

# 支持界面自动生成的模型研究

杨鹤标, 侯仁刚, 田青华

(江苏大学计算机科学与通信工程学院, 镇江 212013)

**摘要:** 为支持界面工程化设计, 提出一种基于模型驱动思想的支持界面自动生成的 ADS 模型。ADS 模型由分析模型、会话模型和显示模型构成。分析模型完成对界面功能和数据信息的建模, 依据给定的转换规则, 由会话模型生成界面的抽象视图, 通过显示模型处理抽象视图的可视化和布局问题, 并用 UsiXML 界面描述语言加以描述, 组合 UsiXML 片段成 XML 文件, 由框架解析 XML 文件生成用户界面。  
**关键词:** 用户界面模型; 任务; 模型转换; 模型映射

## Research on Model Supporting Interface Automatic Generation

YANG He-biao, HOU Ren-gang, TIAN Qing-hua

(School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013)

**【Abstract】** To support engineering-oriented interface design, the ADS model for interface automatic generation is proposed based on the model-driven opinion. The ADS model has three component models named analysis model, dialog model and show model. The analysis model is used to model interface function and data information, dialog model produces the interface abstract view according to the given transformation rules, and show model handles visualization and layout issues of the abstract views. The UsiXML code segments which describe the ADS model are assembled into XML files, and the user interface is generated by parsing these XML files using framework.

**【Key words】** user interface model; task; model transform; model mapping

### 1 概述

据统计, 用户界面在软件开发中占据 50% 左右的编码时间和运行时间<sup>[1]</sup>。而目前软件工程方法主要侧重软件内部模型的需求分析和设计, 涉及到软件外部用户界面时, 除了纯粹概念性的功能分析和表达外, 并未提供完善的理论和表示支持。为解决这个问题, 界面工程设计和自动生成模型的方法成为软件开发者关注的热点。

界面模型大致分为概念模型和陈述模型 2 种。概念模型从抽象的角度论述界面的体系结构和元素间的逻辑关系, 属于理论意义, 缺乏面向设计的形式化描述, 依赖于设计者的认知。其典型的代表有 Seeheim, MVC 和 PAC 等。陈述模型采用信息元作为分析对象, 通过交互分解构建界面的描述。由于缺乏界面布局的准确描述, 难以满足用户界面复杂性和个性化的需求<sup>[2]</sup>。Tadeus, Trident 和 Mastermind 等是基于该类模型的应用。此外, 从软件工程角度还提出了基于 FVI 框架的面向对象的 GUI 设计模型<sup>[3]</sup>和用户界面体系结构模型 EIP<sup>[4]</sup>。

为了支持界面工程设计, 本文提出一种基于模型驱动思想的支持界面自动生成的 ADS 模型及其支撑框架。ADS 模型通过分析模型捕获界面功能和数据信息, 利用会话模型表达信息和控制关系并以抽象视图方式表示元素构成。依据会话模型描述, 显示模型建立抽象元素和外部显示元素对应关系并提供布局描述。最终, 框架以聚合方式创建和管理这些知识, 创建形式化描述, 实现用户界面的自动生成。

### 2 ADS模型总体结构

ADS 模型由分析模型(Analyze)、会话模型(Dialog)和显示模型>Show)构成。模型间的关系如图 1 所示。

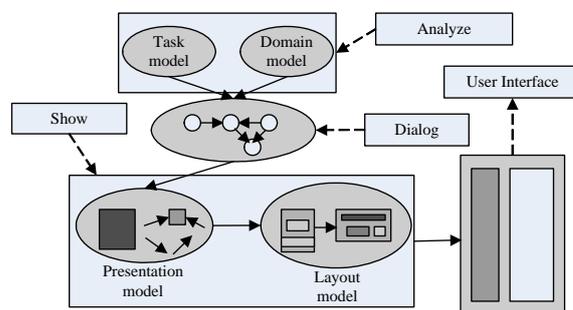


图 1 ADS 模型及内部模型之间关系

Analyze 负责对界面功能建模, 由域模型(DM)和任务模型(TM)组成。任务模型完成对用户任务的分析, 给出任务的层次结构描述以确定界面的功能和界面框架。域模型分析界面状态变化和领域对象, 确定数据信息的需求, 为任务模型和会话模型提供支持。

