136

TP311.56
X44

集成化系统开发的技术基础

萧蕴诗　严隽薇　编著

清华大学出版社　Springer

(京)新登字 158 号

内 容 简 介

本书综合了软件工程、实时系统分析方法、系统集成方法论和组合软件工程等多门技术的精华，将软件工程、实时系统分析技术、系统集成支撑工具、面向对象技术和目前成为热点并正在形成的集成框架技术等一系列系统开发方法串起来，加以精练、浓缩，形成一套能够运用于诸如大型应用系统、企业 CIMS 应用工程等一类集成化系统开发的基本方法和技术。

本书的特点是既注重系统性和科学性，又注重实用性；既有综合性，又有专门深入的论述。本书正文共 7 章。第 1 章是绪论，介绍集成化系统及其开发的任务；第 2 章较系统地介绍了软件开发中成熟的结构化开发方法；第 3 章是实时软件设计方法；第 4 章为面向对象的系统分析；第 5 章介绍系统集成方法与工具；第 6 章是对集成框架技术的介绍；第 7 章给出了一些集成化系统的实例。

本书可作为高等院校机电类专业“系统开发技术基础”等课程的教材或参考书，也可供有一定实际经验的软件工作人员以及需要进行系统分析、开发和系统集成的读者阅读参考。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

书　　名：集成化系统开发的技术基础
作　　者：萧蕴诗　严隽薇　编著
出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.tsinghua.edu.cn
版式设计：刘　路
印 刷 者：北京密云胶印厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印张**：18.75　**字数**：346 千字
版　　次：2002 年 9 月第 1 版　2002 年 9 月第 1 次印刷
书　　号：ISBN 7-302-05707-9/TP・3366
印　　数：0001～4000
定　　价：29.00 元

前言

本教材的基础是“实时软件工程”课程的讲稿。原课程是20世纪80年代后期顺应工业自动化专业对实时系统开发的要求而设置的，前后共讲授8次，并对自动化专业的研究生掌握系统开发方法起到了良好的作用。随着自动化学科的发展，特别是计算机集成制造技术（computer integrated manufacturing system，CIMS）和现代集成制造技术（contemporary integrated manufacturing system，CIMS）的推进，我们感到必须加深与拓宽课程的内容，并在近4年中做了一些改革。例如增加了“软件工程的新动向”，“面向对象技术”以及“集成化系统开发实例介绍”等内容。

该教材对原有课程目标及内容进行改革和更新，使其成为内容涉及自动化领域中的高技术，特别是大型复杂系统开发技术，讲授集成化系统开发的新兴课程教材。其教学大纲的特色主要有两点：第一是综合性强，教材综合了系统集成方法论、实时系统分析方法、面向对象方法学和组合软件工程等多项技术并取其精华，将它们有机地结合，形成集成化系统开发方法整体；第二是工程背景浓厚，教学大纲以工程领域提出的系统规模、集成性、开放性以及复杂性要求为出发点，讲授实际系统开发中已采用或正在兴起的技术，并附以实例。

本书正文共7章。第1章对集成化系统作一引论，这是对全书素材的简要综述，并提出了软件系统、单元系统、复杂系统、集成系统和集成优化系统等5个层次的系统开发方法学。第2章和第3章介绍软件工程传统的结构化开发方法和实时软件开发方法。第4章论述面向对象的分析方法，结合丰富的实例集中讲述面向对象方法学。第5章集中介绍系统集成方法与工具，例如网络技术和数据库技术中的方法、软件重用技术等。第6章则是对新兴

的面向过程与领域的集成框架技术做概要介绍。第 7 章介绍集成化系统实例。

本著作受到国家自然科学基金(课题号 69684003)和上海市研究生课程教材建设计划的支持。同时,本书的主要内容已经过几轮研究生教学实践的检验效果较好。本书的第 2,5,7 章由萧蕴诗编著,第 1,3,4,6 章由严隽薇编著,全书由萧蕴诗统编,严隽薇负责文稿的录入和编辑工作。这里特别要感谢同济大学电子与信息工程学院的研究生刘正权和汤哲川,他们两位为本书第 2 章和第 5 章的编写工作付出了大量劳动。同时本书还得到了同济大学系统工程专业研究生韩宏洲、牟川、陈恒、郭辉以及 00 级和 01 级部分研究生的支持与协助,谨在此对他们表示衷心的感谢。

