

面向企业应用集成的 workflow 引擎实现^{*}

余彬 高鹏 糜宏斌 孙玉芳

(中国科学院软件研究所 北京100080)

摘要 workflow 技术作为一种实现企业过程集成的有效手段,越来越受到重视。本文介绍了一种轻型的面对企业应用集成的 workflow 引擎原型系统,并且引入了对 Script 语言的解释执行来增强 workflow 模型描述能力,加入了动态数据管理来引擎的集成性能。

关键词 workflow, workflow 引擎, Script, 动态数据管理

Implementation of Workflow Engine Oriented Enterprise Information Integration

SHE Bin GAO Peng MI Hong-Bin Sun Yu-Fang

(Institute of Software, Chinese Academy of Science, Beijing 100080)

Abstract Workflow technology is an effective way to implement process integration in enterprises. It is focused much attention and studied widely now. In this paper, a prototype of workflow engine which is oriented enterprise information integration is discussed. The complication of workflow model is increased by explaining and executing a Script language in this prototype. Dynamic data management enlarges the integration of the engine.

Keywords Workflow, Workflow engine, Script, Dynamic data management

1 引言

随着企业信息化、网络化的逐步深入,企业内部的网络基础设施与工作组计算环境日益完善,这为 workflow 技术在企业中的应用、实施提供了可能的条件。本文作者以 workflow 管理联盟给出的 workflow 系统的通用框架—— workflow 参考模型为基础,实现了一个 workflow 引擎原型系统。此原型 workflow 引擎的特点在于:加入了自定义 Script 语言的解释执行功能,用这种 Script 语言协助描述流程,可以极大地增强了流程的多样性和复杂性;灵活的动态数据管理,使本引擎更好地集成到企业应用中去。

2 workflow 基本概念

根据 workflow 管理联盟给出的 workflow 定义^[1], workflow 是一类能够完全或者部分自动执行的经营过程,它根据一系列过程规则、文档、信息或任务能够在不同的执行者之间进行传递和执行。因此, workflow 主要是用来描述经营过程的,可以看成是企业中具体经营过程的抽象或图示化的表示。正如文[2]所述, workflow 中最基本的元素是活动和活动之间的连接关系。另外企业的经营过程还要涉及参与操作的人员、组织、所操作的数据,使用了哪些计算机应用程序等,在 workflow 模型中通过定义活动的角色和组织单元来描述企业的经营活动是由谁完成,通过定义 workflow 应用程序来说明采用什么手段完成经营活动。

3 workflow 元模型

所谓元模型是指描述模型的模型, workflow 管理联盟的 workflow 元模型^[3]描述了 workflow 模型的内在联系,说明了 workflow 模型内部包含的各个对象、对象之间的关系和对象的属性。如

图1所示。

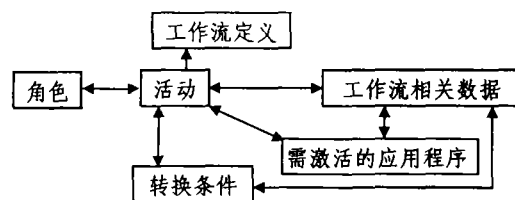


图1

该模型包含了以下几个基本实体:

- (1) workflow 定义(过程模型):反映企业中一个经营过程的目的。一般包含诸如名称、版本号、过程启动和中止条件、系统安全监控和控制信息等基本属性。
- (2) 活动:主要属性有活动名称、活动类型、活动的前后条件、调度约束参数等。活动对应于企业经营过程中的任务。
- (3) 转换条件:主要负责为过程实例的推进提供导航依据,主要参数包括 workflow 过程条件、执行条件和通知条件。转换条件对应于企业经营过程中的业务规则。
- (4) workflow 相关数据:主要属性包括数据名称、数据类型和数据值等,是 workflow 引擎执行任务推进的依据。
- (5) 角色:主要属性包括角色的名称、组织实体、角色的能力等,它决定参与某个活动的人员或组织单元。
- (6) 需激活的应用程序:主要属性包括应用程序的类型、名称、路径及运行参数等。应用主要描述用于完成企业经营过程所采用的工具或手段。

