

一种基于社交网络服务的 mashup 连接子的设计与实现

庄希威 孙艳春 黄 罡

(北京大学信息科学技术学院软件研究所 北京 100871)

(高可信软件技术教育部重点实验室 北京 100871)

摘 要 在 Web 2.0 时代,富客户端应用越来越流行。混搭(mashup)可以将两种或两种以上的互联网资源整合成具有良好用户体验的新型应用。利用 mashup 现有技术,可以实现同一个富客户端上不同富客户端应用之间的信息传递。不过,mashup 对于不同富客户端间的富客户端应用的信息传递仍然无能为力,因此一般情况下,用户无法获得他人经验或集体智慧的帮助而更好地使用富客户端应用。而另一方面,社交网络服务 SNS(Social Network Service)拥有及时性、交互性、广泛性和富含集体智慧的特点。文中基于 SNS 的这一系列特性,提出了一种基于社交网络服务的 mashup 连接子(connector)的设计方案,使得用户可以利用社交网络好友的经验和集体智慧,更为充分地使用富客户端应用,同时也解决了不同富客户端上的应用之间的信息传递问题。然后,实现了基于社交网络服务的 mashup 连接子,并进行了实例研究,验证了该连接子设计的可行性和有效性。

关键词 Connector, Mashup, 富客户端, SNS(Social Network Service)

中图分类号 TP302 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.11.014

Design and Implementation of SNS-based Mashup Connector

ZHUANG Xi-wei SUN Yan-chun HUANG Gang

(Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

(Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

Abstract In Web 2.0 era, rich client applications get popular. Usually, rich client applications are independent, by using mashup, we can create a new application with better user experience from two or more Internet sources. Mashup can help the information transfer between various applications in the same rich client, not different rich clients. However, when using applications, people usually need experienced friends' help. So currently it fails to combine collective experience and intelligence. On the other hand, SNS (Social Network Service) becomes hot Web service with its timeliness, interactivity, universality and collective experience and intelligence. Taking advantage of SNS, we designed one kind of SNS-based mashup connector, so that collective experience and intelligence are added. At the same time, it solves the information transfer problem of applications between different rich clients. Then we implemented it and gave case study, which verifies the feasibility and validity of the mashup connector design.

Keywords Connector, Mashup, Rich client, SNS

1 引言

随着技术的不断发展和人们对用户体验需求的不断增加,传统的互联网应用已经不能满足用户的需求,进而发展出了富客户端应用。富客户端应用将原先在服务器上的部分运算搬到了客户端,客户端可以根据用户需要,对数据进行相对复杂的计算,以更灵活的方式呈现。

目前,开发人员可以根据互联网公司提供的 API(Application Programming Interface)^[1],将复杂的富客户端应用进

行整合,形成各种各样的新应用,这就是通常所指的混搭(mashup)。由于 API 返回的都是结构化的数据,因此在同一个富客户端,不同的应用进行信息传递是可行的。但是,如果应用服务器本身不提供账户进行直接交流,则不同的富客户端之间的信息传递存在问题。而实际上,很多的应用,都需要进行跨富客户端的沟通。比如,当一位用户在富客户端上使用携程(www.ctrip.com)的应用预订机票时,就非常希望与在另一个富客户端上的朋友进行沟通,希望对方推荐携程网上适合的机票、宾馆等信息,甚至希望对方将在携程网上的操

到稿日期:2013-09-18 返修日期:2013-11-16 本文受国家重点基础研究发展规划 973 项目(2011CB302604),国家自然科学基金联合基金项目(U1201252),国家高技术研究发展计划 863 项目(2011AA01A202),国家自然科学基金项目(61073020),国家自然科学基金创新研究群体科学基金(61121063)资助。

庄希威(1989-),男,硕士生,主要研究方向为网构软件中间件,E-mail:zhuangxw12@sei.pku.edu.cn;孙艳春(1970-),女,副教授,主要研究方向为软件工程、服务计算、计算机支持的协同工作,E-mail:sunyc@sei.pku.edu.cn(通信作者);黄 罡(1975-),男,教授,主要研究方向为软件工程、分布式计算、中间件。

