

# 企业级 AJAX 框架设计与实现

荣 耀<sup>1, 2</sup>, 李 昕<sup>3</sup>

(1 南京师范大学 中北学院, 江苏 南京 210046)

2 南京航空航天大学 信息科学与技术学院, 江苏 南京 210016

3. 国家电力公司南京电力自动化研究院, 江苏 南京 210003)

[摘要] 为满足企业级 Web 信息系统的要求, 充分利用网络资源和客户端资源, 扩展浏览器客户端与服务端交互的方案, 改善用户体验, 提出一个创新性的企业级 AJAX 框架 Magic Framework. 该框架采用面向对象特性扩充了 JavaScript 语言, 融合 XHTML 和 CSS DOM、XMLHttpRequest 以及 JavaScript 等现有技术, 创建一种新型的基于浏览器的丰富客户端 (Rich Client) 技术. 提供一套丰富客户端控件以及 3 种客户端与服务端之间的交互方式, 帮助开发者快捷地搭建企业级 AJAX Web 应用. 该框架已成功应用到大规模企业信息系统中, 其性能和带来的用户体验优于传统的 ASP.NET 模型.

[关键词] AJAX 框架, 企业级 Web 应用, 异步通信

[中图分类号] TP393 [文献标识码] B [文章编号] 1672-1292(2007)03-0064-06

## Design and Implementation of Enterprise AJAX Framework

Rong Yao<sup>1, 2</sup>, Li Xin<sup>3</sup>

(1 School of Zhongbei Nanjing Normal University, Nanjing 210046, China)

2 School of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

3 Nanjing Automation Research Institute, Nanjing 210003, China)

**Abstract** In order to meet the demands of enterprise web information system, use the resource of network and client sufficiently, extend the interaction between browser client and server and improve user experience, the paper presents Magic Framework which is a novel enterprise AJAX framework. Magic Framework extends JavaScript language with Object Oriented features and fuses XHTML and CSS Document Object Model XMLHttpRequest and JavaScript to develop a new browser-based rich client technique. The framework provides a set of rich client controls and three interacting ways between client and server. It is helpful for developers to build enterprise web applications more easily and quickly. The framework has been successfully applied to several large-scale enterprise information systems such as Nanjing Automation Research Institute PI3000. The performance and user experience of Magic Framework application have shown that its function is better than traditional ASP.NET models.

**Key words** AJAX framework, enterprise web application, asynchronous communication

## 0 引言

企业级信息管理应用最初采用 C/S (Client/Server, 客户机/服务器) 架构, 为用户提供了强大的交互手段, 但客户端计算任务过重, 部署、维护成本过高, 无法适应广域网计算环境. 随着 Internet 的发展和浏览器的普及, 基于 B/S (Browser/Server, 浏览器/服务器) 架构的 Web 应用和瘦客户端技术兴起, 只需一次性地更新服务器端程序, 即可通过浏览器部署应用, 省去了更新客户端程序的麻烦. 但是 B/S 架构下的 Web 应用仅仅提供了简单的交互元素, 用户界面失去了丰富的交互性, 频繁刷新网页也导致低品质的用户体验. 于是开发者尝试寻求兼具 C/S 和 B/S 优点的解决方案, 最具有代表性的是 Rich Client (丰富客户端). 但是 Rich Client 技术并没有被广泛接受, 原因之一在于不同的 Rich Client 技术需要额外安装不同的客户

收稿日期: 2006-03-19

基金项目: 信息产业部电子信息产业发展基金 (KY050401003) 资助项目.

作者简介: 荣耀 (1971-), 讲师, 博士研究生, 主要从事软件架构和软件工程等方面的教学与研究. E-mail: royao@royabo.com

端运行环境<sup>[1]</sup>. 新一代企业级应用是全面面向企业活动的动态 Web 应用, 需要处理大量的数据信息, 涉及到不同平台之间各种系统的数据传递、调用以及实时更新等问题. 新型企业级 Web 应用需要为用户提供流畅的交互体验, 拥有近乎桌面应用的界面品质, 可以实现数据的异步加载、页面的局部更新以及数据在应用之间的通用交换等功能<sup>[2]</sup>. 新兴的 AJAX 技术不但使 Web 应用具有 Rich Client 的丰富交互体验, 而且仍然是基于浏览器的 B/S 架构技术, 能够充分满足新一代企业级 Web 应用的需求.

