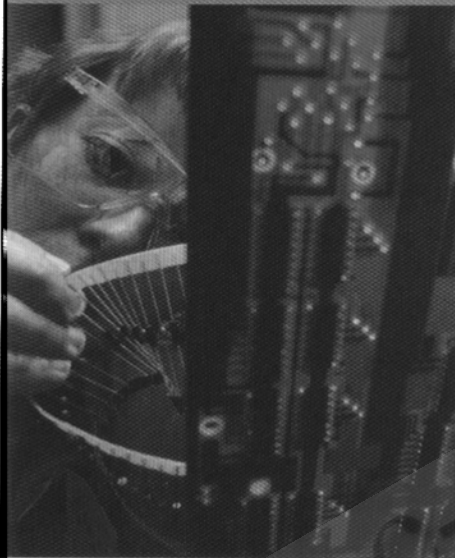


佟玉泉 石博慧 陈 英 (北京理工大学  
计算机科学与工程系 100081)



# UI 设计中面向对象方法 研究与应用

## Research and Apply Object-oriented Method on UI Design

**摘要:** 随着软件规模的日益庞大, 用户界面的设计工作也越来越繁重和复杂, 已经成为软件设计过程中非常关键的部分。本文描述了面向对象方法进行软件UI设计的过程, 并应用到具体的实例之中, 其中还提出了对UI对象建模的概念, 具有一定的理论和应用价值。

**关键词:** UI对象 设计模式

### 1 引言

面向对象是于20世纪80年代形成的一种开发方法, 它比起70年代所流行的结构化开发方法具有明显的优势。尤其在针对大型系统的开发工作中, 应用面向对象方法开发出来的程序更加易于维护和理解, 而且开发效率更高。

所谓对象就是由现实中的事物或概念所抽象出来的, 对象本身具有自己的属性和行为。应用面向对象思想的目的是把现实世界的事物模型化, 这样设计出来的系统无论是对系统开发人员还是系统使用者都更易于把握和理解。

在应用面向对象的开发方法中, 设计和实现是其中最为重要的两个步骤, 其中设计更为关键, 有了好的设计思想才能应用上优秀的实现方法, 更会影响到维护工作的难易。目前相对于面向对象的设计方法, 面向对象的实现方法则更为成熟, 面向对象的编程语言已经被应用的十分广泛, 很多指导面向对象编程的思想和模式也日新月异, 像GOF的设计模式思想, 已经被众多开发人员和设计人员广泛使用。

而随着人们对计算机软件需求的不断增长, 软件的规模越来越巨大, 软件设计在整个软件开发中的重要性也越来越被体现出来。其中软件的界面设计, 即UI设计也越来越引起开发者们的注意。

UI用户界面是软件系统面对用户的终端, UI的结构体系与整个软件结构设计是息息相关、相辅相成的, 而像网站等系统的UI设计则更重要, 往往在软件需求阶段就基本确定了界面的原型结构, 作为与用户交流的有力工具, 最后围绕UI设计进行编码。

### 2 UI 设计中应用面向对象思想

混乱的界面设计将会带来很多麻烦, 比如用户难以使用、代码难以实现高复用性

等。所以有些时候应用面向对象的思想进行界面设计就会很好的解决上述问题。其基本原理是将一组描述同一事物或执行过程的界面作为一个对象来对待, 在进行软件系统分析之前, 首先就是要列出系统所涉及到的用户角色、执行任务, 再通过组织这些用户角色和执行任务的逻辑关系来实现整个系统的功能。例如, 如果要实现一个学生管理系统, “学生”这个对象是一定需要的, 围绕“学生”对象应该还有一些属性, 至少要有个人基本信息, 其中包括学生姓名、学号、性别等; 而学生的基本行为就是上课, 所以又需要“课程”这个对象。这样就构成了一个简单的学生管理系统。

