维普资讯 http://www.cqvip.com

基于分布式工作流技术的校园应用软件集成模型研究*)

王 刚[1] 陈世福[2] 陈兆乾[2] 肖 伟[1]

(四川警官高等专科学校 泸州 646000)[1] (南京大学计算机科学与技术系 南京 210093)[2]

摘 要 工作流技术是实现企业业务过程建模、管理、优化与重组,最终实现业务过程管理与自动化的核心技术。而分布式工作流技术的研究已经成为当前研究的热点。本文描述了分布式工作流技术的特点,利用 Java 和 CORBA 技术给出了基于分布式工作流技术的校园应用集成模型。探讨了多个工作流系统互操作的问题。最后提出了动态创建过程实例的思想和活动池的概念,可以极大地提高整个系统的效率以及资源利用率。

关键词 分布式工作流,活动池,应用代理,应用集成,Java,CORBA

The Study of Campus Application Software Integration Model Based on Distributed Workflow Technology

WANG Gang[1] CHEN Shi-Fu[2] CHEN Zhao-Qian[2] XIAO Wei[1]

(Sichuan Police College, Luzhou 646000)[1] (Department of Computer Science & Technology, Nanjing University, Nanjing 210093)[2]

Abstract Workflow technique is the keystone that can realize modeling, management, optimizing and re-organization for enterprise operation process. And ultimately it can realize the operation process management and automation. The study of distributed workflow technology has been a hot area now. This paper introduces the feature of distributed workflow technology, and campus application integration model using Java and CORBA on the basis of distributed workflow technology is referred. The interoperability of the different workflow system is addressed. In the end, we present dynamically-create process instance and the activity pool, especially using them can increase the efficiency of the whole system a lot.

Keywords Distributed workflow, Activity pool, Application agent, Application integration, Java, CORBA

1 引言

进入 20 世纪 90 年代,随着计算机与网络技术的迅速发展,特别是在 Internet 应用日益普及的情况下,现代企业的信息系统的分布性、异构性和自治性的特征越来越显著,相应的企业信息资源也分布在异构的计算机环境中,信息源之间的连接表现出松散耦合的特点,在这种客观背景下,工作流技术也随之进入了一个新的发展阶段——分布式处理阶段。

目前,在许多高校中,各个部门都有基于各种平台的管理信息系统,相互之间信息共享程度很低,但是不能满足高校多个部门之间业务过程管理与自动化的需求。信息集成解决了由于"信息孤岛"所造成的单位决策困难、信息资源重复和不一致的现象,提高了单位的整体效益。然而,随着高等教育事业的不断发展,呼唤着 21 世纪新技术下的校园办公自动化系统的出现,以提高校、系二级管理的整体水平和办事效率,满足校园日常办公的业务流程自动化或半自动化。单纯的信息集成已不能满足现实的需求,必须进行过程集成,即利用计算机集成支持工具高效实时的实现校园资源共享和应用间的协调工作,将一个孤立的应用集成起来形成一个协调的支持过程自动化的校园 OA 运行系统。过程集成相对于信息集成而言更具柔性,它将应用逻辑和过程逻辑分离,过程建模与具体数据分离,在不修改具体功能的情况下,通过修改模型就可以实现系统功能的改变,从而大大提高了校园 OA 系统的灵活性和高校各部门之间业务过程管理与自动化的需求。

本文根据学校各个部门已有的信息系统,利用 Java 和 CORBA 技术,通过 IIOP,RMI,SOAP 等协议,将多种形式的校园应用进行信息集成和过程集成,以实现校园有些业务流程的自动化或半自动化。

2 系统结构设计

2.1 设计思想

我们进行系统结构设计时,要遵循 WFMC 提出的工作流管理系统参考模型[5],系统主要包含下面三个部分的内容:

(1)软件构件:组成整个系统的软件模块,包括过程建模工具,工作流引擎,任务表管理器,用户界面等。

(2)系统控制数据:系统执行过程中各个构件所用到的数据,有过程定义,企业组织/角色模型数据,任务表,工作流控制数据和工作流相关数据等。

(3)应用数据:系统执行过程中还必须与系统外部的多种应用进行交互,共同完成任务,这些应用及其相关数据是对整个系统功能的重要扩展。

2.2 系统结构

我们开发的基于分布式工作流技术校园应用软件集成系统,将系统的客户端模块完全放置在浏览器端,方便地实现了对工作流过程的分布管理和控制。同时,将 Java 与 Web 充分结合起来,使用 Java Applet、JSP 和 EJB 等 Java 技术来实

