

一种基于 SOA 和 JBPM 的工作流引擎模型*

张辉栋, 卢选民, 杨杰, 李成福

(西北工业大学 电子信息学院, 陕西 西安 710129)

摘要: workflow 技术是实现业务流程自动化的关键技术, 但现有的 workflow 技术已不能满足目前分布式异构和松散耦合的企业信息资源需求。利用 SOA 技术, 结合 JBPM, 提出了一种基于 SOA 和 JBPM 的工作流引擎模型, 并对其流程控制与任务管理进行了分析, 最后, 基于 SOAP、WSDL 协议和 JPD L 实现了此工作流引擎, 证明了其有效性。

关键词: 工作流引擎; SOA; JBPM; 流程实例; 模型

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2013)15-0012-03

A workflow engine model based on SOA and JBPM

Zhang Huidong, Lu Xuanmin, Yang Jie, Li Chengfu

(School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract: Workflow technology is the key to achieve business process automation. But the traditional workflow technology can't meet enterprise information resources' demand for distributed heterogeneous and loosely coupled. A workflow engine model based on SOA and JBPM is proposed in this paper, which utilized SOA technology and combined with JBPM. Then we made an analysis of process control and task management. Finally, this workflow engine model based on SOAP, WSDL and JPD L has been realized in this paper.

Key words: workflow engine; SOA; JBPM; process instance; model

工作流是全部或部分由计算机支持或自动处理的业务过程, 目的是为了提高效率、降低生产成本、提高企业整体的生产管理水平和竞争力。但是现代企业信息资源已呈现出一种分布式异构和松耦合的特点, 传统的工作流已不能满足其功能要求。

面向服务的体系结构 SOA^[1] (Service-Oriented Architecture) 技术将应用程序的不同服务通过这些服务之间定义良好的接口和契约联系起来, 这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。因此, 在传统工作流当中引入 SOA 技术, 发挥两者的优势, 使工作流具有互操作性、强大的开放性及语言和平台无关性, 可以满足跨平台、跨组织、跨部门的分布式的企业应用需求。

本文利用轻量级的开源工作流引擎 JBPM (Java Business Process Management), 结合 SOA, 提出了一种基

于 SOA 和 JBPM 的工作流引擎模型, 并对其流程控制与任务管理进行了分析, 最后, 基于 SOAP、WSDL 协议和 JPD L 实现了此工作流引擎, 证明了其有效性。

1 基于 SOA 和 JBPM 的工作流引擎模型

基于 SOA 和 JBPM 的工作流引擎框架模型如图 1 所示, 该模型主要由 3 部分组成: 服务提供者、服务请求者和服务代理。

该模型的工作流程是: 工作流引擎将流程定义工具产生 XML 格式的流程定义文件解析, 产生流程实例。然后通过 Web 服务接口将这个流程实例反映的流程信息封装成服务形式, 发布于服务代理中, 供服务请求者查找, 服务请求者根据描述信息找到服务提供者并绑定服务提供者, 随后两者间可以消息传递, 工作流引擎根据与服务请求者的消息交互完成流程实例。

(1) 服务请求者: 当服务请求者需要某项服务时, 先在服务代理中查找相关的服务描述信息, 然后它与服务提供者提供的服务进行绑定, 并且与服务提供者信息交

* 基金项目: 国家“863”高技术研究发展计划项目基金 (2009AA01Z430); 国家科技支撑计划 (2007BAH05B02-04)

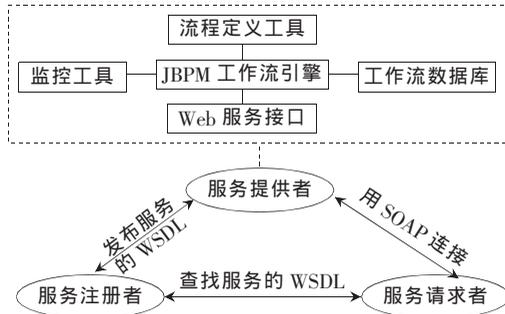


图1 基于SOA和JBPM的工作流引擎模型

换实现业务任务。

(2) 服务代理: 存储服务提供者发布服务的描述, 服务消费者向服务代理查询它所需要的服务, 并根据描述与服务提供者绑定。当所请求的服务是服务代理时, 服务代理也经常充当服务提供者的角色。

