

# 一个支持服务协作的 workflow 管理系统

## Workflow Management System Supporting Service Collaboration

马 华 (湖南涉外经济学院 计算机系 湖南省长沙市 410205)

张红宇 (中南大学 商学院 湖南省长沙市 410083)

**摘要:**面向服务计算环境下, workflow 中的服务不仅是逻辑执行单元,也是业务协作单元。基于一个支持服务协作的 workflow 元模型及其建模语言 SoXPDL,应用条件化有向图方法进一步分析了面向服务工作流的概念模型和实例模型,并提出了支持服务协作的 workflow 引擎驱动算法,最后,基于开源 workflow 项目,设计并实现了一个支持服务协作的 workflow 管理系统原型 SoFlow。

**关键词:** workflow 管理系统 服务协作 引擎驱动算法 SoFlow

### 1 引言

当前,分布式应用集成的范围正逐渐从 Intranet 扩展到了 Internet。同时,激烈的市场竞争要求企业的分布式应用系统的协作更加具有灵活性和动态性。传统的 WfMS,如基于持久消息的 Exotica/ FMQM 系统、基于可移动代理的 DartFlow 系统和基于 CORBA 的 METEOR2 系统等<sup>[1]</sup>,均已无法适应当前的需求。由于 Web 服务具有很好的松耦合和标准化等优点,所以,Web 服务与 workflow 技术相结合已经成为大势所趋。

型<sup>[2]</sup>,我们已经定义了一种支持服务协作的 workflow 元模型以及相应的面向服务工作流建模语言 SoXPDL<sup>[3]</sup>。本文的研究内容,将主要基于文献<sup>[3]</sup>的理论基础,进一步研究如何构建一个支持服务协作的 workflow 管理系统,解决的主要问题包括定义支持服务协作 workflow 的形式化模型、workflow 引擎的核心驱动算法,以及基于开源项目的支持服务协作 workflow 的系统设计与实现。

### 2 workflow 形式化模型

基于我们提出的支持服务协作 workflow 元模型<sup>[3]</sup>(如图 1 所示),以下进一步给出了应用条件化有向图方法定义的概念模型及实例模型。概念模型描述了支持服务协作 workflow 定义时 (Build Time) 的静态工作特征;而实例模型则描述了支持服务协作 workflow 在运行时 (Run Time) 的动态执行特征。

#### 2.1 概念模型

(1) 定义 1。流程定义  $p$ ,是一个六元组  $\langle id, A, T, P, FP, C, Attr \rangle$ ,是业务过程的数学描述。

$id$  是  $p$  的唯一标识(以下均同)。  $A = \{a_1, a_2, \dots, a_n\}$ ,是流程定义中所有活动  $a$  的集合。  $T$  是所有活动间转

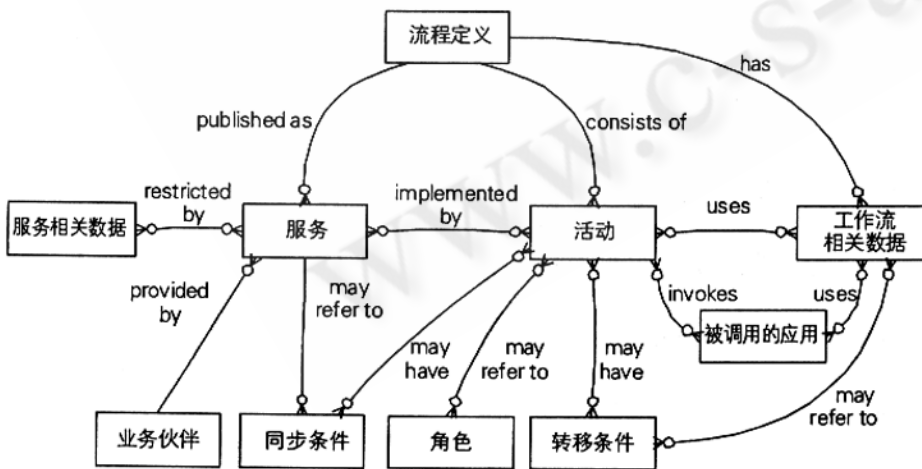


图 1 面向服务工作流的过程定义元模型

面向服务计算环境下, workflow 中的服务已成为一种业务协作的单位。通过扩展 WfMC 的传统 workflow 模

移  $t$  的集合。FP 是形式参数集合。C 为流程容器, 是  $A$  所使用的公共元素集合, C 是一个五元组  $\langle PA, AP, RD, GS, PT \rangle$ , 依次定义了资源/角色、应用、 workflow 相关数据、服务和业务伙伴的集合。Attr 为五元组  $\langle name, accesslevel, version, uthor, created \rangle$ , 是流程属性集合, 依次定义  $p$  的名称、访问级别、版本号、创建者和创建时间。

