

基于 Web 服务的 Mashup 应用的研究与实现

高永兵, 吴纪磊, 胡文江, 魏晓东

(内蒙古科技大学 信息工程学院, 内蒙古 包头 014010)

摘要: Mashup 的出现使得开发更加方便, 随着越来越多的信息提供者公开自己的 API, 用户变成开发者加入到开发 Mashup 的队列中, 各种新型的 Mashup 应用在网上出现。Mashup 作为 Web2.0 的特性, 能够对数据资源进行整合与利用, 提升数据价值。在论述 Mashup 技术特点的基础上, 并讨论采用 Mashup 技术, 实现数据共享和应用开发。提出了基于 Web Service 的商品信息系统技术框架。通过案例研究验证了基于 Web Service 来构建 Mashup 应用的可行性。表明 Web Service 和 Mashup 技术在信息系统中具有巨大的应用潜力。

关键词: Web2.0; Web Service; Mashup

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2010)06-0137-04

Research and Implementation of Mashup Application Based on Web Service

GAO Yong-bing, WU Ji-lei, HU Wen-jiang, WEI Xiao-dong

(School of Information Engineering, Inner Mongolia University of Science & Technology,
Baotou 014010, China)

Abstract: With the emergence of Mashup, development becomes more easily, and more and more information providers publish their API, the user becomes Mashup developers to join the queue to the development of various new Mashup applications appear on the network. As an important character of Web2.0, Mashup technology can integrate data resources and enhance the value of data. Based on characters of this technology, discuss the use of the Mashup technology, data sharing and application development. It also gives a framework for information system technology based on Web Service. Case study shows that Web Service and Mashup technology have great potential in information systems applications.

Key words: Web2.0; Web Service; Mashup

0 引言

Mashup 应用是一个把从多个外部数据源检索到的内容来进行集成的 Web 应用, 当前已经得到广泛应用^[1], 它包括地图 Mashup、视频和图像 Mashup、搜索和购物 Mashup 等类型^[2]。目前, 大多数 Web 服务供应商, 如 Yahoo、eBay、Amazon、Google 都提供了 Web 服务 API^[3], 使得开发人员可以利用 API 来构建相应的 Mashup 应用, 使得这种新型的 Web 应用模式成为了现实。Mashup 又以其更多的重用性、基于 Web、轻量级的和面向终端用户的特点吸引着学术界和工业界^[4]。

文中采用 Amazon 和 Yahoo 提供的 Web Service API 来构建一个 Mashup 应用, 介绍基于 Web Service 的 Mashup 应用在信息共享和应用程序开发中的可行性及其应用潜力。

文章首先介绍相关技术, 然后以一个实例为例, 介绍 Mashup 应用的体系结构和实现过程。

1 相关技术

1.1 XSLT

XSLT^[5]在 1999 年 11 月 16 日成为 W3C 参考标准。XSLT 是 XSL 中最重要的部分, XSLT 是一种把 XML 文件转换成 XHTML 文档或者其他 XML 文档的语言。XSLT 用于将一种 XML 文档转换为另外一种 XML 文档, 或者可被浏览器识别的其他类型的文档, 比如 HTML 和 XHTML。通常, XSLT 是通过把每个 XML 元素转换为 (X)HTML 元素来完成这项工作

收稿日期: 2009-10-20; 修回日期: 2010-01-27

基金项目: 内蒙古自然科学基金(200711020813); 内蒙古自治区高等学校科学研究项目(NJC07109)

作者简介: 高永兵(1974-), 男, CCF 会员, 硕士生导师, 副教授, 研究方向为数据库与 XML。

的。在转换过程中, XSLT 使用 XPath 来定义源文档中可匹配一个或多个预定义模板的部分。一旦匹配被找到, XSLT 就会把源文档的匹配部分转换为结果文档。

