

工作流系统在离散制造业的应用^①

汤幸江 姚明海 (浙江工业大学 信息工程学院 浙江 杭州 310032)

摘要: 工作流技术通过对企业业务过程建模、业务过程管理和集成,规范企业业务流程,优化和重组企业业务,对促进企业管理规范化和信息化具有重要意义。Windows Workflow Foundation(以下简称 WF)为开发基于工作流的应用程序提供了一个通用的框架,通过对 WF 框架的学习研究,结合在离散制造业的应用,开发了基于工作流的离散制造信息系统,提高了企业的工作效率和经济效益。

关键字: WF; 工作流; 信息系统; 离散制造; 流程

Application of Workflow Management in the Discrete Manufacturing

TANG Xing-Jiang, YAO Ming-Hai

(Information Engineering College, Zhejiang University of Technology, Hangzhou 310032, China)

Abstract: Through the enterprise business process modeling, business process management and integration workflow technology could standardize the business processes of the enterprise, optimize and restruct the business processes of the enterprise. It has of great of significance for enterprise to promotes the standardization of management and information. Windows Workflow Foundation provide a common framework for workflow-based application, through the research and study of WF framework. Combined with the applications in discrete manufacturing, the paper develops a workflow-based enterprise information systems, to improve the efficiency and the economic benefits of enterprises.

Key words: WF; workflow; information system; discrete manufacturing; process

1 引言

工作流技术起源于上个世纪 70 年代中期办公自动化领域的研究工作,通过将工作分解定义良好的任务、角色,按一定的规则和过程来执行这些任务并对它们进行监控来提高办事效率降低生产成本,提高企业生产经营管理水平和企业竞争力^[1]。WF 是微软推出的关于工作流的研究框架,它能快速开发 Windows 平台上基于工作流的应用程序。WF 开发框架能够简化工作流程序的开发,提高开发效率,可以使熟悉业务流程的人员经过少许学习就可以在可视化工具中定义流程和规则,而程序员就可以专注于流程的每个功能模块的实现。针对离散制造业的车间生产流程设计了基于工作流的离散制造业信息集成系统。

2 WF引擎体系结构

WF 是一个工作流应用系统的开发框架,为工作流的创建提供了图形化界面,程序员可以利用 WF 快速开发基于 Windows 平台的工作流系统。程序员只需通过自定义活动以及相关的通讯事件及方法,运用 WF 提供的可视化设计界面设计相关业务流程即可运行程序。

WF 体系结构图如图 1 所示共分四层^[2]: 宿主程序、宿主层、运行时层和工作流模型层。

2.1 宿主程序

宿主程序是一个载体或者是一个调用者,而最终用户在使用工作流管理系统的时候接触的就是宿主程序,工作流引擎则运行在服务器上为宿主程序提供服

^① 基金项目:浙江省重大科技专项优先主题重点项目(2008C11112)

收稿时间:2009-12-11;收到修改稿时间:2010-01-22

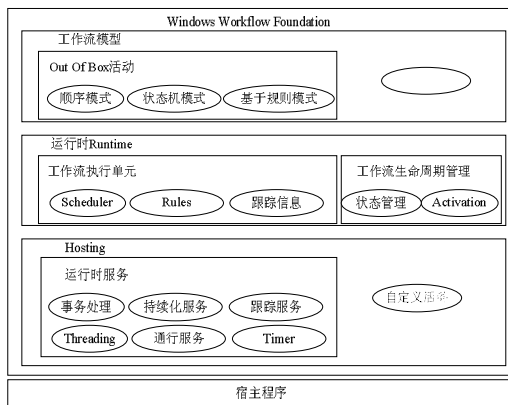


图 1 WF 体系结构图

务并管理工作流。WF 的宿主程序几乎可以是任意的应用程序，比如控制台程序、窗体表单程序、ASP.NET 网站、窗体服务等都可以作为 WF 的宿主程序。

2.2 宿主层

宿主层是在宿主程序层和 WF 的运行层之间提供各种类型的服务。宿主层提供的主要服务有：通讯服务，持久化服务、跟踪服务、定时服务、线程服务与事务处理服务等[3]。

2.3 运行时层

WF 运行时层是核心组件，用于执行业务人员编辑好的工作流(xOML 文件)[4]。是宿主应用程序和工作流实例之间的通道。在 WF 运行时里面，可以加载不同的服务，包括持久化服务等，甚至可以根据不同的需求开发自己编写的服务。在 .NET Framework 3.0 和 3.5 中，有一个专门的类 WorkflowRuntime 类，主要负责管理 WF 运行时、启动和管理工作流实例、管理运行时的服务、处理运行时的事件等。

