

Workflow Management

应用编程接口

数据结构

11

workflow management application programming interface specification

Workflow Management Application Programming Interface Specification

64-67

刘华伟 吴朝晖

TP399

(浙江大学计算机系统工程研究所 杭州310027)

Abstract The document 'Workflow Management Application Programming Interface Specification' is distributed by the Workflow Management Coalition to specify standard APIs which can be supported by workflow management products. In this paper, we first introduce two parts of this interface, then discuss the standardized data structure and functions definition, finally address the future work.

Keywords Workflow management (WFM), Workflow management application programming interface (WAPI)

一、目的

随着 workflow management 技术的迅速发展, 各种各样的 workflow management (WFM) 应用越来越多。然而, 现有的 WFM 产品没有一个统一的标准, 不同的 WFM 产品不能一起工作。WFM 产品之间的不兼容性不仅阻碍了企业流程自动化的发展, 还造成 WFM 开发商不能重用 workflow management 部件, 使开发难度增大, 开发费用增加。有鉴于此, 国际标准化组织 workflow management 联盟发布了《workflow management application programming interface 规范》^[1] (以下简称《WAPI 规范》), 以制定 WFM 产品的应用程序编程接口的标准, 统一众多 WFM 引擎应用编程接口 (API) 的调用方式。这就是下文我们所要介绍的 workflow management application programming interface (Workflow Management Application Programming Interface, WAPI)。

二、概述

在《WAPI 规范》中, WAPI 被分为两部分: 客户端应用编程接口和调用应用接口, 分别对应 workflow 引用模型^[2]中的接口2和接口3。客户端应用编程接口用于编写基于 WFM 引擎功能函数 (workflow 服务) 的客户端应用程序, 比如用户的工作列表桌面。调用应用接口用于激活与完成活动相关的应用程序, 并监控它们的状态, 检测活动是否完成。

2.1 客户端应用编程接口 (接口2)

客户端应用编程接口用于编写基于 WFM 引擎功能函数 (workflow 服务) 的客户端应用程序, WFM 开发者通过标准的 API 接口使 workflow 应用能在不同的 WFM 引擎上工作。这样, 无论系统中存在多少个工作

流引擎, workflow 开发者总能开发出一致的终端用户界面, 实现相同的工作流管理功能。

客户端应用编程接口经常在工作流执行或即将执行时被 workflow 应用程序 (例如, 工作列表处理器、协同应用) 调用。如果其它 workflow 引擎想要和该 workflow 引擎交互, 它也可以调用该 workflow 引擎的 API 函数。

通过接口的函数集, 客户端应用编程接口提供了一套 workflow 引擎支持的工作流服务, 包括: 会话建立、过程控制功能、过程状态功能、工作表/工作项目处理功能、过程监控功能、数据处理功能。接口没有规定特定的用户界面, 使用这些接口构造的用户界面完全取决于 workflow 应用程序的开发环境。

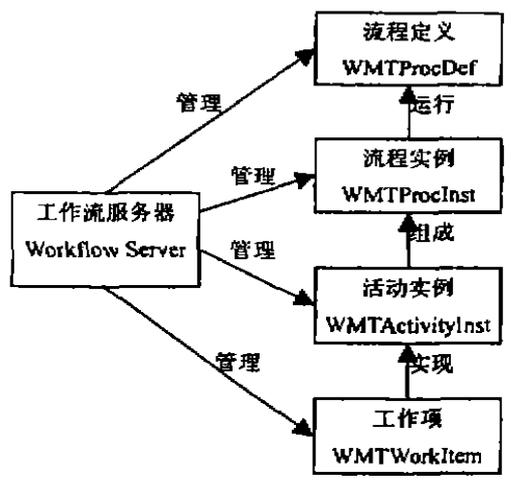


图1 接口2中的对象

图1是客户端应用编程接口主要操作的对象——

WFM(workflow management)对象。WorkflowServer 是 workflow 引擎,提供 workflow 管理服务,WMTProcDef 代表了一个 workflow 的定义,是生成 workflow 实例的模板。在 workflow 实例的执行过程中,workflow 引擎根据活动定义不断产生活动实例。将活动实例指定给一个 workflow 参加者则产生一个 workflow 项。

2.2 调用应用接口(接口3)

