

规则引擎在制造企业 MES 中的研究与应用

闫 欢, 张宜生, 李德群

(华中科技大学模具技术国家重点实验室, 武汉 430074)

摘 要: 在当前制造企业的制造执行系统 (MES) 中, 存在大量灵活多变的业务规则, 这大大增加了这些企业 MES 的实施难度。传统的 MES 系统把业务逻辑和代码逻辑混杂在一起, 使得企业的生产策略和业务逻辑不能灵活改变以适应现实情况。该文通过在 MES 中引入业务规则管理系统 (BRMS), 利用规则引擎及相应的规则语言, 针对企业需求制定业务规则, 把程序代码和业务规则分离开来, 使企业根据市场需求灵活调整企业生产方针, 加快了 MES 系统的开发、升级、维护过程, 增强了 MES 系统的灵活性和适应能力。

关键词: 制造执行系统; 业务规则管理系统; 规则引擎; 规则语言

Research and Application of Rule Engine in MES of Manufacture Enterprise

YAN Huan, ZHANG Yisheng, LI Dequn

(State Key Lab of Plastic Forming Simulation and Die & Mould Technology, Huazhong University of Science & Technology, Wuhan 430074)

【Abstract】 There are so many flexible business rules in MES of modern manufacture enterprises, which makes it difficult to carry out MES successfully. Traditional MES systems mix business rules with programming code, which makes manufacture policy and business rules cannot change flexibly to adapt practical condition. This paper introduces business rule management system (BRMS) into MES and using rule engine and particular rule language. Business rules are defined according to enterprise requirement, programming code and business rules are separated clearly. So the enterprise can adjust producing policy flexibly based on the change of market. Development, update and maintenance of MES can be accelerated; the flexibility and adaptability of MES can be enforced.

【Key words】 MES; BRMS; Rule engine; Rule language

20 世纪 90 年代初, 美国先进制造研究机构(ARM)提出了一种支持企业信息模型集成的概念: 制造执行系统(MES)。该机构将制造执行系统定位于上层计划管理系统与底层工业控制系统之间的、面向车间层的管理信息系统。根据该定义可知, MES 系统是介于企业资源计划(ERP)和生产控制系统(PCS)之间的, 用于企业内部详细资源调配、生产过程控制的系统。MES 所研究的企业日常生产涉及到众多的生产要素和业务规则, 它们的调度和安排是一项复杂的决策过程, 这种灵活性大大增加了 MES 成功实施的难度。本文提出了一种新的基于 Java 规则引擎和 J2EE 多层架构的 MES 解决方案, 可以有效地适应企业业务规则的变化, 增强了 MES 系统的适应能力。

1 传统的 MES 系统

传统的 MES 系统一般是基于 C/S 架构, 即服务器接受客户端的请求, 进行业务逻辑运算, 操作数据库完成相应流程, 最后将结果返回到客户端进行显示; 客户端需要安装特定的程序, 称为“胖客户端”。随着网络技术的发展, 越来越多的企业应用开始转移到以网络为基础的应用上, 由此出现了基于 B/S 分层架构的 MES 系统。在新的架构中, 大量的业务逻辑和企业运算作为中间层, 从客户端抽离出来, 形成专门的中间层应用服务器。客户端通过网络浏览器进行访问; 而中间层作为企业应用的可重用组件, 当企业业务逻辑发生变化时, 只须修改相应的中间层组件即可, 提高了系统的适应能力和扩展性。这两种 MES 系统架构如图 1 和图 2 所示。

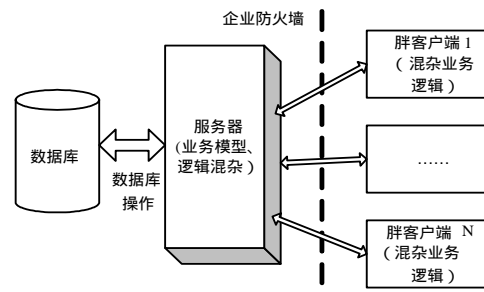


图 1 传统的基于 C/S 的 MES 系统架构

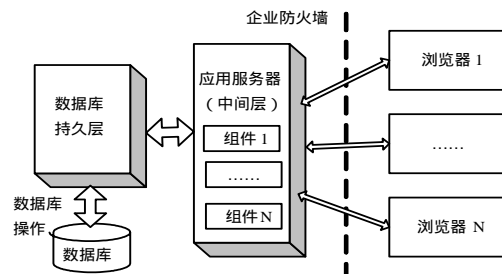


图 2 基于 B/S 分层架构的 MES 系统架构

基金项目: 国家“十五”科技攻关项目“工程塑料精密注射成型模具智能化设计技术研究”(2001BA205B09)

