# 系统架构设计

## 项目概述

本项目业务场景主要围绕“产品定制服务”展开的，用户只需用文字描述理想产品的形状、大小与颜色，即可直接生成对应的3D模型，使商家能够精准理解需求，实现高效的一对一产品定制。彻底改变了传统定制中沟通困难、需求不明的痛点。

针对此需求，我们设计了这样一个集成了自然语言交互、历史对话匹配与3D模型生成的智能系统。用户可通过对话接口提出建模需求，系统自动解析意图、匹配历史记录，并调用3D模型生成服务，最终在Web前端展示生成结果。

## 3D模型选型方案

### 2.1核心需求与评估方案

评估标准如下：

1. 输入支持：支持文本输入；

2. 部署方式：通过API直接调用，无需自行推理部署；

3. 所需成本：调查API调用单价，估计生成任务的预估总成本，在满足需求的前提下，选择成本最优的方案；

4. 生成质量：与输入的一致性，几何质量，模型保真度（细节，真实感）。

### 2.2 详细评估与对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型名称 | 输入支持 | 部署方式 | 成本 | 生成质量 |
| Replicate平台 | 多为文本 | 平台托管多个模型的API | 成本低，模型种类多，有免费额度。但是需要绑定国际银行卡 | 质量因模型而异，需要测试不同模型 |
| Luma AI | 文本，图像 | 官方直接提供API | 价格偏高，提供免费使用额度 | 效果最佳，细节丰富，光影效果逼真 |
| Tripo AI | 文本，图像 | 官方直接提供API | 按生成次数计费，性价比高，有免费额度 | 综合表现均衡，纹理质量好，生成速度快，支持多种输出格式 |
| Kaedim | 图像 | 提供企业级API服务 | 企业级定价，成本高 | 专注于图像转3D，输出网格质量高 |
| OpenAI的Shap-E | 文本 | 通过云平台提供API | 标准云服务计价模式 按token计费 | 纹理细节简单，基础几何 |

### 2.3 综合分析与最终挑选结果

最终选择使用Tripo AI作为本项目的3D生成模型，优势如下：

1.API易用性：有清晰的API文档和Python SDK；

2.集成性：与LangChain等工具配合良好

3.生成质量均衡 ：在速度和质量间取得良好平衡，符合本项目的实际需求；

4.输入输出格式支持全面：输入支持文本或图像，输出支持GLB、OBJ等标准格式；

5.性价比高：成本低，按生成次数计费，有免费额度。

## 三.架构设计

本项目的整体架构如下图所示，主要分为三大模块：基于langGraph的agent设计，基于flask的api设计，前端页面。系统通过三个Agent协同工作，实现从用户输入到3D模型生成的自动化流程。后端Flask API提供稳定的服务接口，前端三个页面分别负责交互、展示和评估，形成一个完整的闭环。这样的架构既保证了系统的可扩展性，也提供了良好的用户体验。

