

图1中集成框架是为了系统容纳多种协同应用信息系统,全面支持协同环境下的工作,对整个 workflow 领域的普遍问题进行统一管理而引入的。它负责和各种通用领域信息系统、外部应用进行协调,提供如群体协同、一致性维护等公共服务;数据接口和转换服务,如对 DBMS 的支持。本文主要研究的工作流应用信息系统版本管理就是在集成框架中实现的,对容纳的多种协同应用信息系统进行统一的版本管理,使业务过程的运行在应用层次上保持一致。

2 基于信息系统的工作流应用信息系统版本一致性管理

2.1 工作流应用信息系统版本一致性管理需求

同正规的软件开发过程一样,协同工作环境下多个协同应用信息系统自身也在不断进行升级和版本的更新。在整个 workflow 实例运行期间,workflow 引擎必须调度正确版本的协同应用信息系统才能有效地完成业务过程。如果没有这样一个维护机制,不可能建立起真正意义上的分布式大型 workflow 管理系统。我们所建立的是基于企业内外部信息系统的工作流管理平台,其中不光是 workflow 管理系统自身组件,更多的是现有的企业信息系统和应用信息系统,更加需要在漫长的工作流业务过程运行期间对涉及参与的企业信息系统和应用信息系统的版本进行统一管理。管理这样一个版本群体实际是非常复杂的,不同于单个应用信息系统的版本管理,需要形式化的理论支持。

2.2 工作流应用信息系统版本数据模型

目前有众多定义 workflow 的模型和方法。本文选择其中简单直观的有向图定义 workflow,将重点放在 workflow 中协同工作的应用信息系统版本信息上,引入应用信息系统版本数据模型。

定义1 工作流是一个有向无回路图 $WF(N, E)$ 。节点 $x \in N$ 是 workflow 中一个活动。边 $(x, y) \in E$ 表示 workflow 由活动 x 到活动 y 。

下面给出我们研究的面向软件开发 workflow 管理的一个简单例子来进行说明,这个例子后面的定义也会用到。

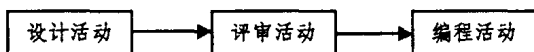


图2 工作流定义示例

定义2 工作流 $WF(N, E)$ 的应用信息系统版本数据模型是个二元组 $MAV-WF(V, R)$ 。 V 是 workflow 涉及的多应用信息系统版本的集合, $x_i \in V$ 是版本为 i 的应用信息系统 x 。 R 是应用信息系统版本之间的关系, $(x_i, y_j) \in R, x_i, y_j \in V$, 表示 workflow 活动由版本为 i 的应用信息系统 x 到版本为 j 的应用信息系统 y 或两个应用信息系统有联系。

注意,我们这里是假设 workflow 活动是原子活动的,如果活动是复杂嵌套活动,那么这个活动自身也可定义应用信息系统版本数据模型,图中的箭头还可以表示活动调用的应用信息系统之间有联系。

假定在上面所定义的工作流中,设计阶段使用应用信息系统 a ,评审需要应用信息系统 b ,编程需要应用信息系统 c ,它的工作流应用信息系统版本数据模型示例如图3所示。

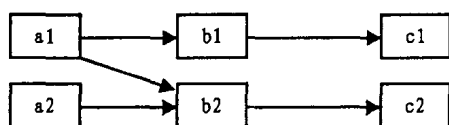


图3 工作流多应用版本数据模型

定义3 工作流 $WF(N, E)$ 的应用信息系统运行实例是一个有向无回路图。节点是 workflow 实例所调用的某一版本应用信息系统,边表示 workflow 实例运行的路线。

图4的工作流应用信息系统运行实例表示了设计活动调用版本为1的应用信息系统 a ,评审活动调用版本为1的应用信息系统 b ,编程活动调用了版本为1的应用信息系统 c 。

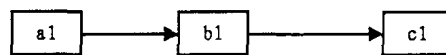


图4 工作流应用信息系统运行实例

定义4 工作流应用信息系统版本协同实例是 workflow $WF(N, E)$ 的应用信息系统版本数据模型的一个连通路径的子集。它代表了一个可协同的工作流应用信息系统实例,即如果 workflow 实例是这样运行的,那么所参与的不同版本的应用信息系统就是可协同的。

