# 面向柔性装配的制造执行系统任务管理的研究与实现

# 摘 要·

XXXX.

**关键词：柔性装配；制造执行系统；SaaS；微服务；**

目录

[面向柔性装配的制造执行系统任务管理的研究与实现 1](#_Toc529998336)

[摘 要· 1](#_Toc529998337)

[第一章 绪论 3](#_Toc529998338)

[1.1 研究背景及意义 3](#_Toc529998339)

[1.2 国内外研究现状 4](#_Toc529998340)

[1.2.1 复杂产品装配的研究现状 4](#_Toc529998341)

[1.2.2 制造执行系统研究现状 5](#_Toc529998342)

[1.2.3 SaaS服务发展研究现状 7](#_Toc529998343)

[1.3 研究目的及内容 9](#_Toc529998344)

[1.3.1 研究目的 9](#_Toc529998345)

[1.3.2 研究内容 9](#_Toc529998346)

[1.4 研究内容与章节安排 10](#_Toc529998347)

[第二章 基于SaaS的应用软件开发模式和关键技术 12](#_Toc529998348)

[2.1 SaaS应用模式介绍 12](#_Toc529998349)

[2.1.1 SaaS模式基本概念 12](#_Toc529998350)

[2.1.2 SaaS模式优势 13](#_Toc529998351)

[2.1.3 SaaS成熟度模型 14](#_Toc529998352)

[2.1.4 基于云计算的其他服务模式 15](#_Toc529998353)

[2.2 基于SaaS模式的装配制造执行系统研发关键技术 16](#_Toc529998354)

[2.2.1 多用户数据管理模式 16](#_Toc529998355)

[2.2.2 服务可配置 18](#_Toc529998356)

[2.2.3 多租户协作 19](#_Toc529998357)

[2.2.4 基于PSL的模型统一化 21](#_Toc529998358)

[本章小结 23](#_Toc529998359)

[第三章 基于SaaS的装配任务管理模式的系统架构设计 24](#_Toc529998360)

[3.1 SaaS服务体系架构 24](#_Toc529998361)

[3.2 微服务框架设计 25](#_Toc529998362)

[3.2.1 微服务理论研究 25](#_Toc529998363)

[3.2.2 微服务架构研究现状 26](#_Toc529998364)

[3.2.3 微服务架构设计方案 27](#_Toc529998365)

[3.3 基于SaaS的微服务系统架构实现 30](#_Toc529998366)

[3.3.1 微服务系统拆分（系统功能需求） 30](#_Toc529998367)

[3.3.2 微服务系统服务定制 32](#_Toc529998368)

[3.3.3 SaaS多租户管理 34](#_Toc529998369)

[3.3.2.1 多租户数据模型设计（数据一致性、数据隔离性、数据传播、数据分组） 34](#_Toc529998370)

[3.3.2.2 多租户协作设计 34](#_Toc529998371)

[3.3.2.3 界面可配置（通过调整界面的参数实现） 34](#_Toc529998372)

[本章小结 34](#_Toc529998373)

[第四章 基于SaaS的装配任务管理系统功能开发 35](#_Toc529998374)

[4.1 功能设计 35](#_Toc529998375)

[4.1.1 订单任务管理服务设计 35](#_Toc529998376)

[4.1.2 生产计划管理服务设计 39](#_Toc529998377)

[4.1.3 在制任务管理服务设计 44](#_Toc529998378)

[4.1.4 任务执行管理服务设计 48](#_Toc529998379)

[4.3 基于SaaS模式的MRP算法优化 50](#_Toc529998380)

[4.3.1 MRP理论研究 50](#_Toc529998381)

[4.3.2 算法详细设计 51](#_Toc529998382)

[4.3.3 不同模块下算法的适应性（参数配置）（物料的库存在不同的地方属于供应链之间的问题） 53](#_Toc529998383)

[4.3.4 数据一致性 53](#_Toc529998384)

[本章小结 53](#_Toc529998385)

[第五章 系统实现与验证 54](#_Toc529998386)

[5.1 系统开发环境与技术 54](#_Toc529998387)

[5.1.1 系统开发环境 54](#_Toc529998388)

[5.1.2 系统运行环境 55](#_Toc529998389)

[5.2 系统架构与关键模块实现 56](#_Toc529998390)

[5.2.1 SaaS服务订制化 56](#_Toc529998391)

[5.2.1.1 订单管理 56](#_Toc529998392)

[5.2.1.2 计划管理 56](#_Toc529998393)

[5.2.1.3 在制任务管理 56](#_Toc529998394)

[5.2.1.4 任务执行管理 57](#_Toc529998395)

[5.2.2 SaaS多租户功能 57](#_Toc529998396)

[5.3 实例展示(主要是界面展示，整个业务流程) 57](#_Toc529998397)

[本章小结 57](#_Toc529998398)

[第六章 总结与展望 58](#_Toc529998399)

[6.1 工作总结 58](#_Toc529998400)

[6.2 工作展望 58](#_Toc529998401)

[致谢 59](#_Toc529998402)

[参考文献 60](#_Toc529998403)

# 第一章 绪论

## 1.1 研究背景及意义

近年来，随着互联网信息时代的进步和先进制造技术的发展，全球工业技术水平、发展模式迎来了重大变革。德国提出“工业4.0”计划通过智能制造刺激制造业的发展。欧盟提出“2020增长战略”重点发展智能制造核心技术。韩国、日本等也纷纷提出发展智能制造相应的战略措施。我国在2015年推出了“中国制造2025”战略强调发展智能制造的重要性。

为了顺应智能制造的发展需要，全球制造企业正在从传统的“制造+销售”生产模式向“技术+管理+服务”的复杂生产模式转型，服务化是制造业发展趋势之一。制造业的全球化、一体化促使现代制造业不得不考虑在制造网络中，通过整合跨企业制造资源来实现制造资源和制造能力的共享、协调，建立灵活、动态适应的大型网络服务平台。在这种背景下，李伯虎院士提出了云的概念，将现有的制造业信息化技术与云计算、物联网等高新技术进行融合，将制造资源和制造能力服务化后进行集中管理，用户通过网络即时获取制造资源或者制造能力，实现制造资源和制造能力的高效共享和协同。云制造的应用将会推进制造业信息化的发展，促进中国制造2025的实现。

然而，云制造服务体系中面临众多的挑战，开发一个大规模复杂的协同应用，需要考虑云计算环境下协同技术的具体实现，一般制造领域的软件开发商显得捉襟见肘。当前云计算服务大致分为三类：基础设施即服务（IaaS）、平台即服务（PaaS）和软件即服务（SaaS）。IaaS服务主要是在云资源层和云系统层展开。PaaS服务主要是在软件层展开，依托于云系统层下的基础资源。SaaS服务相对于IaaS和PaaS而言，主要针对云应用软件层提供服务[1]，SaaS服务是面向整个区域的分布式应用服务,可以充分实现区域内的制造资源的共享,解决制造资源不完备的问题。

现代装配制造企业的生产完全以市场需求为导向，随着客户需求的个性化日益提高，企业一般采用多品种小批量的生产方式。为了加快市场需求的相应。企业不再以单一产品装配功能为主，实现了多企业联合生产，共享生产资源和功能。但是传统的装配制造执行系统不支持多企业联合生产的生产模式，具有局限性，不能适应现代生产装配需求，现代装配制造业急需将制造执行系统部署在云上，根据不同生产企业的需要定制制造执行系统的各个服务，同时协作装配生产。目前，国内制造业装配自动化实现水平不一，针对不同装配产品的装配流程也都不尽相同，基于云服务的制造执行系统的研究还不完善，因此对制造执行系统的设计提出了新的要求与挑战。

综上所述，传统的柔性装配制造执行系统已经不能满足现代装配生产信息化的需求，亟需将装配制造执行系统部署在云上。本课题提出将装配制造执行系统应用SaaS模式部署到云环境上。结合工厂实际生产情况，对柔性装配任务管理模型进行研究，验证装配制造执行系统SaaS化的可行性。

## 1.2 国内外研究现状

### 1.2.1 复杂产品装配的研究现状

复杂产品是指客户需求复杂、产品组成复杂、产品技术复杂、制造过程复杂、项目管理复杂的一类产品，如航天器、飞机、复杂机电产品、武器系统等。装配制造企业根据产品的类型和工艺的组织形式分为集中装配和离散装配。离散装配的产品一般具有较为复杂的产品BOM，由大量不同类型的零部件按照规定的功能和要求组装起来[2]。复杂产品的装配是典型的离散型装配生产模式，具有生产工艺复杂、单件小批量生产、装配周期长且生产不均衡、生产返工返修作业多等特点，使得产品装配的各项任务之间的先行装配关系非常复杂，装配过程数据难以描述且缺乏统一规范的信息模型[3]。国内外学者已经就复杂产品装配这一课题做了许多的研究。

钱芳，扈静[4]等针对机械产品装配过程中的实时物料配送问题，通过对物料配送计划、物料跟踪管理和实时反馈机制的运用，研究与制造执行系统结合的实时物料配送方法，实现装配车间的实时、小批量配送。

傅玉颖，潘晓弘[5]等针对装配制造企业供应链运作过程的不确定性，探讨了装配生产过程中协同组件再订购点和生产批量的问题，用模糊理论对物料库存控制和生产协同问题进行优化求解。

北京理工大学的刘检华、林晓青[6]等人针对装配制造企业的单件小批量离散生产特点，通过引入工作流管理技术，提出基于工作流的装配车间生产和控制方法，实现了生产过程的装配活动和装配数据的统一管理。

Limère, Veronique[7]针对装配过程齐套配套和现场配送物料的物料供应系统，提出针对齐套配套和现场备料两种供应系统，并对这两种模型进行了评估，结果表明使用混合策略具有一定的优势。

常智勇、赵杰、莫蓉等人[8]以复杂产品的装配现场控制为目标，通过数字化的手段，应用三维数字建模，提出了一个适用于复杂产品装配的数字化管理平台并且开发了原型系统。

刘炜、刘峰等人[9]针对航空航天企业装配现状，提出了基于搬运机器人的智能化装配生产模式，通过工位的柔性集成技术实现设备重用和装配工艺兼容，保证装配产品的质量。

Hui Cheng, Yuan Li等人[10]针对复杂产品装配过程中由于缺少装配资源无法完成装配过程的问题，研究了一种基于遗传算法和蚁群算法的装配顺序规划方法，优化装配路径，最后开发了一个仿真系统装配顺序和装配路径验证算法的有效性。

Wang Jia Li, Mao-Gen等人[11]针对复杂产品装配过程难以控制的问题，设计了一种异常控制策略，建立复杂产品装配过程的实时状态模型。最后设计了一个基于工厂现场的异常控制专家系统。

上述的国内外文献综述表明，现阶段对于复杂产品装配过程的研究已经较为成熟。但是目前关于复杂产品装配任务管理的研究主要停留在装配过程物料管理和装配物料配送方面。对于多企业之间建立统一的装配信息模型缺乏相关的研究。然而建立装配过程统一信息模型是实现装配制造执行系统的SaaS模式化的关键，因此对装配过程进行分析，并且利用PSL工具建立装配过程统一模型是本论文研究的重点之一。

### 1.2.2 制造执行系统研究现状

随着制造型企业信息化技术的发展，以企业资源计划（ERP）为代表的管理信息系统开始在制造企业的生产经营中发挥重要的作用。但是随着市场需求的变化和市场竞争的加剧，企业对制造过程的要求不断提高，企业的关注重心从计划的执行结果转移到计划的执行过程，制造执行系统（MES）应运而生。MES的产生弥补了计划和控制系统之间的断层，有效的实现了计划和控制系统的数据传输和共享，同时增强了企业对车间生产的管理和控制。制造执行系统的概念兴起于上个世纪90年代，20多年以来国内外学者对MES系统的发展趋势及应用进行深入的研究与分析。

杨浩、朱剑英[12]等人针对分布式制造执行系统的建模，提出了一种模块化、可配置集成的基于多Agent的MES系统模型，并详细分析了MES的功能模型，实现了MES系统框架。

北航杨建军教授[13, 14]的团队针对敏捷制造企业的车间生产管理系统的研究现状，分析了制造执行系统的功能和应用特点，提出了使用先进的MES系统思想和软件建立敏捷车间系统，同时构建了基于全能体的MES解决MES系统的敏捷性。

周国利[15]等人针对面向订单生产的制造企业，分析了制造企业的功能需求和实施环境，设计了基于订单装配的N公司的制造执行系统。

Hwa Gyoo Park等人[16]针对当前MES系统庞大、冗杂、单一的特点，采用分布式面向对象技术，提出采用系统方法和CORBA基础设施来开发MES系统，将MES变为开放、模块化、分布式和可配置的系统，并且开发了原型系统用来验证。