作　者

2002 年 6 月

第1章

集成化系统及其开发的任务

1.1 集成化系统的形成和发展

在20世纪的后50年中，随着以计算机和通信技术为代表的信息技术的飞速发展，越来越多的复杂系统不断出现，它们对计算机的应用需求发生了巨大的变化。这类复杂系统可以统称为“集成化”系统。集成化系统对计算机的需求带动了系统开发方法学的不断演变与进化。为了了解并掌握这一类复杂系统的开发方法与工具，首先要了解集成化系统的形成与发展。

1.1.1 对软件工程产生历史的回顾

1. 软件危机导致软件工程产生

随着计算机于1945年问世，软件技术也开始在应用中得到发展。60年代中期以后，软件数量急剧膨胀，于是在计算机软件的开发和维护过程中遇到了一系列严重的问题。例如，对程序运行时发现的错误必须予以纠正；对用户的新需求必须做相应的修改；硬件或操作系统更新时，通常需要修改程序以适应新的环境等。人们发现上述这类软件维护方面的工作不仅需要占用惊人的资源消耗比例，更可怕的是许多程序的“个体化软件作坊”特性使得对它们的维护最终成为不可能，这就是“软件危机”的开始。北大西洋公约组织的计算机科学家于1968年在联邦德国召开了国际会议，并使用了“软件工程”这个专用名词。于是，软件工程这门新兴的工程学科诞生了。

软件危机反映出软件开发和维护中的两个问题：如何开发软件以满足不

此为试读，需要完整PDF请访问：www.ertongbook.com

断增长的软件需求；如何维护数量不断膨胀的已有软件系统。具体地说，软件危机有以下 7 种表现：

（1）对软件开发成本进度的估计常常很不准确；

（2）用户对“已完成的”软件系统不满意的现象经常发生；

（3）软件产品的质量往往靠不住；

（4）软件常常是不可维护的；

（5）软件通常没有适当的文档资料；

（6）软件成本在计算机系统总成本中所占比例逐年上升；

（7）软件开发的速度远远跟不上计算机应用迅速普及深入的趋势。

分析产生软件危机的原因有两方面：一方面与软件本身的特点有关；另一方面与软件开发和维护的方法不正确有关。

软件不同于硬件，它是计算机中的一种逻辑部件而不是物理部件。因此在程序编写及成功运行之前，很难估算软件开发的进程，也很难对软件开发的质量加以评价，即软件开发过程是较难管理和控制的。软件虽然是“用不坏”的，但是已运行的软件还可能会发生故障。所以软件维护要比硬件维护困难更多。

软件的另一个特点是其规模庞大，有些软件的规模甚至呈指数增长。例如美国穿梭号宇宙飞船的软件规模就是这种情况，到 70 年代末，该飞船的软件包已包含 4000 万行目标代码，相当于 4000 个人年的程序量。这样规模的程序由于其复杂程度的大大增加，已经不是单纯加大投入编程的人力就能完成的。如何保证所有人完成的工作最终构成一个高质量的大型软件系统，将是今后的主要问题，同时它也是很复杂的问题，不仅涉及技术问题，而且还有软件开发的科学管理问题。

与软件开发与维护有关的很多错误认识和不正确的方法也是使软件问题发展成软件危机的主要原因。错误认识和做法主要表现在忽视对软件需求分析的重要性，认为软件开发就是编出正常运行的程序（程序编写仅为“程序开发”的一部分），轻视软件测试和维护等。

实际上，软件有一个从定义、开发、使用和维护，到最终被废弃的一个漫长的过程，称之为软件生命周期。在问题定义时期，要对问题进行可行性研究和需求分析。需求分析就是深入具体地了解用户的要求，在所要开发的目标系统必须做什么这个问题上和用户取得完全一致的看法，然后才能进入开发时期。若忽视需求分析，仓促地开始编写程序，到了测试阶段或软件交付使用后才发现“已完成”的软件不完全符合用户的需要，那就太晚了。根据美国一些软件公司的统计资料，在后期引入一个变动比在早期引入相同变动需要付出的代价高 2～3 个数量级。图 1.1 定性地描绘了在不同时期引入变动需要付

