

基于工作流的服务组合在电子政务中的应用

文俊浩¹, 饶锡如^{1,2}, 何盼¹, 王玉标¹

(1. 重庆大学 计算机学院, 重庆 400044; 2. 重庆市汽车运输集团 信息中心, 重庆 400010)

(xile@163.com)

摘要:为了解决电子政务系统建设中存在的“信息孤岛”问题,提出了一个基于工作流实现服务组合的系统框架。该框架通过流程服务中心的工作流引擎对遗留系统发布的 Web 服务进行组合,实现了整合遗留系统和跨部门协同办公,从而能够有效地满足业务用户的需求,以及实现对敏捷业务需求的快速响应。最后以跨部门电子政务系统验证了该框架的合理性,结果表明是可行和有效的。

关键词:面向服务架构;工作流;服务组合;电子政务;遗留系统

中图分类号: TP311 **文献标志码:** A

Application of workflow-based service composition in E-government

WEN Jun-hao¹, RAO Xi-ru^{1,2}, HE Pan¹, WANG Yu-biao¹

(1. College of Computer Science, Chongqing University, Chongqing 400044, China;

2. Information Centre, Chongqing Automobile Transportation Group, Chongqing 400010, China)

Abstract: To solve the information silo problem in the construction of E-government system, the authors put forward a frame based on workflow technology to implement service composition, which set up a process service center to compose the service of legacy system and implement the integration of legacy system and trans-departmental cooperative work. In this way, it could be suitable for the individual demand and realize quick response for business agility. Finally, a trans-departmental E-government system verifies the rationality of that frame, which suggests it is feasible and effective.

Key words: Service Oriented Architecture (SOA); workflow; service composition; E-government; legacy system

随着信息技术的高速发展,各级政府部门纷纷建立了自己专属的电子政务系统,但这些政务系统当初只是针对各部门自身的业务需求来设计的,这就造成各个政务系统之间独立建设,开发时所采用的开发环境或应用平台不尽相同,导致各个系统之间彼此异构难以交互,信息资源分割,各自为政,形成大量信息孤岛^[1]。而现今所倡导的电子政务“一站式”服务强调以公众需求为中心构建一个透明的系统,以达到整合政府职能,优化业务流程,提高政府机关办事效率的目的。可见,传统的电子政务系统难以满足“一站式”服务的需求。如何在减少开发与改造成本的基础上有效整合现有政务遗留系统,实现跨部门的政务协同办公成为当前电子政务研究的热点问题。

针对上述存在的问题以及“一站式”电子政务建设目标,利用基于工作流的服务组合设计思想,可有效解决由于系统异构造成的“信息孤岛”问题,实现系统间的分布式协同应用和业务流程柔性搭建^[2]。本文为此设计了一个跨部门协同电子政务的实现框架,并在实际应用中得到了较好应用。

1 相关技术

1.1 SOA 及其实现技术

面向服务架构(Service Oriented Architecture, SOA),是一种分布式软件模型和设计方法,服务是封装成可重用组件的形式,因此服务之间可以进行组合以构建新的服务,通过服务的组合,得到更粗粒度的服务。SOA 的这种设计思想对于遗留系统的集成具有得天独厚的优势^[3]。

Web 服务即 Web service,它是 SOA 的主要实现技术,它提供了在互联网松耦合环境下完成应用之间互操作和集成的技术框架。Web 服务向外界暴露出能够进行调用的 API^[4], Web 服务通过 URL 来确定服务的位置,通过 WSDL 作为其服务接口描述语言,通过 UDDI 协议规范进行 Web 服务的注册和服务查找定位,并借助 SOAP 进行通信。Web 服务技术具有分布式、松散耦合、跨平台性与高度集成性等优点^[5],它的高度灵活性使系统部门间的功能集成与协同变得容易实现。

