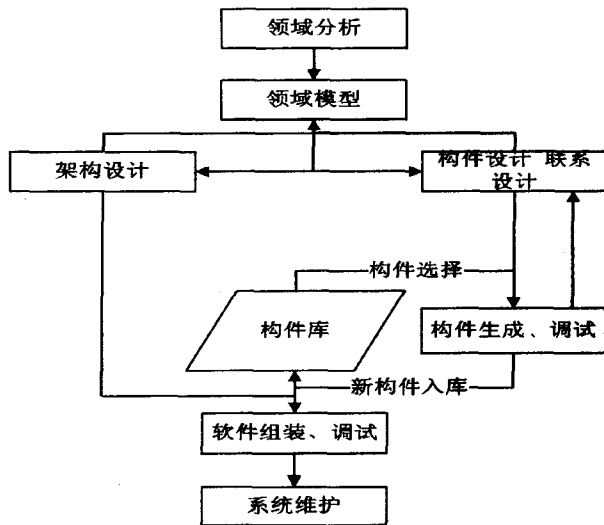


图 1 构件化开发模型

采用构件化开发模型进行应用系统的开发, 过程如下:

1) 行业领域分析。

行业领域分析是待开发领域内的一些共性功能^[8]、共性需求、共性架构以及独立特征、特殊需求进行归纳和标准化描述, 涵盖需求分析和详细设计、概要设计全过程等, 行业领域分析将决定软件应用系统将

实现哪些功能、实现哪些业务处理和数据处理、事务处理。随着软件开发进度的深入,行业领域分析可以随着实际情况进行而具体细化、调整、优化。

2) 行为分析。

行为分析的目的是确定应用软件的总体构架和含括要实现的性能及功能。在行业领域分析的基础上,本过程将详细分析如何具体来实现这些需求。使用面向对象的建模工具和方法后,开发人员可以完全使用典型工具如 UML(统一建模语言)中的用例图来进行行为分析。经过领域建模和行业领域分析^[9],行业领域知识已被按层次、按结构分类,行业领域分析生成的概念结构转化为系统逻辑结构,下一步需要做的工作就是软件构架的搭建。在应用软件系统中,软件构架使得问题域模型变为解域模型,按照这个思路,军用软件系统的软件构架大致可以分为用户交互层、事务处理层、数据处理层、中间层四大层,这些在实际指控项目中非常普遍。

3) 构件设计互联的确定。

单个构件是整个软件架构中基本的事务处理单元和数据处理单元,依据第一步的行业领域分析和领域建模,按照实际开发的需要进行不同粒度构件的分类,并从数学模型、功能、性能等不同角度进行设计和生成构件。在构件与用户界面^[10]、构件与构件的互联过程中,构件标准化接口发挥了至关重要的角色,构件的接口得到确认设计后,其间的互联也将随之明确化,构件接口设计在构件的设计中非常重要,一个具备高度的内聚性和扩展性的构件接口能使软件的伸缩性、维护性得到意料不到的提高。

4) 构件组装和构件复用。

软件构件的接口和互联确认后,下一步进行应用系统的整体开发工作,包括构件的具体开发、现有构件的适应性修改、软件流程实现和连贯、构件的互联等。开发人员将生成的领域构件以及以往的旧有构件加入构件库中,这样一个构件的积累是组装阶段甚至是基于构件化开发方法的重点,这也正体现了软件复用的精髓。开发人员一般可以从已有的企业构件库或商业性机构挑选或购买所需构件,也可以从各种类似应用软件系统中按特征提取。另外,现存的构件也可以进行二次开发,进行适应性修改后以符合新系统的需要。所有的构件完成设计和编码后,末了一步开发人员就是把设计好的构件按定好的系统架构框架组集成为应用系统,一般在主线程中适当编写些调用、冗错等类型的代码即可。

4 软件复用在指控软件中的应用

通过对软件构件及相关技术的研究,将基于软件

构件的复用技术引入到指控系统的设计开发中,结合指控系统设计实践,简化图1的开发模型,基本军用指控软件包含三个主要环节:需求获取、领域分析、软件构件集成。其间关系见图2。

