电子政务系统中 Ajax 应用的模式分析①

周燕霞

(浙江东方职业技术学院 工程技术系，温州 325011)

摘 要：以某机关事务管理局电子政务项目为例，对电子政务项目建设过程中的基于 Web 富客户端技术电子政务业务实现进行阐述。进一步分析了电子政务系统客户端的需求，结合 Ajax 的应用模式阐述客户端各功能模块的具体实现，从中对电子政务系统中富客户端的应用进行拓展和思维扩散，得出了一些常见模式的应用范围，以及总结出了在不同的应用需求下去选择最适用 Ajax 模式的方法。

关键词：富客户端；电子政务系统；Ajax 应用模式

Ajax Application Model in the Electronice Government

ZHOU Yan-Xia

(Engineering and Technological Department, Zhejiang Dongfang Vocational and Technological College, Wenzhou 325011, China)

Abstract: This paper presents the implementation of e-government base on Web rich client example of the government offices administration project. It analyses the system client's needs. According to these needs, it achieves the specific function modules on the client with Ajax application model, and gets the application of these common model, then sums up in a different application requirements on the best way to Ajax mode.

Key words: rich client; e-government; Ajax application mode

Ajax1 本质上是一个涉及一组技术的术语，包括 DHTML 和 XMLHTTPRequest 对象。DHTML 由三个元素组合而成，它们分别是 HTML、JavaScript 代码和级联样式表。在 Web 页面使用 JavaScript 代码，可以动态地改变页面，包括添加、删除或更改页面内容。JavaScript 代码使用 XMLHTTPRequest 对象在加载页面后向服务器请求数据。

在电子政务系统中要做的，就是充分利用 Ajax 的这一特性，来优化客户端的响应质量。从 Ajax 模式的角度进行分析，并对常用模式的适用性加以推广，从而可以对电子政务系统开发中“富客户端”应用的实现，做理论上的指导。

1 缓存控制器模式

缓存控制器模式为调用者提供了一种机制，用来以一致的方式临时保存资源。

电子政务系统是政府部门的信息化平台[2]，政府部门的工作特点，决定了内网办公系统中有许多文件类资源，需要系统进行有效的管理，而对大量信息进行管理最典型、最常见的是信息树[3]，而树结构就是缓存控制器模式一种的应用。如下图 1 左边的目录树，系统需要实现的效果是当点击信息树的节点发出一个请求的时候则读取该节点的下一级目录，而缓存中保留了上一级的目录，保证了已有的数据列表不会被重新加载，这样页面就实现了树的动态更新。并且从服务器端只返回了必要的信息给客户端，实现 Ajax 应用。

图 1 档案管理模块

在页面端通过 WebFXLoadTree()来实现动态树的

① 收稿时间:2010-12-30;收到修改稿时间:2011-02-14

结构，代码如下：

```
<script type="text/javascript">
var xLoadTree_comUser=new WebFXLoadTree("栏目 树 ","mutilModuleOperation Select.do?module Dir=/ne/portal&action=jsp?manager/columninfomation/ci_list.jsp?levelMark={levelMark}*columnName={columnName}&target=main","");
xLoadTree_comUser.target="main";
document.write(xLoadTree_comUser);
</script>
```

而 WebFXLoadTree()是通过 JS 文件对树结构进行封装，在其中单击节点时调用 OperationTreeDataCtrl.java 类，通过这个类，利用缓存一级一级地读取目录。用户在目录操作时，页面就会只刷新树中需要改变的结构，应用“富客户端”4 技术满足用户的良好操作体验。

在电子政务系统的信息管理模块中，应用 Ajax 的缓存控制器模式。缓存控制器模式可以看作是一个请求代理，由它来决定信息应该是直接从缓存中获取，还是应该发送一个请求到服务器端获取 5。本模式在以下环境中使用。

被动式缓存（passive caching）：创建一个被动式缓存的是保留一个已加载数据的列表，用来避免数据被不必要的重新加载。被动式缓存的一个例子是引用配置信息。配置信息在很大程度上是不会改变的，因此被认为是只读的。此外，配置信息不需要预先加载任何其他的数据。配置信息通常作为一个整体一次性地加载，而不是作为多个部分分别加载；

预见式缓存（predictive caching）：预见式缓存实现了被动式缓存的功能，此外还提供了一个附加功能：当发送一个请求时，与请求的信息相关的其他数据也被加载进来。预见式缓存的一个例子是 Google 地图应用。客户端发送一个请求来获取一个地图块。然后预见式缓存将使用一个算法来决定是否加载相关的地图块。重要的是该算法关系到用户有可能执行的操作。

