

软件重用技术及在航天飞行软件中的应用

刘 杰

(北京系统工程研究所)

摘 要 分析软件重用的主要技术问题,介绍国内外航天飞行软件重用技术研究和应用状况,涉及飞行软件的代码重用、任务重用、代码自动生成、软件生产线等技术。据此,探讨飞行软件重用技术的主要研究方向,为飞行软件重用技术研究与应用提供参考信息。

关键词 软件重用 软件生产线 航天飞行软件

1 引 言

软件重用是指重复使用“为了重用目的而设计的软件”的过程。可重用软件是指为了重用目的而设计的软件。

软件开发存在大量可重用的通用基本构件和领域共用构件及相关知识,这些构件和软件知识如果经过事先有目的的重用设计和遵循一定的标准,则可以在不同的软件项目中多次重复使用。据统计,软件项目中大约有 60~80%的部分是可以被重用的。因此,研究软件的重用非常有意义。

通过软件重用,在应用系统开发中可以充分利用已有开发成果,消除包括分析、设计、编码、测试等在内的许多重复劳动,提高软件开发效率,同时,通过重用高质量的已有开发成果,能避免重新开发可能引入的错误,从而提高软件的质量。

通过长期大量的航天飞行任务,各国航天机构在飞行软件研制方面积累了许多成功经验,同时也积累了许多有可能被重用的飞行软件资产,由此,飞行软件重用技术的研究和应用一直受到航天机构的重视,研究内容包括飞行代码重用、飞行任务重用、飞行代码自动生成、飞行软件生产线等方面,并取得了一定成果,以达到节省研制成本、缩短研制周期、提高软件可靠性和安全性的目的。

本文描述了软件重用的主要技术问题,介绍和分析目前国内外主要飞行软件重用技术,指出存在的主要问题,探讨飞行软件重用技术的研究方向,为

软件重用技术在飞行软件中的应用研究提供有益的参考。

2 软件重用技术分析

软件重用涉及多方面的技术,这些技术也在不断的发展。核心技术主要有以下几方面的内容。

2.1 领域工程

领域是指一组具有相似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阶段:

- ①通过领域分析获得领域模型。
- ②通过领域设计获得领域软件构架(DSSA)。
- ③通过领域实现获得将需求翻译到由可复用构件组成的软件系统。

2.2 重用库

软件重用库由可重用资源存放库、查询库中资源内容的查询接口、重用资源的表示方法,以及支持变更管理和质量评估的工具等部分组成。重用库实现技术与一般关系型数据库实现技术不同,可能要采用不同的索引方法、要采用与传统线分类方法不同的面分类法、要采用自由文本的全文检索技术等。

2.3 软件生成方法

生成式重用是软件重用的重要方法,且与领域工程过程紧密相关。对领域知识和相关的系统构建知识进行编码,通过领域相关的应用生成器生成新系统所需要的用目标语言编码的程序。生成过程可能是完全自动的,也可能有人工的干预。

生成式重用的另一个重要进展就是元编译器。

元编译器是一种产生应用程序生成器的程序,能帮助领域相关应用生成器的开发。

2.4 基于构件的开发

基于构件的软件工程对软件重用产生了重大影响,引起了人们的广泛兴趣。目前最主要的构件开发、集成和部署技术有对象管理组织(OMG)的CORBA 构件模型 CCM、Sun 公司的 EJB 技术以及微软的组件对象模型 COM+。这些技术提供了重要的中间件平台,依靠这些平台可以开发可重用的构件、实现构件的动态组装和集成、部署和运行构件化应用。这些构件平台基于容器模型,构件容器不仅负责构件生命周期的管理,还提供一系列的构件服务,如安全、可靠、容错、负载均衡等,并以某种类似面向方面的程序设计(AOP)的方式自动编织到功能构件的切入点,从而实现构件的可信封装与组合。

2.5 编程语言

编程语言的发展与软件重用密切相关。现在,编程语言为程序员提供了更大粒度的程序结构,如模块、类甚至框架等,编程语言的抽象层次更加提高,更加接近人的语言,更加关注领域,因此也更加易于使用。