(2) 定义 2。活动定义  $a$ , 是一个八元组  $\langle name, desc, type, exectrl, implementation, performer, transrestrict, CTX \rangle$ , 表示  $p$  中一个实际或抽象的工作步骤。Executctrl 定义  $a$  的执行控制模式, 决定活动的开始和结束方式。type 定义  $a$  的类型, 可定义为人工交互、自动化、路由、子流程和服务活动类型。implementation 定义  $a$  的实现方式。ActP 是实际参数集合,  $\forall actp_i (\exists rd_i (rd_i \in RD \wedge rd_i.name = actp_i))$ 。performerper 是  $a$  的参与者。transitionrestriction 定义  $a$  的合并类型和分支类型。CTX 是活动上下文。

(4) 定义 4。Web 服务  $ws$ , 是一个七元组  $\langle id, pt, desc, location, operation, FP, gsrdr \rangle$ , 表示被  $a$  引用的服务。pt 定义  $ws$  的提供者。description 是  $ws$  的语义描述。location 是  $ws$  的 URL。operation 指明了要调用的功能名。FP 是形式参数集合。wsrdr 是四元组  $\langle execution, exception, issubtransaction, coordinator \rangle$ , 定义服务相关数据, 依次指明  $ws$  的同/异步执行方式、异常处理策略、是否要求事务保证、服务协调器的 URL。

(5) 定义 5。业务伙伴 (Partner)  $pt$ , 是一个五元组  $\langle name, address, email, telephone, type \rangle$ , 表示参与  $p$  的业务伙伴, 分别定义了  $pt$  的名字、地理位置、联系邮箱、联系电话和服务类型。

(6) 定义 6。前驱活动集  $PreA(x)$ ,  $PreA(x) = \{ a_i | a_i \in A \wedge (\exists t_i (t_i \in T \wedge t_i.from = a_i \wedge t_i.to = x)) \}$ 。

(7) 定义 7。后继活动集  $PostA(x)$ ,  $PostA(x) = \{ a_i | a_i \in A \wedge (\exists t_i (t_i \in T \wedge t_i.from = x \wedge t_i.to = a_i)) \}$ 。

(8) 定义 8。起始活动集  $sa$ ,  $\forall a_i (a_i \in sa \implies |PreA(a_i)| = 0)$ 。

(9) 定义 9。最终活动集  $ea$ ,  $\forall a_i (a_i \in ea \implies |PostA(a_i)| = 0)$ 。

## 2.2 实例模型

(1) 定义 10。流程实例  $pi$ , 表示  $p$  的一次执行过程。它是一个七元组  $\langle id, pid, state, started, lastchanged, CTX, Active \rangle$ 。pid 是  $pi$  所属  $p$  的 id。state 是  $pi$  的当前状态。Started 和 lastchanged 分别表示  $pi$  的启动和最近一次状态改变的时间。CTX =  $\{ ctx_1, ctx_2, ctx_3, \dots, ctx_n \}$ , 为流程上下文。ctx<sub>i</sub> 是一个三元组  $\langle name, value, lastchanged \rangle$ , 依次定义了变量名、当前值和最近一次被修改的时间。  $\forall ctx_i (\exists rd_i (rd_i \in RD \wedge rd_i.name = ctx_i.name) \vee \exists fp_k (fp_k \in FP \wedge fp_k.name = ctx_i.name))$ 。Active 是当前激活的活

