

# 基于 B/S 的跨平台用户界面可配置算法研究

赵双双<sup>a</sup>, 李仁旺<sup>b</sup>, 钱小燕<sup>b</sup>, 张春平<sup>b</sup>

(浙江理工大学, a. 信息学院; b. 机械与自动控制学院, 杭州 310018)

**摘 要:** 目前很多运营特定业务需求的网站间具有很大的模块相似度,针对这种情况,提出在不同软件平台下用户界面模块可配置功能以及模块和数据字典可复用算法。后台通过对平台、页面和模块基本信息进行管理,选择需要的平台、页面、模块、布局、样式模板后,利用拖拽的方式构成页面布局架构,永久存储页面布局数据。同时利用 CSS+DIV+AJAX+JSON+C# 构成模块组件、配置页面,并利用地址重定向技术重组前台页面。实现了一套程序多平台通用,使目标系统能够快速原型化,产生统一的界面风格,提高用户界面的开发效率,并可通过访问页面和模块编号这种分层交叉的处理方式分析页面价值提升流量。

**关键词:** B/S; C#; 可扩展标记语言; 组件; 地址重定向; 模块配置; 页面价值分析

**中图分类号:** TP3      **文献标志码:** A

## 0 引 言

随着计算机的不断普及和应用范围的不断扩展,网站建设在各个领域中已普遍存在。软件规模越大,其开发和维护的成本越大,传统的开发方式难以适应这一变化,目前很多网站间的模块相似度很高,传统的软件开发模式已经不能满足快速建站的需求,因此增加软件模块的复用程度,提高软件开发效率,也就成了软件工程中需要研究的问题。为了实现不同平台用户界面可配置功能,本文采用 CSS+DIV+AJAX+JSON 和地址重定向技术对平台下的页面模块进行分类裁剪、配置、重组形成特定平台。通过此方式可以将网站程序从项目级别发展为产品级别,也缩短了类似项目的开发周期。

## 1 功能设计

### 1.1 设计思想

本文主要采用基于构件的软件开发模型进行网站搭建算法的研究。

据分析和观察,很多商品类网站和门户型网站的界面布局都很相似。网页布局框架<sup>[1]</sup>主要分为:“同”

字型结构布局、“回”字型结构布局、“匡”字型结构布局、“厂”字型结构布局、自由式结构布局、“另类”结构布局。其中,“同”字型结构布局、“回”字型结构布局、“匡”字型结构布局、“厂”字型结构布局一般都具有明显的模块边界,因此本文采用了面向对象的程序模块复用技术,使得系统管理员可对平台、页面、模块进行管理,系统管理员可以在同一套代码下完成多个平台的搭建。系统多个平台,选择一个平台,添加自定义页面,选择一个页面,添加自定义模块生成初始化原型系统。

本文利用 SaaS<sup>[2]</sup> 中的用户界面可配置的设计思想进行网站框架的搭建。

SaaS 的用户界面可配置性包括两方面:

a) 外观的配置:管理员或者用户根据实际的需求和自身喜好决定页面风格。

b) 内容的可配置性:系统管理员或用户可以根据实际需求决定页面上放哪些内容模块。

SaaS 应用中的用户界面可定制满足多用户的需求,根据用户的实际要求显示不同的页面。

网页具有四大要素:布局、样式、内容、动作。其中布局、样式、内容模块可以通过本文功能设计进行自动化配置。后台主要通过 CSS、DIV 和 AJAX 技

术实现页面模块组件的构成和页面布局装配。前台读取后台的页面组装文件和数据库中页面配置信息来组装前台用户控件. ascx 文件, 并利用地址重定向技术来实现页面展现, 初始化原型系统。

同时也可以给访问的每个页面、每个模块添加参数(spm)用来记录平台和模块的访问流量, 根据 spm 值分析页面或者模块的访问量和页面价值<sup>[3]</sup>。

### 1.2 组件功能

目前, 基于组件的软件开发<sup>[4]</sup>尤其引起学术界和开发者的广泛关注。使用基于组件的软件开发方式可提高软件开发效率和软件质量, 充分考虑了组件的可复用性、反复性和易变性, 为软件开发起到了一定的引导作用。

