

基于 CORBA/XML 的工作流系统管理和监控接口技术研究与实践*

梁彦强, 顾明

(清华大学 软件学院, 北京 100084)

摘要: 结合 CORBA 技术强大的跨平台能力, XML 技术良好的灵活性与拓展性, 提出了工作流管理与监控接口的一种实现技术, 能有效地满足工作流系统在开放、分布式处理以及高互操作性上的要求。

关键词: CORBA; XML; 工作流; 管理与监控接口

中图分类号: TP311

文献标识码: A

文章编号: 1001-3695(2006)02-0155-02

Study and Implementation of Administration and Monitoring Interface Technology Based on CORBA and XML for Workflow Management System

LIANG Yan-qiang, GU Ming

(Software School, Tsinghua University, Beijing 100084, China)

Abstract: This article provided an implementation technology of administration and monitoring interface, which is based on the platform-independency of CORBA and the flexibility and extensibility of XML. It can meet the requirement that workflow management system needs in the aspects of open, distribution and high cross-operation.

Key words: CORBA; XML; Workflow; Administration and Monitoring Interface

随着网络技术的日益发展, 分布式技术在软件系统中的应用越来越广泛, CORBA 和 XML 近年来成为了计算机分布式技术研究的两大热点。自 1993 年国际工作流管理联盟 (Workflow Management Coalition, WfMC) 成立以来, 该联盟制定了有关工作流系统的一些技术标准, 其中包括了工作流参考模型中定义的五类应用编程接口 WAPI^[1]。迄今为止, 现有的各种工作流系统的管理与监控接口 (应用编程接口 WAPI 接口 5) 多是通过一种紧耦合的方式实现, 开放性、灵活性和互操作性都较差。本文通过对工作流管理与监控接口结构与功能的分析, 结合 CORBA^[2] 和 XML^[3] 在分布式应用上的优势, 提出了一种基于 CORBA 和 XML 的接口实现技术, 使得工作流管理与监控接口具备了较高的开放性、灵活性和互操作性。

1 工作流系统管理和监控接口的功能

为了实现工作流技术的标准化和开放性, 从而支持异构工作流系统以及不同工作流产品之间的互操作, WfMC 制定了工作流系统的体系结构。在 WfMC 提出的工作流参考模型中, 定义了五类应用编程接口 WAPI, 如图 1 所示。

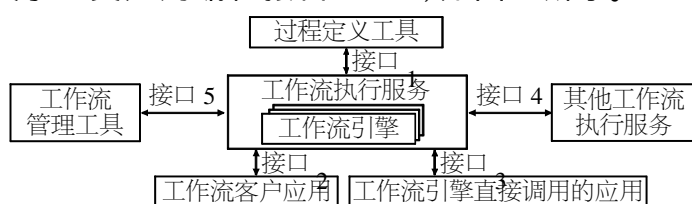


图 1 工作流参考模型

其中, 工作流系统管理工具通过管理和监控接口 (接口 5) 实现对工作流在整个组织内的流动情况的监视, 并提供一系列的管理功能。该接口包括了以下几个方面的内容^[4]:

- (1) 资源控制。设置/取消/修改过程或者活动实例的并发层次。
- (2) 角色管理。定义、删除和修改角色 - 参与者 (Role : participant) 的关系, 修改角色属性。
- (3) 用户管理。对用户或工作组的权限的建立、删除、暂停和修改。
- (4) 过程实例的管理。对过程模型进行实例化, 修改过程实例或者活动实例的状态, 终止过程实例。
- (5) 状态管理。查询过程或活动实例的运行状态。
- (6) 审核管理。查询/打印/删除/启动新的统计过程轨迹或日志。

2 基于 CORBA/XML 的管理和监控接口的实现技术

2.1 管理和监控接口的总体设计

基于 CORBA/XML 实现的工作流管理和监控接口。其总体结构如图 2 所示。



图 2 管理和监控接口的总体设计

2.2 管理和监控接口的工作过程

当 workflow 管理工具通过管理和监控接口对 workflow 系统进行控制时, 管理工具首先将控制命令封装成 XML 命令数据, 然后通过 CORBA 技术找到远程目标对象, 并以 XML 命令数据作为参数调用远程目标对象的处理方法。当远程目标对象的处理方法被调用的时候, 首先解析作为方法参数的 XML 命令数据, 得到具体的控制命令, 然后通知 workflow 执行服务完成相应操作, 并将 workflow 执行服务的操作结果封装成 XML 结果数据返回给调用者。管理工具收到 XML 结果数据之后, 解析得到处理结果, 从而完成控制操作。整个操作过程如图 3 所示。

2.3 IDL 接口定义

CORBA 接口 IDL 定义文件的内容如下:

```
module systemcontrol {
    interface Controller {
        string process(in string xmlCommand);
    };
};
```

IDL 文件定义了一个 Controller 远程目标对象以及 Controller 对象的一个 Process 处理方法。Process 方法以 XML 格式的控制命令字符串为参数, 同时返回一个 XML 格式的处理结果字符串。

