

基于 Silverlight 的电子商务 RIA 平台的研究 与实现^①

Research and Implementation of an E-Commerce RIA Platform Based on Silverlight

程国雄 胡世清 (深圳大学 师范学院 广东 深圳 518060)

摘要: 利用富 Internet 应用(Rich Internet Application)技术设计电子商务系统, 可以解决传统电子商务应用平台在表现商品特点方面的不足。Silverlight 是富 Internet 媒体应用的开发平台和运行环境, 结合 WCF 技术, 可以方便地构建具有丰富用户体验的电子商务系统。提出了 RIA 体系架构, 根据 RIA 体系架构, 运用服务驱动的软件设计模式, 构建了电子商务 RIA 平台。

关键词: 富 Internet 应用 Silverlight WCF 服务 电子商务 即时通讯

1 引言

当前的电子商务平台, 一般采用基于 WEB 的 B/S 架构。这种 B/S 架构与传统 C/S 架构相比, B/S 架构易于开发和部署, 但其建立于 HTML 页面基础上的客户端应用程序, 限制了客户端对程序状态的控制能力和信息表现能力, 缺失了像 C/S 架构所能提供的用户体验。近来 Microsoft 推出的 Silverlight 技术, 用来构建 RIA^[1](Rich Internet Application 富 Internet 媒体应用)平台。以其提供与桌面应用程序一样的程序控制能力和丰富的页面表现能力, 结合传统 WEB 平台易于部署、跨平台的特点, 受到广大软件开发人员的关注。本文介绍电子商务 RIA 平台的结构、工作流程、客户端呈现、数据驱动机制等构建方法和技术特点。特别关注客户端控制与服务器数据提供方式的系统架构。

2 基于 Silverlight 的 RIA 体系架构

Silverlight 作为 RIA 的客户端运行时环境, 其核心表示层实现面向用户界面(UI)和用户交互的组件和服务: 包括用户输入、媒体播放、数字版权管理和数据绑定; 以及用于 Web 应用程序的轻量型 UI 控件, 包括矢量图形、文本、动画和图像。客户端应用程序

框架由程序标记语(XAML)实现界面布局和界面控件描述^[2], XAML 是一种声明性语言, 具有比 HTML 更加丰富的界面(包括 Button、Tree、图表、动画、数据绑定等)描述能力。客户端应用程序逻辑使用 CLR 语言(如 Visual Basic 或 C#)编写代码隐藏文件实现, 实现表示层和和业务层的分开, 达到界面设计^[3]和软件逻辑设计可由设计者(Designer)和开发者(Programmer)分别完成, 以提高软件的开发效率和代码的复用率。基于 Silverlight 技术设计实现的 RIA 系统的一般体系结构如图 1 所示。

Silverlight 客户端通过异步方式连接服务器, 服务器通过 WEB 服务的方式提供客户端所需要的数据。服务器分为应用服务层、业务层、数据访问层^[4]。应用服务层定义并实现访问接口, 提供内部到外部的数据格式转换, 实现客户端与数据业务的隔离, 保证业务访问安全; 业务层实现业务流程控制, 业务操作、业务实体定义等功能; 数据访问层实现数据库连接、数据库访问、数据操作等功能。

RIA 平台通过 Silverlight 客户端实现界面呈现和程序状态维持, 充分发挥客户机的性能; 通过异步方式^[5]与服务器交互, 解决页面因等待数据而造成的画

① 收稿时间:2009-04-02

面停滞。Silverlight 客户端可实现像桌面应用程序一样的拖放, 移动、局部刷新功能和效果。此外 RIA 平台采用三层体系结构, 可实现部署灵活、跨平台交换数据^[6]等特点。