1 AJAX 技术

AJAX 的全称是 Asynchronous JavaScript and XML, 即异步 JavaScript 和 XML (Extensible Markup Language 可扩展标记语言)<sup>[3]</sup>, 是一种基于浏览器的交互式网页应用技术. AJAX 采用 XHTML (Extensible Hypertext Markup Language 可扩展超文本标记语言) 和 CSS (Cascading Style Sheet 级联样式表) 呈现界面, 使用 DOM (Document Object Model 文档对象模型) 作动态显示, 使用 XMLHttpRequest 进行异步数据传输, 最后使用 JavaScript 将这些技术封装在一起<sup>[4]</sup>. 这些技术均非 AJAX 首创, 但 AJAX 将它们有机地融于一体, 最大限度地运用可用的 Web 浏览器技术, 使基于 B/S 架构的 Web 应用焕发出新的活力. 与传统的 Web 应用相比, AJAX 应用的新特性在于它可以利用 JavaScript 实现对网页中需要更新的部分进行局部更新. 客户端脚本在页面内部使用 XMLHttpRequest 以异步方式与服务器之间通信进行数据交换, 交换过程中页面保持原有状态, 无需中断用户操作. AJAX 还能够提供更丰富的交互元素, 用原先基本的 HTML 元素去模拟桌面应用中的各种复杂交互控件功能, 带来的用户体验往往胜过传统的 B/S 架构的应用<sup>[5]</sup>.

本文提出了一个基于 .NET 2.0 的企业级 AJAX 框架——Magix Framework. Magix Framework 被用作某企业 PB000 系统的前期开发基础框架和后期二次开发的平台. 与采用传统 ASP.NET 技术开发的旧版本系统相比, PB000 能够充分利用网络资源和客户端资源, 扩展浏览器客户端与服务端交互的方案, 实现 Web 应用与现有企业内部应用之间的无缝数据交换, 并提供更加丰富的用户交互体验.

2 Magix Framework 架构设计

Magix Framework 由客户端框架和服务端框架两部分构成, 分别采用 JavaScript 和 .NET Framework 实现, 大量的代码集中于客户端脚本, 服务端则负责客户端框架脚本的注册和 XML 序列化的工作.

图 1 展示的是 Magix Framework 的客户端框架结构图, 可以看出框架主要由 5 个部分组成: JavaScript 语言扩充、AJAX 工具、远程服务访问设施、客户端数据集以及丰富客户端控件库. 下面介绍这些组成部分所包含的重要技术成分.

2.1 JavaScript 语言的面向对象扩充

对 JavaScript 脚本语言进行面向对象的扩充是用它来开发框架级软件的前提. Magix Framework 为 JavaScript 提供了面向对象的特性 (如封装、继承、引用、命名空间、事件等)<sup>[6]</sup>, 并扩充了 JavaScript 的固有对象 (如字符串、日期时间、集合等), 使其更加接近 .NET Framework 的编程模型.

2.2 AJAX 工具

Magix Framework 对 AJAX 应用中常用的对象和方法进行了客户端的简化和封装, 引入了一系列方法来帮助 Web 开发者访问 HTML DOM 对象, 这些方法可以根据 ID 来获取指定的 HTML 元素, 还可以便捷地访问 HTML 元素的值、大小和位置等信息. Magix WebClient 类型则封装了 XMLHttpRequest 对象, 开发者可以用它更方便地在客户端同步或异步地访问服务端应用. 该对象还对 HTTP 参数进行了特殊处理, 可

