

# 面向 RIA 的系统重构模型

孙勇<sup>1</sup>, 林菲<sup>2</sup>

(1. 浙江交通职业技术学院信息学院, 杭州 311112; 2. 杭州电子科技大学软件工程学院, 杭州 310018)

**摘要:** 利用软件工程中的重构技术, 提出一种基于抽象数据视图的系统转换模型。采用抽象数据状态图与经典重构策略相结合的方式对系统进行重构, 通过映射方法平滑地将 Web 应用程序转换为富互联网应用(RIA)程序。实验结果表明, 该模型能够提升用户界面的质量与易用性, 降低设计与开发过程中的工作量。

**关键词:** 富互联网应用; 重构; 抽象数据视图; 滚动索引

## System Refactoring Model for RIA

SUN Yong<sup>1</sup>, LIN Fei<sup>2</sup>

(1. College of Information, Zhejiang Institute of Communications, Hangzhou 311112;  
2. College of Software Engineering, Hangzhou Dianzi University, Hangzhou 310018)

**【Abstract】** By the refactoring technology in software engineering, this paper proposes a system refactoring model based on abstract data view. It uses the method which combines abstract data view and classic refactoring stragy, and transforms Web applications into Rich Internet Application(RIA) program by reflection approach smoothly. Experimental results show that the model promotes the quality and usability for user interface, and reduces the workload of the process for design and development.

**【Key words】** Rich Internet Application(RIA); refactoring; abstract data view; scrolling index

### 1 概述

近年来, 随着各种富互联网应用(Rich Internet Application, RIA)技术及工具的兴起, Web 应用程序中的用户界面正由传统的超文本语义型向丰富多彩的桌面型转变。RIA 是集桌面应用程序的用户界面功能与 Web 应用程序的快速、低成本部署以及互动多媒体通信的实时快捷于一体的新一代网络应用程序, 它结合了声音、视频和实时对话的综合通信技术使 RIA 具有前所未有的网上用户体验。开发人员可以利用 AJAX<sup>[1]</sup>, SilverLight<sup>[2]</sup>等较为成熟的 RIA 工具很方便地开发出基于 RIA 的 Web 应用程序。但是, 现实世界中存在大量的已经被广泛应用的传统型 Web 程序, 如何将这些程序转换为 RIA 是许多 Web 设计、开发以及维护人员不得不面对的问题。而关于此类转换, 到目前为止计算机业界还没有一个统一的标准或方法。

在一些文献中提到将 Web 应用转换为 RIA 的方案。例如, RUX 模型法<sup>[3]</sup>将数据与业务逻辑从 Web 应用中提取出来, 并规定一组抽象的用户界面分别用于描述新 RIA 程序的结构和行为。文献[4]利用 AJAX 技术和一个半自动化工具将具有多个页面的 Web 应用程序转换为具有单一页面的 RIA 程序。用此类方法转换来的新 RIA 程序与原始程序差异很大, 使新系统不能完全替换老系统。

由于现实世界中大量传统 Web 应用的存在, 从头开始去搭建 RIA 应用较难实施。针对该问题, 利用重构技术<sup>[5]</sup>提出一种将传统 Web 应用重构为 RIA 应用的建模方法。该方法采用抽象数据状态图与经典重构策略相结合的方式对系统进行重构。通过该模型将 Web 应用以较小的代价平滑地过渡成为 RIA 应用。对该模型进行的实验结果表明该方法的实用性和有效性。

### 2 重构模型

#### 2.1 抽象数据视图

对于一个 Web 应用程序, 将用户界面上不同类型的对象进行分区, 把每一个区域称之为抽象数据视图, 抽象数据视图指定了当前状态下界面的外观与功能, 可以利用方法调用或事件驱动来执行抽象数据视图中的功能, 也可将多个抽象数据视图组成一个复合抽象数据视图来完成一个特定功能, 以实现代码复用, 例如较为常见的地图功能。

图 1 描述了网络视频播放器的抽象数据视图。为每个抽象数据视图规定了对象名称与数据类型, 例如, 代表播放按钮的抽象数据视图的对象名为 Play, 类型为 Button。



图 1 网络视频播放器的抽象数据视图

#### 2.2 抽象数据状态图

抽象数据视图模型化了用户界面的当前状态, 但它是一

**作者简介:** 孙勇(1977-), 男, 讲师、硕士, 主研方向: 软件工程; 林菲, 讲师、硕士研究生

**收稿日期:** 2009-06-13 **E-mail:** syman@zj.com

个静态模型，在真实的应用环境中，抽象数据视图之间会通过用户行为产生联系并使得用户界面发生转变。例如在图 1 中，当执行 Play 对象时，抽象数据视图 Play 被禁用并转换为 Pause，反之亦然，两者不会并存，它们用虚线分隔。为了描述这种动态行为，在抽象数据视图的基础上又定义了抽象数据状态图。

