计算机科学2000Vol.27№.11

工作流互操作规范

Workflow Interoperability Specification

葛羽嘉 石文俊 吴朝晖

(浙江大学计算机系统工程研究所 浙江大学计算中心 杭州310027)

Abstract To support interoperability between different workflow engines, Workflow Management Coalition established the standard [WFMC1021] to provide an abstract specification which defines the principles and functions of interoperability. According to this document, workflow vendors can scale their current products and develop standard ones. This paper covers strategies of implementation, models of interoperability and levels of interoperability on workflow interoperability.

Keywords Workflow, Interoperability, Workflow engine

一、目的

目前不同的工作流产品有各自的工作流引擎，当某个工作流引擎请求另一个工作流引擎对某个已知的流程定义进行选择、实例化、设定等动作时，工作流引擎之间需要互相通信。另外，发出请求的引擎应该收到流程定义执行的返回状态信息。同时，有必要对产生的数据进行审核以保障互操作的顺利进行。

为了提供不同工作流产品间互操作的功能，工作流管理联盟(Workflow Management Coalition)的互操作标准[WFMC1021]定义了相关机制。这一接口是针对工作流产品的生产商，希望他们的工作流产品尽可能符合标准，更易实现互操作，同时采取对用户透明的方式，另外明确了不同的互操作层次，使工作流产品厂商能以此衡量他们的产品。

二、互操作问题描述

2.1 互操作相关概念

在工作流管理联盟词汇定义中[WfMC000]，工作流互操作是这样定义的：

互操作是两个或多个工作流引擎为了协调和执行工作流程实例，而相互通信和互相操作的能力。互操作可发生在以下三种情况：

·两个或多个工作流引擎之间直接互操作；

·在同一个定制服务中的两个或多个工作流引擎之间的互操作；

·在工作流管理范围内的两个或多个工作流服务(即两个或多个工作流引擎运作在两个或多个工作流服务)之间的互操作。

工作流服务是为流程的启动和执行提供运行环境，它利用一个或多个工作流引擎解释并执行流程定义，或与外部资源相互作用。这里的外部资源可以是：①用户代理(以工作流客户端的应用为中介)；②能执行一定任务的软件工具；③其他工作流引擎。

从以上可知，两个有不同运行环境的流程引擎可以当作不同工作流服务。

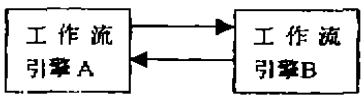

图1 工作流引擎间直接互操作

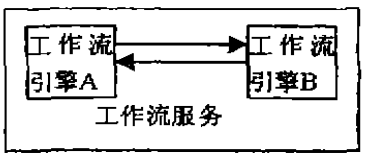

图2 同一服务中工作流引擎的互操作

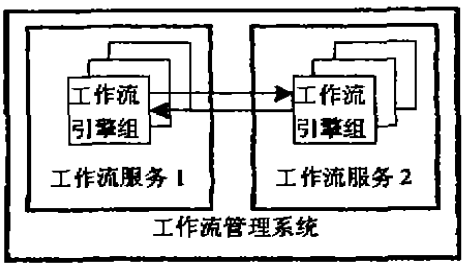

图3 不同服务中工作流引擎的互操作

2.2 互操作实现途径

软件之间的互操作通常认为是共享数据或功能的能力。这里的软件工具是指能实行一定功能的软件，如文字编辑器、信件收发工具、数据库引擎或工作流管理系统。

互操作实现的方式通常为以下几种：

1)工具间直接作用(图4)

2)消息传递(图5)

3)网桥(通过一定形式的封装、转换或网关机制)(图6)

4)利用共享数据存储(图7)

