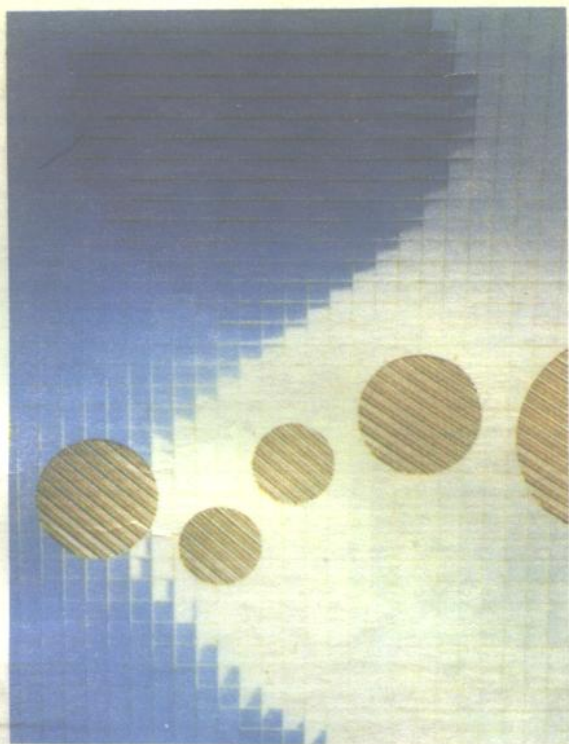




用户接口技术 与 交互系统构造方法

丁茂顺 著

科学出版社



用户接口技术与 交互系统构造方法

丁茂顺 著

科学出版社

1992

(京) 新登字 092 号

内 容 简 介

本书论述了与用户接口相关的概念、技术和方法,同时介绍了进行交互系统研究和开发的一些指导思想和实际知识。本书内容由四部分组成,共十三章。第一部分为概论。第二部分对用户接口中的各项技术分别予以阐述。第三部分介绍了在交互式应用软件的研究与开发中几个典型的基于用户接口软件技术的程序设计方法和软件工程方法。第四部分归纳了软件开发中有关软件接口设计的共性问题。本书对许多技术细节都有重点地给出了相应的数据结构、函数功能和实现思想,有些章节还给出了实用的 C 语言程序。

本书的读者对象为高校计算机软件专业的师生及从事软件开发的科技人员。

用户接口技术与 交互系统构造方法

丁 茂 顺 著

责任编辑 那莉莉

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100707

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1992 年 12 月第 一 版 开本: 850×1168 1/ 32

1992 年 12 月第一次印刷 印张: 9 1/ 8

印数: 1— 7000 字数: 234 000

ISBN 7-03-003095-8/ TP · 227

定价: 7.40 元

前 言

目前，国际上计算机软件正在更新换代，软件工具得到了广泛的使用，友好的用户接口技术在软件界的呼声越来越高。大量的应用软件以用户接口技术为背景，朝着图形化、菜单化和多窗口化的方向发展。用户接口技术的发展造就了一批实用性很强的软件工具系统，反过来它们又促进了交互技术的进一步发展。软件工具的广泛应用使得各种交互系统的研究和开发工作正在突破传统的套路，从而促使人们必须站在方法学的高度上去研究交互系统的构造过程。这显然是软件技术之应用研究的一个引人注目的重要领域。突破并修正受限于机器资源和运行效率的旧式计算机概念，充分地理解、抽象和描述交互软件系统，并进行开拓性的研究和实践，正是用户接口技术和交互系统构造方法所应基于的观点。

本书的大部分内容是作者在所从事的研究工作和教学实践中逐渐形成的，是多年的知识和经验的积累。它论述了与用户接口相关的概念、技术和方法，同时介绍了进行交互系统研究和开发的一些指导思想和实际知识。本书的内容安排贯穿着既动脑又动手的精神，对许多技术的解释，都有重点地给出了相应的数据结构、函数功能和实现思想，有些章节还收进了可实用的 C 语言程序，以此鼓励有兴趣的读者亲自到计算机上去试试，从而逐步学会研究和开发交互软件系统的真功夫。