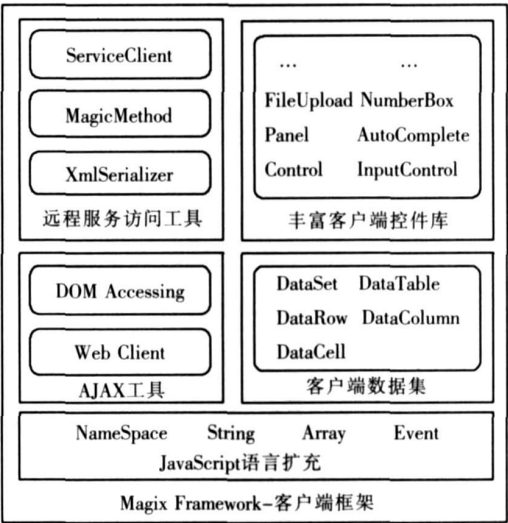


图 1 Magix Framework 客户端框架结构

Fig.1 The client side of Magix Framework

以自动将 JavaScript 中的 Number String Boolean Date JSON 等类型转化为“键值对”形式的 HTTP 参数.

2 3 客户端 XML 序列化

在服务端, 可以利用 .NET 的 Xml Serializer 对任何对象进行序列化. 在客户端, JavaScript 本身并没有提供对象序列化, 因此 Magix 客户端框架中提供了 Xml Serializer 对象, 可以对 Number Boolean String Date Array Object 等 JavaScript 固有类型和 DataSet 类型进行序列化和反序列化.

如图 2 所示, 首先运用服务端 .NET 提供的 Xml Serialier 将一个 C# 对象序列化, 然后在客户端通过 Magix Xml Seriali-zer 对象将服务端序列化后的 XML 结果进行反序列化, 映射到一个对应的 JavaScript 对象上<sup>[7]</sup>. 相反的过程与之类似. 这就做到了使服务端和客户端对象类型、结构和数据保持一致.

2 4 Web 访问

Magix Framework 针对企业应用的不同需求提供了 3 种与服务端交互的方式: WebClient ServiceClient 和 MagixMethod. Magix WebClient 是对 XML HTTP 的封装, 几乎可以访问 HTTP 协议所能访问的任何内容, 并支持简单的参数传递, 获取的服务端返回的结果则可以是普通文本也可以是 XML 数据. Web Service 以其能够在不同系统间无缝地进行通信和共享数据的优点得到越来越多的企业级应用的青睐<sup>[8]</sup>. Magix ServiceClient 允许用户直接调用系统暴露的 Web Service. Magix MagixMethod 是一种通过 XML 序列化实现的 RPC (Remote Procedure Call 远程过程调用) 技术, 可以直接将远程方法嵌入到 ASPX 页面中, 客户端脚本就像使用本地方法那样调用服务端方法.

3 种方案各有所长. WebClient 效率最高, 在客户端无需执行额外的 XML 解析和序列化操作. 当传递复杂数据或对象时, MagixMethod 更加合适. 如果需要与 J2EE 等平台上的应用交互, 则 ServiceClient 为最佳选择. 需要说明的是, WebClient ServiceClient 和 MagixMethod 都支持同步和异步两种远程交互方式. 这是基于这样的设计考虑: 并非所有 AJAX 应用场景都需要运用异步操作, 同步操作对于一些时序要求较高的应用是不可或缺的.

2 5 客户端数据集

在 ADO.NET 中, DataSet 是非常重要的核心组件, 它可以容纳多个数据表, 可以在这些数据表中临时存储数据以便在应用程序中使用, 并维护数据更改的信息. 它还支持 XML 序列化, 可以加载和输出 XML 格式数据. 利用 DataSet 可以实现离线数据库、简单数据查询、数据传输、数据增量保存等功能.

