基于 Canvas 的共享服务流程的研究

吴莹莹,梁正和
(河海大学 计算机与信息学院,江苏 南京 211100)

摘 要:随着信息化技术的快速发展,企业所积累的信息资源持续增长,各企业都在各自的业务支撑领域建立了各式各样的信息系统。人们急需将这些不同部门不同功能的软件系统集成起来,实现信息共享。文中对实现共享服务的方法进行了研究,通过实现一个组装流程将获取不同应用系统中数据的各种方法组装成服务,以实现系统之间数据的交互。基于 HTML5 新特性 Canvas,实现了在浏览器中通过拖动组件的方式组装服务流程,无需安装任何插件,采用异步刷新的方式完成对浏览器的更新。利用直接调用和基于消息方式的调用两种方法,实现了对不同服务接口的统一调用,并利用超级代理的方式对各个组件实现统一调用,有效地降低了逻辑与事务处理之间的耦合度,成功解决了不同服务需要调用不同服务器上的不同组件方法,实现系统之间的数据共享。

关键词:系统集成;共享服务流程;Canvas;服务组装;超级代理
中图分类号:TP311.133.1 **文献标识码**:A **文章编号**:1673-629X(2019)08-0012-06
doi:10.3969/j.issn.1673-629X.2019.08.003

Research on Shared Service Process Based on Canvas

WU Ying-ying, LIANG Zheng-he
(School of Computer and Information, Hohai University, Nanjing 211100, China)

Abstract: With the rapid development of information technology, the accumulated information resources of enterprises continue to grow, and various enterprise have established various information systems in their business support fields. There is an urgent need to integrate these software systems with different functions of different departments for information sharing. Therefore, the method of implementing shared services is studied. By implementing an assembly process, various methods for acquiring data in different application systems are assembled into services to realize data interaction between systems. Based on the new feature of HTML5, Canvas, the assembly process into a service flow by dragging components in the browser without installing any plug-ins is realized, and the browser in an asynchronous manner is updated. The unified invocation of different service interfaces is realized by using direct invocation and message-based invocation. The super-agent is used to implement unified calling for each component, which effectively reduces the degree of coupling between logic and transaction processing and successfully solves the problem that different services need to call different component methods on different servers. Finally, data sharing between systems is realized.

Key words: system integration; shared service process; Canvas; service composition; super agent

0 引 言

随着计算机技术的发展,信息化程度不断提高,企业的信息资源在积累中持续增长。因此,各企业需要建立各种各样的信息系统来保存大量的数据,达到降低企业的运行成本,提高工作效率的目的[1]。而这些信息系统建立在不同的时期,各个时期的信息技术发展水平不同,因此造成了不同时期各企业系统之间的巨大差异,各系统之间的信息交互也十分困难,难以实现信息系统之间的高效集成与交互[2]。

随着信息资源的日益增长,企业急需将各类信息系统进行集成,实现信息共享。对于大多数企业的应用系统,在最开始设计的时候只需要满足当前需求即可,而没有考虑到未来的发展需要。而随着企业业务的变化和发展,企业对系统在集成方面提出了更加严格的要求。因此,由于系统用户需求的变化,导致了系统差异性在集成的过程中出现了多种困难。而 SOA (service oriented architecture) 正是可以满足整合数据需求的有效方法之一[3]。基于 SOA 的思想,应用系统

收稿日期:2018-09-20 **修回日期**:2019-01-23 **网络出版时间**:2019-03-27
基金项目:国家自然科学基金(61272543)
作者简介:吴莹莹(1993-),女,硕士研究生,研究方向为分布式系统;梁正和,副教授,博士,研究方向为 EPR、分布式系统。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20190327.1629.044.html

相关的业务功能被封装成服务,这些服务在经过组装之后可以完成相应的业务流程。SOA 对各个模块通过分解整理再组合的方式,完成对资源的复用。企业子系统以及各企业间数据共享与交互是其架构实施过程中的重要部分。因此,可以通过共享服务的方式,将获取不同应用系统中数据的方法组装成服务,实现系统之间的服务共享。