王炳刚、周伟、饶运清、何非[17]等人对传统制造执行系统的不足和可配置制造执行系统进行了研究,设计了装配流程配置过程的数据模型,提出装配流程配置的方法,根据背景企业的需求,介绍系统各模块的功能,对产品制造流程进行配置,设计了系统软件架构,并对系统进行了开发实现。

沈晓杰、李郡等人[18]基于 MES的统计过程控制系统，利用MES系统的数据采集能力，解决了大多数企业面临的数据采集不实时的问题，并且提高了企业解决质量问题的效率，帮助企业实现了由质量事后检验转变成事前预防。

对MES系统的研究不仅集中在对MES系统的应用进行扩展以及通过系统方法对传统的MES系统的重新设计和优化，同时国内外专家学者分析复杂产品的装配过程，对如飞机、发动机、汽车这样的复杂产品装配过程应用MES系统进行了研究。

吴锋、马里等人[19]针对飞机装配作业生产管理复杂的特点以及生产计划与调度难以实现等问题,在研究装配作业调度系统的基础上,提出面向制造执行系统生产模式的飞机装配作业调度算法。实现的装配作业调度系统以飞机装配过程为核心,以GIM/GRAI方法为计划决策的依据,利用赋时Petri网建立装配生产线模型,按周期计划安排生产等方法实现生产调度优化,从而使生产现场快速组织装配资源并按时完成生产计划。

Gandhi, Prakash等人[20]为了提高装配效率、缩短生产周期、提高质量，采用基于PC的自动化技术和数据库技术来控制柔性装配制造执行系统，实现了功能模块和软件架构的开发。

Z Yang, PK Wong等人[21]针对混合模型装配生产线适应产品快速变化的需求，提出了基于新型的基于RFID的 MES系统，并且提出新的启发式广义拉格朗日分解算法用于模型优化。

Jagdale, Sanjaykumar等人[22]设计了一种智能高效的MES系统，用于从装配工厂车间采集数据，提出了NIRMAN工厂信息系统的设计，控制和监控复杂的装配生产过程。

综观国内外关于MES系统的研究，可以发现对装配MES系统的研究停留在MES系统的优化控制生产过程以及对MES系统架构的优化，丰富MES系统的功能模块，但是MES系统的应用场景一直停留在单一工厂封闭式应用生产的场景，现阶段随着智能制造和云计算的提出，制造业现在偏向于多企业协作生产共享资源的趋势，传统的MES系统已经不能满足日益增长的市场和多变的产品需求，因此，研究将装配制造执行系统部署到云服务器上将是本文的重点之一。

### 1.2.3 SaaS服务发展研究现状

随着服务经济在全球的迅速崛起和云计算的兴起，一种新的软件服务应用模式——软件即服务（Software as a service, SaaS）开始引起广泛关注。SaaS是一种通过Internet提供软件服务的模式，厂商将自己的软件部署在远程服务器上，客户根据自己的需求通过互联网定制所需要的软件服务。通过SaaS模式，企业之间也能够联合制造生产，满足了云制造和“中国制造2025”对制造型企业提出的整合制造能力和制造资源的需求。作为一种软件服务模式，国内外学者已经进行了一系列的研究。

Demirkan, Haluk, Cheng Hsing等人[23]针对SaaS模式下的两个核心竞争力­——应用基础设施提供商（Application Infrastructure Provider, AIP）和应用服务提供商（Application Service Provider, ASP）对系统的动态影响，研究不同的协调策略下这两者对SaaS系统的性能的影响。

Bezemer，Zaidman等人[24]针对SaaS模式实现中多租户的问题进行了研究，提出了一种新的软件架构原则，充分利用制造资源，多个租户共享相同的应用程序和数据库实例。提高了硬件资源的使用率和维护的便利性，同时指出错误的架构可能会导致多租户维护的噩梦。

刘士军、向坚持等人[25, 26]针对中小企业之间的业务协同问题，构建了基于SaaS模式的中小企业协同服务支撑平台，研究了基于Portal的服务集成方法和基于SaaS服务实现企业协同业务的方法，最后通过一个纹织制造企业的订单设计制造协同案例验证基于SaaS服务的业务协同系统的可行性。

李卫、张云勇等人[27]针对电信运营商的SaaS业务发展问题，分析了运营商应用SaaS模式发展业务的优缺点，通过分析目前开展SaaS模式下的电信业务发展情况，提出电信运营商基于SaaS模式发展业务的建议。

范卫锋、吕锋等人[28]以工装过程为研究对象，分析了大型装配制造企业的生产特点，以多租户个性化业务流程定制为目标，提出了基于SaaS模式的工装系统，并进行了系统逻辑结构设计、业务流程设计以及验证模型的实现。

倪能等人[29]设计并实现了一种SaaS化的学校信息管理系统，该系统利用SAAS模式的优势弥补了传统软件应用模式下的弊端。该研究主要用到的关键技术是数据库相关技术、数据模型的设计、数据安全等。学校的信息管理系统相对于制造业管理来说，业务流程较为固定，简单。

罗日新等人[29]实现了一种面向SaaS应用的多租户数据管理系统。该研究设计多租户数据管理的分层模型，屏蔽了多租户数据管理的具体实现。并且针对租户对于业务个性化的需求，提出了基于配置描述符的流程定制方法，实现多租户下的业务定制。该系统在脑卒中电子健康档案管理系统中得以应用。

李新明、廖貅武等人[30]针对SaaS服务链中AIP和ASP间的协调问题，提出协调契约使免费使用效果达到最优，实现SaaS服务供应链的最优绩效，并通过数值模型验证了模型的有效性。

除了上述一些学术性机构发表的文献，许多行业内的巨头公司将SaaS应用到企业信息化和电子商务中取得了巨大的成就。著名的Salesforce公司[31]通过应用SaaS模式商业模式，借助Ajax技术，切入了多数企业必须的客户关系管理系统（Customer Relationship Management, CRM），在CRM领域取得了领先地位。德国软件公司SAP[24]推出了SaaS模式的产品Business By Design, 这是目前最全面的SaaS ERP产品。

自从2003年随着J2EE及.NET技术出现之后，在线租用应用软件技术取得了突破，国内外研究学者对SaaS应用模式的特点和模式应用的研究已经取得了巨大的进步，但是在制造执行系统领域，没有特别的多研究，因此将柔性装配制造执行系统SaaS化将是本课题研究重点之一。

## 1.3 研究目的及内容

### 1.3.1 研究目的

本课题旨在验证柔性装配制造执行系统运用SaaS应用模式架构的可行性，并且研究柔性装配制造执行系统SaaS化的关键技术，并且实现柔性装配制造执行系统中任务管理模块SaaS化。

具体来说，就是围绕基于SaaS应用模式的柔性装配制造执行系统展开。研究基于SaaS模式的软件开发方法，针对SaaS模式对多租户和服务定制化的要求，构建统一的装配制造执行系统任务管理的统一模型，利用微服务的系统框架将制造执行系统的模块拆分成不同的服务，根据不同制造企业现代化智能化程度和业务提供不同的服务配置功能。基于SaaS应用模式，将制造执行系统部署在云服务器上，验证柔性装配制造执行系统SaaS的可行性。

### 1.3.2 研究内容

本课题主要是采用PSL和PSLX统一建模工具对柔性装配制造执行系统中的数据建立信息模型和过程模型，运用SaaS的系统架构和微服务的系统框架对柔性装配系统优化。最后通过对柔性装配制造执行系统的任务管理模块SaaS化验证该思路的有效性。研究内容具体分为以下四个方面：

（1）SaaS应用模式关键技术研究

研究SaaS应用模式的运用场景和技术难点，分析SaaS应用模式下对软件开发的要求，针对SaaS化中的多租户和功能可配置的特点对装配制造执行系统的功能模块进行划分，并且修改底层数据库模型。

（2）基于PSL和PSLX建立柔性装配制造执行系统统一模型

为了适应不同制造企业用户的场景，采用PSL和PSLX对制造执行系统的数据模型定义，并且分析不同租户对装配系统的需要，构建装配制造执行系统统一模型。

（3）SaaS模式下柔性装配系统任务管理的业务流程分析和实现

为了适应SaaS应用模式对软件的要求，分析装配系统中任务管理的业务流程，根据SaaS化下服务可配置的要求，将软件分为不同的模块，基于建立的统一模型实现各个模块的功能需求。

（4）微服务系统框架的搭建

研究微服务系统框架的技术方案，将装配系统任管理模块拆分出来的功能模块采用微服务的系统框架，将功能模块服务化，服务与服务之间解耦，并通过服务解耦的设计实现功能服务可配置化。

## 1.4 研究内容与章节安排

本文的具体组织结构如下：

第一章，绪论。首先论述本文的研究背景和研究意义，然后对复杂产品装配、制造执行系统和SaaS服务发展的国内外研究现状进行分析，并且进行了相关比较。之后阐述了文章研究的主要内容和章节安排，并由此得出了课题研究的技术路线。

第二章，基于SaaS的应用软件开发模式和关键技术。首先介绍了SaaS的基本概念、优势和成熟度模型，并且简单介绍了基于云计算的另外两种模式——IaaS和PaaS。之后研究SaaS服务软件研发方法，主要从多租户的数据管理模式、多租户协作生产、软件服务定制化以及装配过程统一模型几个方面研究SaaS模式软件开发的关键技术。

第三章，基于SaaS的装配任务管理模式的系统架构设计。首先从框架实现的角度，设计SaaS化柔性装配制造执行系统的整体技术方案，选择微服务的技术框架来实现SaaS模式。然后介绍微服务各个模块的功能，并且根据微服务系统框架思路，结合柔性装配任务管理业务流程，将系统划分成各个系统服务模块。最后研究在微服务框架下的各个服务模块远程调用的方法。

第四章，基于SaaS的装配任务管理系统功能开发。首先分析柔性装配制造执行系统各服务模块的业务流程，结合SaaS模式的设计方法，构建各个服务的功能模型。然后结合多租户的设计原则，参考PSLX中对任务管理各个模型的定义，建立SaaS下的柔性制造系统的统一信息模型。最后建立柔性装配系统的过程模型，并且对传统的MRP算法做出了改进。

第五章，系统实现与验证。首先对系统的实现环境和运行环境做了详细的介绍，然后对系统内部的各个服务模块的实现进行了详细的介绍。最后对系统的功能进行验证，验证装配制造执行系统任务管理的业务流程的完备性和正确性，验证SaaS模式下的多租户的独立生产和协作装配以及验证多租户服务定制化的实现。

第六章，总结与展望。对全文的研究进行总结并且对未来的工作进行展望。

全文的技术路线结构如图所示。

# 第二章 基于SaaS的应用软件开发模式和关键技术

## 2.1 SaaS应用模式介绍

### 2.1.1 SaaS模式基本概念

SaaS[32]是Software-as-a-service（软件即服务）的简称，是一种基于互联网提供服务的创新型软件应用模式，随着通信技术和云计算等技术的应用和发展，SaaS应用模式开始兴起。与传统的软件应用模式不同，如图1-a与图1-b所示，企业用户不需要购买软件并在本地安装软件应用， 而是通过服务提供商租用服务软件来管理企业的生产经营。软件服务提供商利用先进的互联网技术为企业用户搭建一个信息化管理公共平台，并且负责所有的软件搭建和维护工作，企业可以根据自己的实际需求，包括软件的功能、软件的租用时间、软件的用户量等，向软件服务提供商订购需要的软件服务。用户从传统的购买软件转变为租用软件服务，不再需要为购买软件支付巨额费用，也不需要搭建软件和硬件运行平台。同时可以通过有效的技术措施保证企业用户的保密性和安全性。当企业的需求发生改变时，不需要重新购买另一套软件，企业只需要在服务提供商的服务提供平台上配置其他需要的服务即可。

SaaS应用模式一般有三层含义[33]：在表现层，SaaS是一种业务模式，意味着服务提供商可以通过租赁的形式向租户提供软件服务，解决软件部署和维护的问题，降低了企业用户购买软件的成本；在接口层，SaaS是一种统一的接口方式，可以方便用户和其他应用在远端进行模块之间的调用，实现服务分离，业务组合，同时不仅是在业务上的一种接口调用，在云计算的三个层次中，位于上层的SaaS服务也能使用IaaS以及PaaS的服务，实现云计算架构的完整实现；在应用实现层，SaaS是一种软件能力，在软件的实现过程中，需要考虑到资源和能力的共享，使得一套软件能够方便多个租户共同使用。