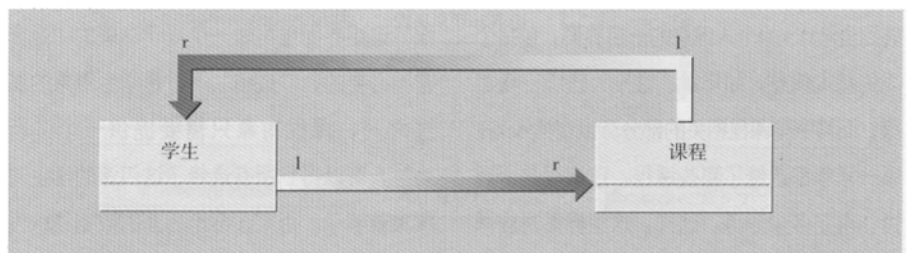


图1 学生管理系统的对象关系示意图

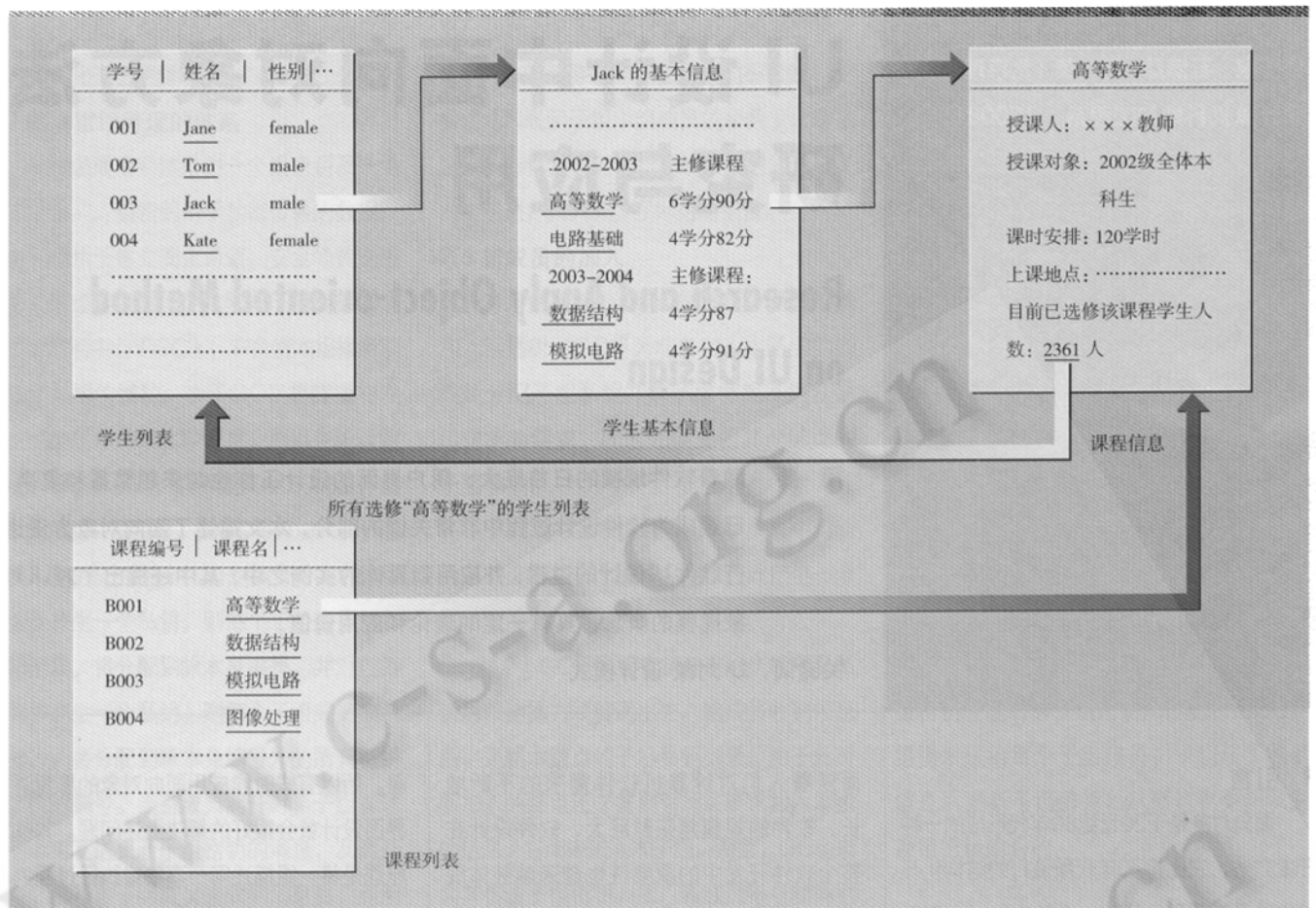


图2 UI跳转关系示意图

图1反映了该系统的基本关系结构，一个学生可以选择1门或多门课程来学习；而一门课程又可以同时被多个学生选修。这样在执行该系统的查询功能时应该可以得到所有学生的列表和所有待选课程的列表，其中把每名学生和每门课程都作为一个对象来对待，这样的逻辑关系组织比较清晰，用户也更容易理解。

学生管理系统的UI跳转示意图如图2所示，当选中其中的一名学生时（如jack），即转到显示jack的个人信息的界面，包括个人的基本信息，如年龄、性别、班级、籍贯等，而其中与课程相关的部分可以包括jack在哪一学年都选修了哪些课程，该课程是否通过，得了多少分等。这样，这组界面对应学生这个对象，Jack就相当于这个对象的实例；Jack的个人基本信息相当于对象的属性或状

态；与课程相关的部分相当于对象的行为或方法，用作得到外部对象的状态和信息。

上述设计似乎是理所当然的，但其实这里要求对对象行为职责的把握要非常清晰，否则细微的变化都会使得结果非常不同。比如，按实际学校中常规的做法应该是教授每一门课程的老师保存一份选修了这门课程的学生的名单，这样的话，在设计界面的时候似乎应该在“课程”这组界面里还要可以显示学生的编号、姓名、成绩，但实际在“课程”这组界面里再做一个显示这些的UI是非常不必要的，“成绩”不过是学生对象的属性之一，课程对象只需要提供一个“行为”，可以得到所有选修了这门课的学生对象集就够了。否则还会给后面的数据结构设计、编程实现都带来不小的麻烦。

另外，在进行UI设计时也可以实现“抽