文中对实现共享服务的方法进行研究,通过实现一个组装流程,即将获取不同应用系统中数据的各种方法组装成服务,以实现系统间数据的共享。基于 HTML5 新特性 Canvas,实现了在浏览器中通过拖动组件的方式组装成服务流程,并不需要安装任何插件[4]。不同的服务需要调用不同服务器上的不同组件方法,传入的参数类型和个数不一致,调用方法的返回值及返回类型有所不同,组件的协议也有所区别,因此文中采用了超级代理的方式对各个组件实现统一调用。

1　共享服务流程的组成和原理

对信息系统进行集成,实际上可以理解为对数据实现共享。传统实现互联的方式是在系统双方之间实现一对一的连接,每个系统都需要与其他任意一个系统建立连接,系统越多,工作量越大,时间和成本耗费越高。随着数据的大量增长,这种系统的集成方式显然是不可用的。

文中将应用系统中获取数据的方法组装成服务流程,通过图形化流程的设计,完成对共享服务的组装。服务设有统一的访问接口,能够相对独立、完整地完成对数据的访问。系统整体架构如图 1 所示。

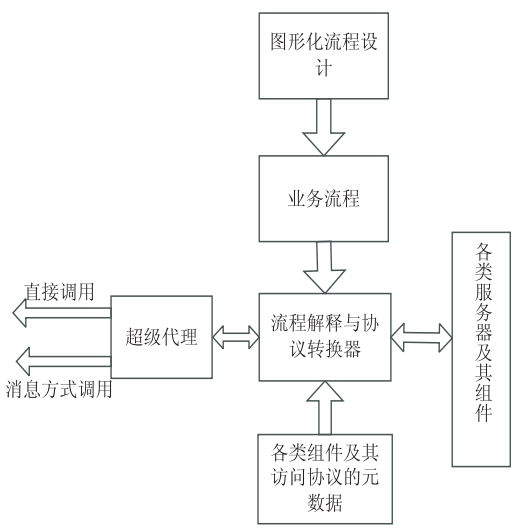

图 1　系统架构

1.1　Canvas 绘图技术

相对于 SOA,HTML5 起步较晚,对使用 HTML5 图形流程绘制的研究并没有 SOA 多。HTML5 拥有跨平台、多设备,无缝集成,及时更新等特点,特别是给图形化业务流程的开发提供了便捷的方式。

Canvas 元素起初是由苹果公司的 Safari 浏览器引入,目前得到了大部分主流浏览器的支持[5]。Canvas 标签是 HTML5 新增的 API 之一,Canvas 拥有大量的绘图 API,能够完成各种图形的绘制[6]。

Canvas 可以认为是放置在网页上的一张“画布”,可以在“画布”上进行“创作”。区别于传统的基于 SVG 的矢量绘图技术,Canvas 属于像素位图的绘制技术。通过 JavaScript 脚本语言可以对 Canvas 实现位图像的动态渲染,完成对图像内容的各项操作[7]。Canvas 可以绘制矩形、文本、路径等,同时可以对图像实现引用、渐变等多种效果[8]。浏览器对 Canvas 的解析与其他标签一样,能够通过 CSS 和 JavaScript 完成对图形的绘制[9]。

浏览器在展示前台页面的时候,通常需要借助视频插件来实现丰富绚丽的网页内容,例如 flash、Silverlight 等[10]。但是安装插件不仅会影响用户体验,而且会增大数据量的传输,降低了运行速度和系统的性能。而 Canvas 绘图则无需安装任何插件,不仅占用系统资源少,而且在传输方面节省了带宽,减少了系统的压力。

1.2　服务流程的组装

一个服务是由多个组件通过服务流程组装而成,服务的组装则是将已有的各种组件方法组装而成,并达到可复用的目的,满足业务需求,实现业务功能。在不同企业,或是企业的不同阶段,采用服务流程进行组装的形式可以迅速对系统进行调整,快速满足需求,避免企业因业务需求的变动而对系统进行重构的问题,降低企业成本。

在服务的调用过程中需要进行传参,包括全局变量、参数变量、转向以及活动。在一个服务中,传过来的参数也可以传递给服务中的其他组件。其中,全局变量和参数变量既可以是 Double、Boolean、LinkedList 等基本包装类,也可以是其他类。转向有三种形式,分别是值条件、无条件、逻辑条件。值条件和逻辑条件只有满足一定的条件才可以向下执行,无条件则是直接向下执行。活动则包含多种,例如:发送消息、接收消息、单个调用、顺序执行、二值分支、多值分支、条件判断、赋值操作。