4 workflow 模型的设计类图

根据 workflow 元模型,设计出 workflow 模型的类图如图2所示。

^{*})本文得到国家自然科学基金重点项目(No. 19831020)的资助。

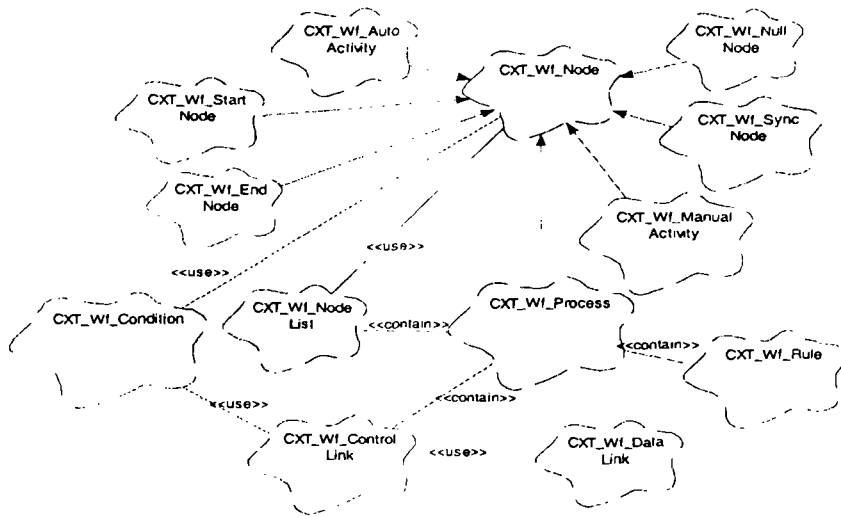


图2

各个类与工作流元模型之间的关系为：

(1)类 CXT_Wf_Node 对应活动，活动是组成业务流程的最基本单元，它包括以下四部分：输入、输出、角色和约束。

a) 输入是保证活动开始的物质条件，由类 CXT_DataElement 实例化的输入容器来表示各个输入数据。

b) 输出是活动的结果，同样是由类 CXT_DataElement 实例化的输出容器来表示各个结果数据。

c) 角色描述的是活动的组织形式。指明活动的执行者、监督者。

d) 约束是指活动执行过程中要满足的条件，包括起始条件、结束条件等，类 CXT_Wf_Condition 对应各种条件。

在统一的活动描述框架下，我们可以定义多种不同类型的活动，如需依靠人员手工或启动应用的方式完成的手工业活动，以及由工作流引擎直接启动应用的自动型活动。还有为了正确理解流程逻辑程序而定义的标志性节点：无前驱的 CXT_Wf_StartNode 和无后继的 CXT_Wf_EndNode 分别标示经营过程的开始和结束。同步节点 CXT_Wf_SyncNode 和空节点 CXT_Wf_NullNode 是为了清晰明确地描述工作流模型的逻辑关系而增加的新节点。在表示“与”关系时，同步节点在执行时将判断它的所有输入连接弧是否已经全部发生转移；空节点只是辅助表示复杂的逻辑关系，没有任何执行过程。

(2)类 CXT_Wf_Process 对应经营过程，由多个活动节点和控制连接弧构成。同时，它又是从 CXT_Wf_Node 类继承而来，本身也可以被称之为一个活动。这样的处理大大增强了模型的表达能力。使模型具有了层次化的概念，并支持自顶向下的建模。类 CXT_Wf_Rule 对应的是在此经营活动中应用的业务规则。

(3)类 CXT_Wf_Control 对应控制连接弧，表达了不同节点元素之间的逻辑顺序关系，它从前驱节点指向后继节点，体现了经营过程的演进。控制连接弧发生转移是有条件的，此转移条件也是由类 CXT_Wf_Condition 来对应的。通常在发生条件转移时，会需要数据流的处理，类 CXT_Wf_DataLink 对应数据处理。

5 工作流的导航

建立起工作流的模型后，企业的经营过程对应着建立起

来的过程流图，而经营过程如何进行则对应着在过程流程图中如何导航。工作流引擎的主要任务就是根据流程图调度经营过程中各个活动的执行。下面给出本文所设计的工作流引擎的导航算法：