图 1-a 传统软件应用模式 图 1-b SaaS软件应用模式

### 2.1.2 SaaS模式优势

SaaS作为一种新兴的软件应用模式，与传统的软件应用模式相比，有以下几个方面的优势。

（1）软件的服务模式从购买转变为租赁。SaaS服务提供商将应用软件部署在统一的服务器上，取消了传统的软件授权收费方式，降低了用户在软硬件方面管理和维护的费用。相对于传统软件而言，SaaS模式在软件部署、管理、升级维护等方面均有较大的优势。

（2）SaaS项目部署周期短、风险低。与传统软件相比，SaaS模式统一管理服务器上部署的软件，即使部署失败，也不会向传统软件部署一样所有前期投入几乎全部浪费，降低了项目投资的风险。

（3）使用方便。传统软件服务模式受限于控件和时间的限制，需要在用户本地的服务器上安装部署软件，并且通过授权才能使用，一旦软件使用过程中需要维护或者用户对软件提出了新的需求，那么就需要软件研发人员实地调试并重新安装部署。而SaaS模式服务软件可以方便租户在何时何地都能通过Internet接入使用，无需安装，从而为租户带来极大的便利。

（4）服务定制。SaaS模式应用软件只需要提供一套软件系统可以同时支持多个用户，用户根据自己的需要定制所需的服务，如果用户对软件的功能需求发生了变化，可以通过更改用户定制的服务达到需求目的。

（5）安全可靠。SaaS模式应用软件运行在服务提供商的统一的服务器中，服务提供商采用安全的数据备份和数据中心等手段保证服务器端的安全可靠，统一管理。而用户是通过浏览器客户端来访问软件服务，即使客户端被木马软件入侵，不会影响到服务器端的数据安全。实现应用和部署分离。

### 2.1.3 SaaS成熟度模型

微软架构师Frederick和Gianpaolo[34]就SaaS软件研发现状，确定了SaaS软件应用的三个特性：可配置、高性能和可扩展。这三个方面决定了应用的可用性、健壮性和灵活性。一般的SaaS应用架构可以根据这三个特性划分成四级应用成熟度。如图2所示的四种不同的SaaS应用架构给出了不同的成熟度模型[35]。 

图 2 SaaS软件成熟度模型

如图（a）所示，第一级成熟度的SaaS模式与传统的应用软件供应商（如ASP）提供的软件应用模式相同，在这种软件成熟度模型中，软件提供商为每个客户定制一套软件，并为其部署。每个客户分别拥有特定的应用程序实例，运行在SaaS服务器上。这一级的成熟度实例与传统的软件应用模式类似，客户的应用实例都是独立的，不同的实例之间是相互隔离的。不同的是软件服务提供商将不同的软件实例统一部署在SaaS平台的服务器上。

如图（b）所示，第二级成熟度的SaaS要求实现软件的可配置性。在第一级的基础上对用户的应用实例采用相同的代码，服务提供商根据用户的需要提供相应的配置，满足不同客户的需求。当客户提出新需求时，只需要对代码做少量的更改就能立刻满足客户的需要。但是客户之间的应用程序仍然是独立的。

如图（c）所示，第三级成熟度上，单实例的应用程序可以支持多租户的需求，也就是多个租户共享同一个软件服务。软件服务提供商通过授权和安全性策略使不同客户的数据能够彼此分开。这种软件架构不再需要服务提供商为每个客户部署专门的单一服务器，充分利用计算机资源，降低成本。

如图（d）所示，第四级成熟度是最高SaaS软件应用成熟度，支持可扩展的多租户SaaS应用架构。通过负载均衡为每一个客户分配到不同的应用实例上，通过多个实例来分担用户的大量访问，可以让应用软件水平无限扩展，提高软件的灵活度。

### 2.1.4 基于云计算的其他服务模式

本课题的研究内容主要是围绕柔性装配制造执行系统SaaS化这一课题展开，旨在利用先进的云计算平台技术软件，为装配制造企业搭建统一的柔性装配系统软件服务平台。而云计算中的云分为基础设施云、平台云以及应用云，反映了云计算架构的基本层次。美国国家标准与技术研究院（NIST）提出云架构包含基础设施层（Infrastructre layer）、平台层(Platform layer)和应用层(Application layer) 3 个基本层次。其每层的功能以服务的形式提供出来，这就是云服务类型分类方式的来源，即基础设施即服务（Infrastructure as a service, IaaS）、平台即服务（Platform as a service, PaaS）、软件即服务(Software as a service, SaaS)。因此，有必要介绍云计算的其他两种服务模式——基础设施及服务（IaaS）以及平台即服务（PaaS）。

基础设施即服务是指将硬件设备等计算资源封装成服务，通过租赁的形式提供给租户使用，允许租户动态申请和释放节点，有着复杂的计费方式和计费模型。运行在IaaS的基础平台之上的服务器是所有租户共享的方式运行，因此计算资源的使用效率很高。

平台即服务是指以服务的形式提供给研发人员研发和部署应用程序的平台，租户可以通过这个平台管理部署的应用程序。这种平台一般包括数据库、中间件以及其他开发所需的工具，所有的工具资源都是以服务的形式通过互联网提供。

在软件的开发过程中，云计算服务提供商专注于云架构中的某一层服务架构开发服务，而无需同时提供三个层次的服务。如Amazon云计算服务EC2[36]，Google的 GAE和Saleforce CRM就只分别向用户提供基础设施层服务、平台层服务和应用层服务。事实上，上层服务的提供者可以利用位于下层的服务架构来实现云计算服务架构，无需服务提供商自己实现所有架构和功能。本课题研究对象是基于应用层架构服务软件的设计，因此重点在于对SaaS架构的研究与实现。

## 2.2 基于SaaS模式的装配制造执行系统研发关键技术

### 2.2.1 多用户数据管理模式

现代装配制造企业的生产完全以市场需求为导向，为了加快市场需求的相应。企业不再以单一产品装配功能为主，实现了多企业联合生产，共享生产资源和功能。基于SaaS模式的装配制造执行系统面向的是整个装配制造企业集群，服务使用的受众愈多，意味着数据量愈大。基于SaaS模式的装配制造执行系统是部署在服务器集群中，通过网络集中访问和存取数据，成本低于部署在本地的情况，但是也就意味着用户与数据在一定程度是隔离的[37]。所以SaaS模式下必须保证数据的安全可靠。同时服务是以单一实例为多用户服务，并且考虑到制造企业之间的协作和资源共享，必须考虑到用户间数据的相互隔离和数据共享。目前有以下几种管理数据的方法：

（1）完全独立架构

租户共享SaaS软件服务提供商提供的服务器上的工具和应用程序，但是使用相互独立的数据库，实现数据在物理和逻辑层面同时隔离，这也是用户认为的数据最安全的方式。因为数据存放在不同的物理机上，不用担心个别租户对数据库底层数据结构的改动（如建立数据表的索引、触发器或视图等）对其他租户的数据造成的影响。同时对个别租户的数据备份和恢复也比较快，不用备份和恢复其他所有用户的数据库，一旦发生物理机的顺坏或者宕机，也不会造成整个SaaS服务的租户同时离线的情况。但是这种方式显然导致SaaS服务管理的数据库过多，从而使得使用的物理机等硬件设备的数量增多，意味着高额的硬件成本。也就是通过高成本换取最佳安全性。



图 3 完全独立数据库架构

因此，这一数据库应用模式是和对数据有着绝对安全性要求的大型企业，如银行、医疗机构等。

（2）部分独立架构

该模式是在完全独立架构的基础上扩大数据共享程度，所有用户给你共享数据库，但是为每一个用户创建各自独立的表结构，形成独立架构。这种模式下，可以使用较少的数据库为大量的用户服务，降低硬件资源成本。独立的表结构也能适应不同用户的需要，扩展数据模型比较方便。但是因为需要为每一个用户创建独立的表结构，使得数据库中表数量过多，在数据安全性和备份恢复方面增加了复杂度，增加管理难度。



图 4 部分独立数据库架构

因此，此模式适用于业务逻辑不复杂，并且数据量和用户不多的应用软件服务。

（3）完全共享架构

该模式下所有用户的数据完全共享，所有用户共享数据库和数据表。多个用户相同数据模型的记录存储在同一张表中，并且通过用户的ID或者企业的域名作为不同用户数据的区别。相对于前两种数据库架构，这种共享模式硬件资源成本最低，支持大量用户。但是因为所有用户共享数据库数据表，导致数据的安全和维护工作变得复杂。



图 5 完全共享数据库架构

因此，该模式性价比最高，适用于以较少服务器满足大量用户的应用场景。

本课题面向的是制造业中众多的装配企业，有大量的潜在用户基础，考虑到硬件资源成本的问题，完全独立架构不适合装配系统的数据库设计。其次考虑到装配系统流程复杂，数据模型较多，需要定义大量的数据表保存数据，且有历史数据的累计，为每一个用户单独设计数据表是不现实的，相对对于数据表的增加，数据记录的增加更能减轻数据库的负载，因此完全共享的数据库架构更适合装配系统的研发。

### 2.2.2 服务可配置

当前各个装配制造企业的装配现代化程度不一，并且装配的业务流程也都不尽相同。因此当前的柔性装配制造执行系统都是独立运行于各个制造企业之中，无法达到数据的共享和业务的平移流转。基于SaaS的装配系统是以单一的装配MES实例为云平台上众多的制造企业服务，但是每个装配制造企业用户对于服务的需求也不尽相同，也就意味着SaaS装配系统软件需要为每一个企业用户提供与传统应用软件实例相同的服务，也就是需要实现软件功能的可配置性。每个用户根据自己企业的装批生产流程和现代化程度定制相应的服务，并且保存配置信息。SaaS软件通过对企业用户的身份验证，提供相应的功能。

用户在SaaS软件平台上定制服务的前提就是服务平台为用户提供一个应用模板，模板以标准的应用服务为基础实现。模板中一部分应用为所有用户都需要的应用，另一部分应用则可以由用户根据自己的特殊需求定制。用户可以定制的服务可以称为可配置服务点[38]。装配制造企业用户需要租用服务时确定需要的配置服务点，完成服务配置工作，得到满足企业需求的软件应用服务，并将企业用户的相关配置信息存储到数据库中。在完成企业用户的服务配置之后，企业内工作人员需要根据人员的权限配置需求的界面、服务等得到最终的企业应用服务。

根据现阶段装配制造企业的业务流程和功能需要，本课题研究的服务可配置点应用从两个方面实现：1. 装配制造业务服务配置，装配制造企业间的装配业务流程不同，需要对装配过程中的订单管理、主生产计划管理、生产指令管理、装配任务管理、反馈等功能组成部分进行专项配置，实现每个装配企业的装配业务流程标准；2. 企业内部员工服务配置，每个装配企业都有一定的员工数量，每个员工都有各自的角色和职能，因此需要为每一个员工配置相应的界面和功能，系统通过权限控制和服务认证进行控制管理。如图为配置服务的整体架构，多租户访问控制策略（超级管理员和企业内部管理员，通过用例图对这些管理员进行分析）：

### 2.2.3 多租户协作

现代装配制造企业为了适应市场的需求，不再以满足单一产品的装配为目标，需要加强多企业之间的联合生产，共享生产能力和生产资源。如飞机的生产装配过程，飞机的装配可以分为构件装配、部件装配和总装配等阶段。构件装配是将一组零件装配成比较简单的构建，如翼梁、翼肋和隔框等；部件装配是将一组零件和构件装配成比较复杂的部件，如机翼、垂尾、机头、机身前段、机身中段、机身后段等；总装配是将所有部件组装成整架飞机。这些装配过程一般都在不同的专门化装配生产，最后在总装厂进行总装配生产。SaaS服务是面向整个区域的分布式应用服务，充分实现区域内的制造资源的共享，解决制造资源不完备的问题。因此采用SaaS模式的软件研发思路能够完全满足多租户协同制造的应用场景。但是，传统的装配制造执行系统是针对企业单一产品生产过程，任务模型和装配过程不适应多企业协作生产的场景。因此需要对任务模型和协作生产过程进行优化在设计。

首先对零件任务的信息模型进行设计，满足多租户协作装配制造的应用场景。装配任务不在仅仅表征任务的信息，还得设置任务的主制对象。在任务模型中如图所示，添加了主制企业和请制企业的信息，表示一个任务可以从一个企业中发布到另一个企业，那么发布任务的企业就是请求制造企业即请制企业，接受任务的企业为主制企业。在实际生产的过程中，不仅要考虑到企业之间的协作生产，还要考虑到企业内部出现的部门之间流转任务协作生产的情况，因此在零件任务模型中增加了主制车间和请制车间的信息。