作者简介: 闫欢(1983-), 男, 硕士生, 主研方向: 企业数字化制造技术的应用; 张宜生、李德群, 教授

收稿日期: 2006-09-10 **E-mail:** island83@163.com

这种多层架构方式和以前的 C/S 架构相比,有了更大程度上的灵活性和重用性,但是随着企业流程的复杂化,以及在全球制造业激烈竞争的环境下,企业需要根据实际情况不断调整生产策略和各种计划安排,这种架构开始表现出一定的局限性,主要如下:

(1)虽然中间层组件更加清晰地表达了企业业务逻辑,但业务模型和业务逻辑仍然混合在一起,并没有彻底将业务逻辑和代码逻辑分离开来,在一定程度上仍然存在逻辑层次不清晰的弊病。

(2)这些中间层组件虽然是可重用的,但是当业务需求发生变化时,需要经过编码、编译、发布等一系列步骤后才能适应业务逻辑的变化,这使得企业 MES 系统的灵活适应能力不能充分发挥出来,增加了升级、维护阶段的复杂程度。

规则引擎的出现解决了这些传统架构的弊端,通过解析可以灵活地配置规则文件, MES 系统能适应企业需求的快速变化,增强系统的灵活性和适应能力,提高企业的生产效率。

2 基于 BRMS 的 MES 系统

因为传统 MES 系统的种种局限性,企业和软件开发人员开始关注能够灵活适应各种业务规则变化的软件框架,业务规则管理系统(BRMS)就是针对这一需求而产生的。借助业务规则管理系统,程序代码和业务规则分离开来,当系统的需求发生变化时,只须修改业务规则,而无须对整个系统进行复杂的升级修改。基于 BRMS 的 MES 系统如图 3 所示。

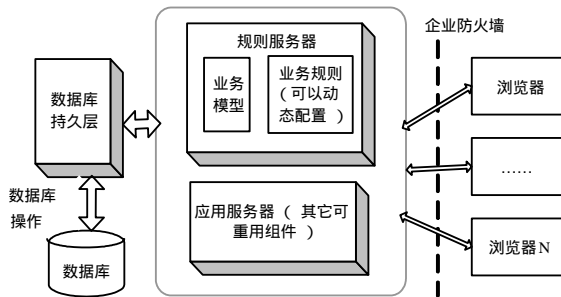


图 3 基于业务规则管理系统的 MES 模型

一个完整的 BRMS 通常包括:规则引擎,规则库,规则语言,规则管理工具 4 个部分。规则引擎是执行业务规则的软件组件,是业务规则管理系统的核心元素。规则库用于存储规则和规则元数据以及与规则有关的属性。规则语言用来定义业务规则,不同的规则引擎有不同的定义格式。规则管理工具用于管理、创建、修改和部署业务规则的图形化工具。

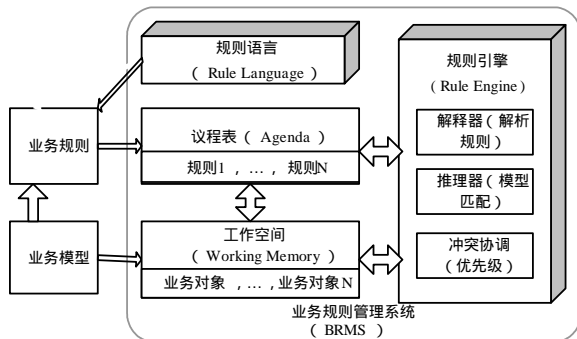


图 4 BRMS 的组成和业务规则的执行过程

开发基于 BRMS 的应用时,需要首先定义系统中涉及到的各种实体,将其抽象为程序中对应的类和对象,并使用特定的规则语言定义业务规则。规则引擎启动后,首先装载待施加规则的对象,这些对象称为 Working Memory。规则引擎负责所定义规则的实例化,根据装载的 Working Memory 按

照特定的算法进行模式匹配,匹配的结果决定业务规则执行的顺序,一条具体的业务规则可以有多个实例。这些实例化的业务规则的组合形成了规则议程表(Agenda)。规则引擎根据 Agenda 中的规则,逐条取出并解析规则含义,施加到 Working Memory 上。整个过程如图 4 所示。

3 应用规则引擎的关键技术

3.1 本系统使用的规则引擎和开发平台

目前规则引擎软件产品在国外的企业生产中已经得到应用,其中 ILOG 公司的 JRules 规则引擎很受欢迎。美国 Sandia 实验室的 Jess 引擎也得到了应用,它的独特之处在于规则语言是基于严谨的 LISP 知识表达语言。除这些商用规则引擎产品之外,还有一些开源的实现,比如 Drools 规则引擎,它利用 XML 文件格式定义规则,大大简化了规则引擎的解析和执行过程。本系统采用的规则引擎是基于 Java 的 ILOG JRules 5.0 规则引擎。

