****

**研究生开题报告**

**题 目：基于web的增材制造预处理平台的设计与实现**

**学 号 M201973007**

**姓 名 张鹏**

**专 业 计算机技术**

**指 导 教 师 李国宽**

**院（系、所） 武汉光电国家研究中心**

**华中科技大学研究生院制**

**填表注意事项**

一、本表适用于攻读硕士学位研究生选题报告、学术报告，攻读博士学位研究生文献综述、选题报告、论文中期进展报告、学术报告等。

二、以上各报告内容及要求由相关院（系、所）做具体要求。

三、以上各报告均须存入研究生个人学籍档案。

四、本表填写要求文句通顺、内容明确、字迹工整。

# 背景、目的和意义

## 课题背景

制造业是事关国民经济命脉的主要支柱产业，代表了国家的科技发展与综合实力水平。于2015年两会上初次提出“中国制造2025”的重要战略规划与目标，新的制造业革命将一触即发，而物联网、云制造、人工智能等前沿潮流信息技术无疑是推动制造业向智能化稳步发展的重要引擎。

虽然当今中国已然位列制造大国之首，但距离成为制造强国仍存在不小的差距。总的来说，我国制造业体系是从工业 2.0 时代建立而来的，随着时代的发展，传统模式下生产效率低下、智能化程度低的弊端日益凸显。“中国制造 2025”中重点强调，要将我国制造业从“制造”向“智造”的方向推进[1]，从而实现面向服务的智能制造[2]，必须充分融合现代信息技术与制造技术，开发具有智能感、主动学习、智慧决策、自动执行等功能的智能设备，从而提高制造业智能化水平。

增材制造，俗称“3D打印”，或快速原型制造（Rapid Prototyping），实体自由制造（Solid Free-form Fabrication），采用分层加工，迭加成型的方式逐层累加材料的方法制造实体零件。3D打印技术被认为将会为个性化产品的设计及生产带来革新，作为快速成型领域的新兴技术，3D打印技术以数字建模文件为基础制造几乎任意形状的三维实体，而不像传统的机械加工技术通过切削或钻孔等工艺或模具完成制造过程，不但能够缩短产品的研制周期从而提高生产率和降低成本，而且在材料耗费、环境保护等方面也有益处。

目前增材制造相关软件都是基于C或C++等强类型的编译语言开发而成，具备处理大量图形数据的能力，而且对于OpenGL有着良好的支持，所以在整个打印流程中会使用到很多本地客户端软件。但是这样对于初学者不友好，而且完全离线操作，无法对处理的数据文件做到统一管理，相同的模型处理方法或过程没有抽象封装，不利于进行个性化定制与平台化应用发展。

到2015年底，全球约有66亿件设备连入互联网，2020年，这一数字将上升到300亿件[20]。随着互联网技术和信息技术的快速发展，以互联网为载体的信息技术已逐渐影响甚至改变了传统生产及商业模式，并进一步地推动了工业技术的革新，引发了全球工业产业的技术升级。

近几年来，基于WebGL的浏览器应用蓬勃发展，使得很多处理三维图形的客户端软件都开始逐步开发Web端甚至移动端应用，比如CAD绘图，在线PS以及在线视频剪辑等，它不仅可以实现便捷快速的虚拟模型构建，也具备良好的交互性，受到越来越多的用户青睐。但是早期的三维图形技术比如Java Applet、Flash存在很多不足，例如资源占用多、性能低下以及诸多跨平台问题。WebGL技术的出现完美解决了这类问题，它不依赖任何浏览器插件，使用JavaScript脚本绘制三维图形，而且利用底层图形硬件加速功能，通过具有跨平台性的OpenGL接口渲染图形。WebGL的出现克服了传统三维全景技术存在的兼容性、高成本，高口槛的缺点，使得相关应用在未来有着广阔的发展空间。

## 课题目的和意义

目前大部分增材制造的方式都是本地、单次、非关联的，通过专业的建模软件读取模型数据，经过计算软件生成打印机可执行的G代码，最后进行实际打印。整个过程都是离线，需要使用到的工具软件繁多，过程中涉及到的数据等信息也是即用即消，无法做到数据追踪，也没有统一的数据存储。随着数据管理和可视化越来越受到相关领域的关注，因此本课题的主要目标是设计一个合适的架构平台去满足增材制造预处理相关应用，由于浏览器的Web平台具有优秀的跨平台特性，以及浏览器引擎不断更新换代后，计算能力大大提高，可以满足在线实时的模型数据处理需求，为3D打印未来的应用发展方向提供一个新思路。

## 1.3研究内容

* 3D打印流程快速原型。
* 增材制造相关附属功能。
* Web应用平台架构选型。
* Web建模技术调研。
* 平台系统实现。

# 国内外研究现状

## 2.1增材制造研究现状

增材制造(additive manufacturing, AM)技术是通过CAD设计数据采用材料逐层累加的方法制造实体零件的技术，相对于传统的材料去除(切削加工)技术，是一种“自上而下”材料累加的制造方法。自20世纪80年代末增材制造技术逐步发展，期间也被称为“材料累加制造”(material increase manufacturing)、“快速原型”(rapid proto-typing)、“分层制造”(layered manufacturing)、“实体自由制造”(solid free － form fabrication)、“3D 打印技术”(3D printing)等。名称各异的叫法分别从不同侧面表达了该制造技术的特点。

美国材料与试验协会(ASTM)F42 国际委员会对增材制造和 3D 打印有明确的概念定义。增材制造是依据三维 CAD 数据将材料连接制作物体的过程，相对于减法制造它通常是逐层累加过程。3D 打印是指采用打印头、喷或其他打印技术沉积材料来制造物体的技术，3D 打印也常用来表示“增材制造”技术，在特指设备时，3D 打印是指相对价格或总体功能低端的增材制造设备。

(1)国外发展状况

国际上增材制造经过 20 多年的发展，美国已经成为增材制造领先的国家，3D 打印技术不断融入人们的生活，在食品、服装、家具、医疗、建筑、教育等领域大量应用，催生许多新的产业。

美国奥巴马总统在 2012 年 3 月 9 日提出发展美国振兴制造业计划，向美国国会提出“制造创新国家网络”(NNMI)，其目的在夺回制造业霸主地位，要以一半的时间和费用完成产品开发，实现在美国设计、在美国制造，使更多美国人返回工作岗位，构建持续发展的美国经济。为此，奥巴马政府启动首个项目“增材制造”，初期政府投资3 000 万美元，企业配套 4 000 万元，由国防部牵头，制造企业、大学院校以及非盈利组织参加，研发新的增材制造技术与产品，使美国成为全球优秀的增材制造的中心，架起“基础研究与产品研发”之间的纽带。美国政府已经将增材制造技术作为国家制造业发展的首要战略任务给予支持。