1987 年夏天，中国科学技术大学研究生院（北京）决定，开设计算机软件接口技术这门课程。从 1988 年春天开始到现在，本书初稿即在该院由作者本人连续讲授过多个学期。此外，以本书所阐述的基本思想为主线，作者也曾在多个单位作过不同类型的学术报告和系列讲座。

本书内容主体由四部分组成，共十三章。

第一部分（即第一章概论）介绍用户接口软件的基本概念、专业动向、研究目标、研究方法以及阅读本书的预备知识。

第二部分（第二章～第八章）对用户接口软件家族中的各个成员，或称为用户接口中的各项管理技术，分别予以阐述。除第八章外，每项管理技术各占一章，内容相近或者重复的部分则加以合并或省略。

第二部分是本书的基本内容，是用户接口的关键技术。一般说来，我们将这些技术分为三类：第一类是事件管理技术（见第二章），讨论事件的相关知识和机制；第二类是屏幕管理技术（见第二章～第六章），分别讨论与屏幕有关的菜单、图符、图象、对话、卷帘以及窗口等各项技术；第三类是资源管理技术（见第七章），论述与用户接口相关的信息存储和映射技术。书中对每项技术的阐述方法大体上都是由外及内，由浅入深，以便于读者有选择性地阅读，对技术细节不感兴趣的读者，可以暂且不读“实现思想”部分。第八章综合前面几章的思想，根据已经建立起来的事件管理、屏幕管理和资源管理等项技术，在较高的层次上研究一种用户接口技术的综合机制，为交互系统的开发者提供一种系统组织模型。

第三部分（第九章～第十二章）介绍了在交互式应用软件的研究与开发活动中，几个典型的基于用户接口软件技术的程序设计方法和软件工程方法。

第九章介绍了一种基于函数库的交互软件的设计和实现方法。实践表明，对于规模不太大的交互软件系统，这种方法实用且高效。

第十章介绍一种基于开发过程模型的交互软件系统构造方法。

第十一章介绍一种基于描述语言的交互软件构造方法。该章参考了第十章的基本思想，综合了近年来有关第四代语言和速成原型法的技术，提出了交互软件系统的一种构造方法。该章主要内容包括构造活动的工作过程模型、具有第四代语言特征的交互

语言以及一个试验系统的简介。

第十二章介绍了一种基于 UIMS 模型的交互系统构造方法。该章阐述了 UIMS 应当采取的组织结构和工作原理，同时还介绍了一个具体的 UIMS 的设计和实现。

第四部分（即第十三章）归纳了软件开发活动中有关软件接口设计的共性问题，提出了系统化的解决办法或想法。这一部分内容仅供从事软件开发的实践者们参考，因为它所涉及的范围较广，似乎已经超出了用户接口技术的范畴。

附录，作为正文的补充，由两部分组成。第一部分包括菜单生成器和图符生成器两个课程练习实例。为避免篇幅冗长，只对它们作了简要介绍。第二部分对本书中所出现的数据类型、工具函数和关键词等列出索引。

本书可作为大专院校计算机软件专业的高年级本科生或研究生的授课教材，也可作为那些从事软件研究与开发工作的教学人员和工程技术人员的参考书。

在本书所讨论的领域内，国内已陆续出现了一些很实用的、服务于具体系统的印刷品（手册或说明书），但正式出版的、较严谨的类似书籍尚少见。比较而言，本书具有如下特点：

(1) 本书强调研究用户接口新技术，以当代国际上实用的用户接口管理系统的最新产品为研究背景，同时也辅以适当的我们自己从事的实际工作的内容。

(2) 本书的内容安排总是力求在较高的层次上使理论与实践相结合，鼓励读者自己上机实习，以便将书本知识迅速地转化为读者的实际本领。

(3) 本书初稿多年来一直作为研究生的授课讲义，注重循序渐进、由浅入深。但是，本书内容的基本部分，特别是前几章所介绍的技术和方法，对于任何从事计算机工作的工程技术人员，都是非常基础和实用的。

