第五章 界面设计



中南大学信息学院

内容提要

- 界面设计原则
- 理解用户
- 设计流程
- 任务分析
- 以用户为中心的界面设计

课程目标

- 掌握图形用户界面的主要思想和设计的一般原则。
- 了解用户、用户体验、用户交互分析以及设计流程。
- 掌握任务分析方法方法,重点掌握:使用行为分析、 顺序分析、协作关系分析。
- 掌握以用户为中心的界面设计方法。

5.1界面设计原则

- □ 根据表现形式,用户界面分为
 - 命令行界面
 - 图形界面
 - 多通道用户界面

A CONTROL OF THE CONT

Lecture 4

5.1.1图形用户界面的主要思想

■ 图形用户界面的三个重要思想

- □ 桌面隐喻(desktop metaphor)
- □ 所见即所得 (What You See Is What You Get, WYSIWYG)
- □ 直接操纵(Direct manipulation)

Lecture 4

1.桌面隐喻(desktop metaphor)

- 桌面隐喻是指在用户界面中用人们熟悉的桌面上的图 例清楚地表示计算机可以处理的能力。
 - □ 图形具有一定的文化和语言独立性,可以提高搜索目标的效 率。
 - 图形用户界面中的图例可以代表对象、动作、属性或其他概念。
 - □ 表达方式: 图例和文字
 - 文字适用于表达某些抽象概念
 - 图例更易于识别,占用较少屏幕空间,可独立于语言

Lecture 4

隐喻的表现方法

- ■静态图标
- 动画
- ■视频

```
SCENE COME TO SERVICE STATE OF THE SERVICE STATE OF
```

隐喻的分类

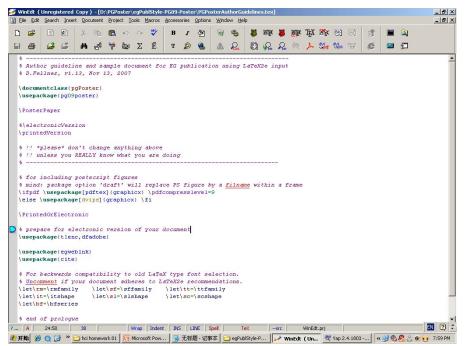
- □ 直接隐喻: 隐喻本身就带有操纵的 对象
 - 如Word中的表格、图表等图标,图标 分别代表了操纵对象。
- □ 工具隐喻: 代表所使用的工具
 - 如用磁盘图标隐喻存盘操作、用打印机 图标隐喻打印操作等,这种隐喻设计简 单、形象直观,应用也最为普遍。
- □ 过程隐喻:通过描述操作的过程来 暗示该操作
 - 如Word中的撤销和恢复图标。

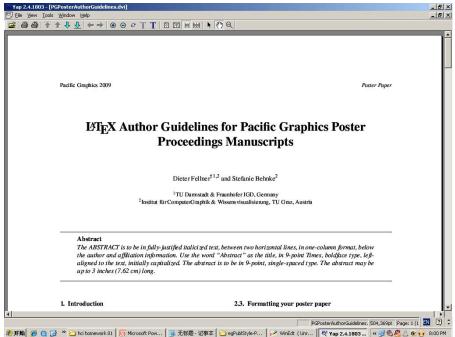




2. 所见即所得(WYSIWYG)

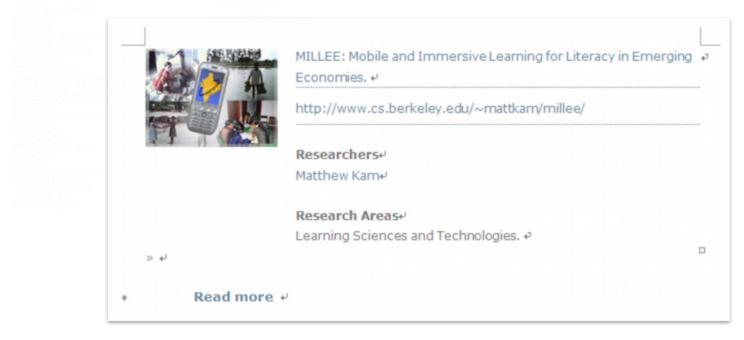
- 在WYSIWYG交互界面中显示的用户交互行为与应用程序 最终产生的结果是一致的。
- 非WYSIWYG的编辑器,用户只能看到文本的控制代码, 对于最后的输出结果缺乏直观的认识。(latex编辑器)





WYSIWYG的弊端

- 如果屏幕的空间或颜色的配置方案与硬件设备所提供的配置不一样,在两者之间就很难产生正确的匹配。
- 文本处理器都提供了定义章、节、小节等的标记,这 些标记显式地标明了对象的属性,但并不是用户最终 输出结果的一部分。



3.直接操纵(Direct manipulation)