据调查，价格低于 2 000 美元的设备多用于科学研究或个人，对行业产值影响不大。行业发展主要依赖于专业化设备性能的提高。目前，专业化设备主要销往美国市场。由于经济不景气隐藏的潜在客户被挖掘，并随着设计与制造的快速增长，快速成型制造行业也得以发展，并呈现以下特点：增材制造产业不断壮大、 新材料新器件不断出现、新市场产品不断涌现、新标准不断更新[12]。

(2)国内发展状况

我国自20 世纪90 年代初，在国家科技部等多部门持续支持下，在西安交通大学、华中科技大学、清华大学、北京隆源公司等在典型的成形设备、软件、材料等方面研究和产业化方面获得了重大进展。随后国内许多高校和研究机构也开展了相关研究，如西北工业大学、北京航空航天大学、华南理工大学、南京航空航天大学、上海交通大学、大连理工大学、中北大学、中国工程物理研究院等单位都在做探索性的研究和应用工作。我国研发出了一批增材制造装备，在典型成形设备、软件、材料等方面研究和产业化方面获得了重大进展，到2000 年初步实现的设备产业化，接近国外产品水平，改变了该类设备早期仰赖进口的局面。在国家和地方的支持下，在全国建立了20 多个服务中心，设备用户遍布医疗、航空航天、汽车、军工、模具、电子电器、造船等行业。推动了我国制造技术的发展。近5 年国内增材制造市场发展不大，主要还在工业领域应用，没有在消费品领域形成快速发展的市场。另一方面，研发方面投入不足，在产业化技术发展和应用方面落后于美国和欧洲。

近 5 年来，增材制造技术在美国取得了快速的发展。主要的引领要素是低成本 3D 打印设备社会化应用和金属零件直接制造技术在工业界的应用。我国金属零件直接制造技术也有达到国际领先水平的研究与应用，例如北京航空航天大学、西北工业大学和北京航空制造技术研究所制造出大尺寸金属零件，并应用在新型飞机研制过程中，显著提高了飞机研制速度。

在技术研发方面，我国增材制造装备的部分技术水平与国外先进水平相当，但在关键器件、成形材料、智能化控制和应用范围等方面较国外先进水平落后。我国增材制造技术主要应用于模型制作，在高性能终端零部件直接制造方面还具有非常大的提升空间。例如:在增材的基础理论与成形微观机理研究方面，我国在一些局部点上开展了相关研究，但国外的研究更基础、系统和深入;在工艺技术研究方面，国外是基于理论基础的工艺控制，而我国则更多依赖于经验和反复的试验验证，导致我国增材制造工艺关键技术整体上落后于国外先进水平;材料的基础研究、材料的制备工艺以及产业化方面与国外相比存在相当大的差距;部分增材制造工艺装备国内都有研制，但在智能化程度与国外先进水平相比还有差距;我国大部分增材制造装备的核心元器件还主要依靠进口。

(3)未来发展趋势

增材制造技术代表着生产模式和先进制造技术发展的趋势，产品生产将逐步从大规模制造向定制化制造发展，满足社会多样化需求。目前增材制造设备在软件功能和后处理方面还有许多问题需要优化。例如，成形过程中需要加支撑，软件智能化和自动化需要进一步提高;制造过程，工艺参数与材料的匹配性需要智能化;加工完成后的粉料或支撑的需要去除等问题。这些问题直接影响设备的使用和推广，设备智能化是走向普及的保证。向组织与结构一体化制造发展。实现从微观组织到宏观结构的可控制造。例如在制造复合材料时，将复合材料组织设计制造与外形结构设计制造同步完成，在微观到宏观尺度上实现同步制造，实现结构体的“设计—材料—制造”—体化。支撑生物组织制造、复合材料等复杂结构零件的制造，给制造技术带来革命性发展。

## 2.2增材制造软件研究现状

増材制造软件是增材制造技术中的关键部分，其中最为核心的是从工件模型到增材制造设备的模型处理和数据转换软件[3]。目前，基于工件三维模型的增材制造的软件系统的研究开发主要有三个方面[4]：三维模型获取软件，模型处理软件和过程监控软件。模型获取软件负责获得模型的基本几何形态和其它所需几何特征数据等；模型处理软件是增材制造软件系统中的核心部分，其用来确定加工位向，进行模型的切片分层，层片轮廓数据处理以及终端设备的加工路径规划等；过程监控软件则负责增材制造加工过程中各类参数的设定以及焊接质量在线检测等。在以上几个部分中，零件模型获取软件可以使用市场上已有的CAD建模软件或自行开发的相关软件，而模型处理软件与过程监控软件一般需要增材制造系统制造方自行开发。而国内外对增材制造的软件研究与开发主要集中于分层切片及路径规划的算法的研究上。

国外各种增材制造设备一般都带有开发商自行开发的软件系统[3]。除了直接嵌入到增材制造设备的可对模型直接处理的软件之外，国外有人开发出一部分第三方软件，作为工件三维模型与增材制造系统之间的转换工具，主要有[5]：比利时Materialise公司的Magic；美国Solid Concept公司的Brige Works，Solid View，PCGO公司的STL-Manager等。

增材制造软件或其中的部分功能模块在国内己有大量研究，部分高校和科研机构己取得较成功的研究成果。上海交通大学的朱晓鹏为激光烙敷再制造过程专门开发了基于STL格式模型的分层切片算法[6]；河北工学院的张志钢开发了基于STL模型的预处理软件，分析了从模型显示到分层切片、层片轮廓分区和设定工艺参数的各个步骤和主要算法实现[7]；华中科技大学的尹西鹏在现有增材制造软件中添加了面向相应设备的反馈控制功能，是增材制造软件系统中过程监控软件的一个较佳实例[8]；朱胜[10]利用机器人等系统的柔性控制功能，发展研究了柔性增材再制造技术，对缺损零件进行了反向建模、再切片分层和重新路径规划，其成型精度控制良好，实现了金属零件的再制造。

哈尔滨工业大学的蒲英钊[3]研究开发了针对KUKA机器人工作站的增材制造离线编程系统，开发了一种点云模型切片方法，并创新性地添加了动画仿真模块，可以让用户预先模拟检查工件的增材制造过程，提高了增材制造系统的可靠性。

北京工业大学的陈树君[9]搭建了基于KUKA机器人的铝合金CMT增材制造系统，使用Robotmaster和MasterCAM建立了用于增材制造的机器人离线编程软件系统，对零件进行分层、轨迹生成和模拟仿真，并进行了铝合金零件增材制造试验。并对熔覆工艺参数（喷嘴高度、送丝速度、焊接速度）与熔覆层高度、宽度的成型规律进行了研究，建立了一个熔覆层尺寸的预测模型。通过铝合金零件成型试验对模型精度进行了验证，发现该模型预测效果良好，成形铝合金零件精度高。

