

## 基于设计模式的复杂界面建模方法

高 腾<sup>1</sup>, 史清华<sup>1</sup>, 刘志杰<sup>2</sup>

(1. 山东大学计算机科学与技术学院, 济南 250101; 2. 鲁东大学计算机科学与技术学院, 烟台 264025)

**摘要:** 当前界面设计模式缺乏统一的规范化描述, 导致在利用多个界面设计模式的嵌套组合构建复杂用户界面模型时, 不能在有限区域展示具有层次结构内容的需求。提出一种复杂界面的建模方法。基于对界面基本元素的抽象实现界面设计模式的规范化描述, 并通过对界面元素的定制, 实现复杂界面的建模和目标代码的自动生成。应用研究表明, 该方法可对复杂界面的设计和实现提供有力的支持, 提高用户界面开发的效率。

**关键词:** 复杂界面; 设计模式; 界面基本元素; 代码生成

## Complex Interface Modeling Approach Based on Design Pattern

GAO Teng<sup>1</sup>, SHI Qing-hua<sup>1</sup>, LIU Zhi-jie<sup>2</sup>

(1. School of Computer Science and Technology, Shandong University, Jinan 250101;

2. School of Computer Science and Technology, Ludong University, Yantai 264025)

**【Abstract】** There are no unified normal forms of the descriptions for the User Interface Design Patterns(UIDP) until now, which leads to the great insufficiency in using nested component of multiple interface design patterns to construct complex user interface. Meanwhile it cannot meet the need of displaying the content of a hierarchical structure in the limited area. This paper proposes a complex interface modeling approach. Based on abstracting the interface basic elements, the standardized descriptions of interface design pattern are constructed. The complex interface modeling and target code automatic generation are implemented by customizing interface basic elements. Application research shows that this approach can greatly support the complex interface design and realization, and enhance the efficiency of the user interface development.

**【Key words】** complex interface; design pattern; interface basic elements; code generation

### 1 概述

当前界面工程研究主要集中在基于模型的界面开发和形式化的界面设计描述 2 个方面, 它们从不同角度和层面建立了界面设计规范和开发过程。单独形式化的界面设计描述难以实现系统化、工程化的界面开发<sup>[1]</sup>。基于模型的界面开发方法缺乏对常用设计知识的有效重用<sup>[2]</sup>, 难以充分发挥其作用, 从而制约了界面工程化开发的进一步发展。JUST-U I<sup>[3]</sup>以一种抽象的描述方法来分析并表达 U I 概念需求, 并没有考虑实现细节, 难以获取与界面展示相关的信息, 不能向用户展示界面的原型, 也不能给出如何由模式生成界面的方法。任务模型<sup>[4]</sup>是一种领域相关模型, 其对象结构和关系对于界面复杂性的影响缺乏直观和全面的研究, 影响了模型的描述能力。FUSE<sup>[5]</sup>较早地提出了一个高度综合和基于工具的开发环境, 对整个用户界面开发过程提供了支持。但是 FUSE 采用了不易掌握和实现的形式化概念描述方法, 难以使用。建立界面设计知识与基于模型的界面开发的有效集成机制, 成为界面工程化开发中亟待解决的问题。

### 2 复杂界面

#### 2.1 复杂界面定义

一个复杂界面可以看作是一个展示区域, 一个展示区域又可以划分为一系列的子区域。复杂界面与子界面的关系可以看作是一棵由展示区域节点组成的树。其中, 每一个节点都可以看做是一个展示区域, 根节点表示整个复杂界面的展示区域; 每个节点的孩子集合表示对该节点描述的界面展示区域的一个划分; 叶子节点表示不可划分的最小展示区域,

它是界面设计模式的容器, 通过向展示区域的叶子节点指派相应的界面设计模式, 完成对子界面的设计。界面设计模式的嵌套和组合就看作是叶节点区域的组合。

#### 2.2 复杂界面设计模式的原理

对界面复杂度可依界面展示内容及其结构分类如下:

(1)界面内容是简单结构还是复杂层次结构;(2)界面内容是独立还是与其他界面内容相关联。

对界面设计模式可分为:

(1)简单模式。该模式中展示对象是普通对象, 也是基本对象, 对象属性为基本类型, 没有对象的嵌套关系。指派该对象给界面后, 直接以一定形式在整个界面内展示对象的可视元素。

(2)简单组合模式。该模式中展示对象是普通组合对象, 该对象由若干普通对象组合而成, 只有一层嵌套。把它指派给某区域后, 该区域可再分割, 分割得到的子区域可指派该对象的子对象。