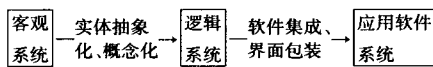


图2 指控软件构件简化开发模型

4.1 需求分析和构件的查找

首先要对业务需求进行分析,然后提炼出需要的构件。在相关领域专家的帮助下,通过不断的交流沟通,将专家的知识提炼成领域知识库,对已有的软件构件进行检索,如:1)通用基本模块,如基本数学函数、统一的坐标变换定义等通用算法;2)统一的指控融合算法、目标综合识别算法;3)各型号指控软件系统专用模块,如航行管理、指挥引导、目标显示、计划拟定等。通过检索后确定重用库构件是否满足知识库需要。

4.2 构件的生成与修改

直接能使用构件库中的构件直接提取使用,有些构件不能直接使用,须对其进行适应性修改,保证构件功能及接口规范的实现。对于新领域的构件则需进行新设计,其基本原则是将构件实现的功能保持粒度最小^[11],通常借助 Use Case 图对系统进行多层次的分析,在提取构件之时,会按构件开发流程进行开发,对软件构件的接口(IDL 文件的设计)、数据算法、数据调用、事务流程进行设计,其中接口的稳定性应予以较大的重视,实际开发中,好的接口设计能适应极强系统功能变化。

4.3 软件构件集成

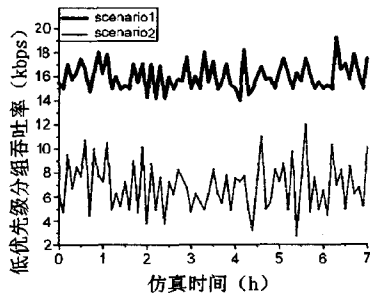
构件的集成是把构件装配成一个整体。需要用开发工具制作一个应用框架^[12],这个框架是整个应用系统的一个骨架,这个骨架在设计时可以通过领域分析方法设计成在某一领域可复用的构架。软件构件集成设计通常采用“配置”的方法,各种构件设计成“基本配置项”和“扩展配置项”,根据系统的实际需求,在一个系统软件界面环境下有选择性的装配。组装结束或构件设计好后,将指控软件移植到相应的分布式环境中进行测试,主要用黑盒测试,使用不同的测试用例查看接口是否满足系统的要求,对出现的问题进行修改满足顾客需要,对于构件内部可以使用开发工具以及专用测试工具进行内部测试。

5 结束语

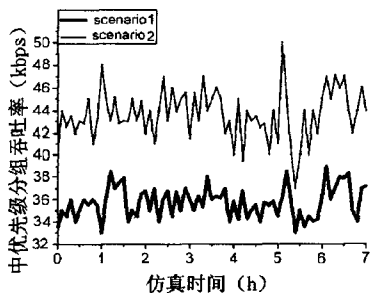
工程实践应用表明,在指控软件开发中应用软件构件复用技术的核心思想能显著提高软件生产效率和

(下转第165页)

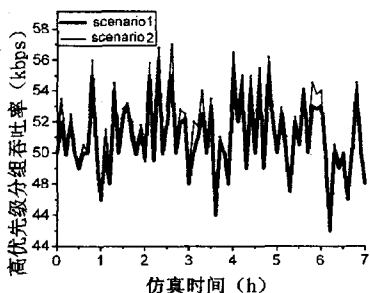
带来端到端延迟增加的问题。所以吞吐率和延迟两者之间如何折衷以获得更好的 QoS 性能,也是需要进一步研究的课题。



(a) 低优先级



(b) 中优先级



(c) 高优先级

图6 高优先级分组吞吐率

参考文献:

- [1] de Pablo D A L. On scheduling models: An overview [M]// Computers & Industrial Engineering. Troyes: [s. n.], 2009: 153-158.
- [2] Liu Hongtao, Cheng Lianglun. Priority-Based Service Differentiation Scheme for Medium and High Rate Sensor Networks [M]// Communication Software and Networks. Singapore: [s. n.], 2010: 392-395.
- [3] 杨志军, 赵东风, 丁洪伟, 等. 两级优先级控制轮询系统研究[J]. 电子学报, 2009, 37(7): 1452-1456.
- [4] 龚跃, 张真真, 黄小珂, 等. 基于动态双向优先级的任务分配与调度算法[J]. 计算机应用, 2009, 29(4): 1131-1134.
- [5] Vukadinovic V, Karlsson G. Video streaming performance under proportional fair scheduling [J]. IEEE Journal on Selected Areas in Communications, 2010, 28(3): 399-408.
- [6] 刘环宇, 侯秀萍. 基于离散粒子群算法的工作流任务调度研究[J]. 计算机技术与发展, 2010, 20(5): 88-91.
- [7] 杨勇, 王雪晶, 陈良臣. QoS 在 IP 中的研究和应用[J]. 计算机技术与发展, 2007, 17(5): 33-36.
- [8] Dovrolis C, Stiliadis D, Ramanathan P. Proportional differentiated services: Delay differentiation and packet scheduling [J]. IEEE/ACM Trans on Networking, 2002, 10(1): 12-26.
- [9] 沈庆伟, 张霖. 基于隧道的 IPv4/IPv6 过渡技术分析[J]. 计算机技术与发展, 2007, 17(5): 171-176.
- [10] 庄正松, 吴家皋, 吴清亮, 陈国风. 互联网基本服务 IPv4/IPv6 过渡的研究与实现[J]. 计算机技术与发展, 2006, 16(8): 13-15.
- [11] 滕腾, 李龙澍. 基于大值堆的自调整粗粒度并行遗传算法模型[J]. 计算机技术与发展, 2007, 17(10): 105-108.
- [12] 李震, 周玮. 一种基于双口 RAM 的环形数据缓存系统[J]. 计算机技术与发展, 2010, 20(1): 201-204.

(上接第 161 页)

产品质量、减少开发成本、提高软件可靠性和可维护性,这些都是军用指控软件迫切需要改善的焦点,指控软件复用技术必将使军用软件产业真正走上工程化的发展道路,军工科研院所应加大软件构件复用技术的研究,为实现软件产业跨越式发展奠定良好的基础。

参考文献:

- [1] 杨美清, 梅宏, 李克勤. 软件复用与软件构件技术[J]. 电子学报, 1999, 27(2): 68-75.
- [2] 杨美清, 王千祥, 梅宏, 等. 基于复用的软件[J]. 中国科学, 2001, 31(4): 367-371.
- [3] 彭思鹏, 孔祥营. 构件技术与指控系统软件开发[J]. 舰船电子工程, 2001(5): 21-25.
- [4] 王刚, 李锋. 基于可重用构件库的机载火控软件的开发与管理[J]. 光电与控制, 2003(1): 40-45.
- [5] 黎姬. 基于构件的软件复用技术应用研究[D]. 重庆: 重庆大学, 2008.
- [6] 张海藩. 软件工程[M]. 北京: 人民邮电出版社, 2002.
- [7] 郑红, 李师贤. 可重用的分布式软件构件模型分析[J]. 计算机工程与应用, 2002(15): 68-71.
- [8] 赵池龙, 杨林, 陈伟, 等. 实用软件工程[M]. 北京: 电子工业出版社, 2006.
- [9] 麦克克劳埃. 软件复用技术: 在系统开发过程中考虑复用[M]. 北京: 机械工业出版社, 2003.
- [10] OMG. OMG unified modeling language specification [EB/OL]. 2005-02. <http://www.omg.org>.
- [11] OMG. The common object request broker: Architecture and specification [EB/OL]. 2005-05. <http://www.omg.org>.
- [12] Microsoft. DCOMtechnicaloverview [EB/OL]. 2006-07. <http://www.microsoft.com/Intserver/library/dcomtec.exe>.