*)本文得到江苏省高技术项目(BG2003026)基金资助。王 刚 讲师,硕士,主要研究方向:分布式对象技术、网络数据库技术、网络犯罪侦查技术;陈世福 教授,博导,主要研究方向:人工智能与知识工程、数据挖掘、工作流技术;陈兆乾 教授,博导,主要研究方向:机器学习、分布式人工智能、工作流技术;肖 伟 主要研究方向:社区警务和保安业务。

维普资讯 http://www.cqvip.com

现客户端的功能，最大程度地提高客户端工具的管理功能。为了实现不同工作流引擎间的通信与协作，我们将每个工作流引擎都封装在 CORBA 对象之中，通过与实现语言无关的接口定义语言(IDL)来定义与外界进行交互的接口。

客户对 EJB 组件的调用请求将首先被传递给 EJB 容器/服务器，然后才由 EJB 容器/服务器间接调用实际的 EJB，使得 EJB 容器/服务器能够有机会提供各种中间服务，如：事务管理、状态管理、安全性管理、持久性管理等。为了整合现存的非 Java 代码的应用系统，可以将这些代码封装成 CORBA 服务对象，然后用 EJB 去调用；或者也可以通过 JNI 去调用。通过 JDBC 或 SQL/J 来访问后台数据库。

系统的开发在遵照 WfMC 关于工作流参考模型及其它有关标准的基础上，利用 Java 和 CORBA 技术，保证了系统的规模化、容错和平台无关性。核心服务采用 CORBA 来实现，并支持 IIOP，RMI，SOAP 等协议，能够与多种形式的应用进行集成。整个系统的体系结构设计如图 1 所示。

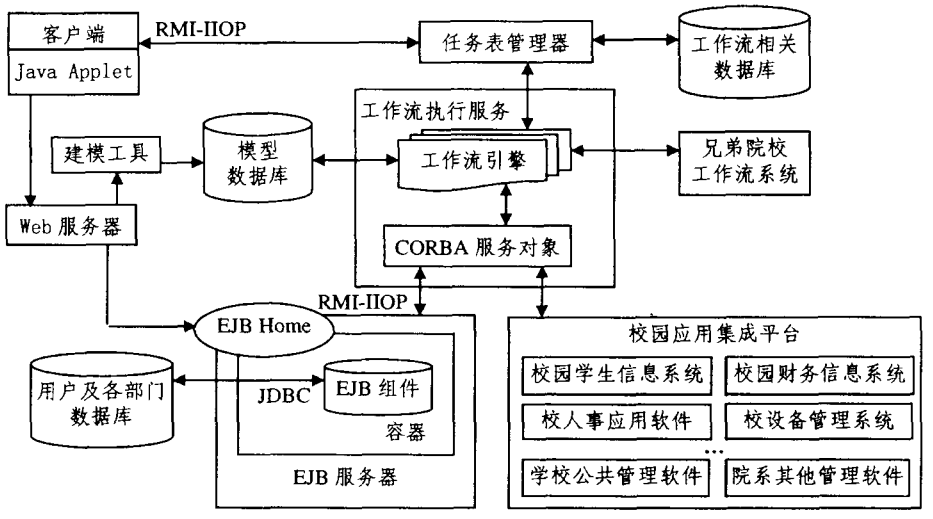

图 1　基于分布式工作流技术的校园应用集成模型

主要部分的工作简述如下：

(1)工作流执行服务　为系统提供运行时服务，包括一个或多个工作流引擎并以此来产生、管理以及运行工作流实例。其中的工作流引擎被定义为工作流实例提供运行时的执行环境的软件服务或“引擎”，它提供了工作流运行服务的控制环境，主要负责过程定义的解释、过程实例的控制、过程活动的导向以及工作流相关数据和控制数据的传送和维护等。