1.2 工作流技术

工作流,就是将一个工作流程中的业务或数据按照一定的逻辑和规则前后组织在一起,实现工程流程的自动化或半自动化处理,使流程构建具有柔性化的特点。工作流引擎是工作流系统的核心,它用来创建新的流程,根据过程定义生成工作项,支持活动的执行以及记录工作流的相关信息^[6]。工作流引擎通常需要满足如下核心功能:1)能够解释建模工具生成的工作流模型并实例化流程;2)用户可以通过控制端对整个流程引擎进行监控;3)在流程执行过程中,可以根据流程定义调用相关应用程序的接口以及与外部资源的交互;4)维护工作流相关数据及操作日志。

2 跨部门协同电子政务系统设计

2.1 跨部门协同电子政务系统设计目标及方案

本文所设计的跨部门协同电子政务系统方案,是考虑到各个部门还需要保留其原有的政务应用系统,不能对原有系统的实现结构与逻辑结构做大的调整,要求以最小的开发代价来实

收稿日期:2009-03-20;修回日期:2009-05-07。 基金项目:国家科技支撑计划项目(2007BAF23B0302)。

作者简介:文俊浩(1969-),男,河南临颍人,教授,博士,CCF 高级会员,主要研究方向:服务计算与面向服务的软件工程; 饶锡如(1979-),男,重庆人,硕士研究生,主要研究方向:服务计算与面向服务的软件工程; 何盼(1984-),女,重庆人,博士研究生,主要研究方向:服务计算与面向服务的软件工程; 王玉标(1981-),男,江西婺源人,硕士研究生,主要研究方向:服务计算与面向服务的软件工程。

现系统的升级改造。在这基础上引入 SOA 架构,只需对原有系统做少量扩充就能无缝过渡到电子政务跨部门分布式协同应用,从而实现“一站式”政务服务的需求。本设计方案是通过设立流程服务中心,以达到分布式流程统一管理的目的,该服务中心由熟悉领域业务规则并且具有相应权限级别的政府工作人员通过系统提供的 GUI(图形用户界面)对工作流引擎进行监控维护,流程服务中心提供服务注册、流程定义、流程监管、流程信息发布等功能,在整个跨部门协同电子政务系统中处于核心地位。另外,各级政务部门的政务系统中将其领域内的业务处理封装成服务并向外界公开,比如政务办事流程、信息发布、项目审批、信息检索等,这些公共服务通过提供统一的 Web 服务接口,以电子政务公共服务资源的形式注册到流程服务中心,由服务中心对这些公开的服务编排组合成新的业务流程,部门政务系统通过查询流程服务中心获得整个跨部门流程的逻辑结构,利用 Web 服务穿越防火墙的优势,实现跨网络的分布式应用目的。在系统实际应用中,用户可以通过任意政务门户获得的全域的政务信息资源,实现了信息共享和跨部门协同办公,使物理上分割的政务系统之间形成了具有逻辑关系的业务流程,这与“一站式”政务服务的理念相符合。整个跨部门协同政务系统逻辑结构如图 1。

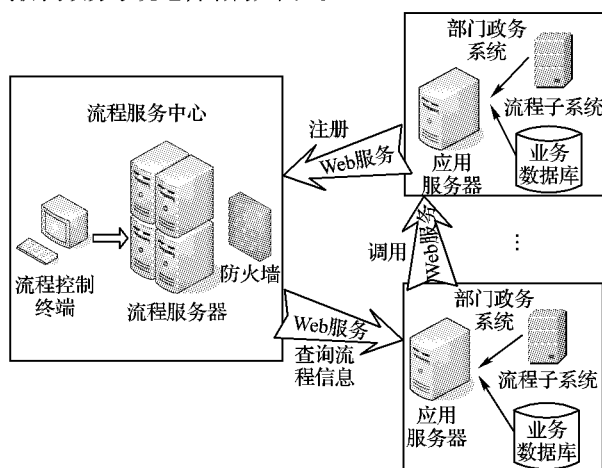


图1 跨部门协同政务系统逻辑结构

2.2 跨部门协同电子政务系统实现框架

根据跨部门协同政务系统的逻辑结构,本文设计了跨部门协同政务系统的实现框架。该实现框架分为四个层次,分别为:遗留系统、Web 服务层、Web 服务组合层、业务应用层。如图 2 所示,下面分别对各层作简要分析。