出的代价的变化趋势。图 1.2 是美国贝尔实验室统计得出的定量结果。

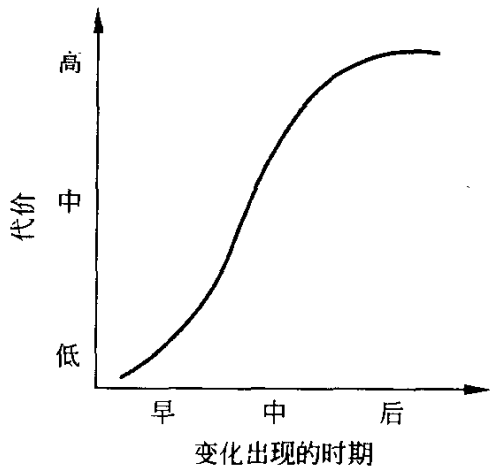


图 1.1　引入同一变动付出的代价随时间变化的趋势

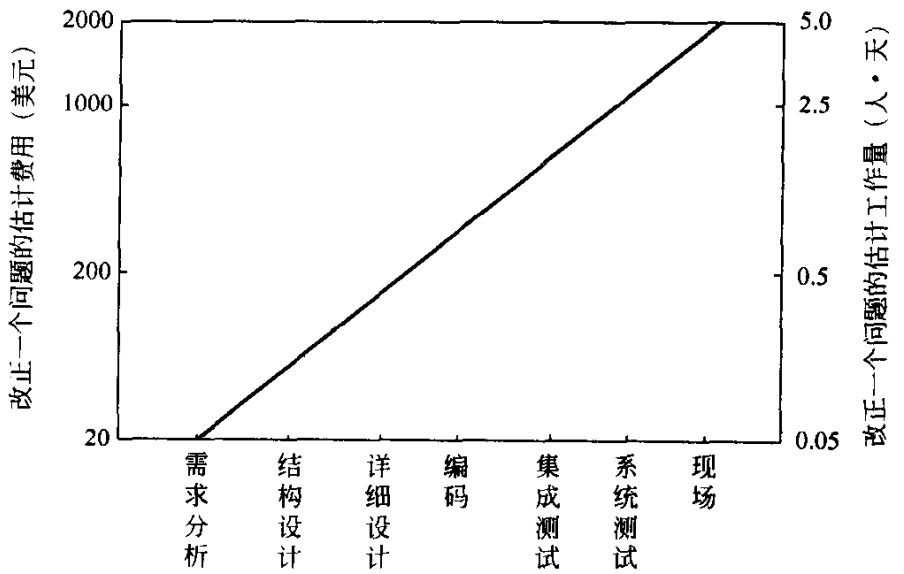


图 1.2　美国贝尔实验室统计出的定量结果

软件的开发要经过总体设计、详细设计后才能进入程序编写阶段，之后程序还必须经过大量的测试方能交付用户使用。这中间，编写程序只占总工作量的 10%～20%，而测试工作通常占总工作量的 40%～50%。可见，程序只是完整的软件产品的一个组成部分。

从图 1.1 和图 1.2 中也可以看到，轻视维护工作是一个最大的错误。统计数据表明，实际上用于软件维护的费用占软件总费用的 55%～70%。

总之，对软件应有如下正确的认识：软件是程序以及开发、使用和维护程序需要的所有文档。这也就是对软件的定义。

解决软件危机有两方面途径：一方面要有技术措施即方法与工具；另一方面要有必要的组织管理措施即管理学。软件工程正是从管理和技术两个方面研究如何更好地开发和维护计算机软件的一门新兴学科。

2. 软件工程的基本内容

软件工程的定义：软件工程是指导计算机进行软件开发和维护的工程学科，采用工程的概念、原理和方法来开发和维护软件，把经过时间考验并证明正确的管理技术和当前能够得到的最好的技术方法结合起来。著名的软件工程专家 B. W. Boehm 于 1983 年提出软件工程的 7 条基本原理，并认为这 7 条基本原理是确保软件产品质量和开发效率的原理的最小集合。这里，仅给出其中最为重要的两条原理，因为这也是软件工程的传统途径强调的两个重要方面。

(1) 生命周期方法学(瀑布型模型)