在 Magix.Data 命名空间中, 利用经过面向对象扩充的 JavaScript 和 XML DOM、XPath 等技术实现了一套数据集机制. 该命名空间中包含有 DataSet DataTable DataColumn DataRow 以及 DataCell 类型, 分别表示数据集、数据表、数据列、数据行和数据单元格. 如图 3 所示, 它们的结构、关系与 ADO.NET 中的 DataSet 数据集类似, 同样具备数据查询、维护更改信息、序列化等功能. 客户端 DataSet 主

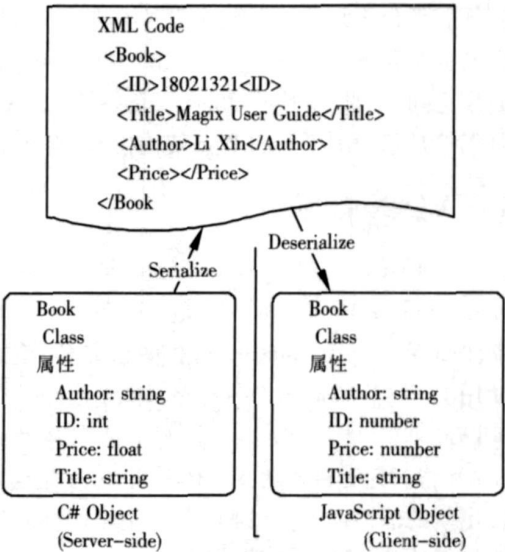


图 2 客户端和服务端的 XML 序列化  
Fig.2 Client XML serialization and server-side XML serialization

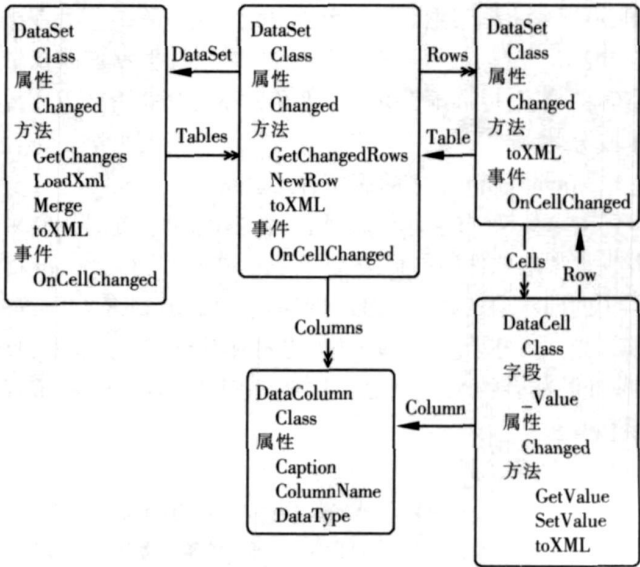


图 3 Magix.Data 命名空间主要成员  
Fig.3 Members of Magix.Data namespace

要用于访问 Web Service, Magix Method 调用和 Magix 控件的数据绑定。

2.6 丰富客户端控件编程模型

Magix 丰富客户端模型是 Magix Framework 中重要的创新技术之一,客户端控件对象可以实现网页局部更新,提供丰富交互体验,而服务端控件则可以有效地对客户端控件的容器和初始化等操作进行包装。

传统的 ASP.NET 模式最大的缺点在于开发人员只能在服务端获得控件对象,客户端只有 HTML 代码。以 ASP.NET 2.0 中的 TreeView 控件为例,它在服务端有非常完善的 API 和树形数据模型,可以对树的节点进行添加、删除、修改等操作,但是在客户端,它没有任何对象模型,只是由服务端控件生成的一堆 HTML、JavaScript 代码,开发者无法在客户端获取或操作节点对象。此外,如果想截获 TreeView 的切换节点事件 (OnNodeSelected),开发者不得不回送 (PostBack) 整个表单数据才能在服务端截获该事件,用户一般无法理解为何仅点击一下节点就需要刷新整个页面。

更好的解决方案是让客户端和服务端使用同一套控件模型编程,同时保证数据的一致性。Magix Framework 中提出了一种丰富控件编程模型:Magix 丰富客户端控件库。在客户端框架的 Magix Controls 命名空间提供了一套类似 ASP.NET 控件模型的丰富客户端控件编程模型,ASP.NET 开发者可以用熟悉的服务端控件模型开发客户端界面。