图 6 零件任务信息模型

其次对企业装配过程优化，如下图所示，在装配任务流程开始之前，企业使用MRP算法对主生产计划分解产生采购需求和生产子任务，对于能在本企业生产的装配需求下发到该企业的生产车间中生产，如果是采购需求或者不在本企业内执行的任务，可以将该任务发放到统一任务管理处，等待协作企业接受任务，然后在该企业生产交付给原企业。如果是本企业车间内部可以进行的生产任务那么下发到车间生产。直到该产品装配所有的需求任务完成后，开始产品的总装配，最后完成交付。



图 7 租户协作生产时序图

### 2.2.4 基于PSL的模型统一化

现阶段装配制造企业对于装配MES系统的使用都是基于本企业对装配过程个性化的需求设计。并且装配制造企业对数据的管理方式大部分采用分散管理，多点集成的形式，因此会产生数据属性、结构、存储不一致的情况。导致产品生命周期各阶段信息很难传递和交换，导致信息闭塞，形成信息孤岛[39]。这种信息闭塞以及装配制造模型的不统一严重阻碍了装配系统SaaS化的进程。解决该问题的一条有效途径是通过过程集成,将先进的管理理念与信息技术相结合，实现产品开发过程的各阶段之间，以及产品开发与产品生命周期下游其他过程之间的信息交互和协同。

PSL是美国标准化技术局（NIST）根据制造企业工程经验提出的一种规范化的过程描述方法，目标是在整个产品生命周期中提供一致的表达方法，是实现异构系统间过程信息交互的统一标准语言。为了满足SaaS模式下装配系统对平台所所有制造企业的制造数据的定义，本文将采用PSL语言规范装配过程描述，然后在装配过程规范描述的基础上，采用面向对象的建模方法，完成装配过程信息模型的构建。PSL理论框架由3个部分组成：PSL核心、基础理论和PSL扩展。PSL核心是一系列用于描述基本过程的公理化语义单元，包括四个原始类(activity，activity-occurrence，timepoint，object)，两个原始函数(beginof，endof)和三个原始关系(before，occurrence-of，Participates-in)。基础理论是对PSL核心进行普遍适用性扩展的推理理论和相关公理，包括集合论、时限理论和情景演算等。PSL扩展是对PSL核心表达能力的扩充，通过加入新的常量和谓词提供核心以外的过程描述方法，如PSL外核扩展、通用活动扩展和排序扩展等。

装配过程描述本质上可以看作是对装配过程中的过程对象和对象之间约束关系的形式化表达，因此引入以过程为核心组织数据的方法，描述过程对象和对象之间的约束关系，并采用PSL核心及扩展中定义的类与关系对装配过程进行规范描述，在产品制造的整个生命周期为装配过程中的过程对象和对象间约束关系提供一致性的语义表达，从而规范信息模型的构建，满足不同企业间以及不同业务场景对装配过程描述的需要，满足装配产品生命周期制造数据一致性的要求。需要指出的是，采用PSL语言描述装配过程的目的并不是将其作为与其他信息系统进行信息交换的中间语言，而是借鉴PSL对过程建模和规范描述的能力。

下面针对产品装配任务执行过程主要的过程对象和对象管线，描述PSL中类与关系之间的映射关系，其于PSL的映射关系具体描述如下：（还需要修改，详细描述）

（1）装配过程中的订单（Order）、主生产计划（MPS）、生产指令（Demand）是最基本的组成部分，对应于四个原始类的Activity类。

（2）工艺活动（routing）与任务活动在装配过程中是定义与实例的关系，对应于PSL中 Activity类与 Activity-Occurence类之间的关系。

（3）生产任务（PartTask）作为活动的实例，在执行的过程中包含了活动的时间信息，对应于PSL四个原始类中的时间（Timepoint）类。

（4）装配过程中对象关系包含主生产计划与物料清单（BOM）、主生产计划与生产指令、生产指令与生产任务、订单明细（OrderDetail）与主生产计划。

（5）装配过程中工艺活动与零件任务活动、工序任务和工步之间的聚合、分解关系，对应于PSL扩展中Subactivity关系。

（6）同一层次的活动与活动之间的串行、并行以及汇合关系，对应于PSL扩展中的follows、start-synchronized、end-synchronized排序关系。

## 本章小结

本章首先对SaaS模式做了详细的介绍，根据柔性装配制造执行系统和SaaS模式的要求，从多租户数据管理模式、服务可配置、多租户协作装配和统一模型的建立四个方面对SaaS模式MES软件研发的关键技术做了详细的介绍。下一章将从这几个关键技术出发，研究基于SaaS的装配任务管理模式的系统架构。

# 第三章 基于SaaS的装配任务管理模式的系统架构设计

## 3.1 SaaS服务体系架构

SaaS服务平台是服务提供商与企业租户之间的中介平台，是装配制造执行系统实现的基础。（SaaS服务体系是什么）如下图所示，SaaS服务体系架构从下至上主要包括以下几个方面：

数据逻辑存储层：在第二章多数据管理模式中已经介绍过，主要是对多租户数据管理的层面，保证应用层租户之间的数据是彼此隔离独立同时又能使用多租户协作生产的场景，实现多租户的数据存储。

数据访问层：是业务逻辑访问数据存储层的接口，实现数据库中记录行的CRUD操作及访问控制。

业务逻辑层：实现装配系统的基本业务逻辑处理，对应系统的基本功能操作，主要是基于访问控制和业务规则对系统的数据进行操作。

组合服务层：具有业务整合功能，将相关联的基本业务功能单位组合成独立、完整的功能服务单元，对应于装配制造执行系统而言，需要将装配系统的功能服务拆分成独立的功能服务单元，是实现SaaS软件功能可配置的关键。

虚拟应用层：在组合服务层拆分出的服务单元的基础上，企业租户配置需要的服务功能单元和用户的界面。相当于为每一个企业配置一个软件应用实例，每个企业都对应一个虚拟应用。

界面控制层：即展现层，用户通过浏览器或者移动设备访问SaaS软件服务，在界面控制层识别登陆用户所属租户及其身份信息，通过登录人员的角色和权限配置用户界面。

可扩展组件：对功能定制、应用配置、数据层扩展、数据安全保证等方面提供支持。服务框架包括消息队列、安全中心、用户交互等服务框架。

图 2 SaaS模式软件架构

在第二章中分析了SaaS软件设计的关键技术主要包括多用户数据管理模式、服务可配置、多租户协作和装配过程统一模型的建立。其中数据管理模式和装配过程模型体现在数据逻辑存储层和数据访问层，服务可配置和多租户协作体现在业务逻辑层和组合服务层。SaaS模式是一种软件设计思路，具体的代码实现还需要有技术方案的支撑，为了适应组合服务层中的服务单元的设计，决定采用微服务的系统技术方案实现软件设计。

## 3.2 微服务框架设计

### 3.2.1 微服务理论研究

随着互联网和信息技术的发展，软件系统架构设计方案从单体式架构到面向服务架构一直在不断演进。近年来，面向服务的架构方式由于其灵活、解耦、规范化等诸多优势。被广泛应用于大型企业的系统架构。但是目前的面向服务架构也存在系统庞大、逻辑复杂、维护困难、研发成本高、可扩展性低等问题[40]。对于面向服务架构的设计缺点，国内外互联网企业探索通过微服务架构方式优化系统服务化的构建和管理。微服务和面向企业服务专门化软件设计方案最大的不同就是舍弃企业服务总线，旨在通过将功能分解到各个离散的服务中易实现对解决方案的解耦，将大型复杂的企业级服务系统应用分割成单一功能的小服务。每一个服务都是一个独立的功能模块，可以在短时间内完成启动部署，并且可以通过分布式部署实现主备防灾。每一服务都是一个独立的实体，可以分别部署在SaaS平台上。服务之间通过http和https网络传输协议进行网络通信，传递数据。

相对于分布式系统和面向服务的架构而言，微服务有以下一些好处：微服务的实现不会受到语言和技术栈的限制，每个服务可以使用最适合当前业务需求的技术来实现；微服务的系统健壮性更强，当系统的某个服务不可用时，不会导致整个系统的级联故障；微服务架构中，每一个单独的模块都可以为不同的应用程序提供服务，可以使用不同的方式构建应用；单一服务的体量小且功能单一，可以更轻易的重构和重写服务。

从SaaS软件四级成熟度模型上看，微服务的系统架构由于将复杂的系统分割成一个个微小自制的服务，服务之间的耦合度降低，各个微服务功能可以灵活的重用，增加了SaaS服务软件重构的灵活性；从SaaS软件服务可配置的要求上看，微服务架构划分组织是通过业务来划分服务而不是技术，租户可以在定制服务时，可以通过各个微服务模块功能的权限对租户开放，实现宏观服务的配置，不需要对具体的业务代码实现用户隔离；从数据隔离上看，微服务之间的数据调用是通过网络服务来实现数据的传递与调用，在数据逻辑处理层面减少了不同租户之间数据交互，增强了数据隔离性和安全性。

经过上述分析，微服务的系统技术方案相对于传统的软件设计方案有众多的优势，并且更适合SaaS模式下的软件设计方案，因此本课题采用微服务来研究SaaS模式下的装配系统实现。

### 3.2.2 微服务架构研究现状

微服务的概念是在2012年出现的，作为加快Web和移动应用程序进程的一种方法，2014年开始收到各方的关注，2015年是微服务的怨念，原来越多的论坛、社区、blog以及互联网行业巨头开始对微服务进行讨论和时间，进一步推动了微服务的发展和创新。国内外研究学者也就微服务的发展和应用进行了很多的研究。

郭栋、王伟、曾国荪等人[41]针对云件技术的发展，基于微服务架构，提出了一种新的云件PaaS平台，在不修改传统软件的情况下，直接将软件部署到远程服务器上，并且通过浏览器远程访问软件服务。通过采用微服务架构设计，使云件平台具有较好的可扩展性、容灾、灵活配置和简易部署等特性。

孟永胜、曹海萍等人[42]针对现代软件系统可扩展性的课题，研究了微服务系统架构的功能分解的过程，实现对复杂软件的解耦解决方案，并且提供了一种基于微服务架构动态扩展软件功能的方法。

彭雪涛、梁桌明等人[43]针对智慧校园的建设课题，分析了云端微服务架构及其应用优势，提出了基于智慧校园建设的微服务云端平台设计，包括信息服务的获取、教育资源的共享、基于学习数据的智能分析与评价等服务。

Hui Kang、Michael Le等人[44]结合新型的容器技术，使用OpenStack作为案例研究，启用动态服务注册于发现，基于容器的微服务式DevOps的效率，并研究了无状态和有状态容器化组件的可扩展性。解决了云基础架构管理设计的局限性。

Khanda Kevin、Salikhov Dilshat等人[45]利用先进的传感器网络以及以Jolie编程语言为中心的分布式微服务架构，研究用于支持智能建筑的应用平台，讨论了基于微服务架构的应用平台的特性和优势，并设计了原型系统。

北京农商银行[46]对微服务架构进行了深入的研究，并从银行的信息系统入手，研究了微服务软件研发路线，将传统的面向单体的软件转变为微服务系统能够架构，并于2016年启动服务网关，冲过应用系统，取得突破性进展。

微服务系统架构自诞生之初开始就在互联网行业掀起了一阵飓风，微服务的设计方案涉及到生活的方方面面，尤其与云计算服务联系紧密，是实现云平台架构的关键技术方案。

### 3.2.3 微服务架构设计方案

当前的微服务系统架构主要有两种实现方式：一种是由alibaba开发的Dubbo框架，另一种是Spring开源社区发布的SpringCloud框架。如图所示，SpringCloud与Dubbo框架相比，SpringCloud框架的系统结构更简单，通过传统的SpringMVC架构加上注册中心的功能实现SpringCloud架构，而dubbo需要使用各种复杂的url，protocol等技术；从性能上看，dubbo的网络消耗小于SpringCloud，但是现在随着网络技术的发展，网络消耗已经不会对系统造成太大问题；从开发难易度上，dubbo的开发难度极大，对jar包的版本和代码库的依赖极大，SpringCloud比较自由，接口管控层面比较高效。

|  |  |  |
| --- | --- | --- |
|  | SpringCloud | Dubbo |
| 开发难度 | 难度小，自由 | 难度大，不便管理 |
| 系统结构 | 简单 | 复杂 |
| 系统应用 | 应用广泛 | 应用广泛 |
| 技术支持 | 开源社区活跃 | 停止技术支持 |
| 性能 | 网络消耗较大 | Rpc调用网络消耗小 |