生命周期方法学遵循“用分阶段的生命周期计划严格管理”的原理，从时间角度对软件开发和维护的复杂问题进行分解，把软件生命周期划分为定义、开发、维护等若干阶段，各个阶段有相对独立的任务，然后逐步完成每个阶段的任务。这种方法使软件开发和维护这一复杂问题的困难度降低，因为每个阶段的任务相对独立，而且比较简单，便于不同人员分工合作。该方法要求软件生命周期的各个阶段都采用科学的管理技术和良好的开发技术方法，并在各个阶段之间加入严格的技术和管理两方面的审查，使整个软件开发工程有条不紊地进行。按照这种传统的生命周期方法学开发软件，各个阶段的工作自顶向下，从抽象到具体顺序进行，犹如山涧瀑布从高处流处低处，奔流不息。因此用瀑布模型(waterfall model)(见图 1.3)来形容传统的生命周期方法学。表 1.1 概括了软件生命周期各个阶段的名称、关键问题和结束标准。

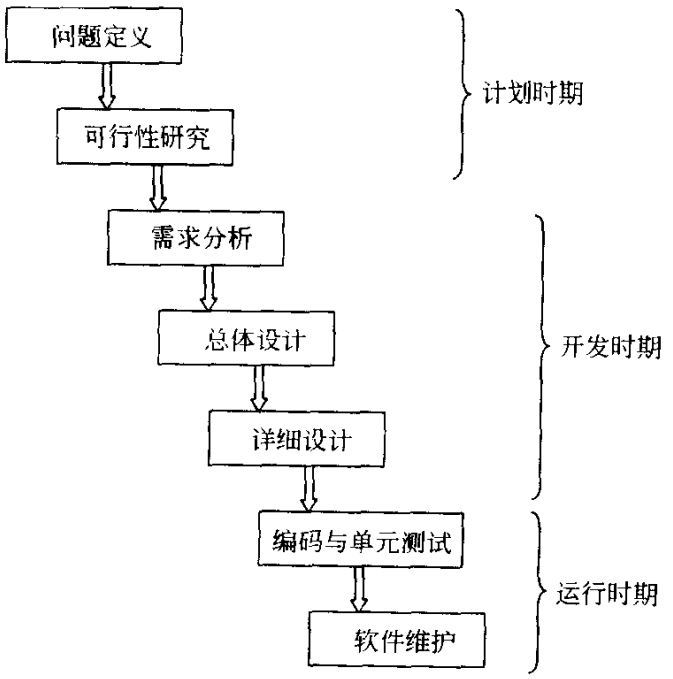
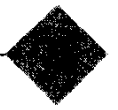


图 1.3 瀑布模型



(2) 结构化分析及结构化设计技术

传统的生命周期方法学在划分各阶段以及在各个阶段中继续分解各个子问题时，都采用结构化分析(SA)及结构化设计技术(SD)。这是一种 60 年代末提出，到 70 年代大家都已公认的先进的程序设计技术。实践表明，采用先进的技术既可以提高软件开发的效率，又可以提高软件维护的效率。本书第 2 章将介绍结构化分析及结构化设计软件开发技术。

3. 小结

“软件工程”学科为克服各种“软件危机”，提高软件的生产效率和质量水平，发挥了巨大的作用。至今软件工程提供的很多方法仍是各种系统开发的有效手段。然而，科学在发展，技术在进步，时代在前进。面对信息时代，为适应计算机应用的新情况，软件工程也一定要不断向前发展。

表 1.1　结构化分析及设计过程小结

阶　段	关 键 问 题	结 束 标 准
问题定义	问题是什么	关于规模和目标的报告书
可行性研究	有无可行解	系统的高层逻辑模型： 　　数据流图 　　成本/效益分析
需求分析	系统必须做什么	系统的逻辑模型： 　　数据流图 　　数据字典 　　算法描述
总体设计	概括地说，应该如何解决这个问题	可能的解法： 　　系统流图 　　成本/效益分析 推荐的系统结构： 　　层次图或结构图
详细设计	怎样具体地实现这个系统	编码规格说明：图或 PDL
编码和单元测试	正确的程序模块	源程序清单；单元测试方案和结果
综合测试	符合要求的软件	综合测试方案和结果；完整一致的软件配置
维护	使软件持久地满足用户的需要	完整准确的维护记录