- 直接操纵是指可以把操作的对象、属性、关系显式地表示出来,用光笔、鼠标、触摸屏或数据手套等指点设备直接从屏幕上获取形象化命令与数据的过程。
- 直接操纵的对象是命令、数据或是对数据的某种操作。

直接操纵的特性

- 直接操纵的对象是动作或数据的形象隐喻
 - 形象隐喻应该与其实际内容相近,使用户能通过屏幕上的隐喻直接想象 或感知其内容。
- 用指点和选择代替键盘输入
 - □ 优点:操作简便,速度快捷。
- 操作结果立即可见
 - □ 操作结果立即可见,用户可以及时修正操作,逐步往正确的方向前进。
- 支持逆向操作
 - 用户在使用系统的过程中,不可避免地会出现一些操作错误,通过逆向操作,用户可以很方便地恢复到出现错误之前的状态。

直接操纵的优缺点

■ 优点:

借助物理的、空间的或形象的表示,而不是单纯的文字或数字的表示。依赖于视觉和手动控制的参与,可以直接操作,有利于解决问题和进行学习。

缺点:

- □ 不具备命令语言界面的某些优点。
 - 例如从用户界面设计者角度看,设计图形比较繁琐,需进行大量的 测试和实验。
- □表示复杂语义、抽象语义比较困难。

5.1.2图形用户界面一般性原则

□ 界面要具有一致性:

在同一用户界面中,所有的菜单选择、命令输入、数据显示和其他 功能应保持风格的一致性。

□ 常用操作要有快捷方式:

不仅会提高用户的工作效率,还使界面在功能实现上简洁而高效。

□ 提供简单的错误处理:

在出现错误时,系统应该能检测出错误,并且提供简单和容易理解 的错误处理功能

对操作人员的重要操作要有信息反馈:

尤其是对不常用操作、至关重要操作要有信息反馈。

5.1.2图形用户界面一般性原则

□ 操作可逆:

■ 对大多数动作应允许恢复(UNDO),对用户出错采取比较宽容的态度。

□ 设计良好的联机帮助:

人机界面应该提供上下文敏感的求助系统,让用户及时获得帮助,尽量用简短的动词和动词短语提示命令。

□ 合理划分并高效地使用显示屏:

- 只显示与上下文有关的信息,允许用户对可视环境进行维护,如放大、 缩小窗口;用窗口分隔不同种类的信息,只显示有意义的出错信息。
- 保证信息显示方式与数据输入方式的协调一致,尽量减少用户输入的动作,隐藏当前状态下不可用的命令,允许用户自选输入方式,能够删除错误的输入,允许用户控制交互过程。

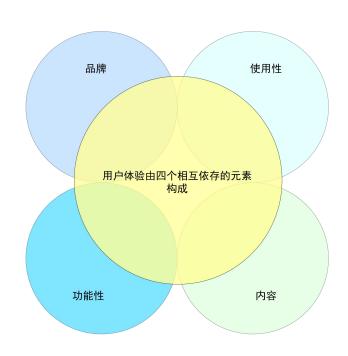
5.2理解用户

■ 5.2.1用户

- □ 用户是使用某种产品的人,其包含两层含义: 1) 用户是人 类的一部分; 2) 用户是产品的使用者。
- 衡量一个以用户为中心的设计的好坏,关键点是强调产品的最终使用者与产品之间的交互质量,它包括三方面特性:
 - □ 产品在特定使用环境下为特定用户用于特定用途时所具有的有效性(Effectiveness)、效率(Efficiency)和用户主观满意度(Satisfaction)。
- 以用户为中心的设计,其宗旨就是在软件开发过程中要紧紧围绕用户,在系统设计和测试过程中,要有用户的参与,以便及时获得用户的反馈信息,根据用户的需求和反馈信息,不断改进设计,直到满足了用户的需求,这个过程才终止。

5.2.2用户体验

- 用户体验 (User Experience, UX) 通常是指用户在使用产品或系统时的全面体验和满意度。
- 用户体验主要有下列四个元素组成(如图5-5):
 - □ 品牌 (Branding)
 - □ 使用性 (Usability)
 - □ 功能性 (Functionality)
 - □ 内容 (Content)



- 用户体验是一个迭代过程,影响用户体验的因素很多·
 - 现有技术上的限制,使得设计人员必须优先在相对固定的UI 框架内进行设计;
 - □ 设计的创新,在用户的接受程度上也存在一定的风险;
 - □ 开发进度表,也会给这样一种具有艺术性的工作带来压力;
 - 设计人员很容易认为他们了解用户需要,但实际情况常常不 是这样。