(3)重叠组合模式。该模式中展示对象是重叠组合对象, 该对象由普通对象和组合对象, 或组合对象和组合对象组合而成, 嵌套层次至少为 2 层。把该对象指派给某区域后, 该区域就变为多个界面重叠而成, 它们可覆盖展示, 也可分页展示, 每次仅显示一个界面。对每个界面, 可指派该组合对

**基金项目:** 山东省科技攻关计划基金资助项目(2008GG10001026)

**作者简介:** 高 腾(1985—), 男, 硕士研究生, 主研方向: 软件与数据工程; 史清华, 副教授; 刘志杰, 助教

**收稿日期:** 2009-01-08 **E-mail:** gaotenga@163.com

象的子对象。指派后，依被指派对象类型又可进行相应操作，直到每个界面都是可直接展示和普通对象。

### 3 界面设计模式

**定义 1** 界面设计模式 UIDP= $\langle$ Data Set, Property Set, Action Set, Present Set $\rangle$ 。即：一个界面可以抽象描述为由数据集(Data Set)、属性集(Property Set)、操作集(Action Set)和展示集(Present Set)4种描述复杂界面的基本元素的组合或者是递归的组合。

#### 3.1 描述复杂界面的基本元素

本文用复合数据集的概念来对复杂界面进行描述和操作，从数据、属性、操作和展示4个方面对界面设计模式进行描述，引入4种模型元素从4个方面描述相关的界面信息：Data Set描述的是界面数据信息；Property Set描述的是界面设计模式中相关控件的属性信息；Action Set描述的是用户进行各种操作的交互信息；Present Set则描述最终形成界面的展示信息。除上述4种外，每一个界面还有一个附加的Flag属性，用于描述界面的复杂性。具体定义如下：

**定义 2** 数据集 Data Set= $\langle$ DName, DFrom, DType $\rangle$ ，其中，DName表示数据的名称，为数据集的主键；DFrom表示数据的来源，表示数据是来自用户输入还是来自外部存储；DType表示数据的类型。

**定义 3** 控件属性集 Property Set= $\langle$ PName, DName, PType $\rangle$ ，其中，PName表示属性的名称，是控件属性集的主键；DName为外键，即需要展示的数据；PType表示属性的类型，即展示对象的控件类型，如：Button, List 和 Tree 等。

**定义 4** 操作集 Action Set= $\langle$ AName, PName, ARS, RType $\rangle$ ，其中，AName为操作的名称，也是这个属性的主键；PName为外键，表示这一操作是针对某一控件的；ARS为操作的结果集，即进行某一操作之后所产生的结果；RType表示结果的返回类型。

**定义 5** 展示集 Present Set= $\langle$ Back, PName, Layout, App $\rangle$ ，其中，Back表示展示区域的整体外观；PName为主键，表示这一操作针对的是某一控件或界面；Layout表示展示区域内控件集合的布局；App表示所展示的控件或界面的外观形式。

Flag属性表示界面或展示区域的展示模式，即简单模式、简单组合模式还是重叠组合模式。若Flag=0，表示展示形式为简单模式；Flag=1表示展示形式为简单组合模式；Flag=2表示展示形式为重叠组合模式。

#### 3.2 复杂界面的抽象描述

根据2.1节所述，界面可以视为一棵由展示区域节点组成的树，一个叶节点区域包含一个或者多个界面设计模式。利用递归公式进行抽象描述：

- $\langle$ 界面 $\rangle := \langle$ 展示区域 $\rangle$  (1)
- $\langle$ 展示区域 $\rangle := \langle$ 展示区域 $\rangle^* + \langle$ 关联 $\rangle^*$  (2)
- $\langle$ 展示区域 $\rangle := \langle$ 界面设计模式 $\rangle^*$  (3)
- $\langle$ 界面设计模式 $\rangle := \langle$ Data Set $\rangle^* + \langle$ Property Set $\rangle^* + \langle$ Action Set $\rangle^* + \langle$ Present Set $\rangle^*$  (4)

其中，展示区域之间的 $\langle$ 关联 $\rangle^*$ 事实上是其被指派的子展示区域之间的关系。

#### 3.3 界面设计模式建模分析

利用复杂界面的基本元素和复杂界面的抽象描述的知识来对第2节所阐述的界面结构进行建模分析。

##### (1)简单模式

平面结构的独立界面可以利用式(1)、式(3)和式(4)进行建

模。如图1所示，即Flag=0的简单模式只需要通过对 $\langle$ Data Set $\rangle$ 、 $\langle$ Property Set $\rangle$ 、 $\langle$ Action Set $\rangle$ 、 $\langle$ Present Set $\rangle$ 的描述就可以完成建模。