抽象数据状态图主要用于描述抽象数据视图之间的联系及传递的数据，与 UML 中的状态图较为相似。抽象数据状态图由一组状态转换组成，状态转换规定了抽象数据视图转换中应遵循的规则。每个状态转换规则由 3 个部分组成：(1)事件，触发状态转换的用户事件；(2)前置条件，用于规定事件触发后完成状态转换需满足的前提条件；(3)后置条件，用于规定结果状态。前置条件与后置条件用布尔型表达式实现，表达式的值可以通过对象方法结合“与”、“或”、“非”等逻辑运算获得。当产生事件时，如果规则集中的规则满足前置条件，则触发相应的状态转换，转换完成后更新后置条件。

图 2 表示图 1 中 Play 与 Pause 2 个抽象数据视图的抽象数据状态图，由 2 个转换规则组成。转换 1 当用户单击 Play 按钮时触发，转换 2 当用户单击 Pause 按钮时触发。假设 Focus()为判断当前对象是否为焦点的方法，Owner 为抽象数据视图的所有者对象。

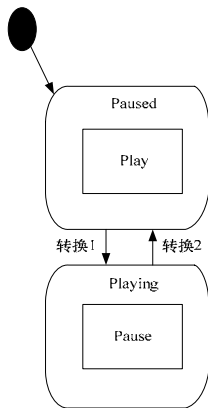


图 2 抽象数据状态

图 2 中箭头 1 描述如下：

事件：MouseClicked  
前置：Focus(Play)  
后置：Owner.IsPlaying()；  
箭头 2 描述如下：

事件：MouseClicked  
前置：Focus(Pause)  
后置：! Owner.IsPlaying()

### 2.3 重构策略

重构是在不改变软件现有功能的基础上，通过调整程序代码改善软件的质量、性能，使其程序的设计模式和架构更趋合理，提高软件的扩展性和维护性<sup>[5]</sup>。每个重构方案主要由名称、概要、动机、作法与范例 5 个部分构成，其中，动机与作法是重构中的重点，动机说明了重构的适用场合，作法简明扼要地分步骤介绍如何实施重构。

面向 RIA 系统重构的动机是将传统的 Web 应用程序以最小代价转换成 RIA 应用程序，重构的作法主要包括以下 4 个步骤：

(1)建立 Web 应用程序中需重构部分的抽象数据视图。

(2)建立 RIA 应用程序的抽象数据视图与抽象数据状态图。

(3)如果 RIA 是新增功能，则将 2 个抽象数据视图合并，还须指明新 RIA 的具体位置。如果 RIA 是替换功能，则将 Web 程序中相应的部分用 RIA 替换。

(4)通过模型映射工具将重构模型映射到接口实现中。

### 3 重构模型的应用

利用第 2 节介绍的重构模型可以将传统 Web 应用程序中的某部分转换为 RIA 应用程序，或添加新的 RIA 程序模块。本节中通过一个将 Web 页面中的列表转换为滚动索引的实例来阐述重构模型的应用方法。

#### 3.1 重构动机

传统电子商务网站通常将商品以线性列表的形式罗列，用户经常需要上下滚动整张网页以浏览各种商品。为了提高用户使用网站的舒适度和突出显示商品项，可以将商品列表以水平可滚动的方式显示，并在列表左右 2 端分别放置一个按钮，使列表左右滚动，当用户单击某个商品项时，将显示该商品的详细信息。通过该方式，用户在浏览商品目录时不必再使用浏览器的滚动条，同时也减少了商品列表占用的页面空间。新 RIA 程序的示例界面如图 3 所示。



图 3 滚动索引界面

#### 3.2 重构方法

滚动索引 RIA 重构模型如图 4 所示。假设 Items 为当前界面显示的商品项数组，长度为 3；AllItems 为所有商品项数组，长度为 Length；OFFSET 为 2 个数组首元素的偏移量。将该模型以替换的方式置入 Web 页面中商品列表的位置，即完成了滚动索引的重构。

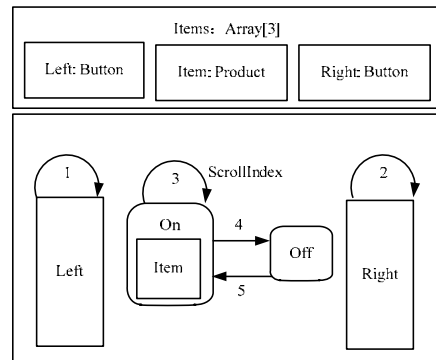


图 4 滚动索引重构模型

图 4 中箭头 1 描述如下：

事件：MouseClicked  
前置：Focus(Left)  
后置：IsShiftLeft()；  
箭头 2 描述如下：

事件：MouseClicked

前置：Focus(Right)  
 后置：IsShiftRight() ;  
 箭头 3 描述如下：  
 事件：MouseClicked  
 前置：Focus(Items[i])  
 后置：Items[i].Opened() ;  
 箭头 4 描述如下：  
 事件：LeftClick  
 前置：OFFSET>0 && ((∇ i)(0 i<3 && Items[i]==Owner.

