

Advanced Technology for Command and Control Systems Engineering

指控系统工程先进技术

(美) Stephen J. Andriole 主编
刘山 姜海格 康立军 主译

国防工业出版社
<http://www.ndip.cn>

Advanced Technology for Command
and Control Systems Engineering

指控系统工程先进技术

(美) Stephen J. Andriole 主编

刘山 姜海格 康立军 主译

国防工业出版社

著作权合同登记 图字:军 — 2003 — 012 号

图书在版编目(CIP)数据

指控系统工程先进技术/(美)安德里奥(Andriole, S. J.)主编;刘山等主译. —北京:国防工业出版社, 2005.1

书名原文: Advanced Technology for Command and Control Systems Engineering

ISBN 7-118-03660-9

I. 指... II. ①安...②刘... III. 系统工程(军队指挥) IV. E1

中国版本图书馆 CIP 数据核字(2004)第 114452 号

Advanced Technology for Command and Control Systems Engineering, Edited by Stephen J. Andriole © AFCEA International

本书中文版由 AFCEA International 授予国防工业出版社独家出版发行。版权所有,侵权必究。

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

腾飞胶印厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 24¼ 569 千字

2005 年 1 月第 1 版 2005 年 1 月北京第 1 次印刷

印数:1—2000 册 定价:56.00 元

(本书如有印装错误,我社负责调换)

国防书店: (010)68428422 发行邮购: (010)68414474

发行传真: (010)68411535 发行业务: (010)68472764

前 言

指挥控制系统对国家安全的重要性已经得到广泛的承认。上到国家最高指挥机构,下至部队的一名排长,指控系统就是指挥者用来领导其部队完成某项任务的手段。指控系统是由指挥者为控制其部队所使用的人员、设备、通信、工具和程序组成的。历史提供了许多实例,说明一位出色的指挥者借助于合适的指控系统支持成功地完成了使命。但是历史也提供了通常更广为人知的一些例证,表明指控系统对任务的结果起到了明显的反面作用。人们不加思索便会想到珍珠港事件、“普柏洛”号事件(Pueblo)、“自由号”事件(Liberty)和“文森斯”号(Vincennes)事件等。

指控系统的重要性从国防预算用于这方面的份额上可以反映出来。最近几年里,用于 C^3 的预算已超过国防预算的7%。而作为指控不可分割的一个要素的情报功能的预算是单列并保密的。随着国防预算的紧缩,如何有效地使用 C^3I 经费变得越来越重要。

指控系统不论从执行任务的有效性还是从效费比上看,重要性都是公认的,因此为指控系统制定出一个知识的基础至关重要。这个知识基础应当为我们提供设计和评估指控系统所需的理论、工具和技术。

在乔治-梅森大学新成立的指挥控制通信和情报(C^3I)卓越中心的目的之一,就是帮助建立这个指控的知识基础。目前由弗吉尼亚发明技术中心、国防通信局、CECOM、ASD(C^3I)、MITRE、PRC/ATI、洛克希德、AFCEA国际和AFCEA NOVA所赞助的研究计划,是一项范围广泛的研究开发计划。该计划包括需求工程、建模和仿真、自动化计划、决策支持系统、先进通信技术和认知工程方面的项目。浏览一下本书目录便可知道,所有这些议题本书都有涉猎。卓越中心的第二个目的是制定一个教学计划,提供一套指导 C^3I 硕士学位的专门课程。该课程开发的结果之一将是关于 C^3I 各个方面的系列丛书。首部面世的关于指控的著作就是由在该中心的创建过程中起过重要作用的斯蒂芬·J·安德里奥教授编辑的。

本书会吸引广大的读者,包括系统需求的制定者、系统工程师、软件系统开发者以及潜在的用户。本书中所讨论的指控系统工程先进技术能够使该领域的工程技术人员创造出更加有效的指挥控制系统来。