表 1 微服务框架对比

经过上述对比，本课题决定采用开发难度较小，技术支持较强的SpringCloud框架来实现本课题的微服务架构。

微服务主要用来实现SaaS服务体系中的组合服务层、数据逻辑访问层以及展现层的设计。其中展现层基于快速开发工具Dorado集成开发，完成用户界面的设计。采用http和https网络传输协议与后台服务解耦，将展现层和服务应用层之间的逻辑分离。通过Restful中间件对服务之间接口调用进行数据交互。使用Axure RP快速原型工具对界面UI进行设计，同时在Dorado的基础上使用H5、jsp等前端技术和BootStrap、React等前端技术框架研发界面。

其次，应用服务层分为业务逻辑服务和架构技术服务。其中架构技术服务主要包注册中心、负载均衡、容错保护、数据通信、服务安全等模块的设计。注册中心提供服务发现的能力，服务启动时向注册中心中注册自己的信息，注册中心是微服务的基础，本课题主要通过SpringCloud下的Eureka组件实现注册中心的功能；负载均衡是SaaS模式第四级成熟度模型的关键，租户的所有请求首先经过负载均衡合理的分配到每一个服务中，减轻服务器的负载，支持多个实例来分担用户的大量访问，可以让应用软件水平无限扩展，本课题通过SpringCloud框架下的Ribbon组件实现基于HTTP和TCP网络传输协议的负载均衡器；服务容错保护机制用来保护服务由于网络原因或者服务自身的问题导致调用服务异常，为了解决这一问题，使用SpringCloud框架下的Hystrix组件实现断路器、线路隔离等保护功能；为了实现各个微小子服务之间的服务通讯，SpringCloud提供了两种通讯方式RestTemplate和Feign，两种通信方式都是基于网络传输，实现HTTP远程调用，功能相同；微服务中所有的Rest传输调用最终都是暴露在公网上，那么需要保证网络上的公司私人业务的安全性，防止信息泄露，本课题利用SpringCloud下的SpringSecurity实现网络传输安全。SpringCloud框架在架构技术层面众多服务要求提供了技术支持，不需要做出太大的改动，因此架构技术层面不是本课题研究的重点，将直接利用这些微服务组件提供支持

应用服务层还包括业务逻辑服务，是将装配制造执行系统任务管理的模块拆分成几个独立的子服务，分别部署在不同的物理机或者虚拟容器中。业务逻辑服务运用架构技术服务提供的整个微服务运行的环境，与数据存储层和用户交互，在业务逻辑层进行逻辑处理，然后将渲染的界面传递给用户，实现人机交互。业务逻辑层是整个装配制造执行系统的核心，为了实现SaaS模式对软件功能可配置的要求以及第四级成熟度模型对软件灵活性和可拓展的要求，需要将传统的装配系统中的业务逻辑拆分成独立的子服务。



图 8 微服务架构技术方案

综上所述，本课题就微服务技术架构方案实现的关键问题在于对业务逻辑服务层面的拆分，以及各个子服务的功能实现。然后在微服务实现的基础上配置用户的业务流程。接下来，本文将具体论述使用微服务的系统技术方案实现装配制造执行系统软件的SaaS化。

## 3.3 基于SaaS的微服务系统架构实现

### 3.3.1 微服务系统拆分（系统功能需求）

20世纪初，制造业从手工作坊开始向大规模生产进化，制造业开始提倡使用泰勒的科学管理理念生产制造。70年代中期市场竞争进一步加剧，制造业开始向大量个性化生产靠拢，要求企业具有柔性和应变能力。但是大部分装配制造企业都是根据本企业生产的产品对象设计专业化的生产流程和生产模型，根据装配形式方面的特征，我国的工程机械装配形式主要有两种[47]：移动装配形式与固定装配形式。同时根据装配现代化程度的不同，部分制造企业严格实现了MRPII

、ERP、MES等现代化管理方法，部分企业根据自身的资金以及生产现状实现的部分现代化管理方法。

用户在SaaS软件平台上定制服务的前提就是服务平台为用户提供一个应用模板，模板以标准的应用服务为基础实现。因此在实现装配制造系统SaaS时，需要考虑到所有企业用户的实际情况，设计一个模板装配制造执行系统，适用于所有企业。每个用户根据自己企业的装批生产流程和现代化程度定制相应的服务，并且保存配置信息。

根据制造企业现代化生产管理的科学方法，结合本课题对柔性装配制造执行系统的研究，对装配系统通用任务管理的功能设计如下图，将传统的装配系统任务管理服务拆分成四个服务：订单管理、主生产计划管理、生产指令管理和任务执行管理。

订单管理是一个企业的生产规划和产品系列大纲，用来协调满足经营规划所需求的产量与可用资源之间的差距，也是一个装配生产过程的开始。企业的订单管理是企业经营战略的具体体现，同时是主生产计划（Master Production Schedule , MPS）的依据。因此订单管理服务是整个装配系统任务管理的开端。订单管理的功能包括对预测和生产订单数据集的管理、设计订单模型

主生产计划是对企业生产订单的细化，用来协调生产需求和可用资源之间的差距，主生产计划是以企业的生产规划即订单为输入，安排详细的进度计划。主生产计划MPS在制造业中广泛应用，驱动整个生产和控制系统，因此主生产计划是装配系统任务管理服务中不可或缺的一环。

主生产计划知识最终产品的计划，但是一个产品可能由成百上千种相关物料组成，需要对主生产计划进行MRP运算得到详细的物料需求计划，也就是生成生产指令，每一个生产指令都是一个单独的物料需求。包括制造需求和采购需求。因此，生产指令也是其中装配任务管理服务中的一环。

最后就是任务执行管理服务，这是所有装配制造业生产流程的核心，车间生产作业控制的目标是通过对制造过程中任务的计划、执行和监控提高生产资源的利用率，达到提高生产率的目的。



图 9 装配系统任务管理功能图

本课题将装配系统任务管理服务拆分成订单管理、计划管理、生产指令管理和任务管理。不同的装配制造企业可以实现不同流程的配置。装配现代化程度较高的企业可以定制所有服务流程，装配系统流程如图10所示，从订单管理开始，订单管理员对订单数据集进行维护。计划员通过订单和计划的关联，将订单产品需求数据通过网络传输协议传送到计划管理服务，经过计划管理的MRP运算产生制造需求和物料需求。然后将生产指令下发到生产作业管理服务中，生产在制任务或者是和当前的在制任务进行关联，然后车间工人拿到任务进行生产，并且将任务的状态返回到上一级的服务，通知任务完成或者报废等信息。计划员根据任务反馈的状态实时反馈给订单层面，通知订单管理员订单的完成情况。

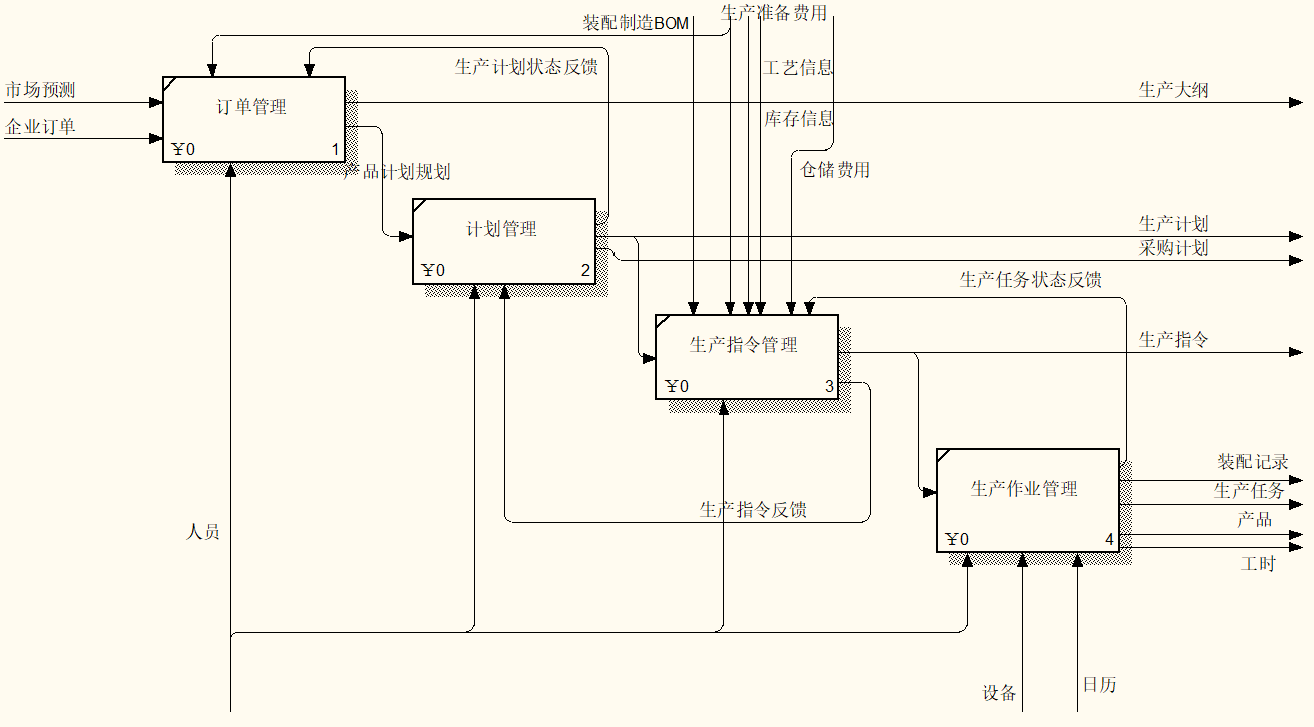


图 10 装配系统功能模型

### 3.3.2 微服务系统服务定制

在微服务系统架构设计的基础上，将传统的装配制造执行系统的任务管理按照业务流程逻辑拆分成四个子服务：订单管理、主生产计划管理、生产指令管理、生产作业管理。但是从装配制造企业的角度上看，只有部分企业完整使用这套科学的制造管理方法。在装配现代化较低的企业，企业的装配管理比较落后，企业的信息化生产过程是从计划层面开始，将订单层面的服务交给ERP或者其他专业系统来实现，在装配MES流程中只需要将生产计划作为输入进行生产。在装配现代化更低的企业，企业直接根据生产指令生产，之间将生产指令下达到车间，不考虑物料、库存等信息，工人根据任务信息直接生产，生产的状态信息直接反映企业的生产状况，不需要将车间的任务状态向上反馈。

因此，需要根据不同的企业级租户配置相应的信息。微服务的一个好处就将传统的复杂业务逻辑拆分成小服务，将服务与服务之间解耦。每一个服务都是一个一个可以独立运行子服务，所以可以从微服务的架构层面实现租户企业级功能的配置。微服务

企业租户在定制服务的基础上，管理员根据企业租户内部人员的职能和权限配置相应的界面，将界面的可配置功能直接交给企业管理，是SaaS软件服务的可配置功能更加灵活。

需要不同的业务流程

②将系统服务拆分

③为什么将系统拆分

④拆分出来的服务怎么实现多租户的可配置

⑤租户内的企业怎么实现界面配置

③通过人员权限配置整个模块服务的可配置

因此需要针对所有的组织、业务和功能维度

用户在SaaS软件平台上定制服务的前提就是服务平台为用户提供一个应用模板，模板以标准的应用服务为基础实现。模板中一部分应用为所有用户都需要的应用，另一部分应用则可以由用户根据自己的特殊需求定制。用户可以定制的服务可以称为可配置服务点

①从组织维度，不同的组织（车间、工段、班组等）需要不同的系统功能。

②从业务维度，不同的人员角色需要配置当前人员需要的系统功能。

③从功能维度，根据不同工厂装配自动化的实现程度，配置系统功能。

不同维度的数据统一模型。

①现在的各种制造企业的制造流程

②说明车间的角色权限

③从组织维度

④绘制总共的业务流程

采用微服务的系统设计方案，将柔性装配任务管理系统拆分出不同的服务模块，保证服务的可用性，服务之间数据的一致性（各服务之间相互调用）。

防止陷入微服务模式化陷阱。

根据当前制造企业的装配现状，计划拆分的服务：

订单任务管理服务

生产计划管理服务

生产指令管理服务

任务执行管理服务

拆分出不同的业务，从组织、业务和功能角度为定制服务的角色分配系统功能。

### 3.3.3 SaaS多租户管理

### 3.3.2.1 多租户数据模型设计（数据一致性、数据隔离性、数据传播、数据分组）

数据隔离性：将不同租户之间的数据分离开。

数据传播：实现不同租户之间协作装配。

数据分组：保证不同租户之间数据资源共享。

### 3.3.2.2 多租户协作设计

云平台功能设计，将不能制作的任务（采购任务）放到云平台上去，通过其他企业接收到任务的状态并且返回交货期，通过更改任务的开始时间和结束时间

### 3.3.2.3 界面可配置（通过调整界面的参数实现）

提供超级管理员：企业服务定制

和企业内部超级管理员：企业内部人员角色维护、权限设定、信息录入

权限对界面实现配置

## 本章小结

本章首先探讨了基于SaaS服务的柔性装配系统的任务管理的需求，根据SaaS模式的特性，提出了SaaS服务的体系架构，并且提出基于微服务框架的系统技术方案。然后介绍微服务各个模块的功能，并且根据微服务系统框架思路，结合柔性装配任务管理业务流程，将系统划分成各个系统服务模块。最后研究在微服务框架下的各个服务模块远程调用的方法。接下来将要介绍根据微服务方案拆分的各个服务的具体功能实现。