2.4 workflow 模型层：

这一层是 WF 的应用开发层。它提供了几种不同的工作流模型，有顺序工作流模型、状态机工作流模型、基于规则的工作流模型。同时提供了大量开箱即用的活动，同样可以根据需求自定义活动。

3 基于工作流的离散制造信息管理系统

3.1 流程分析

在一个企业的生产过程中存在着各式各样的流程

例如：生产部门的生产流程，销售部门的接单流程，销售流程等。而整个企业的生产、销售、结算也是一个大的流程，本文以锯床生产企业为例，将企业的生产、销售、采购、结算的整个流程为例对 workflow 系统的设计做详细介绍。

3.2 系统的业务流程

系统流程从客户有了订单意向，签订意向以后开始一个业务流程，同时产生一个唯一的流程编号，在销售部门确定订单任务，填写根据要求的标准格式填写表单以后，由 workflow 引擎将表单递送到下一个部门进行生产，依次经过 MRP 计算，派工生产、生产入库、仓库发货、到最后客户收货付款，最后等客户付完款项以后该流程结束。workflow 引擎会将表单送到下一个部门的子 workflow 引擎，等某部门完成以后将进展情况送回到主 workflow 引擎，这样每一个订单都能在各部门之间进行流转，而每一个部门都能知道订单的进展情况，系统整体流程图如图 2。

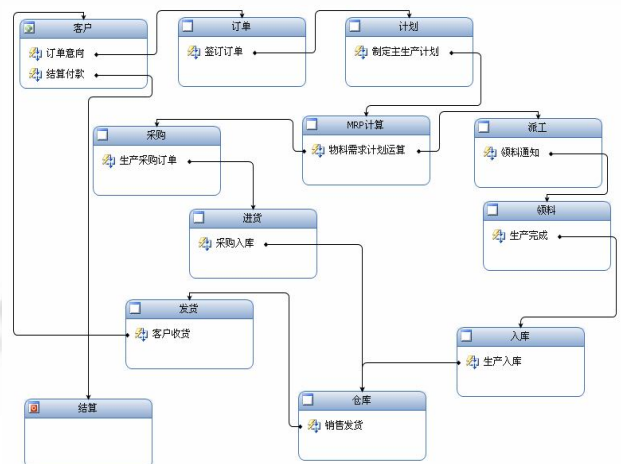


图 2 离散制造信息系统整体流程

3.3 功能描述

系统功能主要包括基础数据管理、仓库管理、销售管理、采购管理、计划管理和生产管理。

3.3.1 基础信息管理

基础数据是系统运行的前提，主要包括部门信息、人员信息、产品信息等基础信息的维护管理。而对基础信息的维护只有经过定义的人员才能进行系统，然

后对数据进行维护，基础数据的录入主要由人工初始化录入，只有把基础数据维护好信息系统才能够准确运行。

3.3.2 销售管理

销售业务流程是企业信息流的源头，也是物流的最后一步。企业根据市场和订单得到需求，将信息传递给计划、采购、生产、仓库等部门，最后从仓库获得所需货物，销售给客户，及时收回货款，办完财务手续，并进行销售业务统计分析，从而完成销售业务流程全过程的管理。销售管理主要内容包括，制定销售计划、订单管理、发货、退货处理、销售统计和分析、客户档案管理、订单执行跟踪等功能，销售管理过程如图 3。

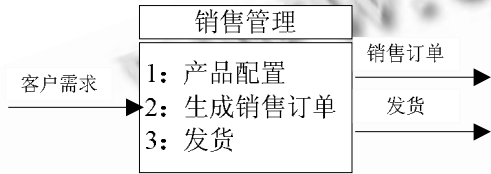


图 3 销售管理

①销售订单(合同)管理

根据客户的需求信息、交货信息、产品的相关信息及其他注意事项制定销售订单(合同),并通过对企业生产可供货情况以及产品定价情况和客户信誉情况的考查来确认销售订单(合同)。销售订单信息传递至生产计划部门,以安排生产,并进行订单跟踪与监控。销售订单是企业生产、销售发货和销售货款结算的依据。对销售订单的管理是销售工作的核心。

