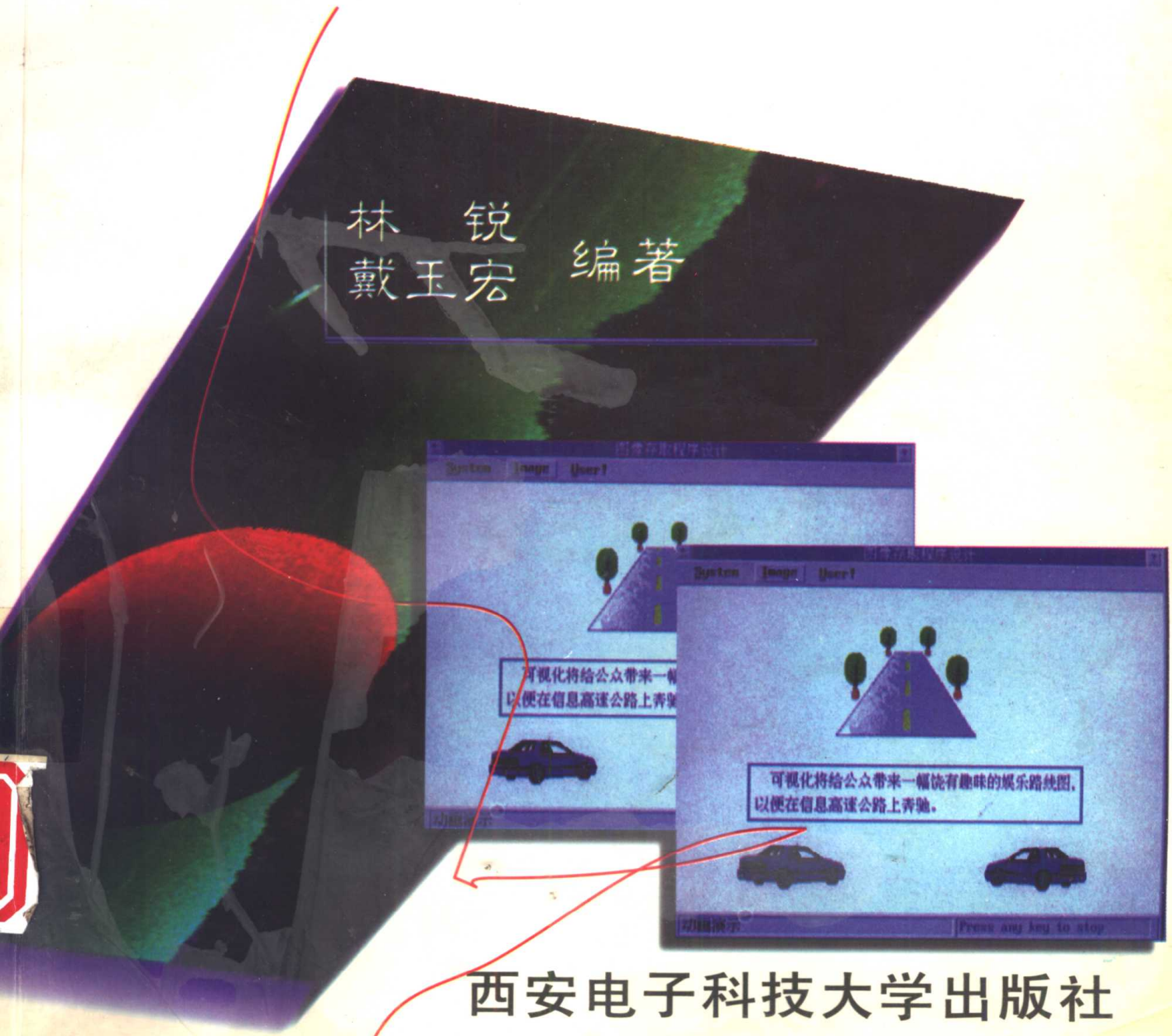
图形用户界面设计与技术

—— 以Borland C++为工具

林　锐
戴玉宏　编著



西安电子科技大学出版社

图形用户界面设计与技术

——以 Borland C++ +为工具

林锐　戴玉宏　编著

西安电子科技大学出版社

1997

（陕）新登字 010 号

内 容 简 介

本书论述图形用户界面(GUI)的设计方法学与技术，旨在帮助读者开发独立自主的DOS 图形用户界面系统。

本书第 1 章主要论述人机界面设计美学与软件工程方法学。第 2 章至第 11 章分别细述界面视感设计、鼠标程序设计、扩展内存与扩展数组设计、窗口系统基础设计、汉字显示程序设计、图像存取程序设计、对话与控制构件设计、窗口系统主控界面设计、对话集成构件设计与交互式绘图程序设计。