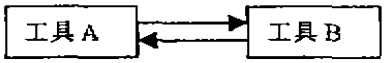

图4 直接互操作

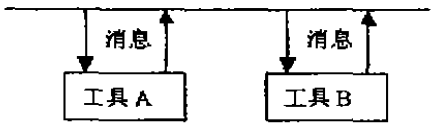

图5 通过消息传递实现互操作

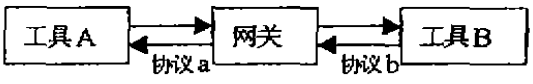

图6 通过网关实现互操作

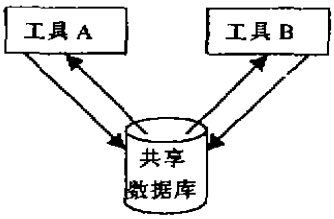

图7 通过共享数据实现互操作

参考模型虽然没有讲述第4种方法，鉴于有些工作流产品通过内部数据库将任务转移，因此，用共享数据库来转移工作包也被认为是一种实现互操作的方法。从一个抽象层次上说这个方法可以看成存储-运送机制(store-forward mechanism)的另一种形式。

三、衡量产品互操作能力的层次

工作流管理联盟互操作白皮书明确互操作有8个层次，各个层次由工作流引擎的体系结构及相关的操作特性区分：

·层次1：无互操作　这个层次的产品没有与其他产品通信的方法，因此没有互操作的功能。

·层次2：共存　这个层次上没有实现工作流产品间互操作的标准方法。然而工作流厂商为了使他们的产品应用在不同平台上，正努力制定工业及全球标准。这个层次特征是工作流产品共享一个运行环境(硬件、操作系统、网络)，存在利用 WAPI 接口实现互操作的工作流产品与没有 WAPI 的工作流产品在同一平台共存的情况。这个层次不要求不同工作流产品任何直接的互操作，但用户可以根据自己的需要，利用适合自己工作的不同工作流产品，组合完成整个流程，不同工作流产品通过人的参与完成，由人决定这个流程结束后开始哪个产品。

·层次3：利用"网关"　"网关"是一种允许工作流产品互相之间进行工作交换的机制(如图6)。在一定的环境中，某个应用可能是两个工作流系统的"网关"，"网关"通过协议转换器将数据和命令从一个域转换到另一个域，"网关"的执行也随着提供转换方式不同而存在差异。"网关"能实现数据对象控制权从一个工作流管理系统转移到另一个系统。当多于两个工作流实例参与时，"网关"必须能执行正确的路由操作。

工作流互操作白皮书定义"网关"的两个层次。一是单一"网关"，这个层次的工作流产品与一定的网关机制(工作流引擎和实例间的路由操作、工作流相关数据的转换和传递、工作流应用数据的转换和传递)一同工作。

二是共同"网关"API。这个层次的"网关"有一个共同的标准 API。这个层次表明，不同网关机制所支持的操作都规格化为标准支持的共同子集，但并不排除超集的可能。

·层次4：有限的子集　这个层次的工作流产品共享一个标准 API，使其直接互操作，在它们之间转移和管理工作。执行这个层次的互操作要求 API 函数的核心集定义在一个公开的标准中，大多数的工作流引擎都能执行那个 API。这个层次的执行模型实际上都比较简单，基于 API 和封装。

·层次5：完全的工作流 API　这个层次的工作流产品都具有一个标准 API，能进行任何工作流管理系统的操作。当然也不排除为了满足一定的市场需求的工作流产品开发特定的功能。因为工作流产品类型的广泛性，有必要对工作流产品做一定限制，而且这层上实现操作的都可以从第四层演进过来。重点要求需要定义一个共同操作集，这些操作必须影射到每个工作流产品的操作上来支持这个层次的互操作。这个层次的映射机制与层次4的执行模型相同。

·层次6:标准的工作流定义形式 这个层次的工作流产品有一个工作流定义的标准形式,包括路由选择、用户访问权限和工作流系统资源的维护,任一个工作流系统产生的每个流程定义、任何工作流引擎都能执行它,并保证流程正确性。我们可以简单认为,所有的工作流产品都将支持一定集合的操作,否则就不是工作流产品。