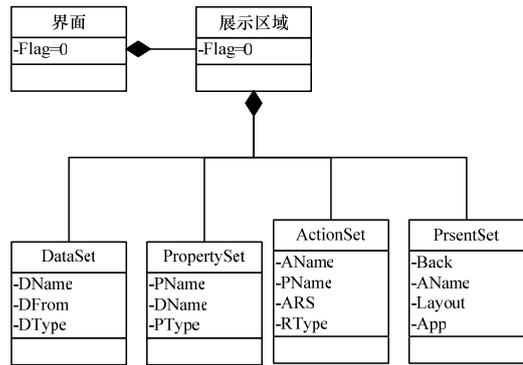


图1 简单模式

##### (2)简单组合模式

平面结构且内容有一定关联或需一起展示，如按钮组、概要与详细信息等，由于有一层内容的嵌套，根据这个模式的特点，可以通过式(1)、式(3)和式(4)进行建模。所建模型如图2所示。

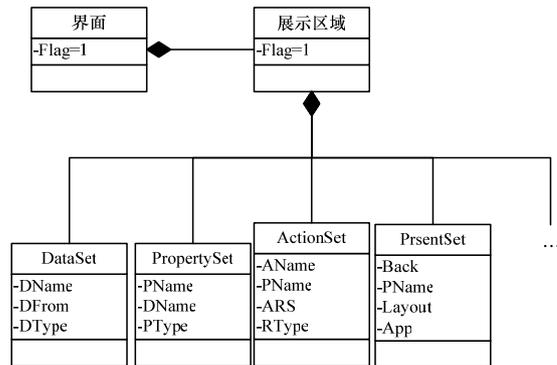


图2 简单组合模式

##### (3)重叠组合模式

多层次结构或需在同一区域展示多项内容，如多级导航菜单、多层属性页，针对这种类型的模式，可以利用式(1)~式(4)对其进行建模。下面对此进行详细说明：

对于一个复杂界面，调用式(1)，即首先将界面看作是一个展示区域。对于整个的展示区域，调用式(2)，可以将整个展示区域分解为若干个子展示区域  $N_1, N_2, \dots, N_m$  以及这些子展示区域之间的关联；分解后的某一子展示区域  $N_p$ ，还可以将其划分为更小的展示区域。对于处在叶节点上的展示区域，调用式(3)，即为该展示区域指派一种界面设计模式，以便对该展示区域进行填充。调用式(4)，对于式(4)中的各个属性集，通过对其中属性的定义和赋值，来完成一个展示区域的建模过程；递归地对其他的展示区域进行建模，最终完成对整个复杂界面的建模。具体过程如图3所示。对于一个Flag=2复杂界面，首先将其分解为一些展示区域，这些展示区域也有多种展示形式，分为简单显示、分页显示和Tab页展示3种，对这些展示方式是通过Present Set中的App属性进行控制的。若App=0表示子界面以简单显示的方式显示；App=1表示展示区域以分页显示的方式展示；App=2表示子界面以Tab页的方式显示。每一个展示区域的设计又是由既定的4种元素组合而成。