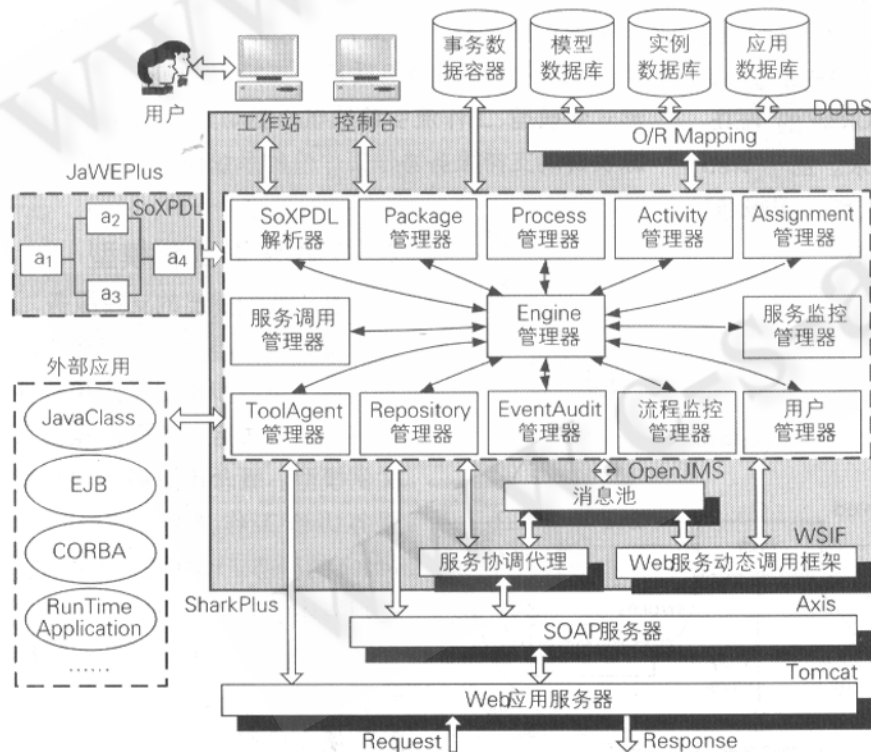


图 2 SoFlow 的总体结构

(3) 定义 3。转移  $t$ , 是一个五元组  $\langle id, a_{from}, a_{to}, type, exp \rangle$ , 表示活动间控制关系与数据流动关系。依次定义了  $t$  的起始活动、终止活动、转移类型和条件表达式。

动实例集合。

(2) 定义 11。活动实例  $ai$ , 表示  $a$  的一次执行过程。它是一个九元组  $\langle id, piid, aid, state, theresource,$

started, accepted, lastchanged, CTX >, 除 id 外依次定义了 ai 所属 pi 的 id、ai 所属 a 的 id、state 是 ai 的当前状态、ai 的实施者、ai 启动时间、最近一次状态改变的时间和 ai 任务被用户接受的时间、活动上下文。

(3) 定义 12. 任务 ass, 表示需要人工交互的工作项。ass 由 ai 产生, ai 激活时将根据 a 绑定的 performer 为角色/组织机构所映射的具体执行人分配任务。它可以表示为一个三元组 < id, aiid, uname >, aiid 是 ass 所属的 ai 的 id; uname 是负责该任务的用

户名。

### 3 workflow 系统 SoFlow

#### 3.1 总体结构

Shark 和 JaWE 是国际组织 Enhydra 的开源 workflow 研究项目<sup>[4]</sup>, 它们严格遵循了 WfMC 的相关规范, 是一个典型的传统 workflow 管理系统。通过对 Shark 和 JaWE 的扩展, 我们开发了一个互操作性好、支持服务协作的 workflow 管理系统 SoFlow, 如图 2 所示。