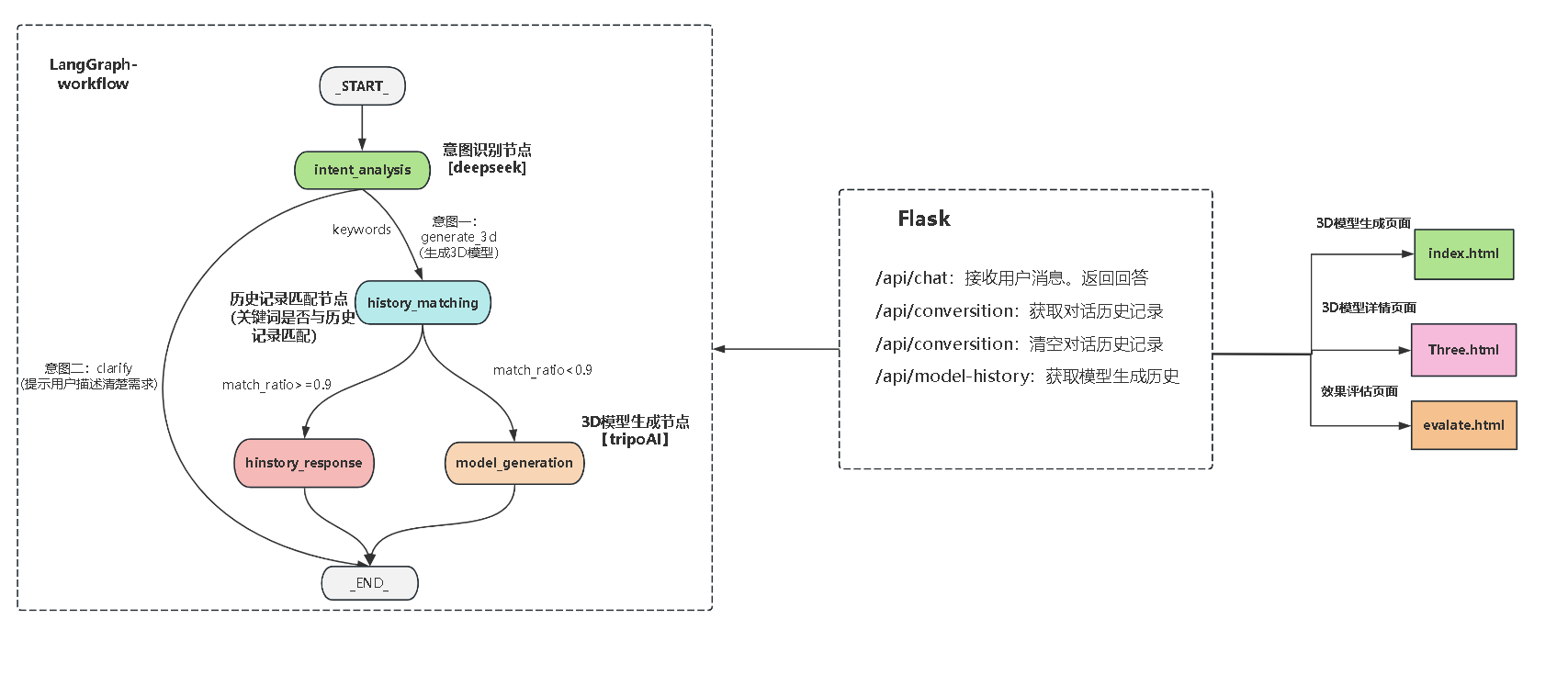


图1 系统整体架构设计

### 3.1 Agent设计

Agent主要是基于langChain和langGraph来构建workflow，是系统的核心处理单元，负责理解用户输入、匹配历史记录和生成3D模型。

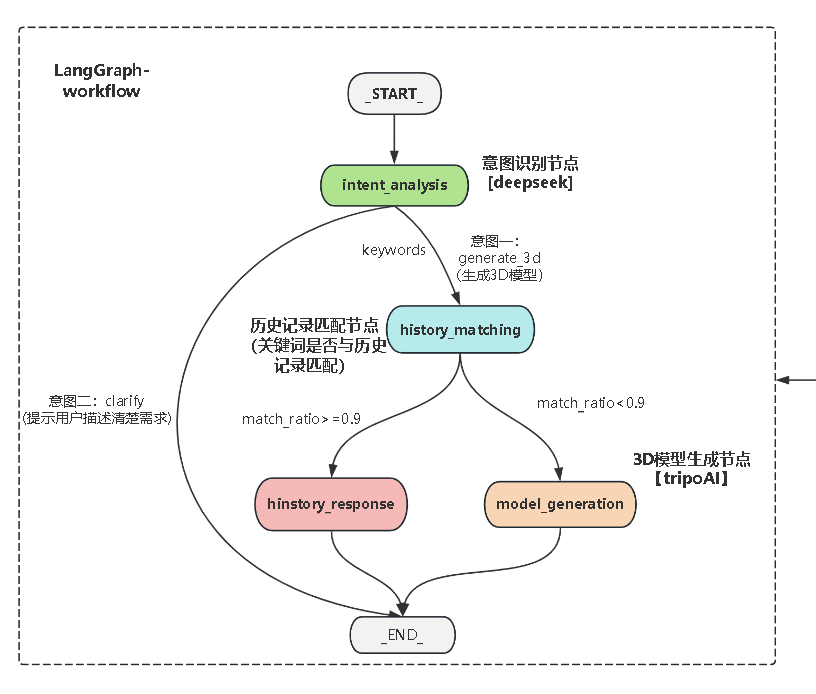


图2 agent设计workflow图

1. **语义理解 Agent (deepseek)**

基于 DeepSeek 大语言模型构建，支持自然语言处理(NLP)和意图识别。

此节点的主要功能是：接收原始用户文本，分析语义意图；从对话中提取3D建模相关的核心关键词，便于后续的基于关键词匹配去查找历史对话记录；另外输出标准化格式便于后续节点处理。

1. **历史记录匹配 Agent**

基于上一步意图识别节点提取到的关键词，针对之前的对话记录进行相似计算。采用Jaccard进行比较。当阈值>0.9，则匹配成功，无需调用3D模型生成API，直接返回之前生成的历史对话记录即可。