随书赠送可视化软件开发工具 VA 3.0，可帮助读者更好地掌握图形用户界面的设计方法、技术与技巧。该软件曾获全国大学生“挑战杯”学术科技作品竞赛二等奖。

本书的读者对象为程序员、学生及软件爱好者。

图形用户界面设计与技术
——以 Borland C++ +为工具

林锐　戴玉宏　编著

责任编辑　徐德源

西安电子科技大学出版社出版发行

西安市秦群印刷厂印刷

各地新华书店经销

开本 787×1092　1/16　印张 13　字数 304 千字

1997 年 5 月第 1 版　1997 年 5 月第 1 次印刷　印数 1－4 000

ISBN 7－5606－0513－3/TP · 0246　　定价: 35.00 元（含盘）

前　言

本书论述图形用户界面(GUI)的设计方法学与技术，旨在帮助读者开发独立自主的 DOS 图形用户界面系统。

图形用户界面作为交互式可视化环境，已成为现代软件系统的核心组成部分。当今流行的微机操作系统都具备了优秀的图形用户界面，诸如 Microsoft 公司的 Windows，IBM 公司的 OS/2，Apple 公司的 Macintosh 等。

DOS 作为微机经典的操作系统，因其单任务、内存限制等先天不足而停滞发展。但 DOS 不因此而淘汰，它经历了数十年的发展，拥有太多的软件财富和用户，这是其他操作系统无法比拟的。况且，DOS 的某些缺陷，从不同需求角度而言却可能是优点。例如 DOS 的简洁性、开放性一直吸引着大批的软件开发者。

Windows 无疑是微机操作系统的主流，所有 Windows 的应用软件都具有良好、一致的图形用户界面。目前高版本的 Borland C++与 Visual C++都提供了高效的图形用户界面开发工具，但遗憾的是它们只运行于 Windows 环境。自然，DOS 用户都希望拥有能运行于 DOS 环境的优秀图形用户界面开发工具。

本书所论述的图形用户界面设计方法学与技术是以随书软件，可视化软件开发工具 VA 3.0 (Visualization Aids for Windows & DOS)为背景的。VA3.0 提供了类似 Windows 95 风格的 DOS 图形用户界面开发工具(VA3.0/DOSGUI)，其核心内容与功能特征如下：

(1) 支持高分辨 256 色图形模式，内置画家配色功能，可用于软件界面的视感设计，真实感场景生成。

(2) 支持高分辨率 256 色模式的鼠标操作。

(3) 特别为科学计算提供了基于扩展内存的扩展数组，可动态定义任意大小的字符型、整型、无符号整型、长整型、浮点型、双精度浮点型等类型的数组。

(4) 窗口系统基于扩展内存，不受 DOS 内存限制。

(5) 支持点阵汉字输出、输入，界面系统兼容中西文。

(6) 支持 BMP、PCX 格式的图像存取。

(7) 提供众多的 Windows 风格的控制构件，可快速设计对话实体。

(8) 提供 WIMP 风格的主控界面，视感与 Windows 95 一致；特别提供浮动式菜单，支持鼠标右键定位。

(9) 提供众多通用的对话集成构件，更方便于用户开发。

(10) 内置交互式绘图功能，可实现类 Windows 画笔软件的功能。

本书第 1 章论述图形用户界面设计方法学及 VA 3.0/DOSGUI 的系统结构。第 2 章至第 11 章逐一细述 VA 3.0/DOSGUI 的技术。第 12 章将结合图形用户界面介绍与可视化应用软件的开发。附录 A 提供了 VA 3.0/DOSGUI 的功能接口，附录 B 为 VA 3.0 的软件说明书，全书彩图均摄取于 VA 3.0 标准版软件。

VA 3.0 是目前深受国内科研人员喜欢的微机可视化软件开发工具。图形用户界面开发工具是 VA 3.0 的核心套件之一(用 Borland C++开发)，其功能、美感及开发效率可媲美于任何一个优秀的 DOS 界面开发工具。结合随书软件，读者可以更好的地掌握图形用户界面

