数字化校园工作流引擎的设计与实现

何 倩[1],孟祥武[1],王 勇[2]
HE Qian[1],MENG Xiang-wu[1],WANG Yong[2]

1.北京邮电大学 网络与交换技术国家重点实验室,北京 100876
2.桂林电子科技大学 网络信息中心,广西 桂林 541004
1.State Key Lab of Networking & Switching Technology,Beijing University of Posts & Telecommunications,Beijing 100876,China
2.Network Information Center,Guilin University of Electronic Technology,Guilin,Guangxi 541004,China
E-mail:treeqian@gmail.com

HE Qian,MENG Xiang-wu,WANG Yong.Design and implementation of digital campus workflow engine.Computer Engineering and Applications,2009,45(25):82-85.

Abstract: Because of the necessarity of the Web service based on workflow engine for digital campus,the system architecture of Digital Campus Workflow engine(DCWFlow) is proposed.Based on database model,the key implement technologies such as the workflow management,parallel connection approval,branch selection,bidirectional flow and restriction,Web service encapsulation and so on are discussed.An example of how to invoke the engine is given and then the pressure test of some services is done by LoadRunner.The experiment and practices show that DCWFlow has good performance and it can offer uniform standard services for digital campus.

Key words: Web service;digital campus;workflow engine;bidirectional flow;parallel connection approval

摘 要:阐述了Web服务工作流引擎对于数字化校园建设的必要性,提出了数字化校园工作流引擎的系统框架,从数据模型出发,详细论述了工作流管理、并联审批与分支选择、双向流转与约束、Web服务封装等关键技术的实现,给出了引擎的调用示例,基于LoadRunner做了压力测试。实验和实践表明该引擎系统性能良好,可以为数字化校园信息系统提供统一标准的服务。

关键词:Web服务;数字化校园;工作流;双向流转;并联审批
DOI:10.3778/j.issn.1002-8331.2009.25.025 **文章编号**:1002-8331(2009)25-0082-04 **文献标识码**:A **中图分类号**:TP311

1 前言

办公自动化适应信息社会化的需要,基于软科学的理论和计算技术、通信技术的普遍应用而得到迅速的发展。近年来,各高校为了提高办公自动化水平,纷纷推行“数字化校园”,“数字化校园”以校园网为背景集教、学、管理、娱乐为一体,让学校和教育管理部门通过信息化手段,实现对各种资源的有效集成、整合和优化,实现资源的有效配置和充分利用,实现教育和校务管理过程的优化、协调,教师与学生实现教学过程与学习过程的优化,从而实现提高各种工作的效率、效果和效益[1-2]。建设数字化的重要工作之一就是发现用户的工作流程,进行分析建模,并把它体现到信息系统的设计中。学校要发挥服务职能、提高工作效率,需要随时调整工作流程,所以即使是完全掌握了当前需求,随着时间的推移,需求必然也会发生变化。固化的采用硬编码的传统系统设计业务流程在业务流程和组织结构发生改变的情况下,需要将系统进行重大修改,甚至重新设计。而基于工作流引擎可以进行灵活定制提高系统的柔性,适应业务流程的变化。对一个学校而言,整个数字化校园是一个开放统一的大系统,需要较强的交互扩展能力,否则容易出现“信息孤岛”问题,作为核心的通用工作流引擎需要具备良好的扩展性。Web服务是一门新兴的网络技术,由服务提供者、服务请求者、服务代理者等三种角色构成,服务提供者可以将业务封装成统一标准的对外服务。在面向SOA和下一代互联网的信息交换和共享中,Web服务技术占据越来越重要的位置,而且可以通过有效的组合实现特定的功能[3],自然成为实现数字化校园通用工作流引擎的上佳选择。

工作流技术[4-5]是针对工作中具有固定程序的常规活动而提出的一个概念,通过将工作活动分解成定义良好的任务、角色、规则和过程来完成执行和监控,达到提高生产组织水平和工作效率的目的。目前成熟的商业工作流软件产品通常功能庞杂、价格昂贵;一些开源工作流软件的流程定义库一般都基于

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60872051);广西省自然科学基金(the Natural Science Foundation of Guangxi of China under Grant No.0575094)。

作者简介:何倩(1979-),助理研究员,博士生,主要研究方向:分布式计算、网络应用系统研究等;孟祥武(1966-),男,教授,博士生导师,主要研究领域:通信软件,下一代互联网;王勇(1964-),教授,博士,主要研究方向:信息管理系统,信息安全。