此节点设置了可配置的匹配阈值，平衡性能与准确性。减少重复计算，优化系统的响应时间。

1. **3D模型生成 Agent (trippAI)**

集成 trippAI 3D生成引擎，支持文本到3D的端到端生成。根据关键词或历史响应生成3D模型，输出标准化的GLB格式文件，确保生成模型的可视化质量和结构合理性；处理并发生成请求，管理生成队列。

### API设计

系统后端使用Flask框架提供API，主要包含以下接口：



图3 接口设计

1. **/api/chat（POST）**

**（1）功能描述：**接收用户消息，触发整个处理流程（包括语义理解、历史匹配和3D模型生成），并返回回答。

**（2）请求体：**包含用户消息的JSON对象。

**（3）响应：**返回处理结果，包括生成的回答、生成ID、3D模型路径（如果生成）或历史匹配结果。

**2.  /api/conversion（GET）**

**（1）功能描述**：获取当前用户的对话历史记录。

**（2）响应**：返回对话历史列表，包括用户消息和系统回复。

**3.  /api/conversion（DELETE）**

**（1）功能描述**：清空当前用户的对话历史记录。

**（4）响应**：返回操作结果。

**4.  /api/model-history（GET）**

**（1）功能描述**：获取3D模型生成的历史记录，包括生成ID、关键词、生成时间等。

**（2）响应**：返回模型生成历史列表。

### 3.3 前端页面设计

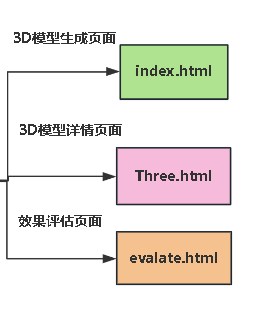


图4 前端设计

前端由三个页面组成，使用HTML、JavaScript和Three.js库实现。

1. **Index.html页面**

**（1）功能描述**：3D模型生成的主页面，提供用户输入界面，展示对话历史和3D模型生成结果。

**（2）技术实现**：使用Ajax与后端API通信，发送用户消息并接收回答。展示对话历史，并当有3D模型生成时，提供链接跳转到Three.html页面查看模型。

**2. Three.html页面**

**（1）功能描述**：专门用于展示3D模型的页面。

**（2）技术实现**：使用Three.js加载和渲染GLB格式的3D模型，使用户可以从不同角度查看模型。

**3. evaluate.html页面**

**（1）功能描述**：数据传输和效果评估页面，用于展示系统生成效果的数据分析。

**（2）技术实现**：通过调用/api/model-history等接口获取生成历史数据。使用图表展示生成时间、用户评分等评估指标。

## 四.模型调用频次优化

### 4.1 优化目标与设计思路

**1. 优化目标**

* **减少不必要的模型调用**：避免对无关请求或重复请求的3D模型生成；
* **提升系统响应速度**：通过缓存和历史复用降低用户等待时间；
* **降低计算资源消耗**：优化trippAI模型的使用频次，控制成本；
* **提高用户体验**：确保相关请求快速响应，无关请求友好处理。

1. **整体设计思路**

系统采用**双层过滤机制**，在请求进入3D模型生成流程前进行两次判断：

* **意图识别过滤机制**：过滤无关对话内容
* **历史记录匹配复用机制**：复用已有生成结果

### 4.2 意图识别过滤机制

**1. 意图定义；**

系统采用二元意图分类体系，将用户输入明确分为两类：

|  |  |  |  |
| --- | --- | --- | --- |
| **意图类型** | **标识符** | **处理方式** | **触发动作** |
| 3D生成相关意图 | Generate\_3d | 继续3D生成流程 | 进入历史匹配/生成 |
| 无关意图 | clarify | 直接返回相应 | 不触发3D生成 |

**2. 意图分类规则：**

**（1）相关意图（触发3D生成）：**

* 建模请求："创建一个现代风格的别墅模型"

**（2）无关意图（直接返回响应）：**

* 问候对话："你好"、"在吗"、"谢谢"
* 系统咨询："这个系统怎么用"、"支持哪些功能"
* 无关问题："今天天气怎么样"、"讲个笑话"
* 功能咨询："你能做什么"、"3D生成需要多久"