Dialog 用于描述交互对象之间的内部逻辑关系, 包含用户和系统交互的静态和动态信息, 在域模型的基础上对界面的构成做一个抽象视图描述。

Show 反映界面的外在形式, 由表现模型(PM)和布局模型(LM)组成。它使用平台无关的具体交互对象替代抽象视图中的抽象元素。LM 作为界面的实例实现, 在 PM 的基础上定义界面可视化外观布局。

**基金项目:** 国家“863”计划基金资助项目(2007AA04Z1B2)

**作者简介:** 杨鹤标(1960—), 男, 教授, 主研方向: 软件工程, 数据挖掘; 侯仁刚、田青华, 硕士研究生

**收稿日期:** 2009-09-23 **E-mail:** hourengang@gmail.com

### 3 分析模型

该模型主要工作如下：通过建模获取界面的功能及应用领域内数据对象，确定界面框架和界面处理的数据。

#### 3.1 任务模型

界面 UI 由任务 T 的集合 {T} 组成, T=(D, F, G) 为三元组, 表示用户对数据对象集合 D 实施功能操作 F 完成某个目标 G 的执行序列。在任务模型中, 界面的功能通过任务和任务之间关系的树型结构来定义。

基于任务的层次性, 本文使用 Coucur Task Tree(CTT)<sup>[5]</sup> 作为任务模型规约的符号。该符号可给出不同任务表示和关系的描述, 是 UI 开发社区广为使用的描述任务符号。常用任务符号及关系符号见表 1 和表 2, 其中, T1 和 T2 表示任务。注: 使能\*表示 T1 和 T2 共同作用于同一数据对象时, 执行 T2 之前必须先执行 T1, 允许数据由 T1 传递到 T2。使能不涉及数据传递。

表 1 任务类型

抽象	交互	应用	用户

表 2 任务之间的关系

选择	交叉	同步	顺序无关
T1 [] T2	T1    T2	T1 [] T2	T1  = T2
可选	禁用	使能*	使能
[T1]	T1 [> T2	T1 [ ]>>T2	T1>>T2

图 2 是根据任务符号描述的邮件应用中发邮件的任务分析树。

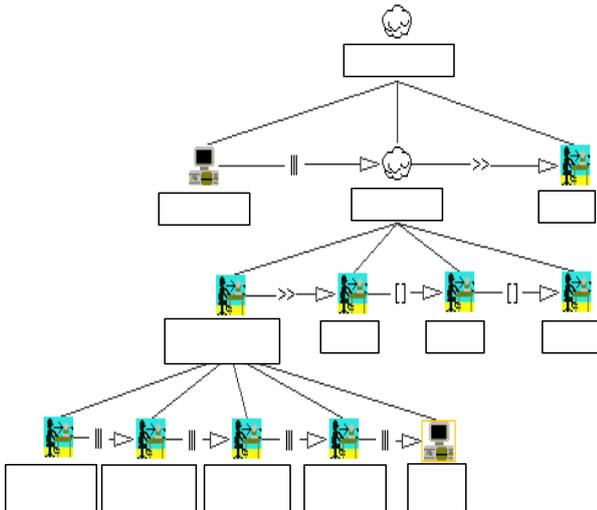


图 2 任务结构

任务和关系的 BNF 范式表示如下:

```
<Task> ::= Des {
<TaskId> id
<TaskName> 名称
<TaskType> 类型
{<Attribute>} 属性
END<Node>
```