象”，但这种“抽象”的目的并不是像程序员在设计C++程序时所做的那样，从具有共性的对象中抽出一个基类，以便于以后的维护与重用；而是从用户接受新事物的角度来考虑，使他们通过对一组界面的接触，能够大概了解这一类事物的界面属性和操作方式。

### 3 基于OO方法的UI设计模式

一个好的用户界面的设计应该具有易于构建、修改和维护的特性，而这些单单依靠面向对象的设计方法是不够的，还需要引入模式的概念。

这里的模式是指一种方法学或一种规则。模式所描述的问题及问题的答案都是具有代表性的，类似的问题会重复出现，使用者就可以将模式应用到不同的环境中去，解决这类问题。

在整个软件设计中的不同阶段，所用到的模式并不相同。按所应用到的阶段不同可分为三种不同层次的模式。

**架构模式 (Architectural Pattern) :** 一个架构模式描述软件系统里的基本的结构组织或纲要。架构模式提供一些实现定义好的子系统，指定它们的职责，并给出把它们组织在一起的法则和指南。

**设计模式 (Design Pattern) :** 一个设计模式提供一种提炼子系统或软件系统中的组件或者它们之间关系的一种纲要设计。设计模式描述普遍存在的在相互通信的组件中重复出现的结构，这种结构解决在一定背景中的具有一般性的设计问题。

**代码模式或成例 (Coding Pattern 或 Idiom) :** 代码模式是较低层次的模式。与编程语言密切相关。代码模式描述怎样利用一个特定的编程语言的特点来实现一个组件的某些特定的方面和关系。

显然，在UI设计中用到的模式应该就是架构模式。常用的架构模式如MVC (Model View Controller)、MVP (Model View Presenter) 等。图3、图4分别是两种模式的示意图。