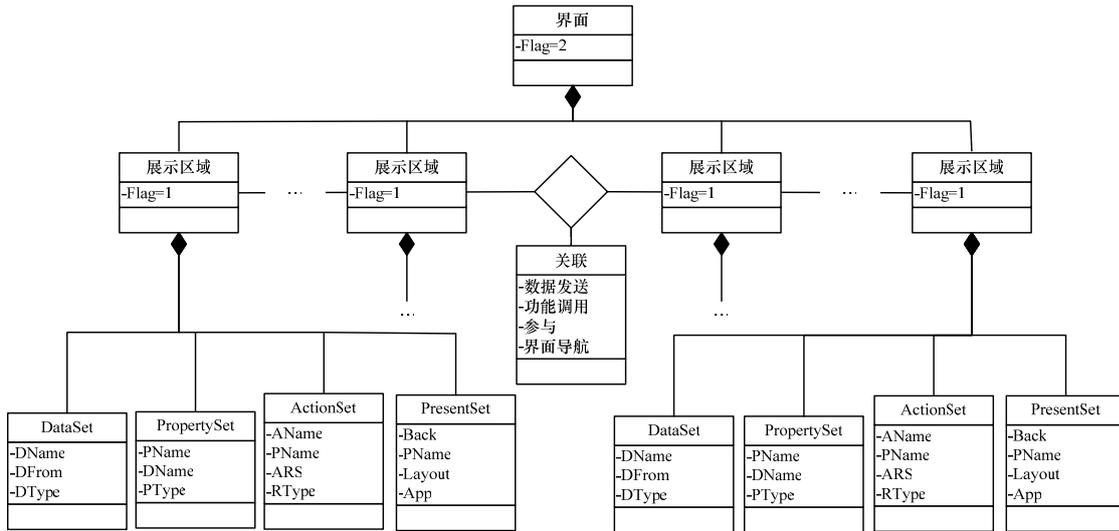


图3 重叠组合模式

#### 4 实验

本部分展示了一个员工管理系统的例子，该例子通过上述的复杂界面模型开发方法建模实现，最终生成基于ASP代码。图4所示为本例 Action Set 中的一个情形，展示了界面不同区域之间的交互以及界面和用户之间的交互。根据本文提出的复杂界面设计模式，图4分为4个部分，位于最顶

层的就是对整个程序进行操作的用户；第二层的查询、添加、编辑、更新和删除则为用例组对象，对界面上提供服务的区域建模；第三层 Employee 是一个汇集模式对象，对界面上展示多条 Employee 信息的区域建模；最后一层详细信息表示的是重叠组合对象，令 Present Set 中的 App 属性等于2，即采用 Tab 页的展示形式。

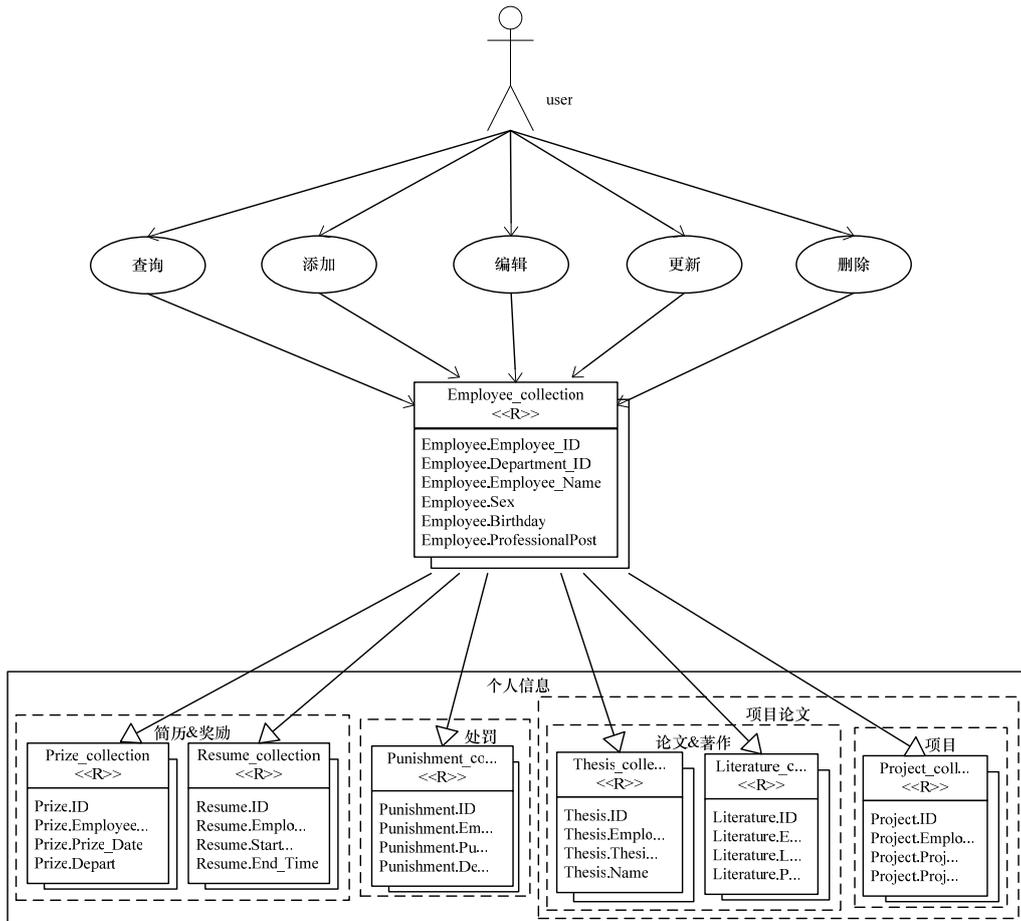


图4 交互模型

用户指向用例组对象的关系为功能使用关系，表示用户使用查询、添加、编辑、更新和删除等功能。用例组对象指

向 Employee 对象的关系为功能调用关系，表示服务模式对汇集模式的数据进行调用。Employee 对象指向组合模式的关系

为数据发送关系,表示将 Employee 的数据发送到组合模式中的各个对象中。

Present Set 的设计包括对子界面划分、子界面内展示对象的指派、子界面内展示对象的布局和数据 Set, Property Set 与子界面的对应编辑等。

对于图 4 设计的一个界面情形,图 5 展示了其中的一种展示方式,设计者可以根据需要定制其布局和外观等展示属性。界面原型上的 3 个区域分别表示用例组对象的按钮展示风格、汇集对象的自由展示风格和重叠组合对象 Tab 页展示风格。