作直接传递过来^[2]。而这个要求,目前没有一般性的解决方案。

另一方面,社交网络服务(Social Network Service, SNS)目前成为了非常热门的网络服务。SNS 拥有信息发布的及时性、信息传递的交互性和使用人员的广泛性等特点。而且, SNS 的高活跃度可以保证众多好友用户的同时在线,而且 SNS 的用户可以很好地进行及时通信和交互。更为重要的是,社交网络服务可以提供集体智慧和他人经验共享,而这为用户更好地使用富客户端应用提供了有力的支持。

本论文的研究思路是,将富客户端的应用进行 mashup 构件化,形成在单一富客户端应用间的信息传递机制。将社交网络服务作为构件化后的应用之间连接和通信交互的连接子(connector),使得用户可以利用社交网络好友的人类智慧,更为充分地使用富客户端应用,同时也解决了不同富客户端上的应用之间的信息传递问题。

学术界有很多涉及到 mashup 的研究,但都是在单个客户端上的 mashup,并没有实现不同客户端之间的 mashup,如 Wijesiriwardana C. 的文章提出的为 mashup 提供的引导框架解决的还是单个客户端上的 mashup 问题^[3]。学术界也有很多涉及到社交网络的研究,但主要是利用数据挖掘技术从社交网络上的海量数据中挖掘出可用信息,如 Kim J. S. 的文章就是利用社交网络产生的大量数据分析出最受欢迎的智能手机和最有趣的智能手机属性^[4];再如 MrinmayaSachan 在文章中提出了通过对社交网络上信息的分析检测出不同主题的社区的方法^[5],但他们并没有将社交网络服务作为他人经验和集体智慧的源泉来研究,分析的对象是数据,而没有利用社交网络上的好友在线提供及时的经验和智慧。也有利用社交网络来方便交流的研究,如 Kukreja N. 讨论了利用 facebook 和 gmail 的机制来方便需求捕获和双赢谈判^[6],但其主要是利用社交网络的机制重新开发系统,而没能将现有的社交网络直接应用进去,这对于用户来说是不方便的,因为原来的好友资源没有被利用。而且,学术界还没有将社交网络服务以连接子的高度来进行过抽象。

本文第 2 节介绍了基于社交网络的 mashup 连接子的设计;第 3 节介绍了基于人人网的 mashup 连接子的设计和实现;第 4 节介绍了携程网的实例研究;第 5 节总结了本文的工作并对未来工作进行了展望。

2 基于社交网络的 mashup 连接子模型

基于社交网络的 mashup 连接子是用来连接经过 mashup 构件化后的富客户端应用的,因此富客户端应用的 mashup 构件化是基于社交网络的 mashup 连接子的使用基础。基于社交网络服务的连接子的构件组装则是具体介绍如何使用连接子。

2.1 富客户端应用的 mashup 构件化

富客户端应用的 mashup 构件化是实现软件复用、增加效率的有效方式。

富客户端应用的 mashup 构件模型^[7]与传统的构件模型一致,同样由构件接口和构件实现两部分构成,如图 1 所示。

构件接口分为视图接口(View Interface)和功能接口(Programming Interface)两个部分。当一个构件实例化之

后,其他构件可以通过视图接口,获取指向实例化后的构件 HTML 根元素的引用。通过引用,其他构件可以对实例化后的构件视图进行修改。功能接口负责构件之间的信息传递。功能接口又分为两个部分,分别是方法(Method)和事件(Event),其中,方法可被其他构件调用,以查询或修改构件的数据。事件用于发布构件的信息,其他构件可以订阅。当构件内状态发生改变,事件可以发布信息,订阅了该事件的构件都可以收到信息,并执行相应的方法。