1.1.2 计算机应用对象的演变与发展

计算机应用所涉及的领域大到宏观宇宙的探索和国家大事的管理，小到微观世界的研究和家庭个人的娱乐；从各种民用产品的生产、经营和管理到军用的飞机、军舰和大炮的驾驶和控制。所以，软件工作者所面临的是一个不断变化的世界，随时会出现新问题，需要新办法来解决。计算机软件面临的是各种各样要求不同的用户，而不单纯是软件设计和开发者。这就是说，软件工程是永远不可能一劳永逸的，必须不断地向前发展。

回顾计算机应用的历史，特别是近十几年来，随着计算机应用领域的不断扩大，计算机应用对象也在不断地演变和发展。其中最明显的是对象的工程目标、对象的特点、开发工作量规模、用户对工期的期望周期以及开发方法与工具等几方面的演变与发展。下面将从上述 5 方面叙述这种变化，从而可以认识到计算机应用对象的复杂化带来的系统开发工作的复杂化。

1. 对象系统应用目标的演变与发展

根据计算机应用工程目标的变化情况，可以列出以下 5 类不同层次的对象系统。

(1) 软件系统

软件系统的工程目标是“软件开发”。在计算机应用的初期和中期，应用对象绝大多数属于此类。经过四五十年的软件开发历程，软件开发类的应用系统成为最广泛、最成熟的系统。特别是随着“软件工程”学科的不断成熟、发展，软件开发的方法与工具日益增多，开发工程的管理也日臻完善，因此软件系统的开发效率与质量不断提高，成本开始得到控制。

软件系统在开发过程中，重视需求模型的建立和软件结构的设计。此类系统以软件开发为主，一般不涉及硬件的开发，但是会包含各类输入输出接口的软件设计任务。可以认为这是一种“单纯软件系统”。这类对象系统的开发工作量多半为中小规模。用户对开发周期并不报有很高的希望，因为它们的开发周期往往会较长。传统的自行开发软件属于此类对象系统，例如企业的工资管理程序、银行的文件管理制表程序、航空公司的飞机票预订系统和医院的患者监护系统等等。

70 年代初期，微电子技术兴起，微型机时代开始，计算机应用系统更加深入、广泛，开始出现更多的实时系统。实时系统的计算机应用目标中，不仅要

求进行实时软件的开发，而且要求建立一个由若干硬件装置和设备组成的独立的系统。这类对象系统就是单元系统。

（2）单元系统

在单元系统的工程目标中，要求构建“系统结构”，特别是在实时的系统对象中要求构建“实时系统结构”。因此单元系统的开发不仅重视需求模型的建立和软件结构的设计，还要求构建对象目标系统的体系结构模型（architecture model），即建造对象的目标物理系统。这类单元系统由于有实时性，因此，需求模型中同时要求反映包括控制流和控制要求说明的控制模型。这类应用对象的开发比第 1 类软件系统的开发难度增加了。由于难度与工作量的增加以及当时又没有太成熟的方法与工具，对它们的开发工作量多半会达到中、大规模，当时用户对开发周期不敢奢望，但不希望太长，因为它们的开发周期处于长短均有的状态。开发周期的长短取决于实际对象及其开发任务的性质与难度，同时很大程度上决定于系统的开发方法、工具以及设计与开发人员的技术水平。这类系统在 70 到 80 年代最为流行，一类由工程技术人员自行设计、开发的单元系统多属于这类对象系统，例如，商业上的自动售货机系统，工厂中的热工过程控制系统等等。这期间不少实时软件与实时系统的开发方法与工具应运而生，本书第 3 章将介绍的 H-P 方法就是一种先进的、有效的实时系统开发策略。

（3）复杂系统

复杂系统的名称已经表示出它的应用工程目标，即开发一个“复杂系统”。这类对象系统是随着计算机应用领域的扩大，逐步渗透到多种领域，特别是制造领域后逐步形成的。之所以称其为“复杂系统”，是因为它具有由多个单元系统合成的体系结构，并由多个相应的软件系统组合成系统的软件结构，因此在复杂系统的工程目标中，一定包含了单元系统的工程目标。除此之外，复杂系统应更重视结构体系的研究，这是为了更好地认识系统，例如，复杂系统结构通常可以描述为分层递阶体系结构，若对带有一定共性的分层递阶体系作深入研究，必将有利于认识系统。为了更好地建立复杂系统的模型，从而为之构造良好的结构，在对复杂系统的开发过程中开始引入面向对象的方法，本书第 4 章对此有专门论述。这类对象系统比第 2 类的单元系统的开发难度与工作量又增加了。难度主要表现在各单元之间的接口开发上，包括硬件接口和软件接口。工作量的增加一方面是由于系统内有若干个单元系统需要开发；另一方面各个单元系统之间的数据交换或通信软件已成为开发工作中的重要内容。