的设计方法、技术与技巧。如果不想自行开发专用的图形用户界面系统，读者可以直接使用 VA 3.0 来开发可视化应用软件。

对微机可视化软件工程的研究得到了浙江大学 CAD & CG 国家重点实验室石教英教授、蔡文立老师的帮助。西安电子科技大学微电子所可视化软件小组的戴玉宏、马佩军、马晓宇同学参加了 VA 3.0 核心系统的开发，马晓华、马晓慧等同学参加了本书的编写工作。

本书的出版得到了西安电子科技大学出版社的支持，感谢徐德源老师的推荐与认真审阅。

林　锐

浙江大学 CAD & CG 国家重点实验室

1997 春

目　　录

第1章 图形用户界面设计方法学

图形用户界面设计方法学可分为美学与软件工程方法学两类主题，前者着重于意识，后者着重于软件实现。

本书强调人机界面设计美学指导的重要性，VA 3.0 图形用户界面的美感是该系统的特色之一。界面美学设计并非是完全跟着感觉走的东西，只有理解了内涵，才可以客观地把握美学设计。界面设计的美学指导绝不是僵化教条，它亦需要灵感，追求创新。

任何软件系统的开发都要有软件工程方法学的指导，而高效的软件工程方法学应依据应用域的软件特征。VA 3.0 的软件工程方法的核心是软件重用，它采用软件系统构件化与集成构造策略，可大幅度提高应用软件的生产率与质量(详见参考文献 1)。

本章将用较多的篇幅论述人机界面设计美学与软件工程方法学；然后将简要介绍VA 3.0 图形用户界面系统的结构(第 2 章至第 11 章将逐一细述该系统的各个模块)；最后是程序设计规范。VA 3.0 的系统核心程序、应用程序及导学程序的设计均遵循此规范。

1.1 人机界面设计美学

界面美的概念很抽象，以致让人无法说清什么是界面的美。但它同时又很现实，以致人人都可以去欣赏和感受界面美，挑剔美中不足。美的界面以计算机科学为技术基础，同时又巧妙地融汇了大众心理学与美学，因此界面设计是一门复杂的艺术。目前国内有关人机界面的研究主要集中在相关软硬件的开发与具体实现上，尚缺乏界面设计理论的发展。

界面美是评价界面的关键因素之一，亦是用户与开发者共同的需求。但是界面设计的美学却并未得到应有的重视，美的设计屈从于其它因素。由于美学并非是量化的科学，轻视美学概念和美学指导，必将导致在其设计中依赖程序员个人的经验与感觉。由于程序员接受的教育主要是如何使计算机完成工作，而不是人如何工作，因此形成的界面往往不能得到广大用户的认可。

人机界面设计的严肃性(科学性)与艺术性(创造美)并不对立，美学成为一种指导思想是十分自然的和必需的，越来越多的用户将通过具有吸引力而令人愉快的人机界面与计算机打交道。界面的美充分体现了人机交互作用中人的特性与意图，将在人机界面发展中起着重要的导向作用。

1.1.1 界面设计中美的需求与导向作用

人们对美的向往和追求是与生俱有的。显然没人愿意丑化他的程序，也没有用户喜欢低劣的界面。软件开发者要设计美，用户要享受美，所以人机界面的美是用户与设计者的共同需求。

美能消除用户由感觉引起的乏味、紧张和疲劳(情绪低落)，大大提高用户的工作效率，从而进一步为发挥用户技能和为用户完成任务作出贡献。一个人机交互系统的成功与否，

往往取决于用户对界面的感官印象和使用界面的体验。一个丑陋的界面能破坏一个本来性能出色的系统，所以人机界面美的设计对该计算机系统是否被用户接受将起重要作用。

从人机界面发展历史与趋势上可以看出人们对界面美的需求，以及美在界面设计中的导向作用。界面设计已经经历了两个界限分明的时代：第一代是以文本为基础的简单交互，如常见的命令行、字符菜单，它们都具有人机交互的功能。但第一代界面并不尽人意，它考虑人的因素太少，用户兴趣不高，这种界面美的层次是较低的。第二代是直接操纵界面，它大量使用图形、语音及其它交互媒介，充分地迎合了人对美的需求。直接操纵界面通过视听、触摸等技术刺激人的感觉器官，使人可以凭借生活常识、经验和空间推理来操纵界面，从而让人在与计算机交互过程中享受了美。更高层次的人机界面甚至模拟了人的生活环境，人们可以直接从计算机系统中感受到美。例如可视化环境中的三维空间，多媒体交互技术中眼的动作、手势、语音等。这与人们日常生活中通过视听得到的美感，如欣赏画、音乐并无多少差别。