关系用 BNF 范式表示如下:

```
<Relation> ::= Des {
<Name> 名称
<Type> 类型
<SourceId> 关系起点任务
<TargetId> } 关系终点任务
END<Node>
```

#### 3.2 域模型

DM 用于表示界面元素的数据对象, 为任务模型提供数据支持。每一个数据对象关联到任务树中一个相关的任务。其 BNF 范式描述如下:

```
<域模型> ::= <名称> <属性> * <方法> *
```

```
<属性> ::= <可见性> <名称> <数据类型> <标签> { <取值范围> <数据来源> } <UI 类型>
```

```
<方法> ::= <可见性> <返回类型> <方法名称> (<参数列表>)
```

域模型描述完成后, 以 XML 的形式存储在库中, 实际运行时由数据对象解析提取各成员属性数据, 并结合任务节点属性绑定抽象交互对象。

#### 3.3 任务模型和域模型之间的映射关系

任务模型生成任务分析树, 树中每个叶节点代表操作(输入或输出), 操作是施于对象的处理。因此, 任务树中表示业务的每个叶节点与域模型中实体的方法和属性存在对应关系。图 3 描述了任务模型和域模型之间的映射关系。上半部分是目标系统的任务模型, 下半部分是域模型的类图表示, 带箭头的虚线表示 2 个模型的映射关系, 图中虚线表示任务到域元素的一种映射关系。域模型为每个任务提供数据支持, 使得任务表述更加具体, 同时任务提供的信息结合域数据又为界面显示提供丰富的信息。

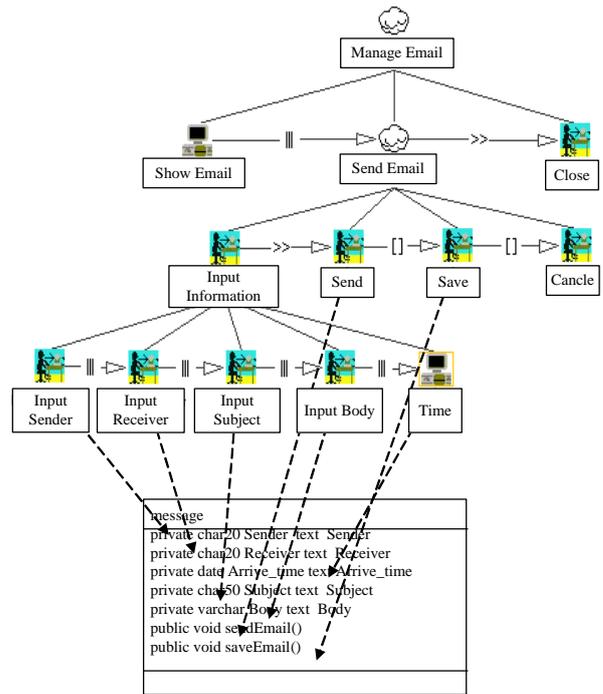


图 3 任务模型与域模型之间的映射

### 4 会话模型

依据数据模型描述, 会话模型使用抽象交互元素表示任务分析树中的节点, 给出界面的抽象视图和各个交互元素的行为定义。具体的方法如下: 用组件抽象视图和组件行为视图表示功能和逻辑上关联的任务模型和域模型元素。前者根据任务分析树和域模型信息生成抽象的界面; 后者进行各个控件的行为定义。

交互组件分为抽象容器(AC)和抽象个体组件(AIC)。容器是一个嵌套的表示单元, 可以嵌套 AC 和 AIC。AIC 代表系统的基本交互功能, 是构成组件抽象视图的基本元素, 有 4 个方面: input(I), output(O), navigation(N), control(C)。各种对象符号见表 3。

表 3 交互元素的抽象符号

I	O	N	C	AC	AIC

从任务分析树和域模型到抽象视图的转换规则定义如下:

**规则 1** 在任务树中,为具有子任务的任务节点创建一个 AC。

**规则 2** 在任务树中,为每个叶节点任务创建一个 AIC。

**规则 3** 为执行输入任务的 AIC 创建一个 input 控件,属性来自于数据对象属性。

注:生成 AIC 类型取决于任务属性和数据对象属性的值。产生 AIC 其他控件的规则类似于规则 3。

规则 1 解析过程如图 4 所示,其他规则类似。

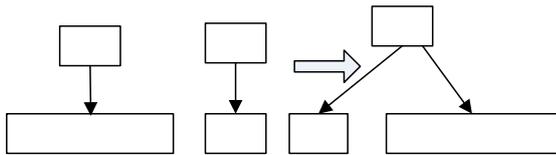


图 4 父任务到 AC 的转换

任务模型和域模型经上述规则转变以后,生成表示界面抽象构成的视图,视图可以以平面方式和树型方式展现结构,图 2 的转换结果如图 5 所示。

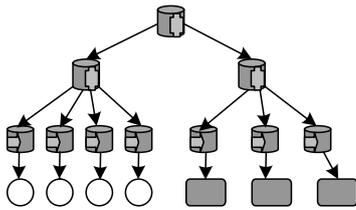


图 5 邮件的组件抽象视图

每一个组件和控件都关联相应的操作,组件行为视图描述每一个组件和控件的行为动作,用户通过该视图能清晰地理解每个元素表达的含义。依据 tasks 1 可获取发邮件会话中的组件行为描述:

- 组件 1: 输入邮件信息
  - 输入控件 1: Sender 输入发件人的 Email 地址。
  - 输入控件 2: Receiver 输入邮件的接收人的 Email 地址。
  - 输入控件 3: Subject 输入邮件的主题。
  - 输入控件 4: Body 输入邮件的主体内容。
- 组件 2: 会话控件
  - 会话控件 1: Send 发送邮件。
  - 会话控件 2: Save 保存邮件。
  - 会话控件 3: Cancel 取消本次邮件。

基于数据对象的支持,会话模型描述界面不再是笼统的任务要求,而是与具体的数据对象建立联系而表示的界面视图。组件行为的描述使得各元素含义更加清晰。

## 5 显示模型

显示模型处理交互对象的可视化和布局问题。其中,PM 负责描述具体容器和控件代替抽象视图中抽象容器和抽象个体组件;LM 位于 PM 之上,描述交互对象可视化属性、结构、风格和布局策略。