1) 遗留系统。这些物理上分割的孤立政务系统是整个系统的基础资源,包括数据资源、业务资源、硬件资源以及网络设施等。它们有各自完善的业务流程子系统,它们是处于一种封闭的运行环境之下,系统彼此之间不能互联互通,而且各个遗留系统的开发环境和运行环境都不尽相当,存在系统之间的异构问题。

2) Web 服务层。该层是将遗留系统的业务处理功能通过 Web 服务包装成为一个个独立的业务服务,它是业务服务的提供者。在系统应用中,通过了验证并具有合法权限的遗留系统将自己业务相关的操作以 Web 服务的方式注册到服务信息库中,需要注册的信息有:Web 服务的 URL 地址、Web 服务所属部门、Web 服务的接口信息和 Web 服务的语义描述等。

3) Web 服务组合层。该层提供跨部门协同政务系统实现的支撑环境。该层主要是通过动态、灵活地组合所需服务来实现业务流程的柔性构建。工作流引擎在此层将孤立的 Web 服务组合成新的业务流程,并向外公布其调用接口信息,工作流

引擎根据业务规则从服务信息库中查找业务相关的 Web 服务,运用流程定义工具,进行流程建模,一般是以图形化的界面表示出流程信息,可通过 Petri 网技术对业务流程图的活性和有界性等进行验证,以保证流程是可执行的。最后,通过映射工具将流程的业务模型映射成格式统一的流程信息并存入流程信息库中,日志管理将对操作进行记录,当工作流引擎运行过程中出现异常,将启动异常处理程序,根据日志信息对异常进行处理,保证流程信息的完整性、安全性和冗错性。

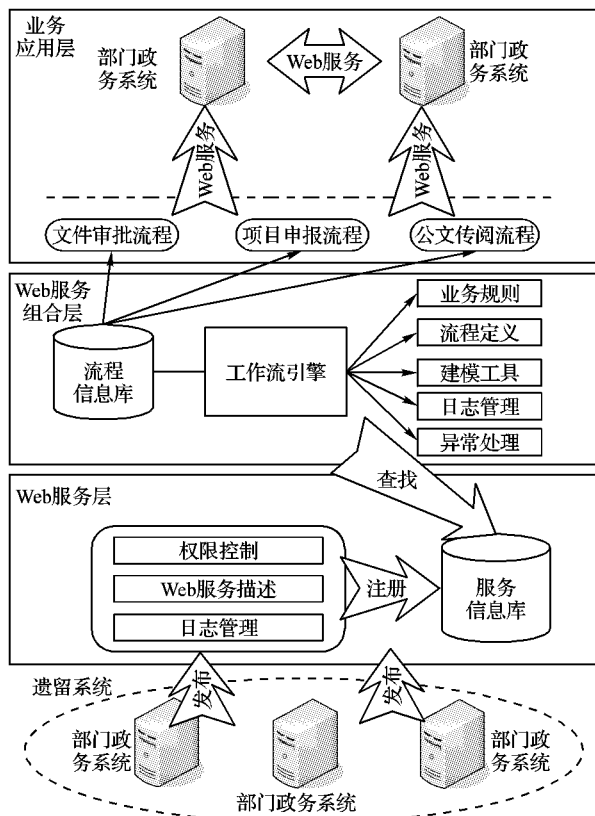


图2 跨部门协同政务系统实现框架

4) 业务应用层。流程信息库将组合后的流程信息,如文件审批流程、项目申报流程、公文传阅流程等以 Web 服务的方式向外界公开,部门政务系统通过调用流程信息服务获得业务逻辑信息,根据流程信息找到执行每一步业务流程所需调用的 Web 服务地址,通过该服务地址与之对应的政务系统进行交互,从而实现政务遗留系统之间的松耦合集成,达到跨部门协同政务的目的。

2.3 跨部门协同电子政务系统对比传统政务系统的优势

1) 采用通用协议标准如 XML,可以适用于分布式异构环境,减轻系统集成的难度和复杂性。

2) 灵活的流程控制,能动态地适应政务流程重组的要求,使工作流搭建具有柔性。当政务流程发生改变时,只需要用流程建模工具重新设计流程定义,存入流程信息库中,就可产生新的流程实例供外部系统调用。