系统中的用户可以通过服务名和参数列表完成对服务的统一调用,使用方便,但同时也面临着安全风险,对此需要对用户的访问权限进行限制。该系统采用对用户进行特定服务的授权方式来限制用户对服务的访问,只有被授权的用户才可以调用特定的服务。用户通过正确的服务名、参数列表、用户名、用户密码

完成对服务的调用,否则无法访问其他服务。在数据库中,通过建立数据表 user_setting 对指定用户设定的服务访问权限进行保存。其中,系统对用户以及服务的授权管理主要包括:新建用户、修改用户、删除用户、给用户添加服务授权、删除选择的服务授权以及获取用户授权的数据服务。

1.3 超级代理服务

在调用组件方法的过程中,由于传参的类型和个数、调用方法的返回值和返回类型、组件的协议有所区别,想要对组件方法实现统一调用,需要通过超级代理的方式来完成。超级代理是一个位于客户端和服务器之间的转发机构,负责各种不同协议的转换,以及各类组件调用的反射机制[11]。超级代理可以分为两个部分,一个代理主机和各个服务器上的子代理,针对不同的服务器,采用不同的超级代理完成调用。这样不仅可以有效减少事务处理和逻辑之间的耦合,而且便于程序日后的维护。

在文件系统中,数据包含了元数据和数据。数据是指文件中的真实数据,而元数据则是用来描述文件特征的系统数据,也可以说是用于提供某种资源的有关信息的结构数据[12]。元数据具有简便易用性、可复用性、可操作性、充分性、交互性、模块性、跨平台和跨库的数据对比、集成和复用等特性[13]。实现各类组件以及访问协议的元数据是实现超级代理的基础。元数据具体包括:组件接口元数据、组件类型元数据、组件部署元数据和协议元数据。可能的组件类型有:java 类、FTPserver、EJB2、EJB3、邮件服务、短信服务等。各类协议包括:ftp、tcp、http、smtp、pop3、udp 等。不同类型的组件对应的元数据也有所不同。例如:消息服务组件的元数据包括服务器 URL、服务器用户名、服务器密码、消息工厂用户名和密码;java 类组件的元数据包括:java 类名、类功能描述、jar 包名称、主机地址。因此,组件类型不同,需要建立的元数据类也不同,需要用统一的目录对象对所有类型的组件元数据类进行包装。

在一个服务流程的过程中,通过依据组件的元数据以及传入的相关参数来调用组件的接口。当超级代理调用组件的方法时,通过对组件元数据、组件实例的参数值和变量值进行调用。组件按照各自的方法通过反射机制、组件的名称、接口、协议、组件的部署、参数值和变量值完成调用。

2 共享服务流程的实现

2.1 流程组装服务

对服务的调用包括直接调用和基于消息的方式调用。调用方式由工厂类的静态方法创建一个服务的类并返回。

在 serviceExeByName( ) 方法中,serviceName 是服务名称,parameterValList 是参数列表,依据服务名来进行方法的调用。userName 和 password 分别是被授权用户的账户名和密码。

serviceExe( ) 方法则是按照服务的唯一标识来进行调用的。例如服务名为 ftpclienttest,参数列表为 serParameterValList,授权的用户名为 Jojo,被授权的用户密码为 123456,服务返回一个二进制结果对象。

服务流程组装模块采用 Canvas 的类库 Zrender,它提供了生动、直观、可交互以及个性化定制的流程图的绘制[14]。Zrender 拥有数据驱动、完整的事件封装、高效的分层刷新、丰富的图形选项、强大的动画支持、易于扩展等优点。用户可以在浏览器中通过对图形元素进行增、删、改等操作,并将变化了的数据更新到应用服务器。可视化架构如图 2 所示。