人机界面的发展历史体现了美设计的重要性，界面设计要继续经历直接操作取代命令串、三维交互取代二维交互、多通道取代单一视觉通道的变化，这些变化无疑受到美的导向作用。随着人们对界面美需求的日益增加，任何宿主机系统的开发者再也不能无视人机界面设计中的美学。

1.1.2 界面美的内涵

界面美的内涵在于其有效表达思想的能力，这也是衡量美的一种标准，它的主要构成因素是合适性、吸引力和风格。

1. 界面的合适性

界面的合适性是指软件界面是否与软件任务相融洽。评价界面的美可以考察其有效表达思想的能力，如果界面并不适合任务，则其表达思想的能力就无从谈起，那么界面美的内涵就会丧失殆尽。所以界面的合适性是界面美的首要因素，它提醒设计者不要片面追求外观漂亮而导致失真或华而不实。界面合适性既提倡外美内秀，又强调恰如其分。

合适性差的界面无疑会混淆软件的意图，致使用户产生误解。尽管它不损害软件的功能与性能，但却让用户产生了不希望的情绪波动。例如软件设计者总喜欢为其作品加一段演示(表演软件特色等)，以便吸引更多的用户。这本是无可非议的，问题在于这演示是否合情合理。如果运行一个程序，它首先表演一套复杂的动画，在后台演奏雄壮的进行曲，电闪雷鸣之后出来的却是一个普通的文本编辑器。整个过程让用户置身于云里雾里，这样的软件广告效果是令人啼笑皆非的。也许初期使用会有些幽默感，但用户很快就会厌烦其浮夸，这也反过来说明合适性极差的界面只会给软件带来厄运。

2. 界面的吸引力

界面的精心制作不仅让用户能用，而且还要让他们喜欢用，因此美的界面既要具有合适性又要具有吸引力。吸引力对软件的推销是至关重要的，显然没人甘愿购买自己不喜欢的东西。

界面是否具有吸引力决定了用户的第一印象好坏，而第一印象往往又是用户的唯一印象。怎样才能增加软件的吸引力，给用户好的第一印象呢？设计者应该预先分析用户在使用该软件时将会产生的感觉与体验，也就是说在界面具备合适性的同时要尽可能地迎合用

户对美感的需求。通常地，视感是用户对软件界面产生美感的主要来源。所以界面的美感常在于其画面清晰、布局自然、颜色和谐、线条得体、动感丰富等可视因素。

在构造界面吸引力时要特别注意界面美设计总体最优原则，即界面整体的美要优于个体因素的美。设计者有时为了增强吸引力，会特别地突出界面中某个美感因素或全部的美感因素，这常常会违反界面美总体最优原则而效果适得其反。例如图形用户界面中的窗口系统，带立体感的窗口要比平面的窗口更具吸引力。但是如果过分地增强窗口的立体感，反而会使人感到心情沉重。这时太突出的立体感(个体美感因素)破坏了界面整体美感。倘若能统筹地处理窗口系统中的各窗口对象，使立体窗口与平面窗口各就其位，密切配合，那么该窗口系统会既具有三维美感又具有操作轻松活泼的气氛。可见只有在界面美总体最优的指导下，才能构造出令人满意的吸引力来。

界面的吸引力与界面是否简单或复杂并无明显关系。因为复杂会使用户感到理解费力，用户总是喜欢轻松自在地操纵界面。

界面的吸引力是多方面的，并不存在绝对的吸引力，也许体现吸引力的某些因素会相互抵触，这时设计者应考虑主次关系来妥善解决。

3. 界面的风格

图形用户界面设计与绘画有许多相似之处，例如在制作界面的立体感时，就采用了光亮和阴影、抛光和着色等绘画技术。但是界面设计风格与专业画家的风格有很大的区别。画家注重的是与众不同，因为对于绘画而言，独具特色的风格比普通风格更具吸引力。观众对某种风格的偏爱会自然地忽视图画中的某些缺陷。