MVC模式将用户界面划分成三个部分，模型、视图和控制器，模型代表域级别的对象，例如，它保持着应用数据并提供访问和修改这些数据的方法；视图则是将模型数据表现在屏幕上的可视化组件；控制器是根据用户输入切换当前所控制的模型。

MVP模式是MVC模式的一个变种，它也是将用户界面分成三个部分，模型、视图和推荐者。其中模型和视图所代表的功能与MVC中相同，这里视图获得较低层的用户输入事件并将之转给推荐者，推荐者根据视图对象来匹配模型和修改模型作为对事件的响应。

相对于整个软件设计，架构模式是在宏观上将软件划分成不同的层次，降低系统组件间的耦合性，提高复用性和维护性。而且只有应用架构模式为根基，UI设计中的面向对象思想才有用武之地。

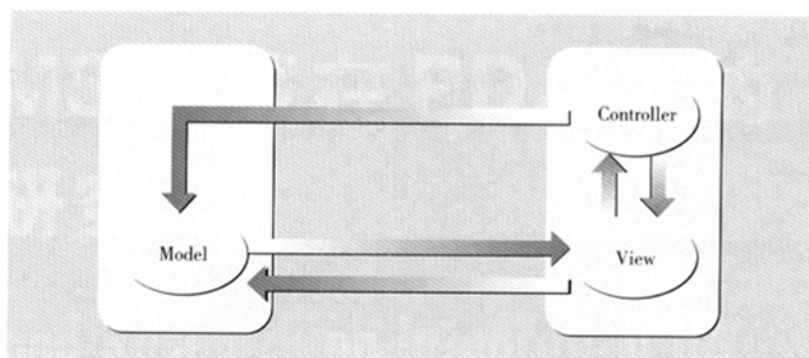


图3 Model View Controller

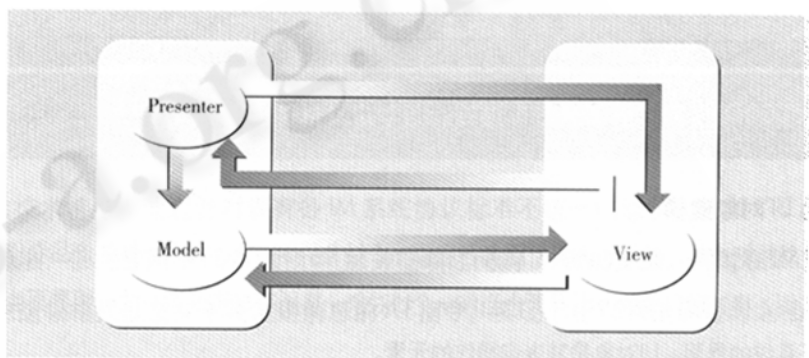


图4 Model View Presenter

#### 4 应用 UI 对象建模辅助 UI 设计

从方法论的角度看：面向对象的方法是面向对象的世界观在方法论中的直接体现，它强调系统结构应该直接与现实世界的结构相应，应当围绕现实世界中的对象来构造系统，而不是围绕功能来构造系统。在构造系统的UI时应用这种方法，也是一样，其主要目的就是使用户觉得系统看上去跟现实世界中的事物更接近、更友好。

下面举个例子，假如现在要设计一个企业信息门户网站，那么在进行完需求分析、核心对象建模等工作之后，就要开始进行系统界面的设计。一般界面的设计很大程度上依赖于设计者的经验，但即使再有经验的设计师也很难一次设计出完全令用户满意的界面，这是由于不同的用户拥有对事物不同的理解方式，有时这种差异可能很大，这样在整个界面的设计过程中，还需要有用户评估界面的程序。整个界面设计周期示意图如图5所示。