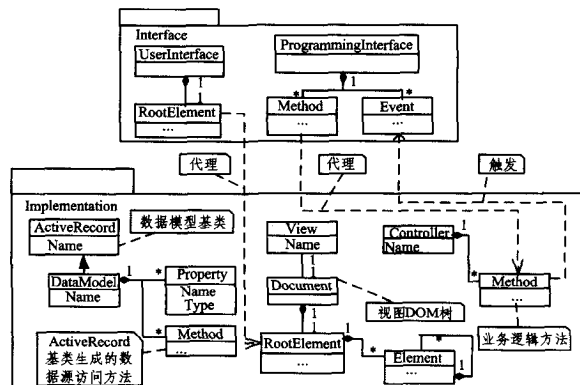


图 1 富客户端应用的 mashup 构件模型

构件实现可以用 MVC 模型进行描述。数据模型(Model)是构件所需的数据类,它的改变会影响最终的数据信息。视图(View)是视图模板中定义的一系列元素。通过修改视图,可以修改构件的页面内容和展示方式。控制器(Controller)是一系列处理数据模型和视图的方法,通过方法调用,可以修改数据和视图内容。

根据数据来源,构件可以分为两个类别。一类是将网页 mashup 构件化形成的构件,其视图和部分控制器方法基本遵循了网页原有的内容。第二类是通过调用 API 来实现的 mashup 应用构件,其视图、控制器和数据类型主要由开发者设定。

2.2 基于社交网络服务的连接子模型

连接子是构件之间的信息交互的抽象。很多时候,连接子会被作为与构件同等重要的实体进行研究。

基于事件的功能组装可以实现同一个富客户端上 mashup 构件的功能组装,模型如图 2 所示。

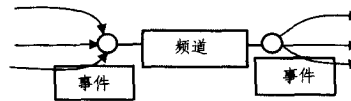


图 2 基于事件的连接子模型

频道(channel)相当于一个连接子,可以连接多个 mashup 构件,实现功能的组装。与频道关联的事件分为传入的事件和传出的事件,每个频道可以定义多个事件。将事件传入频道的 mashup 构件称为发起者,从频道接收事件的 mashup 构件称为订阅者。在 mashup 构件中,定义了各种事件,事件包含各种参数,当发布者发布一个事件后,所有该事件的订阅者会执行相应的事件。这是一个异步的过程。比如,当 Google 的 mashup 构件发布一个包含查询地参数的事件,若 Yahoo 的 mashup 构件收听了该事件,则 Yahoo 的 mashup 构件可以执行相应的事件,如搜索查询地的天气。

基于事件的连接子只能满足同一个富客户端上的

mashup 构件的信息传递,当不同的富客户端上的 mashup 构件需要消息传递时,基于事件的连接子便无能为力了。这里,为了增加经验分享和集体智慧以更好地使用富客户端应用,同时解决跨富客户端的信息传递,本文提出了基于社交网络服务的连接子模型,如图 3 所示。

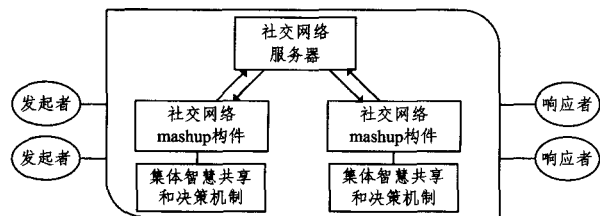


图 3 基于社交网络服务的新型连接子模型

图中的两个发起者和响应者表示连接子可以提供多个接口连接多个应用,如果只需连接携程应用,则只要一个发起者和一个响应者。

我们的整体思路是,将发起者的信息由频道传给社交网络的 mashup 构件(构件 A),通过构件 A 传递到社交网络服务器,再由社交网络服务器将信息传递给另一个富客户端上的社交网络的 mashup 构件(构件 B),构件 B 通过频道将信息传给另一个富客户端上的响应者。

我们要将社交网络应用进行 mashup 构件化。其原因是,基于事件的连接子模型可以保证同一个富客户端上的 mashup 构件能够进行沟通,我们通过 mashup 构件化可以使社交网络应用与发起者和响应者进行通信,进而使得基于社交网络的 mashup 连接子与发起者和响应者能够顺利地流通信息。构件化的过程包括定义事件在内的一系列处理,在第 3 节的具体实现中会讲到,此处不再细说。