如何面向实际的交互系统，并对日益流行和大量应用的用户接口进行分析和研究，从而提供一种合理的运行和管理机制，给

人们在用户接口管理系统以及构造交互系统的实践中以指导，无疑是本书所具有的需求价值和应用前景。在从事交互系统研究和应用的各科研单位，在作为学习和掌握用户接口管理技术基本知识和技能的各教学单位，以及在开发和普及计算机应用的各基层单位，本书的读者都会在不同的层次上针对自己的需求找到答案并受到启迪。

在本书的成稿过程中，曾先后得到中国科学院软件研究所许孔时、张尤腊、董韞美、仲萃豪、陶仁骥、张锦文、王裕国和顾毓清等几位教授的鼓励和支持，得到中国科技大学冯玉琳、北京大学董士海、中国科学院计算技术研究所王庆元、中国科学院计算中心张绮霞和中国科学院自动化研究所戴国忠等教授的真诚帮助。我的同事和研究生宗丽苹、陈晓真、王晓茜、许修平、易章雄、于昕和李新宁等为本书的成稿，花费了一定的时间和精力。宗丽苹仔细地阅读、整理了全部初稿，帮助改写了部分章节，并且和我共同撰写了第十一章。对以上各位老师、同事和朋友们的热情帮助和辛勤劳动，作者谨致以深切的谢意。

目 录

前 言

第一部分 用户接口的基本知识 (1)

第一章 概论 (2)

- 1.1 用户接口的基本概念 (2)
- 1.2 用户接口软件应当具备的特征 (4)
- 1.3 研究用户接口的目标 (5)
- 1.4 用户接口的研究方法 (7)
- 1.5 与用户接口相关的系统配置 (9)
- 1.6 研究用户接口的预备知识 (9)
- 1.7 用户接口相关新技术点滴 (13)

第二部分 用户接口的基本技术 (15)

第二章 事件 (17)

- 2.1 事件的概念 (17)
- 2.2 事件数据结构的描述 (18)
- 2.3 事件管理 (22)
- 2.4 事件驱动程序的控制结构 (26)
- 2.5 程序实例 (30)

第三章 菜单 (37)

- 3.1 菜单的概念 (37)
- 3.2 菜单的分类 (39)
- 3.3 菜单的数据结构 (44)
- 3.4 菜单数据的取值方法 (46)
- 3.5 菜单的显示方法与选取机制概述 (50)

3.6	菜单管理	(53)
3.7	应用软件中菜单驱动机制的要点	(62)
3.8	程序实例	(63)
第四章	图象和图符	(69)
4.1	图象的概念	(69)
4.2	图象映射原语	(71)
4.3	图象的数据结构	(76)
4.4	图象处理	(77)
4.5	图符	(79)
4.6	图象在用户接口中的其他应用	(85)
4.7	程序实例	(88)
第五章	对话	(92)
5.1	对话的概念	(92)
5.2	简单的标准对话应用演示	(93)
5.3	标准对话的数据结构	(95)
5.4	对话管理	(97)
5.5	关于标准对话的几点注记	(100)
5.6	关于订做式对话	(103)
5.7	评述	(105)
第六章	窗口	(106)
6.1	窗口的概念	(106)
6.2	窗口配置简述	(107)
6.3	窗口的数据结构	(111)
6.4	窗口管理	(113)
6.5	多窗口系统控制	(122)
第七章	资源	(124)
7.1	资源的概念	(124)
7.2	资源管理概述	(126)
7.3	系统资源管理	(128)
7.4	应用资源管理	(134)