2.4 XML 控制命令

控制命令 XML 字符串中指定的文档类型定义文件 xmlCommand.dtd 的内容如下:

```
<?xml version="1.0" encoding="UTF-8"? >
<!ELEMENT Command ( ManageRoleCommand | ManageUserCommand | ManageProInsCommand | ManageActInsCommand | ManageCheckCommand ) >
<! ENTITY % ManageRoleCommand SYSTEM " ManageRoleCommand.dtd " >
% ManageRoleCommand;
<! ENTITY % ManageUserCommand SYSTEM " ManageUserCommand.dtd " >
% ManageUserCommand;
<! ENTITY % ManageProInsCommand SYSTEM " ManageProInsCommand.dtd " >
% ManageProInsCommand;
<! ENTITY % ManageActInsCommand SYSTEM " ManageActInsCommand.dtd " >
% ManageActInsCommand;
<! ENTITY % ManageCheckCommand SYSTEM " ManageCheckCommand.dtd " >
% ManageCheckCommand;
```

这段文档类型定义规定: 发送的控制命令可以是管理角色命令、管理用户命令、管理过程实例命令、管理活动实例命令或者管理审核命令。这五种控制命令分别对应了 workflow 管理和监控应用编程接口 WAPI 应包含的五个方面内容, 即角色管理、用户管理、过程实例管理、活动实例管理以及审核管理。各种控制命令中包含的具体控制命令则由对应的 DTD 文件来定义。以管理活动实例命令为例, ManageActInsCommand.dtd 定义文件的内容如下:

```
<?xml version="1.0" encoding="UTF-8"? >
<!ELEMENT ManageActInsCommand ( SetStatusCommand | GetSta-
```

```
tusCommand) >
<! ELEMENT SetStatusCommand EMPTY >
<! ATTLIST SetStatusCommand
InstanceID CDATA #REQUIRED
Status ( completed | suspended | terminated | running) #REQUIRED
>
<! ELEMENT GetStatusCommand EMPTY >
<! ATTLIST GetStatusCommand
InstanceID CDATA #REQUIRED
>
```

这段文档类型定义规定: 管理活动实例命令有设置活动实例状态命令和查询活动实例状态命令两种, 设置活动实例状态命令有活动实例号和预设置状态两个参数; 查询活动实例状态命令只有活动实例号一个参数。

2.5 XML 处理结果

与控制命令 XML 字符串相似, 处理结果 XML 字符串中指定的文档类型定义文件 xmlResult.dtd 的内容如下:

```
<?xml version="1.0" encoding="UTF-8"? >
<! ELEMENT Result ( ManageRoleResult | ManageUserResult | ManageProInsResult | ManageActInsResult | ManageCheckResult ) >
<! ENTITY % ManageRoleResult SYSTEM " ManageRoleResult.dtd " >
% ManageRoleResult;
<! ENTITY % ManageUserResult SYSTEM " ManageUserResult.dtd " >
% ManageUserResult;
<! ENTITY % ManageProInsResult SYSTEM " ManageProInsResult.dtd " >
% ManageProInsResult;
<! ENTITY % ManageActInsResult SYSTEM " ManageActInsResult.dtd " >
% ManageActInsResult;
<! ENTITY % ManageCheckResult SYSTEM " ManageCheckResult.dtd " >
% ManageCheckResult;
```

这段文档类型定义规定: 返回的处理结果可以是管理角色命令处理结果、管理用户命令处理结果、管理过程实例命令处理结果、管理活动实例命令处理结果或者管理审核命令处理结果。这五种处理结果与发送的五种控制命令一一对应。同时, 各种处理结果中包含的具体处理结果也由对应的 DTD 文件来定义, 且与各种控制命令中包含的具体控制命令一一对应。以管理活动实例命令处理结果为例, ManageActInsResult.dtd 定义文件的内容如下:

```
<?xml version="1.0" encoding="UTF-8"? >
<! ELEMENT ManageActInsResult ( SetStatusResult | GetStatusResult ) >
<! ELEMENT SetStatusResult EMPTY >
<! ATTLIST SetStatusResult
value ( true | false) #REQUIRED
>
<! ELEMENT GetStatusResult EMPTY >
<! ATTLIST GetStatusResult
value ( completed | suspended | terminated | running) #REQUIRED
>
```

这段文档类型定义规定: 管理活动实例命令处理结果有活动实例状态设置结果和活动实例查询结果两种(分别与管理活动实例命令中的设置活动实例状态命令和查询活动实例状态命令相对应); 活动实例状态设置结果为成功或者失败; 活动实例查询结果为完成、暂停、终止或者运行。

2.6 管理和监控接口的可拓展性

随着 workflow 技术的发展, WfMC 提出的 workflow 参考模型也将不断的成熟与完善。利用 XML 语言本身具备的特点, 基于 CORBA/XML 技术实现的管理和于监控接口 (下转第 159 页)