基于社交网络的 mashup 的连接子内部是复合结构,由频道、社交网络 mashup 构件、社交网站服务器组装实现。其中频道提供对外的接口。

2.3 基于社交网络服务的连接子的构件组装

基于事件的连接子的组装过程如图 4 所示。

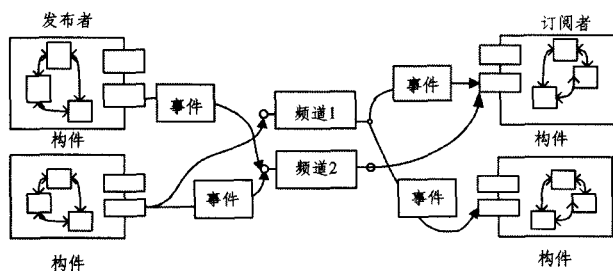


图 4 基于事件的连接子的构件组装

频道传入的事件和传出的事件数量都可以大于 1,且事件可以由不同的构件发出或响应,如图 5 中频道 2 的传入事件由两个不同的构件发出,频道 1 的传出事件由两个不同的构件响应。发生频道的任何一个传入事件,则该频道所有的传出事件都会响应。同一个构件可以由事件连接到不同的频道,如图 5 中左下角的构件的事件被分别传入频道 1 和频道 2。

基于社交网络服务的连接子的构件组装,指的是将基于社交网络服务的连接子与富客户端上的 mashup 构件进行组

装,使得不同富客户端上的构件之间能够实现通信。基于社交网络服务的连接子的构件连接过程如图 5 所示。

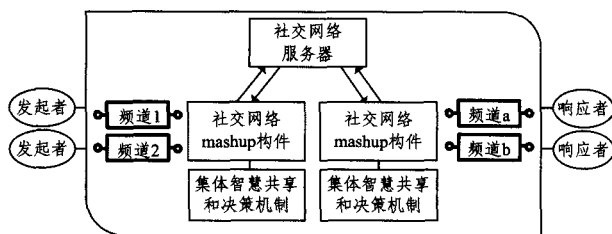


图 5 基于社交网络服务的连接子的构件组装

由于社交网络服务 mashup 构件和发起者、响应者都是相同标准的构件,因此可以通过基于事件的连接子进行功能上的连接。不同富客户端上的应用使用频道的连接过程,与同一富客户端上的构件之间的连接完全一致。发起者通过社交网络的通路,就能实现与另一个富客户端上的响应者进行通信。

基于社交网络服务的 mashup 连接子与 mashup 构件进行组装时,需要满足一定的约束,以保障其功能完整。这里,我们主要从构件的一致性和配置的一致性进行约束。

构件的一致性包括^[8]:

- 所有进行组装的 mashup 构件必须具有唯一的构件名。
- 通过频道连接的构件必须具有相应的事件或方法。
- 连接子中的社交网络应用必须进行 mashup 构件化,且与其他通过连接子连接的构件经过相同的 mashup 构件化过程,满足同一个 mashup 构件模型。

每个构件至少有一个实现。

配置的一致性包括:

- 频道的传入事件和传出事件必须保证事件参数一致,即参数数量和类型保持一致。

3 基于人人网的连接子的实现

iMashup^[9]是基于富客户端中间件,封装中间件功能、提供中间件功能可视化操作界面的富客户端应用框架。iMashup 支持用户以“所见即所得”利用 iMashup 的构件容器和组装机制,完成构件的实例化和组装。它能够分别满足开发者和最终的需要。本文基于社交网络的 mashup 连接子的设计与实现都在 iMashup 的框架上进行。

3.1 基于人人网的 iMashup 连接子的携程应用场景

在基于社交网络的 mashup 连接子设计的基础上,我们选择人人网(www.renren.com)作为社交网络的例子,通过组装携程网应用的 iMashup 构件与基于人人网的 iMashup 连接子,完成一个现实生活中会应用到的功能,该应用场景如图 6 所示,其中:

(1)用户 A 想要从北京出发到上海旅游,于是他使用经过 iMashup 构件化的携程网应用,查询了从北京到上海的机票。他希望得到朋友的一些建议,于是他通过设置频道,将携程网构件与同样经过 iMashup 构件化的人人网应用连接。于是,他在携程应用上的查询内容通过状态发表在了人人网上。

