

# 基于软件体系结构的用户界面设计方法

王帅强, 万建成, 鹿旭东, 卢雷

(山东大学计算机科学技术学院, 济南 250061)

**摘要:**为实现软件体系结构指导下的软件工程设计, 该文提出了一个在体系结构指导下、支持自动生成的用户界面工程开发方法。该方法把界面的体系结构作为工程模型的直接描述对象, 在界面体系结构的引导和约束下, 通过可视化用户干预建立界面的展示模型, 在代码生成阶段采用了界面体系结构指导下的界面设计模式。研究表明, 以体系结构为指导的软件模型化设计是可行的。

**关键词:** 软件体系结构; 基于软件体系结构的软件开发; 抽象用户界面; 用户界面展示

## Approach of User Interface Design Based on Software Architecture

WANG Shuai-qiang, WAN Jian-cheng, LU Xu-dong, LU Lei

(School of Computer Science and Technology, Shandong University, Jinan 250061)

**【Abstract】** To achieve software engineering design under the guidance of software architecture, this paper proposes an approach of user interface design, which is based on architecture and model and can support automatic code generation. In the approach, the architecture of user interfaces is taken as the direct description of engineering model, and the construction process of the user interface presentation models have to be constrained by the architecture, and within the code generation phases, lots of design patterns of user interface concerning different components can be used. It indicates that software engineering design under the guidance of software architecture is feasible.

**【Key words】** software architecture; software architecture based software development; abstract user interface; user interface presentation

### 1 概述

随着软件工程研究的不断深入和发展, 软件开发不再是“算法+数据结构”, 而是“构件开发+基于体系结构的构件组装”<sup>[1]</sup>, 对软件体系结构作用的认识也不断地深化, 软件体系结构不仅仅在高层设计的描述中, 它已经作用到软件开发的全过程中, “并在软件生命周期的各阶段间保持良好的可追踪性”<sup>[2]</sup>。因此, “软件工程正在以工程为中心转向以软件体系结构为中心”<sup>[3]</sup>, 以软件体系结构为指导的工程设计与代码生成已经成为一个新的发展趋势。

模型驱动构架(MDA)<sup>[4]</sup>的蓬勃开展更加推动了这一方向的研究。例如, ABC方法<sup>[2]</sup>中, 利用ABC/ADL方法进行体系结构建模, 并定义了从ABC/ADL到UML的映射规则, 使体系结构模型可以很容易地合成为UML模型并生成EJB代码; SoftArch提出了一种体系结构的建模工具, 并且通过对体系结构模型的精化, 以XML为媒介将其转化为UML模型, 实现了其工程化应用; 文献[5]中针对ADLs在动态、可进化和可扩充特征描述能力的不足的特点, 提出了较完整的软件体系结构理论体系, 为研究软件体系结构的工程化应用奠定了理论基础。

然而在软件体系结构领域的研究中, 体系结构的工程应用研究仍然相当薄弱, 主要表现在缺乏标准和商业化的体系结构建模工具, 使得体系结构模型在包括用户界面设计在内的工程化应用受到很大限制。

用户界面(简称“界面”)是任何软件主要和不可缺少的成分。然而, 至今没有形成商业上成熟的界面工程设计模型。在UML甚至在MDA<sup>[4]</sup>中也是如此。而基于模型的界面开发环境(MB-UIDE)经历了10多年研究, 传统的模型被扩充并结合

新的概念形成界面的陈述模型, 已报道的有关研究有TERISA, JUST-UI, Wisdom, SUPPLE, Pebbles等, 但是至今没有发展完善且功能强大到被商业和工业界接受的界面模型和工具, 界面的设计仍然主要依靠手工操作和代码编制。

为实现软件体系结构指导下的软件模型化工程设计, 针对上述问题, 本文提出了一个在体系结构指导下、支持自动生成、模型化的用户界面工程开发方法。

### 2 用户界面的体系结构描述

为使界面的模型构造直观和易于接受, 本方法把界面的体系结构作为工程模型的直接描述对象。多项研究都指出体系结构需要多层次的描述, 不同层次使用不同的部件和关系类型。这是软件工程技术和环境的现实, 也是软件设计复杂性的由来。考虑到此, 本方法中的界面体系结构采用了与一般有所不同的部件和关系集合。