(2)工作流引擎　是工作流执行服务的核心。它包括 3 个部分：任务容器、模型解释器、工作流引擎接口。其中，任务容器负责调度流程的运行，即创建和管理过程实例的运行、调度活动实例的运行并创建要处理的工作项、维护用户的任务表(记录了当前需要该用户处理的所有任务)，另外它还负责维护工作流相关数据；模型解释器负责解释过程模型；引擎接口为客户端的工作流应用、工作表管理器以及工作流管理工具和任务容器相交互的接口。

(3)任务管理器　负责从工作流任务表中取得任务项，把它们提供给用户进行处理。

(4)建模工具　在该工具中，可以通过图形化控件进行选择和拖动，并由节点(表示活动)和箭线(表示转移)构成工作流过程模型。

(5)容器 EJB 实现(Skeleton 与代理)　容器可以管理分布式通信 Skeleton，该 Skeleton 负责对与客户之间所传输的数据进行编码与解码；容器池存储 EJB 实现实例，并且使用代理在调用被发送给 EJB 实现实例之前执行与特定 EJB 有关的服务管理操作。

(6)CORBA 服务对象　为整合现有的非 Java 代码，可以将这些代码封装成 CORBA 服务对象，然后用 EJB 使用 RMI-IIOP 协议去调用，或者也可以通过 JNI 去调用。

3　实现技术

3.1　工作流引擎

3.1.1　工作流过程的分布处理流程　工作流引擎是工作流管理系统的核心部分，它管理着整个工作流过程，负责实例化工作流过程，管理和控制过程实例运行，负责活动之间的导向，任务的分配等工作。在分布式环境下，整个工作流管理系统有多个工作流引擎协同工作，每个工作流引擎都是一个独立自治的节点，负责处理工作流过程中与该节点有关的活动。

过程定义节点对工作流过程建模完毕后，根据过程中包含的分布信息，将过程的各个部分分配到执行它们的各个节点上去，由节点上的工作流引擎来管理这部分过程的运行与调度，并负责与其他节点间的通信。工作流过程的分布处理流程如图 2 所示。

当一个工作流引擎收到一个过程定义后，即为这个过程定义创建一张过程表，用来记录这个过程定义的具体内容以及所有的相关资源。同时，也为这个过程定义创建一个过程线程，负责维护该过程定义所有实例的运行。过程被实例化运行后，每个过程实例拥有一张实例表，记录了过程实例中各个活动的执行情况以及所有用到的相关数据。

工作流引擎接收到别的节点发送的消息后，过程线程检查消息内容，确定要执行活动，然后从过程表中取出与该活动相关的信息，并在实例表中为该活动创建一条记录，描述活动状态，然后检查活动的开始条件。如果满足开始条件，则创建一个活动线程，管理活动的执行，调用相应的应用查询来完成任务。活动结束后检查活动的返回结果，判断是否满足活动结束条件，如满足则结束该活动，否则重新执行该活动。最后，活动判断其后继活动所在节点，发出消息并传递参数，同

维普资讯 http://www.cqvip.com

时删除实例表中该活动记录。如果活动还有其他条件未满足，则在活动记录中记录已判断的条件，等待新的消息到达。如果活动的开始条件都不可能被满足，则向该活动的后继活动发送消息，通知它们该活动永远不会被执行，删除实例表中的该活动记录。

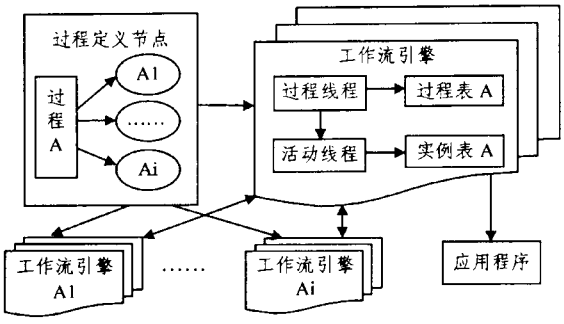

图 2　工作流过程的分布处理流程

3.1.2　*工作流引擎间的互操作*　每个工作流引擎都是一个功能独立的个体，可以分别完成自己的任务，但是一个企业经营过程涉及到多个部门或多个公司间的紧密使用，为了完成整个企业过程，各个工作流引擎间必须相互协作，不断地进行消息通信，这也是实现整个分布式工作流管理系统的关键所在。为此，我们采用了 CORBA 作为工作流引擎间通信的底层技术支持，使用 CORBA 规范中的对象和 IDL 接口定义来实现系统中工作流引擎间的互操作和对数据源的封装，系统中的所有组成部分都以 CORBA 对象的方式实现。

