

# 用户界面设计

## 要点浏览

**概念:** 用户界面 (UI) 设计在人与计算机之间搭建了一个有效的交流媒介。遵循一系列的界面设计原则, 定义界面对象和界面动作, 然后创建构成用户界面原型基础的屏幕布局。

**人员:** 软件工程师通过迭代过程来设计用户界面, 这个过程采纳了被广泛接受的设计原则。

**重要性:** 不管软件展示了什么样的计算能力、发布了什么样的内容及提供了什么样的功能, 如果软件不方便使用、常导致用户犯错或者不利于完成目标, 你是不会喜欢这个软件的。由于界面影响用户对于软件的感觉, 因此, 它必须是令人满意的。

**步骤:** 用户界面设计首先要识别用户、任

务和环境需求。一旦确定用户任务, 则通过创建和分析用户场景来定义一组用户界面对象和动作。这是创建屏幕布局的基础。屏幕布局描述了图标的图形设计和位置、描述性屏幕文本的定义、窗口的规格说明和命名, 以及主要的和次要的菜单项规格说明。可以使用工具来开发原型并最终实现设计模型, 另外为了保证质量需要对结果进行评估。

**工作产品:** 创建用户场景, 构建产品屏幕布局, 以迭代的方式开发和修改界面原型。

**质量保证措施:** 原型的开发是通过用户测试驱动的, 测试驱动的反馈将用于原型的下一次迭代修改。

我们生活在充满高科技产品的世界里。几乎所有这些产品, 诸如消费电子产品、工业设备、汽车产品、企业系统、军事系统、个人计算机软件、移动 App 及 WebApp, 都需要人参与交互。如果要使一个产品取得成功, 它就必须展示出良好的可用性。可用性是指用户在使用高科技产品所提供的功能和特性时, 对使用的容易程度和有效程度的定量测量。

在计算时代的前 30 年里, 可用性并不是软件开发主要关心的。Donald Norman[Nor88] 在其关于设计的经典书籍中曾经主张 (对待可用性的) 态度改变的时机已经到来:

为了使技术适应人类, 必须要研究人类。但我们现在倾向于只研究技术。结果, 人们不得不顺从技术。而现在是时候扭转这个趋势, 使技术适应人类了。

随着技术专家对人类交互的研究, 出现了两个主要的问题。第一, 定义一组黄金规则 (15.1 节)。这些规则可以应用于所有与人交互的技术产品。第二, 定义交互机制使软件设计人员建立起可以恰当实现黄金规则的

### 关键概念

- 可访问性
- 命令标记
- 控制
- 设计评估
- 错误处理
- 黄金规则
- 帮助设施
- 界面分析
- 界面一致性
- 界面设计
- 界面设计模型
- 国际化
- 记忆负担
- 原则和指导方针
- 过程

系统。这些机制消除了机器与人交互方面的一些难题，我们统一称之为用户界面。但即便是在今天，我们还是会遇到这样的用户界面：难学、难用、令人迷惑、不直观、不可原谅。在很多情况下，它们让人感到十分沮丧。然而，仍然有人在花费时间和精力去创建这样的界面，看起来，创建者并不是有意制造麻烦。

## 15.1 黄金规则

Theo Mandel 在其关于界面设计的著作 [Man97] 中提出了三条黄金规则：

1. 把控制权交给用户。
2. 减轻用户的记忆负担。
3. 保持界面一致。

这些黄金规则实际上构成了一系列用户界面设计原则的基础，这些原则可以指导软件设计的重要方面。

### 15.1.1 把控制权交给用户

在重要的、新的信息系统的需求收集阶段，曾经征求一位关键用户对于窗口图形界面相关属性的意见。

该用户严肃地说：“我真正喜欢的是一个能够理解我想法的系统，它在我需要去做以前就知道我想做什么，并使我可以非常容易地完成。这就是我想要的，我也仅此一点要求。”

你的第一反应可能是摇头和微笑，但是，沉默了一会儿后，你会觉得该用户的想法绝对没有什么错。她想要一个对其要求能够做出反应并帮助她完成工作的系统。她想去控制计算机，而不是计算机控制她。

设计者施加的大多数界面约束和限制都是为了简化交互模式。但是，这是为了谁呢？

在很多情况下，设计者为了简化界面的实现可能会引入约束和限制，其结果可能是界面易于构建，但会妨碍使用。Mandel[Man97] 定义了一组设计原则，允许用户掌握控制权。

以不强迫用户进入不必要的或不希望的动作的方式来定义交互模式。交互模式就是界面的当前状态。例如，如果在字处理器菜单中选择拼写检查，则软件将转移到拼写检查模式。如果用户希望在这种情形下进行一些文本编辑，则没有理由强迫用户停留在拼写检查模式，用户应该能够几乎不需做任何动作就可以进入和退出该模式。

提供灵活的交互。由于不同的用户有不同的交互偏好，因此应该提供选择机会。例如，软件可能允许用户通过键盘命令、鼠标移动、数字笔、触摸屏或语音识别命令等方式进行交互。但是，每个动作并非要受控于每一种交互机制。例如，考虑使用键盘命令（或语音输入）来画一幅复杂形状的图形是有一定难度的。

允许用户交互被中断和撤销。即使陷入一系列动作之中，用户也应该能够中断动作序列去做某些其他事情（而不会失去已经做过的工作）。用户也应该能够“撤销”任何动作。

当技能水平高时可以使交互流线化并允许定制交互。用户经常发现他们重复地完成相同的交互序列，因此，值得设计一种“宏”机制，使得高级用户能够定制界面以方便交互。

#### 关键概念

响应时间  
任务分析  
任务细化  
可用性  
用户分析  
WebApp 和移动  
App 界面设计

317

**引述** 依照用户的习惯来设计，比纠正用户的习惯要好。

Jon Meads

318

**引述** 我一直希望计算机能像电话一样易于使用。我的希望已经实现了，我不再想知道如何使用电话了。

Bjarne Stronstrup

使用户与内部技术细节隔离开来。用户界面应该能够将用户移入应用的虚拟世界中，用户不应该知道操作系统、文件管理功能或其他隐秘的计算技术。

设计应允许用户与出现在屏幕上的对象直接交互。当用户能够操纵完成某任务所必需的对象，并且以一种该对象好像是真实存在的方式来操纵它时，用户就会有一种控制感。例如，允许用户将文件拖到“回收站”的应用界面，即是直接操纵的一种实现。

### 15.1.2 减轻用户的记忆负担

319

一个经过精心设计的用户界面不会加重用户的记忆负担，因为用户必须记住的东西越多，和系统交互时出错的可能性也就越大。只要可能，系统应该“记住”有关的信息，并通过有助于回忆的交互场景来帮助用户。Mandel[Man97]定义了一组设计原则，使得界面能够减轻用户的记忆负担。

**减少对短期记忆的要求。**当用户陷于复杂的任务时，短期记忆的要求会很强烈。界面的设计应该尽量不要要求记住过去的动作、输入和结果。可行的解决办法是通过提供可视的提示，使得用户能够识别过去的动作，而不是必须记住它们。

**建立有意义的默认设置。**初始的默认集合应该对一般的用户有意义，但是，用户应该能够说明个人的偏好。然而，“重置”(reset)选项应该是可用的，使得可以重新定义初始默认值。

**定义直观的快捷方式。**当使用助记符来完成系统功能时(如用Alt+P激活打印功能)，助记符应该以容易记忆的方式联系到相关动作(例如，使用要激活任务的第一个字母)。

**界面的视觉布局应该基于真实世界的象征。**例如，一个账单支付系统应该使用支票簿和支票登记簿来指导用户的账单支付过程。这使得用户能够依赖于很好理解的可视化提示，而不是记住复杂难懂的交互序列。

**以一种渐进的方式揭示信息。**界面应该以层次化方式进行组织，即关于某任务、对象或行为的信息应该首先在高抽象层次上呈现。更多的细节应该在用户表明兴趣后再展示。

## SafeHome 违反用户界面的黄金规则

[场景] Vinod的工作间，用户界面设计启动在即。

[人物] Vinod和Jamie，SafeHome软件工程团队成员。

[对话]