图5所示就是定义2应用信息系统版本数据模型例子的两个 workflow 应用信息系统版本协同实例。

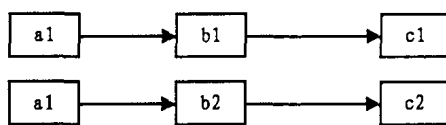


图5 工作流应用信息系统协同实例

定义5 工作流 $WF(N, E)$ 应用信息系统版本是一致的,即 workflow $WF(N, E)$ 所有运行实例所涉及的应用信息系统版本是协同的,如果每个应用信息系统版本属于同一个 workflow 应用信息系统版本协同实例。

2.3 协同环境下基于信息系统的工作流应用信息系统版本一致性检测机制

此机制可以在基于信息系统 workflow 平台的集成框架中实现,用于管理维护多信息系统应用信息系统的当前版本和 workflow 实例运行时的应用信息系统版本,并检测 workflow 运行中应用信息系统版本的一致性。

定理1 基于信息系统 workflow 应用信息系统版本是一致的当且仅当 workflow 应用信息系统运行实例是一个 workflow 应用信息系统版本协同实例的子集。

证明: workflow 应用信息系统版本是一致的,由定义5,组成这些应用信息系统的版本是协同的,必存在一个 workflow 应用信息系统版本协同实例,使这些应用信息系统版本都在其中。

假设 workflow 应用信息系统版本协同实例就是一个 workflow 应用信息系统运行实例,那么由这个 workflow 应用信息系统版本协同实例组成的应用信息系统的版本是一致的。证毕。

2.3.1 工作流应用信息系统版本一致性检测算法

输入:假设对 workflow $WF(N, E)$ 一个应用信息系统运行实例 $RIwf$ 所涉及的应用信息系统版本集合为 Ari 。

输出: $RIwf$ 是应用信息系统版本一致的;否则,输出应用信息系统版本协同实例集合。

1. 对每一个 workflow $WF(N, E)$ 构成的应用信息系统版本协同实例 $CIwf$,其所涉及的应用信息系统版本集合为 Aci ,判断是否 $Ari \subset Aci$;

2. 如果成立,则此 workflow 应用信息系统运行实例版本是一致的;

3. 否则,输出以 workflow 应用信息系统运行实例 $RIwf$ 起

(下转第100页)

点 A 获得动作自主权以后,需要考察本节点任务的执行条件是否满足(比如资源占用情况、角色情况等等)并将当前状态通过消息包发送给 workflow 引擎。

3.2.2 实例运行 系统的主体操作对象包括以下两大类:

1)引擎类:

```
class WfEngine {
    WfProcess createProcessInstance(); //创建过程实例
    List processList; //过程实例表
    ...
}
```

该类主要负责创建执行对象,包括过程对象、活动对象、任务对象,并管理它们的运行,完成相应的任务。这些对象都继承自同一个抽象类,属于完成特定业务逻辑的可执行对象,都支持暂停、继续执行、中止等操作。

2)执行对象类:

```
class WfProcessImpl {
    void start(); //开始/运行过程实例
    boolean executeIncomingTransitions(); //开始运行活动实例前
    //对传入的转移条件进行判断
    void startActivity(); //启动活动实例
    void completeActivity(); //活动实例运行结束
    void executeOutgoingTransitions(); //活动实例运行结束
    void executeSplit(); //调度下个活动的运行
    ...
}
```

```
class WfActivityImpl {
    void createAssignments(); //创建任务
    void complete(); //将活动实例执行完毕
    void start(); //开始执行活动
    ...
}
```

```
class WfAssignmentImpl {
    void completed(); //任务完成
    void accept(); //接受任务
    ...
}
```

通过引擎类,系统可生成过程、活动、任务项,过程对象通过 startActivity 方法开始一个新的活动,并通过 completeActivity 方法在活动结束后进行一些处理工作。活动对象通过 createAssignments 方法可创建一个新的任务项并由 start 方

法启动自己的运行。任务类则通过 accept 方法使活动执行者接受该任务,并开始执行相应的活动。

当一个新的刺绣订单到来时,workflow 引擎为此生成一个刺绣过程实例,并生成一个订单审查任务项,通知相关负责人审查订单内容,根据审查结果生成相应的后继任务。如果订单提供了编针图样,则进行来版识别活动。如果提供了花样图像,则对图像进行处理,随后引擎将图像送入刺绣系统进行编针,并将编针结果送入质检中心,检验编针效果,决定是否重新进行编针、上机刺绣或是继续进行审查。以上即是 workflow 引擎对一个刺绣过程的管理。