这类系统中多个单元的合成是一项新出现的开发难题，而当时现成的开

此为试读，需要完整PDF请访问：www.ertongbook.com

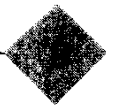
发工具还处于只适用于单元系统开发的水平，因此很多单元合成的难题还要通过攻关才会解决，所以这类系统大都由系统开发专业公司或部门开发，形成一定的商品化软件后提供给系统的开发人员。由于复杂系统的开发在购置商品化软件后仍需进行二次开发，进一步将多个单元系统合成总系统，因此系统的开发工作量多半都达到中、大规模。用户在看到了这种复杂性后，只有希望开发方能尽量缩短开发周期。实际上，复杂系统的开发周期确实不能预料，由于复杂系统开发方法的匮乏，往往又出现系统开发周期、质量和费用难以控制与估计等“危机”情况。看来，研究和创造新的系统开发方法已势在必行。70 年代以后出现的计算机辅助教学系统(CAI)、企业中的计算机辅助设计系统(CAD)、计算机辅助制造系统(CAM)、分布式控制系统(DCS)、车间柔性制造系统(FMS)以及管理信息系统(MIS)等都属于这种复杂系统。

(4) 集成系统

集成系统的应用工程目标很明确：要求“系统集成”。复杂系统尽管复杂，但它还是属于某一类应用的计算机应用系统。当复杂系统进一步拓展、发展，往往要求综合一类以上应用系统。例如，从工程设计类应用跨接到生产制造类应用领域，要求实现从产品设计、工艺设计、数控编程到加工制造的贯通，形成一个集成的新产品试生产系统。这时就提出了工程设计(CAD/CAE)、工艺设计(CAPP)、计算机辅助制造(CAM)和车间柔性制造系统等多系统集成的应用目标。由于跨越不同的应用领域，系统内部不仅有多个单元系统，而且有多类应用系统，因此集成系统的特点是在达到单元系统的工程目标的基础上，更迫切要求解决不同类型系统的集成。集成的任务包括不同类型、不同规模、不同厂商提供的软件产品的数据交换和通信等集成，还包括这些软件各自相应的应用系统管理功能、协作集成要求，最终人员与机构也必然参与集成。

这类集成系统应用对象的出现，是由于 70 年代中，特别是 80 年代后信息技术对制造业、金融业和商业等领域的巨大促进作用，复杂大系统内的多种应用系统也因此在先进的计算机网络和数据库技术支撑下，有可能将物流、信息流和价值流集成为一体，从而能更快、更好、更有效地运作。集成系统是在支撑系统的支持下多类应用系统的合成，同时具有多种人机交互环境。90 年代初兴起的客户机/服务器的数据应用模式为人与机的交互、人与人的协作以及数据共享等功能起了巨大作用。

集成系统的开发除了包括内部多个复杂系统的开发外，还要化相当多的力气研究系统的总体集成问题，即要求系统的总体设计、详细设计考虑更多的



问题；不仅如此，开发人员还必须加快开发进度，更好地合理配置各类资源，建立信息共享的机制，保证系统正常集成运行。集成系统的开发不仅是大规模的，而且开发项目的类型全面、难度大。用户往往会担心开发周期过长，而从系统开发投资效益的角度又不允许过长。

集成系统的开发不仅需要适用的开发方法和工具，还需要用系统工程的方法来指导、管理和控制整个工程开发的过程。系统科学及方法学将成为集成系统开发的重要理论基础。这类集成系统的典型例子是企业的计算机集成制造系统，即 CIMS 工程。CIMS 工程中通常包含工程设计自动化、生产经营管理自动化、制造自动化、质量管理自动化等 4 个功能分系统和计算机网络、数据库两个支撑分系统。每一个分系统中都包含多个相对独立的单元系统。

