

Mashup 在图书馆的应用

肖 铮

(厦门大学图书馆 厦门 361005 中国)

文 摘: Mashup 是随着 Web2.0 的出现而发展起来的一种新的数据应用模式。本文首先介绍了 Mashup 的概念、发展过程、模型架构和主要技术。随后列举了若干国外图书馆的 Mashup 应用案例,并对各个应用的现实作用进行了分析。重点介绍了厦门大学图书馆 OPAC 的 Mashup 应用实现过程,强调了图书馆 Mashup 应用对图书馆现有服务的提高作用。最后提出图书馆在 Mashup 应用中不仅可以作为应用提供方集成第三方的数据资源,也可以作为数据提供方将图书馆的数据资源提供给其它应用进行 Mashup。

关键词: Web2.0; 图书馆 2.0; Mashup; OPAC

中图分类号: TP393; G35

Web2.0 经历了数年的发展,已经从一个高度热门的概念,逐渐转化为一个个清晰的应用,比如博客、Wiki、RSS、社会性网络(SNS)、标签和民俗分类法。OCLC 的副总裁、首席战略家 Lorcan Dempsey 使用了“发散(diffusion)”和“聚合(concentration)”这两个标签来描述 Web2.0^[1]的特点,而 Mashup 应用则正是对这两个标签的最好诠释。

1 Mashup 简介

1.1 Mashup 的概念

Mashup 一词源于流行音乐,指从两首不同的歌曲(通常属于不同的流派)中混合演唱和乐器的音轨而构成的一首新歌。随着 Web2.0 的兴起和发展,许多新的技术和理念也随之而来。Web2.0 这一概念的创始人 Tim O'Reilly 总结 Web2.0 的特性包含:使用者的参与及互动、开放的分享与传递、强调使用者网络的外部延展性,也就是经由服务的提供,形成去中心化的形态^[2]。Mashup 就是 Web2.0 的一个产物,并成为 Web2.0 的一个代表性应用。

了解了 Mashup 一词的起源就不难理解 Mashup 在 Web 应用中的作用。Mashup 是一种新型的基于 Web 的数据集成应用模式,它利用现有的 Web 应用程序或数据来源,重新组合这些资源建立起新的应用。Mashup 利用组织边界外的数据资源和服务,在网页或者桌面应用程序中对这些资源加以利用重组并展示出来。

1.2 Mashup 的兴起

Mashup 的兴起跟 Web2.0 的发展是分不开

的,Web2.0 提出的“开放聚合”理念大大推动了 Mashup 的应用。Web 在经历了数十年的发展,到今天已出现了许多拥有海量数据的 Web 站点,同时各站点间的竞争也日益激烈。要保证在竞争中的优势地位,Web 站点必须要吸引更多的流量。因此,开放成为了他们的必要选择。Mashup 应用的根基正是这些数据提供方的开放策略,他们通过开放数据,创建 API,并推广这些 API 的应用,使其数据覆盖范围不仅仅局限于自身站点内部,而是网络上任何可能用到这些数据的地方,从而吸引更多潜在的用户和流量。这样一来,他们在开放的基础上,也就达到了更好地聚合这一目的。

1.3 Mashup 的架构及相关技术^[4]

如图 1 所示, Mashup 从架构上是由三个不同的部分组成,它们在逻辑上和物理上都是相互分离的(可能由组织边界和网络分隔)^[3]。

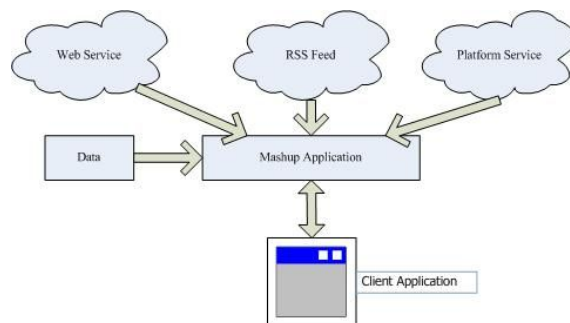


图 1 Mashup 的架构

1) 数据/服务提供方。Web 上拥有数据或者资源的一方,他们使用标准的 WEB 服务规范(如 SOAP 或 REST 规范),将自己的数据或资源以文

本、RSS/Atom、或其他 Web Service 方式提供给第三方进行应用。

数据/服务提供方常用技术有：

a) Web Service: Web Service 是一个概念集，通常指那些将数据传送给客户端的技术，这些技术成为机器与机器，机器与用户间交互的标准。

b) SOAP: 一种基于 HTTP 的消息传递规范。SOAP 规范中有两个关键组件，第一个组件是使用 XML 消息格式进行平台无关的编码，第二个组件消息结构，包括消息头和消息体。消息头用来交换非特定与应用负载（消息体）的相关信息，如认证信息。

c) REST: 一种只使用 HTTP 和 XML 进行 Web 通信的技术。它的简单性和缺少严格配置文件的特性使它与 SOAP 很好地隔离开来，因此被称为轻量级的接口服务。