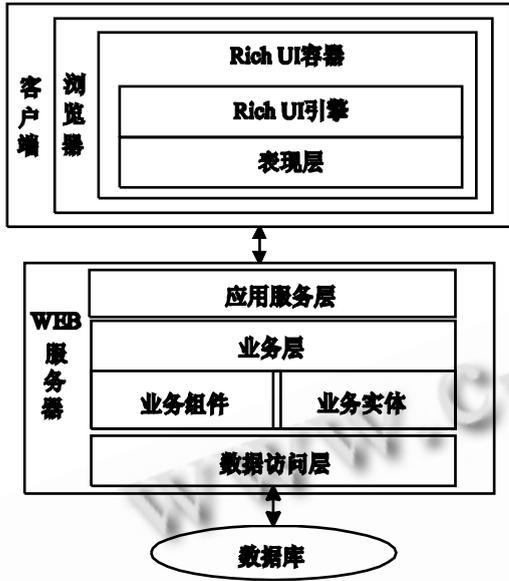


图 1 RIA 体系架构

电子商务^[7]RIA 平台按照 RIA 体系结构的特点进行结构设计, 分为应用层、业务层、数据层三级结构, 各部分之间通过接口实现数据格式转换和数据交换, 达到各逻辑结构之间的有效隔离, 有利于系统维护、数据安全、系统升级和部署。系统总体框架如图 2 所示。

3.2 系统构建

系统采用 Silverlight 和 WCF 构建服务驱动的应用程序, 采用 TCP 通信实现即时通讯服务, 体系结构如图 3 所示。

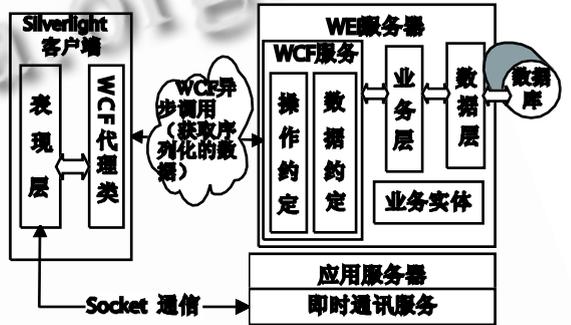


图 3 电子商务 RIA 平台体系结构

3 电子商务 RIA 平台设计

3.1 系统总体框架

3.2.1 服务驱动的软件设计模式

服务驱动的软件设计模式就是通过构建 WCF 服务, 公开可在表现层使用的业务逻辑、实体模型和数据映射代码, 表现层通过与 WCF 的通信实现与业务实体和数据库的交互。这种设计模式既实现 Silverlight 应用程序与数据实体之间隔离, 又使得 Silverlight 应用程序可以与后端业务程序无缝交互。构建 WCF 服务, 核心是要建立能与 Silverlight 应用程序通信的操作约定和数据约定。

(1) 数据约定实现: 以下是本平台 WCF 服务实现数据约定部分代码。Product 类具有 DataContract 属性, 并且它的所有属性都使用 DataMember 属性加以修饰。这将告知 WCF 开始序列化并将实体及其 DataMember 属性提供给 Silverlight 应用程序使用。

```
[DataContract]
public class Product
{ int _productID;
```

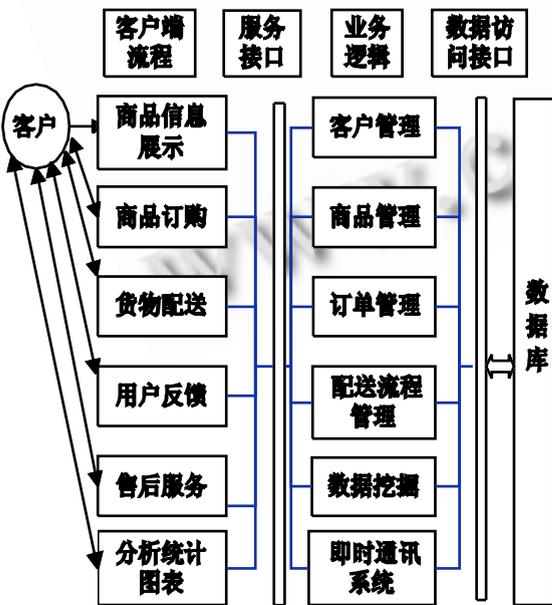


图 2 电子商务 RIA 平台总体框架

```

string _productName;
string _prouctphoto;
[DataMember]
public int _productID {
    get {return _productID; } set { _productID
= value;}
}
.....}