1.2 Amazon Web Service (API)

Amazon Web Service (API) 是 Amazon 公司发布的一个强大的 Web 服务平台。Amazon 公司还允许客户通过评价和评估来扩充和丰富数据库中的内容。例如 Web Service, Amazon 是最早把自己的数据公开的几家公司之一。

目前 Amazon 已经具有了一套丰富的网络接口, 这些数据接口以免费或者按件计费的方式提供, 使得开发人员可以用来创建自己的应用程序。这些都为用户使用 Amazon 提供的 Web Service 来构建 Mashup 带来了便利。但是在调用 Amazon Web 服务平台中的方法时, 则必须参加开发人员计划, 获得开发人员 ID, 才可以使用该 ID 调用每一个 API 的功能。

1.3 Yahoo 搜索 API

Yahoo (<http://www.yahoo.com>) 支持与 Mashup 和 Web 服务有关的开发人员社区已经很长时间了, 除了核心的网络搜索 API, Yahoo 还发布了许多有用的和创新的 Web 服务。

搜索 API 也许是 Yahoo 提供的最核心的 Web 服务, Yahoo 搜索服务可以提供强大的搜索功能, 并且已经扩展到了传统的基于文字的搜索功能之外。Yahoo 购物服务是 Yahoo 提供的基于 REST 风格的 Web 服务之一。Yahoo 从大量的零售商收集商品信息, 并在 Yahoo.com 发布这些数据。通过 Yahoo 购物服务的商品搜索, 获取某个关键字的商品信息。

Yahoo 搜索包含以下几个特性: 音频搜索、新闻搜索、视频搜索、图像搜索和购物搜索等, 每个特性都提供了强大的功能。开发人员可以注册 ID 来方便地使用 Yahoo 发布的 Web Service API。

1.4 Web Service

Web Service 是一些独立的、模块化的应用程序模块, 可以通过 Internet 来描述、发布、定位和调用。它的目的是在现有各种异构平台上构筑一个通用的与平台无关、语言无关的技术层, 从而保证不同平台的应用服务可互操作。Web 服务建立在 XML、WSDL、SOAP 和 UDDI 等协议和标准之上, 具有完好的封装性、松耦合、使用协约的规范性、使用标准协议规范、高度可集成能力等特点^[6]。

随着 Web Service 技术的发展, 可以更多地采用 Web Service 方式提供数据等信息服务, 来支持灵活多样的应用开发。

2 Mashup 技术

2.1 Mashup 技术介绍

Mashup 作为一种新型的基于 Web 的数据集成应用程序正在 Internet 上逐渐兴起。Mashup 技术起源于流行音乐, 从两首不同的歌曲中混合演唱和乐器的音轨而构成的一首新歌。根据 Wikimedia 解释, Mashup 是将多个不同的支持 Web API 的应用进行堆叠而形成的新型 Web 服务。它利用了从外部数据源检索到的内容来创建全新的创新服务, 将来自不止一个数据源的内容进行组合, 创造出更加增值的服务。随着越来越多的 Web 站点公开自己的 API, 利用 e-Bay、Amazon、Google 和 Yahoo APIs 构建 Mashup 的应用不断出现, 使得这种新型的 Web 应用模式成为现实^[7]。

Mashup 的应用结构, 通常, 一个 Mashup 应用程序由三个级别或层次构成^[8]。Mashup 中所使用的服务或数据来源为第一层。根据应用程序的要求, 这个层中可能包括一个或几个资源, 直接在应用程序 (如 Blogger) 中输入数据的用户就是一个常见的数据来源。

第二层即将这些资源组合起来的 Mashup 应用程序。其中可能包含它自己的功能, 或简单呈现数据来源。

最后一层为用于访问网站的用户客户端。客户端脚本可用来改善用户体验。

来自 IBM DeveloperWorks 中国的描述, 一个 Mashup 的应用可以分为三个不同的组成部分: API/内容提供者, Mashup 站点和 Mashup 客户端 (也就是用户的浏览器)。