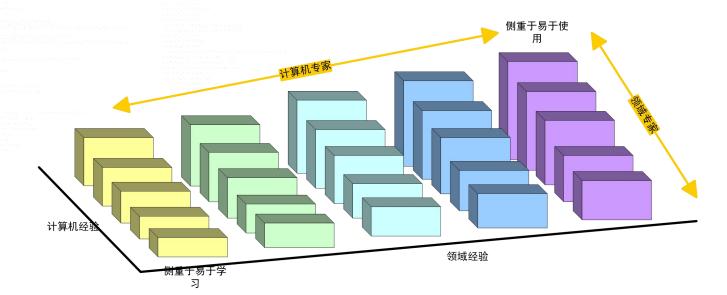
5.2.3用户的区别

■ 1. 用户的分类

- 1) 偶然型用户: 既没有计算机应用领域的专业知识,也缺少计算机系统基本知识的用户。
- 2)生疏型用户:他们更常使用计算机系统,因而对计算机的性能及操作使用,已经有一定程度的理解和经验。但他们往往对新使用的计算机系统缺乏了解,不太熟悉,因此对新系统而言,他们仍旧是生疏用户。
- 3) 熟练型用户: 这类用户一般是专业技术人员, 他们对需要计算机完成的工作任务有清楚地了解, 对计算机系统也有相当多的知识和经验, 并且能熟练地操作、使用。
- 4)专家型用户:对需要计算机完成的工作任务和计算机系统都很精通的,通常是计算机专业用户,称为专家型用户。

5.2.3用户的区别

- 2. 计算机领域经验和问题领域经验的区别
 - 用户界面还必须至少从两个维度迎合潜在的广泛经验,这两个维度指的是计算机经验和领域经验。
 - 计算机经验不仅包括对计算机的一般性了解,还包括对尚待 开发的系统的经验。计算机领域和问题领域经验都不足的用 户所需的用户界面与专家用户的界面将区别很大。



5.2.4用户交互分析

■ 产品策略分析

确定产品的设计方向和预期目标,特别是要了解用户对设计产品的期望是什么;同类型产品的竞争特点,用户使用同类型产品时的交互体验,包括正面的体验和负面的体验,从而得出产品交互设计的策略。

■ 用户分析

深入而明确的了解产品的目标用户(如特定年龄区间、特殊的文化背景、职业特征、计算机使用经验、同类产品使用经验、爱好等等),并找到典型用户。

■ 用户交互特性分析

通过对目标用户群的交互挖掘,得出准确、具体的用户特征,从而可以进行有的放矢地设计。

5.3设计流程

- 5.3.1用户的观察和分析
 - □ 情境访谈 (Contextual Interviews)
 - 走进用户的现实环境,尽量了解你的用户的工作方式、生活环境等情况。
 - □ 焦点小组 (Focus Groups)
 - 组织一组用户进行讨论,让你更了解用户的理解、想法、态度和需求。
 - □ 单独访谈 (Individual Interviews)
 - 一对一的用户讨论,让你了解某个用户是如何工作,使你知道用户的感受、想要什么及其经历等。

■ 5.3.2设计

- 对象模型化: 将用户分析的结果按照讨论的对象进行分类整理,并且以各种图示的方法描述其属性、行为和关系。
 - 比较抽象的视图有利于进行逻辑分析,称为低真视图 (Lowfidelity Prototype);
 - 比较具体的视图更接近于人机界面的最终表达,称为高真视图 (High-fidelity Prototype)

■ 5.3.3实施

- □ 设计师对高真设计原型进行最后的调整,并且撰写产品的设计风格标准(Style Guide),产品各个部分风格的一致性由该标准保证。
- 产品实施或投入市场后,面向用户的设计并没有结束,而是要进一步的搜集用户的评价和建议,以利于下一代产品的开发和研制。

5.4 任务分析

- 在以用户为中心的设计中,关心的是如何从用户那里 理解和获取用户的思维模式,进行充分、直观的表达 ,并用于交互设计。
 - □ 描述用户行为的工具有很多,目前经常提到的是通用标识语言UML (Unified Markup Language)。
 - UML 2.0共有10种图示,分别为组合结构图、用例图、类图、序列图、对象图、协作图、状态图、活动图、组件图和部署图
 - 在任务分析中使用UML工具,可以清晰地表达一个交互任务 诸多方面的内容,包括交互中的使用行为、交互顺序、协作 关系、工序约束等等

名 称	视图	主要符号
组合结构图	表现结构(架构)性需求,主要包括	Part、Port、接口、链接关
(composite-	Part、Port、接口和链接(Link)	系
structurediagram)		
用例图	表现功能需求,主要包括用例和参与	用例、参与者、关联关系
(use case diagram)	者	
类图	表现静态结构,主要包括一群类及其	类、关联关系、泛化关系
(class diagram)	间的静态关系	
序列图	表现一群对象依序传送消息的交互状	对象、消息、活动期
(sequence diagram)	_ 况	
对象图	表现某时刻下的数据结构,主要包括	对象、链接、消息
(object diagram)	一群对象及其间拥有的数据数值	
协作图	表现一群有链接的对象传送消息的交	对象、链接
(collaboration diagram)	互状况	
状态图	表现某种对象的行为,主要呈现一堆	状态、事件、转换、动作
(statechart diagram)	状态因事件而转换的状况	
活动图	表现一段自动转换的活动流程,主要	活动、转换、分叉、接合
(activity diagram)	包括一堆活动及其间的自动转换线	
组件图	表现一群组件及其间的依赖关系	组件、接口、依赖关系、实
(component diagram)		现关系
部署图	表现一堆设备及其间的依赖关系	节点、组件、依赖关系
(deployment diagram)		