本系统采用业界广泛使用的 J2EE(JavaEE)软件平台,一个典型的 J2EE 应用分为应用层、业务层、持久层等。J2EE 具有良好的可扩展性以及与系统平台无关的特点,在企业信息化过程中得到了广泛的应用。JRules 规则引擎是采用纯 Java 的实现,可以作为 J2EE 中间层的一个组件集成到整个系统中。

业务规则需要经过设计、开发、部署 3 个阶段。在设计阶段,业务人员须和系统开发人员一起完成业务对象模型(BOM)的设计,为业务规则的开发提供必要的基本“词汇”;在开发阶段,业务规则开发和系统程序开发可同时进行,业务规则制定者只须熟悉特定的规则语言即可;在部署阶段,规则部署人员须将规则集部署在相应的位置上,使规则引擎可以访问到相应的规则集以实现特定流程。整个过程如图 5。

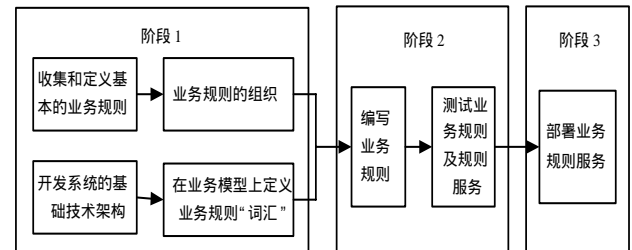


图 5 MES系统中业务规则的开发过程

3.2 MES 系统的模块功能划分

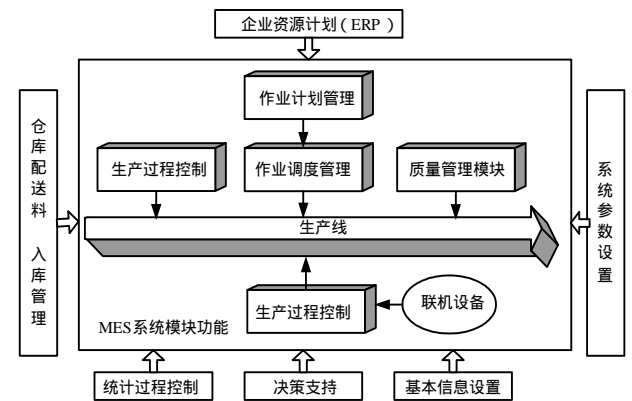


图 6 一个制造企业 MES 的功能模块

MES 的主要功能模块包括工序调度、资源分配、生产单元分配、过程管理、人力资源管理、维护管理、质量管理、文档控制、产品跟踪、性能分析和数据采集等。完备的 MES

系统可以在编制生产作业计划中对作业进行排序和调度来优化车间性能,最大限度地减少生产过程中的准备时间;根据作业任务,分配人员、设备、工具和物料协调地进行生产,以保证企业生产正常进行;并提供资源使用情况的历史记录,跟踪其现在的工作状态和完工情况。本系统主要功能模块如图 6 所示。

3.3 企业业务规则的定义

ILog 定义了自己的规则文件格式,后缀名为 irl。每条规则由规则头(Header)、条件(Condition)、动作(Action)组成。通常,规则的条件部分和动作部分又称为规则的左端(LHS)和右端(RHS)。LHS 包含规则的匹配模式,RHS 则包含该规则被触发时执行的动作。每一个规则由 rule 关键词定义,后边紧跟的是规则的名称,可以用 priority 关键词定义规则的优先级,规则的条件由 when 关键词定义,规则的动作由 then 关键词定义。规则的定义形式如下:

```
rule rule-name {
    priority;
    when{
    condition 1; ...; condition N
    } then {
    action1; ...; action N }
};
```

JRules 规则引擎采用 RETE 匹配算法。在基于规则的语言中,范式是组成规则的元素。RETE 算法通过匹配来决定规则的执行。RETE 算法是一个快速的模式匹配器,是由语言编译器生成的一个静态判别网络。它通过在网络上存储的规则信息来快速地实现匹配。在每一个推理步的循环中,它不是用已知的事实去匹配每一条规则,而是仅仅考察那些事实有变化的匹配。事实只要被匹配一次,范式中的变量就被存储起来。如果另一个事实被断言,则只需简单地比较范式变量的值,如果相等,这个规则就被激活。由于在每一个推理步循环中,那些没有变化的数据可以忽略,因此 RETE 算法能大大地提高事实与规则前件的匹配速度。