2.2 Mashup 技术与传统数据集成系统的区别

随着信息技术的快速发展, 集成信息越来越得到重视, 如企业信息化、EAI (企业应用集成) 等。在集成信息方面, 采用传统的数据集成方案和基于 Web2.0 的 Mashup 技术之间具有很多的不同^[9]:

(1) 对平台与系统的要求。基于传统的系统集成方案需要传统平台 API 层的支持, 如 J2EE 平台和 .NET 平台, 集成过程十分复杂。Mashup 技术构建应用程序, 集成过程相对简单。可以轻松地聚合和重组来自不同地方的内容, 包括服务和数据, 进行业务敏捷开发, 快速满足他们因情景而不断变化的需求。

(2) 对新特性的支持与功能扩展方面。基于传统的数据集成方案需要采用编程的方式完成系统功能扩展, 涉及表示层、业务逻辑层和数据存储层三层或多层结构, 可扩展性差; 而基于 Mashup 技术采用松散耦合的方式来集成信息以创建新的应用和功能, 由于是松耦合的方式, 极大地提高了系统的兼容性和可扩展性。

(3)表示层支持。传统的系统集成方案在传统集成方式下的表现层大多采用图形用户界面 GUI,当客户端发送请求时,需要加载整个页面;基于 Web2.0 的 Mashup 技术使用 Ajax 技术调用 Web 应用程序,浏览器客户端不需要在每次与服务器通信时都重新加载整个页面,动态性强。

(4)底层集成技术支持。基于传统的数据集成方案使用多层企业应用集成技术,涉及不同的集成层次,如业务过程集成、应用集成和数据集成等;基于 Web2.0 的 Mashup 技术采用 SOA(面向服务架构)思想集成底层系统,强调功能的暴露与服务的组合,以 Web Service 形式包装并暴露统一格式的接口和现有系统的能力,这样的集成系统具有模块化、松耦合的特点,并且可扩展性较好。

(5)进行内容聚合的物理位置。传统的 Web 应用程序是基于 HTML 页面、服务器端数据传递的模式,而 HTML 是适合于文本的,随着 Web 应用程序复杂性越来越高,传统 Web 应用程序渐渐不能满足 Web 浏览器更高的、全方位的体验要求,传统的 Web 应用程序往往存在操作复杂性、数据复杂性、交互复杂性等缺点。目前很多 Mashup 站点选择在客户端机器上进行内容聚合,构成所谓的胖互联网应用程序^[10](RIA),RIA 意味着丰富的数据模型和丰富的数据界面,丰富的数据意味着客户端的用户界面能表现和应对更复杂的数据模式,客户端允许用户构建一个高的响应、交互式的应用程序;在客户端进行内容的聚合,数据能够缓存在客户端,从而实现一个比基于 HTML 的响应速度更快且数据往返于服务器的次数更少的用户界面,在无刷新页面之下提供快捷的界面响应时间。

3 基于 Web 服务的 Mashup 案例

Mashup 解决方案的不同之处在于用户角色和用于创建独特价值的聚合的数据集。在本系统中采用 Amazon 和 Yahoo 提供的 Web Service API,并采用 Mashup 技术来创建一个 Mashup 应用,提高数据的共享性和应用的价值。

在此 Mashup 案例^[11]中,将会创建一个搜索界面在一个页面中同时显示从 Yahoo 和 Amazon 中搜索商品的信息。这两种数据源会被合并在一个 XML 文档中。最后将其转化成标准的 XHTML 插入到页面中显示搜索的结果。这样商品的购买者就可以很容易获得商品信息的比较,从而有利于找到物美价廉的商品,做出购买决策。

