

智能化刺绣 CAD 系统中的 workflow 技术研究^{*}

李俊 张华 王崇骏 陈世福

(南京大学计算机软件新技术国家重点实验室 南京210093)

摘要 基于智能化刺绣 CAD 系统向网络化和规模化方向发展的趋势,本文设计并实现了一个面向刺绣 CAD 的工作流框架,在该框架结构下,刺绣任务的各个执行环节被分解为粒度不同的、独立的节点任务,用户根据实际的工作环境定义这些节点任务的执行序列, workflow 引擎根据用户定义控制任务的执行。本文主要介绍了面向刺绣 CAD 系统的工作流结构框架和其中的关键技术,包括过程定义、通信和调度。文中给出了一个基于该框架的系统实例,实际应用表明该框架是可行和有效的。

关键词 智能化刺绣 CAD, 工作流, 过程定义

Research on Workflow in Intelligent Embroidery Compute Aided Design System

LI Jun ZHANG Hua WANG Chong-Jun CHEN Shi-Fu

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract A work flow architecture for computer aided embroidery is designed because the intelligent one is to be developed for network and large-scale integration. In the frame, the embroidery task is divided as plenty of node tasks which are self-governed and different-granularity, then users can define the sequence of these node task according to the needs. The work flow engineering could attemper the process during the running time. This paper describes the frame in briefly and the key technology including process definition, communication and control are informed in detail. A sample based on this frame is given in the paper and the experimental results indicate that this frame is an effective and feasible one.

Keywords Intelligent computer aided embroidery design, Workflow, Process definition

1 引言

刺绣 CAD 系统从20世纪70年代起就得到了软件业和刺绣业的广泛重视,已经研发了多种规格的、适合不同场合的电脑刺绣编程系统,如日本的 Tajima、德国的 ZSK、美国的 Wilcom、中国的天木等等。伴随着制造工艺技术和计算机技术的迅猛发展,人们对刺绣 CAD 系统的要求越来越高,人们希望面向刺绣业的 CAD 系统具备智能化、网络化、规模化的方向发展^[1]。特别是在大规模生产的流水线管理,人们在这方面的要求最终体现在如何快速、高效地完成花样设计作业以及如何将作业进程按业务需求或生产现状进行重组,从20世纪80年代开始基于并行工程提出的并成为研究热点的 CACD (Computer Aided Concurrent Design) 的目标就是通过产品开发过程中的多功能小组的协同作业达到将传统的串行作业流变为并行作业流,从而最终解决企业 P, T, Q, C, S 等问题^[2~4]。

在面向生产的刺绣 CAD 系统中,除了需要考虑并发作业及并发控制外,还需要考虑各个功能小组的通信和协调,考虑到实际生产中的运行环境大多都具有异构 (Heterogeneous)、自治 (Autonomous) 和分布 (Distribute) 的特点,因此,在研究面向刺绣 CAD 系统的并行工程时,很有必要研究基于统一的分布式的模型框架以及在该框架下的并发控制、进程通信和协调,即面向刺绣 CAD 的工作流技术。

工作流的思想最初起源于 JCL (JOB CONTROL LANGUAGE)^[5],目的是让用户定义由一组工作步骤组成的工作流程,全部或部分的由计算机支持或自动处理的业务流程。刺绣 CAD 系统中的 workflow 技术主要包括建模、过程控制及应用工具等三个主要内容。

我们在已完成的智能化刺绣 CAD 系统中采用了 workflow 技术,实现了一个高效的、多功能的智能刺绣 CAD 系统,其实用表明,其效果很好,本文主要介绍了面向刺绣 CAD 系统的工作流结构框架以及相关的核心技术。

2 刺绣 CAD 的工作流模型

工作流模型包含了描述由 workflow 执行服务软件系统执行的过程所需要的所有信息。这些信息包括过程的开始和完成条件、构成过程的活动以及进行活动间导航的规则、用户所需要完成的任务、可能被调用的应用、workflow 引擎的引用关系,以及所有 workflow 相关数据的定义。

针对刺绣 CAD 系统而言,其 workflow 体系结构的主要组成部分包括 workflow 引擎、数据库、过程建模、管理工具、刺绣编程系统以及其他应用工具(如果需要同其他应用程序甚至 workflow 产品进行数据通信或过程调用的话),各个组成部分的关系参见图1所示。