### 4.3 历史记录匹配复用机制

**1. 相似度计算**

基于关键词，采用Jacard进行计算和比较，如果阈值>0.9，进入历史输出节点，否则，调用3D模型API进行生成。

通过计算当前用户请求与历史记录的相似度，判断是否可以直接复用历史生成的3D模型，从而减少对trippAI的调用，提升响应速度。



1. **历史记录存储/状态管理：**

历史记录数据结构定义如下：



通过以上设计，能够高效地存储和复用历史生成记录，从而显著提升系统性能。

## 五.结果评估功能

### 5.1 现有评估指标：

目前系统采用双维度评估体系：

1. **性能指标：**

* **生成时间**：从请求到模型生成的完整耗时（秒）
* **响应速度**：系统处理效率的直接体现

**2，用户反馈：**

* **用户满意度**：0-10分制主观评分
* **使用体验**：用户对生成结果的直观感受

### 5.2 当前系统展示

前端页面的结果评估系统展示如下：



综合得分按照生成时间和用户满意度权重各为0.5，加权融合，最终得到总分。

### 5.3 用户满意度细化

当前系统采用单一的用户满意度评分，未来将细化为多维度评估体系，从五个核心维度全面衡量用户体验：

* **准确度维度：**将评估生成结果与用户描述意图的匹配程度，关注模型是否准确反映了用户输入的关键词和设计要求。这一维度直接关系到系统的核心功能实现质量。
* **质量维度：**重点考察模型的技术细节表现，包括模型结构的完整性、纹理贴图的质量、几何细节的丰富程度等。这一维度反映了系统输出的技术水准和精细程度。
* **美观度维度：**从艺术和设计角度评估模型的视觉表现，包括比例协调性、色彩搭配、整体构图等美学要素。这一维度体现了系统在创造性表达方面的能力。
* **可用性维度**评估模型在实际应用场景中的适用性，包括模型的文件格式兼容性、优化程度、在目标平台上的运行性能等。这一维度关注模型的实用价值。
* **创新性维度**衡量设计的独特性和创造性，评估系统是否能够超越常规模式，提供具有新颖性和独创性的设计方案。

每个维度将采用独立的评分机制，并分配不同的权重系数，最终综合计算得出整体满意度得分。这种细化评估不仅能够更准确地反映用户真实感受，还能为系统优化提供具体的方向性指导。

### 5.4 性能指标扩展

在现有生成时间指标的基础上，性能评估体系将从三个层面进行系统性扩展：

* **时间性能层面：**将细化为多个子指标，包括文本分析处理时间、模型生成核心耗时、渲染输出时间以及系统响应延迟。这种细分有助于精准定位性能瓶颈，为针对性优化提供依据。
* **资源效率层面：**将引入计算资源使用情况的监控，包括GPU内存占用率、CPU利用率、显存使用效率以及磁盘I/O性能。这些指标反映了系统在硬件资源利用方面的优化程度，对于成本控制和规模扩展具有重要意义。
* **系统稳定性层面：**将增加并发处理能力、错误发生率、服务可用性以及负载均衡效果等监控维度。这些指标确保系统在高负载条件下的可靠运行，为用户提供稳定的服务质量。

扩展后的性能指标体系将采用多级加权算法，综合考虑不同指标的重要性差异，生成综合性能评分。同时，系统将建立性能基准线，通过历史数据对比分析性能变化趋势，为容量规划和系统升级提供数据支持。

### 5.5 生成质量自动评估

为了实现更客观、全面的质量评估，系统将引入自动化的质量检测机制，从技术角度对生成模型进行多维度量化评估：

* **几何质量评估：**将自动检测模型的网格结构完整性，包括多边形分布的合理性、法线方向的一致性、模型是否具备水密性等基础几何特性。这一评估确保模型在技术层面的正确性和可用性。
* **纹理质量评估：**通过分析贴图分辨率、UV展开效率、纹理细节丰富度等指标，量化模型在视觉表现方面的技术水平。这一评估关注模型表面细节的质量和真实感。
* **结构完整性评估：**重点检查模型的拓扑结构合理性，检测是否存在悬浮组件、结构冲突、比例失调等设计问题。这一评估确保模型在结构设计方面的专业性和合理性。
* **语义准确性评估：**采用计算机视觉和深度学习技术，分析生成模型与输入关键词的语义匹配程度。通过特征识别和模式匹配，量化模型在表达用户意图方面的准确度。
* **美学协调性评估：**基于设计原则和美学理论，自动分析模型的色彩搭配、构图平衡、风格一致性等视觉要素。这一评估从艺术角度提供质量反馈。

自动化质量评估系统将建立标准化的检测流程和评分体系，为每个生成模型生成详细的质量报告。这种客观评估不仅能够补充用户主观评价的不足，还能在用户评分缺失的情况下提供可靠的质量参考，同时为模型生成算法的持续优化提供数据驱动的基础。