结束语 目前,随着电脑刺绣业的快速发展,国内外电脑刺绣 CAD 系统迫切需要在网络环境下进行 workflow 协同工作,提高自动化程序以及通过网络环境进行协同工作等等,为提高刺绣生产的自动化和效率,促进刺绣产业更进一步的发展,以便满足刺绣行业的要求。我们采用了 workflow 技术,研制了智能刺绣 CAD 系统,采用 VC++ 和 JAVA 编程实现,目前系统已经在电脑刺绣厂投入试用,大大提高了生产效率,其效果很好。

参考文献

- 1 王明强. 计算机辅助设计技术[M]. 北京: 科学出版社, 2002. 242~264
- 2 Reinhard W, Schweitzer J, Volkson G. CSCW tools: concepts and architectures [J]. Compute, 1994, 27 (5): 37~46
- 3 Lindeman D D, Moore B. PDM: An Enabling Technology for Integrated Product Development [J]. In: Proc. Annual Reliability and Maintainability Symposium, 1994. 320~326
- 4 Miller E. PDM in the forefront [J]. Computer-Aided Engineering, 1998, (3): 30~42
- 5 Hollingsworth D. Workflow Management Coalition The Workflow Reference Model. <http://www.wfmc.org/standards/docs/tc003v11.pdf>, 1995
- 6 范玉顺. workflow 管理技术基础[M]. 北京: 清华大学出版社, 2001. 150~153

(上接第96页)

始节点的应用信息系统版本为起点的工作流应用信息系统版本协同实例 CIwf 的集合。

在我们所进行的基于信息系统 workflow 平台研究的课题和面向软件开发的工作流管理系统的开发实践中这些机制和算法已经进行了应用和验证。

2.4 示例说明

下面我们举例说明以上模型和算法的实际运用。在系统中,我们除了记录每个 workflow 实例的活动外,还记录着每个活动实例所调用应用信息系统的版本信息。用这些信息来构造定义2中提到的 workflow 应用信息系统版本数据模型。假设 workflow 应用信息系统运行实例如定义3所示的例子,即活动 A 调用版本为1的应用信息系统 a,活动 B 调用版本为1的应用信息系统 b,活动 C 调用了版本为1的应用信息系统 c。可以看出它是定义2所示的版本数据模型的一个连通路程,所以这样运行的 workflow,其应用信息系统版本是协同一致的。

再假设 workflow 的一个应用信息系统运行实例,它的活动 A 调用版本为2的应用信息系统 a,活动 B 调用版本为2的应用信息系统 b,活动 C 调用了版本为1的应用信息系统 c,可以用图6表示。但是在 workflow 应用信息系统版本数据模型中,找不到这样一个连通路程,说明这样运行的 workflow 应用信息系统实例,其应用信息系统版本是不一致的,需要调整。利

用上面提到的算法提供有效的工作流应用信息系统版本协同实例,供活动选择正确版本的应用信息系统运行调用。

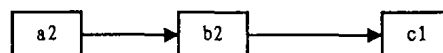


图6 示例说明

结论 协同工作环境下基于信息系统的工作流中所涉及的应用信息系统的版本管理是十分必要和重要的。我们在提出基于信息系统 workflow 平台的概念前提下,给出了 workflow 应用信息系统版本数据模型,并基于此提供了 workflow 应用信息系统版本一致性检测机制和具体算法,不但可以作为协同环境下 workflow 应用信息系统版本管理的理论依据,还可应用于大型分布式复杂信息系统的版本管理上。

参考文献

- 1 Xu H, Furukawa T, Shi Y. Supporting Cooperative Work Based on the Semantics of Workflows. IEEE Computer Society Press Database Applications in Non-Traditional Environments, 2000. 366~369
- 2 Workflow Management Coalition. Workflow management coalition terminology and glossary: [Technical Report, WfMC TC-1011]. Workflow Management Coalition, 1999
- 3 Appina I, Aroubar Shalici U, et al. Correctness of workflow in the presence of concurrency. In: Proc. of the Next Generation Information Technologies and Systems Conf. (NGITS'97). Israel, 1997