对象请求代理(ORB)是 CORBA 对象互操作的中介[2]，它作为对象间服务请求/响应的中间代理，接收对象的请求并把请求转给相应对象，服务完成后又将执行结果或异常返回给请求者。ORB 隐藏了对象的位置、实现和状态，可以使对象以语言、位置和平台独立的方式发出请求和提供服务，相互协同工作，从而建立真正的分布处理，是实现分布对象互操作的核心。工作流引擎间的通信是通过客户方对服务方提出请求，服务方返回结果来进行的，同时，客户方的工作流引擎也可以接收其它引擎的请求而成为服务方。

系统实现对象间的互操作，我们采用了 CORBA IDL，它是一种与编程语言无关的接口定义语言，用来定义对象间的请求/服务接口，描述应用对象所封装的内容及界限。IDL 语言可以被映射为多种编程语言，包括 Java。

工作流引擎和系统的其他部分在运行过程中，通常会和企业现有的应用软件进行交互。企业的现有应用软件大多是用不同的编程语言实现，并运行在不同的操作系统上。为了实现不同应用软件功能和信息的集成，我们利用了 CORBA 对象的位置、实现、执行状态和通信机制的透明性，按照 CORBA IDL 语言对应用软件提供的服务进行描述，使得工作流管理系统同企业应用软件可以透明地交互运行。

另外，WFMC 规范接口 4 定义了工作流引擎间的互操作接口。我们在这些接口规范的基础上进行了工作流引擎的设计，这使得工作流引擎可以同兄弟院校的工作流应用系统进行互操作。

3.2　工作流管理工具

3.2.1　*任务管理器*　整个工作流过程是由许多活动组成，活动可以是一个由工作流引擎自动完成的任务，也可是有用户手动执行的任务。在工作流管理系统内有一张任务表，它记录了系统工程中所有任务的相关信息，如任务的状态，属性，执行者，任务需要调用的应用程序及其地址和要传递的参数等。任务表既可能记录了一个过程实例的多个任务，也可能包含了多个过程实例的多个任务。任务管理器就是用来维护任务表的工具，负责监视系统中任务的执行情况，完成用户对工作流过程的参与。任务管理器也被封装成了 CORBA 对象，提供了对外服务的接口，并且可以同工作流引擎进行交互，改变活动的状态，动态地对工作流模型进行修改，增加了系统的柔性。

3.2.2　*客户端*　主要完成对工作流实例执行过程中各种活动的处理。在过程实例的执行过程中，经常需要用户通过已有的应用程序，完成过程定义所要求完成的处理或操作。

客户端就是管理员和一般用户同工作流管理系统的交互界面，它通过与任务表管理器和工作流引擎的交互来实现提供给用户的各种功能。基于 Web 的设计，使得用户无需安装任何软件，通过网页浏览器就可以实现所有客户端的功能，通过网络用户可以随时随地地参与学校的业务过程。

在系统设计中，根据职责的不同将用户分为管理员和一般用户两种。系统分别为他们提供了不同的功能。管理员需要随时掌握系统的运行和过程的执行情况，他可以维护和修改系统配置，收集系统中所有工作流引擎的执行信息，包括名称、地址、工作状态、负荷情况等。将工作流模型实例化，设定初始的过程参数，开始一个工作流过程的执行。在过程执行时，监控过程实例，可以更改过程或活动的属性，必要时还可能动态地更改执行中的过程模型，以适应企业经营过程的变化。一般用户主要要求能够获取和执行分配给他们的任务，所以他们应该可以通过身份验证从任务表中获取与自己相关的任务信息，可以对任务进行多种操作，包括“开始执行”，“挂起”，“继续”，“完成”等。