# 第四章 基于SaaS的装配任务管理系统功能开发

## 4.1 功能设计

用例图[48]主要从系统用户的角度考虑系统需要提供哪些功能和服务。根据前文分析装配制造执行系统可以分为订单管理服务、计划管理服务、任务管理服务和现场任务执行服务。分析各个服务的功能，确定系统角色，得到用例图中的参与者包括：订单管理员、计划员、任务管理员、操作工人。从而确定每个字服务的系统边界和范围。不同的参与者对应系统的不同业务需求。因此本章将使用用例图确定各个服务的功能边界和业务需求。

用例图是从服务模块的整体功能考虑系统的设计，没有结合系统的业务流程考虑。而状态图是对模型元素的动态行为进行建模，将系统的业务行为进行建模。UML状态图[49]是由类的实例对象的状态和这些状态的变化转移组成。因此本章将从订单、计划、任务对象建立活动图，通过对象状态变化的事件展示服务的业务流程。

最后将结合PSL过程规范语言产品对装配任务执行过程主要的过程对象建立详细的信息模型。并且结合SaaS模式对软件设计的要求对对象模型进行改进。

### 4.1.1 订单任务管理服务设计

4.1.1.1 功能模型

订单任务管理是整个装配任务管理系统的开始，订单使企业自身和客户之间的业务纽带。订单的管理信息有助于企业在制定计划、物流管理、资金管理做出最有效的决策。从用例图分析订单管理服务的基本功能，描述订单任务管理服务的功能模型。



图 11 订单管理员用例图

如图11所示，为订单任务管理模块分配了订单任务管理员的角色，订单管理员主要包括以下职能：

（1）订单数据维护：订单数据集的维护，包括录入制造企业预测的生产需求、从客户上接收到的订单，对下达的订单置取消、下达、完成等状态，还需要录入改订单的详细生产资料。

（2）订单安排：和生产计划数据集进行关联，提供从订单到生产计划的关联关系，可以直接从订单的信息下达生产计划，也可以将订单直接与现有的生产计划关联，通过生产计划的完成状态反馈到订单，通知订单管理员订单的完成状态。

（3）订单状态反馈：根据订单的完成状态，通知订单管理员发货、收货以及结单等操作。

（4）客户信息维护：订单的一个重要信息就是对客户的管理，将发来的订单按照客户分组管理，并且维护客户的信息。

4.1.1.2 过程模型



图 12 订单状态图

从用例图对订单管理服务的边界和功能进行了详细的设计，在订单管理服务中主要是对订单数据集的操作。在订单数据集的操作过程中，订单的状态会根据当前发生的事件以及下游的计划数据集对订单数据的反馈发生转变。因此，绘制了订单状态图，如图所示对订单服务管理里面的事件流程进行详细设计。如表2所示对状态图中的关键事件和状态的描述：

|  |  |
| --- | --- |
| 状态 | 描述 |
| 草稿 | 接收客户订单或者创建新订单。 |
| 已确认 | 对订单进行分析，确定最终装配计划。 |
| 已下达 | 将订单和生产计划关联，将订单的需求反映到生产计划上。 |
| 生产中 | 生产计划开始，反馈到订单开始生产，为生产中订单。 |
| 已中断 | 订单管理员手动置订单状态为已中断，暂停订单需求的生产装配。 |
| 已暂停 | 订单关联的生产计划因为生产能力不足导致生产暂停，置订单为已暂停状态。 |
| 已取消 | 取消订单。 |
| 已入库 | 将已完成的订单，将满足订单的库存打上标识，此时订单状态为已入库。 |
| 已交付 | 将满足订单的库存发给客户，此时订单的状态为已交付 |

表 2 订单状态图

4.1.1.3 信息模型

通过对订单管理服务的功能、流程和事件进行分析，对订单管理服务的功能模型和过程模型做了详细的描述。在订单管理服务中涉及到对订单模型的操作，同时订单模型和计划模型之间需要关联，因此利用PSL规范语言对订单和计划之间的关系进行表达，设计订单管理服务的信息模型：

（1）订单（Order）、生产计划（MPS）是最基本的组成部分，对应于四个原始类中的Activity类。

（2）描述需求的物料（Part）信息、物料清单（BOM）、客户（Customer）信息对应于四个原始类中的Object类

（3）订单和生产计划的对象链接关系，对应于PSL和心中原始类的参与（Participates-in）关系。

（4）订单中包含订单明细对需求的详细描述，他们之间存在聚合关系，这种关系对应于PSL扩展中的子活动关系（Subactivity）。

根据以上结合PSL对订单管理服务中的对象描述，同时为了满足多租户数据模型设计，为每一张表中添加了companyID字段标识每一个租户的数据，建立如图13的关系模型:



图 13 订单服务管理信息模型

### 4.1.2 生产计划管理服务设计

4.1.2.1 功能模型

计划管理服务可以作为订单任务管理的下游，同时也可以作为整个装配过程的开始。计划是生产任务中的关键环节，用来协调生产需求与可用资源之间的差距，充分利用企业资源，协调生产与市场。生产计划管理员负责接收上游订单生成的主生产计划或者主动创建主生产计划，并且维护生产计划数据集。计划管理服务的核心就是对物料需求计划的实现，将产品级别的主生产计划分解出相关需求计划。首先从用例图分析生产计划管理服务的基本功能，描述订单任务管理服务的功能模型。为操作生产计划管理服务的人员分配角色和权限。



图 14 计划管理员用例图

如上图所示，为计划管理模块分配了计划管理员的角色，计划管理员主要包括以下职能：

（1）生产计划的维护：主要包括生产计划的新建、删除、更新，同时还要管理从订单服务的生产计划。管理生产计划和订单之间的关联关系，将生产计划的状态反馈给订单。同时作为SaaS服务的一个可配置服务点，计划管理服务可以直接作为整个装配系统的开端，不需要接收到来自订单的生产计划。企业可以自己通过下达生产计划，然后根据生产计划的状态监控整个生产过程。其次，没有配置订单服务的企业，生产计划的状态也就不会向上继续反馈。

（2）物料需求计划：物料需求计划是整个计划管理服务的核心，因为在生产计划层面，企业下达的是产品级的生产计划，但是一个产品如飞机可能由成百上千的零部件组成，因此需要通过物料需求计划来通过BOM和库存信息实现产品需求的分解，得到制造需求和采购需求。同时还要考虑到企业对于物料需求计划的多种使用模式，包括清除运算结果，多个产品需求同时计算的操作等。关于物料需求计划的算法将在后文详细设计。

（3）生产计划安排：主要是对生产计划和在制任务数据集之间的操作。包括直接根据生产计划对任务投产产生在制。也可以直接和在制任务关联，通过在制任务来交付该生产计划。因为部分企业的现代化生产水平较为落后，可能没有生产计划这一层面的服务，对于物料需求计划都是通过计划员的纸张作业来完成，然后根据计算出来的制造需求再在装配系统中下达任务去执行。这样设计可以满足这样的制造企业的要求，直接下达在制任务。

（4）计划审核：计划员需要时时关注当前计划的执行情况，在审核过程中可能出现意外情况：与分配好的库存因为紧急事件被领走了等等，会出现计划无法完成的情况，这个时候需要计划员通过计划审核功能调整物料需求计划计算出来的结果，或者是补投计划等。

（5）计划跟踪：以产品级计划BOM的形式展开提供计划员审查，同时可以根据车间的反馈状况和库房的物料信息修改计划关联的库房库存和在制任务。

4.1.2.2 过程模型

通过用例图设计了计划管理服务的功能模型，为计划管理服务设计了计划管理员角色，主要负责和计划数据集的交互。在对计划数据集操作的过程中，计划的状态随着计划员操作、上游订单数据的关联和下游任务数据的反馈等事件发生转变，通过这些事件可以对计划管理服务的流程进行详细的设计。因此对计划模型绘制了状态机表征这些事件和状态，从状态图描述服务的过程模型，如图15所示。



图 15 生产计划状态图

生产计划状态图以生产计划为研究对象，描述了计划对象的生命周期，描绘计划的状态及引起计划状态转换的事件，表示订单管理服务的行为，下表对这些行为及事件做了完整的描述，如表3所示。

|  |  |
| --- | --- |
| 状态 | 描述 |
| 草稿 | 对于新建的生产计划为草稿状态。 |
| 已分解 | 生产计划经过MRP运算之后为已分解状态。 |
| 已下达 | 将独立需求的生产计划下达或者将独立需求的相关需求下达，那么该独立需求的生产计划状态为已下达。 |
| 生产中 | 已下达的生产计划关联在制任务之后，或者根据生产计划下达在制任务，生产计划为生产中的状态。 |
| 已暂停 | 计划员根据调整人工暂停当前生产计划，生产计划为已暂停。 |
| 已中断 | 由于生产能力的不足，生产计划关联的在制任务被迫停止，生产计划为已中断的状态。 |
| 已取消 | 将计划标为取消状态，表示生产计划作废。 |
| 可交付 | 生产计划独立需求关联的所有在制任务都已完成，那么此时的计划状态是已交付状态。 |
| 已交付 | 生产计划的独立需求全部完工交检出库，是生产计划最终完工的节点。 |

表 3 生产计划状态描述

4.1.1.3 信息模型

通过对计划管理服务的功能、流程和事件进行分析，对计划管理服务的功能模型和过程模型做了详细的描述。在计划管理服务中涉及到对计划模型的操作，同时订单数据集和计划数据集、计划数据集和任务数据集之间存在关联，因此利用PSL规范语言对订单和计划之间的关系进行表达，设计订单管理服务的信息模型：

（1）订单明细（Order）、生产计划（MPS）是最基本的组成部分，对应于四个原始类中的Activity类。

（2）描述需求的物料（Part）信息、物料清单（BOM）、库存信息（Warehouse）对应于四个原始类中的Object类。

（3）计划（Demand）和计划（Demand）的对象链接关系、生产计划（Demand）和在制任务的对象连接关系，对应于PSL和心中原始类的参与（Participates-in）关系。

（4）主生产计划（MPS）和相关需求计划（Demand），他们之间存在聚合关系，这种关系对应于PSL扩展中的子活动关系（Subactivity）。

（5）任务（PartTask）作为活动的实例，在执行过程中包含了活动的时间信息，对应于PSL中四个原始类之一的Timepoint(时间点)类。

（6）工艺（Routing）与任务（PartTask）在装配过程中是定义与实例的关系，对应于PSL中Activity类与Activity-Occurrence(活动实例)类之间的Occurrence-of(活动发生)关系。

根据以上结合PSL对订单管理服务中的对象描述，同时为了满足多租户数据模型设计，为每一张表中添加了companyID字段标识每一个租户的数据，建立如下图的关系模型:



图 16 计划管理服务信息模型

### 4.1.3 在制任务管理服务设计

4.1.3.1 功能模型

在制任务管理服务可以作为计划任务管理的下游，同时也可以作为整个装配过程的开始。任务是整个生产管理的核心，也是现场工人实际操作的对象。目前制造业偏向于制造资源和制造能力共享，这也是当前SaaS软件对多租户资源共享的要求。因此在设计在制任务管理的功能模型时，需要考虑到企业间任务的流转和生产。同时协作任务不仅在企业间需要考虑，同时需要考虑在制任务在一个企业内部不同工厂间外协加工。从用例图分析在制任务管理的基本功能，描述任务管理的功能模型。为在制任务计划管理服务的操作人员分配角色和权限。



图 17 任务管理员用例图

如图17所示，在制任务管理模块分配了任务管理员的角色，任务管理员主要包括以下职能：

（1）车间任务管理：需要一批中的零件任务打上标识，用来跟踪零件的生产装配过程。将零件任务根据工艺分解成工序任务，然后将工序任务分派给现场的车间工人加工装配。还需要维护任务数据集，提供直接下达任务的操作，可以不经过生产计划和指令的关联，适用于生产装配现代化落后的装配企业，企业直接下达装配任务生产。