#### 2.1 用户界面体系结构的定义

**定义 1** 设 $ADT = \{ADT_1, \dots, ADT_m\}$ 是一个抽象数据类型集合;  $IO = \{IO_1, \dots, IO_n\}$ 是一个抽象界面输入/输出对象集合;  $ID$ 为数据项的名称;  $DR = \langle \text{取值域定义} \rangle$ ;  $DD = \langle \text{默认制定义} \rangle$ ;  $DS = \{ \text{常量, 表} \}$ ;  $L$ 为标签;  $MU$ 为度量单位;  $V = \{ \text{visible, invisible} \}$ 为可见性, 则数据项集合 $DI = \{DI_1, \dots, DI_n\}$ , 数据项 $DI_i = ID \times ADT \times IO \times DR \times DD \times DS \times L \times MU \times V$ 。

**定义 2** 设 $G = \{G_1, \dots, G_k\}$ 为有序的数据项分组,  $G_i = \{DI_m\}$

**基金项目:** 山东省科技发展基金资助项目(20051014)

**作者简介:** 王帅强(1981-), 男, 博士研究生, 主研方向: 软件工程, 形式化方法; 万建成, 教授、博士生导师; 鹿旭东, 讲师、博士研究生; 卢雷, 副教授、博士研究生

**收稿日期:** 2006-11-19 **E-mail:** wangshuaiqiang@mail.sdu.edu.cn

$\{DI_m\} G_i$ , 且  $G_i \neq \Phi$ , 当  $i \neq j$  时,  $G_i \cap G_j = \Phi$ , 令  $G = G_1 \dots G_k$ , 则  $G = DI$ 。

**推论 1** 数据项分组  $G$  是数据项集合  $DI$  的一个有序的划分。

**定义 3** 界面体系结构模型  $UIA = \langle Actor, UC, O, P, U, R_{UIA} \rangle$ 。其中,  $Actor = \{Actor_1, \dots, Actor_m\}$  是一个用户或角色的集合;  $UC = \{UC_1, \dots, UC_n\}$  是一个可见或不可见用例的集合;  $O = \{O_1, O_2, O_3\}$  是一个可见或不可见对象或汇集的集合, 且  $O \subseteq DI$  (见定义1); 其中,  $O_1$  为普通数据对象集合;  $O_2$  为数据查询对象集合;  $O_3$  为控制参数对象集合;  $P = \{P_1, \dots, P_g\}$  是一个可见或不可见组件的集合;  $U = \{U_1, \dots, U_r\}$  是一个用户界面的集合。

**定义 4** 界面体系结构关系  $R_{UIA} = \{R_1, R_2, R_3\}$  是界面体系结构模型包含的关系集合, 其中:

- (1)  $R_1: Actor \rightarrow UC \mid Actor \rightarrow O \mid Actor \leftarrow O \mid Actor \rightarrow P$  为角色对直接可见的用例、对象、汇集或组件的使用;
- (2)  $R_2: UC \rightarrow O \mid UC \rightarrow P \mid UC \rightarrow U \mid UC \rightarrow UC$  为对象、汇集、组件、用例或界面的使用: 关系调用对象、汇集或者组件的一个或者多个方法, 并在符号上显示方法名称;
- (3)  $R_3: O \rightarrow UC \mid O \rightarrow O \mid O \rightarrow P \mid O \rightarrow U$  为事件的激发行为和操作: 对象包含一系列的事件, 当事件到来时, 触发事件响应; 并且在符号上显示时间的名称。

## 2.2 用户界面体系结构

体系结构模型的图示符号如图 1 所示。数据项具有可见性属性, 可见数据项的符号在界面上具有可视的表示。所有可见的对象和汇集都具有特别预定义的抽象事件和操作行为, 可用来指示事件的激发、行为的参与和操作, 包括对对象消息和属性消息的时间事件响应。对象关系指体系结构模型中对象之间的操作关联, 它包括对象或接口的使用、功能调用、界面导航、消息发送、数据传输和对象参与行为等。