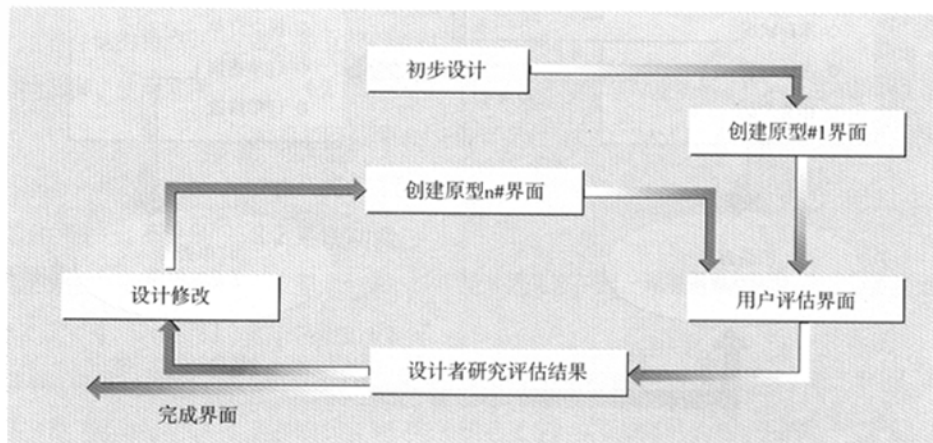


图5 界面设计周期示意图



### 4.1 根据核心对象创建导航栏目

假设该系统的核心对象已确定，分别是产品、新闻、服务和人员，那么应该根据这些核心对象在网站的首页上来创建一个导航栏目，使用户访问该网站时一目了然的清楚这里所提供信息的主题。

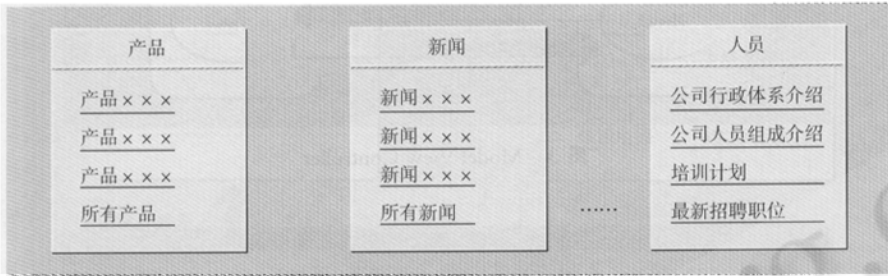


图6 导航栏目界面示意图

### 4.2 为UI对象建模

这里就把产品、新闻、人员、服务各看作一个抽象的UI对象，其中具体的产品看作对象的具体实例，将处理订单的一组界面也当作一个UI对象；将所有与查询相关的一组界面当作一个查询子系统的界面，UI对象是其查询操作的元素。

图7示出了针对产品对象及其上的操作的UI对象模型图。其中所用到的图形符号与UML中所定义的图形符号很相似，因为无论是UML还是UI对象建模，都是以面向对象的理论思想为根基的，所以其中用到的跟UML中相似的图形符号所代表的意义与其在UML中所代表的意义也是相似的。只不过所描述的基本元素都是UI。例如：椭圆形符号代表功能系统的界面；矩形符号代表UI对象；圆形符号代表抽象UI对象；箭头上<<.....>>中的字指明两者之间的关系。