REST 支持 POST/GET/PUT/DELETE 这几种简单的操作，可通过 HTTP 直接对一个由 URI 定位的资源进行操作。

d) RSS/Atom: 基于 XML 的格式标准，可以方便地向订阅 Feed 的用户提供数据，也可以方便地在机器间传递数据。

2) Mashup 应用实现方。根据 Mashup 实现的位置不同，可分为服务器端 Mashup 和客户端 Mashup。

服务器端 Mashup 指在服务器上使用后台编程语言（JAVA/PHP/.NET/...）对数据提供方的数据进行调用处理，并根据需要将数据与原有服务整合，所有数据调用处理工作均在服务器端完成，最终用户直接从服务器上获取新的服务。

客户端 Mashup 指在用户浏览器上使用 HTML+Javascript 代码，通过 Ajax 技术，形成富 Internet 应用程序，在数据表现层对内容进行整合，以增强交互式用户体验。

两者相比较，客户端 Mashup 使用富 Internet 应用程序具有明显优点，表现为：对 Mashup 应用的服务器所产生的负载较轻（数据可以直接从数据提供方传送到用户，而不通过应用服务器），具有更好的无缝用户体验（用户不必转向内容提供方的网站，而获得该网站所提供的内容）。

Mashup 应用层常用到的技术有：

a) Ajax: Ajax 本身就是一个 Web 应用模型，它包括几种关注内容的异步加载和呈现的技术，将这些技术结合起来使用时，它们的目标是通过与内容服务器交换少量的数据为用户创造平滑、良好的 Web 体验，而不用在用户执行某些操作之后重新加载并重新呈现整个页面。

b) 用于网页编码规范化的 XHTML 和用于确定表现层风格的 CSS。

c) 为在浏览器中实现动态显示和交互所提供的 DOM（文档对象模型）API。

d) XML，作为数据传递基础，实现异步数据交换。

e) Javascript，浏览器端脚本，实现富 Internet 应用程序的编程代码。

3) 用户层：用户最终使用 Mashup 的地方，通常是客户端程序或 Web 浏览器，以图形化的方式呈现应用程序，通过 Mashup 与用户进行交互，使用客户端脚本改善用户体验。

2 Mashup 在图书馆中的应用实例

图书馆在开放的网络环境中面临着服务模式的转变，Web2.0 的发展为图书馆进行这一转变提供了多种可借鉴的方法。在图书馆界，Blog 的兴起拉近了图书馆跟读者的距离，使图书馆跟读者形成了良好的互动关系。RSS 的应用架起了图书馆与读者间的信息桥梁，使读者能够及时主动地获取图书馆的最新信息。Tag 的引入铺垫出一条读者创建组织图书馆资源的道路，使读者能够按照自己的方式组织图书馆的资源。Mashup 做为 Web2.0 下的一个重要应用，也日益吸引着图书馆参与其中。

2.1 NCSU Library 的 Google Maps Mashup^[5]

NCSU，美国北卡罗莱纳州立大学，共占地 101000 英亩，其五所图书馆分布在三个校区之中。NCSU 图书馆根据自身分馆多、分布散的实际情况，利用 Google Maps API，将图书馆位置标注在 Google Map 上，并使用 Mashup 将此位置图混搭在图书馆网站中。用户不但可以在图书馆网站上直接查到图书馆位置信息和地图信息，还可以利用 Google Maps 提供的服务获取行车路线，或通过 Google Maps 的卫星视图直接地看到图书馆馆舍和周围建筑鸟瞰图。

2.2 UIUC Library 的 Twitter Mashup^[6]

UIUC，伊利诺大学香槟-厄班纳分校。该校图书馆使用 Twitter^[7]发布最新的图书馆公告，任何人只要在自己的 Twitter 帐号上 Follow 图书馆的 Twitter 帐号就可以通过 IM、手机或网页即时获取图书馆最新公告。与此同时，图书馆网站则通过 Mashup 其在 Twitter 上发布的消息，使图书馆实现了在 Twitter 一处发布消息，在其它各处调用，避免了重复发布信息的成本和信息更新不同步的问题。

2.3 Danbury Public Library^[8] 的 Library

Thing^[9] Mashup

Danbury 公共图书馆是 LTFL (Library Thing For Libraries^[10]) 的第一个用户, 在 Danbury 图书馆的 OPAC 书目信息页面中, 通过 LTFL API 服务, 调用 Library Thing 网站中该书的 Tag 数据和 Similar Books 数据, 大大丰富了自身 OPAC 书目页面, 使用户可以通过 Tag 数据找到由用户自主分类的同类图书, 通过某本书的 Similar Books 数据找到类似图书。Danbury 图书馆的 OPAC 在利用 Library Thing 提供的数据的基础上, 将原本孤立的 OPAC 的书目信息有效地联系起来, 加强了图书的揭示功能。