### 5.1 表现模型

表现模型以独立于平台的方式具体化会话模型,使用一组 CIO(Concrete Interaction Object),根据转换规则替代会

话模型中的抽象元素。由于 AIC 的类型与数据类型具有多种不同的组合,易产生多种不同的组件的需求,因此对交互组件选择用下述的规则加以约束。

**规则 4** 使用具体容器代替抽象容器。

**规则 5** 为每一个只有一个输入值的输入型控件创建一个文本输入框。

**规则 6** 为每一个具有多值但不多于 N 的只能选择一个值的输入型控件创建一个单选按钮。

规则 6 解析过程如图 6 所示,其他规则类似。

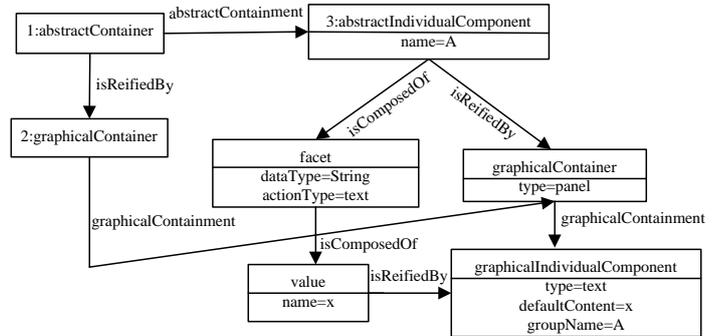


图 6 输入型控件到文本输入框的转换

会话模型中抽象视图采用树型结构表示,经过 CIO 具体化后的表现模型在结构上保持不变。本文使用扩展语义树表示该种结构,扩展语义树对树进行语义扩展使其表示的元素具有生长方向。表现模型由若干个 CIO 构成,CIO 分为原子 CIO 和复合 CIO,原子 CIO 是最基本元素,每个 CIO 在表示界面元素时具有不同的排列结构(生长方向),用递归方式形式化描述如下:

```
<PM> ::= V(<复合 CIO>)|H(<复合 CIO>)
<复合 CIO> ::= (<原子 CIO>)|<原子 CIO>,{V(<复合 CIO>)|H(<复合 CIO>)}|{V(<复合 CIO>)|H(<复合 CIO>)}|<原子 CIO> //V 表示垂直布局, H 表示水平布局
```

这种树型表示与 XML 在结构上具有很大的相似性,采用界面描述语言 UsiXML<sup>[6]</sup>描述模型,可方便地进行模型间的转换。对于上述输入文本框,本文给出一段 xml 描述片段,如下:

```
<abstractContainer name="A" type="panel" alignment="vertical" >
  <graphicalIndividualComponent type="text" dataType="String" label="default" defaultContent="x" groupName="A" />
</abstractContainer>
```

### 5.2 布局模型

显示模型中未涉及到元素的显示属性以及整体布局描述,这些属性须由布局模型定义。布局模型以表现模型的 XML 文件为输入,结合布局模型对元素的布局安排,组合表现模型的各个 UsiXML 片段。为使得各个 UsiXML 片段合并在一起仍保持外观一致,使用布局策略 Welia<sup>[7]</sup>限制各个片段,为满足个性化需求,须对每个交互元素设置样式属性,父节点属性作用全部子节点,子节点属性覆盖父节点属性。布局模型最终由 XML 片段组成,由框架解析生成界面。

## 6 界面生成过程

基于模型的界面生成过程如图 7 所示,生成过程分为 4 个阶段。

第 1 阶段,需求分析阶段,主要任务将界面需求转化为任务模型和域模型。

第 2 阶段, 会话模型生成阶段, 框架映射转换引擎将任务模型和域模型转换成界面的抽象视图。

第 3 阶段, 显示模型生成阶段, 框架引擎根据抽象视图生成使用具体交互对象表示的表现模型, 使用编辑器编辑布局模型。

第 4 阶段, 代码生成阶段, 设置目标语言类型, 框架结合相应规则和算法生成可执行的界面代码。

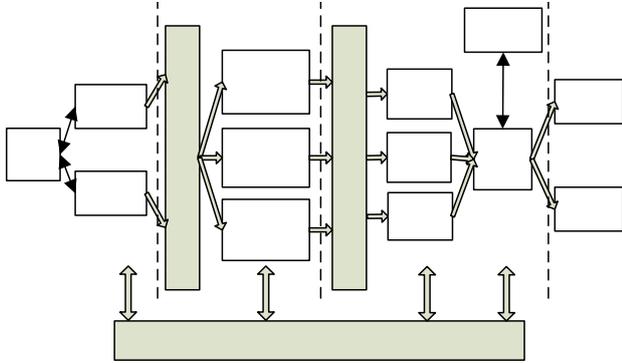


图 7 界面生成过程

## 7 结束语

本文分析了界面模型的特点和界面建模的技术, 提出了支持界面自动生成的 ADS 模型。模型从用户任务分解入手, 构建任务分析树, 给出了任务分析树到界面的抽象视图的转换规则, 由显示模型完成抽象视图到 UsiXML 文件的合成, 最终由框架解析生成界面。

建模  
工具

任务  
模型

转换  
映射

会话模型  
视图2

转换  
映射

表现  
模型1

表现  
模型2

编辑器

布局  
模型

编辑

布局  
模型

Java  
发生器

(上接第 78 页)