当前，国内外增材制造软件主要存在下四个方面的不足[8]：

①增材制造软件无标准化。绝大多数的增材制造软件往往是根据增材制造设备本身的特点而专门开发的，只能与设备制造者的本身增材制造特征相适应，而现有增材制造软件各自都有不同的开发方式，不同软件间的联接与数据共享几乎无法实现。

②难于对现有软件进行二次开发。目前的增材制造软件一般只适用于相应的固定设备，没有标准的数据处理与交换格式，无法满足用户根据具体情况进行具体调整或二次开发，且一般不能集成不同的工艺。

③单独开发工件三维模型与増材制造设备之间的转换软件具有较大困难性，故而有时需要采用第三方软件作为中间的数据过渡转换，这会増加増材制造软件的复杂度。

④价格昂贵，功能不足。当前绝大多数较完善的增材制造软件的加工都十分髙昂[14]，且大多只具备模型信息分析、分层切片、加工路径生成等基本功能，后续的加工控制和过程监控模块则只能由用户另行解决。

## 2.3 Web服务的研究现状

Web服务的概念目前并没有统一的标准。万维网联盟(World Wide Web Consortium，简称W3C)总结的概念为：通过统一资源标识(Uniform Resource Indication，简称URI)的软件应用，就是Web服务，能经由XML文档来发现、描述与定义它的绑定、接口；通过互联网协议直接交互其余软件，其能对以可扩展标记语言(Extensible Markup Language，简称XML)为基础的消息提供支持[21]。

IBM给出的概念是：模块化、自包容应用就是Web服务，其能经由互联网、尤其WWW等把调用完成，可以被定位、发布、描述。Microsoft给出的定义是：XML Web服务对其它应用程序进行逻辑性服务、数据提供，其主要属于经由规范数据格式、Web协议可编程访问的Web构件[22]。

Web服务作为互联网上的API，具有以下特点：

平台无关性，由于Web服务是使用SOAP协议来实现请求和反馈的，所以对于Web服务的调用，无需再建立其他平台或协议之间的桥接，直接利用SOAP协议的平台无关化即可实现跨平台交互，很好的实现了系统的可集成性。因此在选取测试工具和脚本开发语言方面有了更多的选择和更好的适应性[24]；

松散耦合：Web服务提供者和服务使用者是松散耦合关系。以面向服务体系结构为基本依据构建而成的系统中，需求如果有所改变，仅需进行操作修改或业务服务流程调整就可，根本无需对服务提供接口进行修改，因此在测试过程中利于自动化测试的引用，很大程度上降低自动化测试的维护成本[23]；

标准化、粗粒度接口：应用功能在Web服务前提下能够经由提供WSDL标准化接口，并在JMS与HTTP的标准化传输方式基础上，通过SOAP标准化协议进行调用。这就为了测试过程中将各个部件互相隔离提供了出色的手段。在做系统集成测试之前，可以执行其他的相关测试，如性能测试、压力测试、负载测试等[25]。

## 2.4 WebGL应用的研究现状

在过去的十年里，互联网技术有了突飞猛进的发展，逐渐参透到人们生活的各个角落，使人们的生活方式产生了巨大的改变。在互联网的各个领域中，发展和变化最快的就是Web应用的发展，已经成为当今网络技术的研究重点。随着人们对网页体验的要求不断提高，网页正在逐渐地从传统的二维平面网页向交互式三维网页发展。

但是，早期的技术并不成熟，比如Java Applet所实现的非常简单的Web交互三维图形程序，不仅需要下载一个巨大的支持环境，而且画面非常粗糙，性能也很差，其主要原因就在于Java Applet在进行图形渲染时，并没有直接利用到图形硬件本身的加速功能。

后来，Adobe的Flash Player浏览器插件和微软Silver Light技术相继出现，逐渐成为了Web交互式三维图形的主流技术。与Java Applet技术不同，这两种技术直接利用操作系统提供的图形程序接口，来调用图形硬件的加速功能，从而实现了高性能的图形渲染。但是，这两种解决方案都存在一些问题。首先，它们是都是通过浏览器插件形式实现的，这就意味着对于不同的操作系统和浏览器需要下载不同版本的插件；其次，对于不同的操作系统，这两项技术需要调用不同图形程序接口。这两点不足在很大程度上限制了Web交互式三维图形程序的使用范围。

普通的Web浏览器正日益成为能应用丰富3D交互式程序的一个平台。2014年10月，万维网联盟完成了HTML5的标准制定，而作为HTML5标准之一的WEBGL很好地解决了上述两个问题：首先，它通过JavaScript脚本实现Web交互式三维图形制作程序的设计与实现，无需任何浏览器插件支持；其次，它利用底层图形硬件的加速功能进行的图形渲染，是通过统一的、标准的、跨平台的OpenGL ES2.0实现的。

WebGL是一种3D绘图标准，这种绘图技术标准允许把JavaScript和OpenGL ES 2.0 结合在一起，WebGL可以为HTML5的Canvas标签提供硬件 3D 加速渲染，这样Web开发人员可以借助系统显卡来在浏览器里更流畅地展示 3D 场景和模型。

WebGL 解决了现有的 Web 交互式三维动画的两个问题： 第一， 在无需安装浏览器插件的情况下，通过JavaScript脚本语言实现Web交互式三维图形绘制和动画制作 ; 第二，通过统一的、标准的、跨平台的OpenGL接口实现利用底层的图形硬件加速功能进行的图形渲染。

# 关键技术及主要工作

## 技术方案

①对系统进行快速原型建模。

②web平台搭建，实现对模型的各种交互式操作，以及通用函数的封装。

③实现切片算法以及轨迹路径生成算法，并基于目前C++编写的算法进行改进设计。

④对模型以及相关操作设计数据库存储。

⑤分布式部署，满足并发式使用。

⑥针对市面主流的几款打印机器，生成相对应的G代码。

## 系统原型建模

原型建模是解决软件危机的一种行之有效的方法，主要是在需求分析阶段先行设计一个与需求尽可能匹配的示意版系统，然后通过实际的需求分析沟通，进行多次原型迭代和细化，从而准确表达用户的真实需求，并可以降低系统开发的重构成本。