(2)用户 B 在另一台机器上登录着 mashup 构件化后的

人人网应用,并且看到了好友 A 发布的状态。由于 B 曾经也去过上海旅游,于是他点击人人网应用上的建议按钮。这时,控制器会自动产生一个携程网构件的实例,并且将携程网构件与人人网构件在用户 B 的计算机上通过频道进行连接。用户 B 在携程应用中搜索了北京到上海的机票,并且在返回信息中勾选了自己推荐的航班,人人网应用会自动生成一个状态回复记录,用户 B 点击后就可以发布。同样地,用户 B 可以在携程应用中搜索上海的宾馆,并对其中熟悉的宾馆进行推荐。人人网同样会生成对应的状态回复。

(3)用户 A 在人人网应用上看到了用户 B 对自己的建议后,可以点击采纳,则该回复记录的用户 B 对应的操作在用户 A 中会呈现。例如,用户 A 点击了用户 B 对于宾馆的建议后,用户 A 的计算机上的携程应用就会出现在用户 B 推荐的宾馆的界面上,则用户 A 看到详细信息后,可以进一步操作,预订或者换一家。



图 6 基于人人网的 iMashup 连接子的携程应用场景

3.2 人人网服务封装

人人网是国内非常热门的综合性社交网站,用户量在 2012 年底达到 2 亿^[10]。

本文通过数据模型、视图和控制器(MVC)3 个方面来介绍人人网构件的包装。

(1)数据模型

数据模型(Model)是构件所需的数据类,它的改变会影响最终的数据信息。由于直接包装了传统 Web 应用,因此网页上的很多数据都存在于构件中。为了事先附加的功能,增加了 replyId 等全局的数据。

(2)视图

视图(View)是视图模板中定义的一系列的元素。而人人网构件主要采用了人人网网页本身的视图。通过运行时的控制器的操作,视图会发生变化。如在与携程相关的状态后面添加了建议按钮,在相关回复后面添加了采纳按钮。人人网构件的视图模板定义如表 1 所列。运行时实际视图如图 7 所示。

表 1 人人网构件视图模板定义

```

<div>
  <div class="renren">
    <iframe dojoAttachPoint="if1" id="if1" src="
      http://www.renren.com" width=490 height=
      470>
    </iframe>
  </div>
</div>

```

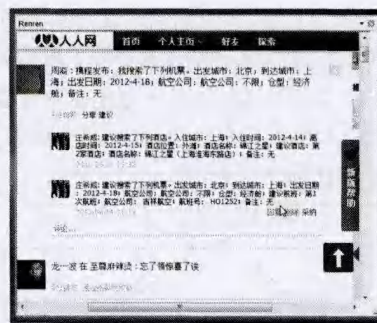


图 7 人人网构件运行时实际视图

(3)控制器

控制器(Control)是一系列处理数据模型和视图的方法,通过方法调用,可以修改数据和视图内容。人人网网页本身有很多的处理方法,这些方法是网页应用所必须的,就不在此列举。而作者在构件化过程中增加了一些事件和方法,这是人人网构件对外的主要接口。事件和方法如表 2 所列。

表 2 人人网构件的事件和方法

```

methods: {
  "setLocation" : {Function: "setLocation", CustomMethod: "/ * argu-
    ments[0];String * /"},
  (!--携程搜索后,人人网构件填充状态的方法--)
  "setSuggestedFlight" : {Function: "setSuggestedFlight", Custom-
    Method: "/ * arguments[0];String * /"},
  (!--携程搜索建议航班后,人人网构件填充状态回复的方法--)
  "setSuggestedHotel" : {Function: "setSuggestedHotel", CustomMethod: "/
    * arguments[0];String * /"},
  (!--携程搜索建议酒店后,人人网构件填充状态回复的方法--)
  },
events: {
  "acceptSuggestion" : {Function: "acceptSuggestion", CustomMethod: "/
    * arguments[0];String * /"},
  (!--用户点击建议按钮后发生的事件--)
}

```

人人网构件本身就是一个独立的构件,可以将人人网应用在构件实例上运行,保持了所有的人人网功能。同时,通过人人网构件,可以形成基于人人网的 Mashup 连接子。基于人人网的 Mashup 连接子可以将人人网用户使用的富客户端应用通过人人网连接在一起,两亿的注册用户就可以形成信息的流通,分享经验和集体智慧。