1) workflow 引擎 这是 workflow 系统的核心模块。它将建模工具定义的刺绣过程模型实例化,负责任务的调度和监控,并

^{*} 本文得到江苏省高新技术研究(BG2003026)资助。李俊 硕士生,主要研究领域是 workflow 技术。张华 硕士生,主要研究领域是机器学习、workflow 技术。王崇骏 博士生,主要研究领域是软计算、模式识别。陈世福 博导,主要研究领域是人工智能、数据库技术。

完成任务节点间的导航,同时调用刺绣编程系统完成具体的刺绣工作, workflow引擎模块还提供了与企业其他应用工具进行交互的接口,两者互相协助,共同完成整个刺绣业务流程。

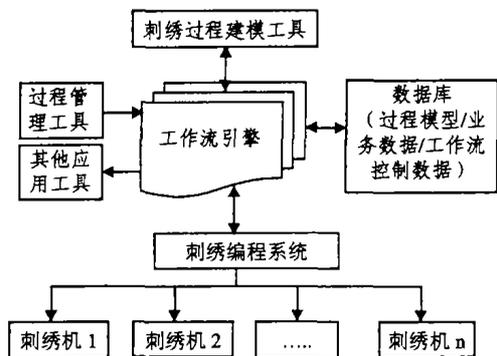


图1 刺绣CAD系统工作流模型

2)数据库 刺绣CAD workflow系统的底层存储模块,用以存放各种刺绣过程模型及其相关的业务数据和工作流控制数据。其中业务数据为过程执行期间所要用到的企业资源,而 workflow控制数据用来控制过程逻辑,完成任务间的导航。每次过程执行期间的各种状态、事件都被作为历史记录存放在数据库中,这些数据为以后设计出更高效的过程模型提供了历史经验。

3)建模 建模是根据刺绣过程的特点确定刺绣任务的分解,根据任务划分粒度的大小,定义刺绣 workflow过程中所包含的具体任务及其执行顺序和干预规则。本系统采用 WfMC (workflow管理联盟)提出的 WPD L (Workflow Process Definition Language)规范作为 workflow过程的存储格式。通过这个规范,一个过程模型就可以在不同的 workflow产品上被解释执行,为不同的 workflow产品间的互相协作打下基础。

4)管理工具 一个全局的提供人机交互的界面,通过该界面,用户可以参与到 workflow过程中来。管理员定义或管理一个刺绣 workflow过程的执行,包括启动、中断、终止一个过程和其中的任务,查看或修改其状态等等。工作人员则用它完成接收和执行分配给自己的任务。

5)刺绣编程系统 具体刺绣任务的控制与完成的主要模块。它与多台刺绣机相连,负责花样编针并对各个刺绣机进行有效的管理以完成相应的刺绣任务。

6)其他应用工具 整个刺绣过程是由企业现有的各种应用工具共同协作来完成的,因此刺绣 workflow系统中必须提供统一的接口来管理和调用这些工具。

3 实现的关键技术

刺绣CAD系统 workflow中的最主要的关键技术是建模(过程定义)、模型的解释和运行(任务调度)。

3.1 过程定义

3.1.1 相关概念 过程定义是设计具体的 workflow实例并输出一个由节点任务组成的执行序列的 workflow实例。本系统过程定义至少要包含如下几个内容:

1)工流类型定义(Workflow Type Definition) workflow过程名、版本号、过程开始、结束条件。

2)活动(Activity) 活动名、活动类型、进入动作和离开动作、活动前提和后继条件。当 workflow运行在分布的环境下时,活动的属性中还应包括执行该活动的 workflow机的位置。

3)转移条件(Transition Condition) workflow引擎在活动

之间进行导航的依据。

4) workflow相关数据(Work Flow Relevant Data) 企业业务运行过程中使用的资源,包括资源名称和具体访问路径,它是 workflow引擎执行任务推进的依据。

5)角色(Role) 每项任务都有一个角色属性,只有具有相同角色属性的用户才有权限操作该任务。角色可能是一个具体的用户名称,或者是某一级别的组织实体。

6)应用程序调用(Program Evoke) 过程定义中必须给出所有在过程运行时将调用到的应用程序的类型和名称、执行参数、本地或者访问路径,以便由 workflow引擎直接调用应用程序完成任务,或者由用户手工启动应用程序完成分配给他的任务。