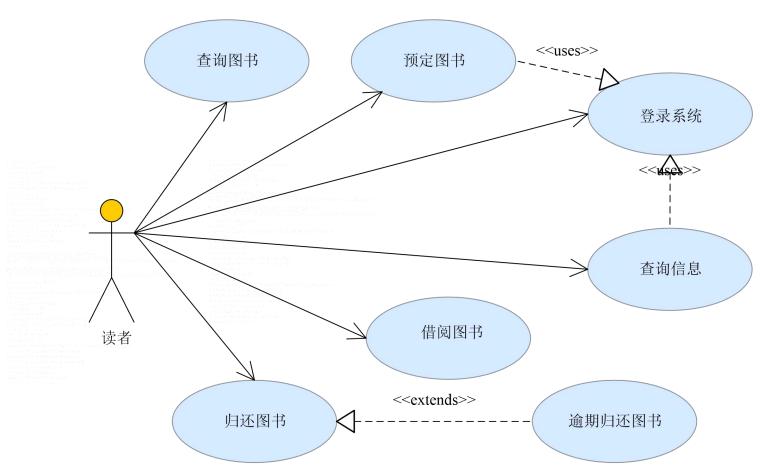
例子

- 图书馆管理系统为例说明任务分析的过程,这个用例 从读者提出想要借书开始,经过如下五个交互步骤:
 - 根据系统提供查询功能,读者可以在系统界面中输入关键字 查询图书;
 - □ 系统通过交互界面列出可借用的图书供读者选择;
 - □ 如果读者选定了图书,系统提示读者输入借书证号和密码;
 - 如果最后读者确定借阅关系,系统处理并通知读者借书成功,并给读者一个确认;
 - 当确认信息出现时,整个图书借阅的交互过程就结束了。
- 下面,我们详细介绍如何用UML来对这些交互任务进行分析。

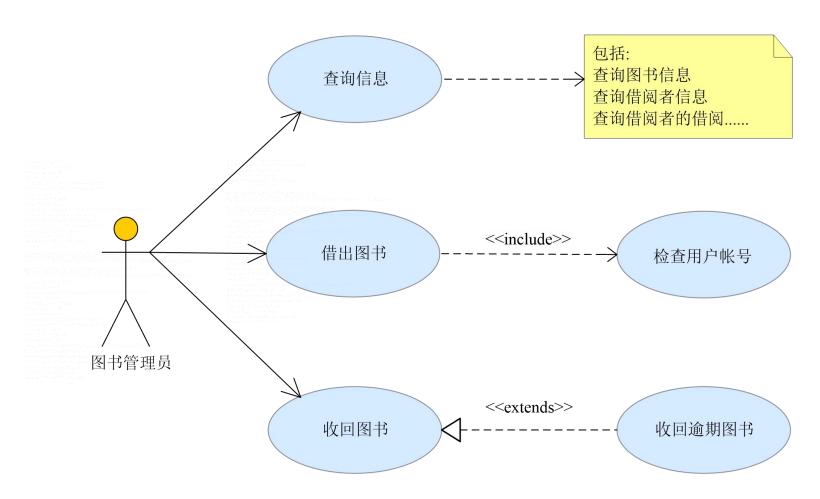
5.4.1使用行为分析

- 使用行为分析就是要理解系统中每个参与者及其所需完成的任务,即分析系统所涉及的问题领域和系统运行的主要任务,分析使用该系统主要功能部分的是哪些人,谁将需要该系统的支持以完成其工作。
- 使用行为分析一般使用用例图描述,它从参与者的角度出发来描述一个系统的功能,主要目的是帮助开发团队以一种可视化的方式理解系统的功能需求。
- 以图书馆管理系统为例,其参与者主要包括:
 - □ 读者(借阅者);
 - □ 图书管理员;
 - □ 图书馆管理系统的系统管理员。