在客户端，缓存控制器是一种最小限度的实现，它接收服务器端使用的一个参考数字，这个数字指示出是否有新的内容要被发送。

2 内容分块模式

内容分块模式使得增量地建造一个 HTML 页面成为可能，从而允许单个 HTML 页面的逻辑分布在不同的地点，由用户来决定内容的加载时间和逻辑。

内容分块模式在系统的实现中应用得最为广泛。如图 1 所示，页面的左侧负责显示信息树，右侧负责相关的信息条目刷新显示，两个模块各自负责自己的内容。在应用页面中使用下面方法来划分一个区域：

```
<script language="JavaScript" type="text/javascript">
var asynchronous = new Asynchronous();
asynchronous.complete=function(status,statusText,responseText,responseXML){eval(responseText);}
</script>
...
<body>
<button onclick="asynchronous.call('/chap03/chunkjs01.html')">Get Script</button>
<table>
  <tr><td id="insertplace"></td></tr>
</table>
</body>
```

要实现信息条目的实时刷新，在 Jsp 页面端使用 showGrid 方法调用 buffalo，这样在文件信息发生变化，或者需要查看不同目录的文件信息的时候，页面根据 infomationLinkManager.findAllInfomationList 方法只会动态更新显示文件列表的区域。

用 Ext 实现页面布局，系统会在页面载入时，通过相应的 js 文件对整个页面的布局和数据的处理进行控制。这样可以把控制与页面实现相分离。在页面的布局方式上 Ext 是通过如下的代码段实现的：

```
var configDesktopLayout = function(){
    desktopLayout.beginUpdate();
    desktopLayout.add("north", new Ext.ContentPanel("title-bar", "North"));
    desktopLayout.add("west", new Ext.ContentPanel("function-tree", {title: "West"}));
    desktopLayout.add("center", new Ext.ContentPanel("workspace", {
    title: "Workspace",
    fitToFrame:true,
    autoScroll:true,
    resizeEl:id_Workspace_iframe,
    closable: false
    }));
desktopLayout.endUpdate();  }
```

这样的页面布局可以在同一个页面中完成相关联的一系列操作，从而使得用户的思路更连贯，操作更

流畅。内容分块模式有着更多不同的适用环境：

当因为网站的性质而无法知道 HTML 页面看起来应该是什么样的时候。每个区域的内容是未知的，但是内容将会放在哪个区域中是已知的；

当将要下载的内容太大了，会导致用户过多的等待的时候。例如，执行一个搜索，等待将所有找到的元素收集为一个结果集，整个返回给用户，并不是可取的做法。一种更好的方法是在执行搜索的同时显示找到的部分元素；

当显示的内容之间彼此不相关的时候。Yahoo、MSN 等是门户应用，它们将一些彼此不相关的内容并列显示。如果内容是由单个 HTML 页面生成的，服务器端的逻辑不得不包含大量的条件判断，判断哪些内容需要加载，哪些内容不需要加载。一种更好的方法是将每一块内容当作一个分离的部分，然后分别加载。

与其它模式相比，内容分块模式是任何 Ajax 应用的核心模式 6。使用内容分块模式独特的地方是它总是遵循相同的处理阶段：生成事件、请求、响应、分块的注入。所以在客户端设计时，要首先从整体的角度考虑好客户端页面的内容分块布局。

3 持久通信模式

持久通信模式提供了一种机制，支持客户端和服务器在持久稳固的基础上进行通信。客户端能够发出消息到服务器，服务器也能发送消息到客户端。

在电子政务系统中也有对持久通信模式的简单应用，用户注册界面就是其中一个。通常用户需要填写很多个人注册信息，而各类注册信息都有一定的格式要求，每个网站又不尽相同。这就导致了用户注册一个“合法”身份，需要提交和修改很多次注册信息，从而造成了操作的繁琐，同时也加重了网络的负担。

现在有了 Ajax 技术，可以提供无刷新和异步提交的功能，就可以使得用户注册这类功能越来越人性化。

用一个功能点来说明 Ajax 的应用，注册时用户名必须是唯一的，所以这个字段必须和服务器端的数据库进行交互。为了不造成用户把所有信息填写完成，提交注册时服务器端才告知用户名已被使用的尴尬。页面可以在用户填写完用户名，用户名文本框失去焦点的时候让页面自动去服务器端验证用户名是否已经被注册，并反馈提示信息给用户，方便其及时更改。

在页面上的具体实现是通过一个 buffalo 方法来进行用户名验证：

```
var END_POINT="<c:url value='bfapp'/>";
var buffalo = new Buffalo(END_POINT);
function test() {  var p1 = $("tname").value;
  buffalo.remoteCall("arithmeticManager.test",[p1],
function(reply) {
    var res = reply.getResult();
    alert(res);});    }
```

而服务器端是通过 LoginTestManager.java 类的 test 方法来对比数据库，并返回验证信息。据此分析，只通过关键信息与服务器进行交互，并及时地提供服务器反馈信息，在客户端应用中是一种非常有效的方式，从而保持客户端与服务器端的良好通信。这种模式还有如下更多的适用性：