由以上三例应用可以看到, 图书馆利用 Web 上的开放数据与服务, 通过 Mashup 与自身数据或服务相融合, 快速地实现了新的服务。这些服务一方面将用户与自身熟悉的 Web 服务联系起来, 使用户感到亲切, 增强了用户对图书馆网站的使用体验。另一方面, 图书馆内在的数据与服务也因为和新数据的融合实现了信息增值。可以说, Mashup 的应用以较低的成本投入, 获得了较高的收益。

3 厦门大学图书馆 Mashup 应用实例

目前厦门大学图书馆采用的 OPAC 系统在功能上还存在某些不足, 比如 OPAC 提供的元数据信息还仅仅限于文献外部信息的描述, 缺少对文献内容的描述, 读者通过 OPAC 获取到的信息不足以帮助他们进行快速挑选和剔除。但受限于书目 MARC 数据本身的规范性, 很难直接将对内容的描述信息加入其中。而且如果将对文献本身描述的信息加入到 MARC 中, 维护 MARC 记录的成本也会大大增加。

而如今, 在网络上出现了诸如 Amazon, Google, Library Thing, 豆瓣这类拥有海量数据的网站, 并且在 Web2.0 理念的作用下, 他们逐步为第三方开放了自身的数据, 这就为图书馆利用这些数据加强自身服务提供了良好的契机。

3.1 Google Books Search API^[11]

GBS (Google Books Search) 项目提供的 API 使第三方可以调用 GBS 的图书封面图片、全文可获得的状态、到图书搜索结果的链接。此 API 提供静态链接和动态链接两种方式调用 Google Books 相关内容^[12]。静态链接直接通过 ISBN/LCCN/OCLC 控制号链接到 Google Books 中的信息页面, 这种方式简单易用, 但缺点是无法得知 GBS 的查询结果, 就无法判断 GBS 中是否存在该书记录, 某些时候会影响用户体验的效

果。动态链接使用 REST 风格的消息传递模式, 客户端通过使用相关参数向 GBS 发出查询请求, 查询结果以 XML 格式返回, 开发者可根据返回结果选择性地将相关信息展现给用户。

3.2 豆瓣网 API^[13]

豆瓣网是目前国内最为成功的 Web2.0 网站, 已有注册用户 700 多万, 用户可自行书籍/音乐/电影创建标签, 添加评论。豆瓣中关于书籍的数据可分为以下四类: 书目信息、用户评价、图书推荐和常用链接。豆瓣提供的 API 使第三方可通过编程搜索并查看书籍信息, 添加、更新、删除用户收藏, 查看、发布、修改、删除用户评论等等。因此, 在 OPAC 页面中, 可以使用 Ajax 技术, 通过 Javascript 动态查询某本书, 然后根据返回 XML 数据结果, 提取其中的图书封面信息、内容简介、用户标签等相关内容展示在 OPAC 页面。

3.3 具体实现步骤

厦门大学图书馆 OPAC 系统使用 PHP 编程, 为二次开发和修改提供了极大地便利。选用客户端 Mashup 方式, 在 OPAC 系统的用于显示图书详细信息页面 item.htm 中加入调用 GBS API 和豆瓣网 API 的 Javascript 程序。当用户浏览到该页时, 会自动加载 Javascript 程序代码, 程序通过 API 调用 GBS 的查询结果和豆瓣网中图书的封面图片地址、内容简介和常用标签, 再使用 Ajax 动态操作页面 HTML DOM, 将取回的内容放置到合适的位置, 供用户使用。程序流程图如图 2 所示:

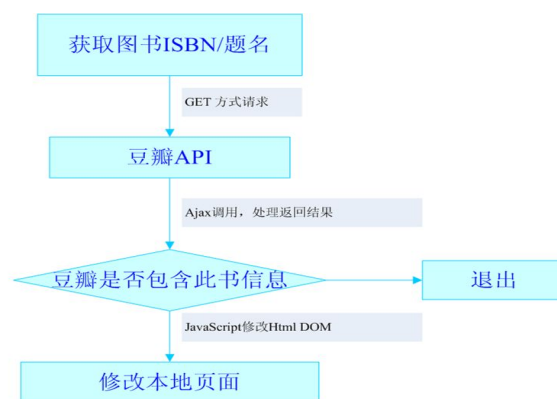


图 2 OPAC 与豆瓣网 Mashup 应用程序流程图

程序代码片段及说明:

1) 加载豆瓣 API Javascript 类库

```
<script type="text/javascript"src="http://www.douban.com/js/api.js?v=2"></script>
<script type="text/javascript" src="http://www.douban.com/js/api/parser.js?v=1">
```

```
</script>
```