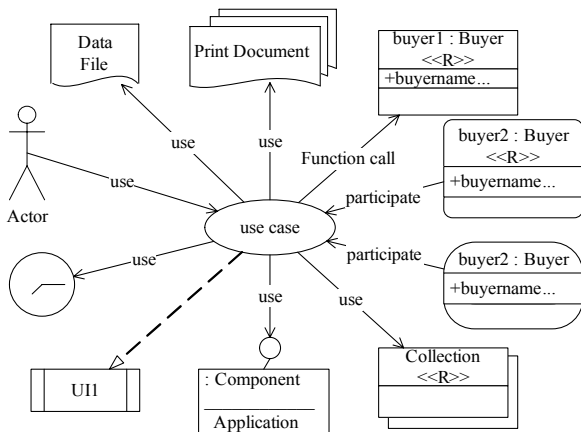


图 1 界面体系结构模型的概念和符号

### 2.2.1 汇集、对象、外部实体/组件

汇集是一种特殊的界面数据对象, 它是数据对象或视图对象的集合。数据对象分为形成查询条件的查询对象、形成控制信息参与用例行为操作的控制对象。汇集和对象都有数据源属性, 如果该属性值是数据库表, 在对象关联的情况下, 来自多个对象, 其界面操作需要访问数据表的特别代码支持。

外部实体或组件指参与界面行为但通常不直接可见的资源。它通常代表的是支持系统工作资源, 诸如数据库服务器、服务组件、时钟、网络连接等。通常, 它们是代表并出现在界面中的系统资源或功能支持。

### 2.2.2 关联、界面、界面功能、界面导航和用例

由于交互关系复杂化, 因此对象间的关联关系是界面复杂化的一个因素。关联是具有特定语义约束的对象间关系。图 1 中没有给出这种关联表示, 但界面代码生成需要考虑这个因素。

界面功能和界面导航是界面需求分析的初始结果, 体系结构模型通过各种对象描述了它们的组成。界面功能是作用在对象上的, 是与对象相关联的, 方法调用、消息发送、事件激发、数据传输和界面导航使得界面功能关联明确且清晰; 界面导航是界面的调用关系, 可以传入或传出参数, 分为模式和非模式两类。

在体系结构模型中, 界面功能由代表操作的用例构成。通常, 不同对象是通过用例而“关联”在一起的, 可见用例对应于界面上的菜单项或命令按钮, 不可见用例对应于内部的服务支持, 如函数或过程; 如同其他可见对象, 模型中可以控制可见用例的“使能”状态。用例及其关系表达了一个算法的外围或体系结构——与关系关联的对象将参与用例所描述的行为操作, 并在用例中以一定次序得到执行, 这与 UML 协作图的对象操作关系类似; 参与用例行为的操作是由标志在关系上的标签内容所确定的。

### 2.3 举例

图 2 给出一个界面体系结构模型的例子。它是学生成绩查询设置界面, 描述了该界面的概念构成。两个对象之间存在数据库的外码关系, 即 Student.ID 为 Score 的外码; 当用户要查找某位同学的成绩, 即要发生 Retrieve 用例操作时, 由查询条件对象得到条件, 然后搜索汇集 Student 中的对象, 并得到满足条件的记录; 汇集 Student 再通过它与汇集 Score 的关系从而得到该同学各门功课的成绩。