4 携程网实例研究

4.1 携程网的 iMashup 构件化

携程网是中国领先的在线旅行服务公司。携程旅行网向超过五千万注册会员提供包括酒店预订、机票预订、度假预订、商旅管理、高铁代购以及旅游资讯在内的全方位旅行服务。

因为携程网的账号只是对用户本身的信息进行记录,而没有用户之间的信息共享机制,所以无法完成信息直接交流。而且未注册携程网的用户,同样可以使用携程网的应用,通过基于人人网的 iMashup 连接子与人人好友进行信息传递。

本文通过数据模型、视图和控制器(MVC)3 个方面来介绍携程网构件的包装。

(1)数据模型

数据模型(Model)是构件所需的数据类,它的改变会影

响最终的数据信息。由于直接包装了传统 Web 应用,因此网页上的很多数据都存在于构件中。为了事先附加的功能,增加了 acceptArray 等全局的数据。

(2)视图

视图(View)是视图模板中定义的一系列元素。而携程网构件主要采用了携程网页本身的视图。通过运行时的控制器的操作,视图会发生变化。如携程宾馆搜索界面和航班搜索界面增加了用于建议的勾选框。携程网构件的视图模板定义如表 3 所列。运行时实际视图如图 8 所示。

表 3 携程网构件视图模板定义

```

<div>
  <div class="ctrip">
    <iframe dojoAttachPoint="if1", id="ctripFrame", src="http://www.ctrip.com", width=490, height=470>
  </iframe>
</div>
</div>

```

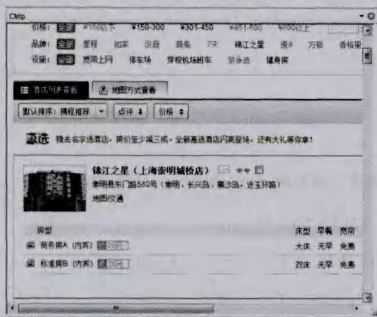


图 8 携程网构件运行时实际视图

(3)控制器

控制器(Control)是一系列处理数据模型和视图的方法,通过方法调用,可以修改数据和视图内容。携程网网页本身有很多的方法,这些方法是网页应用所必须的,就不在此列举。而作者在构件化过程中增加了一些事件和方法,这是携程网构件对外的主要接口。事件和方法如表 4 所列。

表 4 携程网构件的事件和方法

```

methods:{
  "acceptSuggestion": {Function:"acceptSuggestion", CustomMethod:"/* arguments[0];String */"},
},
<!-- 人人网点击采纳按钮后,携程网重现建议过程的方法 -->
events:{
  "MarkerInformation": {Function:"MarkerInformation", CustomMethod:"/* arguments[0];String */"},
  <!-- 用户在携程网搜索航班或酒店后的事件 -->
  "suggetFlight": {Function:"suggetFlight", CustomMethod:"/* arguments[0];String */"},
  <!-- 用户在携程网推荐特定航班后的事件 -->
  "suggetHotel": {Function:"suggetHotel", CustomMethod:"/* arguments[0];String */"},
  <!-- 用户在携程网推荐特定酒店后的事件 -->
}