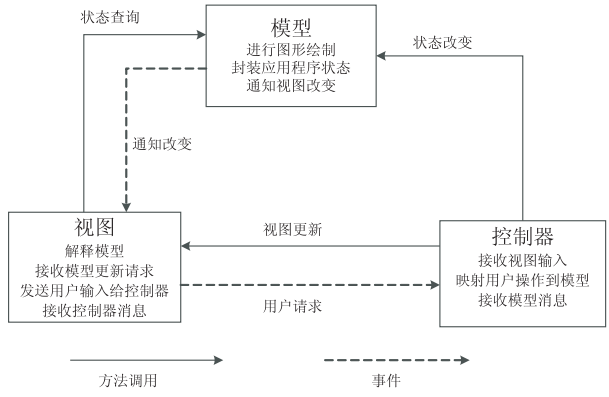

图 2 可视化架构

2.1.1 异步刷新

Ajax 的主要应用是对页面进行局部刷新,也就是对网页上其中一个标签里的内容进行刷新,从而不影响整体的效果。最常见的例子莫过于 Google 地图,对用户来说,实时的异步刷新既快速又方便。在该系统的服务组装过程中,也需要用到异步刷新。当用户拖动图形元素到工作区时,浏览器会自动生成箭头用来连接所选的图形元素,协助开发人员完成流程组装。当用户调整图形元素的位置时,箭头会随即进行相应的调整,并将信息发送至服务器。此时只有被操作的内容发生变化,而不是整个页面进行刷新。这部分功能主要由 serviceShareFlow. js 完成,主要功能描述如下:

在 serviceReqFlow(cmd, paraRst) 方法中:第一个参数是操作标识,有四种:“selectedSa”、“selectedFt”、“move”、“reSelectedSa”;第二个是参数结果集,通过“post”形式将前端的数据发送给“serviceFlowServlet”流程控制器,服务器会根据接收到的参数做出相应的处理。若服务器的状态发生改变,则调用

serviceChangeFlow( ) 方法。serviceChangeFlow( ) 方法中含有 5 个参数,“serviceId”指服务唯一的标识号,“serviceType”指“活动”或“转向”,“objId”指被选中图形的 Id,“x”与“y”分别为图形所在的坐标值。当服务器状态改变时,调用 serviceChangeFlow( ) 方法。在此方法中,当网页返回的状态码显示为 200 时,即服务器已经成功处理了相应的请求,并返回 xml 页面对结果进行解析。

2.1.2　流程图绘制

流程图主要包括两种元素,分别是连接弧元素和图形元素。图形元素指的是除了连接线以外的其他形状,例如:表示执行操作的矩形,表示条件判断的菱形等,而连接弧则是用来连接这些图形元素的连接线。在实现过程中,将 Unit 类作为基类,其他元素都是 Unit 的子类,继承自 Unit 类。系统的整个流程管理是通过 serviceFlowGraph 类来实现的。

将 zrender 的 dragable 属性设置为 true,即可对图形元素进行拖动,但是图形元素之间的连接线并不会随着图形的拖动而移动。因此,还需要建立一个监听机制 zrenderListener. on( ),用 dragingActivity( ) 来定义当前正在拖放的节点,包括对开始事件、结束事件的拖动,用 onUnitMouse( ) 来定义鼠标的移动事件。另外,每个图形元素拥有八个连接点,连接弧并不是必须和某个图形元素相连接的,而是在连接过程中按照图形元素所在的位置信息,找到最靠近的那一个点。这里用 getAnglePosition( ) 方法来获取角点的所在位置,用 refreshActivityTransitions( ) 来定义对连接弧进行刷新的方法。

2.1.3　流程控制器

该系统采用 MVC 的设计模式[15],serviceFlowServlet 作为流程组装的 Controller,相当于一名调度者,根据“init”标识和服务的唯一标识号,调用 serShowFlowXml( ) 方法,将服务流程页面组装成 xml 页面,经过 JavaScript 解析后再显示到页面。

一个服务包括四个部分:参数变量、全局变量、活动和转向。活动和转向是共享服务流程组装的主要部分。若要对转向和活动进行操作,则向 Controller 传入相应的服务标识号、对象类型(转向或活动)、对象 id、移动后产生的新的 x、y 坐标,根据对象的类型不同做出相应的处理,将活动或转向移动到新的位置。serShowFlowXml( ) 方法中传入的参数分别是服务的 id、是否是选中状态、活动的 id、转向的 id。如需要添加活动,则根据活动的 id 和服务唯一的标识号,调用该方法将服务流程显示到页面,传入的参数分别为 serviceId、true、serActivityId、null;当需要拖动转向对流程有所操作时,传入的参数为 serviceId、true、null、serTransformId;当删除一个服务时,也需要调用该方法进行显示。