乔治-梅森大学
信息技术、电气和系统工程系名誉教授
C³I卓越中心主任
海利·L·范特里斯

目 录

绪论	1
第一部分 指控信息系统工程	7
指控系统中的软件重用	8
指控软件系统需求工程	19
需求不断变化的技术规格书	29
基于超媒体的需求工程设计	39
指控的系统级需求规格书:风险管理与降低易变性的方法	52
灵活的多学科指控信息系统工程生命周期	60
第二部分 指控建模与仿真	83
指控培训的战争模拟、建模与仿真	84
假设分析型作战仿真对制定军事计划的支持	90
对于指控系统模拟士兵在回路的要求	109
军事指挥控制中人的效能	119
效能评估框架	139
作战 C ³ I 仿真与建模的新技术和新工具	157
第三部分 智能系统技术	165
在指控环境中由专家系统向智能系统技术的过渡	166
具有可证实特性的自动化对抗性计划搜索程序	175
指控用实时专家咨询系统评估:一般方法和个案研究	193
用于指控的智能系统	202
用于情报报文分析的自然语言处理系统	211
第四部分 辅助决策与支持系统技术	219
陆军战区计划和反恐危机管理的群体决策支持系统样机	220
ISMAUT 及其相关技术用于国防通信决策支持	242
评价决策支持系统:在研发过程中融入综合评价方法	256
下一代决策支持的先进信息技术	267

第五部分 指控需求分析与建模	281
MECCA:指挥和控制资产的多方面评估	282
军事计划编制和技术	292
总统还是计算机:谁将最终决定 SDI 的使用.....	300
利用分析方法辅助进行训练资源的有成本效益的分配	306
第六部分 先进的通信技术.....	315
陆军战术指挥控制系统:集成的 C ³ 环境	316
采用异构系统进行集成分布式计算	327
视频远程会议:应用与发展趋势	335
第七部分 指控系统工程中的问题	341
计算机辅助系统工程在多学科指挥与控制中的应用	342
系统设计重点的转移——从机器到人	351
认知系统工程的前景	362
多学科系统工程培训和教育:设计和挑战.....	371
论文作者名单	379
关于作者	383

绪 论

本书力图对大量零散的材料进行综合。综合工作是多学科系统工程(MSE)的本质。指挥和控制(C²)及其主要的支持系统——通信系统、情报搜集和处理系统、平台和武器——是复杂的,动态的,从概念、解析和操作方面看很大程度上是难以捉摸的。在同一种应用中,神经网络、专家系统或线性规划,到底哪一种技术更能解决问题,存在数不清的指控“模型”,永远有争议的效用测量手段和激烈的争论。在许多方面,各种 C² 理论、模型、工具、方法和技术组成了谚语中所说的(圣经中没有建成的)“巴贝尔”通天塔。

本书的目的是要理出一个头绪来。核心概念是“系统工程”,尽管只有少数几个作者用到这个术语。本书还力图对某一个特定时间的技术进行全面的描述,随着技术和手段不可阻挡的进步,这种描述也势必要更新。本书反映了该领域的广度和深度,介绍了指控系统设计和开发方面丰富的方法和光明的前景,也提出了尚未解决的问题以及面临的挑战。

正确认识系统工程

什么是系统工程?为什么它对指控系统的设计、开发和部署至关重要?

系统工程是一种以行为学、计算机、工程、管理和数学科学为基础的解决问题的方法。根据至少一个资料来源(1986年 Chambers 引用的国防部 1974 年的定义),系统工程方法是一种:

“……能将作战需求转化为系统性能参数的描述和优化系统配置的各种活动和决策的逻辑顺序……”。

同一资料来源认为,系统工程方法有以下 9 个重要阶段。

1. 使命需求分析;
2. 功能分析;
3. 任务分配;
4. 综合;
5. 后勤保障工程;
6. 寿命周期成本分析;
7. 优化;
8. 生产工程分析;
9. 生成系统规格书。

其他资料来源(1988年 Eisner)则把系统工程方法分析得更细,包括以下“要素”:

1. 需求分析;
2. 需求分配;
3. 功能分析;

4. 功能分配;
5. 技术指标分析;
6. 技术指标分配;
7. 技术指标开发;
8. 初步设计: A. 系统级;
B. 分系统级;
9. 接口定义;
10. 工期制定;
11. 预先成本分析;
12. 技术性能测量;
13. 不同方案权衡分析;
14. 预先计划产品改进;
15. 最终设计: A. 系统级;
B. 分系统级;
16. 进度调整;
17. 成本调整;
18. 生产;
19. 编码;
20. 预测试;
21. 排除故障和重新配置;
22. 测试和集成;
23. 更新调整: A. 进度;
B. 成本;
C. 技术性能测量;
24. 文件;
25. 培训;
26. 生产。