AllItems[OFFSET+i]))  
 后置：((∇ i)(0 i<3 && Items[i]==Owner.AllItems [OFF-  
 SET+i-1])) ;  
 箭头 5 描述如下：  
 事件：RightClick  
 前置：OFFSET<Length-3 && ((∇ i)(0 i<3 && Items[i]==  
 Owner.AllItems[OFFSET+i]))  
 后置：((∇ i)(0 i<3 && Items[i]==Owner. AllItems[OFF-  
 SET+i+1]))

#### 4 模型映射方法

重构模型可以利用 XSL 模板技术<sup>[6]</sup>自动地映射为具体实现，本文主要研究模型设计，对于实现方法在本节中做简要阐述。

假设重构模型中抽象数据视图的外观与行为已分别由 HTML 与 JavaScript 实现。为每个重构模型编写一个 XSL 模板，通过该模板可以将重构模型以参数传递的形式来实例化。

重构模型实例化的过程如图 5 所示。

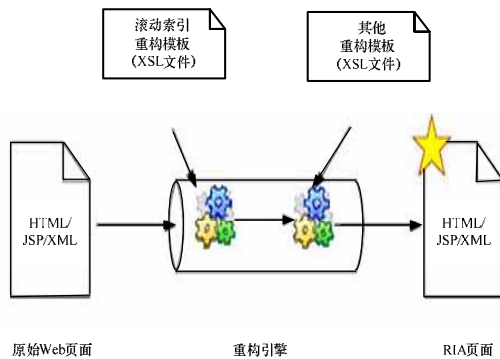


图 5 重构过程

对于在第 2 节提及的滚动索引重构模型的实例化具体步骤如下：

(1)提取原始 Web 页面中要重构的部分，示例 HTML 代码如下：

```
<html><body><table>
<s: iterator value="index"><tr>
<td><s: property value="item"/></td>
</s:iterator></table></body></html>
```

(2)编写滚动索引重构模板，示例 XSL 代码如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl: stylesheet version="1.0" xmlns:xsl=" "
```

```
<xsl: template name="ScrollIndex">
<!-- Left Button -->
<s: iterator value="index"><tr>
<xsl: call-template name="ItemView"/>
</s: iterator>
<!-- Right Button -->
</xsl: template></xsl: stylesheet>
```

在名为“ScrollIndex”的 XSL 模板中封装了滚动索引的实现，ItemView 为描述商品项水平列表视图模板的 XSL 文件。

(3)编写重构引擎，并自动生成 RIA 程序，示例 XSL 代码如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl: stylesheet version="1.0" xmlns:xsl=
"http://www.w3.org/1999/XSL/Transform">
<xsl: include href="ItemView.xsl">
<xsl: template match="//*[@id='index']">
<xsl: call-template name="ScrollIndex"/>
</xsl: template></xsl:stylesheet>
```

该模板文件利用 XPATH 表达式规定了标识名(id)为“index”的窗口部件将被名为“ScrollIndex”的模板替换。

#### 5 结束语

本文提出一种有效的将传统 Web 程序转换为 RIA 程序的系统建模及实现方法，系统建模主要基于软件工程领域中重构技术并结合文中提出的抽象数据视图概念，实现利用 XSL 技术通过对模型的映射，并结合一个滚动索引的实例加以阐述。

重构以标准化的形式记录了一些重要的设计经验，在今后的工作中，面向 RIA 的重构系统将进一步完善，将加入更多种类的重构模板，并考虑开发针对抽象数据视图与抽象数据状态图建模的软件工具。

#### 参考文献

- [1] Garrett J J. Ajax: A New Approach to Web Applications[EB/OL]. (2005-11-03). <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
- [2] 丁士锋. 精通 SILVERLIGHT: RIA 开发技术详解[M]. 北京: 人民邮电出版社, 2008.
- [3] Linaje M, Preciado J, Sanchez F F. Engineering Rich Internet Application User Interfaces over Legacy Web Models[J]. Internet Computing, 2007, 11(6): 53-59.
- [4] Mesbah A, Deursen V A. Migrating Multi-page Web Applications to Single-page AJAX Interfaces[C]//Proc. of the 11th European Conference on Software Maintenance and Reengineering. Washington D. C., USA: IEEE Computer Society, 2007: 181-190.
- [5] Fowler M. 重构——改善既有代码的设计[M]. 侯捷, 译. 北京: 中国电力出版社, 2003.
- [6] Khun Yee Fung. XSLT 精要——从 XML 到 HTML[M]. 北京: 清华大学出版社, 2002.

编辑 陆燕菲