Jamie：我已经在考虑监控功能的界面了。

Vinod(微笑)：思考是好事。

Jamie：我认为我们可以将其简化。

Vinod：什么意思？

Jamie：如果我们完全忽略住宅平面图会怎么样？它倒是很华丽，但是会带来很多开发工作量。我们只要询问用户要查看的指定摄像机，然后在视频窗口显示视频就

可以了。

Vinod：房主如何记住有多少个摄像机以及它们都安装在什么地方呢？

Jamie(有点不高兴)：他是房主，应该知道。

Vinod：但是如果不知道呢？

Jamie：应该知道。

Vinod：这不是问题的关键……如果忘记了昵？

Jamie：哦，我应该提供一张可操作的摄像机及其位置的清单。

Vinod：那也有可能，但是为什么要有一份清单呢？

Jamie：好的，无论用户是否有这方面的

320

要求，我们都提供一份清单。

**Vinod**：这样更好。至少用户不必特意记住我们给他的东西了。

**Jamie**（想了一会儿）：但是你喜欢住宅平面图，不是吗？

**Vinod**：哈哈。

**Jamie**：你认为市场营销人员会喜欢哪一个？

**Vinod**：你在开玩笑，是吗？

**Jamie**：不。

**Vinod**：哦……华丽的那个……他们喜欢迷人的……他们对简单的不感兴趣。

**Jamie**：（叹口气）好吧，也许我应该为两者都设计一个原型。

**Vinod**：好主意……我们就让客户来决定。

### 15.1.3 保持界面一致

用户应该以一致的方式展示和获取信息，这意味着：（1）按照贯穿所有屏幕显示的设计规则来组织可视信息；（2）将输入机制约束到有限的集合，在整个应用中得到一致的使用；（3）从任务到任务的导航机制要一致地定义和实现。Mandel[Man97]定义了一组帮助保持界面一致性的设计原则。

允许用户将当前任务放入有意义的环境中。很多界面使用数十个屏幕图像来实现复杂的交互层次。提供指示器（例如，窗口标题、图标、一致的颜色编码）帮助用户知晓当前工作环境是十分重要的。另外，用户应该能够确定他来自何处以及存在哪些转换到新任务的途径。

在完整的产品线内保持一致性。一个应用系列（即一个产品线）都应采用相同的设计规则，以保持所有交互的一致性。

如果过去的交互模型已经建立起了用户期望，除非有不得已的理由，否则不要改变它。一个特殊的交互序列一旦变成事实上的标准（如使用 Alt+S 来存储文件），则用户在遇到每个应用时均会如此期望。如果改变这些标准（如使用 Alt+S 来激活缩放比例），将导致混淆。

本节和前面几节讨论的界面设计原则为软件工程师提供了基本指南。在下面几节中，我们将考察界面设计过程。

**引述** 看起来不同的事物产生的效果应该不同，而看起来相同的事物产生的效果应该相同。

Larry Marine

## 信息栏 可用性

在一篇关于可用性的见解深刻的论文中，Larry Constantine[Con95]提出了一个与可用性主题非常相关的问题：“用户究竟想要什么？”他给出了下面的回答。

“用户真正想要的是好的工具。所有的软件系统，从操作系统和语言到数据录入和决策支撑应用软件，都是工具。最终用户希望从为其设计的工具中得到的东西，与我们希望从所使用工具中得到的是

一样的。他们想要易于学习并能够为自己的工作提供帮助。同时，他们想要的系统应该能提高工作效率，不会欺骗或困扰他们，不会使他们易于犯错误或难于完成工作。”

Constantine 指出，系统的可用性并非取决于设计美学、交互技术的发展水平或者内置的界面智能等方面，而是当界面的架构适合于将要使用这些界面的用户的需

求时,才能获得可用性。

正式的可用性定义往往令人有些迷惑。Donahue 和他的同事 [Don99] 给出了如下的定义:“可用性是一种衡量计算机系统好坏的度量……便于学习;帮助初学者记住他们已经学到的东西;降低犯错的可能;使得用户更加有效率,并且使得他们对系统感到满意。”

确定你所建系统是否可用的唯一办法就是进行可用性评估和测试。观察用户与系统的交互,同时回答下列问题 [Con95]:

- 在没有连续的帮助或用法说明的情况下,系统是否便于使用?
- 交互规则是否能够帮助一个知识渊博的用户工作得更加有效率?
- 随着用户的知识不断增多,交互机制是否能变得更灵活?
- 系统是否已经过调试,使之适应其运行的物理环境和社会环境?

- 用户是否意识到系统的状态?在工作期间,用户是否能够知道其所处的位置?
- 界面是否是按照一种合理并且一致的方式来构建的?
- 交互机制、图标和过程是否在整个界面中一致?
- 交互是否能够提前发现错误并帮助用户修正它们?
- 界面是否能够容错?
- 交互是否简单?

如果上述每个问题的回答都是肯定的,那么可以认为这个系统是可用的。

可用性好的系统带来的诸多好处在于 [Don99]: 提高销售量和用户满意度、具有竞争优势、在媒体中获得良好的评价、获得良好的口碑、降低支持成本、提升最终用户生产力、降低培训费用、减少文档开销、减少来自不满意用户的投诉。

## 15.2 用户界面的分析和设计

用户界面的分析和设计全过程始于创建不同的系统功能模型(从外部看时对系统的感觉)。首先将完成系统功能的任务分为面向人的和面向计算机的,然后考虑那些应用到界面设计中的各种设计问题。可以使用各种工具来建造原型并最终实现设计模型,最后由最终用户从质量的角度对结果进行评估。

### 15.2.1 用户界面分析和设计模型

分析和设计用户界面时要考虑四种模型:工程师(或者软件工程师)建立用户模型;软件工程师创建设计模型;最终用户在脑海里对界面产生映像,称为用户的心理模型或系统感觉;系统的实现者创建实现模型。不幸的是,这四种模型可能相差甚远。界面设计人员的任务就是消解这些差距,导出一致的界面表示。

用户模型确立了系统最终用户的轮廓(profile)。Jeff Patton[Pat07]在《用户为中心的设计》(user-centric design)的前言中写道:

事实是,设计者和开发者(包括我自己)都经常考虑到用户。然而,在缺少特定用户有力的心理模型的情况下,开发者和设计者会以自我来替代用户。自我替代并不是用户为中心,而是自我为中心。

**网络资源** 可以在 [www.nngroup.com](http://www.nngroup.com) 找到用户界面设计信息的优秀资源。

**引述** 如果用户界面有一点瑕疵,那么整个用户界面都会被破坏。

Douglas  
Anderson

为了建立有效的用户界面，“开始设计之前，必须对预期用户加以了解，包括年龄、性别、身体状况、教育、文化和种族背景、动机、目标以及性格”[Shn04]。此外，可以将用户分类：新手，对系统有了解的用户，间歇用户或对系统有了解的经常用户。

用户的心理模型（系统感觉）是最终用户在脑海里对系统产生的印象，例如，请某个餐厅评级移动 App 的用户来描述其操作，那么系统感觉将会引导用户的回答，准确的回答取决于用户的经验（新手只能做简要的回答）和用户对应领域软件的熟悉程度。一个对餐厅评级应用程序有深刻了解但只使用这种系统几次的用户，可能比已经使用该系统好几个星期的新手对该应用程序的功能描述回答得更详细。

实现模型组合了计算机系统的外在表现（界面的观感），结合了所有用来描述系统语法和语义的支撑信息（书、手册、录像带、帮助文件）。当系统实现模型和用户心理模型相一致的时候，用户通常就会对软件感到很舒服，使用起来就很有效。为了将这些模型融合起来，所开发的设计模型必须包含用户模型中的一些信息，实现模型必须准确地反映界面的语法和语义信息。

**建议** 即使用户是新手也会有使用快捷键的需求；即使是经常使用系统的用户有时候也需要指导。他们的要求都要满足。

**引述** 注意用户的行为而不是他们的言语。

Jakob Nielsen

## 15.2.2 过程

用户界面的分析和设计过程是迭代的，可以用类似于第4章讨论过的螺旋模型表示。如图15-1所示，用户界面分析和设计过程开始于螺旋模型的内部，且包括四个不同的框架活动[Man97]：（1）界面分析和建模；（2）界面设计；（3）界面构建；（4）界面确认。图15-1中的螺旋意味着每个活动都将多次出现，每绕螺旋一周表示需求和设计的进一步细化。在大多数情况下，构建活动涉及原型开发——这是唯一实用的确认设计结果的方式。

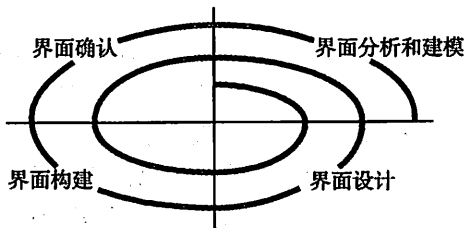


图 15-1 用户界面的设计过程

323

界面分析活动的重点在于那些与系统交互的用户的轮廓。记录技能级别、业务理解以及对新系统的一般感悟，并定义不同的用户类别。对每个用户类别进行需求引导。本质上，软件工程师试图去理解每类用户的系统感觉（15.2.1节）。

一旦定义好了一般需求，就将进行更详细的任务分析。标识、描述和细化（通过绕螺旋的多次迭代）用户为了达到系统目标而执行的任务。15.3节将对任务分析进行更详细的讨论。最后，用户环境的分析着重于物理工作环境的特征（例如地理位置、采光、位置约束）。

作为分析动作的一部分而收集的信息被用于创建界面的分析模型。使用该模型作为基础，设计活动便开始了。

界面设计的目标是定义一组界面对象和动作（以及它们的屏幕表示），使得用户能够以满足系统所定义的每个使用目标的方式完成所有定义的任务。界面设计将在15.4节详细讨论。