编程实践中的接口与实现分离技术已被广泛采纳,该技术与设计模式等的结合,使得程序组成成分之间耦合程度更小、灵活性更强,更加有利于重用。

2.6 CASE 技术

随着软件工程思想日益深入人心,以计算机辅助开发软件为目标的 CASE 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。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。

2.7 重用度量

与软件重用概念相关的重要问题是如何对软件重用和可重用性进行度量,如何确定重用对软件质量和生产率的影响,这就涉及到重用的建模、度量和试验等一系列问题,包括重用类型分类模型、重用库度量、费效模型、成熟度评价模型、失效模式模型、可重用性评价模型、重用度量元及数量确定等。所谓模

型,就是要建立作为变量的度量元之间的明确关系,度量则是对软件产品或过程属性的定量化测量,而试验则是建立一组度量元对另一组度量元影响的过程,如软件重用与软件质量、软件生产率的关系,软件重用对软件生产成本、投放市场时间、项目完成时间等的影响。

2.8 软件重用过程

软件过程是软件生存周期内为达到一定目标而必须实施一系列相关过程的集合。一个定义良好的软件过程对软件开发的质量和效率有着重要影响。高质量的过程能产生高质量的产品。软件生命周期过程的质量直接影响所生产的软件产品质量。对软件重用来说,软件开发人员需要一个标准的过程去识别、分析、设计、实现、测试、部署和维护可重用的软件构件,以及基于可重用构件的软件开发。

IEEE Std.1517 定义了软件重用过程最基本的要求,它为软件重用定义了一个标准的过程框架,该框架可以融入任何的软件生命周期过程,如 ISO 12207 软件生命周期过程。

3 飞行软件重用技术

飞行软件重用技术研究可分为基于航天器飞行时间和事件的软件自动生成技术研究和用于空间快速反应的软件生产线研究。一些研究项目已取得实质性进展,并部分应用于航天飞行任务,代表了现阶段飞行软件重用技术的研究及应用方向。

下面通过两个实例介绍软件重用技术在航天飞行软件中的应用情况。

3.1 飞行控制系统软件自动生成

2006 年北京航天自动控制研究所发表的相关研究结果认为:“对于控制系统飞行控制软件而言,存在着大量的继承性和被继承性,已经形成了相对稳定的基本结构。他们的工作范围、承担的功能、采用的技术、使用的理论基础,存在着类似和相同、相通的部分。而且,航天事业经过多年的发展,有许多经过多次飞行试验考验的成功型号软件,这些软件的模块、结构,甚至它本身,如果能在新的型号软件中重用,将节省软件研制经费,缩短研制周期,而且大大提高软件的质量和可靠性。”同时,“软件的重用是

建立在已有可被重用的软件的基础上的, 可被重用的软件是在设计中专门为重用而设计的。”基于这些原则, 北京航天自动控制研究所提出以下三个解决方案, 并实现了飞行软件原型系统。