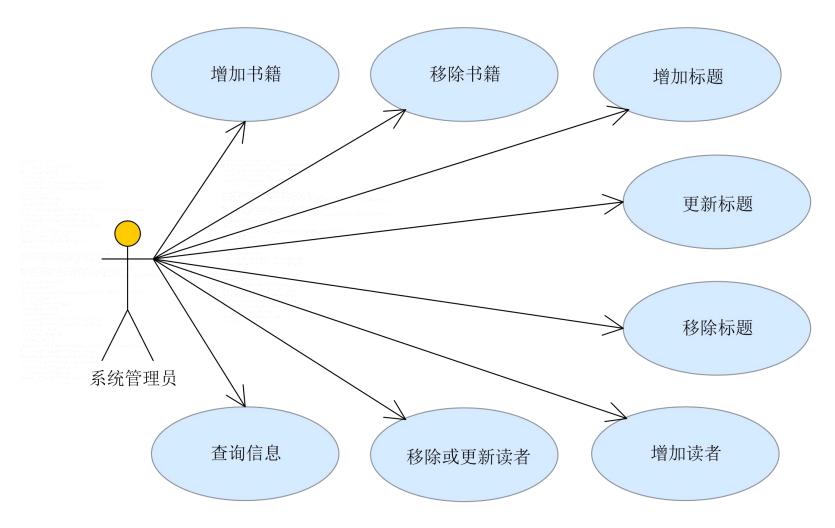
■ 读者使用图书馆管理系统的用例



■ 图书管理员处理借书、还书的用例



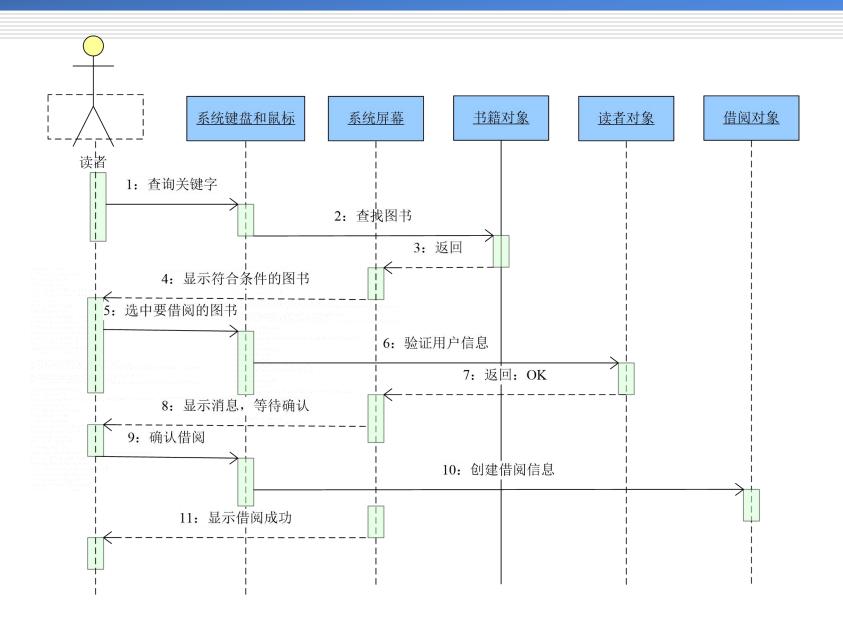
■ 系统管理员进行系统维护的用例



5.4.2顺序分析

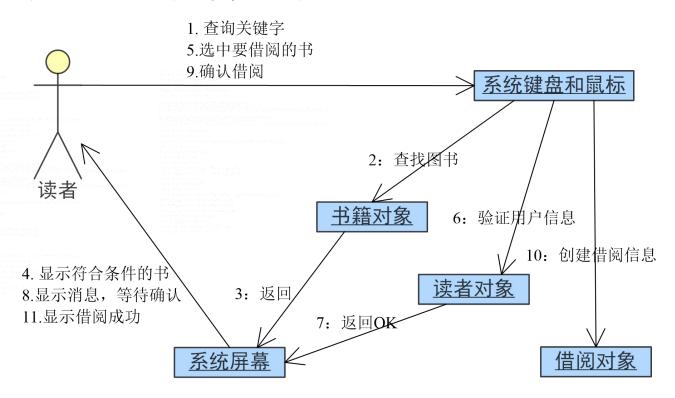
- 每个使用行为都是由若干步骤组成的,这些步骤可以 使用顺序图进行描述。
- 顺序图描述了完成一个任务的典型步骤;
- 它可以按照交互任务发生的时间顺序,把用例表达的需求转化为进一步、更加正式层次的精细表达;
- 用例常常被细化为一个或更多的顺序图。

读者借书时序图



5.4.3协作关系分析

- 协作图着重显示了某个用户行为中各个系统元素之间 的关系,而不再重点强调各个步骤的时间顺序。
 - □ 读者借书过程中,几个交互对象之间的协作关系:



5.4.4工序约束陈述

- 用户完成任务的步骤又被称为工序,某些工序之间的顺序是由一些逻辑关系的。工序约束陈述是工序分析的最直接的方法。
- 本案例中可能存在如下工序约束:
 - □ 系统管理员必须先增加借阅者信息,读者才能登陆。
 - □ 系统管理员必须先增加书籍信息,读者才能查阅。
 - 读者借阅信息生成后,图书管理员图书管理员才能去书库取书。
 - □ 读者必须先在系统中办理借阅,才能取书。
 - □ 读者必须先借书才能还书。