收稿日期:2009-03-18 **修回日期**:2009-04-17

XML 文件，难以实现对大容量数据的管理，而且大都不提供 Web Service 接口。

结合项目实际,在数字化校园开发过程中以数据库为存储介质,基于 Web Services 技术,采用 C# 语言设计了一套工作流引擎(DCWFlow)。该引擎提供标准的 Web 服务接口,调用简单,有利于简化数字化校园其他信息系统的开发和维护。

2　系统总体设计

2.1　交互关系

数字化校园一般包括教务系统、学生管理系统、科研管理系统等,各个系统实现自己相关的信息输入和管理界面,对于流转的定义和控制都通过一个统一的 DCWFlow 工作流引擎来完成,如图 1 所示。

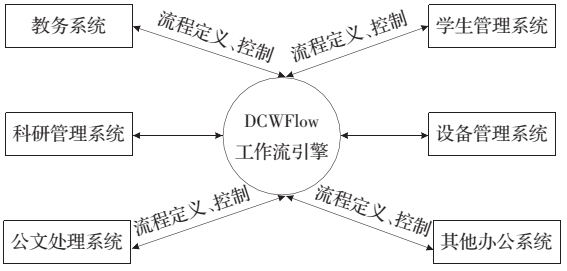

图 1　DCWFlow 与数字化校园其他系统的关系

2.2　系统框架

整个 DCWFlow 工作流引擎的系统框架如图 2 所示，由图形化管理界面、工作流后台、Web 服务封装构成。图形化管理平台提供了一个管理工作流引擎的图形化操作界面。DCWFlow 工作流后台分成 3 个主要部分:引擎支撑子系统、工作流日志采集器、数据分析器。引擎支撑子系统是 DCWFlow 的核心,实现一个工作流管理系统正常工作的所有功能，包括模板设计、工作流管理、流转控制等三个模块;工作流日志采集器负责收集引擎支持子系统中所有的操作日志;数据分析器分析系统日志,主要使用频率统计的方式提炼出基于工作流的角色和工作模式等，然后自动生成模板。工作流后台的接口访问是 DLL 的,Web 服务封装模块把工作流里的 DLL 接口用 Web 服务的方式封装起来对外提供统一标准的接口。主要讨论引擎支持子系统与 Web 服务封装的实现。

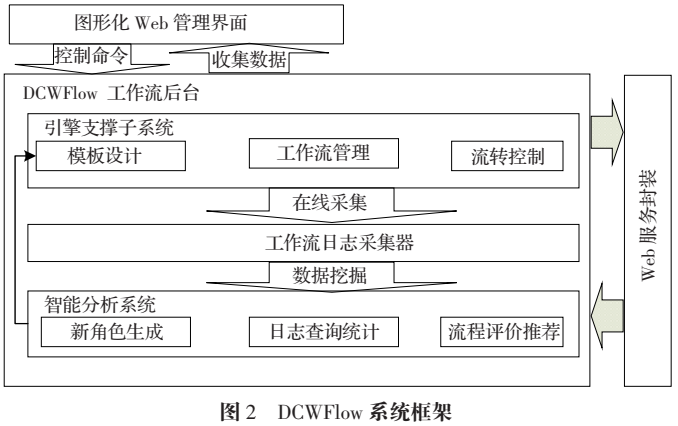

图 2　DCWFlow 系统框架

2.3　数据库设计

DCWFlow 核心数据模型主要包括办公事务表(OBusiness),事务操作表(OBOperation),事务实例表(OBInstance),流程实例表(OBIFlow),操作实例表(OBIOperation)等 5 个实体(如图 3)。其中,OBusiness 表里存放办公事务的业务类别;OBOperation 表用于定义某类办公事务可以进行的操作以及对操作的限制,例如公文处理事务可以定义拟稿、部门审核、会办等操作,会办操作之前必须完成部门审核的约束等;OBInstance 表描述一个具体的办公事务,比如一次具体的公文流转;OBIFlow 表用于控制完成一项具体的事务实例的整个流转过程;OBIOperation 表用于选择具体完成流转的操作人(角色或用户),并存放对于事务处理的结果。DCWFlow 数据模型如图 3 所示。