界面构建通常开始于创建可评估使用场景的原型。随着迭代设计过程的继续，用户界面开发工具（15.5节）可用来完成界面的构造。

界面确认着重于：（1）界面正确地实现每个用户任务的能力，适应所有任务变化的能力以及达到所有一般用户需求的能力；（2）界面容易使用和学习；（3）作为工作中的得



力工具，用户对界面的接受程度。

324

如我们已经提到的，本节描述的活动是以迭代方式开展的。因此，不需要在第一轮就试图刻画所有的细节（对分析或设计模型而言）。后续的过程将细化界面的任务细节、设计信息和运行特征。

### 15.3 界面分析<sup>①</sup>

所有软件工程过程模型的一个重要原则是：在试图设计一个解决方案之前，最好对问题有所理解。在用户界面的设计中，理解问题就意味着了解：（1）通过界面和系统交互的人（最终用户）；（2）最终用户为完成工作要执行的任务；（3）作为界面的一部分而显示的内容；（4）任务处理的环境。在接下来的几节中，为了给设计任务建立牢固的基础，我们来检查界面分析的每个成分。

#### 15.3.1 用户分析

在担忧技术上的问题之前，用户界面这个词完全有理由要求我们花时间去理解用户。之前，我们提到每个用户对于软件都存在心理映像，而这可能与其他用户的心理映像存在着差别。另外，用户的心理映像可能与软件工程师的设计模型相距甚远。设计师能够将得到的心理映像和设计模型聚合在一起的唯一办法就是努力了解用户，同时了解这些用户是如何使用系统的。为了完成这个任务，可以利用各种途径（用户访谈、销售输入、市场输入、支持输入）获得的信息。

下列一组问题（改编自 [Hac98]）将有助于界面设计师更好地理解系统的用户：

- 用户是经过训练的专业人员、技术员、办事员，还是制造业工人？
- 用户平均正规教育水平如何？
- 用户是否具有学习书面资料的能力或者是否渴望接受集中培训？
- 用户是专业录入人员还是键盘恐惧者？
- 用户群体的年龄范围如何？
- 是否需要考虑用户的性别差异？
- 如何为用户完成的工作提供报酬？
- 用户是否在正常的办公时间内工作或者一直干到工作完成？
- 软件是用户所完成工作中的一个集成部分，还是偶尔使用一次？
- 用户群中使用的主要交流语言是什么？
- 如果用户在使用软件的过程中出错，结果会怎么样？
- 用户是否是系统所解决问题领域的专家？
- 用户是否想了解界面背后的技术？

**提问** 我们如何知道最终用户的人数和特征？

325

这些问题和类似问题的答案将帮助设计师了解：最终用户是什么人，什么可能令他们感到愉悦，如何对用户进行分类，他们对系统的心理模型是什么样子，用户界面必须具有哪些特性才能满足用户的需求。

<sup>①</sup> 因为需求分析问题在第 8~11 章已经讨论过，所以有理由把这一节放到这几章中去。本节之所以放在这里，是因为界面的分析和设计紧紧相连，两者的界限常常模糊不清。

### 15.3.2 任务分析和建模

任务分析的目标就是给出下列问题的答案：

- 在指定环境下用户将完成什么工作？
- 用户工作时将完成什么任务和子任务？
- 在工作中用户将处理什么特殊的问题域对象？
- 工作任务的顺序（ workflow ）如何？
- 任务的层次关系如何？

为了回答这些问题，软件工程师必须利用本书前面所讨论的分析技术，只不过在此种情况下，要将这些技术应用到用户界面。

**用例。**在前面几章中，我们提到过用例描述了参与者（在用户界面设计中，参与者通常是某个人）和系统的交互方式。作为任务分析的一部分，设计用例用来显示最终用户如何完成指定的相关工作任务。在大多数情况下，用例采用第一人称并以非正式形式（一段简单的文字）来书写。例如，假如一家小的软件公司想专门为公司室内设计师开发一个计算机辅助设计系统。为了更好地理解他们是如何工作的，实际的室内设计师应该描述特定的设计功能。在室内设计师被问到“如何确定室内家具摆放位置”的时候，室内设计师写下了如下非正式的用例描述：

我从勾画房间的平面图、窗户与门的尺寸和位置开始设计。我非常关心射入房间的光线，关心窗外的风景（如果它很漂亮，就会吸引我的注意力），关心无障碍墙的长度，关心房间内活动空间的通道大小。我接下来会查看客户和我选取的家具清单……接着，我会为客户画出一个房屋的透视图（三维图画），让客户感受到房间看起来应该是什么样的。

这个用例给出了计算机辅助设计系统中一项重要工作任务的基本描述。从这个描述中，软件工程师能够提炼出任务、对象和整个交互流程。另外，系统中能够使得室内设计师感到愉悦的其他特征也被构思出来。例如，可以将房屋中每一扇窗户的风景都拍摄成一张数码相片。在画房屋透视图时，通过每扇窗户就可以看到窗外的真实景象。

**关键点** 用户的目的是通过用户界面来完成一个或多个任务。为了实现这一点，用户界面必须提供用户达到目标的机制。

**网络资源** 可以在 <http://web.eecs.umich.edu/~kieras/docs/GOMS/> 找到不错的用户建模信息资源。

326

### SafeHome 用户界面设计的用例

**[场景]** Vinod 的工作间，用户界面设计正在进行。

**[人物]** Vinod 和 Jamie，SafeHome 软件工程团队成员。

**[对话]**

**Jamie：**我拦住我们的市场部联系人，让她写了一份监视界面的用例。

**Vinod：**站在谁的角度来写？

**Jamie：**当然是房主，还会有谁？

**Vinod：**还有系统管理员这个角色。即使是房主担任这个角色，这也是一个不同的视角。“管理员”启动系统，配置零件，

布置平面图，安置摄像机……

**Jamie：**当房主想看视频时，我只是让她扮演房主的角色。

**Vinod：**好的，这只是监视功能界面主要行为之一。但是，我们也应该调查一下系统管理员的行为。

**Jamie（有些不满）：**你是对的。

（Jamie 离开去找销售人员。几个小时以后她回来了。）

**Jamie：**我真走运，找到了市场部联系人，我们一起完成了系统管理员的用例。我们应该把“管理”定义为可以应用所有其他



SafeHome 功能的一个功能。这是我们提出的用例。

(Jamie 给 Vinod 看这个非正式的用例。)

非正式用例：我想能够在任何时候设置和编辑系统的布置方案。当我启动系统时，我选择某个管理功能。系统询问我是否要建立一个新的系统布置方案，或者询问我是否编辑已有的方案。如果我选择了一个新建方案，系统呈现一个绘画屏幕，在网格上可以画出建筑平面图来。为了绘画简便，应该提供墙壁、窗户和门的图标。我只是将图标伸展到合适的长度。系统将把长度显示为英尺或者米（我可以选择度量系统）。我能够从传感器和摄像机库中进行选择，并且将它们放置在平面图中。我

标记每个传感器和摄像机，或者系统自动进行标记。我可以通过合适的菜单对传感器和摄像机进行设置。如果选择编辑，就可以移动传感器和摄像机，添加新的或删除已有的传感器和摄像机，编辑平面图并编辑摄像机和传感器的设置。在每种情形下，我希望系统能够进行一致性检查并且帮助我避免出错。

Vinod (看完脚本之后)：好的，对于绘画程序，可能有一些有用的设计模式（第 16 章）或可复用的图形用户界面构件。我打赌，通过使用可复用构件，我们可以实现某些或大部分管理员界面。

Jamie：同意！我马上进行查看。

**任务细化。**在第 12 章中，我们讨论了逐步求精（也称为功能分解或者逐步细化），把它作为一种细化处理任务的机制，而这些任务是软件完成某些期望功能所要求的。界面设计的任务分析采用了一种详细阐述的办法来辅助理解用户界面必须采纳的用户活动。

327

首先，工程师必须定义完成系统或应用程序目标所需的任务并对任务进行划分。例如，考虑前面讨论的为室内设计师开发的计算机辅助设计系统。通过观察工作中的室内设计师，软件工程师了解到，室内设计由一系列的主要活动组成：家具布置（在前面用例设计中提到过）、结构和材料的选择、墙壁和窗户装饰物的选择、（向客户）展示、计算成本、购物。其中任何一个都可以被细化成一系列的子任务。例如，使用用例中的信息，可以将家具布置任务细化为下面的子任务：（1）根据房屋的尺寸画出平面图；（2）将门窗安置在合适的位置；（3a）使用家具模型在平面图上描绘相应比例的家具轮廓；（3b）使用饰件模板在平面设计图上勾勒相应比例的饰件；（4）移动家具和饰件轮廓线到达理想的位置；（5）标记所有的家具和饰件轮廓；（6）标出尺寸以显示其位置；（7）为用户勾画透视图。也可以应用类似的方法对其他主任务进行细化。

上面的每个子任务都可以进一步细化。其中子任务 1～6 可以通过在界面中操纵信息和执行各种动作来完成。另一方面，子任务 7 可以在软件中自动完成，并且几乎不用直接与用户交互<sup>①</sup>。界面的设计模型应该以一种与用户模型（典型室内设计师的轮廓图）和系统感觉（室内设计师期望系统自动提供）相一致的方式来配合这些任务。