3.4 规则引擎的执行和系统的技术架构

业务规则开发完毕后,在应用程序中启动规则引擎需要新建一个规则集对象,它是 IrlRuleSet 类的实例;然后把规则集文件读入到一个 InputStream 对象中去,用 IrlRuleSet 的 parse 方法进行解析,如果解析成功,规则集文件所包含的业务规则就绑定到该规则集对象上。再新建一个规则引擎上下文对象,即 IrlContext 类的实例,该对象使用刚才已经绑定业务规则的 IrlRuleSet 对象作为构造参数,同时可以用一个 IrlParameterMap 的对象将应用程序中的对象作为参数传递到规则引擎上下文中,利用这种方法可以在应用程序和规则引擎之间传递数据。

准备完毕后,通过 IrlContext 对象的 insert 方法,可以把应用程序需要用到的业务模型对象(BOM)依次插入到规则引擎上下文中,形成 Working Memory。每一个对象插入时,规则引擎根据规则的匹配条件,采用 RETE 算法进行模式匹配,按照规则定义的优先级触发一系列的规则形成 Agenda 列表。执行完毕后,通过调用 IrlContext 对象的 retractAll 方法,将所有留在 Working Memory 中的对象全部撤销,然后用 end 方法结束规则引擎的运行。

本系统使用开源的 Webwork 框架作为 Web 层的 MVC 框架,用 Spring 框架作为业务层的框架,规则引擎执行完毕后,

利用 Hibernate 开源框架进行对象的持久化工作,将结果保存在数据库中,整个过程如图 7 所示。

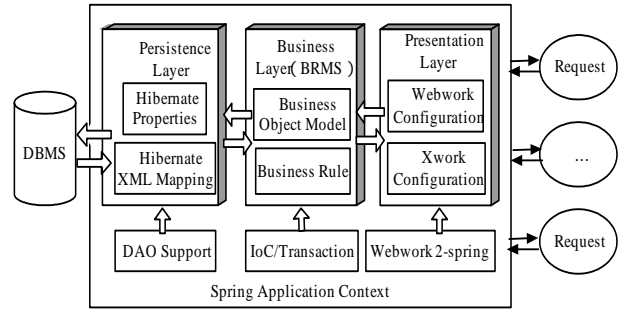


图 7 基于 BRMS 和开源框架的 MES 技术架构

3.5 基于 BRMS 的 MES 系统与传统模式的对比

本系统通过基于 Hibernate 开源框架的持久层将数据持久化到数据库,然后以网页的形式将结果展现给用户。MES 注重企业资源的合理调配,通过规则引擎根据既定的企业规则进行自动安排,可以优化企业生产流程,充分发挥企业生产资源的功效,合理制定企业生产计划,节约企业生产成本。

本系统在根据订单状况决定生产计划的流程中存在的规则:如果同一月份内两个订单所订购的产品和需要使用的钢卷类型均相同,那么这两个订单可以安排到一个生产计划中,该生产计划的产品数量为每个订单订购数量的总和,而完成日期为订单中最靠前的日期。传统的方式是把业务规则混合在程序代码内,如果随后企业需求发生改变,假如要求完成日期比原定方案提前若干天,那么对于既定策略的任何改变,都必须经过修改源代码、编译、发布等一系列步骤;而通过引入规则引擎和规则语言,只需要修改对应的规则即可,如图 8 和图 9 所示。

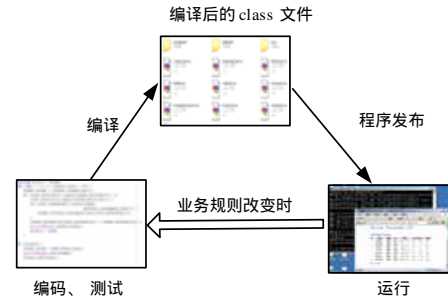


图 8 传统模式升级时需重复整个开发过程的示意图

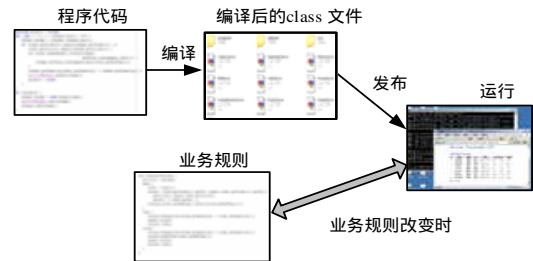


图 9 引入 BRMS 后只需修改对应的业务规则的示意图

在日常生产安排流程中,不同日期、时段的生产需要安排当天工作负荷最小的机械、模具、人员。传统的方式是由工作人员手动排班,排班的合理程度依赖于生产计划制定人员的个人经验和熟练程度;本系统利用规则引擎自动完成这些流程,使得生产资源的配置按照确定的企业规则进行,因

(下转第 224 页)