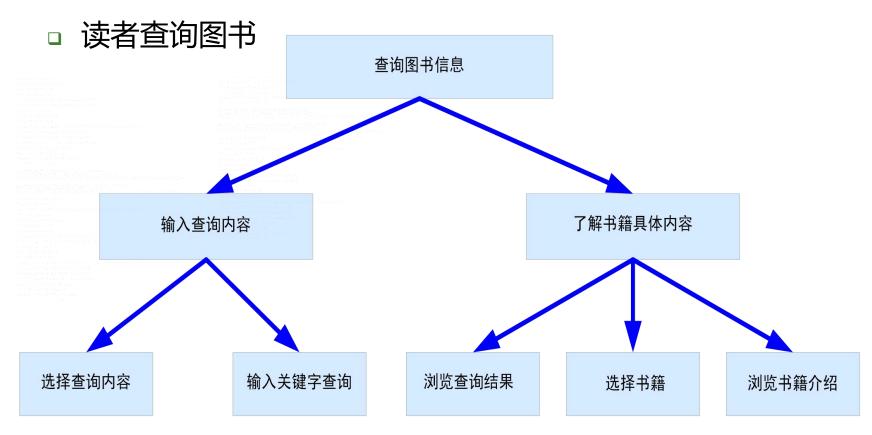
5.4.5用户任务一览表

当所有任务分析完毕,就可以用一览表的形式描述系统中的所有用户及其可能需要完成的所有任务:可以一目了然的展示所有用户的交互任务信息,并且便于更改和调整。

任务	读者	图书馆管理员	系统管理员
书籍信息查询、读者信息查询	$\sqrt{}$	V	
借书	V	V	
还书	$\sqrt{}$	V	
书籍预定	$\sqrt{}$	$\sqrt{}$	
增加、删除或更新书目			V
增加、删除书籍			V
增加、删除或更新读者帐户信息			V

5.4.6任务金字塔

任务金字塔描述了不同层次的任务之间的关系。任何 一个任务都可能包括若干子任务,从而构成金字塔状 的结构。



5.4.7故事讲述和情节分析

- 通过描述实际的任务场景可以非常直观的进行任务描述,便于与用户的交流,并可以帮助分析设计者和真正用户之间对任务的不同理解。
- 故事讲述(story telling)可以是真实的案例,也可以 是虚构的情节,甚至可以是对理想场景的虚构,关键 是使这些故事能够典型的反映交互任务,具有充分的 代表性。
- 情节分析 (scenario analysis) 是对故事所反映的交 互任务的理性分析,分离出故事中所描述的角色、目 标、环境、步骤、策略、感情等诸方面的因素。

例子

■ 学生借书的过程描述图书馆里系统的交互故事

- □ 角色: 刘凡, 图书馆读者; 图书管理员。
- □ 目标:完成书籍的借阅或预定。
- 环境: 图书馆借阅大厅,有查询电脑可供查询使用;借书处,取到借阅的书籍。
- 步骤:查询书籍,浏览图书信息,确定要借阅的书籍;然后 在系统中办理借阅,并等待从借书处取书。
- □ 策略: 如果图书在馆,则借阅;否则,可以预定图书。
- 情感:交互系统的交互过程简洁、顺畅,信息提示充分、清晰,用户对完成任务的过程感到满意。

5.5 以用户为中心的界面设计

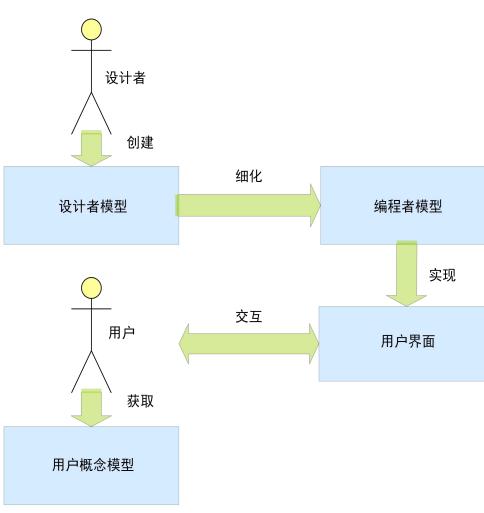
- 以用户为中心设计的四个重要原则。
 - 及早以用户为中心: 设计人员应当在设计过程的早期就致力于了解用户的需要。
 - □ **综合设计**:设计的所有方面应当齐头并进发展,而不是顺次 发展,使产品的内部设计与用户界面的需要始终保持一致。
 - 及早并持续性地进行测试: 当前对软件测试的唯一可行的方法是根据经验总结出的方法,即若实际用户认为设计是可行的,它就是可行的。通过在开发的全过程引入可用性测试,可以使用户有机会在产品推出之前就设计提供反馈意见。
 - 反复式设计: 大问题往往会掩盖小问题的存在。设计人员和 开发人员应当在整个测试过程中反复对设计进行修改。