引入一个 workflow 管理系统总是意味着集成现有的 IT 环境,或者说“工作流化”。因此,这层“工作流化”接口在一定程度上必须支持现有的软件系统或者调用已有的应用程序,然而在现有情况下,不同的 workflow 管理工具集成应用和交换信息使用的是不同的机制。这些机制为了特殊的用户要求而编写,系统集成公司和第三方开发商如果想要使用别人的 workflow 管理工具就必须重新实现这些机制。一个看上去简单的普通应用的“工作流化”,其 workflow 环境通常包括一系列在异种操作系统运行的不同应用程序。结果,它们的工作量加重,开发费用也增加。市场促使 workflow 管理系统和集成平台需要通用的标准化的应用程序接口,因此,workflow 管理联盟制定了一个特别的应用程序驱动接口——“调用应用接口”。借助于该接口调用和控制应用程序,workflow 管理联盟标准化了 workflow 产品和其它软件系统之间的交互协议。

与其它的接口不同的是:调用应用接口关系到 workflow 系统集成商和第三方软件商。这是因为调用应用并不是 workflow 管理特有的功能,与 workflow 引擎关系不大。由于这个特点,调用应用接口可以用支持 workflow 的“工具代理”实现。“工具代理”使用各种各样的通信技术例如 DDE、OLE、CORBA 等,调用应用程序,实现信息交换。通信技术取决于底层体系结构和应用程序定义的特有接口。

三、接口描述

本节介绍 WAPI 的接口定义标准。《WAPI 规范》使用 C 语言作为定义语言。在它的附录中还有用 OLE 和 OMG IDL 定义的 WAPI 标准。

3.1 数据结构

标准化 WAPI 用到的数据结构包括基本数据类型、WFM 对象等。它们是 WAPI 函数输入、输出参数的数据类型。

基本数据类型是与操作系统或者平台相关的数据类型,如表 1 所示。除此之外,字符串是大小为 STRING_SIZE=64 字节的字符数组,包括字符串结束符 '\0'。标志符 WMTID 是大小为 UNIQUE_ID_SIZE=64 字节的字符数组。

表 1 WAPI 的基本数据结构

	WAPI	操作系统
8位整数	WMTInt8	Char
16位整数	WMTInt16	Short
32位整数	WMTInt32	Long
无符号整数 (8、16、 32位)	WMTUInt8 WMTUInt16 WMTUInt32	Unsigned char……
字符	WMTText	Char
指针	WMTPointer	字节地址
布尔	WMTBoolean	Char

WFM 的对象(图 1)用 C 语言中的结构实现。以下仅举流程实例的定义,其它对象详见《WAPI 规范》。

```
//流程实例的定义
typedef struct
{
//产生该实例的流程名称;
WMTText process_name[STRING_SIZE];
//流程实例的标志符;
WMTID proc_inst_id;
//产生该实例的流程标志符;
WMTID proc_def_id;
//流程实例的运行状态;
WMTProcInstState state;
//流程实例执行的优先级;
WMTInt32 priority;
}WMTProcInst;
//指向流程实例对象的指针;
typedef WMTProcInst * WMTProcInst;
```

与查询 WFM 对象相关的数据结构是过滤器和查询句柄。过滤器(WMTFilter 结构)用来说明查询的条件:attribute_name + comparison + filterstring。查询句柄(WMTQueryHandle 结构)为查询结果提供上下文,以区别同一 workflow 引擎中的不同的查询。

```
//过滤器;
typedef struct
{
//过滤器的类型,1-255为 WFMC 规范的保留值,其值规定了查询的特殊意义,比如某个参加者的工作列表;
WMTInt32 filter_type;
//查询条件中的属性名称
WMTText attribute_name[STRING_SIZE];
//比较运算符<,>,……
WMTUInt32 comparison;
//查询条件,可用*、?等模糊查询
WMTText filter_string;
}WMTFilter;
//查询句柄;
typedef struct
{
WMTUInt32 query_handle;
}WMTQueryHandle;
```

所有的 WAPI 函数的返回值都是一个错误结构:

```
typedef struct
{
WMTInt16 main_code;
WMTInt16 sub_code;
}WMTErrRetType;
```

这个结构是一个有两个 16 位变量的 32 位的整数。main_code 变量返回错误主值,sub_code 变量包含更具体的错误代码。例如,若 main_code = WM-IN-