组件, 又叫构件或部件, 是对数据和方法的简单封装。组件可以有自己的属性和方法<sup>[5]</sup>。使用组件可以实现拖放式编程、快速的属性处理以及真正的面向对象的设计。使用基于组件的软件开发模型是鼓励软件功能的复用, 形成可重用的构件库。

界面配置<sup>[6]</sup>是指系统运行以前, 通过可视化界面配置工具对应用系统的初始化界面进行配置, 从而实现对系统功能的裁剪、重组、扩充以实现特殊的业务需求。为了实现系统管理员对界面的配置, 必须将界面信息、组件对象属性和状态信息持久存储的应用程序之外, 在程序启动时, 从数据库中读取界面和组件信息并生成用户界面。

如果将界面配置信息存储在注册表内, 会因为系统的重新安装而造成数据丢失, 因此可以将用户界面信息永久保存在 DB 中或 XML 中。本文选择将界面配置信息存储在 DB 中。

### 1.3 AJAX

AJAX 指异步 Javascript 及 XML(Asynchronous

JavaScript and XML)<sup>[7]</sup>, 是一种运行在浏览器端的用于创建更好更快以及用户体验更强的 Web 应用程序的技术, 以实现异步刷新的效果。

### 1.4 JSON

JSON(JavaScript Object Notation)属于一种轻量级的数据交换格式<sup>[8]</sup>, JSON 采用完全独立于语言的文本格式, 此特性使得 JSON 成为理想的数据交换语言<sup>[9]</sup>。JSON 易于人阅读和便携, 同时也易于机器解析和生成。

JSON 是完全动态的, 允许在 JSON 结构中的中间改变表示数据的方式<sup>[10]</sup>, 在处理 JSON 格式的数据时, 没有需要遵守的预定义的约束<sup>[11]</sup>。

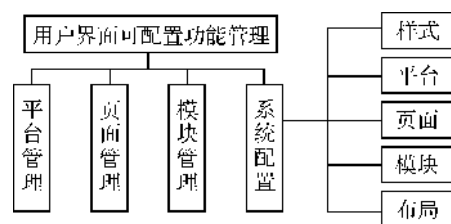


图1 系统功能框架

### 1.5 系统功能架构

### 1.6 设计工具和技术

本文算法研究基于 C# 的 B/S 体系结构。

使用开发工具: Microsoft Visual Studio 2010; 采用技术 CSS、DIV、JavaScript、C Sharp、Asp. Net 和地址重定向技术; 采用的数据库管理系统 SQL Server 2008; Web 服务器采用 IIS6.0。

### 1.7 数据库设计

基于以上设计思想, 该功能设计中具有 4 个实体: 平台、页面、模块、页面配置信息, 如下实体关系 E-R 图, 如图 2 所示。

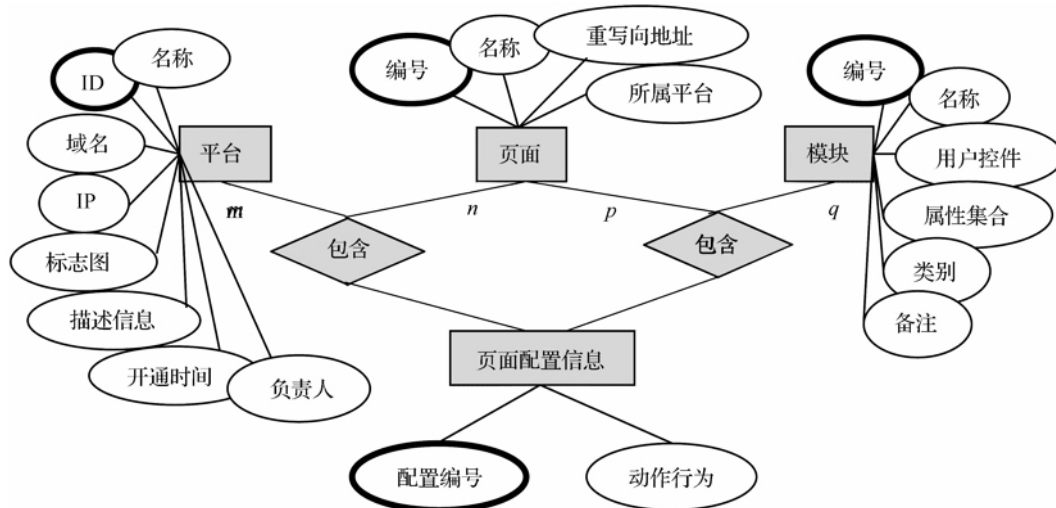


图2 E-R 概念设计

关系模式如下:

平台(平台编号,名称,域名,IP,是否启用,标志图,描述信息,开通时间,负责人,备注)

页面(页面编号,名称,重定向地址,所属平台)

模块(模块编号,名称,用户控件,属性集合,类别,备注)

页面配置信息(配置编号,平台,页面,显示模块,动作行为)

根据上面关系模式得出如下表 1—表 4 所示

(前面加 \* 字段表示主键)。

a) 平台信息表:记录平台/投放系统信息,如表 1 所示。

b) 页面信息表:记录各个平台的页面信息,如表 2 所示。

一个页面不论属于多少个平台只记录一条数据, P\_ID 为所属各个平台编号 power(2,P\_ID)之和。

c) 模块信息表:记录各个模块信息,如表 3 所示。

d) 配置表:记录每个平台页面模块的配置信息,如表 4 所示。

表 1 平台信息表

字段名称	说明	类型	长度	允许为空	约束
* P_ID	平台编号	int	10	No	主键,IDENTIFITY(1,1)
P_Name	名称	varchar	50	No	无
P_Domain	域名	varchar	50	Yes	无
P_IP	IP	varchar	50	Yes	IP 地址格式约束
P_Enabled	是否启用	int	1	No	默认约束,默认启用
P_SignLogo	标志图	varchar	100	Yes	无
P_Description	描述信息	varchar	100	Yes	无
P_Remark	备注	varchar	100	Yes	无
DevelopTime	开通时间	datetime		Yes	无
DevelopAuthor	负责人	varchar	50	Yes	无

表 2 页面信息表

字段名称	说明	类型	长度	允许为空	约束
* PP_ID	页面编号	int	10	No	主键,IDENTIFITY(1,1)
PP_Name	名称	varchar	50	No	无
PP_URL	重定向地址	varchar	250	No	无
P_ID	所属平台	Int	10	No	无

注:一个页面可能属于多个平台,减少了数据记录存储量。

表 3 模块信息表

字段名称	说明	类型	长度	允许为空	约束
* PM_ID	模块编号	int	10	No	主键,IDENTIFITY(1,1)
PM_Name	名称	varchar	50	No	无
UserContrUrl	用户控件	varchar	250	Yes	无
ContrProperty	属性集合	Varchar	1000	Yes	无
Module_TypeID	模块类别	Int	1	No	无
Remark	备注	Varchar	100	Yes	无

注:“ContrProperty”字段数据存储格式:{WordLimit: \$ 20 \$ ,RowCount: \$ 10 \$ }。

表 4 页面配置信息表

字段名称	说明	类型	长度	允许为空	约束
* Web_ID	配置编号	int	10	No	主键,IDENTIFITY(1,1)
P_ID	平台 ID	int	10	No	外键,平台表主键
PP_ID	页面 ID	int	10	No	外键,页面表主键
ShowModule	模块集合	varchar	200	No	无
Behaviour	动作行为	varchar	1000	No	无

注:ShowModule 字段数据存储格式:[1,2,3,4,5],每个数字代表模块的 ID 编号,模块编号和编号之间用逗号间隔,并用大括号括起来。

Behaviour:字段数据存储格式:

{ root: { ew \_ position:” widget-place-4 = identifierwidget-17 | widget-place-1 = identifierwidget-1, identifierwidget-2,identifierwidget-3,identifierwidget-5”, ew \_ layout:” widgetsStyletwo. css”, ew \_ stylesheet:”

BasenetwidgetsStyle. css”}},其中,ew\_position:标识每个框架中包含模块,其中 widget-place-4 表示框架布局,identifierwidget-1 表示显示的模块标识。ew \_ layout:选择的布局样式文件。ew\_stylesheet:选择的模板样式文件。