界面设计更注重一致性，即与别的同类应用软件一致。设计者必须密切注意在相同应用域中以及在最成功的软件中所使用的技术，并且必须尊重用户使用这些软件的习惯。这就像软件习惯于设置F1键为帮助热键那样。如果某个设计者却别出心裁地让F1成为程序终止热键，那么在用户渴望得到帮助而伸手击F1键的一刹那，他的工作就此结束。可以想象这种不一致性将产生怎样一个糟糕的后果。

界面设计要发展，势必要提倡采用新思想和新技术。有独到之处的界面自然要比泯然于众的界面更具有吸引力和生命力。这正如界面由传统的字符命令发展到今天的多媒体，其发展前景依然激动人心。可见界面设计中的灵活性(与一致性对比而言)亦是重要的原则，灵活性是界面脱颖而出的主要因素。

界面设计风格中的“一致性”和“灵活性”原则具有双重和明显的矛盾，这要通过一些精心设计来解决。界面的风格一部分可以通过模仿而获得。如分析同领域中一些优秀软件,学习它们的技巧，提炼它们的思想。但在模仿的同时不能限制自己的创造力，即应该在取得一致性的前提下再创造出独具特色的风格。

1.1.3 界面美与可视化——一种界面设计潮流的美学剖析

界面设计的根本目的是帮助人和计算机交流信息(即人机交互作用)。可视化是一种面向信息的图形设计，界面设计走向可视化是人类对自身视觉潜能的一种开发与应用。当今流行的界面杰出代表如Microsoft公司的Windows, Apple公司的Machintosh, IBM公司的OS/2, 工作站上的Motif, 它们均是可视化的图形用户界面。这些界面的畅销盛行充分地体现了界面美与可视化的密切关系。

Xerox Parc系统科学试验室对人机交互作用的研究，奠定了目前图形用户界面的基础，形成了使用高密度的图形显示器，实现多窗口多视图的WIMP风格界面，即以窗口(Windows)、图标(Icons)、菜单(Menus)和指示装置(Pointting Devices)为特征的界面。

WIMP界面中的图标设计可以很好地解释可视化技术在具体实现美时存在的问题。如果图标的图像很逼真的话，那么图标操作就是一种很有效的人机交互方式。因为不管是有经验的还是无经验的用户，都可以运用直觉来理解和使用图标。(这种技术已被Apple和Microsoft等公司广泛采用，并有形成一种超越语言障碍的国际对话语言的趋势。)

图标以逼真产生涵义，对于有形有架的物体很好用。但当表现较抽象的概念时，它们表达思想的能力就很差。即使图标很复杂而且形象逼真，它仍有可能模棱两可或者没有直接意义。图形可视化实质上受到用户对图像理解的限制。图标会产生多义性问题，被不同的人解释为不同的含义，所以为了防止多义性，大多数图标系统中常加一些文字说明。

1.1.4 界面设计广义美

尽管界面的美并没有增加软件的功能与性能，但却又是必不可少的。用户使用界面时，除了直接的感官美感(如视、听、触觉)外还有很大一部分美感是间接的，它们存在于人们的使用体验中，诸如界面的友好、方便、实用等。与图形界面相比，命令行是最原始的界面，它难记又难看懂。可是命令行仍显得生命力顽强，它并没有因缺乏直接美感而被淘汰。实际上，对于熟练用户而言，他们乐于使用命令行以获得高效率。命令行因具有执行高效率而赢得了专业人员的喜爱，UNIX 就是一个彻头彻尾的命令系统。

可以说，一切有利于人机交互的界面设计因素都具有界面广义美，它们是普适的而非专有的。

界面设计的一些特殊考虑也体现了广义美，如使程序能为能力较差的人使用。开发一个计算机软件系统，没有理由不让程序残障人使用。IBM公司和Apple公司都已表示关心能力较差的用户。IBM公司在1985年已经创建了残障人国家支持中心，Apple公司的专门教育办公室则提供了一些有利于程序残障人使用的计算机信息产品。