3) 松散耦合。不依赖于使用的平台和编程语言,保证了可互操作性。

4) 穿越防火墙。Web 服务使用了 HTTP 作为底层的传输协议,从而避免了防火墙问题。

3 系统实现

3.1 构造服务

本文在实际开发中使用的是 MyEclipse6.5 自带的 JAX-WS(Java API for XML Web service)工具进行 Web 服务的开

发。JAX-WS 属于 Sun 最新开发的 Java EE 平台的一部分, JAX-WS 在使用上比较简单,符合面向服务的开发需求。

3.1.1 遗留政务系统端

遗留政务系统根据跨部门协同电子政务业务需求,需要发布以下主要 Web 服务。

1) FileSubmitService。即文件信息提交服务,远程系统调用此服务将提交的文件信息写入服务提供端的本地数据库中。

2) FileFeedbackService。即文件反馈信息服务,通过此服务可以把文件处理后的信息(如审批结果、材料反馈、文件检索结果等)写入服务申请端系统的本地数据库中,这些反馈信息可作为流程跟踪的信息来源。

3) BusinessProcessService。即业务处理服务,该服务封装了部门系统的业务具体处理功能,不同部门的业务处理不尽相同(如审批、检索、申报、公示和竞标等),但对外都提供统一格式的调用接口。

3.1.2 服务流程中心端

服务流程中心作为跨部门协同电子政务系统的中枢,起到联系各个遗留政务系统,生成跨部门流程实例的作用,提供服务注册,服务组合,流程信息发布等功能。服务流程中心端需要封装以下主要 Web 服务。

1) RegistWsInfoService。即 Web service 信息注册服务,该服务的作用是将外部遗留政务系统发布的 Web 服务信息注册到服务流程中心的服务信息库中。其中该服务接口的接收参数是 WsInfo 的 JavaBean 类型,Wsinfo 是用于维护 Web 服务的相关信息类,该类实现了序列化,方便通过网络进行二进制的传输。

2) QueryFirstStepService。即查询第一步流程步骤信息,该 Web 服务根据流程 ID,查询到第一步的流程步骤信息,将第一步所调用的 Web 服务 URL 以 String 类型作为返回值,该服务将启动一个业务流程实例。

3) QueryNextStepService。即查询下一流程步骤信息,该 Web 服务根据参数流程 Id 和步骤 Id 检索流程服务中心的流程信息库,当流程步骤记录项的 status 为 false 则表示非最后一步,流程将继续流转;当 status 为 true 表明为最后一步,整个流程结束。

3.2 基于工作流的服务组合实现

通过上面介绍的 RegistWsInfoService,在流程服务中心的服务信息库中注册了大量遗留政务系统发布的 Web 服务接口信息,这些零散、彼此没有关联的 Web 服务需要重新组合形成更大粒度的服务接口以形成新的业务流程,才能产生更大的 IT 增值,这也是整个跨部门协同政务系统实现的关键一环。本文采用基于二维关系表的轻量级工作流引擎来实现服务组合,即工作流引擎中的数据模型全部通过关系结构来表达。在该工作流引擎中,Web 服务被当作工作流中的一个活动,利用工作流机制实现对 Web 服务的组合、执行和监控功能。基于二维关系表的轻量级工作流引擎有如下优点:1) 流程相关的数据的完整性以及事务操作的原子性可以由数据库管理系统来维护;2) 可以方便地利用数据库管理系统提供的各种 DML 语句来操纵工作流引擎所需的各种数据;3) 使用数据库异常处理机制,可保证工作流引擎运行异常的有效捕获;4) 利用数据库备份机制,可以保证在出现异常的情况下,各集成应用系统的流程能够得到一致恢复。