一般在业内通常使用的是Axure，它通过制作出逼真的项目高保真原型, 使开发团队在软件开发前, 就能真实地体验和直观地展示未来软件的效果与核心逻辑功能, 从而实现精确的需求分析。Axure能便捷地创建基于目录组织的原型文档、功能说明等, 并自动生成原型的Html文件和页面的需求与设计的Word文档。原型Html文件经优化后, 可直接用于后期逻辑代码的开发工作, 有机地将需求工作成果应用于后期逻辑编码工作, 使原型建模不只是完成了需求分析工作, 而是完成了主要的项目表示层的编码工作。

原型法软件开发过程一般大致分为：需求沟通和市场调研、项目评估、原型设计、软件设计、开发编程、测试和产品交付等6个阶段, 如图3.2.1所示。

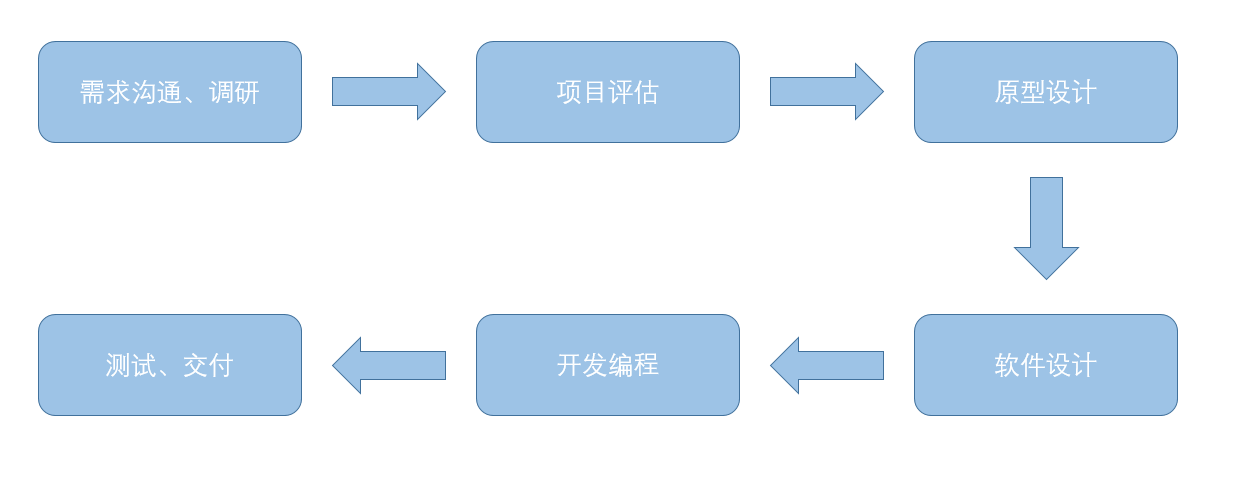


图3.2.1原型法软件开发过程

针对该系统需求绘制了初步的原型图：



图3.2.2 登录注册页面



图3.2.3 模型管理页面