(3) 服务提供者: 提供工作流引擎的功能, 把工作流引擎提供的服务相关描述信息 WSDL 发布于服务代理中, 供服务请求者查询绑定。由于本模型中主要是对工作流引擎功能封装成服务, 所以本模型的服务提供者即是工作流引擎模块。

此工作流引擎模块主要包括 JBPM 工作流引擎和 Web 服务接口。Web 服务接口主要是将工作流引擎发布为服务和服务请求者交互, 最终使得整个流程可被封装为 Web 服务并完成服务请求者的需求。

上述模型中的 3 种角色之间的交互是通过以下 3 种基本操作来实现的。

(1) 发布服务: 服务提供者向服务代理发布他们的服务。其发布的信息包括服务的接口定义、服务提供者所处的位置以及其他可能用到的支持信息和文档。

(2) 查找服务: 服务请求者向服务代理查找他们需要的服务。

(3) 绑定服务: 服务消费者与服务提供者提供的具体服务进行绑定, 这个绑定的过程主要是用户的身份验证和服务确认。

在上述的服务操作中, 服务的发布或查找是通过 WSDL 协议实现, 该协议主要描述 Web 服务提供的功能、访问该服务的数据格式与协议要求以及该服务如何定位等内容。服务提供者与服务请求者间的绑定是通过 SOAP 协议实现的, SOAP 是一种轻量的、简单的、基于 XML 的协议, 它被设计成在 Web 上交换结构化的和固化的信息。而工作流引擎的流程定义是遵循 JPD L 协议实现的, 在 JPD L 中提供了任务(tasks)、待处理状态(wait states)、计时器(timers)、自动处理(automated actions)、事件(event)等术语, 并通过图形化的流程定义, 很直观地描述业务流程。

2 工作流引擎的流程控制与任务管理

2.1 流程调度控制

工作流的最显著的优点是实现了流程的自动控制。

在工作流引擎中的表现是工作流引擎把流程定义文件解析、实例, 生成流程实例也就是一套对象, 而这套对象的结构则反映了流程的结构。当流程实例创建完成并进入运行状态时, 流程实例的运行全部依靠工作流引擎的控制。

在 JBPM 中, 一个运行标记 token 代表一个流程实例中的一次执行, 它是运行时的概念, 用来维护流程设计图中节点的指针。token 需要一个信号(signal)来让流程向前执行。当接收到一个 signal, token 会通过路径(弧 transition)离开当前所在的结点到达下一个节点。若 token 已经在 Start-State 节点, 那么就可以开始往前推进, 从而促使流程实例往前运行。对于外部操作, 触发流程实例往下运行的操作有两个:

- (1) 强制执行 ProcessInstance 的 signal 操作。
- (2) 执行 TaskInstance 的 end 操作。

如图 2 所示, 这两个操作主要是调用内部的 token.signal 方法来实现流程实例的推进。

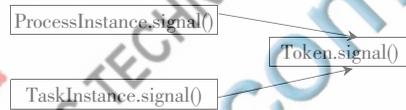


图2 两操作的内部调用

token.signal() 方法的一些内部操作如图 3 所示。即: 确定转向节点、确定转向节点的上下文对象、离开当前节点、获取转向节点的上下文对象、执行节点。当执行完本节点流程会循环进入 token.signal() 方法, 直至完成流程实例。