(5) 集成优化系统

集成优化系统的工程目标是在建立集成系统的过程中力求“集成系统的优化”。“优化”一直是控制科学、系统科学追逐的最高目标，并且已有很多优化理论和智能优化方法的研究成果。集成系统优化的重点放在集成系统总体上对各项重要指标的优化。以制造业中的集成系统为例，这类集成优化系统就是制造行业中各类企业的综合自动化大系统。它们的优化目标就是企业追求的优化目标。这些目标可以归纳为“T，Q，C，S”4 个字，其具体含义如下：

T：最短的产品上市时间(time to market)；

Q：最好的质量(quality)；

C：最低的成本(cost)；

S：最好的服务(service)。

这 4 个字是企业为走向市场，赢得竞争而面临的 4 个永恒的主题，是达到企业最优总体经济效益和社会效益的 4 大法宝。可见集成优化系统必须以企业的产品为龙头，建立产品全生命周期的各类系统集成的基础结构。为了使企业内部的物流、信息流、价值流全面集成并优化，必须全面准确地为企业建立模型和优化的指标体系，通过工程再造(re-engineering)等手段，获得优化的系统模型。

为了提高集成优化系统中系统的开发速度与质量，系统将采用软件重用技术，遵循软件的各类标准，甚至采用最先进的预集成软件系统工具。集成优化系统是比一般集成系统规模更大、更全、开发难度更高的对象系统。目前还没有什么能从根本上缩短开发周期的办法，开发者努力去做的是采用正确的系统方法论与系统工程方法，寻求各种成熟的技术先进、管理科学的集成优化

使能软件。有了科学的理论与工程方法论作指导，又能借助先进的集成优化使能工具，就可能使系统优化，同时缩短开发周期。对这类集成优化系统开发的例子还尚少，有下面一种形式可算是一例：在某企业的 CIMS 应用工程中选用整套可集成的应用软件包和集成工具集，并得到厂商和技术依托单位对系统实施的支持以及技术培训，同时对工程开发过程实施监理制度，那么该 CIMS 应用工程就可能成为一个集成优化系统。

所谓集成优化使能软件，是指整套可集成的应用软件包和集成工具集，或称之为预集成软件的系统工具，也就是一种集成框架系统（第 6 章将作介绍）。使能工具（也可称之为“中件”）的研制与开发方兴未艾，例如，企业制造资源计划 MRP-Ⅱ、生产数据管理系统 PDM 以及用于开发计算机协同工作环境的群件 Lotus Notes 系统等。

集成优化系统涉及大量的技术问题，还有很多管理问题。目前已有很多可供企业优化的新的管理理念和模式在世界上流行，下面列举若干：

精简化生产（lean production，LP）；

准时制生产（just in time，JIT）；

并行工程（concurrent engineering，CE）；

供应链管理（supply chain management，SCM）；

敏捷制造（agile manufacture，AM）；

虚拟制造（virtual manufacture，VM）。

这些新理念、新模式正在被引入企业的 CIMS 工程，它们将促使企业的全面优化。

尽管集成优化系统复杂，又涉及管理方面的问题，但作为系统开发，都可以归结为计算机应用系统的开发，因此，集成优化系统是在前述 4 类对象系统的基础上综合应用各个层次的方法与技术的，当然在如何全面集成以及如何综合优化这两个方面会有相当难度。

我们已经将计算机应用的对象按 5 个层次分别叙述了它们在工程目标、对象的特点、开发工作量规模、用户对工期的期望周期等方面的演变与发展过程。表 1.2 简明地列出了这 5 类计算机应用对象系统的演变过程。