图 4 举例比较了传统服务端控件和 Magix 控件的运作方式。假设有一个菜单,当菜单项被选中时会做出不同的反应(由服务端决定做什么反应)。传统做法是在服务端编写一个 Menu 对象,在 ASPX 页面中对它进行实例化并初始化属性和事件,最终通过 ASP.NET 引擎呈现给客户端一段菜单的 HTML 代码。当用户选中菜单项后,客户端会将整个表单提交回送到服务端,然后服务端根据提交的菜单动作做出不同的反应,再将更新的整个页面的 HTML 代码传回客户端,客户端浏览器刷新后将结果呈现给用户。而 Magix 服务端控件则是在客户端生成一段 JavaScript 代码,用来初始化客户端的 Menu 类型对象 menu1,然后调用 Render 方法运用 DOM、CSS 等技术将控件呈现在客户端。当用户点击某个菜单选项时就会触发客户端 Menu 对象的 OnClick 事件,这时事件处理程序 menu1.OnClick 首先会在客户端检查是否需要更新,如果当前所选不是原来选中的菜单项,便会通过 XML HTTP 向远程服务器发送请求,然后将返回的数据更新在客户端界面上,最后调用 Render 方法自我更新。

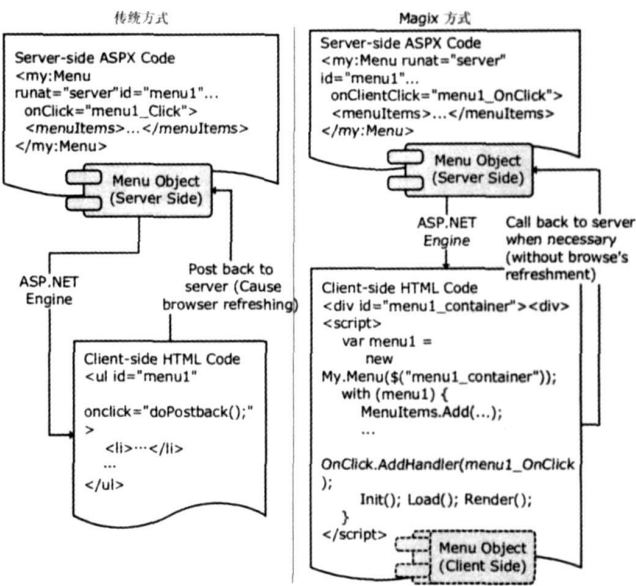


图 4 传统服务端控件与 Magix 控件运作方式比较  
Fig.4 Traditional server-side controls vs. Magix controls

3 Magix 企业级解决方案模型

现在描述怎样运用 Magix Framework 构建企业级 Web 应用。如图 5 所示,假定某企业当前已经拥有完善的大型数据库以及登录服务、邮件服务、短信服务等子系统,并建立了传统 Web 门户,它们都与企业应用服务相连,而应用服务则可以调用任何子系统所暴露出来的对外服务,从而实现各子系统间的互通。此外应用服务还将子系统的服务集成封装、统一接口,通过一个安全级别较高 Web Service Block (Web Service 阻塞)来发布所有可用的企业服务。企业内部应用,例如 CRM (Customer Relationship Management 客户关系管理)系统、ERP (Enterprise Resources Planning 企业资源计划)系统等,只需调用它就可以访问整个信息系统服务。现在企业需要建立一个基于 ASP.NET 2.0 的全新的动态 Web 门户网站,该网站面向企业内部和企业的客户,可以显示公司新闻,进行用户统一身份验证,收发公司内部邮件,并可以导航到各个子系统的 Web 站点。此外,可能还需要显示 ERP、CRM 系统中的数据信息,并做必要的操作。

传统的做法是,在 Visual Studio 中向网站添加特定的 Web Service Facade 引用,这样就会在当前项目下自动生成一个对应的 Web Service 代理<sup>[9]</sup>,然后在 ASPX 页面中就可以在本地调用该 Web 方法.如图 6 中左图所示,当用户提交表单时,会将整个表单回发到服务端,这时服务端再调用 WebService 取得下一条客户信息,并将这些信息绑定到控件上,然后将整个页面重新绘制以 HTML 的形式返回到客户端,浏览器再解析 HTML 并刷新当前页面.过分依赖服务器的后果是每当用户按下导航按钮页面都会整体刷新一次,由于应用需要经过 ASPX 页面才能调用 Service Facade 因而在速度和体验上往往让用户难以接受.