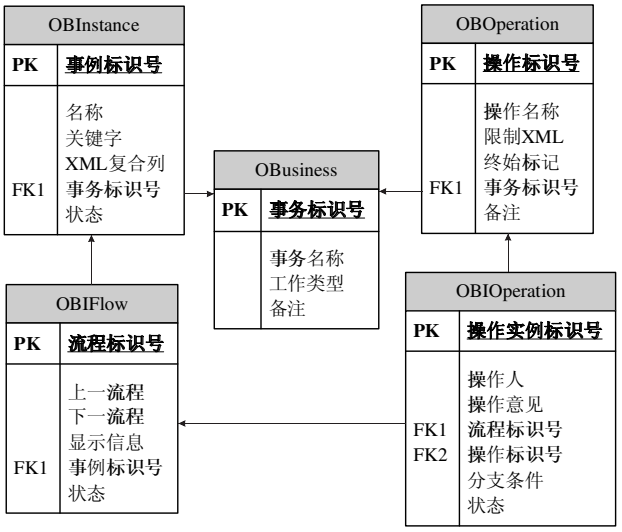

图 3　DCWFlow 核心数据模型

3　系统实现

DCWFlow 采用 C# 语言在 Visual Studio 2005 开发完成,数据库使用 MS SQL SERVER 2005，运行于 ASP.net 2.0 的 Web 之上。

3.1　工作流管理

(1)办公事务初始化

在普通用户实际进行办公业务之前,需要初始化办公事务和对应的操作，这项工作主要是通过对 OBusiness 和 OBOperation 表的维护来实现的。算法如下:

[1.1] 增加一项新的办公事务;

[1.1.1] 在 OBusiness 表里新增一条记录,填写事务名称,设置工作类型等;

[1.1.2] 循环往 OBOperation 表新增该事务的操作步骤,直至完成所有;

[1.2] 修改一项办公事务;

[1.2.1] 拷贝旧的办公事务作为一个副本进行修改,新增事务副本的版本号在旧版本的基础上加 1;

[1.2.2] 实施修改;

[1.3] 删除一项办公事务,不实施物理上的修改,只更改版本的可使用状况。

以上算法对于事务的修改和删除都采取了版本控制,这样可以避免基础数据的更新,影响已有的办公事务实例。

(2)办公事务实例管理

办公事务类型按流程、操作的变化频率特点可分成三大类:常变的、不常变、恒定。对于常变的和不常变的可以由进行办公事务的第一个发起用户设计一个全新的流程或调用流程

模板后修改再设计,对于恒定的办公事务则由系统调用默认模板生成流程和操作步骤实例。办公事务实例管理主要是通过对OBInstance、OBIFlow 和 OBIOperation 等三个表的维护来实现的。算法如下:

[2.1] 增加一项新的事务实例;

[2.1.1]在 OBusiness 表中选择一项事务,在 OBInstance 新增一项工作;

[2.1.2]如果选择从流程模板自动构建流程转入 2.2;如果选择自设计流程转入[2.1.3];

[2.1.3] 在 OBIFlow 中新增一步,选定多个操作增加到 OBIOperation 中,循环进行[2.1.3],构成新增事务实例的所有流程;

[2.2] 修改一项事务实例;

[2.2.1]实施修改,如果涉及到流程修改转[2.1.3];

[2.3]删除一项事务实例。

3.2 并联审批与分支选择

并联审批是指多个单位或个人能够同时对一项事务进行操作。具体实现方法是在 OBInstance 表新增一步流程后,多次循环在 OBIOperation 表进行新增操作,这些新增的操作可以并发执行。

分支选择可根据上一流程不同的结果状态,程序自动地选择一条合适的路径。具体方法是在进行流程设计的时候利用 OBIOperation 表的“分支条件”字段,存放上一流程必须的结果,由这个结果来控制操作的发生。

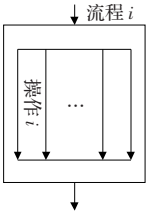

图 4 并联审批

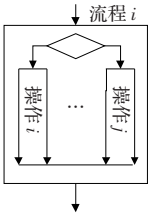

图 5 分支选择

3.3 双向流转

如图 3 所示,办公事务流程实例表通过定义流转的前趋(上一流程)和流转的后继(下一流程)实现了一个双向数据库链表,这样 DCWFlow 定义的工作流可以支持双向流转。

(1)在办公事务活动表中建立触发器。当同一分支的活动所有操作者都完成了操作的时候,流程可以自动进入下一步,当前流程状态设为 2。

(2)当授权的活动操作者要求后退重审时候,事务回到上一流程,将当前流程状态重设为 0,上一流程的状态设为 1。

3.4 流转约束限制

许多工作都提出了前提和后继的限定,这种限定的实现主要借助 OBOperation 表的“限制 XML”字段。“限制 XML”的 Schema 如图 6,其中 previous 表示该项流转的前提流程,next 表示后继流程,当用户创建办公事务实例的时候需要根据这两个信息进行生成,SQL Server2005 已经支持对 XML 字段的直接查询,所以这种查询的实现是方便的。