VALID_PROCESS_INSTANCE(非法的流程实例), 则它的 sub_code 必须包含更具体的错误代码, 指明为什么流程实例(Process Instance)是非法的。WAPI 规范定义了主要的 main_code 代码, 其它的 main_code 代码和所有的 sub_code 代码由 workflow 开发商实现。

3.2 函数定义

接口2和3的成员函数按功能分为八大类, 分别是: WAPI 连接函数; WAPI 流程控制函数; WAPI 活动控制函数; WAPI 流程状态函数; WAPI 活动状态函数; WAPI 工作表函数; WAPI 管理函数; WAPI 调用应用函数。

3.2.1 WAPI 连接函数 WMCConnect/WMDisconnect 函数用来绑定应用程序调用 WAPI 时的工作环境, 定义如下:

```
WMTerrRetType WMCConnect(
    In WMTPConnectInfo pconnect_inf,
    Out WMTPSessionHandle psession),
WMTerrRetType WMDisconnect(
    Out WMTPSessionHandle psession);
```

WMCConnect 有一个输入参数 pconnect_inf。它是 WMTPConnectInfo 指针类型, 指向用户连接信息, 包括用户的标志符、登录口令、用户连接的工作流引擎名称(如果系统中存在多个工作流引擎)。WMCConnect 返回一个参数——会话句柄 psession。会话句柄的值是唯一的, 标志着应用程序与工作流引擎之间存在一个连接。这个连接允许应用程序访问 workflow 数据, 直到 WMDisconnect 被调用。

函数声明中的 In/out 表明参数传递的方向:

In 参数把数据从调用者传向 API 函数

Out 参数把数据从 API 函数传向调用者

需要注意的是, 在 C 语言中, 标志着“out”的参数需要调用者向 API 函数传递指针, 然后 API 函数才能将返回值放在指针所指向的地址空间。

3.2.2 WAPI 流程控制函数 流程控制函数改变一个或多个流程实例的状态, 包括创建、启动、终止等。有些 API 函数会涉及到多个参加者, 因此它们只有管理员才能调用。下面列出了几个函数的定义, 其他函数见《WAPI 规范》。

·已知条件 pproc_def_filter 查询流程定义列表, 返回查询结果的句柄。如果 count_flag 为 true, 则 pcount 返回查询结果的个数。当过滤器为保留值时, 该函数有特殊的意义, 比如得到某个用户可以启动的流程列表。

```
WMTerrRetType WMOpenProcessDefsList(
    In WMTPSessionHandle psession,
    In WMTPFilter pproc_def_filter,
    In WMTPBoolean count_flag,
    out WMTPQueryHandle pquery_handle,
    out WMTPInt32 pcount);
```

·生成一个流程实例。然后可以调用 WMStartProcess 函数启动该流程。

```
WMTerrRetType WMCreateProcessInstance(
    In WMTPSessionHandle psession,
    In WMTPProcDefID pproc_def_id,
    In WMTPText pproc_inst_name,
    out WMTPProcInstID pproc_inst_id);
```

·启动已创建实例的流程。该函数将启动流程的第一个活动并创建活动实例。

```
WMTerrRetType WMStartProcess(
    In WMTPSessionHandle psession,
    In WMTPProcInstID pproc_inst_id);
.....
```

3.2.3 WAPI 活动控制函数 (与 3.2.2 类似, 略)

3.2.4 WAPI 流程状态函数 流程状态函数用来显示与一个 workflow 参与者或应用程序相关的某一状态的工作。一个常规用户或者管理员可以通过状态查询来察看工作的进展情况。流程状态函数实现了全局工作的查询和单个流程实例的查询:

1	与该流程相关的所有流程实例	WM (Open + Fetch + Close) PrOcessInstancesList
2	单个流程实例	WMGetProcessInstance

3.2.5 WAPI 活动状态函数 (与 3.2.4 类似, 略)

3.2.6 WAPI 工作列表函数 工作流参加者用工作列表函数得到被指定的工作。WFM 模型已经描述过, 流程由一定顺序的活动组成, 活动关联着应用, 应用关联着工作流参加者。根据 WFM 产品实现的不同, 一个参加者在某一时刻可能被指定了一项或多项工作。指定给一个参加者的一项工作被称为“工作项(work item)”, 而指定给一个参加者的所有的工作项被称为参加者的“工作列表(worklist)”。因此, 工作列表定义为: 用预定义的查询在工作项空间查询的结果的集合。

```
WMOpenWorkList //打开工作列表
WMFetchWorkItem //取工作列表中的下一个工作项
WMCloseWorkList //关闭工作列表
WMGetWorkItem //返回参加者选定的工作项
WMCompleteWorkItem //告诉工作流引擎工作项已经完成
WMReassignWorkItem //把工作项转给他人
.....
```