界面广义美设计的目的是要发展一种与人的特性、需求真正相适应的交互技术。在交互式软件系统中，用户界面要优先予以考虑。由于涉及了大量人的不确定因素，界面的设计需求不能一次性得到严格定义，这使得界面设计具有无结构性。传统的结构化分析和设计方法总是首先考虑系统功能分解和数据关联，它并不适合于界面设计反复迭代的特征。界面原型开发是解决界面设计无结构性的有效策略。它提倡快速构造一个界面雏形(即原型)，再从该雏形出发不断进行修改和扩充，而后演变成最终界面。采用面向对象程序设计技术来实现界面原型是受到普遍推荐的。

1.2 人机界面设计中的软件工程方法学

1.2.1 软件工程的核心问题与软件重用

在计算机系统中，软件的发展是远远落后于硬件的。现代软件工程的目标是实现软件生产的工业化，如何提高软件的生产率与质量是软件工程的核心问题。软件的开发者(或生

产厂商)追求高生产率以获得高效益(包括经济效益与社会效益)，用户则更关心软件的质量。从供需角度而言，生产率与质量存在着相互依存的关系。从软件生产过程而言，生产率与质量则存在内在的一致性：高生产率必须以高质量为前提，高质量将显著降低软件维护代价而必然导致高生产率。因此，寻求一种高效的软件工程方法学，旨在同时提高软件的生产率与质量。

软件重用是一种提高软件生产率与质量的有效途径。

从直观上理解，软件重用是显著提高软件生产率的关键。道理是浅显的，简而言之，通过软件重用，构造新的软件系统可以不必从零做起，直接使用已有的软件成分，加以合理修改，即可组装成新系统。重用合理化简化了软件开发过程，减少了总的开发工作量与维护代价，降低了软件的成本，同时也提高了生产率。从本质上理解，重用可获得很好的软件内部质量，这种内在因素提高了软件的外部质量，从而获得软件的高质量与高生产率。软件系统构件化与集成构造则是具体实现软件重用的策略。

软件重用不是一个新概念，在未被大力提倡之前，它更多地存在于人们的潜意识中而被不自觉地使用。许多经验丰富的程序员都有他自己的程序库。据调查，某些高效率软件开发人员的生产率达到平均数的 5 倍，这主要归功于大量地使用可重用的软件成分。目前许多软件公司与工厂的软件重用活动已经取得一定的经验，获得了明显的效益。

软件重用的意义比简单的代码重用要广泛得多，重用形式可以是设计重用、形式化规格说明重用，甚至是人员重用等。显然，较高级的软件重用提供了更大的优势。

1.2.2 软件质量因素

软件质量不能用单纯的定义来表述，应该把它看成“由各方面来确定”的这样一个质量观念。其中每一方面都要用一系列因素来描述。

人们在评价一个软件产品的质量时，有些因素如易用性、可靠性等是要由用户来检测的，这类质量因素称之为外部质量因素或可见质量因素。这里“用户”是一种泛称，它不仅包括直接使用该软件的最终用户，而且还包括购买、投资或改进软件的客户。因此，能反映用户需求(如正确性、容错性等)或比较容易适应需求变化(如可扩充性、兼容性等)的质量因素都属外部质量因素。

软件产品中那些只有程序开发人员才能理解的质量因素，如模块化、程序设计风范等都属于内部质量因素。

应该注意的是，软件的外部质量才是人们关心的最终目标。人们在使用软件时，很少关心该软件的程序实现，然而内部因素是保证外部因素能否满意的关键。因此，软件开发人员只有运用先进的方法学与技术获得很好的内部质量，才能确保软件的外部质量(同时提高了软件生产率)。

下面介绍一些常见的外部质量因素，为具体的软件实现设立目标。

1. 正确性

正确性是指软件准确执行需求说明中所规定任务的能力。很清楚，正确性是首要的质量因素。如果一个软件不能准确执行指定给它的任务，那么，其它的一切都是不能说明问题的。只要采用完全形式化的方法表达系统要求，那么满足这些要求就是正确的。

2. 容错性

容错性是指在异常条件下软件仍能运行的能力，又称鲁棒性或健壮性。容错性是讨论在不可预料的环境下的软件行为，本质上比正确性难以把握。容错性的作用在于如果发生异常情况，它应有能力使系统正常终止，或拒绝执行而返回正常态，或进入“降级运行”模式，确保系统不发生灾难性事件。