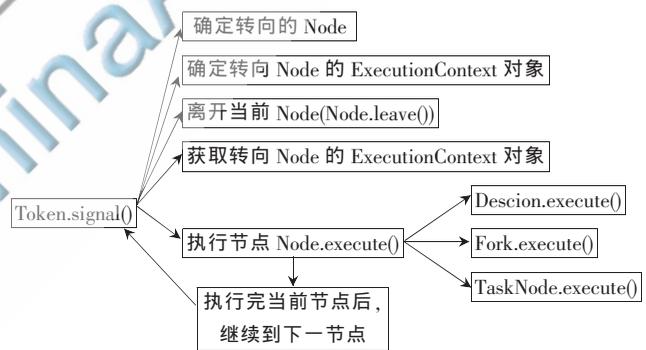


图3 signal方法的内部操作

2.2 任务管理

工作流引擎的一个核心功能是为人类管理任务和任务列表, 其主要是在 Task 节点当中定义的。当工作流引擎执行到 Task 节点时, 根据任务节点上定义的任务产生任务实例。一个标准的任务实例共分为初始态、运行态、挂起态、完成态和终止态 5 个阶段。在任一时刻, 任务只能处于一个任务状态。任务状态之间的关系如图 4 所示。

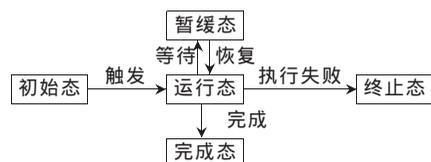


图4 任务状态生命周期

初始态指任务实例被创建成功的状态,当执行任务的条件满足后就进入到运行态,任务实例在运行态时,如果由于某些依赖性或者约束性的限制,如必须等待某些任务实例的执行,必须等到它所需要的资源都满足时才能继续运行,那么该任务实例就进入到挂起状态,直到依赖或者约束性的限制得到满足时,任务实例重新进入运行态,任务实例的运行由于某些原因可能执行失败,任务实例进入到终止态,当任务成功执行后进入到完成态,任务的执行结束。

任务实例的一个作用就是支持用户交互,把数据显示给用户并从用户那里收集数据。任务实例需要从服务请求者输入数据,然后 workflow 引擎对数据进行处理操作,最后把信息再通过服务消息传递到服务请求者显示给用户。JBPM 工作流引擎支持对任务实例分配处理器进行扩展开发,实现对特定数据的特定处理操作,同时其支持对任务实例分配给用户、候选者或泳道。

3 基于 SOAP、WSDL 协议和 JPD L 工作流引擎的实现

本文以数字档案归档流程为例,对图 1 所示的模型进行了实现。

(1) SOAP

在服务提供者与服务请求者之间的消息交换是通过 SOAP 协议进行的。SOAP 是一种轻量的、简单的、基于 XML 的协议,它被设计成在 Web 上交换结构化的和固化的信息。SOAP 协议可以与现存的许多因特网协议和格式结合使用,包括超文本传输协议(HTTP),简单邮件传输协议(SMTP),多用途网际邮件扩充协议(MIME),它还支持从消息系统到远程过程调用(RPC)等大量的应用程序。同样,SOAP 中消息头和消息体的机构确保基础架构消息和服务应用消息的完整性。

下面是 SOAP 协议部分代码:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body xmlns:m="http://localhost:8080/
    FilearchivingProcess">
    <m:GetArchiveTable>
    </m:GetArchiveTable>
    ...
  </soap:Body>
</soap:Envelope>
```

(2) WSDL

WSDL 是 Web 服务描述语言,主要描述 Web 服务提供的功能、访问该服务的数据格式与协议要求以及该服务如何定位等内容。服务提供者把自己的服务转换成 XML 格式的描述语言发布于服务代理中。当服务请求者在服务代理中查找到需要的服务的 WSDL 时,服务请

求者解析其内容,同时确定服务提供者的位置、协议、需要的数据格式、方法信息,最后生成符合的 SOAP 文件并发送给服务提供者,此时两者实现连接。

下面是归档流程的 WSDL 部分代码:<wsdl:message name="archiveTable">