系统使用 CORBA 作为分布的工作流引擎之间的底层通讯支持，校园中原有的系统被封装成 CORBA 服务对象，对外提供标准的 IDL 接口。根据系统应用集成需要，开发出的应用程序作为 CORBA 客户端，根据统一的 IDL 接口以及工作流引擎来调用 CORBA 服务对象。Web 浏览器可以通过 Java Applet 方式下载 CORBA 客户方程序来访问 CORBA 应用服务，客户端的管理软件可以通过任务表管理器或 Web 服务器与工作流引擎进行通信。

3.3　活动池

为了更好地实现动态创建，提高效率，我们进一步提出活动池的概念。由于在动态创建过程中，当过程实例需要运行一个尚未创建的活动实例时，它这时就要创建该活动实例，仍然需要消耗创建一个活动实例的代价，而一个活动实例在运行完就销毁了，我们可以在系统再开辟一个存储区，它用来保存那些已经用完准备销毁的活动实例。系统没有必要把过程实例立刻删除，让垃圾收集器把它回收，而是把活动实例放到存储区中以备后用。这样，在过程实例需要用到一个活动实例时，它就可以先从该存储区中寻找有无匹配的活动实例，这样就省掉了创建那个过程实例的代价。当然该过程实例一开始还是要被创建一次的，但后来可以被反复使用。

3.4　应用代理

在工作流系统中，活动分为人工型的和系统自动调用的，一般由于学校各部门的分布性，工作流系统不可能对所有的应用直接启动，但是也有很多应用可以直接集成到工作流管理系统中，完全由系统来控制应用的执行，这就是工作流系统

（下转第 108 页）

维普资讯 http://www.cqvip.com

而 VoiceXML 频繁地加载语法文件会对系统性能造成很大的影响，因此，对 Web 端维护的语法文件，我们可以通过一个定时任务在一段时间后对所修改的语法进行统一加载，这样就不会影响系统性能。

语法文件在 VoiceXML 中被加载后进入语音信息收集阶段，我们通过事先定义的 Callee 和 PhoneType 字段来收集用户的语音输入，当用户的输入匹配语法文件时，返回对应的字符串给应用程序，应用程序通过数据库操作查询出用户好友的对应对话号码，如马波的手机，然后上报给智能网设备进行后续接续。

下面代码演示了我们所构建的 VAD 系统最基本的呼出流程：

```
……
〈grammar src="〈%=sFinalGrammarPath%〉"/〉
〈field name="Callee"〉
〈prompt bargein="true"〉〈audio src="〈%=sFinalAudioPath%〉
1.wav"〉请问您要找哪位？比如：找张三的手机。查询好友电话号码
请按 1，维护好友电话号本请按 2……〈/audio〉〈/prompt〉
〈/field〉
〈field name="PhoneType"/〉
〈filled namelist="Callee"〉
〈if cond="PhoneType==undefined"〉
〈assign name="PhoneType" expr=" MobilePhone "/〉
〈if cond="Callee==1"〉
   〈submit next="queryfriendphone.jsp"/〉
……
〈else/〉〈! - 如果是语音输入 -〉
      〈clear namelist="getfriendphone"/〉
〈/if〉
……
〈subdialog name=" getfriendphone" src=" getfriendphone.jsp"
   namelist="CallerNo Callee PhoneType" expr=true 〉
   〈filled〉
      〈if cond="getfriendphone.Ret==0"〉
      〈submit next=" returnevent.jsp" namelist=" CallerNo
      CalleeNo"/〉
      〈else/〉
      〈prompt〉〈audio src="〈%=sFinalAudioPath%〉2.wav"〉对
      不起，您未设定该电话号码。〈/audio〉〈/prompt〉
      〈clear namelist="Callee PhoneType"/〉〈reprompt/〉
      〈/if〉
   〈/filled〉
〈/subdialog〉
```