2. 5 类对象系统的开发方法及工具

应用对象系统的演变必然导致它们所采用的开发方法与工具的不断变化和发展。表 1.3 给出了这种演变过程。

表 1.2 计算机应用对象系统的演变过程

应用对象名	工 程 目 标	系 统 特 点	工作量规模	对开发周期的期望	实 例
软件系统	(实时)软件开发	仅软件系统需求模型＋软件结构(SE)	中、小	允许较长	传统的自行开发软件属此例 工资管理程序、银行文件管理制表程序、机票预订、患者监护系统
单元系统	(实时)系统结构	软件＋硬件 RM(需求模型)＋AM(结构模型)	中、大	不希望太长	自行设计、开发的单元系统(例如一个自动售货机系统或热工过程控制系统)
复杂系统	复杂系统	多个单元合成的系统结构 多个软件系统组合分成层递阶的体系结构(O-O)	中、大	希望能缩短周期	某领域的某项系统 例 CAD,CAI,DCS 或 FMS,MIS 等
集成系统	系统集成	多类应用系统合成加上支撑系统 人机交互系统(C/S)	大、全	不允许太长	某企业的 CIMS 工程 例:经营生产管理(MIS,MRP-Ⅱ,DSS,DA)＋工程设计自动化(CAD/CAPP/CAM)＋制造自动化＋…＋支撑系统
集成优化系统	集成系统优化(T,Q,C,S)	产品全生命周期的各类系统集成的基础结构 系统建模方法 软件重用 软件标准化 (预集成软件系统(集成框架))	更大更全、复杂	必须缩短开发周期	厂商的全面解决方案和第三方的监理与技术依托,整套可集成的应用软件包＋集成工具集＋实施支持＋培训 例: MRP-Ⅱ,PDM,OANotes 又例: LP,JIT,CE,SC,AM,VM

表 1.3 系统开发方法和工具的演变过程

对　　象	开发方法和工具
软件系统	“软件工程方法”SA,SD 等 (实时)操作系统或采用数据库技术 一般 CASE 工具和语言工具(第 3 代)
单元系统	SA,SD,RP 法,RP 法的 CASE 工具 (实时)操作系统＋数据库系统 IDEF 方法、Hatley-Pirbhai 方法
复杂系统	第 2 代 CASE、CASE＋O-O 方法、体系结构理论 网络技术＋数据库系统(单体型→分散型)
集成系统	新型的 CASE,集成支持工具(技术、管理) 网络互联＋数据库系统互联(分散型或协同型,C/S)
集成优化系统	商品化开发平台 集成支持环境,集成框架技术 C/S 结构→分布式数据库结构 关系型→O-O 型

3. 小结

上述 5 类计算机对象系统处于不同的层次,从低到高越来越复杂,而每提高一层都继承前面各层对象系统所采用的方法与技术,关于这一点将在 1.2 节中详细叙述。另一种更简洁的归类方法是将 5 类对象系统分为以下 3 类对象系统:

(实时)软件开发 → 软件系统

(实时)单元系统(系统结构)、复杂系统 → 计算机应用系统

集成系统、集成优化系统 → 集成化系统

无论是 5 类还是 3 类都说明同样的问题,那就是通过了解对象系统的演变与发展过程可以进一步认识到计算机应用系统的开发是如何从“软件开发”向“集成化系统开发”演变的。结论:

① 软件面临有各种各样要求的不同用户，而不单纯是软件设计和开发者。

② 软件工作者面临的是一个不断变化和发展着的世界，随时会出现需要用新方法解决的新问题，即软件工程和系统开发方法要不断地进展。

③ 计算机应用对象的复杂化带来系统开发方法的不断变革和创新：

- 突出表现：系统规模的变化带来的系统开发任务的变化。
- 最大的变化：系统要求集成。
- 集成的程度：从小到大，逐步深入。

总之，系统**特点**的不断变化，形成了不同**规模**的系统，产生了不同开发**周期**采用不同系统开发**方法**及**工具**的**目标**不同的开发任务。

本书将就系统开发方面的技术基础问题展开讨论，其中主要是关于系统开发的方法和工具，这些方法和工具适用于前述 5 类之一的对象系统的开发。为进一步简化，统称这 5 类对象系统为集成化系统，并将“集成化系统开发的技术基础”定为本书书名。

1.2　集成化系统开发的任务

1.2.1　软件系统开发的任务

软件系统的任务是：采用软件工程的方法，完成从问题定义、可行性论证、需求分析、总体设计直至软件测试和试运行等阶段，实现软件系统预定的目标。软件工程中传统的方法适合第 1 层次软件系统的开发。这些方法如下：

- 瀑布模型法：SA，SD。
- 快速原型法：RP(rapid protocol)。

软件工程中提供的分析、设计、测试工具可以用于开发这类软件系统，除此之外还用到下列工具：

- CASE 工具(computer aided software engineering)。
- 第 3 代语言工具。
- (实时)操作系统。
- 数据库系统。

此为试读，需要完整PDF请访问：www.ertongbook.com