```
<wsdl:part element="tns:archiveTable" name="
  parameters">
</wsdl:part>
</wsdl:message>
<wsdl:portType name="FilearchivingProcess">
<wsdl:operation name="submit">
<wsdl:input message="tns:archiveTable"
  name="archiveTable">
</wsdl:input>
<wsdl:output message="tns:archiveResponse" name="
  archiveResponse">
</wsdl:output>
</wsdl:operation>
</wsdl:portType>
<wsdl:service name="FilearchivingProcessImplService">
<wsdl:port binding="tns:
  FilearchivingProcessImplServiceSoapBinding"
  name="FilearchivingProcessImplPort">
<soap:address location="http://localhost:8080/
  FilearchivingProcess"/>
</wsdl:port>
</wsdl:service>
```

(3) JPD L

JPD L 其实是一套 XML 语言,这套语言有特定的节点,每个节点代表不同的流程定义规则。通过 XML 对流程进行描述,JBPM 去解析这套 XML 语言,然后把得到的信息存储到数据库中。

下面是归档流程的 JPD L 部分代码,主要是开始状态填写归档表,指定管理员,并且有提交动作,同时 JBPM 需要进行数据处理。

```
<start-state name="填写归档表">?
<task name="填写归档表">
<assignment
  class="com.archive.assign.ApplicantHandler">
</assignment>?
</task>?
<transition to="主管审批" name="主管审批">
<action name="Submit"
  class="com.archive.action.SubmitHandler">
  </action>
</transition>?
</start-state>
...
```

对于传统工作流已不能满足分布式异构和松散耦合的企业信息资源的要求,本文在传统的工作流技术基础上,利用 SOA 技术,结合 JBPM,提出了一种基于 SOA 和 JBPM 的工作流引擎模型。本文目前主要研究的是模型的构造与工作流引擎的流程管理与任务控制方面,今后主要工作将是对服务并发数的控制与均衡进一步完善。

参考文献

- [1] 毛新生.SOA 原理·方法·实践[M].北京:电子工业出版社,2007.
- [2] 胡奇.JBPM4 工作流应用开发指南[M].北京:电子工业出版社,2010.
- [3] 蔡孝武,韩永国,蓝科.一种轻量级工作流引擎的研究与设计[J].计算机工程,2010,36(20):30-33.
- [4] 朱永华,张倩,吴俊杰.基于操作的轻量级工作流模型的研究与设计[J].计算机工程与设计,2012,33(1):153-157.
- [5] Guo Xiaofeng, Shen Jianjing, Wu Shanming. On Workflow Engine Based on Service -Oriented Architecture [C]. 2008International Symposium on Information Science and Engineering, Shanghai, 2008.
- [6] Wang Huijiao, Huang Tinglei, Chen Zhifen. Research and Design of the Workflow Engine based on Web Services[C]. Intelligent Computing and Integrated Systems, Guilin,2010.
- [7] Yu Jie, Yu Zhang. The Research and application of Web-Application Based Workflow Engine [C]. 2011Communication

Software and Networks, 2011.

- [8] HEINIS, THOMAS. Design and evaluation of an autonomic workflow engine[C]. Autonomic Computing, Seattle, WA. 2005.
- [9] Liu Peng. Research on workflow patterns based on JBPM and JPDL[C]. Computational Intelligence and Industrial Application, 2008.
- [10] Liu Guanjun, Jiang Changjun, Zhou Mengchu. Interactive Petri Nets[J]. Systems, Man, and Cybernetics: Systems, 2013(43):291-302.
- [11] KIM K P. BPM-supported social networks discovery and analysis techniques[C]. Kyonggi Univ.Advanced Communication Technology (ICACT),2013.
- [12] DU Y, TAN W, ZHOU M C. Timed compatibility analysis of web servicecomposition; a modular approach based on petri nets[J]. Automation Science and Engineering, 2013:1-13.

(收稿日期:2013-05-06)

作者简介:

张辉栋,男,1989年生,硕士研究生,主要研究方向:工作流。

卢选民,男,1972年生,博士后,副教授,主要研究方向:智能信息处理,多媒体通信与计算机网络。