7.5	资源管理的一般设计原则	(138)
7.6	资源管理技术的实用示例	(139)
第八章	用户接口技术的综合机制	(141)
8.1	概念	(141)
8.2	作为用户工作空间(屏幕)组织者的窗口技术	(141)
8.3	作为用户操作时序组织者的事件技术	(145)
8.4	作为用户工作后备支持的资源管理	(147)
8.5	交互软件系统的组织	(147)
8.6	程序实例	(147)
第三部分	开发和构造交互软件的方法学	(151)
第九章	以组织库函数为特征的交互软件开发方法	(154)
9.1	概述	(154)
9.2	以组织库函数为特征的交互软件开发步骤要点	(154)
9.3	交互软件系统的结构框架	(158)
9.4	关于事件驱动机制的补充说明	(159)
9.5	评述	(159)
第十章	基于一种开发过程模型的交互软件构造方法	
——AIH		(161)
10.1	概述	(161)
10.2	环境 AIH 的系统配置情况简介	(161)
10.3	交互语言 IL 简介	(163)
10.4	AIH 方法学的开发过程模型	(168)
10.5	应用实例	(169)
10.6	评述	(173)
第十一章	基于描述语言的交互软件构造方法	
——BLISS		(175)
11.1	概述	(175)
11.2	构造交互系统的工作过程模型	(176)
11.3	状态转换图和中间语言的特性	(178)

11.4	构造方法的环境试验系统	(182)
11.5	评述	(188)
11.6	一个简单的应用实例——Diagram-Builder	(189)
11.7	ML 语言文法	(227)
第十二章 基于 UIMS 模型的构造方法		(233)
12.1	概述	(233)
12.2	UIMS 的三大部件及其职责	(234)
12.3	表现部件的原理和结构	(236)
12.4	对话控制部件的原理和结构	(239)
12.5	应用接口模型的原理和结构	(244)
12.6	系统实现概况	(246)
12.7	评述	(250)
第四部分 软件开发活动中的接口设计考虑		(251)
第十三章 软件开发活动中的接口设计考虑		(252)
13.1	概述	(252)
13.2	子系统之间的接口设计考虑	(253)
13.3	模块之间的接口设计考虑	(257)
13.4	接口文件设计考虑	(260)
13.5	用户接口设计考虑	(262)
附录 A	菜单生成器概述	(263)
附录 B	图符生成器提纲	(267)
附录 C	索引	(269)
参考文献		(276)

第 一 部 分

用户接口的基本知识

这一部分概述研究用户接口软件时所应当掌握和具备的基本知识，介绍用户接口软件的基本概念、专业动向、研究目标、研究方法、用户接口新技术以及阅读本书的预备知识。

第一章 概 论

1.1 用户接口的基本概念

我们现在来系统地考察一下计算机软件家族中的一名重要成员——计算机软件用户接口，简称用户接口（user interface）。用户接口是一类软件工具或软件系统，由它们帮助实现计算机系统、应用软件系统和使用系统的人（即用户）三者之间的恰当联结和协调一致；也可以说，用户接口是专门处理人机交互活动的软件。这里的用户主要是指终端用户（end user）；而应用软件则主要是指那些在其执行过程中频繁地与终端用户对话的软件，通常称之为交互软件或交互系统（interactive system）。下面再进一步解释一下这个定义的几个侧面：

(1)用户接口这一术语已经在计算机界广泛使用，它指的是软件，与硬件上常说的线路与插件等接口并不相干。

(2)接口一词顾名思义，表示它介乎于诸软件成员和人之间，发挥联结与协调的功能。

(3)用户一词强调人的因素，突出表明用户接口理所当然地服务于基本上属于人机交互类型的环境；换句话说，传统的批量作业环境很难得到用户接口的服务。

(4)用户接口作为软件，其大小与完善程度完全视环境与需求而定。简单的用户接口可以是一组软件工具，以源码或目标库的形式提供给使用者；比较复杂的用户接口可以根据用户的需求说明自动地生成所需要的可执行系统。但不管是简单的还是复杂的，用户接口都必须具备一个最基本的特征，即最大限度地将交互式应用系统中具有共性的人机交互部分游离出来，自成体系，以便为更多的应用系统服务，从而大大地简化应用系统的开发过