可以看出,系统工程方法如能正确实施,需包括一系列步骤,当这些步骤合在一起时就能解决一个特定的问题。系统工程方法从性质上和用途上看是通用的;具体的应用领域如指控,则借用通用方法,并从通用方法中按具体问题需要提取一些要素。

多学科系统工程工作者的定位

了解不同层次的人员,如系统分析员、系统工程师、“其他”工程师、编程人员和学习一般系统理论的大学生之间有什么区别,是很重要的。让我们从编程人员说起,因为他们最经常——不管是对还是错——与指控信息系统的设计和开发联系在一起。当编程人员面对整个信息系统开始开发软件时,他们从本能上就“自底向上”做起。他们会把编程看成是完成系统工作的某一个或某一些方面。他们从微观角度看系统,只要涉及他们个人的软件系统的设计和开发,他们就会改变角度看问题。好的编程人员都是自觉不自觉地采取一种系统方法来完成整个系统难题中由他负责的那些模块程序的设计和开发。这种不

太恰当的处理问题的方法乍一看似乎不好理解,但仔细想来又确实很有道理。编程人员本来就不负责整个系统,而只是负责系统中的一部分,只有当你负责一套相互关联的部件,把它们组合在一起应能完成某项特定功能时(如一个计算机程序),采用“自顶向下”的系统方法才真正有意义。

各计算机编程人员在系统设计和开发过程中的作用是由系统分析员自顶向下分多个级别来评定的,系统分析员关心的是所有部件怎样更好地组合在一起。米尔斯(Mills, 1985)则走得更远,将系统设计师和“系统总体设计师”区分开来。系统总体设计师对整个的设计具有“控制”权,至少在项目最初阶段。

“其他工程师”则是非系统工程师的总称,他们看待系统也是自底向上的,除非面对他们自己负责的系统。机械工程师关心的是液压系统或他们选用的设备结构上的完整性,电气工程师则关心那些保障大型装备运转的电路,而化学工程师可能会专注于某些能导致下一步过程的化学反应。所有这些工程师关心的都是他们的“小”系统,只有系统总体设计师才关心所有那些小系统组合起来匹配的程度如何。

系统工程师关心每一样东西。他是出色的自顶向下解决问题的专家,系统工程师应用系统方法对复杂的系统进行设计和开发。系统工程师是总体设计师,是分析师,是包罗万象的工作者。

系统工程师的工作是进行领导、集成、综合、先后排序、度量、试验、制作样机和监控。他们遵守一套既明确但又不精确的工作程序。系统工程既是一门应用科学,也是一门艺术。不论你相信与否,毕竟有一个系统工程的“标准”定义(MIL-STD-499A):

“系统工程是在如下方面的科学和工程成果的应用:(a)通过定义、综合、分析、设计、试验和评估的反复迭代过程的应用,将作战需求转化为系统性能参数的一个描述和一种系统配置;(b)对相关技术参数进行综合,确保所有的物理、功能和程序接口的兼容性,以便达到整个系统定义和设计的最优化;(c)将可靠性、维修性、安全性、生命力、人的和其他因素综合进整个的工程工作之中,以满足成本、进度和技术性能的目标。”

政府的系统工程管理指南(DSMC, 1986)用图形描述了这一过程,见图1。该过程的各个步骤都是通用的,即建议系统工程师应如何设计和开发复杂的系统。更确切地说,该图告诉人们应如何“设计”各种设计。

系统工程师应用这个生命周期来解决各种问题。他们可以是城市系统工程师、运输系统工程师、污水系统工程师、航天站系统工程师、指控系统工程师。所有这些应用都要求对图1所示的通用生命周期进行详细的判读,尽管它们都有一些重要的东西是共同的。

首先是完成一个通用的或特定问题的系统工程生命周期所需的工具和技术的性质:它们总是多学科的。今天,要想设计和开发一个系统而不借助多学科的观点是根本不可能的。生命周期的所有阶段要求采用多种方法和途径,不可能在单一的(甚至是几个)学科范围里全部找到。有意思的是,许多关于系统工程的文章都没有强调系统工程的这个方面。第二个共同点是迭代性。复杂的系统从来都不是一次就能设计出来的。第三个重要的共同点是需要进行效益和成本/效益分析。第四个共同点则是所有的设计决策都必须与有效需求进行协调。