3.1 系统架构

系统架构如图 1 所示。

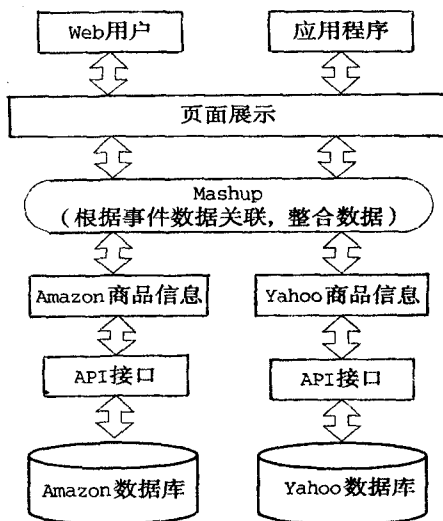


图 1 系统架构

3.2 工作流程

第 1 步:Web 用户提交请求,通过页面提交表单到 Mashup Web 服务器。

第 2 步:Mashup Web 服务器处理请求,分别向 Amazon API 和 Yahoo API 发出 HTTP 请求。

第 3 步:Amazon 和 Yahoo 分别响应 XML 的格式的结果给 Mashup Web 服务器。

第 4 步:Mashup Web 服务器把这 2 个 XML 混合到一个 XML 文档中,并且使用 XSL T 将这个文档转化为一段 XHTML 代码,然后把这段 XHTML 代码插入到此网页中。

第 5 步:服务器再次呈现这个网页,但仅包含了 2 个结果内容。

3.3 案例实现

可以看出,基于 Web 服务的 Mashup 开发的优势非常明显,在此 Mashup 实例中涉及到 Amazon 和 Yahoo API,调用 Web 服务 API 返回 XML 文档,然后把 XML 文档混合在一起使用 XSLT 构成一种 HTML 复合界面显示。

本地服务器关键代码片段如下:

(1) 调用远程服务器的 Web 服务并返回 XmlDocument 对象:

```
HttpRequest myRequest = (HttpRequest)
WebRequest.Create(strURI);
```

```
HttpResponse myResponse = (HttpResponse)
myRequest.GetResponse();
```

```
Stream myResponseStream = myResponse.GetResponseStream();
```

```
XmlDocument myDoc = new XmlDocument();
```

```
myDoc.Load(myResponseStream);
```

```
return myDoc.
```

(2) 获得来自 Amazon 和 Yahoo 的数据将其合并到一个新的文档 myResults:

```
XmlDocument myAmazonDoc = webUtility. GetUri ("http://webservices. amazon. com/onca/xml? Service = AWSECommerceService & AWSAccess KeyId = AKIAI7N2BKWZF2LCISTA & Operation = ItemSearch & ResponseGroup = Medium & AssociateTag = jileiwu & SearchIndex = " + strSearchIndex + "&Keywords = " + strKeywords);
```

```
XmlDocument myYahooDoc = webUtility. GetUri ("http://api. shopping. yahoo. com/ShoppingService/V1/product-Search? appid = jileiwu&query = " + strKeywords + "&results = 10&start = 1");
```

```
XmlDocument myResults = new XmlDocument();
```

```
XmlNode myRoot = myResults. CreateElement ("MashupRoot");
```

```
myResults. AppendChild(myRoot);
```

```
XmlNode amazonTmp = myResults. ImportNode(myAmazonDoc. DocumentElement, true);
```

```
myResults. DocumentElement. AppendChild (amazonTmp);
```

```
XmlNode yahooTmp = myResults. ImportNode(myYahooDoc. DocumentElement, true);
```

```
myResults. DocumentElement. AppendChild (yahooTmp);
```

(3) 将(2)返回的新的 XML 文档通过转换函数 DoXSLTransformation() 用表转换成 HTML (转换模板比较复杂, 因此没有列出)。核心代码如下:

```
XslCompiledTransform myProcessor = new XslCompiledTransform();
```

```
myProcessor. Load(System. Web. HttpContext. Current. Server. MapPath(strXsl));
```

```
StringWriter myWriter = new StringWriter();
```

```
myProcessor. Transform ( myDoc, ( XsltArgumentList) null, myWriter);
```

```
return myWriter. ToString();
```