(1) 模块化通用设计

模块化通用设计是在对任务进行功能分解的基础上, 设计、实现、测试可被重用的通用模块, 形成可重用模块库, 如图 1 所示。

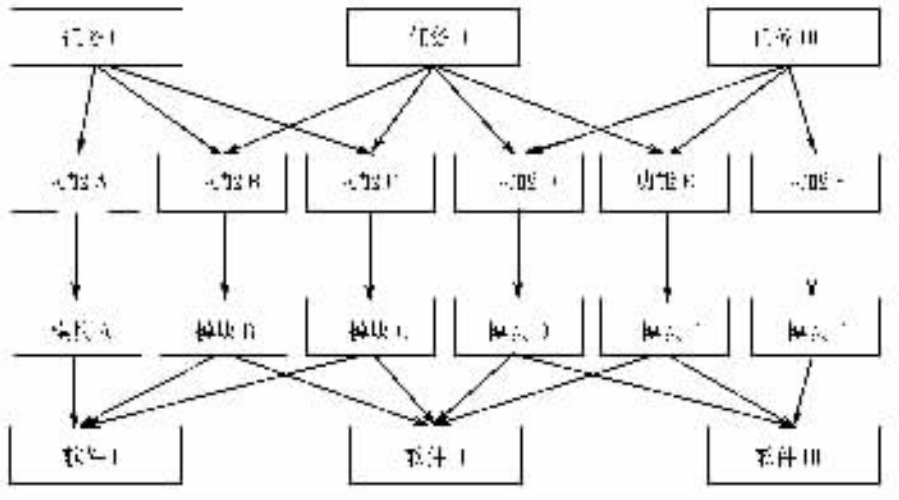


图 1 模块化通用设计

(2) 任务级通用化设计

任务级通用化设计是在模块化通用设计的基础上, 提出一种基于控制时序的、通用的飞行控制软件框架模型。

飞行控制软件框架模型的基本原理是, 飞行软件按照时间轴顺序执行, 当运行到相应的时间顺序交叉点时, 即调用相应的模块代码, 完成该事件功能, 如图 2 所示。

针对一个飞行控制软件任务书, 设计飞行任务时间-事件图, 依据飞行控制软件框架和通用模块库, 实现该任务的飞行控制软件原型。

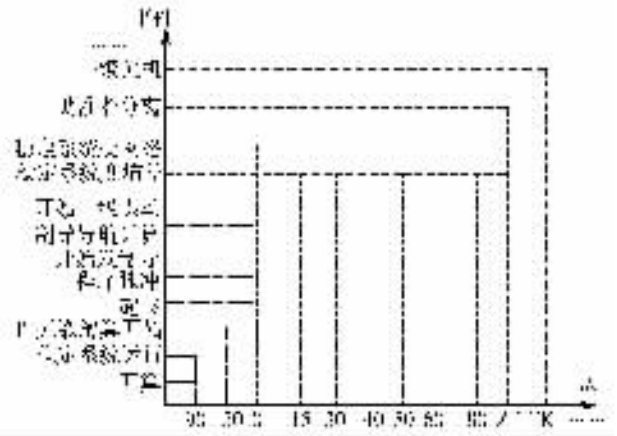


图 2 飞行任务时间-事件图

(3) 自动生成的实现

自动生成是指在模块化通用设计、任务级通用化设计的基础上, 系统设计人员和软件设计人员共同进行飞行控制软件设计, 确定任务的事件、弹道数

据、导航参数和控制律等信息, 建立时间-事件表, 并输入到飞行控制软件自动生成工具中, 自动生成飞行控制软件原型, 工作原理如图 3 所示。

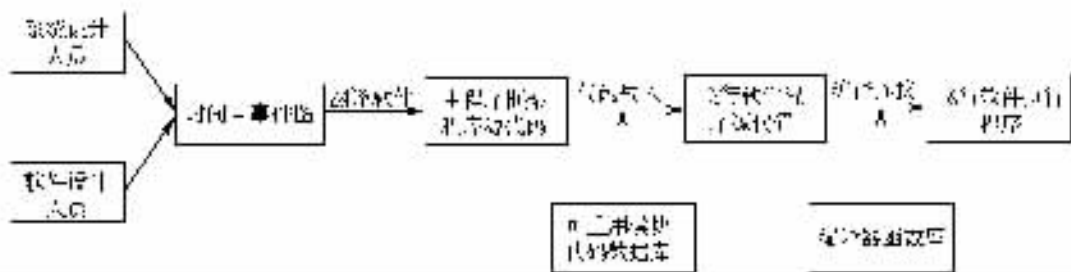


图 3 飞行控制软件自动生成原理

3.2 空间快速反应核心飞行系统

戈达德航天飞行中心飞行软件部长期从事飞行软件的研制工作,包括飞行软件重用技术研究,主要有飞行软件生产线研究,以及用于空间快速反应的核心飞行系统研究。

戈达德航天飞行中心认为,核心飞行系统是一条基于构件的软件生产线,也是一个与平台无关的飞行软件环境。核心飞行系统具有许多先进实用的功能,代表了一类飞行软件生产线的应用和研究方向。核心飞行系统的主要功能包括。支持快速装配、支持动态发现(Run time discovery)和集成、支持自动生成文档和测试用例:

- 支持从构件库中选择构件并快速集成,支持生成文档、测试用例,生成遥测/命令数据库。
- 支持生成基于运行平台的快速原型,包括:开发符合核心飞行系统应用程序接口的构件,快速集成和测试;支持构件修改,以及在系统运行中的再装载。
- 消息中间件具有动态网络机制,支持按照系统需求插入/删除硬件设备和测试构件。
- 在桌面系统开发的应用系统,可以通过再编译、连接、装载到目标机执行。

· 构件库支持多平台,并可扩充,其中,嵌入式系统包括 VxWorks、Linux、RTEMS、LynxOS 操作系统;桌面系统包括 Macintosh、Linux、Windows (Cygwin)。

核心飞行系统由一个可重用的核心飞行执行程序(core flight executive,CFE)、软件构件库、一个集成开发环境(Integrated Development Environment, IDE)组成。

(1)核心飞行执行程序(CFE)

可重用的核心飞行执行程序(CFE)具有层次体系结构、标准的中间件/总线、标准的应用程序接口,如图 4 所示,主要特点如下。

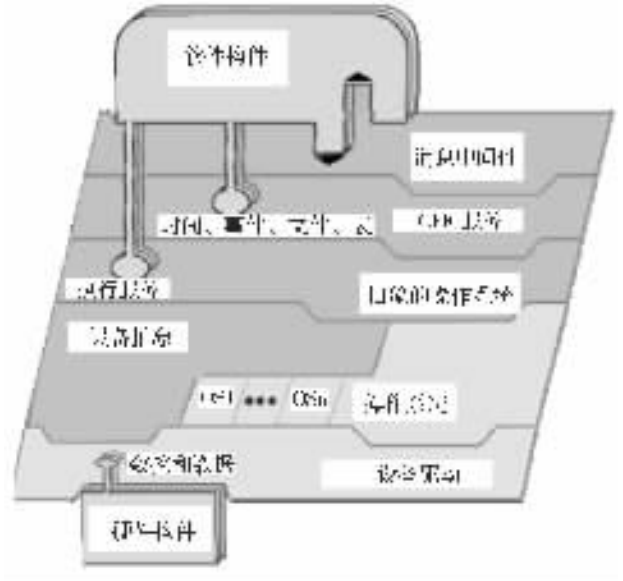


图 4 核心飞行执行程序

- 至关重要的软件分层:可以改变一个软件层的实现,但不会影响其它软件层及其接口。
- 先进的消息处理:淘汰了人工配置飞行软件,可以自动配置飞行软件软件构件和硬件构件。
- 标准、抽象的程序接口:飞行软件受到飞行硬件、操作系统、应用程序改变的影响最小化。

(2)可重用软件构件库

软件构件库用于管理和提供可重用构件,实现即插即用(Plug and Play),软件构件库中主要包括以下内容,如图 5 所示。

- 抽象的飞行软件的公共功能,及其构件和服务。
- 构件的测试和证明信息,以及相关文档。
- 建立飞行软件所需的核心服务、可重用的构件、一个任务的专用构件、改写的遗产构件。
- 可被嵌入到即插即用硬件设备的软件构件。
- 可在中心处理器执行并与插入设备相连接的软件构件。

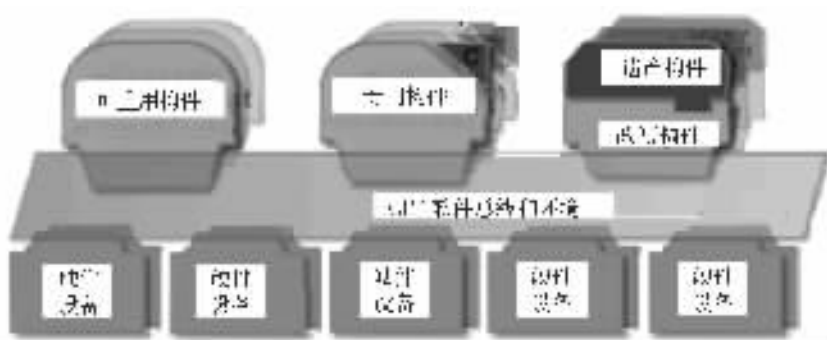


图 5 可重用软件构件库

(3)集成开发环境(IDE)