所有这些共同点存在于每一个系统工程项目中。我们已经建立了行为、管理和计算机科学对需求分析的相互关系。样机制造意味着迭代,而好的系统工程师总是不断从各

种复杂设计中寻找效费比好的解决方案。但更重要的是,系统工程师是受一个自顶向下的需求规格书牵引的,如图 2(DSMC,1986)所示。系统工程师是使用广泛领域和学科的方法、工具和技术来进行系统的设计和开发的。

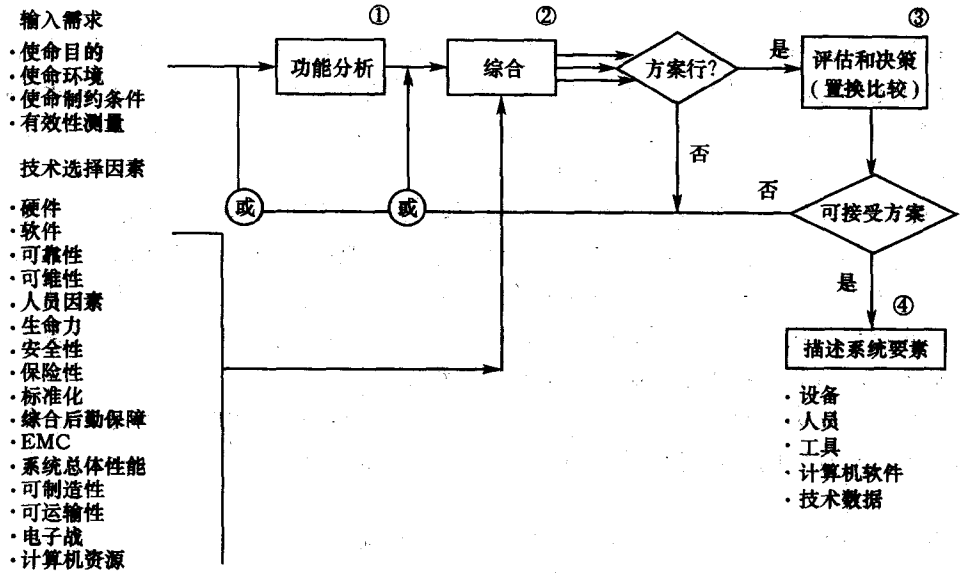


图 1

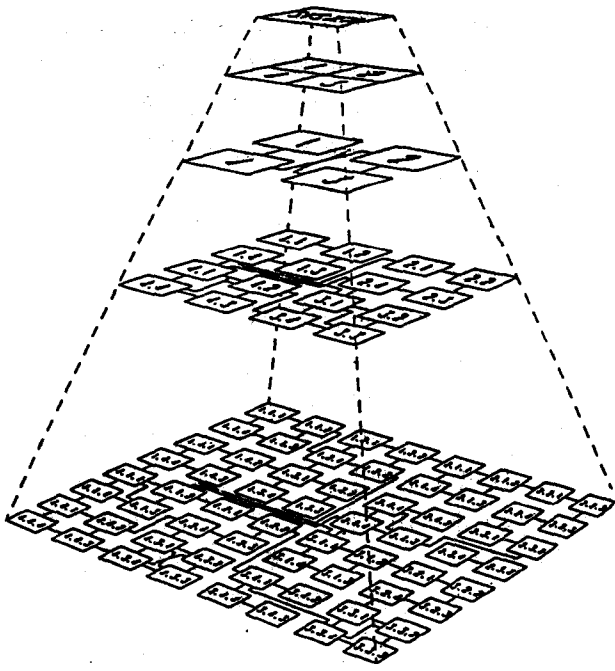


图 2 系统需求自顶向下的按级分解

多学科系统工程的思想方法是成功地进行系统设计和开发的关键。它所寻找的信息是系统的用途、系统开发背后的动机、用户对性能的期望和管理上的限制条件。多学科系统工程从不规定需求,从不处理试验或评估问题,从不开发与需求无关的维修计划。一个系统要存在的最大理由应是一个过滤器,所有设计的决策都应该顺利地通过。

多学科系统工程的关键指导原则

多学科系统工程的关键指导原则,即“秘诀”是什么?