利用 Magic Framework,在客户端页面上可以直接调用 Web Service Facade 执行操作或获取数据对象,因此只要获得各业务的 Web Service 地址,就可以在客户端调用子系统内的业务服务.如图 6 中右图所示,可以采用 Magic 控件实现表单和导航栏,在客户端截获导航按钮单击事件.当用户按下按钮时,可以运用 ServiceClient 对象绕过 ASPX 页面,直接在客户端异步调用 Service Facade 取得一个包含客户信息的 DataSet 然后将新数据集绑定到表单的各个控件上,而无需整页刷新.使用 Magic 解决方案的另一个好处是数据缓冲.对于频繁更新的数据而言,可以在客户端建立一个 DataSet 类型的本地缓存,将每次浏览的数据合并到缓存中,从而当查看之前浏览过的信息时,可以使用本地缓存数据而无需调用远程服务,从而大幅提高系统运行速度.

由于 Web Service 一般使用 SOAP ( Simple Object Access Protocol 简单对象访问协议 ) 和 HTTP ( Hypertext Transfer Protocol 超文本传输协议 ) 进行访问,它最大的优越性在于拥有穿过防火墙的能力,但也带来了相应的安全隐患<sup>[10-11]</sup>.可以使用 SSL ( Secure Sockets Layer 安全套接层 ) 协议来保证 Web 通信的安全. SSL 是一种用于在不安全的网络上安全地传输信息的协议,可以建立用户与服务端之间的加密通信,确保所传递信息的安全性.

通过针对同一需求构建的两种不同实现方案的对比不难发现,与传统 B/S 架构相比, Magic Framework 充分利用了网络资源和客户端资源,实现了数据和界面的分离,改善了用户体验,最重要的是,它扩展了浏览器客户端与服务端交互的方案.

4 结束语及后续研究方向

实践证明, Magic Framework 提供的全新企业级 Web 应用解决方案可以大幅提高 Web AJAX 应用的开发效率,为用户带来更好的体验.该框架已成功运用于某企业的 P13000 系统之上. Magic Framework 在更多的方面存在发展的空间,将来框架服务端会延伸到 Java 平台上,并提供更多的丰富客户端 ( Rich Client ) 控件. Magic Framework 将运用于协同工作平台、基于浏览器的即时消息系统以及客户服务中心系统等领域,为更多的用户带来新体验.

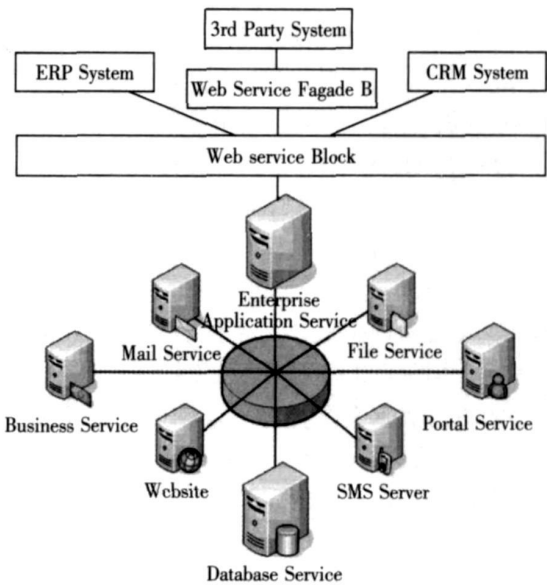


图 5 企业现有系统框架体系  
Fig.5 The architecture of existing enterprise system

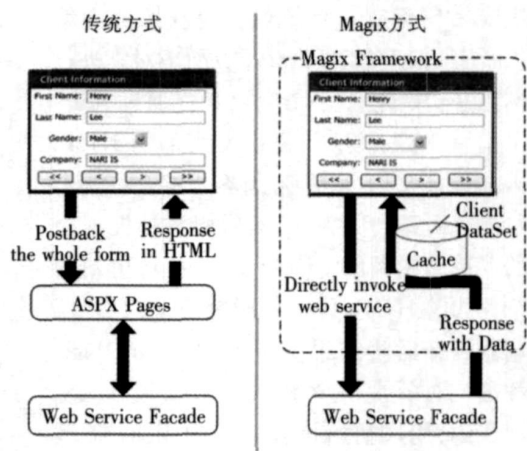


图 6 传统 B/S 架构方案与 Magic 解决方案对比  
Fig.6 Traditional B/S architecture vs. Magic Solution