2.2　超级代理服务

文中通过 SuperAgent 类来实现超级代理的服务,其中 serGroupMethodExecute( ) 方法完成了对组件方法的调用。该方法需要传入的参数有:服务活动的二进制对象、需要调用的组件对象的二进制数据、服务实例化的标识号、组件参数列表的二进制数据、参数值集合的二进制对象、对象转换规则的二进制对象。

除去参数服务实例化的标识号以外,当其他五个参数不为空时,转换成原对象。转换后,若服务的活动、服务实例号和组件对象存在,并且服务对象必须是系统中已经存在的组件的形式,此时,传入服务的活动对象、组件对象、组件参数列表以及参数值,通过 serRepresentMethods 类直接调用相应组件的调用方法,结果返回给 returnObject。

超级代理通过调用应用服务器的组件方法把获取到的对象封装在本地对象中,被访问的数据对象在调用者系统中并不存在,通过 transferObjByRule( ) 方法可以解决这个问题。一种方式是将当地对象转换为统一的标准对象,另一种方式是将本地对象转换为 Json 中间对象。Json 对象可以转换成系统可以使用的对象,以此来实现异构对象的转换。因此需要判断 returnObject 是否需要转换,如果需要转换,并且转换规则不为空时,直接调用 transferObjByRule( ) 方法,转换后的对象赋给 returnObject。当服务对象不是 javaclass 等已存在的组件形式时,调用失败。当服务、组件对象、服务实例号中的某个参数为空时,服务调用失败,具体流程如图 3 所示。

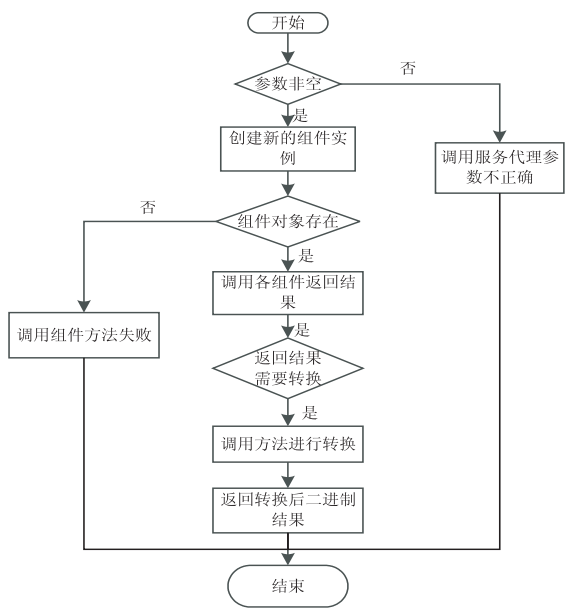

图 3　组件方法的调用流程

超级代理调用了不同应用服务器的组件方法,同

时各个应用服务器上又安装了超级代理组成了子代理层,因此系统也需要对应用服务器进行管理。对应用服务器的管理主要通过 ServerInfoReginBean. java 类完成,该方法会在构造器中获取到应用服务器的连接,主要方法及功能如表 1 所示。

3 实验结果与分析

系统通过在浏览器中拖动和编辑各图形元素,完成对服务的组装。用户对服务的调用方法之一是直接调用,这种方式下不需要接收消息,其业务流程如图 4 所示。该图中定义的服务具有三个参数,一个全局变量 returnVariable,两个局部变量 parameterValue 与 inputData。该服务包括条件判断、调用 java 类方法、对变量赋值、对返回值赋值等方法。调用该服务后会得到一个返回值,返回变量是 returnVariable。

表 1 应用服务器管理相关方法

方法名	具体功能
serverInfoTableSelectAction	获取所选应用服务器的详细信息
addServerAction	新建应用服务器登记
serverSettingAction	修改应用服务器登记
serverConfirmAction	确认新建、修改应用服务器登记
delServerAction	删除所选应用服务器登记
delServerConfirmAction	确认删除选择的应用服务器登记
serverAgentInsTestAction	应用服务器安装代理测试功能