②销售发货管理

按销售订单的交货期,通过库存信息的查询组织货源,下达发货通知,并组织发货,并将发货情况转给财务部门。

③销售与市场分析

销售统计分析是对各种销售信息进行汇总统计分析。对各种产品的订单订货情况、销售情况、订单收款情况、销售发货情况、销售计划完成情况以及销售赢利情况等,从地区、客户、销售员及销售方式等多角度进行统计与分析。销售与市场分析的结果将成为

企业领导决策的依据。

3.3.3 生产管理

生产管理主要是指在企业的销售订单或者产品销售计话单的得到需求信息以后,在各类基础数据和系统参数设置完整、正确的前提下,进行日常业务主要包括根据生产计划制定相应生产任务单,生产领料,生产入库,生产退料,并形成企业管理必须的各种统计分析报表等。

①派工任务单管理

车间主任根据 MRP 计算得到的计划加工清单生产派工任务单,指定生产小组进行组装生产,同时决定生产起始时间和生产完成的时间。

②领料通知单

生产小组根据得到的生产派工单到仓库领料,进行生产。

③生产入库通知单

由 MRP 计算可以得到最终生产的成品入库清单,等车间生产完成以后,产生入库通知单,并将产品交付给仓库。

3.3.4 采购管理

采购管理在企业经营管理中占据了非常重要的位置。任何企业要向市场提供产品或服务都离不开原材料或消耗品的采购。采购管理子系统主要从供应商信息管理、采购及资金计划管理、请购管理、缺料维护、询价管理、采购订单管理、来料验收管理等方面管理采购业务,采购管理的过程如图 4。

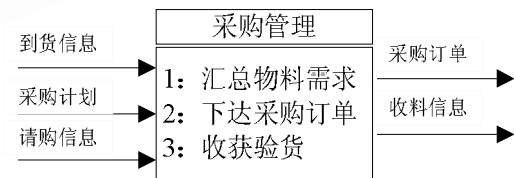


图 4 采购管理

①采购申请管理

采购申请单据(请购单),是采购需求来源之一。

②缺料维护

对主生产计划后的物料需求进行维护,是采购订单的主要来源之一。

③询价管理

采购人员对供应商进行价格咨询并商谈有关交货数量、交货期、质量要求与技术要求等问题,生成询价单据。

④采购订单管理

根据订货批量、采购提前期、库存量、运输方式、采购计划及计划外物料申请进行物料合并,生成采购订单,经过确认后进行订单输出,最后下达给供应商。也可以按销售订单需求直接向供应商下达采购订单。

采购业务人员下达采购订单后要对其进行跟踪,系统可以设置跟踪周期,形成订单跟催计划。在跟催过程中要了解供应商的生产进度及质量情况,并及时对供应商予以支持。

⑤来料验收管理

可以由采购业务员对来料按订单与发票进行验收,并录入收货单与发票信息。也可以直接由来料检验人员(或物料管理员)对来料按订单验收,并对不按计划交货的供应商进行管理控制。采用何种处理过程和方式可以由用户自行在系统中定义。

3.3.5 仓库管理

库存管理子系统通过对库存物品的入库、出库、移动和盘点等操作进行全面的控制和管理,帮助企业的仓库管理人员管理库存物品,以达到降低库存,减少资金占用,杜绝物料积压与短缺现象,提高客户服务水平,保证生产经营活动顺利进行的目。库存管理子系统管理库存物品的数量、库存成本和资金占用情况,以便企业可以及时了解和控制库存业务等各方面的准确数据。

①物料入库管理

采购订单的来料入库根据采购订单安排检验、接收物料,办理入库手续,开收料入库单(收货单、入库单),分配材料库存货位,同时监督来料是否与订单相符。对于其他情况的来料收货和入库,处理流程与采购订单的来料收货和入库不同。此类材料入库无需采购订单,经过审核收货且检验合格后,即可安排入库(进行仓库、货位分配)。