**对象细化。**软件工程师这时不是着眼于用户必须完成的任務，而是需要检查用例和来自用户的其他信息，并且提取室内设计师需要使用的物理对象。这些对象可以分为不同的类。需要定义每个类的属性，并且通过对

**建议** 尽管对象细化十分有用，但它应当作为独立的方法去使用。任务分析的过程中，应当考虑用户的聲音。

① 然而，事实可能不是这样。室内设计师可能想要指定所画的透视图、缩放比例、色彩的运用和其他信息。与透视渲染相关的用例将提供解决这些问题的信息。

每个对象动作的评估为设计师提供一个操作列表。例如，家具模板可能被转换成一个名为 Furniture 的类，这个类包括 size、shape 和 location 等属性。室内设计师会从 Furniture 类中选择对象，将其移动到平面图（在此处，平面图是另一个对象）中的某个位置上，拖曳家具的轮廓，依此类推。任务选择（select）、移动（move）、拖曳（draw）等都是操作。用户界面分析模型不能对任何一种操作都提供文字实现。然而，随着设计的不断细化，对每个操作的细节都会进行定义。

**工作流分析。**当大量扮演着不同角色的用户使用某个用户界面时，有时候除了任务分析和对象细化之外，还有必要进行工作流分析。该技术使得软件工程师可以很好地理解在涉及多个成员（角色）时，工作过程是如何完成的。假设某个公司打算将处方药的开方和给药过程全部自动化。全部过程<sup>①</sup>将围绕着一个 WebApp 进行考虑，医生（或者他们的助手）、药剂师和病人等都可以访问这个应用系统。用 UML 泳道图（活动图的一种变形）能够有效地表示工作流。

328

下面只考虑工作过程中的一小部分：当病人请求重填处方时发生的情形。图 15-2 给出了一个泳道图，该图表明了前面提及的三个角色的任务和决定。这些信息可以通过访谈或每个角色书写的用例获取。不管怎样，事件流（图中显示的）使得界面设计师认识到三个关键的界面特征：

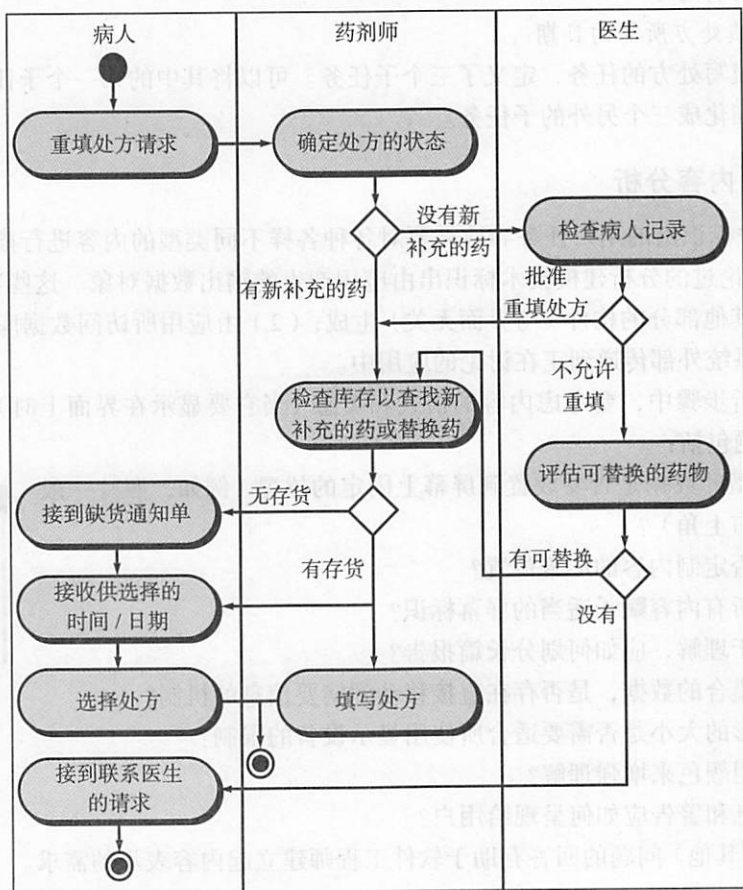


图 15-2 处方重填功能的泳道图

① 这个例子选自 [Hac98]。

1. 每个用户通过界面实现不同的任务，因此，为病人设计的界面在感觉上上与为药剂师或医生设计的界面有所不同。
2. 为医生和药剂师设计的界面应该能够访问和显示来自辅助信息源的信息（例如，药剂师应能够访问库存详细清单，而医生应能够访问其他可选药物信息）。
3. 泳道图中的很多活动都可以采用任务分析和对象求精使其进一步细化（例如，“填写处方”隐含着邮购支付、访问药房，或者访问特殊药品分发中心）。

**层次表示。**在界面分析时，会产生相应的细化过程。一旦建立了工作流，就为每个用户类型都定义一个任务层次。该任务层次来自于为用户定义的每项任务的逐步细化。例如，考虑重填处方的用户任务请求，任务层次如下：

重填处方请求

- 提供辨识信息
  - 指定姓名
  - 指定用户 ID
  - 指定个人身份识别号码 (PIN) 和密码
- 指定处方序号
- 指定重填处方所需的日期

为了完成填写处方的任务，定义了三个子任务。可以将其中的第一个子任务“提供辨识信息”进一步细化成三个另外的子任务。

**引述** 使技术适应用户要比用户适应技术好。

Larry Marine

329  
1  
330

### 15.3.3 显示内容分析

15.3.2 节中标识出的用户任务导致需要对各种各样不同类型的内容进行描述。在第 9 章和第 11 章中讨论过的分析建模技术标识出由应用产生的输出数据对象。这些数据对象可能：（1）由应用中其他部分的构件（与界面无关）生成；（2）由应用所访问数据库中存储的数据获得；（3）从系统外部传递到正在讨论的应用中。

在界面分析步骤中，要考虑内容的格式和美感（当它要显示在界面上时）。其中需要提问和回答的问题包括：

- 不同类型的数据是否要放置到屏幕上固定的位置（例如，照片一般显示在右上角）？
- 用户能否定制内容的屏幕位置？
- 是否对所有内容赋予适当的屏幕标识？
- 为了便于理解，应如何划分长篇报告？
- 对于大集合的数据，是否存在直接移动到摘要信息的机制？
- 输出图形的大小是否需要适合所使用显示设备的限制？
- 如何使用颜色来增强理解？
- 出错信息和警告应如何呈现给用户？

对这些（和其他）问题的回答有助于软件工程师建立起内容表示的需求。

**提问** 作为用户界面设计的一部分，我们如何决定所显示内容的格式和美感？

### 15.3.4 工作环境分析

Hackos 和 Redish[Hac98] 在讨论工作环境分析的重要性时这样写道：“人们不能孤立地

完成任务。他们会受到周围活动的影响,如工作场所的物理特征,使用设备的类型,与其他人的工作关系等。”在某些应用中,计算机系统的用户界面被放在“用户友好”的位置(例如,合适的亮度、良好的显示高度、简单方便的键盘操作),但有些地方(例如,工厂的地板和飞机座舱)亮度可能不是很适合,噪音也可能是个问题,也许不能选择使用键盘、鼠标或触摸屏,显示方位也不甚理想。界面设计师可能会受到某些因素的限制,这些因素会减弱易用性。

除了物理的环境因素之外,工作场所的文化氛围也起着作用。可否采用某种方式(例如,每次交互所用时间、交互的准确性)来度量系统的交互?在提供一个输入前,两个或多个人员是否一定要共享信息?如何为系统用户提供支持?在界面设计开始之前,应该对上述问题和更多的相关问题给予回答。

[331]

## 15.4 界面设计步骤

一旦完成了界面分析,最终用户要求的所有任务(对象和动作)都已经被详细确定下来,界面设计活动就开始了。与所有的软件工程设计一样,界面设计是一个迭代的过程。每个用户界面设计步骤都要进行很多次,每次精细化的信息都来源于前面的步骤。

**引述** 交互设计是图形艺术、技术和心理学的无缝结合。

Brad Wieners

尽管已经提出了很多不同的用户界面设计模型(例如[Nor86]和[Nie00]),但它们都建议结合以下步骤:(1)定义界面对象和动作(操作);(2)确定事件(用户动作),即会导致用户界面状态发生变化的事件;(3)描述每个状态的表示形式;(4)说明用户如何利用界面提供的信息来解释每个状态。

### 15.4.1 应用界面设计步骤

界面设计的一个重要步骤是定义界面对象和作用于对象上的动作。为了完成这个目标,需要使用类似于第9章介绍的方法来分析用户场景,也就是说,撰写用例的描述。名词(对象)和动词(动作)被分离出来形成对象和动作列表。