3.4 案例运行界面

在搜索界面文本框中输入搜索关键字 (如 Harry Potter), 并在下拉列表中选择搜索索引 (如 DVD), 便可以得到图 2 所示的结果。从图 2 的结果可以看出左列 Amazon 和右列 Yahoo 得到的同类商品价格的最低

价格是有差别的, 从商品购买者的角度来看, 方便了他们比较商品信息, 做出购买决策。

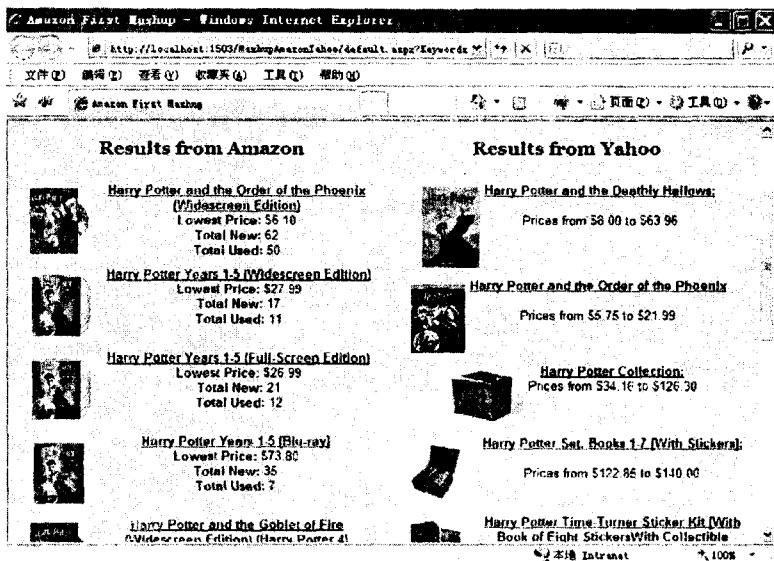


图 2 搜索结果

4 结束语

文中基于 Web 服务构建 Mashup 应用, 并且使用轻量级的技术构建在不同的数据源上, 区别于以往的以数据库为中心的开发方式。这种方法构建 Mashup 应用相对简单, 实际的优势也很明显, 能够有效地实现数据共享, 提升数据价值。采用这种新的技术解决问题, 可以发现如何访问一些数据, 并把这些数据和其他数据集成在一起创建一个新的、有趣的 Internet 应用。

参考文献:

- [1] Jackson C, Wang H. Subspace: Secure cross-domain communication for web Mashups[C]//Proceedings of the 16th International Conference on World Wide Web WWW'07. New York: ACM, 2007: 611-620.
- [2] Merrill D. Mashups: Web 应用程序新成员[EB/OL]. 2006-08. <http://www.ibm.com/developerworks/cn/xml/x-mashups.html>.
- [3] 李峰, 李春旺. Mashup 关键技术研究[J]. 现代图书情报技术, 2009(1): 44-49.
- [4] Liu Xuanzhe, Hui Yi, Sun Wei, et al. Towards Service Composition Based on Mashup[C]//2007. IEEE Congress on Services. [s.l.]: [s.n.], 2007: 332-339.
- [5] 郭小明, 雷电. 基于 XML 的业务指令分解技术研究[J]. 计算机技术与发展, 2009, 19(7): 234-237.
- [6] 柴晓路, 梁宇奇. Web Services 技术、架构和应用[M]. 北京: 电子工业出版社, 2003.