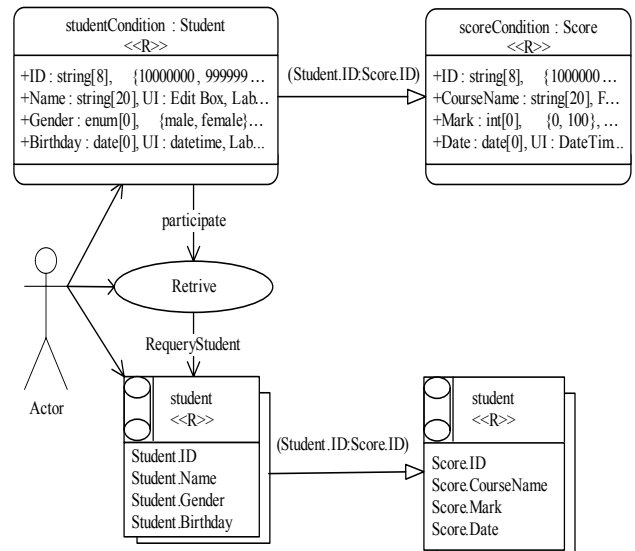


图 2 信息检索界面的体系结构模型

## 3 体系结构约束下的界面展示描述

界面展示描述是在体系结构指导下的处理界面的可视化展示形式和布局规范。它是层次化的, 最底层是体系结构模型, 规范了可视界面对象的细节; 最上层是界面模板, 规范了界面的宏观布局。体系结构模型中界面对象的展示结构、风格和布局策略, 界面模板为整个界面的组成和布局规范提供了表达模型。

### 3.1 界面展示区间, 展示单元和界面模板

界面通常是在一个矩形区间内展示的, 这就是界面展示

区间。为建立具有多成分复杂界面的展示描述，需要对展示区间进行分割以容纳多个展示对象。一个界面展示描述是由多个矩形展示区构成的。定义如下：

**定义 5** 屏幕的矩形区间  $A = \langle W, H \rangle$ 。其中， $W$  为区间的宽度； $H$  为区间的高度。令

$$\text{Condition1} = (\exists A_{i1}) \dots (\exists A_{ik}) (A_{i1} \neq \emptyset \wedge \dots \wedge A_{ik} \neq \emptyset \wedge (\forall ip, iq) (ip, iq \in [i1, ik] \wedge A_{ip} \cap A_{iq} = \emptyset \wedge A_{ip} \cup A_{iq} = A_i))$$

$$A_i = \begin{cases} A_{i1} \cup \dots \cup A_{ik} & \text{Condition1} \\ A_i & \text{其他} \end{cases}$$

**推论 2** 矩形区间  $A$  形成一个树结构。

**证明** 如果条件  $\text{Condition1}$  不成立，则  $A_i$  不变，为叶结点；如果  $\text{Condition1}$  成立，则  $A_i$  可包含一系列子结点  $\{A_{i1} \dots A_{ik}\}$ ，称为  $A_i$  的孩子结点，而称  $A_i$  为它们的父结点；这些子结点每一个又可以包含一系列子结点，直至全部成为叶结点为止。这满足树的定义，所以区域  $A$  形成树结构。证毕。

**推论 3** 矩形区间  $A$  形成的树结构的叶子节点是  $A$  的一个划分。

**定义 6** 假设  $I = \langle A, C, O, P, U, R \rangle$  是被设计界面 UI 的体系结构模型 UIA， $I'$  是一个  $I$  中所有可见项目的集合， $D = \{D_1, \dots, D_n\}$  是  $A$  的一个划分，其中  $D_i$  是该划分树结构的叶节点。界面模板 UIT 是一个经过修饰的关系  $T: I' \rightarrow D$ 。其中， $I'_s \rightarrow$  代表针对  $I'$  中项目的可视化形式所具有的控制参数和布局策略。

**定义 7** 展示单元 PU (presentation unit) 是一个叶结点展示区和展示在该区间的展示对象集合的结合。

由定义 7 可知，界面模板是一个从体系结构模型的界面对象到经过划分的界面展示区间的叶结点的关系，其中包含了展示特性和全局展示策略设置。体系结构模型的所有可见对象将作为出现在展示单元中的展示对象。

### 3.2 体系结构模型与界面模板之间的关系

界面模板是以体系结构模型为指导的一个界面的抽象展示形式。这样，通过展示区间的划分和与体系结构模型中界面对象关系的指派，界面模板描述了一个界面的全局展示和布局结构。

体系结构模型和界面模板之间的关系如图 3 所示。其中， $PU_i$  表示  $i$  号展示单元。每一  $PU_i$  与一或多个界面对象关联；对于数据对象，其扩展对象模型给出了对象的构成和可能的展示细节。依据对象成员的分组关系、导出关系、行为关联、分组和成员次序，在布局策略的约束下，可以决定一个界面对象的局部或细节构成和展示问题。