第一个是“需求的功能层次分解”。多学科系统工程需要对多层次的需求进行识别、定义和确认,弄清各种需求的相互关联性,如何通过设计来满足。

另一个关键原理是迭代,或者说认识到不可能通过一个系统概念一下子就抓住所有的需求。多学科系统工程在设计和开发过程中的所有阶段都需要反复迭代。快速的和不那样快的制作样机都需要根据用户需求、各种系统概念、软件需求、试验和评估,甚至文件来进行反复迭代。

再一个关键概念是综合,或将分散的组件集成于一个比各组件总和更大的、相关效费比更好的整体之中。正是在这一点上,一般的系统工程和专用的多学科系统工程变成了一种艺术。创造性的综合过程已有描述(史密斯,1988),但是没有正式发表。虽然已有建议说明设计要素综合的内容是什么,在哪里进行综合,但尚极少有教导信息系统工程师如何进行综合的指南。不过,综合仍然是多学科系统工程方法的一个重要组成部分,它很适用于多学科观点。

权衡分析永远是多学科系统工程方法的一部分。由于需求、限制条件和设计备选方案之间的各种组合,需要不断进行评估、排序和试验。每制作一台样机都必须进行权衡分析。每一次决定用一段新的设计程序代替原有的时候,就是在进行权衡。设计挑战越复杂,需要做的分析权衡就越多。面临的限制条件越多,分析权衡就变得越加重要。

最后,是尽可能保持多学科特性,这是最基本的需要。它要求具有跨社会学、行为学、计算机、数学、工程学、管理学和物理学等学科择取所需知识的能力。然而请注意,不能期望系统工程师懂得所有这些领域和学科的全部知识,但是至少应了解现存的唾手可得的这部分。对人类知识成果采取接受的态度——不论它们来自哪个种族、信仰或宗教——就是从思想上敞开大门,学会创造性地解决问题。如果对别人的方法、工具和技术一概采取关门主义,那就会大大减少成功的机率。

指控系统工程用的先进技术

本书具有多学科系统工程的价值。同时它承认需要新的思想、观念和创新的应用技术。

第一部分重点介绍指控信息系统工程。讨论了系统、用户和软件的需求分析,提出了新的思路和方法,还介绍了一种新的信息系统工程生命周期。

第二部分重点介绍指控建模和仿真。建模和仿真是必不可少的系统工程工具。指挥过程模型和行动过程仿真能够为系统设计提供有价值的启示。这部分还介绍了战争模拟,人类能力建模,用于需求分析、系统设计和培训的仿真。

第三部分研究可供多学科指控系统工程师使用的人工智能系统技术的范围。对“传统”的人工智能(AI)学术团体提出了某些挑战,介绍了几个应用实例。评估也是这部分

的一项重要内容:怎样才能知道一个专家咨询系统在帮助解决一个问题时起的是促进作用还是促退作用?

第四部分研究一种特殊的指控系统和该系统工程化的手段。这部分探讨一些案例,强调评估的重要性,并介绍了用于下一代决策支持系统工程的某些先进技术。

第五部分的重点是需求分析、指控过程和功能的建模。该部分还涉及到一些有争议的系统,一些棘手的分配问题,以及技术怎样才能勾画出用于战术计划系统工程的方法。

第六部分描述了某些先进通信技术,以及它在支持态势评估和指挥决策方面过去是如何发挥作用的,将来又有怎样的影响,重点放在将来帮助指控通信系统工程革命化的技术上。

第七部分讨论了其他一些问题。描述了计算机辅助系统工程(CASE)工具,提出了“系统设计重点的变化”(从机器转到人),探索了“认知系统工程的前景”,考察了多学科系统工程教育和培训的性质。