## 2 关键算法

根据图1系统功能框架,要组装成一个页面,需要选择平台、页面、页面布局、模块集合、样式模板基本信息。(注:以下代码部分为伪代码)

### 2.1 页面结构配置算法

在该系统中,系统后台管理员可以对架构的平台信息、页面信息、模块信息进行管理和维护。同时也可以对布局、样式进行个性化定制管理。

平台信息管理设计:具有平台管理权限的系统管理员可以添加、编辑、删除、查询同一代码框架下的多平台信息,平台信息属性主要包括:平台域名、IP、标识是否已经启用、平台Logo、平台开通时间、平台维护负责人、平台简介等。在程序设计中主要用到“表1平台信息表”。

页面信息管理设计:具有页面管理权限的系统管理员可以添加、编辑、删除、查询某一平台或者所有平台下的页面信息,页面信息属性主要包括:页面名称、重定向地址、所属平台编号。在程序设计中主要用到“表2页面信息表”。

模块信息管理设计:具有系统模块管理权限的系统管理员可以添加、编辑、删除、查询在同一代码框架下的所有模块信息,同时也可以对模块进行模型分类,主要包含文章模型、下载模型、图片模型、视频模型,可以自定义模块属性信息,比如对列表类别模块的显示数据条数、显示字数限制、前台显示标题、模块引用样式文件等属性信息进行自定义。模块的属性主要包括:模块名称、用户控件地址(.ascx文件)、控件属性集合、模块类别、模块描述。在程序设计中主要用到“表3模块信息表”。其中模块属性集合存储特定模块下的自定义属性信息,支持属性数量可扩展性。存储格式采用了JSON的数据格式,例如:{WordLimit: \$20 \$, RowCount: \$10 \$},其中WordLimit表示模块中显示文章的条数,RowCount表示每篇文章标题显示的字数限制。实现了前台显示的可定制功能。

#### 2.1.1 加载页面组件集合

加载页面组件时,首先从平台列表和页面列表中获取要选择的平台(ID)和页面(ID),清除相应页面的挂件并重新定位,并进行所有选择模块的初始化,形成挂件,并将挂件追加在选择的页面中,同时也要加载选择页面的布局样式文件和模板样式,利用JavaScript定义方法“ShowPlat”处理页面配置信息。主要算法如下:

```
function ShowPlat() {
    var isinit = true; var _wlayout = ""; var _wstyle = "";
    var _platID = $("#branchPortal").val();
    //分平台控件对象
    var _pageID = $("#portalPage").val(); //
    获取页面控件对象
    //请求处理数据文件
    var _url = "/BranchPlatManager/WidgetsHandler.ashx?rdm=" + Math.random(10000);
    $.ajax({
        //ajax 进行挂件定位和页面架构生成
        //ajax 处理,JSON 参数_platID 变量值和 _pageID 变量值; //请求页面变量 _url
        beforeSend: function() {
            //1. 选择页面后,清除所有挂件重新定位
            $("#widget-place-1").empty(); //布局框架区域 1
            $("#widget-place-2").empty(); //布局框架区域 2
            $("#widget-place-3").empty(); //布局框架区域 3
            //2. 所有点击的模块初始化
            //设置模块为选中状态
            $("div[name^=identifierwidget-][checked=true]").attr("checked", "false");
            $("div[name^=identifierwidget-]").removeClass("divclick"); //移除点击事件
        };
        success: function(data) {
            _wlayout = data.root.ew_layout; //获取布局文件
            _wstyle = data.root.ew_stylesheet; //获模板样式文件
            //3. 追加挂件
            $.each(data.ew_showlist, function(i) {
                AddWidget("WebGets.aspx?PageID=" + _pageID + "&index=" + i + "&WidgetsID=" + data.ew_showlist[i].ID, 'widget-place-1');
            });
            //4. 加载选择的布局样式文件
            RadioLayout(_wlayout);
        }
    });
}
```

### 2.1.2 模块列表标识

每个模块所在 div 标识形式:

① PM\_ID 表 3 字段 PM\_ID 值;