3.5 Web 服务封装

Microsoft .net 技术对于 Web 服务的实现做了的简化,开发人员可以方便地完成对已有模块的 Web 服务封装[6]。在 DCWFlow 采用了分层的设计思想,由 DataLib 库实现实际的业务逻辑,Web 服务封装的时候只要简单地调用 DataLib 库链接,如办公事务管理的 Web 服务可封装如下:

```
using DCWFlow.DataLib;
…
[WebMethod]
public bool manageOfficeWork(OBusinessE ob)
 {
        return WFWrapper.manageOfficeWork(ob);
}
```

OBusinessE 是一个序列化的办公事务实体类,与 OBusiness 表相对应:

```
[Serializable]
public class OBusinessE
{  public EActionType actionType;//操作类型枚举
    public Guid businessID;//事务标识
    public String businessName;//事务名称
    public int businessType;//工作类型
    public String memo//备注
}
```

```
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:element name="action-condition">
<xs:complexType><xs:sequence>
<xs:element name="previous">
<xs:complexType>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
<xs:element name="instance-id" type="xs:string"/>
</xs:sequence></xs:complexType>
</xs:element><xs:element name="next"><xs:complexType>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
<xs:element name="instance-id" type="xs:string"/>
</xs:sequence></xs:complexType></xs:element>
</xs:sequence></xs:complexType></xs:element>
</xs:schema>
```

图 6 操作限制模式图

```
<action-condition>
<previous><instance-id>xxxxx <instance-id></previous>
<next><instance-id>xxxxx <instance-id>…</next>
</action-condition>
```

图 7 活动限制 XML 的样例图

4 应用示例与系统测试

在数字化校园系统中,公文处理是一个独立的子系统,用户在公文处理系统中录入公文、上传文件,实施界面操作,公文处理的流转后台,例如流转路径模板的选择、流转路径的修改定制以及工作步骤的自动流动等,则是通过调用通用工作流引擎 DCWFlow 的 Web 服务来实现。在公文流转处理中,每次的参与人可能不同,需要每次为这样的工作生成不同的办公事务实例,具体的公文流转处理后台服务调用序列如图 8 所示。

在一台 Intel 双核 E4600(2.4 G)处理器,2 G 内存、Windows 2003 SP1 的 PC 机上安装 SQL Server2005 和 IIS 6.0 服务器并部署 DCWFlow 工作流引擎。压力测试软件使用 LoadRunner 8.1[7],并发用户数为 300。做办公事务读取(GetFlowAll)、工作流程的增加(InsertFlow)和流转(GoToNext)等三个重要服务的性能测试,持续运行 60 分钟,总共完成服务调用各 39 146 次,没

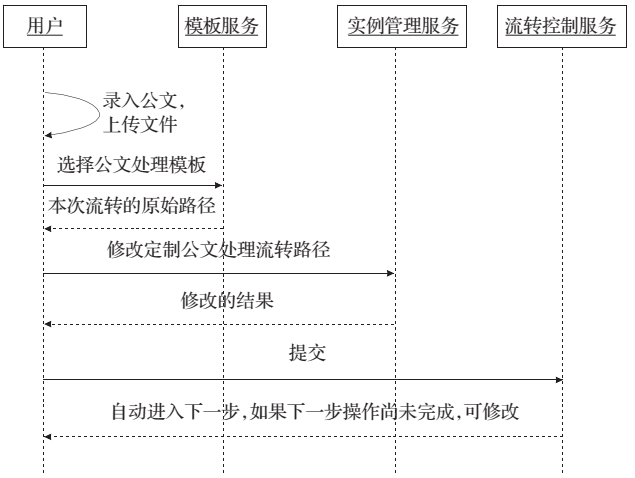

图 8　公文流转处理后台服务调用序列

表 1　DCWFlow 服务完成时间　s

Transaction Name	Min	Avr	Max	Std. D
GetFlowAll	0.010	7.143	19.312	3.398
InsertFlow	0.006	1.112	18.843	0.370
GoToNext	0.011	0.787	18.687	0.358