算法 flowNavigating

输入：过程名 pname，过程输入参数 input，上下文 context
返回：流程状态

算法流程：

```

初始化，实例化过程模型，所有活动的状态设置为 INITIAL；
启动事务 TransactionManager.startTransaction();
事件(START, FINISHED-OK)加入 eventQueue;
loopEnd=FALSE;
while(true)
{
    从 eventQueue 移去第一个事件→aEvent;
    switch(aEvent)
    {
        case(A, FINISHED-OK);
            A 加入 completedActivityQueue;
            计算 A 的后继可运行节点集合→M
            if M≠{END}
            {
                flowState=FINISHED;
                loopEnd=true;
                break;
            }
            for(M 中的每个活动 a)
            {
                a 的状态←EXECUTABLE;
                CALL a 的实现;
            }
            break
        case(A, FINISHED-FAIL)
            break;
    }
    //end switch
    if (loopEnd)
        break;
}
//end while
如果流程正常结束，确认事务；
如果流程异常结束，做事务回退或事务补偿；
    
```

6 Script 语言的应用

在 Linux 平台上，我们应用 YACC 和 LEX 工具实现了对 Script 语言——CL 语言的编译和解释功能，用于工作流模型的描述和执行中，极大地增强了流程的多样性和复杂性。LEX 和 YACC 配合使用的情况参见文[4]。

CL 语言经 LEX 的词法分析和 YACC 的语法分析后，生成中间码，然后由类 VM 实现的虚拟堆栈机器解释执行。在此机制下，CL 语言的功能有定义变量和给变量赋值、表达式计算、内部函数的调用、对象方法的调用。

(下转第92页)

O(1),FIFO,LRU 的时间复杂度也为 O(1),且空间复杂度也相同,可见我们提出的算法复杂性没有增加。

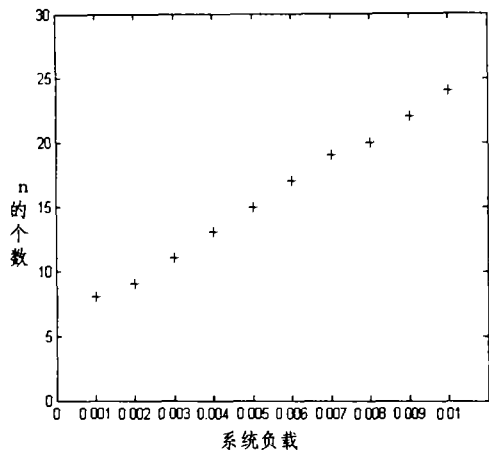


图6 系统负载与 n 的个数的关系

结论 证书访问处理是实现 Internet 公钥基础设施中安全认证和授权的重要一环。而证书服务器中的现有方法并没有解决好 Cache 的置换问题,本文提出了一种新的方法,该方法能有效地缩短网络对用户的实时性响应的的时间,最大程度也节约了系统资源,改变了以前被动的替换策略,使替换更加合理迅速。

参考文献

1 Housley R, Ford W, Polk W, et al. Internet X. 509 Public Key

Infrastructure Certificate and CRL Profile. RFC2459,1999-01
 2 Aggarwal C, et al. Caching on the world wide web [J]. IEEE Transaction on knowledge and data engineering, 1999. 11
 3 Nelson N. Accelerated life testing-step-stress models and data analysis[J]. IEEE Transactions of Reliability, 1980, R-29(2)
 4 Nishida S. Failure Analysis in Engineering Applications [M]. Oxford: Butterworth-Heinemann Ltd. 1992. 22~2
 5 Zhimei J. An adaptive network prefetch scheme [A]. In: IEEE Intl. Conf. on Communications [C] Part 1, 1997. 8~12
 6 Hyokyung B, Kern K, Sam H N, Sang L. Efficient Replacement of Nonuniform Objects In Web Caches. IEEE Journal of Computer Science, 2002, 6: 65~73
 7 Tanenbaum A S. Computer Network [M]. Third Edition Prentice Hall International, Inc
 8 Frankel Y, Gemmell P, MacKenzie P D, et al. Optimal-resilience proactive public-key cryptosystems. In: IEEE symposium on Foundations of Computer Science, 1997. 384~393
 9 Nussbaumer J P, et al. Networking requirement for interactive video on demand. IEEE Journal of Selected Areas in Communication, 1995, 13(5): 779~787
 10 Zipf G K. Human Behavior and the Principle of Least Effort. Addison-Wesley, 1949
 11 Wolf J, et al. Disk load balancing for Video-on-Demand systems. Multimedia system, 1997, 5(6): 358~370
 12 龚诗松, 王玲玲. 可靠性统计[M]. 上海: 华东师范大学出版社, 1984
 13 唐三平, 等. 一种处理服务器连接的控制模型. 计算机研究与发展, 2002, 39(6): 668~671
 14 李勇, 等. 大规模 VOD 及其 Cache 机制. 计算机工程与科学, 1999, 21(5): 52~55
 15 Access Control Library (ACL) Library File, Getronics Government Solutions. http://www.getronicsgov.com/hot/acl-home.htm