图3.2.4上传模型页面

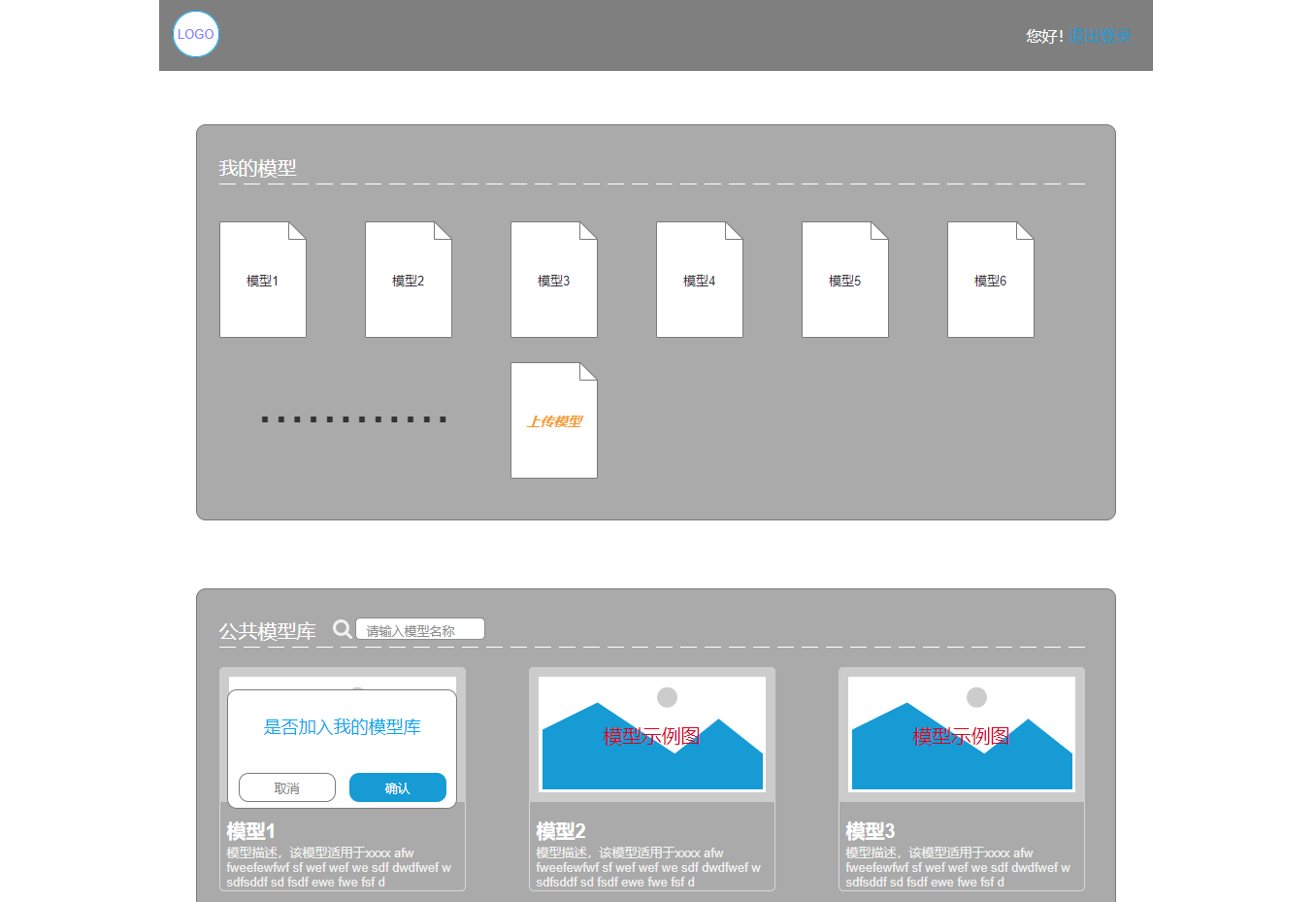


图3.2.5选择系统模型页面



图3.2.6模型交互页面



图3.2.7生成切片锥面菜单

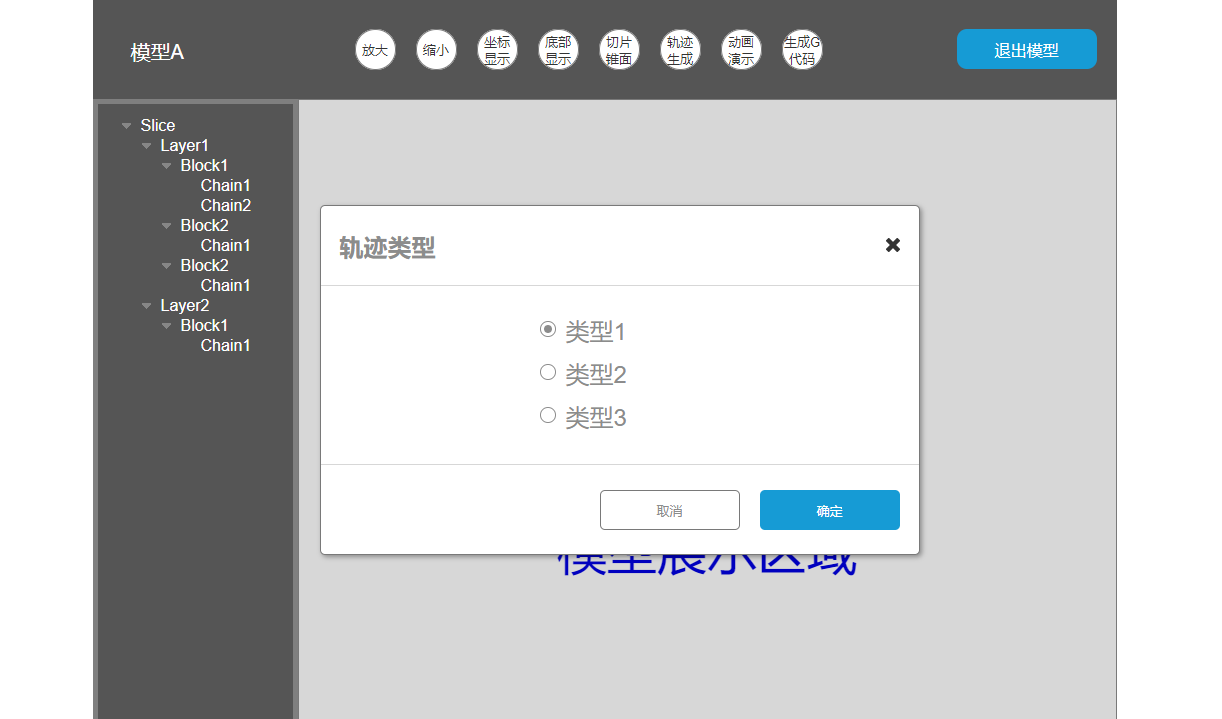


图3.2.8生成轨迹菜单

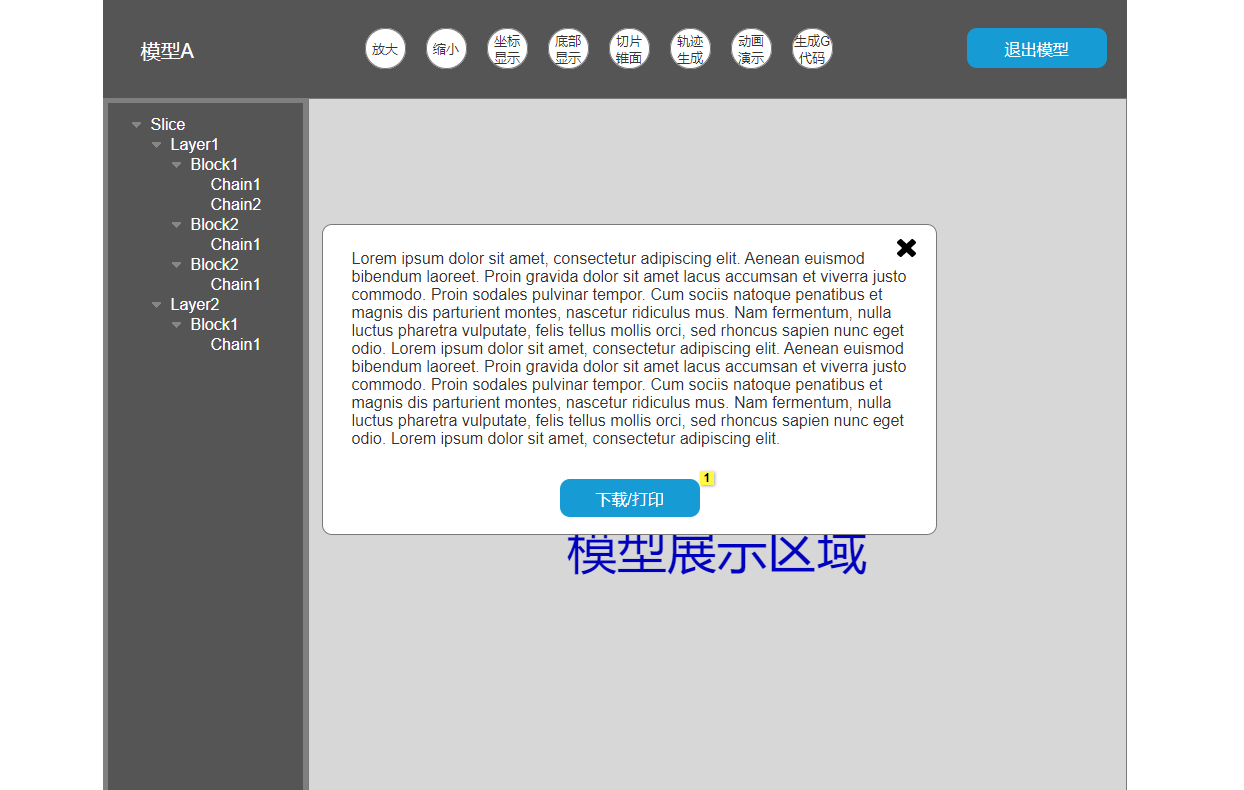


图3.2.9生成G代码

### 3.2.2增强式交互

物理实体是实际存在的客观对象，操作人员可以直观方便地与之交互。而数字孪生体是构建于信息空间的虚拟对象，难以与操作人员形成类似物理空间中的交互与理解。为了使孪生体在显示形态与交互便利性上更接近于物理实体，增强式交互技术被广泛运用在数字孪生系统的对外显示与交互中，主要技术包括OpenGL、WebGL甚至有VR、AR和混合现实等。良好的人机交互页面会降低研究人员上手操作的难度，也可以让应用得以广泛传播。