```

4.2 携程网 iMashup 构件与基于人人网的 iMashup 连接子的组装

用户 A 实例化携程网构件和人人网构件,并且设置频道,使得携程应用上的搜索过程能传递到人人网,该过程如图 9 所示。

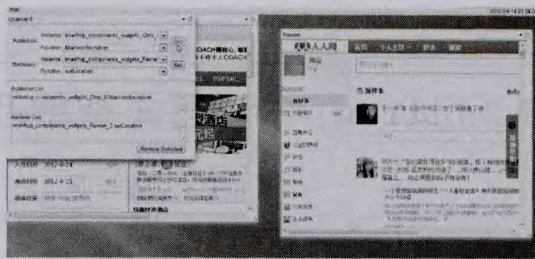


图 9 频道设置

用户 A 在携程网应用中搜索从北京去上海的机票,我们实现的人人网应用会生成对应操作的状态。用户 A 点击即可发布状态,该过程如图 10 所示。

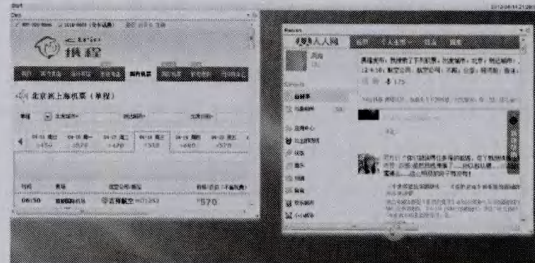


图 10 状态传递

用户 B 在人人网应用中看到了用户 A 发布的状态,点击建议按钮,经过 iMashup 构件化的人人网构件会产生一个携程网构件的实例,并且自动生成 3 个频道,将携程网构件与人人网构件在用户 B 的计算机上进行连接,该过程如图 11 所示。

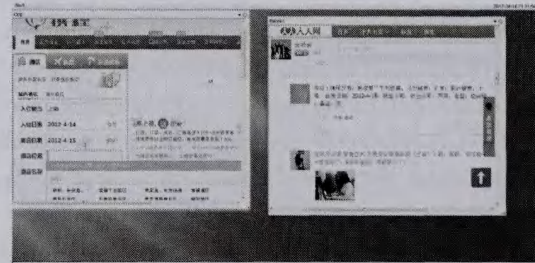


图 11 点击建议

用户 B 在携程应用上搜索航班和酒店,并且在自己推荐的项目上划勾,则人人应用会产生相应的推荐回复内容。用户 B 点击即可以发布回复,该过程如图 12 所示。

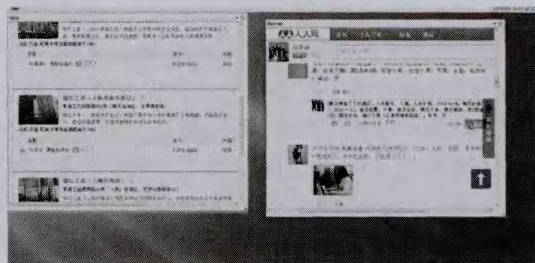


图 12 推荐信息传递

用户 A 看到用户 B 的回复后,在自己认为合适的回复后点击采纳按钮,人人网构件会自动生成一个频道,使得人人网应用的建议内容传送到携程网应用中。携程网会自动显示人

(下转第 98 页)

动态演化过程。系统的性质和行为的动态演化通过应用场景的变迁来体现,当发生特定的事件时,应用场景会变迁到新的场景,构件在不同的场景下的行为方式也随之发生变化,这为系统的演化和复用在场景大粒度上提供了很好的支持。我们将进一步研究复杂软件系统中场景自动机的设计和图形化建模,为系统自演化提供支持。

参考文献

[1] 杨芙清,梅宏,吕建,等. 浅论软件技术发展[J]. 电子学报,2002,30(12A):1901-1906
 [2] 谢仲文,李彤,代飞,等. 基于 Petri 网的面向动态演化的软件体系结构建模[J]. 计算机应用与软件,2012,29(10):36-39
 [3] 冯耀东,黄罡,梅宏,一种自适应软件体系结构建模及其实施方法[J]. 北京大学学报:自然科学版,2008,44(1):67-76
 [4] Gough P A, Fodermiski F T, Higgins S A, et al. Scenarios: An Industrial Case Study and Hypermedia Enhancements[C]//1995 IEEE International Symposium on Requirements Engineering (RE'95). Los Alamitos CA: IEEE Computer Society Press.

1995:10-17
 [5] Carroll J M. Making Use: Scenario-Based Design of Human-Computer Interactions[M]. Cambridge MA: MIT Press,2000
 [6] Cockburn A. Writing Effective Use Cases [M]. MA: Addison-Wesley,2001:159-176
 [7] Sutcliffe A G, Maiden N A M, Minocha S, et al. Supporting Scenario-Based Requirements Engineering [J]. IEEE Transactions on Software Engineering, 1998, 24: 1072-1088
 [8] Jarke M, Bui X T, J M. Carroll Scenario management: An interdisciplinary approach [J]. Requirements Engineering, 1998, 3(1):155-173
 [9] Rolland C, et al. A proposal for Scenario Classification Framework [J]. Requirements Engineering, 1998, 3(1):23-47
 [10] Wang Mao-guang, Mei Hong, Jiao Wen-pin, et al. Multi-agent System Collaboration Based on the Relation-Web Model [C]//AICI 2010. 2010:132-144
 [11] 王茂光. 基于自主构件的自适应网构软件开发方法[D]. 北京:北京大学,2011