# [参考文献] (References)

- [ 1] David Chappell . NET 大局观 [M]. 2版. 荣耀,译. 北京:电子工业出版社, 2006  
David Chapell Understanding NET[M]. 2nd ed Rong Yaq Translator Beijing Electronics Industry Press 2006 ( in Chinese)
- [ 2] 孔莉,郭曦,宋戈. 企业动态网站应用程序开发 [ J]. 工具技术, 2005 11( 39): 51-54  
Kong Li Guo Xi Song Ge Development of application program of enterprise dynamic website [ J]. Tool Engineering 2005 11( 39): 51-54 ( in Chinese)
- [ 3] 克拉恩,帕斯卡雷洛. AJAX 实战 [M]. Ajaxcn.org 译. 北京:人民邮电出版社, 2006  
Dave Crane Eric Pascarelbam es Ajax in Action[M]. Ajaxcn.org Translator Beijing Posts and Telecom Press 2006 ( in Chinese)
- [ 4] 杨国瑞,张思博. 基于 AJAX 的 Web 应用架构设计 [ J]. 现代电子技术, 2006( 15): 97-98  
Yang Guonui Zhang Sibao AJAX-based web application architecture design[ J]. Modern Electronics Technique 2006( 15): 97-98 ( in Chinese)
- [ 5] Bruno Eric J Ajax asynchronous JavaScript and XML[ J]. Dr Dobb's Journal 2006 31( 2): 32
- [ 6] 张军林,阳富民,胡贵荣. JavaScript 语言解释器的设计与实现 [ J]. 计算机工程与应用, 2003( 30): 124-125  
Zhang Junlin Yang Fumin Hu Guanrong The design and implementation of a javascript interpreter[ J]. Computer Engineering and Applications 2003( 30): 124-125 ( in Chinese)
- [ 7] 黄芳,孙建伶. 基于模式的 XML 文档元素的序列化标识 [ J]. 计算机工程与应用, 2004( 17): 181-183  
Huang Fang Sun Jianling Schema-based sequential identification for XML elements [ J]. Computer Engineering and Applications 2004( 17): 181-183 ( in Chinese)
- [ 8] 施伟斌,孙未未,施伯乐. XML 数据的结构化处理方法 [ J]. 计算机研究与发展, 2002 39( 7): 819-826  
Shi Weibin Sun Weiwei Shi Baile A method for structurization of XML data [ J]. Journal of Computer Research and Development 2002 39( 7): 819-826 ( in Chinese)
- [ 9] 叶军,李志蜀,殷锋,等. Web Services 在企业信息系统集成中应用架构的研究 [ J]. 计算机应用研究, 2007, 24( 6): 295-298  
Ye Jun Li Zhishu Yin Feng et al Research on enterprise information system integration architecture using web services[ J]. Application Research of Computers 2007, 24( 6): 295-298 ( in Chinese)
- [ 10] 祁涛,陈定方,张云如. Web 服务的安全性设计与实现 [ J]. 计算机与现代化, 2006( 1): 91-94  
Qi Tao Chen Dingfang Zhang Yunnu Design and realization of web service security[ J]. Computer and Modernization 2006 ( 1): 91-94 ( in Chinese)
- [ 11] 李海峰,杨小虎. 基于 RPC 风格与 Message 风格 SOAP 消息的 Web 服务性能比较 [ J]. 计算机应用与软件, 2007( 1): 79-80  
Li Haifeng Yang Xiaohu A performance comparison of SOAP RPC-style and Message-style web services[ J]. Computer Applications and Software 2004( 1): 79-80 ( in Chinese)

[责任编辑: 严海琳]