3. 可扩充性

可扩充性是指软件适应需求变化的难易程度，它是与软件规模有关的问题。对于小型软件而言，修改程序是容易做到的。但对于大型软件而言，能按需求变化修改并且不留下隐患是非常困难的事。通过提高内部质量来改善可扩充性，应遵循一些基本原理，如:

(1) 设计简单。在具备完成相同任务的能力的情况下，简单的系统结构总比复杂的系统结构更易于适应修改。

(2) 模块自主、凝聚。对某一模块的简单修改，不致于引起整个系统的连锁反应。

4. 可重用性

可重用性是指软件成分可以全部地或部分地应用到新系统中的能力。软件重用可以提高生产率与质量，可重用性同时又是软件的一个外部质量因素，它影响着软件质量的其它方面。

5. 兼容性

兼容性是指软件产品与其它软件组合成一个整体或相互协作的难易程度。由于软件产品不是在真空中开发的，它们需要与其它软件相互作用而更好地发挥作用。但是相关软件不兼容的情况是屡见不鲜的，不少操作系统所支持的文件格式就常常不能兼容。

兼容性的关键在于设计同构。对软件所操作的所有重要实体定义标准化的存取协议，是更通用的解决兼容性问题的方法。这是面向对象技术中数据与过程抽象的内在思想。

6. 其它质量因素

上面论述的正确性、容错性、可扩充性、可重用性与兼容性是软件质量的关键因素，它们反映现代软件工程实践中最重要的问题。构件化策略改善软件质量关键因素最为有效。从不同侧重点而言，软件质量的其它一些因素也是很重要的。

(1) **可靠性。**是指软件系统在特定的环境(条件)下，在给定的时间内，不发生故障工作的概率。目前对软件可靠性的研究一般采用数理统计法，还不能提供可以直接提高相应软件内部质量的技术路线。为了易于工程实现，最好把可靠性理解为正确性与容错性的综合。构件化策略有助于获得很高的“内建可靠性”。

(2) **效率。**是指对资源利用的合理程度，如软件的时空效率(不同于开发效率)。

(3) **可移植性。**是指把软件产品转移到其它硬件与软件环境的难易程度。

(4) **安全性。**是指软件系统的各个成分(程序、数据和文档等)，免受非授权人员存取和修改的保护能力。

7. 各种因素的权衡

在很多情况下，可能会发现问题的解要满足明显矛盾的不同因素。此时必须综合考虑，恰当权衡，把准则表达得很清楚。

1.2.3 软件系统构件化与集成构造策略

1. 软件系统构件化

软构件可以看成是软件中的标准件，它是应用域中具有一定集成度的通用或专用软件逻辑单元。生产软构件的直接目的是获得软件重用，即把软件设计过程变换成构造软件逻辑单元的活动，从而使程序可用现存的标准单元进行组合构造，就如同“零件”对于机械设计师，集成电路对于电子工程师一样。“软构件”亦是软件开发人员的基本财富。

研究软构件的目的就在于尽可能地在软件生产中使用现有的软件，而尽量少重复编写具有类似功能的软件。最好有一种集成机制，可以将已经成熟的软件单元制成一个个相对独立的实体，使它们可以不加改动或只做很少改动就可以加入于新软件系统中。只有这样，软件生产才有望减少重复劳动。所以软构件实质上是面向集成的软件重用单元。

2. 软件系统集成构造策略

计算机的硬件设备是由许多插件板连接而成的，而这些插件板又是将许多具有独立功能的集成电路按设计要求组装而成的。要构造计算机系统的硬设备，设计人员不必透彻地了解每个集成电路的内部结构及其实现，只要了解使用要求、功能说明等外部特征，便可以在板上安装集成电路。这样，构造计算机硬设备的主要工作是插件板的设计。计算机硬件的迅速发展与这种构造特征及集成电路的发展是分不开的。

软件系统集成构造的基本思想是：构造者只要对整个系统的构造方案与原则进行精心描述和制定(相当于主板的设计)，将所需软构件按这些原则组织起来(相当于将集成电路装入插件板)，便可以推出一个新的软件系统。这样软件生产率将会获得很大的提高。

软件系统集成构造的特点是：构件合成的结构与问题结构相对应，开发人员把精力放在系统结构的设计上而不是在控制流的设计上。

1.2.4 软构件特征

在工程实现过程中，软件的外部质量是靠内部质量来保证的，软构件特征反映了良好的内部质量因素，因此采用构件化与集成构造策略的软件系统具有良好的可扩充性、可重用性和兼容性等外部质量。