(上接第 73 页)

人网中建议的项目所在的页面。用户 A 可以进行预订或者放弃。该过程如图 13 所示。

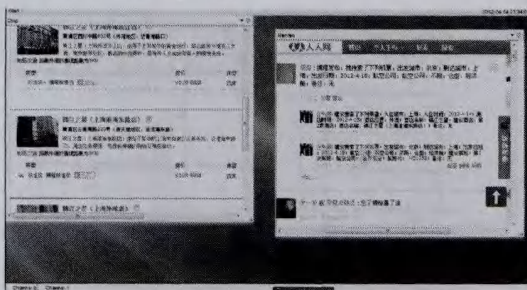


图 13 采纳建议

结束语 在 Web 2.0 时代,富客户端应用兴起。混搭可以解决同一个富客户端的富客户端应用之间的信息流通,但是,除非应用本身具有服务器账户间的直接信息传递机制,否则 mashup 对于不同富客户端间的富客户端应用的信息传递仍然无能为力。因此,一般情况下,用户无法获得他人经验或集体智慧的帮助而更好地使用富客户端应用。

本文利用社交网络的广泛性、交互性、及时性和汇聚集体智慧的特点,提出了基于社交网络服务的 mashup 连接子的设计,使得用户可以利用社交网络好友的人类智慧,更充分地使用富客户端应用,同时也解决了不同富客户端上的应用之间的信息传递问题。

同时在基于中间件的富客户端应用 iMashup 框架内,实现了基于人人网的 mashup 连接子。通过连接子和携程网的 mashup 构件组装,完成了一套基于人人网的 iMashup 连接子的携程应用场景,验证了连接子设计的正确性。

目前利用人人网的状态发布机制进行信息传递,将来可以利用更加丰富的传递形式,如照片、日志、站内信等形式,更加充分地利用他人经验和集体智慧。

可以开发更多的富客户端构件,与基于社交网络的 mashup 连接子进行组装,实现更为复杂的应用场景。

参考文献

[1] Qi Zhao, Gang Huang, Huang Ji-yu, et al. An on-the-fly approach to web-based service composition [C]// Congress on Services Part II, Beijing, China, 2008:208-209
 [2] Liu Xuan-zhe. Composing data-driven service mashups with tag-based semantic annotations [C]//Proceeding of the IEEE International Conference on Web Services (ICWS). Washington, DC, USA, 2011:243-250
 [3] Wijesiriwardana C, Ghezzi G, Gall H. A guided mashup framework for rapid software analysis service composition [C]//2012 19th Asia-Pacific Software Engineering Conference (APSEC). Hong Kong, China, 2012:725-728
 [4] Kim J S, Yang M H, Hwang Y J, et al. Customer preference analysis based on sns data [C]//2012 Second International Conference on Cloud and Green Computing (CGC). Hunan, China, 2012:609-613
 [5] Sachan M, Contractor D, Faruque T A, et al. Using content and interactions for discovering communities in social networks [C]// Proceedings of the 21st international conference on World Wide Web. Lyon, France, 2012:331-340
 [6] Kukreja N. Winbook: a social networking based framework for collaborative requirements elicitation and winwin negotiations [C]// 2012 34th International Conference on Software Engineering (ICSE). Zurich, Switzerland, 2012:1610-1612
 [7] 赵祺. 富客户端运行环境自适应中间件研究[D]. 北京:北京大学信息科学技术学院软件研究所,2012
 [8] 杨杰. 基于软件体系结构的网构软件组装技术研究[D]. 北京:北京大学信息科学技术学院软件研究所,2007
 [9] A mashup tool for creating personalized, web-based and service-oriented mashup systems in the way of WYSIWYG [OL]. [2013-05-30]. <http://www.github.com/sakinijino/imashup>
 [10] 陈一舟. 人人网用户量今年将达 2 亿 [EB/OL]. [2012-02-14]. <http://www.techweb.com.cn/it/2012-02-14/1151875.shtml>