一旦完成了对象和动作的定义及迭代细化,就可以将它们按类型分类。目标、源和应用对象都被标识出来。将源对象(如报告图标)拖放到目标对象(如打印机图标)上,这意味着该动作要产生一个硬拷贝的报告。应用对象代表应用中特有的数据,它们并不作为屏幕交互的一部分被直接操纵。例如,邮件列表被用于存放邮件的名字,该列表本身可以进行排序、合并或删除(基于菜单的动作),但是,它不会通过用户的交互被拖动和删除。

当设计者满意地认为已经定义了所有的重要对象和动作(对一次设计迭代而言)时,便可以开始进行屏幕布局。与其他界面设计活动一样,屏幕布局是一个交互过程,其中包括:图标的图形设计和放置、屏幕描述性文字的定义、窗口的规格说明和标题,以及各类主要和次要菜单项的定义等。如果一个真实世界的隐喻适合于该应用,则在此时进行说明,并以补充隐喻的方式来组织布局。

为了对上面的设计步骤提供简明的例证,我们考虑 SafeHome 系统(在前面几章讨论过)的一个用户场景。下面是界面的初步用例(由房主写的)描述。

**初步用例:**我希望通过 Internet 在任意的远程位置都能够访问 SafeHome 系统。使用运行在笔记本上的浏览器软件(当正处于工作或者旅行状态时),我可以决定报警系统的状态、启动或关闭系统、重新配置安全区以及通过预先安置的摄像机观察住宅内的不同房间。

[332]

为了远程访问 SafeHome，我需提供标识符和密码，这些定义了访问的级别（如并非所有用户都可以重新配置系统）并提供安全保证。一旦确认了身份，我就可以检查系统状态，并通过启动或关闭 SafeHome 系统改变状态。通过显示住宅的平面图，观察每个安全传感器，显示每个当前配置区域以及修改区域（必要时），可以重新配置系统。通过有策略地放置摄像机以观察房子内部。通过对每个摄像机进行摇动和变焦以提供房子内部的不同视角。

基于这个用例，确定房主的任务、对象和数据项如下：

- 访问 SafeHome 系统。
- 输入 ID 和密码实现远程访问。
- 检查系统状态。
- 启动或关闭 SafeHome 系统。
- 显示平面图和传感器位置。
- 显示平面图上的区域。
- 改变平面图上的区域。
- 显示建筑平面图上的视频摄像机位置。
- 选择用于观察的视频摄像机。
- 观察视频图像（每秒 4 帧）。
- 摇动或变焦摄像机。

从房主的这个任务清单中抽取出对象和动作。所提到的大部分对象都是应用对象。然而，视频摄像机位置（源对象）被拖放到视频摄像机（目标对象）以创建视频图像（视频显示的窗口）。

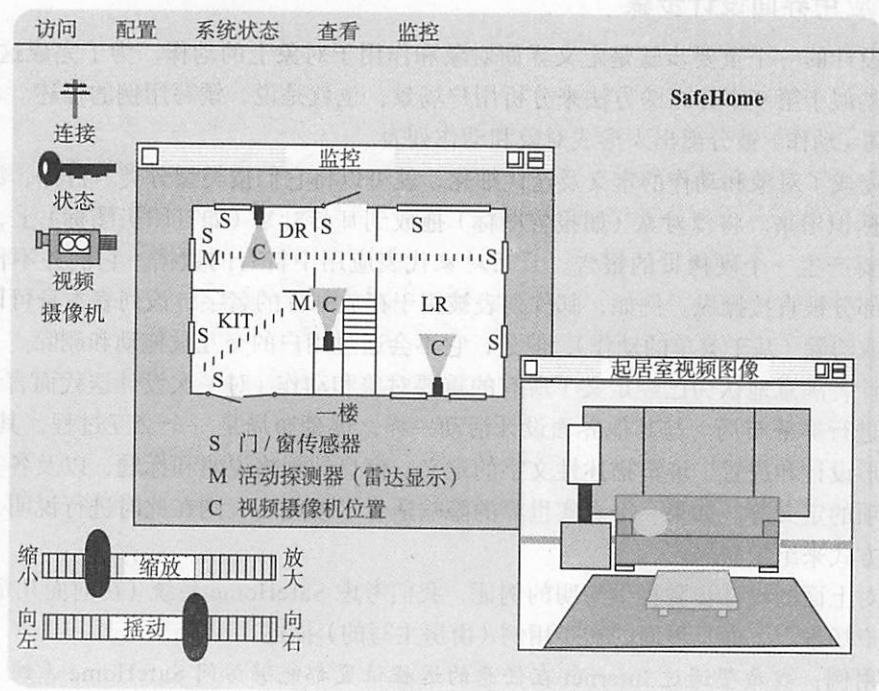


图 15-3 基本的屏幕布局

为视频监控设计的屏幕布局初步草图如图 15-3 所示<sup>①</sup>。为了调用视频图像,需选择显示在监控窗口中的建筑平面图上的视频摄像机位置图标 C。在这种情况下,起居室(LR)中的摄像机位置被拖放到屏幕左上部分的视频摄像机图标处,此时,视频图像窗口出现,显示来自位于起居室中的摄像机的流视频。变焦和摇动控制条用于控制视频图像的放大和方向。为了选择来自另一个摄像机的图像,用户只需简单地将另一个不同的摄像机位置图标拖放到屏幕左上区域的摄像机图标上即可。

所显示的布局草图需以菜单条上每个菜单项的扩展来补充,指明视频监控模式(状态)有哪些可用的动作。在界面设计过程中,将创建用户场景中提到的房主的每个任务的一组完整草图。

**建议** 尽管自动化的工具在开发布局原型中十分有用,但有时候铅笔和纸张也是需要的。

### 15.4.2 用户界面设计模式

图形用户界面已经变得如此普遍,以至于涌现出各式各样的用户界面设计模式。设计模式是一种抽象,描述了特定的、界限明确的设计问题的解决方案。

作为通常碰到的界面设计问题的一个例子,考虑用户必须一次或多次输入日历日期这种情况,有时候需要提前输入月份。对于这个简单的问题,有很多可能的解决方案,为此也提出了很多种不同的模式。Laakso[Laa00]提出了一种称为 CalendarStrip 的模式,此模式生成一个连续、滚动的日历,在这个日历上,当前日期被高亮度显示,未来的日期可以在日历上选择。这个日历隐喻在用户中具有很高的知名度,并提供了一种有效的机制,可以在上下文环境中设置未来的日期。

在过去的十年间,人们已经提出了很多用户界面设计模式。在第 16 章中有关于用户界面设计模式的更为详尽的论述。此外,Ericksen[Eri08]提供了许多基于 Web 的文献资料。

### 15.4.3 设计问题

在进行用户界面设计时,几乎总会遇到以下四个问题:系统响应时间、用户帮助设施、错误信息处理和命令标记。不幸的是,许多设计人员往往很晚才注意到这些问题(有时在操作原型已经建立起来后才发现问题),这往往会导致不必要的反复、项目拖延及用户的挫折感,最好的办法是在设计的初期就将这些作为设计问题加以考虑,因为此时修改比较容易,代价也低。

**响应时间。**系统响应时间包括两个重要的属性:时间长度和可变性。如果系统响应时间过长,用户就会感到焦虑和沮丧。系统时间的可变性是指相对于平均响应时间的偏差,在很多情况下这是最重要的响应时间特性。即使响应时间比较长,响应时间的低可变性也有助于用户建立稳定的交互节奏。例如,稳定在 1 秒的命令响应时间比从 0.1 秒到 2.5 秒不定的响应时间要好。在可变性到达一定值时,用户往往比较敏感,他们总是关心界面背后是否发生了异常。

**帮助设施。**几乎所有计算机交互式系统的用户都时常需要帮助。现代

**网络资源** 已经有大量的用户界面设计模式,访问 <http://www.hci-patterns.org/patterns/borchers/patternindex.html> 可以找到它们的站点链接。

**引述** 当试图设计一些十分简单的东西时,人们经常犯的共性错误就是低估了笨人的智慧。

Douglas Adams

① 注意这里的实现与前几章讲到的这些特性的实现有所不同。这里应该是第一次设计的草图,可以考虑提供备选的设计草图。



的软件均提供联机帮助, 用户可以不离开用户界面就解决问题。

**错误处理。**通常, 交互式系统给出的出错消息和警告应具备以下特征: (1) 以用户可以理解的语言描述问题; (2) 应提供如何从错误中恢复的建设性意见; (3) 应指出错误可能导致哪些不良后果 (比如破坏数据文件), 以使用户检查是否出现了这些情况 (或者在已经出现的情况下进行改正); 应伴随着视觉或听觉上的提示, 并且永远不应该把错误归咎于用户。

**菜单和命令标记。**键入命令曾经是用户和系统交互的主要方式, 并广泛用于各种应用。现在, 面向窗口的界面采用点击 (point) 和选取 (pick) 方式, 减少了用户对键入命令的依赖。但许多高级用户仍然喜欢面向命令的交互方式。在提供命令或菜单标签交互方式时, 必须考虑以下问题:

- 每个菜单选项是否都有对应的命令?
- 以何种方式提供命令? 有三种选择: 控制序列 (如 Alt+P)、功能键或键入命令。
- 学习和记忆命令的难度有多大? 命令忘了怎么办?
- 用户是否可以定制和缩写命令?
- 在界面环境中菜单标签是否是自解释的?
- 子菜单是否与主菜单项所指功能相一致?
- 有适合于应用系列内部的命令使用约定吗?