（2）协作任务管理：提供外协任务，可以将任务发给同企业的其他车间生产，也可以接受从其他车间发来的协作任务。同时可以对任务处理，完工出入库等操作。

（3）委外任务管理：和协作任务管理类似，不同的是委外任务针对的是不同的生产企业，协作任务针对的是同一生产企业内部的不同生产车间。企业通过委外任务可以实现多企业联合生产，共享生产资源和功能。

4.1.3.2 过程模型

通过用例图设计了在制任务管理服务的功能模型，为任务管理服务设计了计划管理员角色，主要负责和任务数据集的交互。在对任务数据集操作的过程中，任务的状态随着任务管理员操作、上游计划数据和下游工人数据的反馈等事件发生转变，通过这些事件可以对计划管理服务的流程进行详细的设计。因此对计划模型绘制了状态机表征这些事件和状态，从状态图描述服务的过程模型，如图18所示。



图 18 零件任务状态图

零件任务状态图以零件任务为研究对象，描述了零件任务对象的生命周期，描绘零件任务的状态及引起任务状态转换的事件，表示在制任务管理服务的行为，下表对这些行为及事件做了完整的描述，如表4所示。

|  |  |
| --- | --- |
| 状态 | 描述 |
| 草稿 | 新建的零件任务或者从计划管理中生成的在制任务为草稿状态。 |
| 已关联 | 与计划关联之后的在制任务状态为已关联状态。 |
| 已分解 | 将零件任务按照工艺分解成工序任务，在制任务状态为已分解。 |
| 已齐套 | 对零件任务齐套分析，可以齐套的零件任务为已齐套的状态。 |
| 未齐套 | 对零件任务齐套分析，不能齐套的零件任务为已齐套的状态。 |
| 已分派 | 对零件任务分解产生的工序任务分派给现场工人，零件任务状态为已分派。 |
| 已开始 | 分派给工人的工序任务开始加工，对应的零件任务状态为已开始。 |
| 已中断 | 任务管理员手动悬挂任务，任务状态为已中断。 |
| 已暂停 | 任务由于生产能力不足导致任务暂停，任务状态为已暂停。 |
| 已完成 | 所有的任务需求完成，需求数等于计划数，任务状态为已完成。 |
| 已交检 | 将已经完成的任务交给检验处检验，任务状态为已交检。 |
| 完工入库 | 检验完成，将加工完成的产品入库，任务结束。任务状态为完工入库。 |

表 4 任务状态描述表

4.1.3.3 信息模型

通过对在制任务管理服务的功能、流程和事件进行分析，对任务管理服务的功能模型和过程模型做了详细的描述。在任务管理服务中涉及到对任务模型的操作，同时计划数据集和零件任务数据集、零件任务数据集和工序任务数据集之间存在关联，因此利用PSL规范语言对订单管理服务的信息模型进行设计：

（1）描述需求的物料（Part）信息、人员（Employee）信息对应于四个原始类中的Object类。

（2）计划（Demand）和计划关系（DemandLink）的对象链接关系、生产计划（Demand）和在制任务（PartTask）的对象连接关系，对应于PSL和心中原始类的参与（Participates-in）关系。

（3）主生产计划（Demand）和相关需求计划（Demand），他们之间存在聚合关系，这种关系对应于PSL扩展中的子活动关系（Subactivity）。

（4）零件任务（PartTask）和工序任务（OperationTask）作为活动的实例，在执行过程中包含了活动的时间信息，对应于PSL中四个原始类之一的Timepoint(时间点)类。

（5）工艺（Routing）与任务（PartTask）在装配过程中是定义与实例的关系，对应于PSL中Activity类与Activity-Occurrence(活动实例)类之间的Occurrence-of(活动发生)关系。

根据以上结合PSL对订单管理服务中的对象描述，同时为了满足多租户数据模型设计，为每一张表中添加了companyID字段标识每一个租户的数据，建立如下图的关系模型:



### 4.1.4 任务执行管理服务设计

4.1.4.1 功能模型

任务执行管理服务是专门为现场工人加工装配设计的服务，旨在加快装配过程无纸化的进程，传统的离散型手工装配在装配过程中需要填写大量的纸质装饰记录，企业需要大量的库存去存放这些装饰记录，一旦需要对某件产品的装配过程进行追溯，那么会非常的困难。同时工人在车间现场装配时，需要花费大量时间填写装饰记录，严重影响生产效率。因此通过对现场工人的用例图分析任务执行管理服务功能模型。



图 19 操作员用例图

如图19所示，为在任务执行模块分配了操作员的角色，操作员主要包括以下职能：

（1）领料：领取当前任务所需要的物料。

（2）填写装饰记录：根据工序中对装配的要求，填写装饰记录。

（3）装配作业：这是现场工人的操作核心，需要通过装配制造系统开始和完成装配。

4.1.4.2 过程模型

通过现场操作员的用例图分析了现场操作人员的职能，然后对现场工序任务的状态分析，绘制现场执行的工序任务状态图



图 20 工序任务状态图

工序任务状态图以工序任务为研究对象，描述了工序任务对象的生命周期、工序任务的状态及引起任务状态转换的事件，表示工序任务管理服务的行为，下表对这些行为及事件做了完整的描述，如表5所示。

|  |  |
| --- | --- |
| 状态 | 描述 |
| 已创建 | 零件任务经过工序分解产生工序任务，此时为已创建状态。 |
| 已分派 | 将分解产生的工序任务分派给车间现场工人。 |
| 已开始 | 操作工人开始装配，工序任务为已开始的状态。 |
| 已完成 | 操作工人完成装配，工序任务为已完成的状态。 |

表 5 工序任务状态描述表

4.1.4.3 信息模型

现场任务执行管理服务主要是对无纸化的一个实现，操作工人针对工序任务数据集进行维护，实时记录任务的完成情况，并且向上游服务反馈数据进度，使每一个服务的管理员都能监控，因此利用PSL规范语言对订单管理服务的信息模型进行设计：

（1）描述需求的物料（Part）信息、人员（Employee）信息对应于四个原始类中的Object类。

（2）计划（Demand）和计划关系（DemandLink）的对象链接关系、生产计划（Demand）和在制任务（PartTask）的对象连接关系，对应于PSL和心中原始类的参与（Participates-in）关系。

（3）主生产计划（Demand）和相关需求计划（Demand），他们之间存在聚合关系，这种关系对应于PSL扩展中的子活动关系（Subactivity）。

（4）零件任务（PartTask）和工序任务（OperationTask）作为活动的实例，在执行过程中包含了活动的时间信息，对应于PSL中四个原始类之一的Timepoint(时间点)类。

（5）工艺（Routing）与任务（PartTask）在装配过程中是定义与实例的关系，对应于PSL中Activity类与Activity-Occurrence(活动实例)类之间的Occurrence-of(活动发生)关系。

根据以上结合PSL对订单管理服务中的对象描述，同时为了满足多租户数据模型设计，为每一张表中添加了companyID字段标识每一个租户的数据，建立如下图的关系模型:

## 4.3 基于SaaS模式的MRP算法优化

### 4.3.1 MRP理论研究

在制造业的生产经营中，一方面为了生产的连续进行，需要对生产过程中的在制品、半成品以及采购的原材料、毛坯、零部件进行仓储，满足市场的需求。另一方面，在仓储的过程中，库存占用了大量的资金，为了加快制造企业的资金周转，需要降低库存，因此在企业生产中就会产生库存和生产需要之间的矛盾。为了解决这种矛盾，物料需求计划（Material Requirement Planning，MRP）应运而生，MRP是一种精确的生产计划系统，同时也是有效的物料控制计划。MRP的目的就是为了在满足物料需要的前提下，是库存水平保持在最低值，协调库存和生产需要之间的矛盾。（装配需要使用到MRP算法）复杂产品的装配过程一般由总装、部装（大部装、小部装）、组装、零件装配过程组成，那么在对复杂产品装配过程中会占用大量的库存资源用来仓储装配过程中需要用到的原材料、零部件以及生产中的在制品和半成品。因此为了协调装配过程中的仓储和生产之间的矛盾，本文将研究并实现MRP算法。

MRP算法于20世纪60年代兴起于美国，发展到现在已经较为成熟，国内外学者对于MRP算法的研究与具体实现都已经发展的比较完备。柯耀杰、韦拥欧等人[50]针对汽车装配制造企业，分析了汽车在装配过程中物料需求计划制定和实施过程中存在的问题，从信息系统和管理流程和降低成本方面进行研究，优化了汽车装配过程中的物料需求计划。张迪、党少杰等人[51]在JIT思想的基础上，提出了基于生产节拍的物料需求计划的计算方法，从Job Shop作业车间的物料计划优化入手，确定配送物料的物料需求计划。Plenert, Gerhard等人[52]着眼于当前MRP算法对企业生产制造的影响，从JIT、优化生产技术（OPT）、约束理论（TOC）和瓶颈分配方法（BAM）等方面优化MRP算法，并寻找可以替代MRP的方案。

制造企业内部的物料需求[53]可以分为独立需求和相关需求两种类型。独立需求由企业外部决定，企业生产一般可以分为备货生产和订单生产。因此企业可以根据已经接到的订单以及企业内部对产品需求的预测，得到对物料的独立需求。而相关需求可以根据独立需求的产品结构组成关系得到。

MRP的基本原理[54]就是根据独立需求的产品结构各层次物料的从属和数量关系，以企业对产品需求的预测和订单为计划对象，以产品的交货期为时间基准倒排计划，推算出制造装配所需物料的准确时间和数量，这是一种优先计划的方法。物料需求计划MRP的原理如图所示。MRP有三个输入：主生产计划（Master Production Schedule, MPS）、物料清单（Bill of Material, BOM）和库存状态。主生产计划MPS是针对企业生产对象的生产计划，包括生产数量和交货期。生产数量是已经下达的订单和市场预测得到的产品需求。物料清单BOM表示产品的组成机构，也是MRP分解必须的因素。库存状态表示企业仓库中零部件及产品的物料数量。经过MRP计算之后可以得到物料的采购计划和自制组件的加工计划。



图 21 MRP逻辑关系

### 4.3.2 算法详细设计

目前MRP算法在计算机辅助生产管理软件上的实现，多采用国际上比较通用的MRPII的思想：先对BOM表进行分解，确定各物料在BOM中的层级关系，不同层级零部件的层次以最底层层数为基准，如图所示的眼镜的BOM树，螺钉分别在第1层和2层中出现，那么螺钉的层次就为2，然后搜索分解表，访问主生产计划中该物料的需求数量和交货期，根据分解表和库存信息逐层推算出自制件的加工计划和采购件的采购计划。



图 22眼镜的BOM树

但是传统的MRP算法并没有在SaaS场景下使用的经验，当前的制造型企业内部的物料需求计划是针对当前企业内部独立制造生产设定的，没有考虑到多企业共享制造资源和制造能力的情况。因此为了使SaaS环境下的多企业之间制造资源和制造能力的共享协作，需要对传统的物料需求计划MRP进行调整。同时，当前的MRP计算都是针对制造企业内部的制造业务流程设计的，但是不同企业的制造业务流程和装配能力的现代化程度都不尽相同，部分制造企业没有完全实现I-MES的标准业务流程

③列出需求点，分点去论述（图），最后通过流程图和公式设计

③在SaaS应用模式下的对MRP提出的新要求（能够适应不同装配模式的要求）（为了适应多租户，可以查找到另一个用户的数据，主要是库存的共享，对库存进行分析，不要和供应链扯上关系）。

①流程图

②计算公式

### 4.3.3 不同模块下算法的适应性（参数配置）（物料的库存在不同的地方属于供应链之间的问题）

### 4.3.4 数据一致性

## 本章小结

本章针对微服务拆分出来的订单任务管理服务、生产计划管理服务、生产指令管理服务以及现场任务执行服务，结合SaaS模式对应用软件的要求，分别从功能模型和信息模型进行了详细的设计，然后对整个装配过程的过程模型进行了描述。最后对传统的MRP算法进行了改进，是MRP算法跟能适应SaaS软件下多租户的要求。下一章将对整个系统的实现进行验证。

# 第五章 系统实现与验证

## 5.1 系统开发环境与技术

### 5.1.1 系统开发环境

* Windows7、8、10 32/64位操作系统
* Java语言开发工具包(JDK1.8+)
* Eclipse J2EE版本+Dorado7插件
* Google浏览器、360浏览器或者使用谷歌内核浏览器
* Oracle11g数据库
* SpringCloud微服务框架
* Tomcat服务器
* Maven仓库
* Git代码版本工具