希望本书所介绍的技术能在将来更大型的指控系统工程中发挥更大的作用,各章节能成为扩展您概念上的和方法上的工具箱。

参 考 文 献

- 1 Chambers, GJ (1986) "The Systems Engineering Process: A Technical Bibliography." *IEEE Transactions on Systems, Man and Cybernetics*, Volume 16, Number 5, September - October 1986, pp. 712-721
- 2 Defense Systems Management College, *Systems Engineering Management Guide*, US Government Printing Office, 1986
- 3 Eisner H, *Computer-Aided Systems Engineering*, Prentice-Hall, 1988
- 4 Mills, JA (1985) "A Pragmatic View of the Systems Architect," *Communications of the ACM*, Volume 28, Number 7, pp. 708-717
- 5 Smith, CU (1988) "Applying Synthesis Principles to Create Responsive Software Systems," *IEEE Transactions on Software Engineering*, Volume 14, Number 10, pp. 1394-1408
- 6 United States Department of Defense, *Engineering Management*, MIL-STD-499A (USAF), May 1, 1974, pp. 11-18

第 一 部 分

指控信息系统工程

指控系统中的软件重用

Christine L. Braun and Alan B. Salisbury

引言

许多文件都已证实国防部内的“软件危机”。每年花在新软件开发和现有软件维护上的费用高达几十亿美元。虽然最直观和最伤脑筋的问题主要都是与所有新软件有关的实际日常成本增加和开发周期延长的问题,但是,日益增加的大量正在使用中的软件的维护成本不断攀升,至少也具有同等的重要性。指控系统用的软件也不例外,并可能在这方面更严重。

软件“重用”即同一个软件用于多个系统将有可能使新系统软件开发成本大大降低,使维护费用适度降低。有几个有利因素支持这一结论。首先,是各种指控系统之间高度的功能通用性,从高层次的应用通用性(如战斗命令)到通用保障功能(如自动报文处理),直到更通用的功能(如人机界面)。这些领域都含有大量通用的子功能基础,为软件跨系统重用提供了充分的潜在空间。第二个因素是作为指控系统应用语言的 Ada 语言的使用日益广泛。Ada 编译程序对各种操作系统的广泛适用性大大方便了对可重用的设计和程序进行移植重用。

除了节省开发成本和时间外,软件在指控系统中的重用还有其他一些好处。例如,系统共享通用软件可以更容易实现互操作性,尤其是大量互操作软件用于满足两个系统之间数据元、数据代码和不同信息(报文)格式的转换需求。根据通用软件的共享程度,这样的转换可以部分或全部取消。类似的好处还表现在可靠性、维修性和可用性等领域。

陆军战术指控系统(ATCCS)给出了一个在指控系统中的软件重用具有巨大潜力的很好的实例研究。ATCCS 包括 5 个系统,每个系统控制 5 个主要战场功能区域中的一个(机动、火力支援、情报/电子战、防空和作战部门支持),再加上一个全面覆盖的分布式综合系统,为指挥员提供“部队级”控制。每个系统都已经单独开发出来,有几个系统在不同的处理器上使用了不同的语言,许多是用不同的方法实现同样的功能,没有什么通用性。在一位综合项目执行官领导下花了很大功夫,识别出所有五个系统中能使用重用软件完成的通用功能。这样,ATCCS 就可以指望与时俱变,使用层次越来越高的通用(重用)软件,从而明显地展示出可重用软件的优越性。

从许多方面看,软件重用并不是什么新事物。例如,通用操作系统和数据库管理系统(DBMS)的利用就是一种比较成熟的重用形式。但是,通过在软件库中用严格的方法(过程)进行功能(领域)分析、软件编目和检索是一个新的方向。它对系统开发人员和维护人员的潜在回报是非常高的。

什么是软件重用?

软件重用可以简单定义为“软件成分使用一次以上”。这是一个非常宽泛的定义。一个“软件成分”可以是软件开发过程的任何产品——一组代码、一份设计规格书、一个测试案例等。一个成分使用一次以上,可以表示从一项设计被其设计者非正式重用,直到一套大型软件包如 DBMS 的广泛使用等任何事情。该成分可以不必改动就重用,也可以进行修改以适应新的应用。那些能帮助自动生产程序的技术,如应用发生器、4GL 等,也可以被看作是重用的方法。

当今软件重用已较普遍。在低层次上,使用数学程序一类的标准软件已经非常普遍。在较高层次上的典型应用则包括商业操作系统、DBMS 系统等。通常我们并不把这些标准活动称做“软件重用”,但它们确实就是软件重用。软件重用研究者的目的是把这种活动扩展到更多的应用中去。在了解了上述已有案例中重用是如何运作之后,我们就可以找到如何使这些活动制度化的方法了。