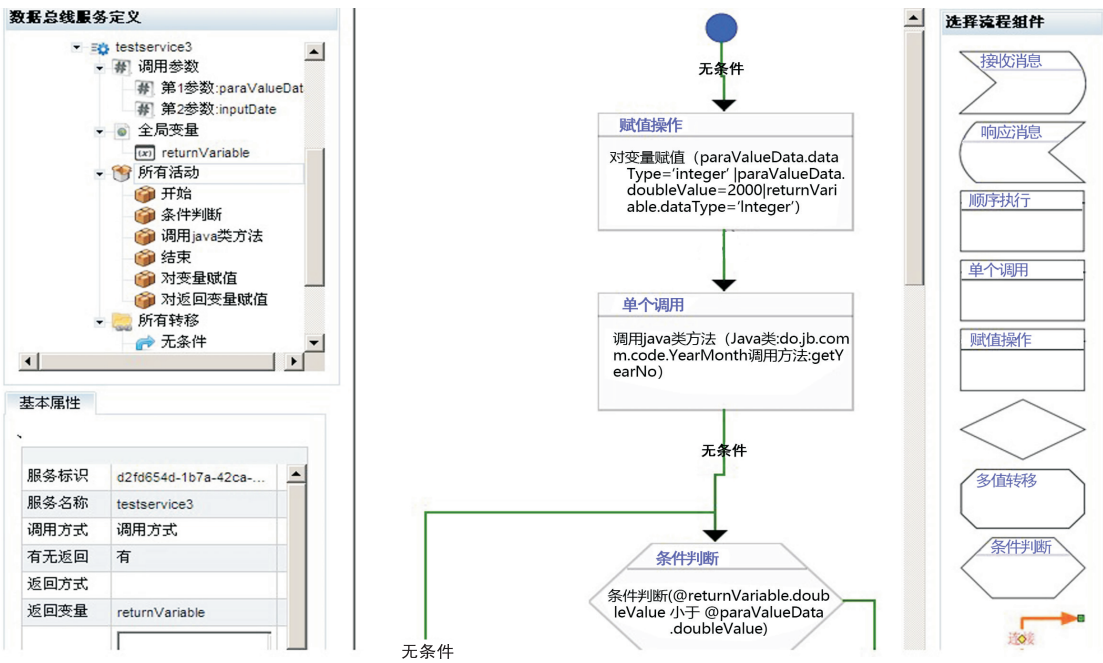

图 4 直接调用服务的组装流程

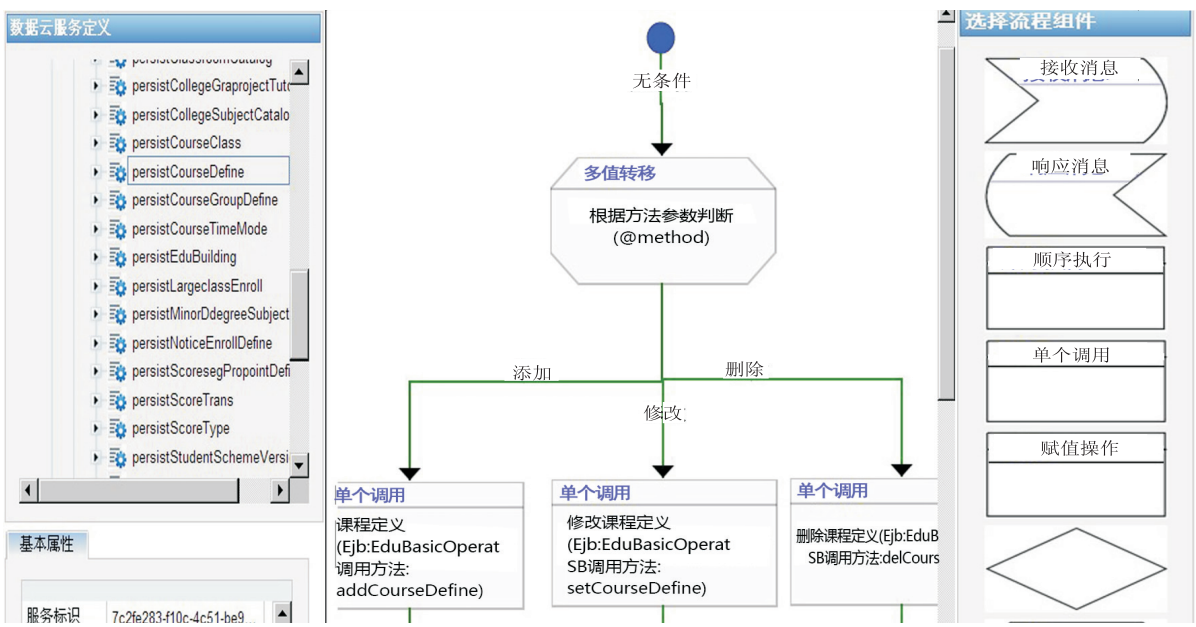

图 5 基于消息方式调用服务的组装流程

图 5 是基于消息方式进行调用的,展示了一个定义好的实现数据对象持久化的服务。该服务可以调用 EJB 组件对课程进行添加 addCourseDefine、修改 updCourseDefine 和删除 delCourseDefine,并实现数据对象持久化功能。

4　结束语

文中采用共享服务的思想,将获取不同应用系统中数据的方法组装成服务,采用 Canvas 绘图技术实现图形化流程的设计,根据用户需求自定义选取、编辑组件元素,灵活组装,完成共享服务流程的组装。当图片发生变化时,通过异步刷新的方式完成对浏览器的更新。为了保证系统的安全性和可靠性,对用户进行特定服务的授权方式来限制用户对服务的访问。利用直接调用和基于消息方式的调用两种方式,实现了对不同服务接口的统一调用。最后,使用超级代理完成对组件方法的调用,实现不同协议之间的转换以及各类组件调用的反射机制,有效地降低逻辑与事务处理之间的耦合度,使其不受组件的类型和协议等的影响,最终实现基于 Canvas 的共享服务流程的数据共享。

然而,系统仍然存在不足之处,比如流程的设计上还不够美观、合理,超级代理还可以调用更多的应用服务器的组件方法等,这些问题在以后的学习过程中会加以改进,使系统更加完善。

参考文献:

[1] 杨宇东. 基于信息化的企业业务流程再造[J]. 中国管理信息化,2010,13(11):59-61.

[2] 费　奇,余明晖. 信息系统集成的现状与未来[J]. 系统工程理论与实践,2001,21(3):75-78.

[3] 王卫星,王晨光. 基于 SOA 的企业信息系统集成框架[J]. 计算机工程,2010,36(18):29-31.

[4] 徐卓揆. 基于 HTML5、Ajax 和 Web Service 的 WebGIS 研究[J]. 测绘科学,2012,37(1):145-147.

[5] 李　龙. 基于 HTML5 技术的实时图形图像处理研究[J]. 信息通信,2017(6):134-135.