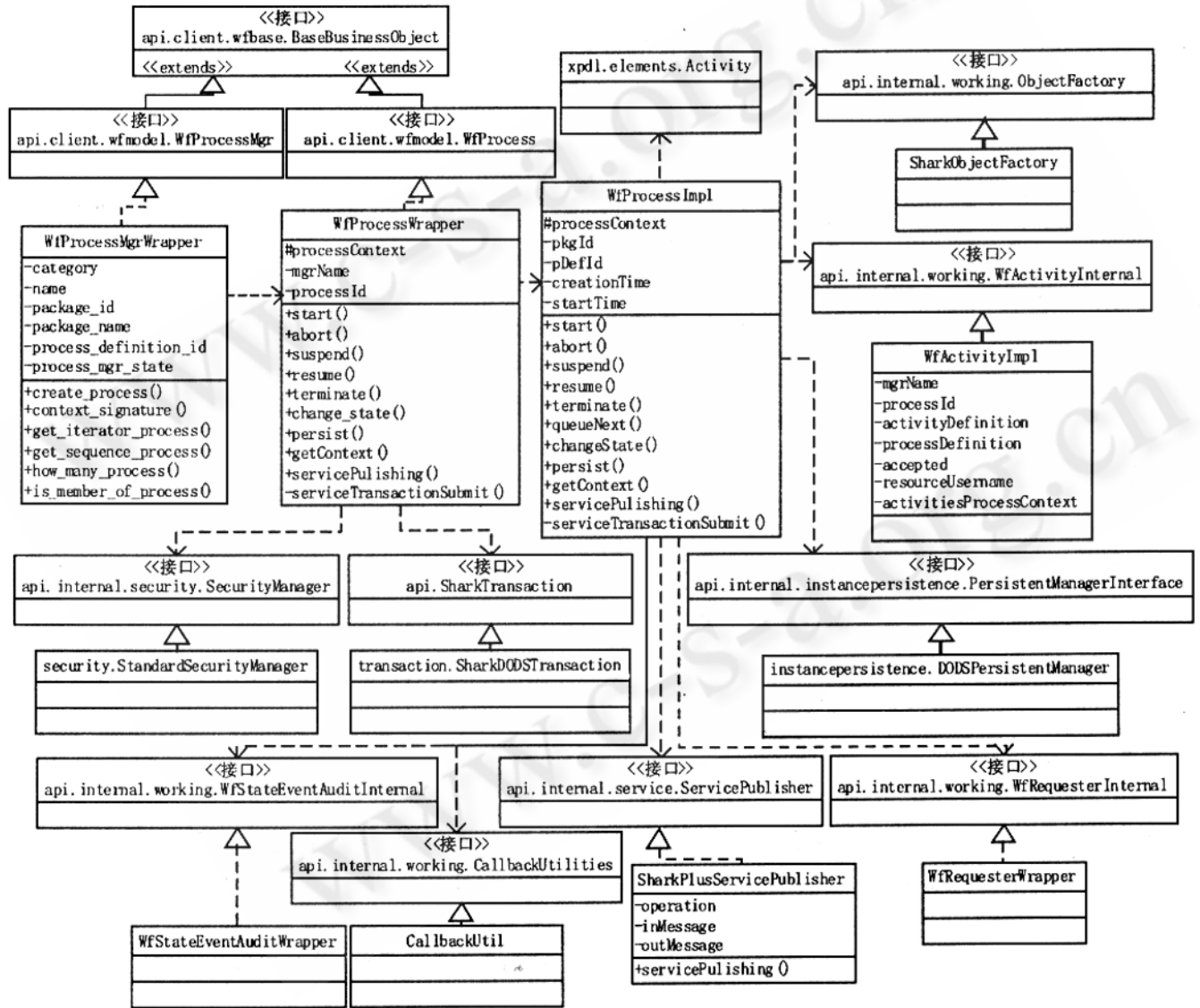


图 3 SharkPlus 的 Process 管理器

SoFlow 主要由面向服务 workflow 引擎 SharkPlus 和建模工具 JaWEPlus 组成。在 SharkPlus 中, 我们基于开

源 JMS 项目 (OpenJMS) 设计的消息池实现了对服务的异步协调支持, 而基于开源 WSIF 框架, 我们实现了

对服务运行时的动态调用支持,并设计了服务协调代理来处理支持服务协作工作流中的事务一致性要求。JaWEPlus 工具支持我们定义的 SoXPDL 语言。

### 3.2 系统设计

SharkPlus 在 Shark1.1.2 版本的基础上进行开发,通过保持对 WfMC 的应用编程接口规范<sup>[5]</sup>的兼容性,实现了一个互操作性好、支持 SoXPDL 的工作流引擎。SharkPlus 包括了 Engine 管理器、SoXPDL 解析器、Process 管理等九大核心组件和消息池、WSIF、服务协调代理等辅助组件。在类的设计上,严格遵循了开放封闭原则、依赖性倒置原则、接口分离原则和最少知识原则等,并灵活地应用了包装器模式、抽象工厂模式和单例模式等设计模式。在包的设计上,遵循了通用闭包原则、通用重用原则、无环依赖原则和稳定抽象原则等。本节主要介绍 Process 管理器的类图设计(SharkPlus 其它部分及 JaWEPlus 的介绍略),如图 3 所示。Process 管理器主要由两个包装器类(WfProcessMgrWrapper 和 WfProcessWrapper)来实现。前者是流程定义类,用于管理流程模型及实例化流程;流程实例的操作则主要通过后者来实现。

## 4 总结

目前,SoFlow 的研究成果已经在湖南省的多个电子政务和办公自动化系统中得到了初步的应用,有效

地实现了面向服务计算环境下的服务协同。本文所建立的面向服务工作流形式化模型,为深入研究支持服务协作的工作流技术的其它问题提供了理论基础,通过分析面向服务工作流引擎的驱动算法,在开源项目基础上实现的支持服务协作的工作流系统 SoFlow,为面向服务计算环境下工作流技术研究提供了新的思路和方法。下一步我们将研究支持 SoFlow 的事务、异常处理模型和安全模型。同时,进一步完善 SharkPlus 和 JaWEPlus,从而最终实现 SoFlow 的实用化。

### 参考文献

- 1 范玉顺、罗海滨、林惠萃等, 工作流管理技术基础 [M], 北京: 清华大学出版社, 施普林格出版社, 2001.
- 2 WfMC. The Reference Model and API Specification [R]. 1996.
- 3 马华、张红宇、李建华, 支持服务协作的工作流元模型和建模语言[J], 计算机应用. 2007, 27 (02): 409-412.
- 4 ObjectWeb. Enhydra [OL], <http://www.objectweb.org/>.
- 5 WfMC. Workflow Management Application Programming Interface Specification [R]. 1998.