集成开发环境(IDE)提供配置管理、需求跟踪、开发等工具,提供开发标准主要特点如下:

- 集成开发环境是可扩充的。
- 软件工具通过开放源程序的 Eclipse 平台集成,并支持多软件工具厂商。
- 支持图形化界面的环境配置与部署。
- 提供软件需求、设计、编码、单元和测试功能,及其用户指南,提供用 XML 描述的遥测/命令数据库。

4 主要问题和研究方向

4.1 主要问题

根据上述研究和分析,我们认为,在飞行软件重用技术方面目前存在以下问题:

- 可重用飞行软件的分析、设计、实现技术尚未成熟,飞行软件设计主要针对一个任务,很少考虑未来多航天飞行任务的重用需求。
- 飞行软件的软件体系结构、接口规范、信息交换规则有待标准化。
- 重用飞行软件(构件)对新飞行系统功能、性能、可靠性、安全性等多方面综合影响的评估技术尚未成熟,未能有效地实现有计划、大规模的软件重用。
- 可重用飞行软件的动态连接、装载技术尚未得到实际应用。

4.2 研究方向

为了保证航天飞行软件的可靠性和安全性,实现空间快速反应,增强国家的空间竞争能力,在现有飞行软件重用技术的基础上,有必要进一步加强可重用飞行软件标准、开发理论与技术、验证理论与技术、开发管理的研究,以有效解决飞行软件重用中的关键性技术问题。

(1)标准研究

进一步研究和完善飞行软件重用标准,主要包括:可重用飞行软件的分类、可重用飞行软件的开发过程模型、可重用飞行软件的验证和确认、可重用飞

行软件的接口与信息交换、可重用飞行软件安全保障等的标准化。

(2)可重用飞行软件开发理论与技术研究

可重用飞行软件开发理论与技术研究的主要内容包括:可重用飞行软件的分析理论与技术、可重用飞行软件的设计理论与技术、可重用飞行软件的新型体系结构及理论与方法、可重用飞行软件的动态连接与装载技术、可重用飞行软件可靠性的实时评估与预测的理论和方法等。

(3)可重用飞行软件验证理论与技术研究

可重用软件验证技术应涵盖整个软件生命周期,以提高可重用软件的可信度,主要研究内容包括:飞行软件可重用度量技术、可重用飞行软件的可靠性验证理论与技术、可重用飞行软件安全性验证理论与技术、可重复发射航天器软件及计算机系统安全验证技术、现货软件的验证与安全性证明、可重用飞行软件的测试要求与技术等。

(4)可重用飞行软件开发管理研究

进一步加强对可重用飞行软件开发的管理,有计划地进行软件重用,主要研究内容包括:飞行软件生产线开发过程管理、可重用飞行软件库管理、现货软件提供方的资质证明与合同管理、可重用飞行软件(包括现货软件)的运行与维护管理等。◇

参 考 文 献

- [1] William B. Frakes and Kyo Kang, Software Reuse Research: Status and Future, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL.31, NO.7, JULY 2005
- [2] 杨美清,梅宏,李克勤.软件复用与软件构件技术.电子学报,1999,27(2):68-75
- [3] 孟小丰,马伟华.控制系统飞行控制软件重用技术研究.航天控制, Vol. 24, No.6, 2006年.
- [4] Jonathan Wilmot. NASA Goddard Space Flight Center. Implication of Responsive Space On the Flight Software Architecture. 2006年4月.
- [5] Jane Marquart. NASA Goddard Space Flight Center. Flight Software Technology Roadmap.
- [6] 军用软件安全性指南.中国人民解放军总装备部, GJB/Z 142-2004.
- [7] 军用软件测试指南.中国人民解放军总装备部, GJB/Z-141, 2004.