由于受篇幅所限,下面仅列出流程引擎部分关系表 ER 图,如图 3 所示。

wsInfo 表存储的是 Web 服务相关信息,主要包括 Web 服务的地址,Web 服务的详细描述(用于服务的匹配查

找),Web 服务的服务提供者信息等。

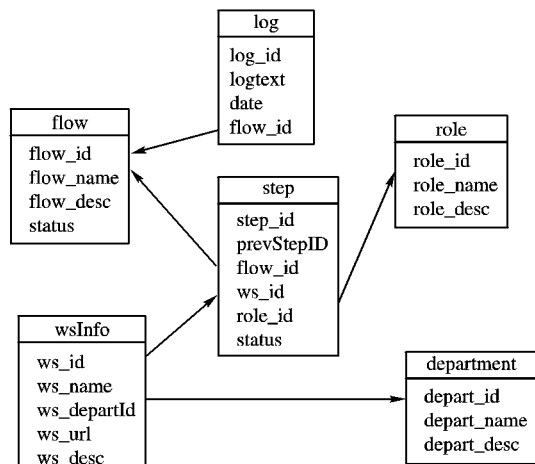


图 3 流程引擎关系表 ER 图

flow 表存储经组合后的流程信息,包括流程名,流程描述等,流程服务中心将 flow 表里的流程信息以 Web 服务的方式向外界公开,供外部调用 flow 列表,并从中选择自己业务相关的流程信息。

step 表存储某个流程的具体执行步骤,以及每个步骤所调用的外部 Web 服务。在这张表中将对 Web 服务的执行顺序进行编排组合。其中 prevStepID 字段标明流程前一步的 step_id,通过该字段可以将每一步骤联系起来;status 字段标明该步骤是否为流程的最后一步,作为流程是否结束的标志。

log 表将存储流程操作日志信息,包括增加、修改、删除以及操作者信息。

图 4 是电子政务中最常见的文件审批流程图,直观地说明了用户启动了一个流程服务后,在轻量级工作流引擎机制下的审批流程执行过程。

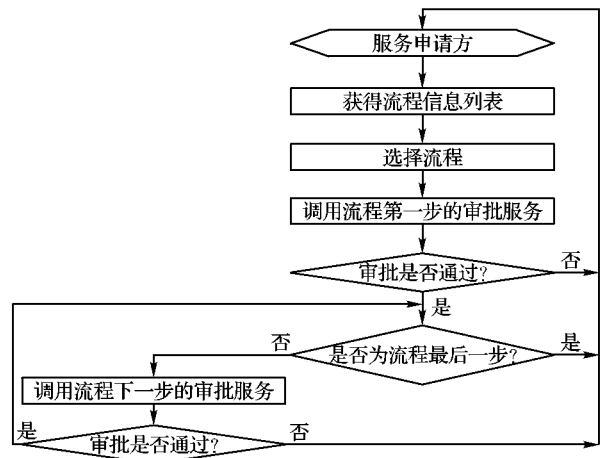


图 4 审批流程

该流程图反映在业务处理 Action 里的代码片段如下:

```
QuerynxyStepService service = new QueryNextStepService();
QueryNextStep Delegate delegate = service.getQueryNextStep Port();
String stepInfo = delegate.queryNextStep(flowId, stepId);
try{
    //通过 DOM 解析 stepInfo, 获得下一步所要调用的服务 URL
    //以及 status
    builder = factory.newDocumentBuilder();
    Document doc = builder.parse(new InputSource(
        new StringReader(stepInfo)));
    ...
    String url = nurl.getFirstChild().getNodeValue();
    Boolean status = nstatus.getFirstChild().getNodeValue();
```

```

If( status == false)           //判断流程是否为最后一步
{
    //对审批文件进行业务处理
    ...
    //调用下一流程步骤的 Web 服务
    BusinessProcessService bps = new BusinessProcessService ( url,
        qname);
    ...
    //调用反馈服务,将处理结果返回到流程发起端系统
    FileFeedbackService ffs = new FileFeedbackService ( sourceurl,
        qname);
    ...
}
else{
    //对审批文件进行业务处理
    ...
    //调用反馈服务,将处理结果返回到流程发起端系统
    FileFeedbackService ffs = new FileFeedbackService ( sourceurl,
        qname);
    ...
}
}