由于每个用户都有其相应的语法文件，因此语法文件的信息我们存放于数据库中，当用户拨打电话进入 VAD 系统，通过用户鉴权后从数据库中取出该用户对应的语法文件路径，然后通过 grammar 元素的 src 属性指定该用户对应的语法文件路径，并对此语法文件进行加载，此语法文件中包含该用户好友及其各种联系方式，语法的作用域覆盖 Callee 和 PhoneType 两个字段。业务播放相应的提示音后，开始收集用户输入，当用户的输入不匹配语法时，提示用户继续输入，如用户的输入匹配所加载的语法文件，则从语法文件返回定义的对应值给变量 Callee 和 PhoneType，在程序中，我们对 PhoneType 字段进行了灵活处理，用户可以只说好友姓名而不说联系方式，这时，可以把好友的某种联系方式当成默认的查找类型，如上述代码中，当用户不说联系电话类型时，我们默认处理为手机，然后流程根据 Caller（用户电话号码，从电话交换设备传进业务中）、Callee 和 PhoneType 字段的值进行数据库操作，查找该用户所查询的好友的相应的联系方式，如果查找失败，则进行后续错误处理，如提示用户继续查找或播放提示音后退出系统等。如查找成功，则上报给智能网，智能网根据业务中上报的好友电话号码和主叫号码进行接续和计费操作。

总结 随着基于 VoiceXML 的语音增值业务的规模逐步扩大，应用领域越来越广泛，用户对语音业务的需求也越来越多元化、个性化，传统的电话按键输入已越来越不满足语音业务发展的需要，通过 ASR 实现语音输入将成为语音增值业务发展的新方向。

我们基于 Nuance 识别系统所构建的 VAD 系统，具有实用、方便、易操作等特点，在国内正处于推广和试用阶段，部分地区已开始商用，特别在面向企业和集团的应用中，其效益和价值十分明显，这也是我们推广业务的一个重要方向。我们相信：在不久的将来，VAD 业务必将因其便利性、实用性和效益得到更多用户和运营商的青睐，从而得到越来越广泛的应用，成为语音增值业务新的亮点。

参 考 文 献

1 Voice Extensible Markup Language (VoiceXML) Version 2.0. http://www.w3.org/
2 Nuance Grammar Developer's Guide. http://www.nuance.com/
3 VoiceXML Programmer's Guide . http://cafe.bevocal.com/docs/vxml/index.html
4 Beasley R. Voice XML 语音应用程序开发[M]. 北京：机械工业出版社，2002

（上接第 96 页）

的直接启动应用接口。使用应用代理会为工作流管理系统的开发和集成都带来很大方便。应用代理与工作流引擎之间的数据交互和消息传递都可以使用标准化的 API 和数据格式完成。而应用代理和直接启动的应用之间的数据交互和消息传递利用专门的面向特定应用的集成接口来完成。这样可以大大提高工作流系统的柔性。

结束语 随着高校规模的不断扩大和校园各部门对应用软件要求的不断提高，校园日常办公的业务流程日趋复杂，基于分布式工作流技术的应用软件集成系统是对高校现有信息系统资源融合的有效方式，必将对高校各处室、各院系等部门之间业务流程的自动化或半自动化、提高工作效率发挥积极的作用。本文利用 Java 和 CORBA 技术提出了基于分布式工作流技术的校园应用集成模型，在此模型的基础上并遵循工作流系统相关标准，我们正在开发的校园应用软件集成系统，具有良好的开放性和一定的灵活性，可应用于目前高校的办公自动化系统中。其主要优点如下：

(1)可扩展性好：应用扩展时，只需添加相应的分布式处理结点即可；

(2)健壮性较强：当分布式环境中的某台处理结点出现错误后，只有那些流经该结点的过程的运行受到影响到，而不至于使整个系统陷于瘫痪；

(3)能够自然地应用在大规模、分散性强的企事业环境中。

参 考 文 献

1 Hollingsworth D. Workflow Management Coalition: The Workflow Reference Model Document Number TC00-1003 Document Status-Issue1.1, Jan. 1995
2 Cichocki A, Helal A S, Rusinkiewicz M, Woelk D. Workflow and Process Automation-Concepts and Technology. Kluwer Academic Publishers, 1998
3 CORBA. The OMG Management Architecture. The Object Management Group. http://www.omg.org
4 The IBM Corporation, WebSphere MQ Workflow. http://www-306.ibm.com/software/integration/wmqwf/
5 MfMC 接口规范，URL：http://www.mfmc.org
6 (美)Justin Couch，等著，考迟，马琳，等译．J2EE 宝典．北京：电子工业出版社，2002