```

(2) 操作约定实现: 以下是本平台 WCF 服务实现操作约定部分代码, 操作约定包含服务及其服务方法, 服务使用 `ServiceContract` 属性进行修饰, 其方法使用 `OperationContract` 属性修饰。

```

ServiceContract(Namespace = "")]
public interface IProductService
{ [OperationContract]
    Product FindProduct(int productId);
    [OperationContract]
    List<Product> FindProductList();
    ..... }

```

(3) Silverlight 2 客户端应用程序可以添加对此服务的引用, 从而创建代理类。随后可以异步调用该服务。以下代码查询商品信息异步调用 WCF 服务的代码。

```

var proxy = new ProductServiceClient(); //
创建代理
proxy.FindProductCompleted += new
EventHandler<FindProductCompletedEvent
Args>( FindProductCompleted); //添加异步调用
结束时的事件处理程序
proxy. FindProductAsync(productId); //开
始异步调用

```

(4) 业务层: 主要实现对数据的有效性校验、处理定单等数据并将数据传送到数据访问层。实现订购、付款、送货、回执等商务流程和商务逻辑控制; 设计数据挖掘模式和算法, 完成从数据库中提取和分析数据。

(5) 数据层: 负责完成与数据库的数据交换, 以下代码是利用 LINQ To SQL^[8]实现商品信息数据收集

的方法。

```

public List<Product> FindProductList()
{ ProcutDataContext db = new
ProcutDataContext();
try { var products = from b in
db.Products orderby
    b.productID select b;
return products.ToList< Products>();
}
catch { return null; }
}

```

3.2.2 Silverlight 客户端设计

(1) 界面实现: 客户端采用 XAML 语言对界面布局和控件外观进行描述。Silverlight 提供了丰富的控件库, 这些控件支持用户界面 (UI) 开发, 主要包括可视化控件和容器控件。Silverlight 允许用户通过 `ControlTemplate`(控件模板)自定义现有控件的外观, 并且可通过 `ControlTemplate` 中添加 `Visual State` 对象实现控件可视化行为, 为控件添加动画效果。以下代码是商品展示界面的 XAML 描述。

```

<Grid x:Name="LayoutRoot" Opacity=
"0.96">
    <ListBox Grid.Column="0" x:Name=
"CategoryList1" />
    <ListBox x:Name=" ProductList2"
Style="{StaticResource ListboxStyle2}"
/>
    <StackPanel x:Name="ProductDetail" >
    <controls:Expander x:Name="Product
Select"
Header="当 前选择商品" >
<controls:Expander.Content>
<MultiScaleImage x:Name="dZObj"
Source="dzc_product.xml" /> //商品详细
信息
</controls:Expander.Content>
</controls:Expander >
</StackPanel>

```

</Grid>

(2) 数据模板: 数据模板(DataTemplate)定义控件中的动态数据项, 以数据绑定方式显示从服务器获得的数据, 数据绑定可以是单向的, 也可以是多向的。

```
<DataTemplate x:Key="DataTemplate2"> //数据模板
    <ContentControl Content="{Binding _prouctphoto }"/>
    <ContentControl Content="{Binding Path=_prouctCategory }"/>
    <ContentControl Content="{Binding Path=_product Name }"/>
</DataTemplate>
```

(3) 高清图像: 为了增强用户体验, 使用户对商品外观有比较清晰的了解, 当前的选中商品的图像显示用深度缩放(Deep Zoom)技术实现, 图像深度缩放是 Silverlight 中用来实现对任意大的图像提供高效缩放功能的一种技术, 它以很大或很小的比例显示图像, 但不影响图像的显示效率, 图像以渐进的方式加载, 解决高分辨率图像的传输和显示问题。MultiScaleImage 控件的 Source 属性指明图像集合源, 其形式如下。

```
<MultiScaleImage x:Name="dZObj"
    Source="dzc_product.xml" />
```