2) OPAC 页面获取图书 ISBN

```
var isbn = document.getElementById('book_isbn').innerHTML;
isbn = isbn.replace(/-/g, "");
```

3) 调用豆瓣 API

```
var subject_id;
DOUBAN.apikey = 'doubanAPIkey';
DOUBAN.searchBooks({
  keyword:isbn,
  callback:function(re){
    if (re.entry.length != 0)
    {
      subject_id=re.entry[0].
      id.$t.substring
      (re.entry[0].id.$t.lastIndexOf('/')+1); }
  }
});
```

4) 处理返回的 JSON 对象

```
DOUBAN.getBook({
  id:subject_id,
  callback:function(re){
    var subj = DOUBAN.parseSubject(re);
    If (typeof(variable)=='undefined')
    db_cover =
    (typeof(subj.link.image) == 'undefined') ? " : "<a
    href='http://www.douban.com/subject/'+ subject_id
    +' target=_blank></a><span
    id="gbs_intro">豆瓣提供<br/>相关介绍</span>";
    db_summary=
    (typeof(subj.summary)=='undefined')? " : "<dt>
    内容简介:</dt><dd>"+subj.summary+"</dd>";}
```

5) 改写 OPAC hml 页面

```
element=document.getElementById('db_cover');
element.innerHTML = db_cover;
element=document.getElementById('db_api');
element.innerHTML = db_summary;
```

经过 Mashup 后的 OPAC 页面如图 3 所示，在新的页面中，融合了豆瓣图书封面、图书简介、图书标签，Google 图书预览服务。对于用户来说他们并不关心数据从哪里来，而真正关心数据是否对自己有用。通过 Mashup 混搭给用户以更多选择，丰富了 OPAC 页面内容，弥补了原有系统中的某些不足，使图书馆服务更加贴近用户。



图 3 厦门大学图书馆 OPAC 与豆瓣网的 Mashup 应用

4 结语

Mashup 为图书馆提供了一条简单易用的方式利用 Web 上的海量数据，改进图书馆自身的某些不足，帮助图书馆在网络环境下更好地为读者服务。但就目前的情况看，图书馆在 Mashup 的框架中，还多是扮演 Mashup 应用创建者的角色，而在 Web2.0 的理念下，图书馆也应该以更加开放的心态面对来自各方面的竞争，积极向数据提供者的角色转变，主动将图书馆的数据开放出来，在更广阔的网络中聚合更多用户。

参考文献

- [1] Lorcan Dempsey. Lorcan Dempsey's weblog[EB/OL]. (2008-10-10). <http://orweblog.oclc.org/archives/001556.html>
- [2] Tim O'Reilly. What is Web2.0:Design patterns and business models for the next Generation of Software,2005[EB/OL]. (2008-10-10). <http://www.oreillynet.com/lpt/a/6228>
- [3] Larry Clarkin, Josh Holmes. 企业级 Mashup [EB/OL].(2008-10-10).<http://msdn.microsoft.com/zh-cn/library/bb906060.aspx>
- [4] Duane Merrill. Mashups:Web 应用程序新成员[EB/OL]. (2008-10-10). <http://www.ibm.com/developerworks/cn/xml/x-mashups.html>
- [5] NCSU 图书馆位置地图[EB]. [2008-10-10]. <http://www.lib.ncsu.edu/liblocations/>
- [6] UIUC 图书馆网站[EB]. [2008-10-10]. <http://www.library.uiuc.edu/ugl/>
- [7] Twitter[EB]. <http://twitter.com>
- [8] Danbury 图书馆 OPAC[EB]. <http://cat.danburylibrary.org/search>
- [9] Library Thing[EB]. <http://www.librarything.com>.
- [10] Library Thing for Libraries 服务. [EB] http://www.librarything.com/wiki/index.php/LTFL:Libraries_using_LibraryThing_for_Libraries
- [11] Google Books Search API[EB] <http://code.google.com/apis/books/>
- [12] 编目精灵. 用 Google 图书搜索 API 增强[EB/OL]. [2008-10-10]. [OPAChttp://catwizard.blogbus.com/logs/17095011.html](http://catwizard.blogbus.com/logs/17095011.html)
- [13] 豆瓣 API 使用指南[EB]. [2008-10-10]. <http://www.douban.com/service/apidoc/guide>

Mashup in Library

Xiao Zheng

(Xiamen University Library, Xiamen 361005, China)

Abstract: This paper introduces Mashup, a new application of Web2.0, gives its concept, development, module. Then the paper gives some examples of Mashup in foreign library. Meanwhile take Xiamen University OPAC Mashup implement as an example to show how to use it improve library service. At last authors show a prospect of position transformation of library in Mashup module in the future.

Key words: Web2.0;Llibrary2.0;Mashup;OPAC