目前,主要的建模技术有数据流程图(DFD)、角色行为图(RAD)、事件流程链(Event-Process Chain)、Petri网等。其中 Petri网具有极强的模拟能力,有着坚实的理论基础和比较成熟的分析方法,如:可达图、完全性、死锁等特性分析,使得过程模型在运行前被检查,从而避免错误,可利用分析结果发现模型缺点,从而不断优化整个过程模型^[6]。图2即为我们使用 Petri网技术设计的一个简单刺绣业务的过程模型。在建模过程中,我们采用的条件和事件的概念,图2中,圆圈代表条件,方框代表事件。一个事件有多个输入输出,分别代表事件的前提条件和后继条件。

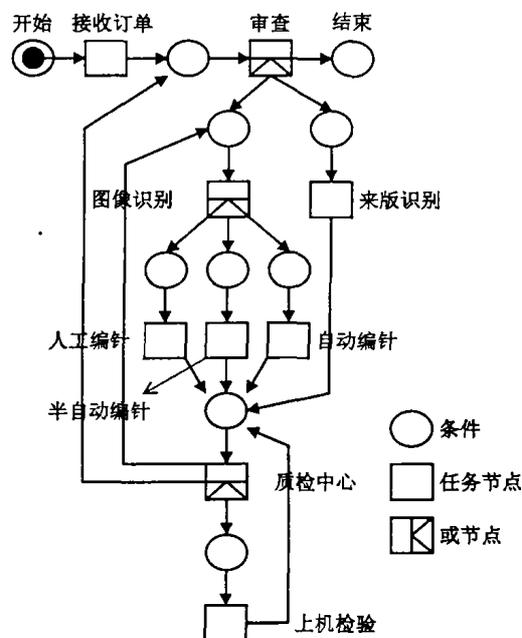


图2 CAD刺绣系统的一个刺绣过程模型

现在市场上很多的工作流产品都有自己的过程定义语言,使得每个过程模型只能为自己的 workflow引擎所解释,缺乏通用性。为此,WfMC提出了 WPD L规范,统一过程模型的表示方式,同时也给出了 XPD L (XML Process Definition Language)的工作流模型的公共交换格式。用 XPD L格式来表示过程定义,则一个过程模型可以运行于多个 workflow系统中,将建模工具与 workflow管理系统独立开来^[5]。

3.1.2 基本数据结构 对应于上述内容,该系统在实现的时候使用 Java语言定义了一系列的数据结构,首先是最基本的 WFElement的数据类,它是该 workflow中元素的最小抽象单位,定义如下:

```
class WFElement {
    String id; //id号
    String name; //名称
}
```

```
String description; //描述
Map extendedAttributes; //扩展属性列表
...
}
```

从这个基本的数据类出发,我们派生出其它的数据对象,其中最关键的是 WorkflowProcess, Activity, Transition, Condition 这四个数据类,具体定义如下:

```
class WorkflowProcess {
    List formalParameters; //形参列表
    List dataFields; //数据
    List participants; //参与者列表
    List applications; //应用列表
    List activities; //活动列表
    List transitions; //转移条件列表
    ...
}
class Activity {
    Implementation implementation; //活动的具体实现
    String performer; //活动执行者
    AutomationMode startMode; //活动启动方式
    AutomationMode finishMode; //活动结束方式
    List transitionRestrictions; //转移条件列表
    ...
}
class Transition {
    Condition condition; //转移条件
    String from; //转移起点
    String to; //转移目标
    TransitionLoopType loopType; //转移类型
    ...
}
class Condition {
    List xpressions; //条件表达式列表
    ConditionType type; //条件类型
    ...
}
```

通过这样的定义,我们就可以把一个任务节点描述清楚,系统根据用户在建模时对该节点(对象)的定义(属性和动作等),能将过程定义实例化为可执行的执行序列。

为了满足复杂的刺绣业务过程,需要在过程模型中表示出串行、并行、条件选择、循环等多种过程逻辑,本系统中定义了一些结构化的组件来实现这些功能。同时,为了便于用户的操作,我们根据刺绣行业的业务流程的具体情况抽象出一系列的任务节点库,并将其以组件的方式进行封装,用户可以通过从库中拖取组件来完成模型的设计。

3.2 workflow引擎

workflow引擎是刺绣CAD workflow系统中的软件服务器,负责整个刺绣workflow实例的运行并维护实例的执行环境。一般具有解释过程定义、控制过程实例的运行状态、控制workflow实例的执行顺序、维护workflow数据、提供控制、管理和审查功能等等。图3是CAD刺绣系统中workflow引擎的结构。