(4) 图表: 图表能够直观地表达数据信息, 由于 HTML 的缺陷, 传统的商务网站很难实现数据的图表透视。Silverlight 提供了丰富的图表控件, 对于用图表来表现商品销售过程中的各种信息, 提供了很好的解决方案。以下是建立商品销售情况图表的 XAML 代码。

```
<charting:Chart x:Name="TVChart"
    Title=" 平板电视销售情况表 " Style="{Static Resource ChartStyle1}" >
    <charting:ColumnSeries
        IndependentValueBinding="{Binding
```

```
Path=Year}"
        DependentValueBinding="{Binding
Path=Value}"
        Title="台数" />
</charting:Chart>
```

3.2.3 即时通讯系统的实现

Silverlight 的重要功能之一是支持 TCP 套接字的通信。由于 Silverlight 不能是侦听器, 因此它不能接受入站套接字连接。即时通讯系统将创建侦听端口 4530 的 Chat 服务器(套接字服务器), 并维护连接池, 其中每个连接代表一个用户。Silverlight 套接字运行时从策略服务器下载策略文件, 以实现对所有套接字连接实施跨域可选策略。即时通讯系统体系结构如图 4 所示, 其工作流程按 1 到 7 的顺序实现。

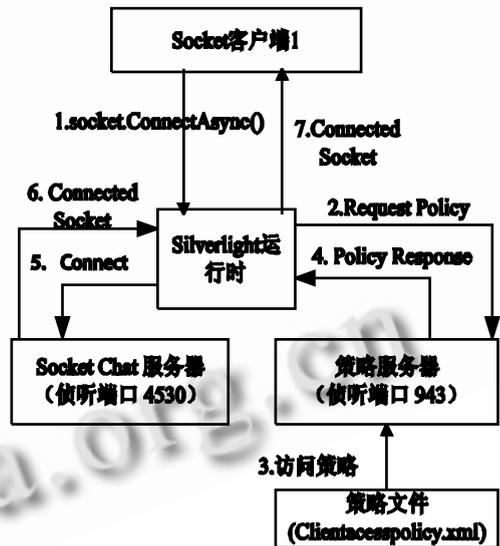


图 4 即时通讯系统体系结构

3.2.4 工作界面

图 5 展示的是电子商务 RIA 平台商品展示页面, 左边为商品分类列表; 用户选择相应类别后, 中部显示该类别的所有商品列表; 右边上部为已选择商品信息, 中部为当前选择的商品详细信息, 商品图片可以是分辨率较高的图片, 采用图像深度缩放技术实现其显示和传输, 可通过鼠标或键盘实现图片缩放和移动, 使客户能更好地了解商品的特征。下边窗口是与商品销售、受欢迎程度相关的图表信息, 使客户更直观获

取相关的商品信息。



图 5 电子商务 RIA 平台商品展示页面

4 结语

基于 .Net 的 Silverlight 开发和运行平台,具有跨系统、跨平台等特性,结合 XAML 语言强大的媒体描述能力和界面信息描述能力,由服务器保存并传输界面描述文件, Silverlight 运行时在客户端实现媒体表现,减少了信息传输流量,充分利用客户端运算和处理能力,提高网络带宽的利用效率。Silverlight 异步工作模式,结合 WCF 服务和 XAML 技术,可以很方

便地利用 RIA 技术开发电子商务系统。

参考文献

- 1 Steam B. A new approach for developing rich Internet applications. IEEE Internet Computing, 2007,11(3):67-73.
- 2 吕高旭. SILVERLIGHT/ASP.NET 与 AJAX 开发实战. 北京:清华大学出版社, 2008.
- 3 戴侃,杨小虎. 基于 J2EE 和 FLEX 技术构建 RIA 系统的探索与实现. 微电子学与计算机, 2006,23(5):22-24.
- 4 王非. RIA 系统研究与实现. 微计算机应用, 2008, 29(11):100-104.
- 5 Lawton G. New ways to build rich Internet applications. IEEE Computer, 2008,41(8):10-12.
- 6 Linaje M, Preciado JC, Sánchez-Figueroa F. Engineering rich Internet application user interfaces over legacy Web models. IEEE Internet Computing, 2007, 11(6):53-59.
- 7 鲍蓉. 基于 JADE 的智能电子商务系统设计. 计算机工程与设计, 2008,29(10):2561-2563.
- 8 侯利军. 精通 LINQ 数据访问技术——基于 C#. 北京:人民邮电出版社, 2008.