[6] XIE Yue. Transmission and display technology for vital signs based on HTML5 canvas and COMET machanism[C]//International conference on computer science & network technology. Harbin,China:IEEE,2012:2760-2763.

[7] 丁立国,周　斌,熊　伟. 基于 Html5 的 Web Map 矢量渲染技术研究[J]. 测绘工程,2017,26(8):62-67.

[8] BOULOS M N K,WARREN J,GONG J,et al. Web GIS in practice VIII:HTML5 and the canvas element for interactive online mapping[J]. International Journal of Health Geographics,2010,9(1):14-15.

[9] 陆　钢,区洪辉,梁柏青,等. 面向移动终端的 HTML5 应用运行环境研究[J]. 电信科学,2013,29(5):40-44.

[10] 丁　宏. Flash 动画在网页制作中的应用与发展探讨[J]. 计算机光盘软件与应用,2012(11):162-163.

[11] 赵宏利,李秀冰,李大林. 基于反射机制的插件系统软件设计[J]. 计算机工程与设计,2010,31(2):348-351.

[12] LINKERT M,RUEDEN C T,ALLAN C,et al. Metadata matters:access to image data in the real world[J]. Journal of Cell Biology,2010,189(5):777-782.

[13] 黄如花,邱春艳. 国内外科学数据元数据研究进展[J]. 图书与情报,2014(6):102-108.

[14] 任建文,江贤康,崔　悦. 基于通用电网框架模型的电力系统图形自动绘制实现[J]. 电力系统保护与控制,2012,40(7):138-142.

[15] YUAN M L,HUANG Y B,HUANG J L,et al. The research and application of MVC Software architecture based on J2EE[J]. Application Research of Computers,2003(3):147-149.