### 3.2.3应用平台架构

增材制造作为一项技术，将其转化应用、为企业创造价值才能真正体现这项技术的价值。目前市面上对于增材制造的落地产业应用都趋向小众，并且还未形成一个规范的平台应用产业链，要形成一个完善的工业化产业链体系，还需要广大学者与企业进行合作交流，促进相应标准的制定与出台，并且暂时也没有比较有实验性的平台进行推广使用，所以这块市场还处于蓝海阶段，市场潜力巨大。

大致可以确定出一个这样的整体架构，如图4所示。

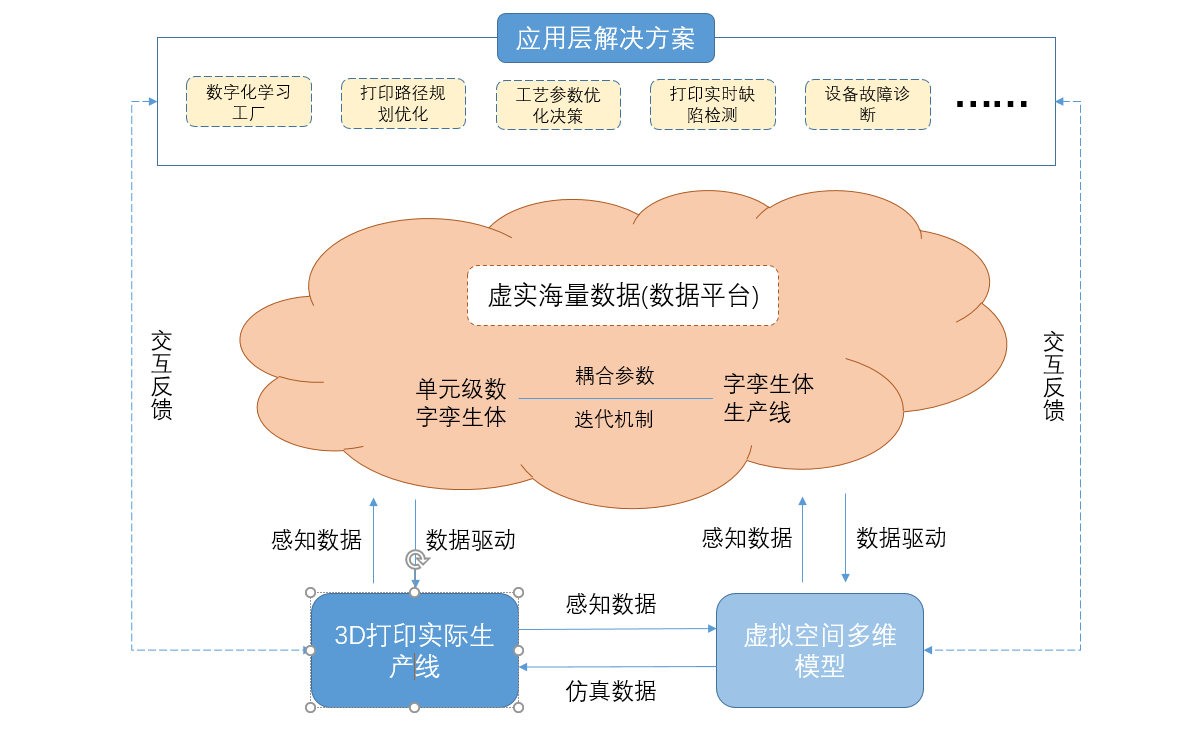


图4增材制造相关应用平台架构图

## 平台技术选型

随着信息技术的不断发展，传统的C/S架构的开发模式已经逐渐被B/S架构模式所取代，同时随着业务需求越来越复杂化和多样化，前端工程化以及前后端分离的开发方式已经深入人心，被越来越多的开发者所采纳。因此本系统基于单页面应用程序(SPA)开发，采用前后端分离模式，前端组件化、工程化，后端数据化；前端实现人机交互逻辑，提供业务数据展示，后端为前端提供业务数据支撑；前后端通过RESTful接口进行数据交互。

### 3.3.1前端开发选型

前端架构基于MVVM的模式，如图5所示，在该模式下，视图和模型是不能直接通信的，ViewModel通常要实现一个observer，也就是观察者模式，来监听数据变化，然后通知对应的视图做自动更新，而当用户操作视图时，ViewModel也能监听到视图的变化，然后通知数据做改动，从而实现双向绑定。

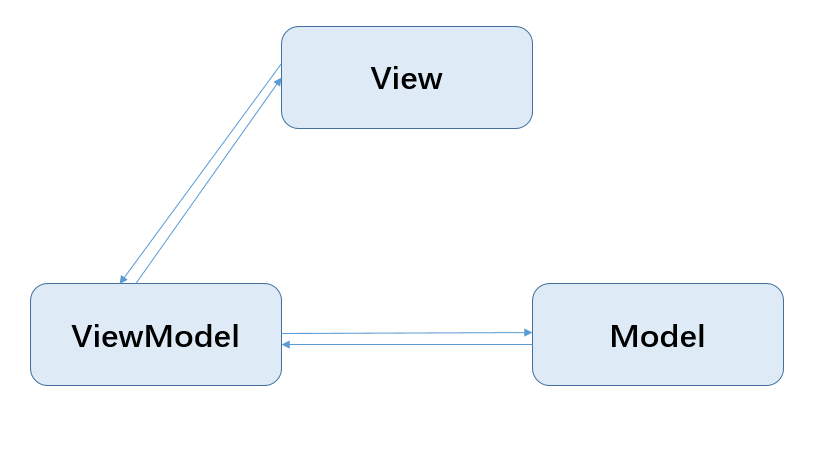


图5MVVM架构示意图

市面上主流的基于MVVM模式的开发框架有Angular、React、Vue，在这里我们选择Vue作为前端的主体框架，框架依赖的中间件有Vue-router、Vuex、Axios以及样式库Antd-Design-Vue。整个前端技术架构如图6所示：