② PM\_Name 表 3 字段 PM\_Name 值;

③ 点击相应模块形成挂件,放在相应页面布局中。

SelectModel(par1,par2,par3)方法:

par1: identifierwidget-①,即选择模块的多个属性值;

par2: ②,即模块名称,作为挂件的标题;

par3: this,定位选中的模块 div 对象,用来标识选中 div,

```
<div checked="false" value="①" onclick="③"
name="identifierwidget-①" title="②" >③</div>.
```

说明:如果模块名称长度大于 6 截取只显示 6 个字符。

### 2.2.3 形成挂件原理算法

根据 SelectModel(par1,par2,par3),获取挂件的 ID 和标题,通过.ashx 后台程序处理。

挂件形成原理:

①根据 SelectModel 传递的值获取挂件的 ID 和标题;

②根据挂件的 ID 截取整数值读取表 3 ContrProperty 字段值,将其以冒号分割,显示挂件的各个属性和值,追加到选择的平台页面中。

## 2.2 前台组装算法

### 2.2.1 根据 URL 地址指定页面

用户在打开某个平台或者选择某个平台后根据 Url 重定向的地址读取表 2,然后根据选择平台 ID,页面 ID,读取表 4 页面配置信息表,获取特定平台下特定页面的模块组合前台布局和模板 CSS 文件组装成前台显示页面。

### 2.2.2 获取页面布局代码段

```
public void BindLayout(string MID, HtmlControl hc)
{ //参数 1:加载模块 ID,参数 2:展现布局负载体
//显示模块信息的 JSON 变量
string strShowModuleJson = string. Empty;
//显示页面整个配置信息变量
string strBehaviorJosn = string. Empty;
//根据条件平台 ID 和模块 ID 从表 4 中查询模块的
配置数据
赋值给变量 strShowModuleJson|
//声明脚本序列化对象
```

```
JavaScriptSerializer Serializer = new JavaScriptSerializer();
List<ucControlProperty1> CPList = new
List<ucControlProperty1>();
//获取控件或者模块的属性列表
Serializer. Deserialize< List< ucControlProperty1
>>>(strShowModuleJson);
//遍历模块属性信息,根据模块编号查询表 1-3 获
取控件的地址和模块字段 ucControlProperty 信息,
并通过方法 RanderControl 动态加载用户控件。
}
```

### 2.3.3 动态加载用户控件代码段

```
private void RanderControl(string ucIdentityID, string
ucName, IDictionary < string, string > _ dict,
HtmlControl hc)
{//参数 1:模块 ID,参数 2:模块名称;参数 3:控件
页根据参数 ucName 加载用户控件.ascx 控件,返回
UserControl 对象,并根据 ModuleID 值获取控件模
块的属性信息,并设置该控件的 ID 为“
identifierwidget-”+ ucIdentityID.代码如下:
PropertyInfo ptModuleID = ucType. GetProperty
("ModuleID");
ptModuleID. SetValue(ucControl, "identifierwidget-”
+ ucIdentityID, null);//获取模块属性信息
PropertyInfo rcProperty = ucType. GetProperty(temp);
rcProperty. SetValue(ucControl, _ dict[strKey].
Replace(" ", ","). Replace("¥", ":"), null);
hc. Controls. Add(ucControl);//向实际页面中追加控件
}
```