**应用的可访问性。**随着计算型应用变得无处不在, 软件工程师必须确保界面设计中包含使得有特殊要求的用户易于访问的机制。对于那些实际上面临挑战的用户 (和软件工程师) 来说, 由于道义、法律和业务等方面的原因, 可访问性是必需的。有多种可访问性指导方针 (如 [W3C03])——很多都是为 WebApp 设计的, 但这些方针经常也能应用于所有软件——为设计界面提供了详细的建议, 以使界面能够达到各种级别的可访问性。其他指南 (如 [App13]、[Mic13]) 对于“辅助技术”提供了专门的指导, 这些技术用来解决那些在视觉、听觉、活动性、语音和学习等方面有障碍的人员的需要。

**国际化。**软件工程师和他们的经理往往会低估建立一个适应不同国家和不同语言需要的用户界面所应付出的努力和技能。用户界面经常是为一个国家和一种语言所设计的, 在面对其他国家时只好应急对付。设计师面临的挑战就是设计出“全球化”的软件。也就是说, 用户界面应该被设计成能够容纳需要交付给所有软件用户的核心功能。本地化特征使得界面能够针对特定的市场进行定制。

软件工程师有多种国际化指导方针 (如 [IBM13]) 可以使用。这些方针解决了宽度设计问题 (例如, 在不同的市场情况下屏幕布局可能是不同的), 以及离散实现问题 (例如, 不同的字母表可能生成特定的标识和间距需求)。对于几十种具有成百上千字母和字符的自然语言的管理, 已经提出的 Unicode 标准 [Uni03] 就是用来解决这个挑战性问题的。

**引述** 来自地狱的界面——修正这个错误并且继续进行, 请输入任一个 11 位的素数……

作者不详

**网络资源** 开发可访问软件的指导原则可以在 <http://www-03.ibm.com/able/guidelines/software/access-software.html> 找到。

## 软件工具 用户界面开发

**[目标]** 用户界面开发工具使得软件工程师只需做有限的定制开发就可以建立复杂的

图形用户界面。这些工具提供了对可复用构件的访问, 并且通过选择工具上预定义

的功能就可以建立用户界面。

**[机制]** 现代用户界面由一组可复用的构件组成, 这些构件与一些提供特殊特性的定制构件相结合。大多数用户界面的开发工具能够通过使用“拖放”功能来完成界面的设计。换句话说, 开发人员选择预定义的功能(例如, 表格构造器、交互机制、命令处理), 并将这些功能放置在所创建界面的环境中。

**[代表性工具]**<sup>①</sup>

- LegaSuite GUI。由 Seagull Software (<http://www-304.ibm.com/partnerworld/gsd/solutiondetails.do?solution=1020&expand=true&lc=en>) 开发, 能够创建基于浏览器的

的图形用户界面(GUI)并且提供了对过时界面的再造功能。

- Motif Common Desktop Environment。由 Open Group ([www.osf.org/tech/desktop/cde/](http://www.osf.org/tech/desktop/cde/)) 开发, 是一个集成的图形用户界面, 用于开放系统桌面计算。它对数据、文件(图形化桌面)和应用系统的管理提供了单一的、标准的图形化界面。
- Alita Design 8.0。由 Altia ([www.altia.com](http://www.altia.com)) 开发, 是一种可以在多种平台(例如, 自动的、手持的、工业的)上创建图形用户界面(GUI)的工具。

## 15.5 WebApp 和移动 App 的界面设计

无论是为 WebApp、移动设备、传统的软件应用、消费产品设计的用户界面, 还是为工业设备设计的用户界面, 都应该展示出本章前面所讲的特性。Dix[Dix99] 认为 Web 工程师设计的 WebApp 和移动界面必须能够回答用户三个主要问题: 我在哪里? 我现在能做什么? 我去过哪里? 我能够去哪里? 这些问题的答案使用户理解交互环境并且使应用更为有效。

**引述** 如果一个站点非常好用, 但却缺少美观、合适的设计风格, 同样会失败。

Curt Cloninger

### 15.5.1 界面设计原则与指导方针

WebApp 或移动 App 的用户界面是它的“第一印象”。不管它的内容、处理能力、服务以及应用自身的整体效益如何, 一个设计糟糕的用户界面将会使潜在的用户失去信心。事实上, 用户甚至可能转向使用别的应用, 因为几乎在每个主题领域内, WebApp 和移动 App 的竞争都是十分激烈的, 用户界面应当迅速“抓住”潜在用户。

当然, WebApp 和移动 App 之间具有重要的差异。由于小型移动设备(如智能手机)的物理限制, 移动界面设计师必须以集中的方式来压缩交互。然而, 本节讨论的基本原则仍然适用。

Bruce Tognozzi [Tog01] 定义了一组可用性更高的基本的设计原则。<sup>②</sup>

**预测。**应用应当能够预测出用户的下一个动作。例如, 假设用户已经请求了一个内容对象, 此对象显示出针对最新版本操作系统的打印机驱动程序信息。WebApp 的设计者应该预测出用户可能会请求下载该驱动程序, 并且直接提供下载的导航辅助。

**传达。**界面应该能够传达由用户启动的任何活动的状态。传达可以是直接的(例如一条

**提问** 在设计图形用户界面时是否有须遵循的一些基本原则?

[337]

① 这里提到的工具只是此类工具的例子, 并不代表本书支持采用这些工具

② 本书对 Tognozzi 的最初准则进行了修改与扩展。这些原则的进一步讨论参见 [Tog01]。

文本消息),也可以是间接的(例如在打印机中移动的纸张表明打印机正在工作)。

**一致。**导航控制、菜单、图标和美学风格(例如颜色、形状和布局)的使用应该在整个应用系统中保持一致。例如,如果一个移动 App 在显示屏的底部使用一组四个图标(用来表示主要功能),这些图标应该出现在每个屏幕上,并且不应移动到顶部来显示。图标的含义在应用范围内应该是不言而喻的。

**自律。**界面应该辅助用户在整个应用中移动,但也应该坚持使用已经为应用建立起来的导航习惯,以这样的方式来辅助用户。例如,对内容的导航应该受到用户 ID 和密码的访问控制,而不应该提供能使用户改变这种控制的导航机制。

**效率。**应用的设计和界面应该优化用户的工作效率,而不是优化设计与构建 WebApp 的 Web 工程师的效率,也不是优化运行系统的客户/服务器环境的效率。Tognozzi[Tog01]在讨论这一问题时写道:“这个简单的事实就是,为什么对于参与软件项目的每个人来说,认识到提高用户生产率的重要性及理解开发有效率的应用和提高用户效率的根本区别是非常重要的。”

**引述** 最好的旅程应该有最少的步骤,能够缩短用户和他们要到达的目标之间的距离。

作者不详

**338** **灵活性。**界面应该足够灵活,既能够使其中一些用户直接完成任务,

也能够使另一些用户以一种比较随意的方式浏览应用。在每种情况下,界面能够使用户认识到自己在哪里,并且给用户提供了撤销错误及从选错的导航路径返回的功能。

**关注点。**界面(及界面表示的内容)应该关注用户正在完成的任务。这个概念对移动 App 来说格外重要,如果设计师试图做得太多会使得界面变得非常杂乱。

**人机界面对象。**对于 WebApp 和移动 App,已经开发了大量可复用的人机界面对象库。使用这些对象库。能被最终用户“看到的、听到的、接触到的以及用别的方式感知到的”[Tog01]任何界面对象都能从大量对象库的任何一个中获得。

**缩短等待时间。**应用不应该让用户等待内部操作的完成(例如,下载一个复杂的图形图像),而应该利用多任务处理方式,从而使用户继续他的处理工作,看起来就像前面的操作已经完成一样。除了减少等待时间,如果有延迟事件发生,则必须通知用户,从而使用户知道正在发生的事情,包括:(1)在选中选项后,如果应用没有立即做出响应,则应该提供声音反馈;(2)显示一个动态时钟或进度条表示处理工作正在进行中;(3)当处理过程很长时,提供娱乐活动(例如动画或文本演示)。

**学习能力。**应用应将学习时间减到最少,并且一旦已经学习过了,当再次访问此应用时,将所需要的再学习时间减到最少。一般而言,界面应该侧重于简单、直观的设计,将内容和功能分类组织,这样对于用户来说很直观。

**隐喻。**只要隐喻适合应用和用户,使用交互隐喻的界面就更容易学习和使用。隐喻应该采用用户熟悉的图片和概念,但是并不要求是现实生活的精确再现。

**易读性。**界面展示的所有信息对于老人和年轻人都应该是易读的。界面设计者应该着重选择易读的字型式样、字体大小以及可以增强对比效果的背景颜色。

**跟踪状态。**在合适的时候,应该跟踪和保存用户状态,使得用户能够退出系统,稍后返回系统时又能回到退出的地方。一般而言,可设计 cookies 来存储状态信息。然而,cookies 是一种备受争议的技术,别的设计方案也许对某些

**建议** 隐喻是一种出色的想法,因为隐喻能够反映现实世界的经验。只是要确保你选择的隐喻是最终用户所熟悉的。

用户来说更合适。

可见的导航。设计合理的界面提供了这样的设想,“用户待在同一个地方,工作被带到他们面前”[Tog01]。使用这种方法后,导航就不再是用户关心的事情了,用户检索内容对象,并选择功能,这些功能都是通过界面显示并执行的。

339

## SafeHome 界面设计评审

[场景] Doug Miller 的办公室。

[人物] Doug Miller, SafeHome 软件工程技术经理; Vinod Raman, SafeHome 产品软件工程技术成员。

[对话]

Doug: Vinod, 你和你的团队是否有可能评审 SafeHomeAssured.com 电子商务的界面原型?