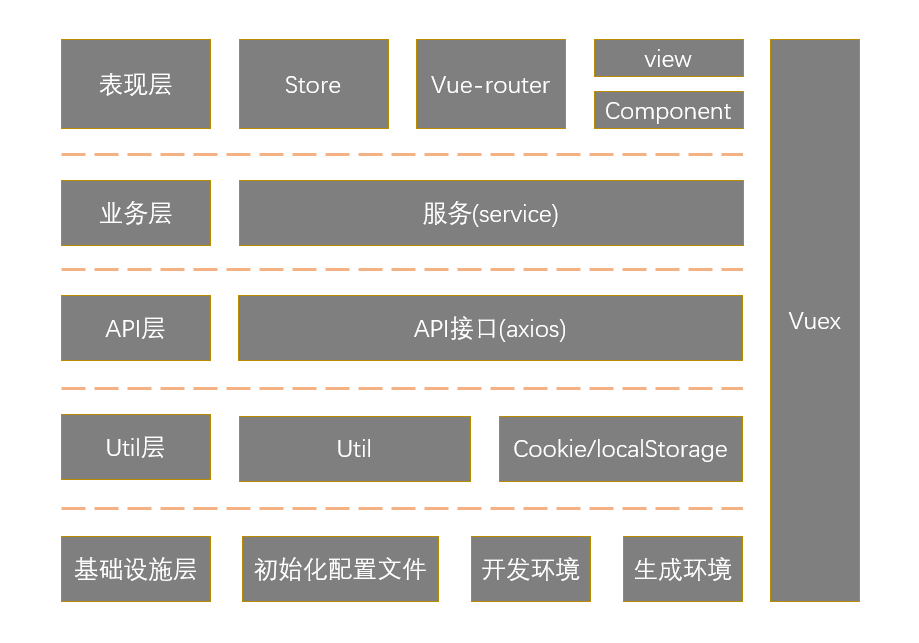


图6前端技术架构示意图

### 3.3.2模型交互技术选型

互联网产业的蓬勃发展使得二维图形无法满足软件业务需要。于是，基于Web三维图形的交互式全景漫游技术应运而生，它不仅可以实现在线虚拟场景漫游，而且具有杰出的交互性，受到众多用户青睐。但是早期三维图形技术比如Java Applet、Flash存在诸多不足之处，例如占用资源多、性能低、跨平台性问题[13]。WebGL技术的出现完美解决了这类问题，它不依赖任何插件，使用JavaScript脚本绘制三维图形，而且利用底层图形硬件加速功能，通过具有跨平台性的OpenGL接口渲染图形，WebGL能够很好地帮助我们在一些需要强视觉交互的领域做出好应用。

这里我们选择的是Three.js，它是基于WebGL的第三方库，封装了WebGL的接口，抽象出一些简单的对象，比如camera、light、mesh、material等等，并基于此提供了非常多的3D显示功能，方便开发者进行三维场景构建。

在Three.js的三维场景中，它定义了几种基本部件：场景、渲染器、相机、物体。下面对此做简要介绍：

1.场景：是三维模型的显示容器，即将三维空间映射到二维平面上，让我们可以在二维的视野范围内看到三维图像。

2.渲染器：即负责将空间三维图像投影到二维平面上。在这一过程中我们还可以加入比如光源(Light)来模拟现实中的自然光。

3.相机：可以理解为在三维模型中人的眼睛，即我们怎么看它，在Three.js中主要包括透视投影和正投影。透视投影与我们在日常生活中观察物体的方式一致，即离视点越近的物体投影越大，越远的物体投影越小；正投影就是不考虑物体与视点的距离，将物体按统一大小绘制。

4.物体：即在场景中显示的几何模型，比如常见的立方体、球体、圆柱体等等，在这基础上我们才可以构建负责的几何模型，同时我们还能在物体上加入材质等参数以此来模拟现实中的物体。

综上所述，Three.js大幅度降低了在浏览器中实现3D模型的技术门槛，通过这些简洁易懂的对象，我们可以将任意的模型按照自己需要的方式绘制在网页中。

### 3.3.3服务端技术选型

后台采用Java语言，并选择Spring Boot框架，因为它是当前Java语言社区中一个非常流行的、先进的、实现了MVC思想的开发框架，它相对于传统的SSH框架更具有优势。Spring框架能够为Web应用程序的开发提供一系列的解决方案，而Spring Boot框架简化了基于Spring的Web应用程序的体系结构，它在拥有Spring框架功能的同时，又大大简化了程序的配置方式。

通过使用Spring Boot开发框架，程序员可以创建独立的Spring程序，并且Spring Boot框架中有内置服务器，比起SSH框架来能够大大减少需要程序员进行构建配置的工作。在使用Spring Boot开发框架时，程序员可以通过注解的方式来配置一些包括像数据库的连接信息等的动态参数。Spring Boot框架能够支持自动地配置应用程序所需要的Spring框架以及第三方的库，并且程序员不再需要编写大量的XML代码来进行构建配置，这大大降低了程序员的额外配置工作，从而减少了程序员的工作量。

因为Spring Boot框架中集成了Tomcat、Jetty等服务器组件，程序员在进行Web应用系统开发时，不再需要将应用程序发布到单独的JBoss、Tomcat等HTTP服务器上。程序员可以跟启动一个普通的Java程序一样在Spring Boot框架中启动其开发的Web应用系统。

### 3.3.4数据库选型

Mysql和MongoDB都是开源的常用数据库，一个是传统的关系型数据库，MongoDB则是非关系型数据库，也叫文档型数据库，是一种NoSQL的数据库。两者差异对比如图7所示：



图7 MongoDB与MySQL的差异对比

鉴于本系统会有很多json格式的模型数据以及数字孪生体所需大量数据的背景下，MongoDB的高速读写特性，以及未来的高扩展性，都是我们系统数据存储方案的不二之选。

### 3.3.5 WebAssambly

WebAssembly是一种由多家浏览器厂商共同推出的新的可移植二进制代码形式，它可以为Web端应用程序提供接近本地层代码的执行速度[15][16]，并且可以作为 C/C++，Rust[17]等语言的编译目标。正因为这样，它为以前难以移植到 Web 端的应用提供了解决方案，并且提供了高效的运行环境[18]。

WebAssembly允许使用者采用C，C++或Rust等代码，并将其编译生成相应的WebAssembly模块。用户可以将其加载到Web 应用程序中并通过JavaScript 调用它，它不是JavaScript的替代品，它需要与JavaScript一起使用。目前，Emscripten提供了一个框架，用于将C/C++等编译生成WebAssembly，并且生成在JavaScript中嵌入WebAssembly模块所需的执行环境。通过这个框架，就可以在JavaScript运行过程中调用通过C/C++等实现的函数。