有发生一次失败,表 1 列出了各项服务的完成时间。

以上数据均是在模拟用户每次都重新刷新页面的条件下取得，因为 Web 服务的初始连接耗时较长，如果手工打开页面，然后多次重复的调用服务，在 300 个并发用户的压力下，GetFlowAll 的响应速度一般能保持在 0.3 秒左右,GetFlowAll 的平均响应时间较 InsertFlow、GoToNext 大许多的原因主要是由于它的测试位置是同一次循环的第一个服务。

5　结束语

通用工作流引擎的应用可以简化开发过程,提高数字化校园信息系统的集成度,减免“信息孤岛”问题。针对数字化校园,基于 Web 服务实现了一个标准通用的 DCWFlow 工作流引擎,经测试,系统性能良好,能够满足中小规模大学校园用户的需求。目前,DCWFlow 已经在实际的系统中得到应用,实践证明该引擎灵活、稳定、兼容性好,具备较好的推广应用价值。下一步将继续完善引擎的人机接口,提高系统的并发性能。

参考文献:

[1] 张学旺,汪林林,马中峰.数字化校园综合应用软件平台的关键技术[J].计算机工程,2007,33(23):267-269.

[2] 谭义东,张亚伟.关于文科类院校数字化校园建设问题的探讨[J].桂林电子科技大学学报,2008,28(6):547-550.

[3] 温嘉佳,陈俊亮,彭泳.基于目标距离评估的启发式 Web Services 组合算法[J].软件学报,2007,18(1):85-93.

[4] 罗海滨,范玉顺.工作流技术综述[J].软件学报,2000,11(7):899-907.

[5] wfmc 主页[EB/OL].[2008].http://www.wfmc.org.

[6] 李敏波.C# 高级编程[M].3 版.北京:清华大学出版社,2005.

[7] Mercury Interactive,LoadRunner[EB/OL].(2007).http://www.mercury.com/us/products/loadrunner/.

[8] Fischer L.Workflow Handbook[M].[S.l.]:Future Strategies,2005.

(上接 77 页)

8　结束语

为了进一步说明 ESDCM 构件模型在实时嵌入式领域 CBSE 开发的可行性及适用优点。表 1 根据构件模型的评价要素给出 ESDCM 构件模型与一些主流构件模型的评价比较。

表 1　ESDCM 构件模型与主流构件模型的比较

构件模型	评价要素					
	领域复用	实时性描述	非功能描述	开放性	可靠性	移植性
PBO		√	√		√	
Koala		√	√	√	√	
PECOS		√	√		√	
DRSCDE	√	√	√			
ESDCM	√	√	√	√	√	√

提出了一种新的具有较好普适性的面向实时嵌入式系统的软构件模型——ESDCM，采用形式化方法给出了一套较为完整可行的模型描述理论体系,首先,提出构件模型及模型元素定义并给出接口规约;其次,提出构件组装机制及组装规约;最后，从实时性描述及验证角度给出构件时间性的推理方法。目前模型已成功应用于一些采集及监控系统中,并成功复用到其他系统的开发中。但在平台验证和组装动态演化方面仍有不足,今后将进一步研究 ESDCM 构件模型平台验证机制及组装动态演化方案。

参考文献:

[1] Stewart D B,Volpe R A,Khosla P K.Design of dynamically reconfigurable real-time software using port-based objects[J].IEEE Transactions on Software Engineering,1997:759-776.

[2] van Ommering R,van der Linden F,Kramer J,et al.The koala component model for consumer electronics software[J].IEEE Computer,2000,33(3):78-85.

[3] Winter M,Genbler T,Christoph A,et al.Components for embedded software:The PECOS approach[C]//Proc of the 2nd International Workshop on Composition Languages,2002.

[4] 刘晓燕,张云生,Schwarz J J,等.复杂实时系统软构件对象设计[J].计算机工程与应用,2003,39(31):119-121.

[5] Hoare C A R.Communicating sequential processes[C]//Comm ACM21,1978.

[6] 任洪敏,钱乐秋.构件组装及其形式化推导研究[J].软件学报,2003,14(6):1069-1070.

[7] Davies J,Schneider S.A brief history of timed CSP[J].Theoretical Computer Science,1995,138(1):243-271.

[8] Allen R J.A formal approach to software architecture[D].Pittsburgh:School of Computer Science Carnegie Mellon University,1997.

[9] 刘晓燕,张云生,Schwarz J J,等.基于 C/S 关系的实时系统构件交互规约[J].计算机工程与应用,2007,43(17):104-107.