Vinod: 是的……我们所有人都从技术角度对它进行了仔细检查,而且我还做了一些记录。

昨天我将这些记录发给了 Sharon (SafeHome 电子商务网站外包供应商 Web 工程团队的经理)。

Doug: 你和 Sharon 可以在一起详细讨论一下……给我一份重要问题的总结。

Vinod: 总的来说,他们已经做得很好了,没有遇到什么阻力。但是,这是一个典型的电子商务界面,具有高雅的美学设计、合理的布局设计。他们已经完成了所有重要功能……

Doug (可怜地微笑): 但是?

Vinod: 是的,有些小问题。

Doug: 例如……

Vinod (给 Doug 看界面原型的序列情节故事板): 这是一些显示在主页上的主要功能菜单。

学习 SafeHome

描述你的住宅

获得 SafeHome 构件建议

购买 SafeHome 系统

获得技术支持

问题并不在于这些功能,它们都没有问题,但是抽象级别不太合适。

Doug: 它们是主要功能,对吗?

Vinod: 没错。但是有这样一个问题,你可以通过输入构件列表来购买系统,如果你不想描述房子,就没有必要描述。我建议在主页上创建 4 个菜单选项:

学习 SafeHome

确定你所需要的 SafeHome 系统

购买 SafeHome 系统

获得技术支持

当你选择了“确定你所需要的 SafeHome 系统”时,你会有下面的选项:

选择 SafeHome 构件

获得 SafeHome 构件建议

如果你是一个有经验的用户,那么将从一组分好类的下拉菜单中选择构件,包括传感器、摄像机、控制面板等。如果需要帮助,可以请求系统提供建议,那时系统需要你描述一下你的房间。我认为这样更合理。

Doug: 我同意。关于这个问题你和 Sharon 谈过了吗?

Vinod: 没有,我想先和市场部讨论一下,然后我会给她打电话。

Nielsen 和 Wagner[Nie96] 提出了一些实际可行的界面设计指导原则(基于他们对重要 WebApp 的重新设计),这些原则很好地补充了本节前面提出的准则:

- 不要迫使用户阅读大量的文本信息,特别是当文本的内容是解释 WebApp 的操作,

或者是辅助导航时。

- 除了不可避免的操作外，不要让用户进行滚动操作。
- 设计不应该依赖于浏览器的功能来辅助导航。
- 美学效果绝不应该取代功能性。
- 不要强迫用户搜索显示如何链接到其他内容或服务。

340

好的界面设计能够提高用户对网站提供的内容或服务的理解程度，它并不一定要有闪烁的动画，但总应该是结构良好的，且具有符合人机工程学的声音。

**引述** 人们对于那些设计糟糕的WWW站点几乎没有耐心。

Jakob Nielsen,  
Annette Wagner

### 15.5.2 WebApp 和移动 App 的界面设计工作流<sup>①</sup>

本章前面曾经提到用户界面设计首先要确定用户、任务和环境需求。一旦确定了用户任务，就可以创建和分析用户场景（用例），并定义一组界面对象和活动。

需求模型包含的信息构成了创建屏幕布局的基础，屏幕布局描述图标图形设计和位置、描述性屏幕文本的定义、窗口标题定义及规格说明、主菜单和子菜单项目的规格说明。接着使用工具创建原型，并最终实现用户界面模型。下面的任务代表了一个基本工作流程：

1. 对需求模型中的信息进行评审，并根据需要进行优化。
2. 开发 WebApp 界面布局的草图。如果界面布局已经存在（在需求建模时开发了一个原型），则应该根据需要进行评审和优化。
3. 将用户目标映射到特定的界面行为。对于大多数 WebApp 和移动 App 来说，用户的主要目标相对比较少。应该将这些目标映射到特定的界面行为，如图 15-4 所示。实际上，界面设计人员必须回答下面的问题：“界面是如何让用户完成每个目标的？”

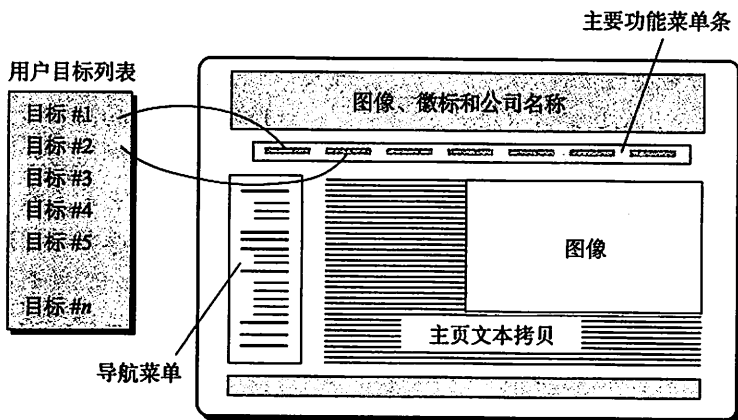


图 15-4 将用户目标映射到特定的界面行为

4. 定义与每个行为相关的一组用户任务。每个界面行为（例如购买商品）与一组用户任务相联系。在分析建模的过程中已经确定了这些任务。在设计期间，它们必须与明确的交互建立对应关系，这些交互包括导航事件、内容对象和应用功能。
5. 为每个界面行为设计情节故事板映像。在考虑每种行为时，应该创建序列情节故事板映像（屏像），来描述界面是怎样响应用户的交互行为的。应该明确内容对象（即使它们还没有设计和开发），并确定导航链接。

341

① 在第 17 章和 18 章将更详细地讨论 WebApp 和移动 App。

6. 利用美学设计中的输入来细化界面布局和情节故事板。在大多数情况下, 粗略的布局和情节故事板是由 Web 工程师完成的, 但是重要商业网站的美学外观通常是由专业绘图师而不是技术专家完成的。
  7. 明确实现界面功能的界面对象。这一任务可能需要在现有对象库中搜索, 找到那些适合界面的可复用对象(类)。另外, 在此时定义任何需要的自定义类。
  8. 开发用户与界面交互的过程表示。这一可选的任务利用 UML 顺序图或活动图(附录 1)描述用户与 WebApp 交互时的活动流(和决策流)。
  9. 开发界面的行为表示法。这一可选的任务利用 UML 的状态图(附录 1)表示状态转换和引起状态转换的事件, 并定义控制机制(即通过用户可用的对象和行为改变一个应用系统的状态)。
  10. 描述每种状态的界面布局。利用在任务 2 和任务 5 中开发的设计信息, 把确定的布局和屏幕图像与任务 8 中描述的每个 WebApp 状态联系起来。
  11. 优化和评审界面设计模型。界面的评审应该以可用性为重点。
- 值得注意的是, Web 工程团队所选择的最终任务集必须适合待构建应用的特殊需求。

## 15.6 设计评估

一旦建立好可操作的用户界面原型, 必须对其进行评估, 以确定满足用户的需求。评估可以从非正式的“测试驱动”(比如用户可以临时提供一些反馈)到正式的设计研究(比如向一定数量的最终用户发放评估问题表, 采用统计学的方法进行评估)。

[342]

用户界面评估的循环如图 15-5 所示。完成设计模型后就开始建立第一级原型; 用户对该原型进行评估<sup>①</sup>, 直接向设计者提供有关界面功效的建议, 如采用正式的评估技术(比如使用提问单、分级评分表), 这样设计者就能从调查结果中得到需要的信息(比如 80% 的用户不喜欢其中保存数据文件的机制); 针对用户的意见对设计进行修改, 完成下一级原型。评估过程不断进行下去, 直到不需要再修改为止。

原型开发方法是有效的, 但是否可以在建立原型以前就对用户界面的质量进行评估呢<sup>②</sup>? 如果能够及早地发现和改正潜在的问题, 就可以减少评估循环执行的次数, 从而缩短开发时间。界面设计模型完成以后, 就可以运用下面的一系列评估标准 [Mor81] 对设计进行早期评审:

1. 系统及其界面的需求模型或书面规格说明的长度和复杂性在一定程度上体现了用户学习系统的难度。
2. 指定用户任务的个数以及每个任务动作的平均数在一定程度上体现了系统的交互时间和系统的总体效率。
3. 设计模型中动作、任务和系统状态的数量体现了用户学习系统时所记忆内容的

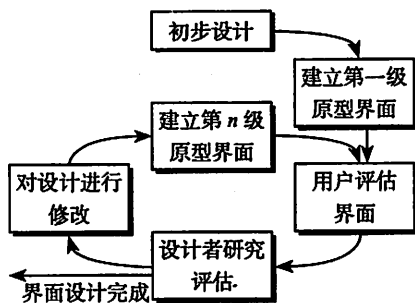


图 15-5 界面设计评估循环

① 注意, 人类工程学和界面设计方面的专家也可对界面进行审查。这些审查叫作启发评估或认知走查。

② 一些软件工程师更倾向于开发一个简单描画设计的用户界面模型, 称之为纸上原型, 使相关人员在交付任何程序资源之前先测试验证 UI 的内容。具体过程参见 [http://www.paperprototyping.com/what\\_examples.html](http://www.paperprototyping.com/what_examples.html)。



343

多少。

4. 界面风格、帮助设施和错误处理协议在一定程度上体现了界面的复杂度和用户的接受程度。

一旦第一个原型完成以后,设计者就可以收集到一些定性和定量的数据以帮助进行界面评估。为了收集定性的数据,可以进行问卷调查,使用户能够评估界面原型。如果需要得到定量数据,就必须进行某种形式的定期研究分析。观察用户与界面的交互,记录以下数据:在标准时间间隔内正确完成任务的数量、使用动作的频率、动作顺序、观看屏幕的时间、出错的数目、错误的类型、错误恢复时间、使用帮助的时间、标准时间段内查看帮助的次数。这些数据可以用于指导界面修改。

有关用户界面评估方法的详细论述已超出了本书的范围,有兴趣的读者可以参考 [Hac98] 和 [Sto05] 等文献。

## 15.7 小结

用户界面可以说是计算机系统或产品的最重要元素。糟糕的界面设计可能会严重地阻碍用户挖掘系统的计算能力,也会阻碍用户挖掘应用程序的信息内容。事实上,即使应用具有良好设计和可靠实现,糟糕的界面也可能导致该系统失败。

三个重要的原则可用于指导有效的用户界面设计:(1)把控制权交给用户;(2)减少用户的记忆负担;(3)保持界面一致性。为了得到符合这些原则要求的界面,必须实施有组织的设计过程。

用户界面的开发首先从一系列的分析任务开始。用户分析确定了各类最终用户的概况,并且使用了从各种业务和技术资源收集来的信息。任务分析定义了用户任务和行为,其中使用了细化或面向对象的方法、用户用例的应用、任务和对象的细化、 workflows 分析和层级任务表示等,来获得对人机交互的充分理解。环境分析可厘清界面必须操作的物理结构和社会结构。

一旦任务确定下来,就可以通过创建和分析用户场景来定义一组界面对象和动作。这为创建屏幕布局提供了基础,屏幕布局描述了图形的设计和图标的放置、描述性屏幕文字的定义、窗口的规格说明和标题以及主菜单和子菜单项规格说明。诸如响应时间、命令和动作结构、错误处理和帮助设施等设计问题应该在细化设计模型时考虑。很多实现工具可以用于创建供用户评估的原型。

344

像传统软件的界面设计一样,WebApp 和移动 App 界面的设计体现了用户界面的组织结构和屏幕的布局,是对交互模式的定义,也是对导航机制的描述。在设计布局和界面控制机制时,界面设计原则和界面设计工作流为 WebApp 或移动 App 设计者提供了向导。

用户界面是软件的窗口。在很多情况下,界面塑造了用户对系统质量的感知。如果这个“窗口”污点斑斑、凹凸不平或破损不堪,用户也许会选择其他更有效的计算机系统。

## 习题与思考题

- 15.1 描述一下你操作过的最好和最差的系统界面,采用本章介绍的相关概念对其进行评价。
- 15.2 在 15.1.1 节的基础上,再给出两条“把控制权交给用户”的设计原则。
- 15.3 在 15.1.2 节的基础上,再给出两条“减轻用户的记忆负担”的设计原则。
- 15.4 在 15.1.3 节的基础上,再给出两条“保持界面一致”的设计原则。

15.5 考虑下面几个交互应用（或者导师布置的应用）：

- a. 桌面发布系统。
- b. 计算机辅助设计系统。
- c. 室内设计系统（如 15.3.2 节所描述的）。
- d. 大学课程自动注册系统。
- e. 图书管理系统。
- f. 基于网络的公共选举投票系统。
- g. 家庭银行系统。
- h. 导师布置的交互应用。

对上面给出的每个系统，开发用户模型、设计模型、心理模型和实现模型。

15.6 选择习题 15.5 中所列的任何一个系统，使用细化或面向对象的方法进行详细任务分析。

15.7 在 15.3.3 节提供的内容分析列表中至少再添加 5 个问题。

15.8 继续做习题 15.5，为你所选择的应用定义界面对象和动作。确定每个对象类型。

15.9 对于在习题 15.5 中所选的系统，开发一组带有主菜单和子菜单项定义的屏幕布局。

15.10 针对 SafeHome 系统，开发一组带有主菜单和子菜单项的屏幕布局，可以选择一种不同于图 15-3 的方法。

15.11 对于在习题 15.5、习题 15.7 和习题 15.8 中所完成的任务分析设计模型和分析任务，描述你采用的用户帮助设施。

15.12 举例说明为什么反应时间变动是一个问题。

345

15.13 开发一种能自动集成错误消息和用户帮助设施的方法。即系统能自动识别错误类型，并提供帮助窗口，给出改正错误的建议。进行合理且完整的软件设计，其中要考虑到合适的数据结构和算法。

15.14 开发一个界面评估提问单，其中包括 20 个适用于大多数界面的通用问题。由 10 名同学完成你们所有人使用的交互系统的提问单。汇总你们的结果，并在班上做介绍。

## 扩展阅读与信息资源

尽管 Donald Norman 的著作（《The Design of Everyday Things》，reissue edition, Basic Books, 2002）不是专门阐述人机界面的，但其中涉及了进行有效设计的心理学，可以应用于用户界面的设计。对于那些非常关心高质量用户界面设计的人员，我们推荐此读物。Weinschenk 的著作（《100 Things Every Designer Should Know About People》，New Riders, 2011）并不特别侧重于软件，而是富有洞察力地提出了以用户为中心的设计。Johnson 的著作（《Designing with the Mind in Mind》，Morgan Kaufman, 2010）利用认知心理学开发有效的界面设计规则。

图形用户界面在现代计算世界中是无处不在的，无论是在 ATM、移动电话、汽车电子仪表板、Web 站点，或者是在商业应用中，用户界面都为软件提供了窗口。正因如此，关于界面设计的书籍有很多，其中值得借鉴的有：

Ballard, 《Designing the Mobile User Experience》，Wiley, 2007。

Butow, 《User Interface Design for Mere Mortals》，Addison-Wesley, 2007。

Cooper 和他的同事, 《About Face 3: The Essentials of Interaction Design》，3rd ed., Wiley, 2007。

Galitz, 《The Essential Guide to User Interface Design》3rd ed., Wiley, 2007。

Goodwin 和 Cooper, 《Designing for the Digital Age: How to Create Human-Centered Products and Services》，Wiley, 2009。

Hartson 和 Pyla, 《The UX Book: Process and Guidelines For Ensuring a Quality User Experience》, Morgan Kaufman, 2012。

Lehikonen 和他的同事, 《Personal Content Experience : Managing Digital Life in the Mobile Age》, Wiley-Interscience, 2007。

Nielsen, 《Coordinating User Interfaces for Consistency》, Morgan-Kaufmann, 2006。

Pratt 和 Nunes, 《Interactive Design》, Rockport, 2013。

Rogers 和他的同事, 《Interactive Design: Beyond Human-Computer Interaction》, 3rd ed., Wiley, 2011。

Shneiderman 和他的同事, 《Designing the User Interface : Strategies for Effective Human-Computer Interaction》, 5th ed., Addison-Wesley, 2009。

Tidwell, 《Designing Interfaces》, O'Reilly Media, 2nd ed., 2011。

Johnson 的著作 (《GUI Bloopers: Common User Interface Design Don'ts and Do's》, 2nd ed., Morgan Kaufmann, 2007 和 《GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers》, Morgan Kaufmann, 2000) 对那些通过检查反例来实现高效学习的人提供了有用的指导。Cooper 编写的广受欢迎的书 (《The Inmates Are Running the Asylum》, Sams Publishing, 2004) 讨论了为什么高技术产品常令人感到疯狂, 以及如何设计不让人疯狂的产品。

在网上有大量关于用户界面设计的信息, 有关用户界面设计的最新参考文献列表可在 SEPA Web 站点 [www.mhhe.com/pressman](http://www.mhhe.com/pressman) 找到。