WebAssembly被创造的主要目的是在Web端获取更好的执行性能，它的二进制文件形式比JavaScript文件拥有更小的体积，因此加载和执行速度也更快。JavaScript是一种弱类型语言，在编写代码时不需要进行变量类型的定义，也不需要提前编译。这使得代码的编写变得简单快捷，但是这也意味着JavaScript引擎需要承担许多工作。它必须在浏览器端对代码进行解释执行。解释执行 JavaScript的过程需要将纯文本的代码解析生成抽象语法树的数据结构，并且将其转换成二进制格式[19]。而WebAssembly以二进制的形式提供，解析速度更快[20]。并且WebAssembly是一种静态类型语言，它与JavaScript不同，不需要在执行的过程中进行复杂的数据类型的推导，而且大多数的优化都是在编译源代码期间，甚至在进入浏览器之前发生的。这就使得在浏览器端可以减少额外的性能开销。并且 WebAssembly和C/C++这样的语言一样，内存都是手动管理的，并不提供垃圾回收机制。所有的这些机制都是为了提供了更好，更可靠的性能。所以通过 WebAssembly可以将大量与性能相关的计算密集型代码移植到 Web 端运行。

## 系统评价指标

①前端首屏加载时间：

0-3秒：用户体验最好，打分100

3-8秒：用户可以容忍，从第3秒开始，每超过1秒减5分

8-15秒：用户不能忍受，从第8秒开始，每超过1秒减10分

②并发用户数：

基于本系统暂时只考虑满足100人以内的小型实验室使用，所以只要服务端能同时满足100次以内的计算请求且响应时长不超过2s就可以接受。

0-2秒：用户体验最好，打分100

3-5秒：用户可以容忍，从第3秒开始，没超过1秒减5分

5-10秒：用户不能忍受，从第5秒开始，每超过1秒减10分

③界面渲染帧率：

因为使用到了WebGL的应用系统，所以模型交互的帧率是必不可少的衡量标准，一般来说画面在 60fps 的帧率下效果比较好。换算一下就是，每一帧要在 16.7ms (16.7 = 60/1000) 内完成渲染。因此，我们的首要任务是减少不必要的性能消耗。 越多的帧需要渲染的，意味着有越多的任务需要浏览器处理，所以掉帧就出现了，这是达到 60fps 的一个绊脚石。如果所有动画都无法在 16.7ms 渲染完毕，就考虑用略低的 30fps 帧率来渲染。

60fps以上：用户体验优秀，打分100分

30-60fps：用户体验良好，打分80分

低于30fps：用户体验一般，打分60分

④服务器响应时间：客户端发送一个HTTP请求至收到HTTP响应头信息消耗的时间(因为此处涉及到常规后端查表请求和后端数据计算请求两种不同类型，因此时间标准无法做一致性判断，可以通过增加loading样式去减轻用户等待的体验)。

⑤模型数据在经过webGL处理后是否会产生点丢失。

⑥前端解析渲染效果与单机版客户端渲染效果对比。

# 研究进展计划

|  |  |
| --- | --- |
| **时间** | **任务** |
| **2020.9 – 2020.10** | 阅读文献，撰写开题报告 |
| **2020.11 – 2020.12** | 总体方案设计、关键技术研究与实现 |
| **2021.1 – 2021.2** | 系统平台搭建与测试 |
| **2021.3 – 2021.4** | 文档整理、撰写硕士学位论文 |
| **2021.5 – 2021.6** | 硕士论文审定、修改及答辩 |

# 参考文献

1. 路甬祥.中国智造与中国创造[J]. 全球化, 2016(9):5-13.
2. 陶飞，戚庆林.面向服务的智能制造[J]. 机械工程学报, 2018.
3. 蒲英钊．机器人弧焊增材制造离线编程系统研究［Ｄ］．哈尔滨：哈尔滨工业大学，2014.
4. Y.M.Zhang, Y.Chen, P.Li，A.T.Male. Weld deposition-based rapid prototyping a preliminary study[J]. Journal of Materials Processing Technology, 2003, 135:347~357.
5. 张嘉易，刘伟军，王天然，等．快速成型数据处理系统研究[J]．机械设计与制造，2004，(3)：95~97.
6. 朱晓鹏．激光熔覆再制造过程中的分层切片方法[D]．上海；上海交通大学，2013．
7. 张志钢.SLS快速成型系统预前处理软件研究[D]．天津：河北工学院，2008．
8. 尹西鹏．选择性激光熔化快速成型系统设计与实现[D]．武没：华中科技大学，2008.
9. 陈树君，赵昀，肖珺等．铅合金电弧熔积成形机器人増材制造系统[J].焊接·焊接增材制造专题，2016，（4）：9~12．
10. 朱胜. 柔性增材再制造技术[J].机械工程学报，2013，49(23)：1~5.
11. 卢秉恒,李涤尘.增材制造(3D打印)技术发展[J].机械制造与自动化,2013,42(04):1-4.
12. ToddZakiWarfel, 沃菲尔，汤海，等.原型设计:实践者指南[M]. 北京:清华大学出版社,2013.
13. 赵孔阳,朱军,尹灵芝.基于WebGL的虚拟场景网络漫游研究[J].测绘与空间地理信息,2015,38(12):40-43.
14. Park J T, Kim H G, Moon I Y. WebAssembly Performance Analysis for Multimedia Processing in Web Environment[C]// International conference on future information & communication engineering.2018,10(1): 288-290.
15. Watt C. Mechanising and verifyingthe WebAssembly specification[C]//Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs. ACM, 2018: 53-65.
16. Matsakis N D, Klock II F S. The rust language[C]//ACM SIGAda Ada Letters. ACM, 2014, 34(3):103-104.
17. Heil S, Siegert V, Gaedke M. ReWaMP: Rapid Web Migration Prototyping Leveraging Web Assembly[C]//International Conference on Web Engineering. Springer, Cham, 2018: 84-92.
18. Guha A, Saftoiu C, Krishnamurthi S. The essence of JavaScript[C]//European conference on Object-oriented programming. Springer, Berlin, Heidelberg, 2010: 126-150.
19. Rossberg A. WebAssembly: high speed at low cost for everyone[C]//ML16: Proceeding of the 2016 ACM SIGPLAN Workshop on ML. 2016.
20. 沃尔克马尔·邓纳尔. 工业4.0将融入中国制造2025[J]. 经理人，2015(7):24-24.
21. W3C Working Group. Web Services Architecture Requirements [EB/OL], https://www.w3.org/TR/2002/WD-wsa-reqs-20020819#IDAIO2IB. 2002.2016.3.12.
22. JM Butle. Web service grid architecture[P]. United States Patent 8756320,06/17/2014.
23. 徐秀勤. Web Service(Web服务)初探[J]. 硅谷，2014(18):197-197.
24. Ferris C，Farrell J. What are Web services? [J]. Communications of the Acm, 2003, 46(6):31-31.
25. Yan M, Sun H, Wang X, et al. WS-Taas: A Testing as a Service Platform for Web Service Load Testing[J]. 2012, 90(1):456-463.

|  |  |  |
| --- | --- | --- |
| **研究生签字**  **指导教师签字**  **院(系、所)领导签字** |  |  |

**年 月 日**