在完成了界面的建模以后,模型基于 Data 和 Property 信息构成界面的静态信息,即图 4 所示的 Action Set 信息以及图 5 所示的界面展示信息,对用户设计的界面进行 ASP 代码生成,如图 6 所示。



图 5 界面模版



图 6 ASP 代码

## 5 结束语

本文针对现有方法对复杂界面设计方面的不足,提出了基于设计模式的复杂界面建模方法,通过在界面设计模式中对界面基本元素 Data Set, Property Set, Action Set, Present Set 的定制,克服了当前基于设计模式的软件开发中对复杂界面支持不足的缺陷,提高了开发的能力,同时为界面目标代码的自动生成提供支持。下一步的主要工作是对界面基本元素支持能力的扩展,此外,对生成基于不同平台的代码也是重要的研究方面。

### 参考文献

- [1] Bowen J, Reeves S. Formal Refinement of Informal GUI Design Artefacts[C]//Proc. of the 2006 Australian Software Engineering Conference. [S. l.]: IEEE Publication, 2006: 221-230.
- [2] Sinnig D, Gaffar A, Reichart D, et al. Patterns in Model-based Engineering[C]//Proc. of the 5th International Conference on Computer-aided Design of User Interfaces. [S. l.]: Springer Netherlands Publisher, 2005: 197-210.
- [3] Molina P J, Melia S, Pastor O. Just-UI: A User Interface Specification Model[C]//Proc. of the 4th International Conference on Computer-aided Design of User Interfaces. Dordrecht, Holland: Kluwer Academics Publisher, 2002: 31-37.
- [4] Mori G, Paterno F, Santoro C. CTTE: Support for Developing and Analysing Task Models for Interactive System Design[J]. IEEE Transactions on Software Engineering, 2002, 28(8): 797-813.
- [5] Lonczewski F. Providing User Support for Interactive Applications with FUSE[C]//Proc. of the 2nd International Conference on Intelligent User Interfaces. [S. l.]: ACM Press, 1997: 253-256.

编辑 任吉慧

(上接第 66 页)

- [2] Droste D B, Allman B. Anatomy of the Next Generation of ATE[M]. Autotestcon, USA: IEEE Press, 2005: 560-569.
- [3] 钟德明, 刘 斌, 阮 谦. 嵌入式软件仿真测试环境软件体系结构研究[J]. 北京航空航天大学学报, 2005, 31(10): 1130-1134.
- [4] 蒋崇武, 杨顺昆, 刘 斌. 面向嵌入式软件测试的仿真建模[J].

计算机工程, 2008, 34(4): 87-89.

- [5] Raskin S. How to Port Python to VxWorks[EB/OL]. (1999-03-12). <http://mail.python.org/pipermail/python-list/1999-May/003929.html>.

编辑 金胡考

(上接第 69 页)

果判定为错误的情况下,其错误的来源,通过数据分析来弥补主观判断的不确定性,具有重要的工程意义。

## 4 结束语

本文在对粗糙集理论研究的基础上,提出基于信息熵的改进算法,得到最优属性约简,并将改进算法应用于列控中心软件测试平台的测试数据分析中,发现该数据集中潜在的数据模式,获取决策规则,从而进一步理解数据的本质。

### 参考文献

- [1] Frank W, Hans T. The Application of Rough Sets Analysis in Activity-based Modeling: Opportunities and Constraints[J]. Expert

Systems with Application, 2004, 27(4): 585-592.

- [2] 苗夺谦, 李道国. 粗糙集理论、算法与应用[M]. 北京: 清华大学出版社, 2008.
- [3] Pawlak Z. Rough Sets[J]. International Journal of Computer and Information Science, 1982, 11(5): 341-356.
- [4] 王国胤. 粗糙集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001.
- [5] 徐章艳. 关于“两种新的决策表属性约简概念”的记注[J]. 小型微型计算机系统, 2007, 28(9): 1686-1689.

编辑 金胡考