(上接第60页)

CL 语言在工作流模型中有以下两种应用:

(1) 将 CL 语言的逻辑表达式计算功能应用于起始条件、中止条件、转移条件的逻辑值计算中。类 CXT_Wf_Condition 中的条件可以用 CL 语言写成的复杂逻辑表达式, 当需要计算此表达式的值时只需调用 CL 语言的解释执行程序。

(2) CL 语言能对变量进行定义和赋值, 并且能对对象的方法进行调用, 这样用 CL 语言写成的数据处理程序就能方便地被解释执行, 使得工作流模型中的数据流处理变得更为容易。

由于 CL 语言的应用, 工作流模型可以处理非常复杂的各种逻辑条件以及数据映射, 变得更为灵活。

7 数据模型管理

数据模型实现了对元数据的管理与动态的数据总线, 应用实例在运行状态下可以依据事务的处理状态动态拼装数据。我们采用了 Composite 模式^[5]来设计数据模型, 类图如图 3 所示。

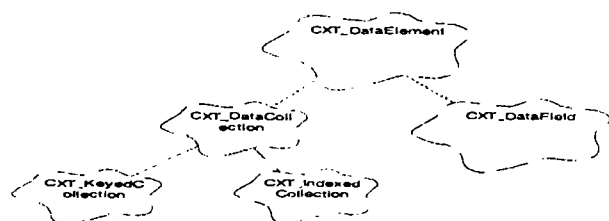


图3

1) CXT_DataElement 是最基本的基类, 简单数据类

CXT_DataField 和复杂数据类 CXT_DataCollection 都由它派出来。它提供了数据操作的统一接口。

2) 类 CXT_DataField 代表最基本类型的数据, 如整型、浮点型、字符串型等。

3) 类 CXT_DataCollection 代表复合型的数据, 可衍生出 CXT_KeyedCollection 和 CXT_IndexedCollection 两类复合型数据。

4) 类 CXT_KeyedCollection 由有序的 CXT_DataElement 集合组成, 各个 CXT_DataElement 可以是不同类型的数据, 类似于 C 语言中的 struct 结构。

5) 类 CXT_IndexedCollection 是由相同的 CXT_DataElement 复合而成, 相当于 C 语言中的数组, 各个 CXT_DataElement 子项以下标标识。

在已定义的数据字典的基础上, 工作流模型中的活动可以用这些表示数据的类动态地拼装出所需要的数据结构并赋予相应的数据, 极大地增加了数据管理的灵活性和可用性。

结论 本文着重讨论了面对企业应用集成的工作流引擎的设计和实现, 以及如何通过对 Script 语言的解释执行和灵活的数据管理来提高工作流模型的复杂程度和表述能力。对工作流模型的支持标准的支持和工作流引擎执行性能的提高是需要进一步研究的问题。

参考文献

1 Workflow Management Coalition. The Workflow Reference Model. WfMC TC00-1003, 1994
 2 Leymann F, Roller D. Production Workflow Concepts and Techniques. ISBN 0-13-021753-0, 2000
 3 Workflow Management Coalition. Workflow Management Coalition Specification: Terminology & Glossary. WfMC TC1011, 1944
 4 吕映芝, 张素琴, 蒋维杜. 编译原理. ISBN 7-302-02732-3, 1998
 5 Booch G. Design Patterns