## 3 功能实现

高级管理员、系统管理员可以通过图 3—图 6 所示的界面配置用户界面,配置系统界面后,前台选择某平台,会根据后台配置显示前台页面。

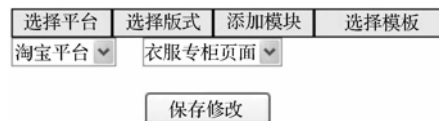


图3 选择平台和页面效果

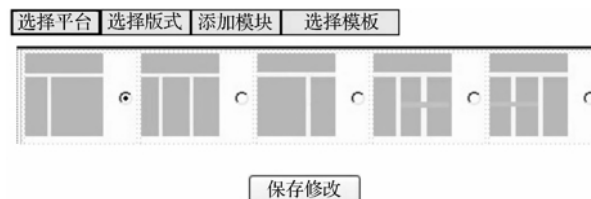


图4 选择版式效果

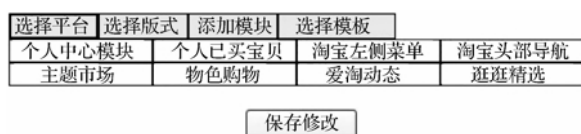


图5 选择模块效果



图6 选择模板效果

#### 4 结 语

本文采用基于C#语言的B/S体系结构,基于构件的信息类系统的设计模式,进行了多网站平台的用户界面可配置算法的研究,该算法设计使得多平台系统可以使用同一套程序,用来实现多网站架构的复用,将程序从项目级别升级为产品级别,大幅度增加了软件的生产率。解决了多数类似门户网站或者信息类网站的开发成本,主要减少了时间成本,同时也实现了软件的用户可定制性的特点。在该系统架构上,用户可添加自己业务需要的界面,不需要技术开发人员的参与。但不足之处在于,虽然本文实现了样式部分可定制功能,但目前算法只能获取到一些系统定义好的样式,软件管理

员不能直接在后台进行页面样式的个性化定制操作,非前端开发专业人士只能选择定义好的样式。

#### 参考文献:

- [1] 王 建. 精通 Web 标准建站[M]. 北京: 人民邮电出版社, 2007.
- [2] 及东兴. SaaS 平台构件方法研究[D]. 西安: 西安电子科技大学, 2010.
- [3] 梵 易. 页面价值分析: 2013 阿里技术嘉年华[EB/OL]. [2013-7-14]. <http://adc.alibabatech.org/carnival/history/schedule/2013/detail/main/25>.
- [4] 吴 坤. 基于组件的可配置用户界面管理系统[D]. 杭州: 浙江大学, 2005.
- [5] 百度百科. 组件[EB/OL]. [2013-6-25]. <http://baike.baidu.com/link?url=kMSbvKEhgfojhXGjswke3OHvaLB-Ak-sIOHkNND0LiRN7ZFdEXUkAcOL9QnNFWUYL>.
- [6] 王云鹏. 支持协同设计的 CAD 开发平台[D]. 北京: 北京航空航天大学, 2003.
- [7] W3School. AJAX 教程[EB/OL]. [2014-06-18]. <http://www.w3school.com.cn/ajax/2006>.
- [8] 韩义波, 宋 莉, 宋俊杰. Ajax 技术结合 XML 或者 JSON 的使用比较[J]. 电脑知识与技术, 2009, 5(1): 101-102.
- [9] 彭利章. 基于 SaaS 的通用后台内容管理[D]. 广州: 华南理工大学, 2012.
- [10] 高 静, 段会川. JSON 数据传输效率研究[J]. 计算机工程与设计, 2011, 32(7): 2267-2269.
- [11] 王小强, 程耕国. 基于 AJAX 和 JSON 的批量数据传递[J]. 软件导刊, 2010, 9(5): 187-188.

## Configurable Algorithm Research of B/S-Based Cross-Platform User Interface

ZHAO Shuang-shuang<sup>a</sup>, LI Ren-wang<sup>b</sup>, QIAN xiao-yan<sup>b</sup>, ZHANG Chun-ping<sup>b</sup>

(a. the School of Information Science and Technology;

b. Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**Abstract:** At present, many websites needed by special business have great module similarity. In allusion to such situation, configurable function of user interface module and reusable algorithm of data dictionary are proposed under different software platforms. The backstage selects necessary platform, page, module, layout and style template through managing basic information of platform, page and module. Layout framework forms through dragging method to permanently store layout data. Meanwhile, CSS+DIV+AJAX+JSON+C# forms module component and configuration page. Besides, URL redirection is utilized to reconstruct front-stage page. A set of procedures can be used by multiple platforms so that multi-platform can achieve rapid prototyping, generate uniform interface style and improve development efficiency of user interface. Besides, hierarchical crossing handling is used to analyze page value and improve network flow through visiting page and module numbering.

**Key words:** B/S; C#; XML; component; URL redirection; module configuration; page value analysis

(责任编辑: 陈和榜)