上述这些开发技术核心依托于SpringCloud开源框架。在底层数据库层面，借助spring+hibernate数据库ORM持久化框架，搭建数据库表记录和程序代码实体间的映射。通过hql数据操作语言实现对数据库记录的增删改查等操作。在展现层，基于Dorado7的eclipse插件，对用户界面进行开发，同时引入H5、JavaScript和JSP等前端网页语言与用户操作交互。对于图标的显示，引入了基于Jquery的Enterprise-Charts插件进行数据的可视化。利用tomcat服务器作为前端网页的容器，提供处理浏览器请求的web服务。利用Maven仓库管理整套系统的jar包构建，通过git版本控制工具实现代码的迭代。整个系统的技术架构如图所示。



图 23 系统机构图

### 5.1.2 系统运行环境

（1）客户端

* 操作系统没有限制
* Google浏览器、360浏览器或者使用谷歌内核浏览器

（2）服务端

基于SaaS应用环境用户只需要客户端进行远程访问。

SaaS服务提供者系统运行环境：

* 前台服务服务器：
* apache-tomcat-8.0.26 服务容器
* Java语言开发工具包(JDK1.8+)
* 后台微服务服务器：
* Java语言开发工具包(JDK1.8+)
* 数据库服务器：
* Oracle11g数据库
* Redis缓存

## 5.2 系统架构与关键模块实现

### 5.2.1 SaaS服务订制化

根据各租户装配流程和装配现代化的实现程度定制以下服务。

主要实现以下几点：1.服务之间数据的传输。2.根据租户的装配流程实现从订单🡪计划🡪生产指令🡪生产任务🡪反馈的流程，也可以自定义需要的服务，比如部分制造业企业是直接下达生产计划，那么可以从计划🡪生产指令🡪生产任务🡪反馈的流程。某些制造企业的装配现代化程度较低，那么可以直接配置从下达生产指令或者生产任务开始，来配置整个服务。因此可以实现从装配服务的模块化到制造企业装配MES的实例化。3.根据各个用户之间的协作关系，管理多租户的制造数据，并且实现各个租户定制的柔性装配制造执行系统之间业务流转，制造外协。

### 5.2.1.1 订单管理

①订单数据集管理

②订单和计划数据集关联

### 5.2.1.2 计划管理

①计划数据集管理

②物料需求计划MRP改进

③计划和生产指令数据集关联

### 5.2.1.3 在制任务管理

①生产指令数据集管理

②生产指令和生产任务数据集关联

③生产指令层级关联

### 5.2.1.4 任务执行管理

①生产任务数据集管理（AO，FO）

②生产任务完工反馈管理

### 5.2.2 SaaS多租户功能

①多租户独立生产

②多租户协作

## 5.3 实例展示(主要是界面展示，整个业务流程)

①SaaS化多租户应用展示

②SaaS化定制服务展示

③业务功能展示（从下达订单到任务完工反馈整个流程的展示）

④用户界面展示

## 本章小结

# 第六章 总结与展望

## 6.1 工作总结

## 6.2 工作展望

# 致谢

光阴似箭，日月如梭。转眼间就到了即将毕业的时间，在论文完成之际，对三年来帮助我的人表达由衷的谢意。

首先，要衷心的感谢我的导师杨建军教授在研究学习中对我的谆谆教导，本论文实在杨老师的细心指导下完成的。在研究生期间，杨老师学识渊博、治学严谨、良好的专业素养和高效的做事风格，将使我终生受益并铭记于心。

其次，对机械工程及自动化学院工业与制造系统工程辛勤付出的各位老师，表达深深的敬意与感谢。正是各位老师不遗余力地帮助我们解决在学习科研和生活中遇到的各种问题，才能使我们在学术氛围浓厚的学习科研环境中不断进步。感谢你们孜孜不倦的教学态度和辛勤工作的敬业精神。

然后，我要感谢我所在的实验室的各位兄弟们。感谢邓必超、周勇、韩宝安、黄壮、刘欣等师兄在软件设计和科研问题上的热心指导，让我学到了很多课本上学不到的知识，不仅帮助我提高了技术水平，还提高了我的专业技能。同时，我要感谢我的同窗刘晨硕、温春生，他们是我学习和生活中的强大助力，遇到困难时我们相互探讨相互鼓励，一起解决问题度过难关。感谢刘诚悦、张文昊、闫成龙、冉小川和孔祥龙等师弟，在我的课题完成中提供了莫大的帮助。

借此机会，我还要感谢一直陪伴在我身边的家人。感谢我的父母对我的支持，他们是我坚实的后盾，在每一个难熬的日子，给我鼓励，给我动力，是我在科研路上能够奋发向前。

最后，感谢各位评审老师，在百忙之中评阅我的论文，并提出宝贵意见，谢谢！

# 参考文献

**校对报告**

当前使用的样式是 [北京航空航天大学学报]

当前文档包含的题录共56条

有0条题录存在必填字段内容缺失的问题

所有题录的数据正常

**参考文献**

[1] 邢蕊. 基于SaaS的天津市制造业信息化平台的研究[D]. 天津理工大学, 2012.

[2] 吴坤. 物联网环境下的整车装配过程运行管理方法及应用研究[D]. 合肥工业大学, 2015.

[3] 张勤学，杨建军. 面向复杂产品的装配过程可视化管控系统[J]. 成组技术与生产现代化. 2016(03): 18-24.

[4] 钱芳，扈静，葛茂根，等. 面向机械产品装配过程的物料配送方法研究[J]. 机械工程师. 2011(05): 34-37.

[5] 傅玉颖，潘晓弘. 模糊不确定下多物料库存控制与生产批量优化[J]. 浙江大学学报(工学版). 2008(06): 1046-1050.

[6] 刘检华，林晓青，刘金山，等. 基于工作流的装配车间生产过程计划和控制技术[J]. 计算机集成制造系统. 2010(04): 755-762.

[7] Limère V. To kit or not to kit: optimizing part feeding in the automotive assembly industry[J]. 4OR. 2013, 11(1): 97-98.

[8] 常智勇，赵杰，莫蓉. 复杂产品装配执行过程数字化技术[J]. 南京航空航天大学学报. 2009(05): 564-569.

[9] 刘炜，刘峰，倪阳咏，等. 航天复杂产品智能化装配技术应用研究[J]. 宇航总体技术. 2018(01): 33-36.

[10] Hui C, Yuan L, Zhang K F. Efficient method of assembly sequence planning based on GAAA and optimizing by assembly path feedback for complex product[J]. International Journal of Advanced Manufacturing Technology. 2009, 42(11-12): 1205.

[11] Liu M Z, Wang J L, Mao-Gen G E, et al. Research on abnormal control strategy oriented to complex product assembly process[J]. Journal of Hefei University of Technology. 2011.

[12] 杨浩，朱剑英. 基于多Agent的分布式制造执行系统的建模[J]. 中国机械工程. 2004(11): 33-37.

[13] 张书亭，杨建军，邹学礼. 面向敏捷制造车间的制造执行系统研究[J]. 新技术新工艺. 2000(12): 2-4.

[14] 周华，杨建军，邓家盚. 基于全能体的MES构建[J]. 制造业自动化. 2001(02): 13-16.

[15] 周国利. 基于订单装配的制造执行系统（MES）在N公司的应用[D]. 湖南工业大学, 2013.

[16] Cheng F, Shen E, Deng J, et al. Development of a distributed object-oriented system framework for the computer-integrated manufacturing execution system: Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)

Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)[Z]. 19982116-2121.

[17] 王炳刚，周伟，饶运清，等. 装配流程可配置的制造执行系统的研究与应用[J]. 现代制造工程. 2007(11): 8-12.

[18] 沈晓杰，李郡. 基于制造执行系统的统计过程控制在质量管理上的应用[J]. 工业控制计算机. 2012(09): 108-109.

[19] 吴锋，马里. 面向MES的装配作业调度系统研究[J]. 杭州电子科技大学学报. 2010(04): 72-76.

[20] Gandhi P. A Manufacturing Execution System using Siemens' PC Based Automation Technology[J]. 2003.

[21] Addition I. Modeling of RFID-Enabled Real-Time Manufacturing Execution System in Mixed-Model Assembly Lines[J]. Mathematical Problems in Engineering,2015,(2015-1-14). 2015, 2015(1): 1-15.

[22] Jagdale K S, Patil S A, Parchandekar S K. A Smart Manufacturing Execution System[J].

[23] Demirkan H, Cheng H, Bandyopadhyay S. Coordination Strategies in an SaaS Supply Chain[J]. Journal of Management Information Systems. 2010, 26(4): 119-143.

[24] Bezemer C P, Zaidman A. Multi-tenant SaaS applications:maintenance dream or nightmare?[C]. 2010.

[25] 刘士军，张勇，杨成伟. 基于SaaS服务的中小企业业务协同系统[J]. 东南大学学报(自然科学版). 2011(03): 458-462.

[26] 向坚持，陈晓红. SaaS模式的中小企业客户关系管理研究[J]. 计算机工程与应用. 2009(19): 232-235.

[27] 李卫，张云勇，郭志斌，等. 电信运营商SaaS业务发展研究[J]. 电信科学. 2012(01): 132-136.

[28] 范卫锋，吕锋，贾现召，等. 基于SaaS模式的大型装备制造业工装信息系统研究[J]. 矿山机械. 2014(08): 117-120.

[29] 赵立君，范晓晖. SaaS技术的发展和演进[J]. 现代电信科技. 2007(12): 46-48.

[30] 李新明，廖貅武，刘洋. 基于SaaS模式的服务供应链协调研究[J]. 中国管理科学. 2013(02): 98-106.

[31] 霍小军. SaaS模式在电子政务中的应用初探[J]. 电子政务. 2008(01): 73-87.

[32] 刘蒲. 基于SaaS模式下中小企业信息化建设研究[D]. 延边大学, 2014.

[33] 赵立君，范晓晖. SaaS技术的发展和演进[J]. 现代电信科技. 2007(12): 46-48.

[34] Kwok T, Nguyen T, Lam L. A Software as a Service with Multi-tenancy Support for an Electronic Contract Management Application[C]. IEEE, 2008.

[35] 陈强. 基于SaaS模式的题库平台研究[D]. 首都经济贸易大学, 2017.

[36] 袁玉宇，胡文博. 基于IaaS云计算平台的弹性计费模型[J]. 中兴通讯技术. 2012(06): 34-37.

[37] 叶利娜. 基于SaaS模式的工艺设计服务研究[D]. 浙江大学, 2010.

[38] Mietzner R, Leymann F. Generation of BPEL Customization Processes for SaaS Applications from Variability Descriptors[C]. 2008.

[39] 吕海洋，杨建军. 基于PSL的制造过程信息模型研究与应用[J]. 制造业自动化. 2010(04): 14-17.

[40] 谭一鸣. 基于微服务架构的平台化服务框架的设计与实现[D]. 北京交通大学, 2017.

[41] 郭栋，王伟，曾国荪. 一种基于微服务架构的新型云件PaaS平台[J]. 信息网络安全. 2015(11): 15-20.

[42] 孟永胜，曹海萍. 基于微服务架构扩展软件功能的方法[J]. 电脑知识与技术. 2018(23): 105-106.

[43] 彭雪涛，梁卓明，赵金媖. 基于微服务的中小学智慧校园云平台设计[J]. 教育信息技术. 2018(09): 9-13.

[44] Kang H, Le M, Tao S. Container and Microservice Driven Design for Cloud Infrastructure DevOps[C]. 2016.

[45] Khanda K, Salikhov D, Gusmanov K, et al. Microservice-Based IoT for Smart Buildings[C]. 2017.

[46] 喻德禄，郭世龙，丁树晶. 中小银行基于微服务架构的应用服务体系研究与实践[J]. 中国金融电脑. 2018(03): 30-33.

[47] 王广信. 论述工程机械装配现状与发展趋势[J]. 内燃机与配件. 2018(9): 188-189.

[48] 肖海蓉. 基于统一建模语言的软件开发实例[J]. 计算机系统应用. 2013(07): 141-143.

[49] 李小亮. 时序扩展的UML状态图的测试用例生成研究[D]. 西南大学, 2014.

[50] 柯耀杰，韦拥欧. 汽车企业物料需求计划优化控制研究[J]. 企业科技与发展. 2013(14): 25-27.

[51] 张迪. 基于节拍预测的Job Shop作业车间物料需求计划研究[D]. 合肥工业大学, 2017.

[52] Plenert G. Focusing material requirements planning (MRP) towards performance[J]. European Journal of Operational Research. 1999, 119(1): 91-99.

[53] 洪跃山，胡燕. 物料需求计划的实现方法[J]. 电脑与信息技术. 2002(01): 50-52.

[54] 程控革扬. Mrp Ⅱ/ERP原理与应用[M]. 清华大学出版社, 2006.