一个重用系列

识别和重用软件成分有许多种方法。一种方法我们可以称作“自下而上与自上而下结合的方法”。

自下而上重用寻找的对象是小的广泛可用的底层组件,如数学软件包、分类和搜索程序等。有许多公用的库都能提供这些成分。有些通用的 C 程序手册也可以利用;许多作者都把源代码放在一个磁盘中。Ada 软件组件可以从政府的 SIMTEL20 库中免费得到,也可从各经销商处购得。自下而上重用的好处在于组件使用的范围广。

自上而下重用依靠对系统结构的整体观念识别出可以被重用于特定级别系统的较大规模的组件。它寻找的是那些能够用作设计框架并允许大规模重用的通用体系结构。数据库管理系统(DBMS)就是一个例子。DBMS 概念的最初设计者认识到,许多系统都包含大批可以提取和打包的数据处理工作。用一个商用软件包来提供这些服务能够节省设计者的大量工作,但它要求设计必须与 DBMS 提供的功能和接口相匹配。实际上,DBMS 的发明已经对某一类系统规定了一种通用的体系结构。自上而下重用的好处在于重用的规模大,但这种重用的典型实例很少。

当把这两种方法结合起来时,就可以最大限度地得到重用的好处。结合法是将一个通用的设计体系结构作为出发点,然后利用较低层次的成分菜单来填充体系结构内的各种专用功能。显然,只有当我们将重点集中在某一特定的应用领域时,这种方法才有意义。

特定领域的重用

软件重用在某些领域比其他领域的应用要广泛得多。编译程序开发领域被普遍认为是成功重用实践的一个范例,值得认真分析。一个通用的编译程序体系结构(前端检查程序的语法和语义,后端产生目标码、列表等)几乎是被全球所接受的。有了这样一个通用

编译程序体系结构,在编译程序制作中就会有如下的大量重用:

- 编译程序设计者再也不必从画草图开始,设计重用是标准的,所有设计都从同一个顶层结构开始构建。

- 由于所有编译程序都有相同的结构,编译程序各部分之间的接口也就趋于标准化。例如,在 Ada 前端和后端之间就有标准接口,称为戴安娜树。

- 通用体系结构为描述编译程序各部分而建立了一个公用词汇表。如果一个程序设计员说“我需要一个能在戴安娜树框图上运行的 Ada 总体优化程序”,其他编译程序设计者都能懂得其含义。

- 由于所有编译程序都建立在具有标准接口的相同功能成分基础上,而设计者在提到它们时使用的都是共同的词汇,因此大大简化了成分的重用。例如,同一语言的各种编译器通常都重用一个编译器的前端界面。

在其他应用领域,通过建立类似的通用体系结构,也有可能 在软件重用方面取得重大进步。尤其是指控领域,非常适于上述这种方法,我们将在后面进一步说明。

重用导向的设计

有了以上描述的通用体系结构和实施该体系结构各主要部分的一套组件,便有可能对设计活动进行导向,以便最大限度地重用现成的组件。类似情况对硬件领域也是非常适用的。当今电子系统的主要子功能在集成电路上或“芯片”里都是现成的。硬件设计者设计系统时不必设计到门逻辑级,也不用去查看哪些芯片能够完成其设计的一部分。相反,他们知道哪些芯片可用,于是引导自己的设计去适应这些芯片。很清楚,后一种方法导致更多的重用,从而节省了设计成本。然而,当今的软件设计全都一味追随先前的模式。我们一直设计到详细级,然后再尽量把设计与现有的软件组件相匹配。我们发现根本没有完美的匹配。在重用领域的一个目标就是推进导向方法,以便最大限度地实现重用,我们相信,这种方法对指控系统是可行的。

为什么在指控应用中要推广重用?

指挥和控制领域确实是软件重用的一个广阔天地。一方面因为此类系统的性质和结构为大量重用提供了机遇;另一方面由于重用的潜在好处确实非同一般。下面几节对某些机遇和好处进行简要讨论。

机遇

系统规模 and 成本

指控系统规模大,成本昂贵,而且这种趋势与日俱增。由于硬件变得越来越便宜和强大,会大大鼓励人们制造越来越多的软件,以便使硬件得到充分利用。不过,制造软件是劳动密集型的工作,正是由于劳动力的价格不断上涨,软件才变得越来越贵。关键是通过