```

3.3 实现结果

根据上述框架具体实现后,在流程服务中心对注册服务进行编排组合,建立流程模型。图形化建模效果如图 5。

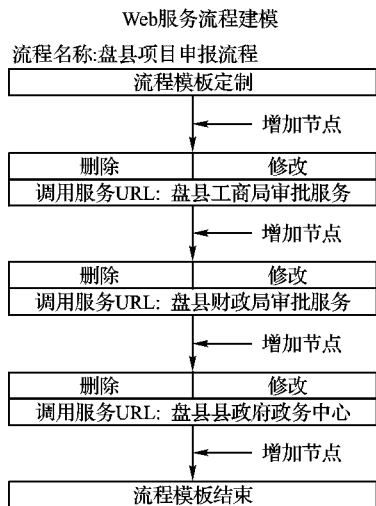


图5 流程动态建模

当系统启动一个流程,并按流程事先定义好的顺序执行完毕后,根据流程状态追踪,生成下面的实现效果,如图 6。



图6 审批流程追踪效果

实践证明,该跨部门政务协同办公系统能达到预定目标,实现了电子政务的分布式集成应用和业务流程的柔性构建,在对遗留系统的改造升级所需成本较低,运行效果良好。

4 结语

本文针对“一站式”电子政务建设目标以及现有政务遗留系统存在“信息孤岛”的实际问题,运用基于二维关系的工作流引擎技术与 Web 服务技术相结合,以服务组合的方式实现了各个政务遗留系统之间的高效整合,达到了跨部门协同办公和流程柔性搭建的目的。并在实践中取得了良好的效果,提高了政府部门为社会公众服务的效率和能力,这对当前政府部门的信息化建设有一定的借鉴作用和参考价值。

参考文献:

- [1] 刘星沙,马华,夏明伟,等.支持跨区域协同的政务审批系统研究[J].计算机工程与设计,2007,28(16):3981-3984.
- [2] WANG SHU-YING, SHEN WEI-MING, HAO QI. An agent-based Web service workflow model for inter-enterprise collaboration[J]. Expert Systems with Applications, 2006, 31(4): 787-799.
- [3] 张一川,汪德帅,刘莹,等.基于业务服务的企业遗留系统集成框架[J].计算机应用,2008,28(6):263-265.
- [4] 迟文学,吴信才,方坤,等.面向服务的搭建式开发技术研究[J].计算机应用研究,2008,25(6):1717-1719.
- [5] LEE S P, CHAN L P, LEE E W. Web services implementation methodology for SOA application [C]// Proceedings of 2006 IEEE International Conference on Industrial Informatics. Washington, DC: IEEE Computer Society, 2006: 335-340.
- [6] 于勇,彭岩.工作流管理系统柔性机制[J].计算机工程,2008,34(24):40-42.

(上接第 2498 页)

- [28] WEIMER W, NECULA G. Mining temporal specifications for error detection [C]// TACAS 2005: Proceedings of the 11th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Edinburgh, UK: [s. n.], 2005: 461-476.
- [29] JOHNSON T A, EIGENMANN R. Context-sensitive domain-independent algorithm composition and selection [C]// Proceedings of the 2006 ACM SIGPLAN Conference on Programming Language Design and Implementation. New York: ACM Press, 2006: 181-192.
- [30] DILLIG I, DILLIG T, AIKEN A. Static error detection using semantic inconsistency inference [J]. ACM SIGPLAN Notices, 2007, 42(6): 435-445.
- [31] ACHARYA M, XIE TAO, PEI JIAN, et al. Mining API patterns as partial orders from source code: From usage scenarios to specifications [C]// Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT

Symposium on the Foundations of Software Engineering. New York: ACM Press, 2007: 25-34.

- [32] THUMMALAPENTA S, XIE TAO. NEGWeb: Static defect detection via searching billions of lines of open source code [EB/OL]. [2009-01-11]. <http://people.engr.ncsu.edu/txie/publications/TR-2007-24.pdf>.
- [33] LO D, KHOO S C, LIU CHAO. Efficient mining of iterative patterns for software specification discovery [C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2007: 460-469.
- [34] YE YUN-WEN, YAMAMOTO Y, NAKAKOJI K, et al. Searching the library and asking the peers: Learning to use Java APIs on demand [C]// Proceedings of the 5th International Symposium on Principles and Practice of Programming in Java. New York: ACM Press, 2007: 41-50.