状态更新：一个状态更新是客户端感兴趣的一段全局信息，保存在服务器上。多个客户端看到的是数据的相同表现；

存在检测：当多个客户端访问相同的全局资源时，可以应用存在检测。除了资源的状态依赖于正在查看资源的客户端之外，全局资源为所有的客户端提供的表现都是相同的；

服务器推送：当多个客户端注册到一个全局资源上，但仅仅被分配了唯一的资源时，可以应用服务器推送。由服务器决定如何在已识别的用户与唯一资源之间建立交叉建立引用。

4 状态导航模式

状态导航模式提供了导航 HTML 内容的基础设施，当从一部分内容导航到另一部分内容时，状态维持不变。

在电子政务系统中，审批管理用于政府内部的办公和业务系统，如申请项目过程；文件、报告、合同的审批、传阅等需要流程化处理的业务。电子政务的内网办公系统根据事件的整个处理过程包含定义、发起、会签、延续、审批、跟踪、查询、归档、委托和数字查询等功能。

而行政审批模块主要依托于流程，所以单独的页面并不能完成业务流程的完整处理。以前电子政务系统的处理方式是一步步进行页面跳转，其缺点是用户的思路会被页面不断的跳转所打断，流程的运行过程也会被打乱，使得整个系统不具备良好的操纵性。

从客户端开始，状态导航层的目的是发送和获取状态。不使用状态导航层而绕过这个模式，会导致状态的破坏。通常实现状态导航模式的事件顺序如下：

①点击，检查，或者一些其他的 HTML 动作触发另一个 HTML 动作，产生一个 HTML 事件。

②HTML 事件可以是一个 HTMLpost 或按钮

onclick，这个事件创建了一个状态，客户端的状态导航层使用这个状态创建一个请求。

③这个请求被发送到服务器，服务器端的状态过滤器层可能处理或不处理这个请求，随后将请求传递给处理器。

④如果状态过滤器处理了请求，这个过滤是透明的，不会修改请求的内容。

⑤处理器生成包含一个链接的内容，状态导航模式使用这个链接加载下一个资源。

在这些客户端的事件顺序中，状态导航模式包含了 3 种职责：使用状态来组装表现；从表现中生成状态；并且在必要时重定向页面。

在系统的行政审批模块中，把流程定义在 xml 文件中，然后通过 XML 格式在客户端和服务器端来传递状态和引用信息。这样可以很好的控制一系列动作的状态。状态导航模式有着如下的适用性：

状态导航模式可以应用于可以编辑的状态与 HTML 页面相关联的情况中。在 HTML 表现表单中的状态必须不能被编辑，因为可以应用表现变形模式将静态表现转换为可编辑表现。

这里还存在着绑定状态和非绑定状态，非绑定状态是用来呈现绑定状态的。例如，与邮箱相关的非绑定状态可以指示出如何对显示在邮箱中的电子邮件进行排序。呈现电子邮件的资源是一个电子邮件列表，这个列表被认为是绑定状态。在没有绑定状态的情况下，非绑定状态可能会丢失。

5 无限数据模式

无限数据模式[7]用来从表面上无限的数据集中及时展现数据。该模式的思想是以增量的方式生成结果。

通过客户端页面多样化的布局，系统能够良好的满足行政审批模块复杂的交互性操作需求。而这些页面又能够通过 buffalo 与服务器端进行通信。

在页面中，页面的布局模式是通过 js 文件中的代码段来控制，而其他一些功能按钮类的页面实现则是通过 grid 来控制的。

当然，要实现数据的实时刷新，则要保证客户端界面与服务器的良好通信，通过 buffalo 来从服务器端获取数据：

```
var dataStore = new Ext.data.Store({
    proxy:new
    Buffalo.Ext.DataProxy(buffalo,"login.hello",[test],true),
    reader: myReader});
```

通过 buffalo 从服务器端获取数据，并且通过 dataStore 单元对这些数据的显示进行控制。

这些是通过 EXT2.0 的表单来满足用户所需要的数据交互，可以看出无限数据模式对大量数据信息的处理提供了一种良好的操作方式。无限数据模式有着较广的适用性：

查询或者操作非常庞大的数据集，当操作花费的时间比用户可以接受的时间更长的时候。

服务器端的操作去查询另一个生成不完整数据集的远程服务。

花费很长时间来执行的操作，可以划分为更小的操作，可以看作是一种智能猜测。

能够被转换为异步操作，允许应用在数据到达时生成结果。

6 结语

这些模式的应用并不是孤立的，各个模式之间都相互依赖，当系统设计人员从模式的角度考虑客户端实现时，必须要结合多个模式综合来解决问题。

通过对应用模式的分析，系统设计者们能够在电子政务系统客户端设计时，根据系统需求和技术情况，选择适当的模式，作为系统实现的指导，有效应用“富客户端”技术，丰富用户界面，满足用户需求。

参考文献

1 Garrett JJ. Rich Internet Applications and AJAX Selecting the best product. http://www.developer.com/design/article.php/3526681,2007.

2 王卫国,闾国年,王爱萍.电子政务系统.北京:科学出版社,2007.224–229.

3 孔敏.电子政务应用框架研究.南京:南京大学出版社,2006.

4 李子拓.Eclipse Rich Client Platform 简介.程序员,2005,(3).

5 Gross C. Ajax 模式与最佳实践.北京:电子工业出版社,2007.

6 Holzner S.Ajax 宝典.北京:人民邮电出版社,2007.

7 Cooper JW. Java 设计模式: a tutorial.北京:科学出版社,2004.