(下转第 151 页)

力,取消了 Snort 对缓冲区溢出的规则。

为了使系统缓冲区溢出,被用来攻击的字符串总长度应达到 4096 个字符以上。由于攻击程序所产生的字符串的最后一部分是真正产生攻击的。因此,取攻击字符串的最后 75 个字符作为导致程序异常行为的程序输入来训练神经网络。另一方面,正常的 lpr 输入字符串一半长度不长于 75 个字符,因此,以输入的前面的 75 个字符作为程序正常行为的输入来训练神经网络。

所以,神经网络的输入层节点个数为 75,根据仿真实验使用输出层的神经元的个数为 20 * 20 个。

实验中,共获得数据 2000 份,其中有 600 份是导致程序异常的数据,将数据平均分为两部分,每一部分包含 300 份异常数据,拿出其中的一个部分作为训练神经网络,剩下的一部分用来检验神经网络,试验的结果如表 1 所示。

表 1 神经网络算法数据对比

	BP	SOM	SOM+LVQ
检测率	86.1%	90%	94%
误报率	22%	18%	14%

经过试验数据的比较,发现:

1)文中的混合神经网络算法相对于传统的 BP, SOM 正确率有所提高。

2)混合神经网络可以有效地应用专家指导信号。SOM 网络先前的聚类可以减少 LVQ 的训练时间。国内的一些技术虽然也能达到较高的检测率,但是以较长的训练时间为代价的。

3)本方法仍有较高的虚报率,在于先前训练的数据还不能够完全反映系统服务程序 lpr 被攻击的行为,所以算法需要在数据的选择方面加以改进。

3 结束语

入侵检测的误用检测,作为早期的主流检测技术,

一直以它的高速来获得人们的青睐,但由于新型入侵技术的日益增加,误用检测已无法单独保护现在的网络安全。异常检测优势在于能够检测出以前没有检测出的入侵,与误用检测不同的是,异常检测的模式库需要利用程序以前的运行行为来自动的建立,需要庞大的训练样本。其核心在于对以前程序的行为类的边界是否明晰,但是现有的技术是无法完全将现有攻击技术通过异常检测技术训练成模式。

文中将误用检测和异常检测相结合,通过 Snort 检测已知的人侵行为,然后根据 SOM 算法对网络数据先聚类,再利用 LVQ 在聚类的基础上分类的方法,使分类的边界更加明晰,试验证明混合入侵检测系统在防御已知网络攻击和检测未知攻击上都有了明显的提高。

参考文献:

[1] Chenfeng Vincent Zhou. A Survey of Coordinated Attacks and Collaborative Intrusion Detection[J]. Computers & Security, 2009(6):1-17.

[2] 张 铮. 改进贝叶斯分类算法在入侵检测中的研究[J]. 计算机技术与发展, 2007, 17(1):174-176.

[3] Schultze E E. The Future of Intrusion Prevention[J]. Computer Fraud & Security, 2007(8):11-13.

[4] Ganame A K. A Global Security Architecture for Intrusion Detection on Computer Networks[J]. Computers & Security, 2008(3):30-34.

[5] 青华平. 基于模式匹配和神经网络的分布式入侵防御系统的研究[J]. 计算机安全, 2006(2):33-35.

[6] 廖光忠. 基于 Linux 的网络入侵防御系统的研究和设计[J]. 计算机技术与发展, 2008, 18(6):134-136.

[7] 张广斌. 基于神经网络的入侵防御系统研究[J]. 微计算机信息, 2007, 24:64-66.

[8] 孙吉贵. 聚类算法研究[J]. Journal of Software, 2008, 19(1):48-61.

(上接第 140 页)

[7] 王天亮,陈 刚,徐宏炳. 基于共享数据库的数据共享技术[J]. 计算机工程与设计, 2007, 28(8):1923-1926.

[8] 汪明申,王 强. Mashup 系统构建研究[J]. 现代图书情报技术, 2009(5):34-38.

[9] 廖建尚. 论 Mashup 技术与传统系统集成方案的异同[J].

电脑知识与技术, 2009, 12(4):3249-3250.

[10] 邓启辉,赵 英. RIA 技术在数字图书馆中的应用[J]. 图书馆学研究, 2008(6):27-30.

[11] Shanahan F. Mashups Web2.0 开发技术[M]. 吴宏泉译,北京:清华大学出版社, 2007:129-150.

中国计算机学会会刊、中国科技核心期刊
《计算机技术与发展》欢迎订阅,邮发代号:52-127