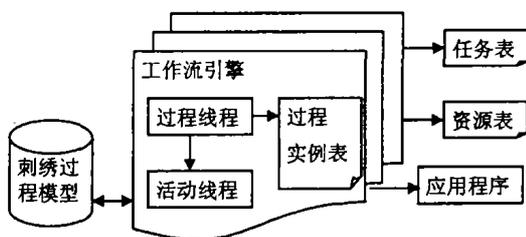


图3 CAD 刺绣系统的工作流引擎结构

开始运行一个刺绣过程模型时,workflow引擎从模型库中读取刺绣过程模型,并根据XPDL公共格式进行解释。workflow引擎维护一张过程实例表,记录过程运行期间的状态等信息,同时为每一个实例化后的刺绣过程模型单独运行一个过程线程,管理这个过程实例的运行,维护其运行状态和导航其中的各个活动。过程线程为过程中的每个活动创建一个活动线程来管理活动的执行,并在过程表中添加活动记录来保存

活动的运行信息。在过程实例化后,workflow引擎将过程中的活动内容整理成记录存入任务表中,包括了活动的具体内容、需用到的工具、现在的运行状态以及当前已经满足的和未满足的开始、结束条件。在过程运行期间,实时的修改任务表中的内容,与过程运行状态保持一致。图4显示了一个过程实例的运行过程。

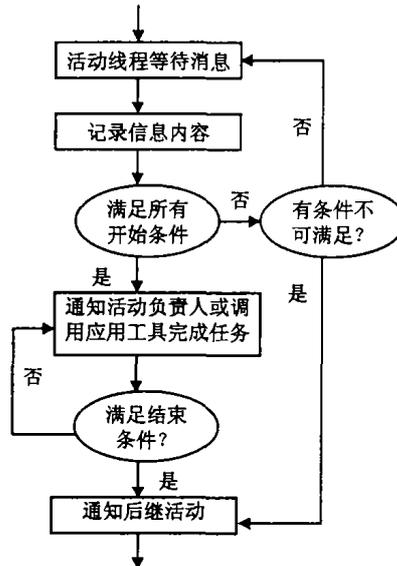


图4 过程实例运行过程

根据一个活动执行过程中传递的开始其他活动的消息,过程活动启动一个活动线程。如果活动的开始条件都已满足,活动线程通知活动执行者或者自动调用应用程序完成任务,如果满足结束条件,则通知过程线程启动下一个活动,否则重新执行活动。如果活动的开始条件已经不可能被满足了,则通知它的后继活动,不必再等待该活动的执行结果。

3.2.1 节点任务的通信

由于在workflow运行时期,workflow引擎负责各个节点任务的通信和调用,因此节点任务之间的通信对于workflow引擎来说都是可控的和可观测的。

一般而言,节点任务的通信方法有主机模型(Host based Model)、共享文件存储模型(Shared File Store Model)、电子邮件模型(Electronic Mail Model)、过程调用或消息传递模型(Procedure Call or Message Passing Model)等多种常用的方式^[7]。在本系统中将主要使用电子邮件模型和消息传递模型。对需要由人工参与的活动,刺绣workflow引擎将根据活动的角色属性给它的职工或者满足该角色属性的所有人发送电子邮件,告知他要完成的任务。同时用户可以使用管理工具,从任务表中提取所有与他有关的任务,查看其执行状态,并控制其执行。

我们定义了两个通信元语用于节点之间的通信。其定义描述如下:

- 1) Send(WfActivity activity, Message package): 当前任务节点向任务节点 activity 发送消息包 package;
- 2) Send(Message package): 任务节点向workflow引擎发送消息包 package;

执行情况是这样,每当一个节点退出的时候就将结果执行情况打包发送到下一个指定的节点,这里包括两种情况:(1)每个节点任务执行完毕以后只将消息发送给workflow引擎,以后具体的操作由workflow引擎执行;(2)每个节点任务都可以看作是一个可以运行的应用程序实例,每个节点执行完以后,它就根据过程定义情况将消息传递给下一个节点 A。此时,节

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 引擎为此生成一个新的刺绣过程实例,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。

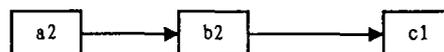


图6 示例说明

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