用户为中心的设计方法

- 图形用户界面设计与评估(Graphical User Interface Design and Evaluation, GUIDE)
- 以用户为中心的逻辑交互设计 (Logical User-Centred Interaction Design, LUCID)
- 用于交互优化的结构化用户界面设计(Structured User-Interface Design for Interaction Optimisation, STUDIO)
- 以使用为中心的设计 (Usage-Centered Design)
- OVID设计

OVID方法

- 通过对用户、目标和任务的分析,系统地指导人机交互界面设计,以达到用户满意的设计要求
 - 设计者模型就是用对象、对象间的关系等概念来表达目标用户意图的概念模型;
 - 编程者模型广泛应用于面向对象的开发方法中,用于表示和实现构成系统的类;
 - 用户概念模型表示用户对系统的理解,它依赖于用户的交互经验

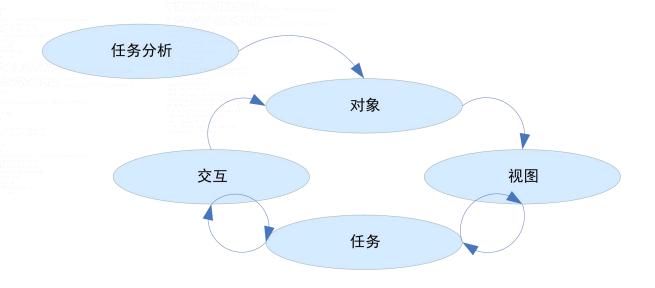


OVID方法的关键

- 确定交互中涉及的对象,并把这些对象组织到交互视图中。
 - □ 其中, 对象来自用户的概念模型
 - □ 视图是支持特定用户任务的对象的有机组合
 - 交互就是那些在交互界面中对对象执行的操作。

OVID中的活动循环

- 对象从用户概念模型的任务分析中获得,并被转化到设计者的对象模型中。
- 交互就是那些界面中执行对象操作的必须动作。
- 如果该模型能够有效地设计和实现,用户就可以通过与系统的交互理解设计者模型所要表达的信息;这些模型可以使用面向对象概念去表达,如统一对象建模语言(UML)等。

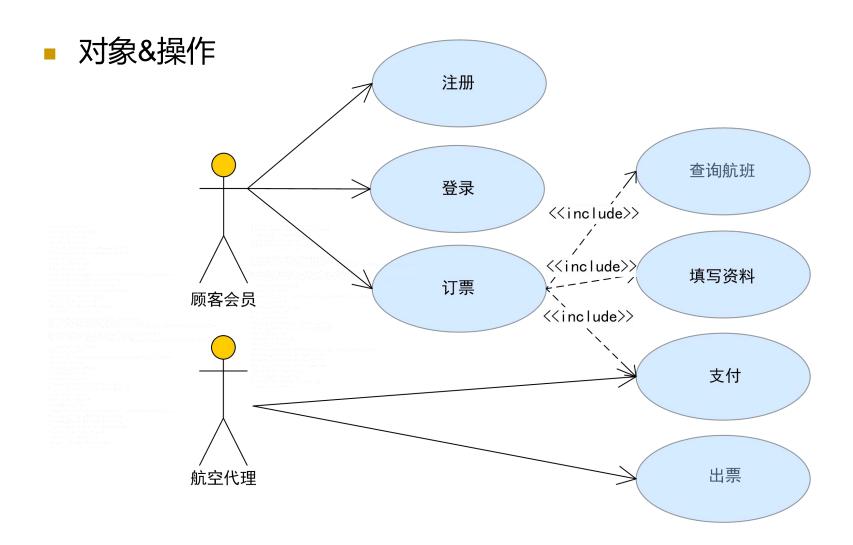


■ 下面以一个网上机票订购系统的界面开发为例,简单 说明OVID方法的过程

```
Committee of the commit
```

5.5.1 对象建模分析

- 建模是将系统任务的某些概念及其关系用图的方式直 观综合地表达出来;分析则是将系统的对象抽象为类 ,列出对象或类的属性、行为、以及对象间的关系。
 - 对象: 乘客会员、航空代理、航班、机票、会员账户、航班 列表等;
 - 操作(交互过程):用户注册、登陆、查询航班信息、填写 预定信息、支付、出票等



5.5.2 视图抽象设计

- 视图抽象设计阶段就是仔细研究系统的对象模型,列出其系统状态,对每个视图抽象出其中涉及的对象,以及对象的属性和行为。
 - □ 用户查询航班视图
 - □ 航班信息列表视图
 - □ 用户选中的某个具体航班的信息视图
 - □ 订购信息填写视图
 - □ 支付视图
 - □ 交易成功反馈和出票视图等