实现这个互操作标准有两个步骤:1)定义普通功能集:具有所有工作流产品的共同属性;2)定义不同工作流类型的特定的功能操作:特殊的功能用来支持解决不同问题领域。

工作流管理联盟定义的流程定义语言(WPDL)[WFMC0020]完成第一步的工作。附加功能可以用扩展语言定义,包含类型、函数、编译和运行的库函数。

·层次7:协议兼容 这个层次假定所有的客户/服务器的API通信都是标准化的,包括定义转换、工作流事务、恢复。为了达到这个层次的互操作,工作流产品必须提供各种机制来实现这样的互操作。

·层次8:通用界面 这个层次假定除了前面各个层次,所有工作流产品呈现给用户相同的标准界面,起码用户使用感觉是相同的。但由于商业和实际的某些原因,这个层次可能永远都不能实现。

四、流程间互操作模型及实现

4.1 链锁的流程

主工作流引擎启动另一个工作流引擎的子引擎,但不需要等待子流程完成,就可继续执行。如图8。

工作流引擎A必须完成:

1)选择一个工作流引擎B管理或能访问的已知流程

2)传递数据给工作流引擎B

3)请求工作流引擎B启动该流程

可能有以下几种情况:

1)工作流引擎A管理流程定义,在需要时,传递给工作流引擎B.

2)工作流引擎B管理流程定义,工作流引擎A知道并可要求启动它。

3)流程定义保存在一个共享区内,任一流程都可访问它。

为了标准的需要,简化为两种:1)工作流程定义可由一个工作流引擎传递到另一工作流引擎;2)工作流程定义可由一个启动它的工作流引擎访问.

由于标准[WfMC0020]定义了1)所要求的机制,这儿在假定情况2)下。因此,上图的两个工作流引擎所需要的操作如下:

工作流引擎	操作
A	创建流程实例
B	回应创建流程实例
A	设置流程实例属性
B	回应设置流程实例属性
B	获得流程实例属性
A	回应获得流程实例属性
A	启动流程
B	回应启动流程
A	撤消流程
B	回应撤消流程

另外,子流程还可以启动其他子流程。

4.2 嵌套的子流程

主工作流引擎启动另一个工作流引擎的子引擎,等待子流程完成,才可继续执行。如图9。

我们可以利用工作流管理工具来返回当前流程的状态,实现相对于不同工作流引擎的流程调度。用WAPI接口2[WfMC1009]和WAPI接口3[WfMC0013]可以实现工作流管理工具。

4.3 平行同步的流程

两个工作流平行执行,当一个工作流程达到指定点时,等待另一个流程的到达那一点,然后进行一些交互活动,然后各自继续完成。

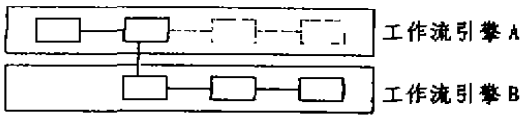

图8

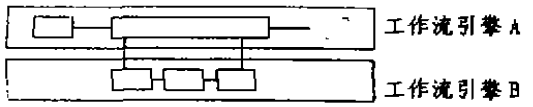

图9

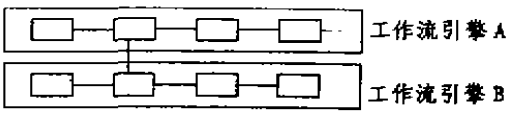

图10

4.4 用接口定义语言(IDL)实现

4.4.1 实现方法 各个互操作模型所需的操作需要详细描述,互操作标准用IDL(Interface Define Language)来描述,IDL描述是工作流引擎间产生互操作所必需的操作的抽象表示。消息描述是工作流引擎之间所需要传递的信息,消息的格式描述将支持同步和异步的工作流互操作。有三种不同类型的消息:

请求消息：当一个工作流引擎需要另一个工作流引擎执行一定的动作；

响应消息：告诉提出要求的工作流引擎的结果，即被要求的行为是否执行及原因；

通知消息：在一个互操作过程中，子过程需要通知父过程它已经达到了一定事件，若是子流程由于某种原因失败，父流程可根据消息继续执行。每一个通知消息由一个回答消息来回应。

4.4.2 操作描述 标准定义了连接操作、流程控制交互作用的16个操作，包括：

1）产生流程实例；

2）获得流程实例的状态和改变流程实例状态；

3）设置和获得流程实例的属性；

4）启动、中断、取消和终止流程实例；

5）列出流程实例；

6）通知工作流引擎实例已经启动、中断和终止；

7）状态、属性已改变。

下面举例说明它们的标准描述：

操作：生成一个流程实例

```
WMAReturnCode WMRequestCreateProcessInstance(
    in engine_identifier          WMAEngineID,
    in process_definition_id      WMAObjectID,
    in return_flag                WMABool,
    in parent_pid                 WMAObjectID,
    in activety_id                WMA_Activity_id,
    out sub_process_id            WMA_Process_instance_id,
    out user_id                   WMAResourceID,
    out role_id                   WMAResourceID
```

请求消息格式

域值	描述
ProcessDefinitionID	流程定义标识符
ReturnFlag	True：嵌套的子流程 False：链锁的子流程
ParentPID	父流程标识符
ProcessID	要求产生新流程的工作流引擎上流程标识符
ActivityID	要求产生新流程的活动标识符
Timestamp	用于工作流引擎传递请求时
SourceNodeID	发起请求的工作流引擎的标识符
SourceUserID	与请求有关的用户号
SourceRoleID	与请求有关的角色号
TargetUserID	与目标工作流引擎的流程有关的用户号
TargetRoleID	与目标工作流引擎的流程有关的角色号
SourceBusinessDefinition ProcessName	请求流程的商业定义
MessageRoutingInfor mation	路由信息

这个操作确认了一个流程定义，使一个工作流引擎可以让另一个工作流引擎启动某一个有明确定义的流程，并执行它。同时，当一个工作流引擎产生新流程时，它需要知道是否返回状态信息给要求的工作流引擎。当此标志为 TRUE 时，所有的状态信息都通知那个工作流引擎，否则不返回新流程的信息。

响应消息格式

域值	描述
ReturnCode	Success|Failure|No_operation
Timestamp	用于当被请求流程创建时
ProcessID	新产生流程号
UserID	主要用户的用户号
RoleID	主要用户的角色
TargetProcessBusiness DefinitionName	可为空值
TargetState	新流程的状态
DomainID	互操作工作流引擎的域
TargetNodeID	目标工作流引擎节点
Message Routing Information	路由信息

另外，一些工作流引擎动作相关记录会保存下来，说明此操作顺利完成。

五、总结和展望

工作流互操作标准的制定，使各个工作流产品更好地进行交互，不同工作流产品间的工作切换提供了一定的灵活性，提高了一个组织整体的效率和反映能力。

另外，在工作流标准（WFMC-TC-1018）中，提供了一种抽象描述来实现多个工作流引擎之间的互操作。这个文档给出了两个工作流引擎间类型定义和消息的形式，定义必要的功能。它用 e-mail 的 MIME 编码作为传送机制。WFMC1012中描述了实现互操作的消息具体定义，它将消息的抽象定义映射到 e-mail 的简单文本接口中。

我们正在开发的工作流产品符合工作流管理联盟制定的标准接口，所以有能力达到互操作的第六个层次。各厂商的工作流产品若能尽可能符合标准，将更易实现互操作。

参 考 文 献

1 Workflow Management Coalition, Workflow Standard-Interoperability Abstract Specification (WFMC-TC-1012), 1996.10

2 Workflow Management Coalition, Workflow Standard-Interoperability Internet e-mail MIME Binding (WFMC-TC-1018), 1998

3 Workflow Management Coalition, Terminology & Glossary (WFMC-TC-1011), 1996.6