程。

(5)计算机系统通常是指计算机硬件和系统软件两者的总称。

为了以后的叙述方便和避免混淆，我们对那些与用户接口有关人员称谓作一个大体上的约定：

(1)终端用户。指的是在机器上操作（通过终端）而实际使用软件系统的人。这里的终端可以是大型分时系统的终端，也可以是个人计算机或工作站的一组键盘、鼠标器和显示器等。终端用户可以是操作员，也可以是有关领域的专业人员或管理人员，例如，银行系统的终端用户就常常是会计师、出纳员或银行经理。

(2)软件工程师。原来是指作系统分析和系统设计等项工作的软件人员；按工作性质，它又细分为应用（软件）工程师、系统（软件）工程师和用户接口（软件）工程师。习惯上常常把这一类人员统称为系统开发者，甚至简单地称为程序员或高级程序员。此处应当强调指出的是，必须清晰地终端用户和系统开发者之间划一道分界线，这一点在全书的叙述中是至关重要的。

本书所叙述的用户接口知识、技术和系统工程方法等内容，基本上是 80 年代才发展起来的。尽管这种发展、演变过程包含了非常丰富的细节，但大体上仍然可以用图 1-1 形象地表示出来。图中“宿主系统”这一术语主要是指操作系统、编译程序以及相关的系统实用程序，另外还包括机器硬件；“计算部分”指纯属应用软件本身的内容，而主要的用户接口部分已经游离出来了。

计算机图形学的迅速发展，以及若干种带有图形显示的软件产品的相继问世，使得用户接口作为一类独立的软件日益成熟起来。用户接口软件所具有的成分和特征，通过一些著名的软件产品（例如 Smalltalk 和 Macintosh）得到体现；随着这些软件产品的推广使用，用户接口软件也就逐渐地为人们所广泛接受。在学术界，关于用户接口的研究和讨论，近几年来一直活跃不衰，有关的国际会议也非常频繁。笔者印象较为深刻的是美国乔治·华盛顿大学的 J.Foley 教授等人所做的工作——他们实现了图形

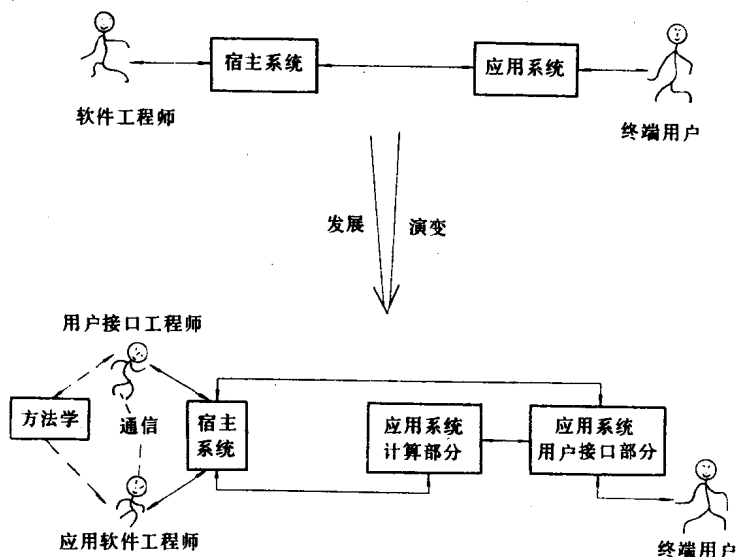


图 1-1 用户接口发展演变示意图

学的一个标准，并在此基础上发展了有关用户接口的、带全局性的思想以及相应的软件工程方法学。Foley 教授在 1983 年曾写道：

“我们称那些从事用户接口设计的人员为用户接口建筑师 (architect)。我们有意识地选择这个名称，是因为这种人与楼房建筑师的传统职责有着很多的类似之处……不幸的是，只有几所（美国）大学和学院开设了较强的教育和研究计划，为培养未来的用户接口建筑师打基础。”