返回的记录形成一个完整且表述一致的记录集合。去重归并处理主要包括 2 步: (1)对 4 个 Deep Web 返回的数据记录按名称、资源作者、格式、大小排序; (2)对连续相同的 2 个记录进行比较, 若名称、资源作者、格式、大小都相同, 则认为是重复记录, 对其按来源进行区分, 以优先保留次序(新华社新闻资源库<K12 资源库<中央电教馆资源库<上海教育资源库)为规则删除重复记录。通过以上处理形成检索记录集并写入跨库检索记录数据库, 利用结果显示模块将查询结果(结果界面如图 5 所示)反馈给查询者。



图 5 跨库检索结果界面

## 参考文献

- [1] Lozano M D, Gonzalez P, Ramos I. User Interface Specification and Modeling in an Object Oriented Environment for Automatic Software Development[C]//Proc. the 34th International Conference on Technology of Object-oriented Languages and Systems. Santa Barbara, USA: [s. n.], 2000: 373-381.
- [2] Schlungbaum E. Model-based User Interface Software Tools: Current State of Declarative Models[R]. Graphics, Visualization and Usability Center, Georgia Institute of Technology, Tech. Rep.: #96-30, 1996.
- [3] 孙晓平, 郭腾冲. 基于 UML 的面向对象的图形用户界面设计模型[J]. 计算机科学, 2003, 30(5): 108-112.
- [4] Wan Jiancheng, Lu Xudong, Lu Lei. A Model of User Interface Design and Its Code Generation[C]//Proc. of IEEE International Conference on Information Reuse and Integration. Las Vegas, USA: [s. n.], 2007: 128-133.
- [5] Paternò F. Model-based Design and Evaluation of Interactive Application[M]. [S. l.]: Springer-Verlag, 1999.
- [6] Limbourg Q, Vanderdonckt J, Michotte B, et al. UsiXML: A Language Supporting Multi-path Development of User Interfaces[C]//Proc. of EHCI-DSVIS'04. Dordrecht, Holland: Kluwer Academics Publisher, 2005: 207-228.
- [7] Welie M. Patterns in Interaction Design[EB/OL]. (2007-02-04). <http://www.welie.com/pattern/>

会话模型  
视图1

会话模型  
视图2

会话模型  
视图3

转换  
映射

转换  
映射

结束语

以上系统环境: 服务器操作系统 Redhat Enterprise HTML  
客户端环境: JDK5.0 以上, 应用服务器 tomcat5.5. 发生器

本文设计充分利用了 Deep Web 集成技术发展的最新成果, 较好地解决海量教育资源的“信息孤岛”问题, 在实践中应用该思想开发了上海教育资源库的智能检索门户。随着 Deep Web 技术的发展, Web 数据库发现技术及查询结果归并去伪技术将取得新的突破, 构建一个从教育领域 Web 数据库自动发现、抽取模式自动生成、记录归并、返回结果集成处理等全程自动化处理的教育资源检索门户将逐步成为现实, 并进一步推动我国教育资源均衡发展战略的实施。

## 参考文献

- [1] 教育部基础教育课程教材发展中心. CELTS-42-2006 基础教育教学资源元数据应用规范[S]. 2006.
- [2] 徐和祥, 王鑫印, 王述云, 等. 基于知识的 Deep Web 集成环境变化处理的研究[J]. 软件学报, 2008, 19(2): 257-266.
- [3] Liu Wei, Meng Xiaofeng, Meng Weiyi. Vision-based Web Data Record Extraction[EB/OL]. (2006-06-30). <http://www.cs.binghamton.edu/~meng/pub.d/WebDBCamera.pdf>.
- [4] Weber R. HTTPClient 客户端编程工具包应用[EB/OL]. (2007-12-02). <http://wiki.apache.org/jakarta-httpclient/HttpClient>.

编辑 陈 晖