3.2.7 WAPI 管理函数 管理函数集包含了一套 workflow 系统管理和维护的超级用户的函数。这些函数可以异常终止 workflow 实例, 改变任意 workflow 实例或活动实例的状态和属性。

```
WMChangeProcessInstancesState //改变任意流程实例的状态
WMChangeActivityInstancesState //改变任意活动实例的状态
WMTerminateProcessInstances //正常终止任意的流程实例
WMAssignProcessInstancesAttribute //改变任意流程实例的属性
```

工程产品

CAD

信息集成

任务元

(18)

计算机科学2000Vol. 27No. 11

工程设计

基于任务元的工程产品 CAD 信息对象集成模型

CAD Info Object Integration in Engineering Product Based on Task Cell

67-70

储 备¹ 武俊峰² 蔡 青¹

TB21

TP391.72

(西北工业大学 CAD/CAM 研究中心 西安710072)¹(西安市市政设计研究院 西安710068)²

Abstract It is necessary for product info to be communicated, exchanged and shared in engineering design (ED). So it is quite necessary to study collaborative product info. According to feature in ED, the paper presents CAD info object integration in engineering product, with an idea of "task cell", definition of drawing cell object and integration of collaborative links. The experiment in engineering validates the rationality and validity of the ideas.

Keywords Task cell, Drawing cell object, Integration of collaborative links, Product info model

工程设计是工程技术人员根据约束条件及有关规范,利用自身经验、专业知识及一系列分析、综合、评价得到满足特定要求的设计结果所进行的一种智能活动,它是一个十分复杂的过程,一般要经历方案设计、初步设计和施工图设计三个阶段,包括设计构思、结构计算、设计细化、施工图绘制、工程概预算等,这些活动需要各类产品信息的通讯、交换和共享,因此研究协同产品信息模型十分必要。本文针对工程设计的特点,提出并建立了工程产品 CAD 信息对象集成模型。最后

工程实例验证了该模型的合理性与有效性。

1 工程设计的特点

从工程的设计过程不难看出,工程设计并不是一个纯逻辑的推理过程,而是依靠设计经验与领域知识,综合运用抽象思维及形象思维的动态求解活动。工程设计有如下特点^[1,2]:

·病态性,工程设计问题往往无法建立完整的数字模型,相关问题的初始、目标状态在开始时是一些非公

```
WMAssignActivityInstancesAttribute
//改变任意活动实例的属性
WMAbortProcessInstances
//异常终止任意的流程实例
```

3.2.8 调用应用函数 WAPI 管理函数提供关于工具代理的服务:激活并控制与工作项关联的应用。

```
WMTAConnect&WMTADisconnect
```

```
//连接和断开工具代理
```

```
WMTAInvokeApplication
```

```
//让工具代理启动或调用应用程序
```

```
WMTARequestAppStatus
```

```
// workflow 引擎查询激活的应用程序的状态
```

```
WMTATerminateApp
```

```
//让工具代理终止激活的应用程序
```

4. 展望

以上介绍了《 workflow 管理应用编程接口规范》。自从1995年 workflow 管理联盟颁布版本1.0以来,作为一系列 workflow 管理规范的核心,该规范发展迅速,先后增加

了接口的对象绑定、数据统计等说明。为了进一步完善 workflow 管理标准, workflow 管理联盟还将在以后的版本中增加以下内容:

·引入同步点处理(Synchpoint processing)增加可恢复性(recoverability)。

·增强安全机制。

·增加共享数据的锁机制。

·保证 workflow 数据的完整性。

参 考 文 献

- 1 Workflow Management Coalition—Workflow Management Application Programming Interface (Interface 2&3) Specification [Document Number WFMC-TC-1009] July-98
- 2 Workflow Management Coalition—The Workflow Reference Model [Document Number TC00-1003] Jan-95
- 3 Workflow Management Coalition—Terminology & Glossary [Document Number WFMC-TC-1011] June-96