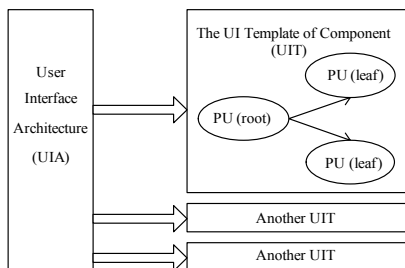


图 3 界面体系结构模型与界面模板的关系

### 3.3 举例

以图 2 作为体系结构模型，可以建立多种界面模板。其中，涉及查询对象可以有两种展示风格：(1)逻辑的合取范式

风格；(2)逻辑组合风格。对于查询结果的展示可以有 5 种风格，其中有汇集的 Free-form、Grid、对象列表、Tree 和 Graph。此外，每一个对象或汇集可以同时使用多于一种的展示风格。

图 4 给出了其中的一种界面模板和生成的界面。其中，展示区间被划分为 5 个子区：查询对象，检索操作的用例，Student 汇集，两个为同一个 Score 汇集。Student 汇集采用 Grid 风格展示。Score 汇集分别以 Grid 和 Graph 风格展示。查询对象以合取范式风格展示。以上这些和其他各种设置都是在界面模板的交互设计环境中进行的，包括汇集的曲线、圆饼图和直方图的展示风格设置。最后生成的代码包含了查询条件的形成、查询操作的实现和关联汇集的 Grid 和 Graph 风格的展示刷新操作。

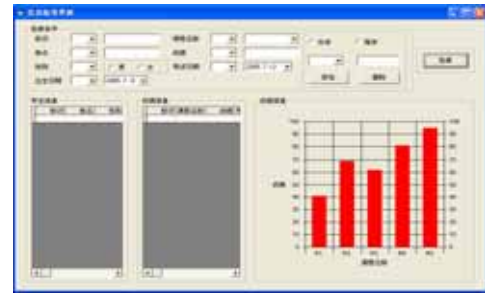


图 4 信息检索界面的界面模板

## 4 体系结构约束下的界面代码生成

正如 MVC 倡导的，每一个界面设计模式有其内部的数据模型、展示构成和行为。

**定义 8** 界面设计模式  $UIDP = \langle M, P, O \rangle$ 。其中， $M$  是抽象的内部体系结构模型 UIA； $P$  是界面模板中对体系结构模型中可见项的相关的组成和参数； $O$  是体系结构模型中定义的操作函数集合。

界面展示是受体系结构约束的，对拥有同一个  $M$  的不同界面设计模式而言，对模型而言是一样的，具有可替换性。

## 5 总结和今后的工作

本文建立了界面体系结构的构造和展示软件环境，并且界面代码生成也是基于体系结构的，即每个界面模板都在它自身的体系结构模型的指导下生成以 Visual Basic, ASP.Net 和 J2EE 为目标平台的代码，使得体系结构模型具备了不同目标平台间的可移植性。研究表明，以体系结构为指导的软件模型化设计是可行的。

今后需要进一步完善界面体系结构模型，总结和丰富界面设计模式。另外，目前的模型和生成的代码中尚缺乏事件响应和用例操作的逻辑描述，对它们的操作逻辑建模和代码模型转换是下一步最重要的工作。

### 参考文献

- 1 杨芙清. 软件复用及相关技术[J]. 计算机科学, 1999, 26(5): 1-4.
- 2 梅宏, 陈锋, 冯耀东. ABC: 基于体系结构、面向构件的软件开发方法[J]. 软件学报, 2003, 14 (4).
- 3 Bosch J. Architecture Centric Software Engineering[C]//Proceedings of the 24th International Conference on Software Engineering. 2002.
- 4 Frankel D S. Model Driven Architecture: Applying MDA to Enterprise Computing[M]. Wiley, 2003.
- 5 赵会群, 王国人, 高远. 软件体系结构抽象模型[J]. 计算机学报, 2002, 25(7): 730-736.