入库业务还包括生产完工入库、生产剩余物料入库以及销售退货入库。这几种入库在库存管理子系统

中分别按不同的流程来处理,对各种入库方式都可以通过自定义来实现。完工的物品有半成品和成品。生产完工入库后进行生产成本的计算,并将数据转到财务部门。销售退货有不同的处理方式,如扣减货款、换货等处理,相关的数据都需要转到财务。

②物料出库管理

物料出库有生产领料、非生产领料与销售提货。生产计划的领料按车间生产订单(加工单、工票、派工单或装配计划等,它们都来源于主生产计划和生产排程)与分工序用料,并可以根据物料清单(BOM)与工艺路线自动生成工序领料单或物料配送单。非生产领料有多种形式,系统可以自定义物料的种类。销售提货按销售订单或合同生成出库单据,并生成销售订单与合同的出货单。生成的销售出货单可以追溯相应的资料,如单据、订单、生产加工单与原材料信息,从而实现质量管理要求。这些过程都向财务传递相关数据。

③物料移动管理

物料的移动是库存之间的物料调拨,这种物料可以不经检验,也可以通过设置系统参数进行检验控制。

④库存盘点

库存盘点是对库存物品的清查,是对每一种库存物料进行清点数量、检查质量及登记盘点表的库存管理过程,其目的主要是为了清查库存的实物是否与账面数目相符以及库存物资的质量状态(即可用库存)。实物数量与账面数量有出入的,要在财务部门的共同协调下调整物料的账面数量,做到账物相符,并且应遵守相应的管理流程。每种库存物料都设立相应的盘点周期,并可以通过系统自动输出到期应盘点的物料。盘点方法提供冻结盘点和循环盘点两种。正在冻结盘点的物料须停止进行入库、出库操作,而循环盘点时允许同时进行入库、出库处理。系统支持对仓库、物料的盘点方式和范围选择,并生成盘点报表。

⑤库存物料信息分析

从各种角度对库存物料信息做分析,如日常的物料进、出、存的业务数据分析,物料占用资金分析,物料来源和去向分析,物料分类构成分析等,并可以分析各种库存管理指标,生成收发存明细账等多种统

计报表。这些信息对企业的库存与其他管理工作有监督和促进作用。

3.3.6 计划管理

根据主生产计划、物料清单、产品结构、库存等信息进行 MRP 计算以后得到生产任务计划,采购计划以及生产入库计划,是企业生产的核心部分。MRP 计算只需选择需要进行计算的生产计划,然后点击计算即可,对于已经计算过的生产计划系统会自动进行重新计算。

3.4 系统实体属性表:

表 1 系统实体属性表

实体号	实体名	实体属性
E01	产品表	编号, 名称, 单位库存等
E02	客户信息	编号, 名称, 联系电话, 联系人等
E03	订单	编号, 客户编号等
E04	订单明细	订单编号, 产品编号, 数量等
E05	发货单	编号, 客户编号等
E06	发货单明细	发货单编号, 产品编号, 数量等
E07	主生产计划	计划编号等
E08	主生产计划明细	计划编号, 产品编号, 数量, 生产日期等
E09	派工单	派工单编号, 车间编号, 制定时间等
E10	派工单明细	派工单编号, 产品编号, 数量, 开始时间, 结束时间等
E11	部门信息	部门编号, 部门名称, 上级部门等
E12	入库单	入库单编号, 仓库编号, 入库时间等
E13	入库单明细	入库单编号, 产品编号, 数量等
E14	出库单	出库单编号, 客户编号, 仓库编号等
E15	出库单明细	出库单编号, 产品编号, 数量等
E16	采购订单	采购订单编号, 采购计划编号, 供应商编号等
E17	采购订单明细	采购订单编号, 产品编号, 数量等
E18	物料清单	文件编号, 产品编号, 数量等

3.5 系统信息模型:

系统信息模型如图 5。

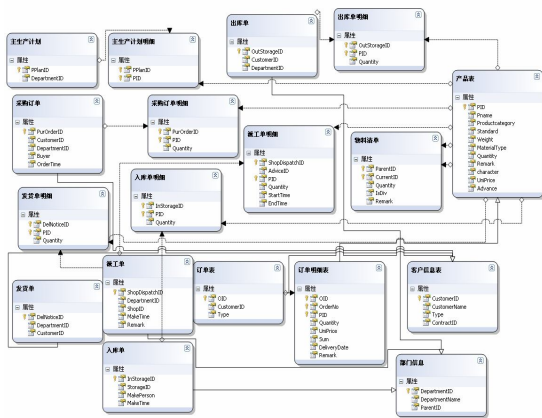


图 5 系统信息模型

4 应用实例

4.1 创建 WorkflowRuntime 对象

在 WF 环境中执行任务时,需要 Workflow Runtime

对象来监管执行的过程。一般情况下,我们需要整个应用程序保持一个工作流运行时的实例。一个管理工作流运行时普遍的方法是使用 Global.aspx 文件中的 Application_Start 和 Application_End 事件,将 WorkflowRuntime 对象保存在 Application 对象中。

```

void Application_Start(object sender, EventArgs e)
{
    //创建 workflowRuntime 对象
    WorkflowRuntime
    workflowRuntime =new WorkflowRuntime();
    ManualWorkflowSchedulerService
    manualService =new
    ManualWorkflowSchedulerService ();

    workflowRuntime.AddService(manualService);
    workflowRuntime.StartRuntime();
    // 将 workflowRuntime 对象保存到
    Application 对象中
    Application["WorkflowRuntime"]=
    workflowRuntime;
}

void Application_End(object sender,
EventArgs e)
{ //获取 workflowRuntime 对象
    WorkflowRuntime workflowRuntime =
    Application["WorkflowRuntime"]
    as
    WorkflowRuntime;
    //停止工作流
    workflowRuntime.StopRuntime();
}

```

4.2 使用 WF 引擎创建工作流实例

在 WEB 页面中添加一个按钮,在响应事件中通过 Application["workflowRuntime"] 中 获取 WorkflowRuntime 对象,创建工作流实例。启动工作流相应代码如下:

```

protected void StartRuntime_Click(object sender,
EventArgs e)
{
    WorkflowRuntime workflowRuntime =
    Application["WorkflowRuntime"]
    as
    WorkflowRuntime;

```

```

ManualWorkflowSchedulerService
manualScheduler
=workflowRuntime.GetService(typeof(ManualWorkflo
wSchedulerService))
as
ManualWorkflowSchedulerService;
WorkflowInstance instance =
workflowRuntime.CreateWorkflow(typeof(ASPNetSeq
quentialWorkflow));
instance.Start();
manualScheduler.RunWorkflow(instance.InstanceId);
}

```

4.3 持久化

由于日常流程中,参与流程中的人员往往不止一个,而一个工作流实例的运行时间可能很久,可能是几天或是一个月,WF 为了提高运行速度把工作流实例保存在内存中,将暂时不用的工作流实例移出内存,当工作流实例需再次激活的时候可以重新将工作流实例载入内存。WF 提供 **SqlWorkflowPersistence Service** 服务,将该服务加载到工作流引擎中以后即可实现持久化。在以下几种情况下工作流会被持久化^[5]:

① **Idle** 的时候 (如等待外部事件,使用 **Delay Activity**)

② 工作流完成或是终止。

③ 当 **TransactionScopeActivity** 完成的时候。

④ 当 **CompensatableSequenceActivity** 完成的时候。

⑤ 当一个装饰有 **PersistOnCloseAttribute** 的自定义活动完成的时候。

⑥ 当你手动的去调用 **Unload** 或 **TryUnload** 方法的时候。

5 结语

WF 自发布以来广受程序员的关注,但是由于 WF 实际上是引入了人一种新的编程方式,使人们开始很难理解,同时也由于 WF 在企业级的应用上缺乏稳定性,使得该项技术并没有像 WCF, WPF 那样备受青睐,但是很快人们就会体会到 WF 为编程方便以及效率的提高。本文将 WF 技术应用到信息管理系统中,发挥工作流管理对于流程控制和设计方面的优势,通过在离散制造业的实际应用,对企业的流程重组和再造起到了积极的作用,提高了企业工作效率以及经济效益。

参考文献:

- 1 范玉顺.工作流管理技术基础.北京:清华大学出版社,2001.
- 2 张瑾.WWF 开发指南.北京:机械工业出版社,2008.
- 3 蔡昭权.基于 WWF 的自定义工作流研究与实现.计算机应用化学,2008,25(10):1280-1282.
- 4 Kitta T.陈宇寒译.Windows workflow foundation 高级编程.北京:清华大学出版社,2008.