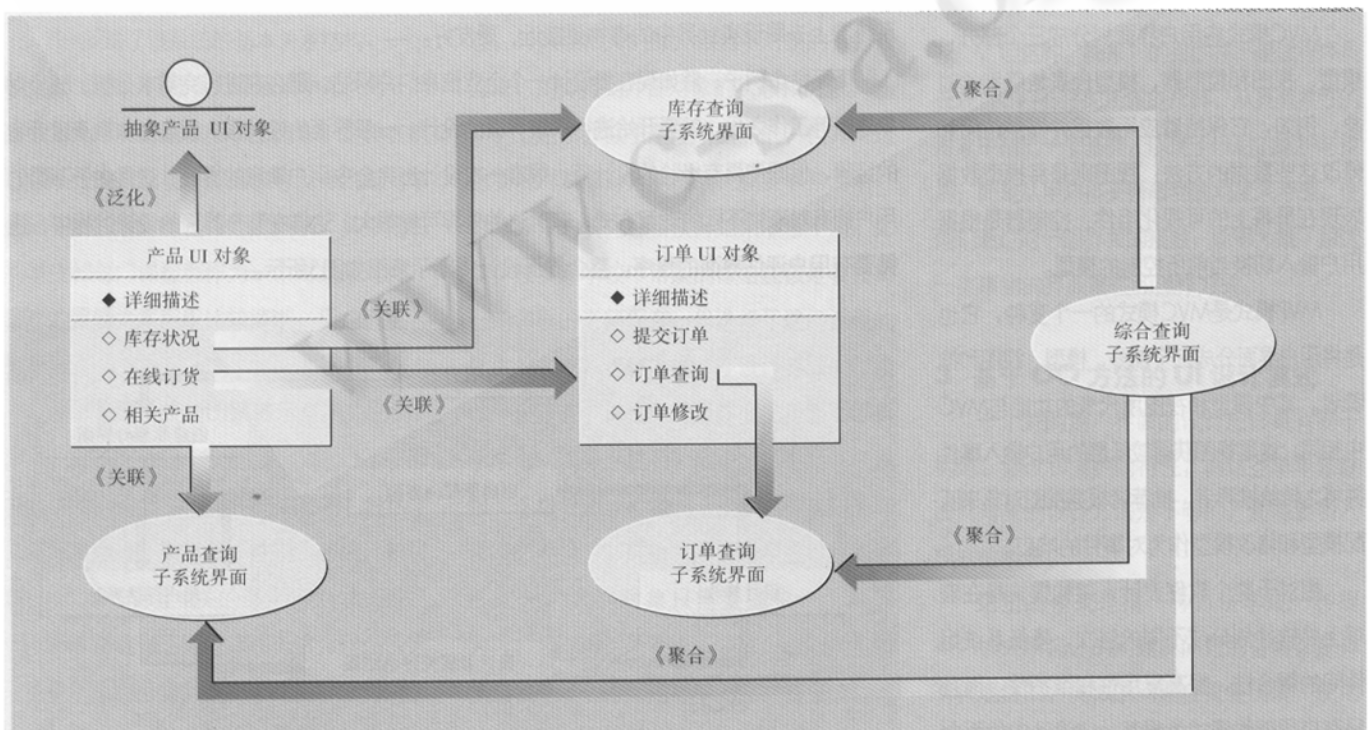


图7 UI对象模型图

### 5 结束语

本文所介绍的面向对象UI设计方法，是将面向对象的设计思想融合于UI的设计当中，改进了传统界面设计方法的许多不足之处。在UI设计过程中增加了UI对象建模的过程，这使UI设计过程更具规范化，并且有助于提高开发效率和增强系统的复用性。而且以面向对象思想设计出的UI，与现实世界的操作模式更为接近，更容易被用户所接受，减少了设计工作的反复。

### 参考文献

- 1 Dorin Sandu . User Interface Pattern , August 4, 2001.
- 2 Martijn van Welie , Hallvard Tr?tteberg . Interaction Patterns in User Interface 1998.
- 3 Roger S.Pressman . Software Engineering 1999.
- 4 Java 与模式, 阎宏, 2002.