Foley 教授在这方面的研究工作，对于后文中用户接口技术、方法的研究具有一定的指导意义。

1.2 用户接口软件应当具备的特征

作为一种独立的软件类别，用户接口软件应当具备什么样的特征呢？以下四点是必不可少的：

(1)使用简单。用户接口软件作为工具，要让“建筑师”们用起来简单、顺手；在组装成应用系统时，终端用户用不着花太多的时间和精力就可学会使用这些工具（或用户接口工具函数）。

(2)响应合理。用户接口软件的基本使命是设置一定的机制，以真正实现终端用户和系统之间的交互活动。交互活动的质量主要是看系统响应是否合理：响应时间不可太久，太久了会使终端用户不耐烦；响应的内容要合理，例如，合乎逻辑的出错处理和提示、帮助信息等。

(3)一致性（consistency）。一致性的特征表现在系统的各种机制、术语甚至关键词、符号等方面，一经约定含义，就自始至终不加改变。

(4)熟悉性（familiarity）。熟悉性的特征表现在系统开发时不是人为地杜撰一些令人费解的行话，而是尽可能地使用那些人们已经用惯了的话语。

1.3 研究用户接口的目标

用户接口的研究已经成了当前软件界的热门话题之一。尽管研究的内容和方法在表述上千差万别，但大家的研究目标却可以归纳为以下几个方面。

1.3.1 将分散的用户接口成分从现有的应用软件中游离出来

将散布于各个现有的（交互）应用软件之中的具有共性的用户接口成分游离出来，成为独立的软件成分，为其他软件所共享，在目前有着巨大的意义。

这里所说的“游离”，显然包含着很丰富的实际内容——它是对原有的应用系统中的用户接口部分进行识别、抽象、设计和验证的过程。找出应用软件中的一个（些）部分，这个（些）部分只与人机交互活动有关而与应用本身的特定内容无关，这就是识别；将所识别出的各个部分进行比较鉴别，从相同或相似各部

分之中归纳出一个共同的功能，这就是抽象；从所得到的功能出发，设计出一个独立的软件部分（例如函数或子程序等），这就是设计；将所设计的结果代入到原有的应用软件中（有时可能需要对应用部分作较大的修改），考察其执行效果是否已达预期目标，这就是验证。对于这种游离，可以用操作系统的 I/O 加工来类比——目前的 I/O 加工就是对早先分散于应用系统中的对数据变量及其格式的描述作输入输出处理的部分进行游离而成的（当时还没有操作系统）。

进行上述的游离，其好处是显而易见的：

(1)减少重复开发，提高软件生产率。这是因为既然系统中有关用户接口的部分已经游离出来了，那么就没有必要让每个应用软件再去背一个加工用户接口的包袱。

(2)提高应用软件的可移植性。这是因为原有的、依赖于设备与环境的软件部分已经划归为用户接口所管辖，因而就使得应用软件较少地依赖于甚至根本不依赖于具体的设备与环境，从而提高了系统的可移植性。

(3)提高软件质量。由于分工明确，所以使得在应用软件和用户接口两个方面都进行深入细致的研究成为可能，从而在总体上提高了软件系统的质量。

要达到将用户接口成分从应用软件中游离出来的目标并非易事，问题往往出在那些用传统的方法开发应用系统的人身上。在这类系统中，开发者经常将应用的计算部分和用户接口部分无规则地搅和在一起，很难分开。

1.3.2 开发具有共性的用户接口软件工具

考察用户接口的共性，开发出相应的一批用户接口软件工具，为多种应用服务。例如，在应用这一级，组织那些已得到的用户接口软件工具，快速地生成新的应用系统的原型。这种方法通常称为速成原型法（rapid prototyping），我们也称之为“搭积木”法。采用这种方法时，由于在开始阶段，应用软件部分尚无