□ 用户查询航班视图

■ 对象: 航班对象

属性: 出发城市、到达城市、航空公司、起飞日期时间、机票 类别以及出票城市等,

操作:主要是查询,即要从后台数据库中查出符合条件的航班信息,以及重置查询条件等操作。

5.5.3 概要设计

针对特定的操作系统或交互方式,对抽象的视图设计 做进一步的具体设计,产生视图的概要设计。实际设 计中,这些视图通常是用铅笔画在纸上,这样做速度 快,而且修改起来也比较方便。

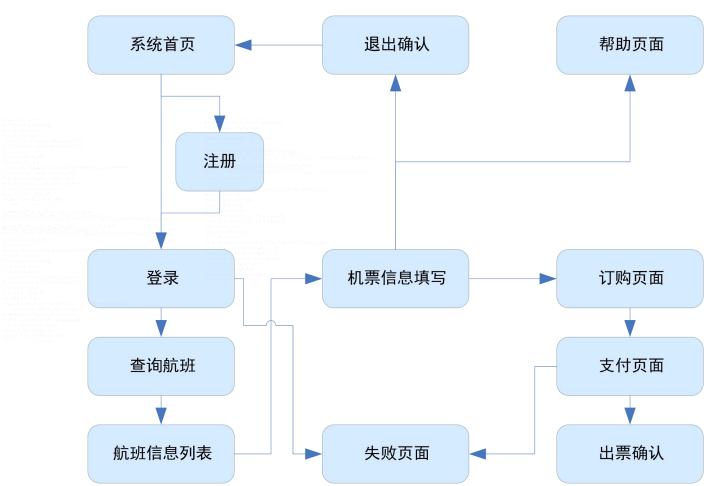
航班号	起飞城市	到达城市	起飞时间	到达时间	全票票价	剩余票额	
CA1100	济南	北京	17: 35	19: 10	500	20	
	ì	丁票	打印	保存	上一页	下一	页
	查询条件	=			新查询	结果中面	查询

- 例如对于视窗系统界面,在用户执行航班查询后,要 在屏幕上显示航班信息和可能的进一步交互动作
 - 航班信息如何展现?以什么顺序显示?已经订满的航班还需不需要显示?如果要显示的话,是不是用不同的颜色?如果信息超出一屏是用滚动条还使用分页的方式?这些都要通过与用户一起分析来确定,最大限度的满足用户可用性。
 - 进一步的操作如何展示?进一步查询的条件如何输入?是否提供打印功能?用户通过什么方式确定要订购的航班?是否允许用户把查到的信息存入收藏夹?
 - 考虑与其他界面可能的关联接口。对于其他的用户界面,如电话语音订票或手机短信订票,返回信息的表示方式就大为不同,所以需要针对不同界面进行交互设计,但这些设计都是来自同一个抽象设法

5.5.4 视图的关联设计

- 很多交互任务需要从一个状态转化为另一个状态,这 就要考虑用户完成任务所需的信息和功能,并将不同 交互视图之间的联系和状态转换关系整理清楚。
- 对一个具体的交互视图进行关联性设计一般就要考虑 以下因素:
 - 该视图的前一个或几个视图是什么?用户怎样由前面的视图 到达该视图?
 - 该视图后面的视图是什么? 也就是用户下一步可以进入哪些视图?
 - 如何从一个视图转移到另一个视图,即转移的条件或操作是 什么?

订票业务网站的桌面交互实现中,用户可能在整个业务相关的多个交互视图中进行转换



5.5.5 视图的全面设计

确定各个视图的具体内容和大致布局,并在每个视图 上明确体现与其他视图的关系,保证系统的整体性和 和谐性。然后可以借助具体的开发工具进行界面的实 际设计。

```
Supplies to the control of the contr
```

- □ 浏览器交互方式下,Web界面视图的整体性主要通过下列几点保证 ·
 - 使用相同的界面风格,包括颜色、字体、布局、行距,间距,导航条等;
 - 使用相同的识别标志,如公司LOGO,底纹图案,版权和联系方式等;
 - 系统视图结构清晰,在每个界面上明确表示当前视图与整体系统的关系;
 - 使用一致的术语,特别是在不同语言的版本之间保持信息翻译的一致。
 - 视图之间的转化是通过用户的交互动作实现的,如在交互元素上双击鼠标左键,单击按钮或链接等;在全面设计阶段要将各个视图间的转换条件和执行的交互动作予以明确说明。

习题

- 5.1请简要论述界面设计的一般原则。
- 5.2请描述任务分析主要包括哪些内容?
- 5.3 利用本章介绍的人机交互界面设计方法,完成网上银行系统的交互界面分析和设计,包括账户查询、存款、取款、转账等业务流程。该系统要能够同时支持浏览器方式和电话银行方式(可参考互联网上实际网上银行的设计)。
- 课堂作业:
 - □ 画出ATM系统用例图
 - □ 画出ATM取款时序图