目前很难给软构件特征下定义，但可以从以下几个方面来论述。

1. 系统可分解性

如果一个设计方法能帮助我们把一个新问题分解成若干问题，如果有必要则可继续细分，则称这种方法满足可分解性。系统可分解性是指把系统分解成若干软构件的能力，亦即系统构件化能力。

系统分解通常是重复进行的，直至子系统成为易于理解的构件为止。系统构件化可以降低原有系统的复杂性。由分解得到的不同软构件可以由不同的人去实现。

2. 软构件可集成性

软构件的集成是系统分解的逆过程。系统分解是为了降低系统构造的复杂性，当子系统转换成软构件并在工程上予以实现时，这些软构件必须能被集成为原系统，如图 1.1 所示。

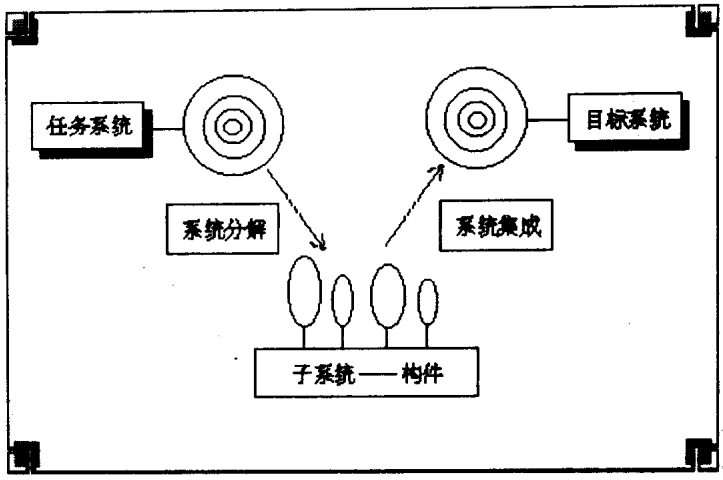


图 1.1　系统分解与集成示意图

前面已经提到，软构件实质上是面向集成的软件重用单元，利用软构件来构造新系统的方法称为集成构造。

构件集成的思想可以应用于设计软构件本身。对于已存在构件不作修改，而仅增加适量的结构元件与功能元件，或直接耦合相关构件，来创造功能更强的软构件，称为软构件的组合设计。一个重用软构件的规模与被重用带来的实际效益是成正比的。软构件的组合设计表达了如下原则：在满足软构件可重用的前提下，尽可能扩大构件的规模，避免产生一些无完整语义的构件碎片。事实上，面向对象的语言中的类组合与类继承(或派生)机制直接支持软构件的组合设计。

3. 软构件的连续性

在构件集成的系统中，当需求说明有些较小的修改时，系统只要求对一个或少数几个相关的构件进行修改，而不必影响整个系统的结构与行为，这个性质称为软构件的连续性。

连续性这个术语借用了数学分析中连续函数的概念(粗略地说，如果一个函数的自变量有一个较小的变化，其函数值也只有一个较小的变化，则称这个函数是连续的)。软构件的连续性特征提高了系统的可扩充性。在构造大系统的过程中，随着项目的进展，几乎所有的规格说明都是会变化的，软件的修改将变得非常困难。软构件的连续性意味着局部的修改不会引起系统的大波动，亦即提高了系统的可扩充性，如图 1.2 所示。

4. 软构件的开放—封闭性

如果一个程序单元仍然可以进行扩充，则称之为开放的，如果一个程序单元可以作为一个相对的独立体被其它程序引用，则称之为封闭的。软构件可以被扩充或被组合成新构件，它具有开放性；软构件是自主与凝聚的，它具有封闭性。

软构件的“开放—封闭”特征看起来是矛盾的，但是在项目实现中是很有效的。当开始着手一个新问题时，我们很难掌握所要解决的问题的全部含义，因此开放性需求是不可避免的。这意味着，在写一个程序之前，先要纵观问题的一些重要方面，同时作好在以后增加新项的准备。但是封闭性也是需要的，因为我们不能等掌握了问题的全部信息后再把程序做成他人能用的构件。这样等待是不现实的，没有封闭性一样不能得到综合性较强的软构件。