

《软件体系结构》课程大作业

基于大模型的软件架构风格智能助手

一、作业目标

设计并实现一个软件体系结构风格智能助手，要求结合大语言模型、智能体（Agent）和微服务技术，实现以下核心功能：

- 需求分析：**接收用户自然语言描述的软件功能需求，进行语义理解与关键特征提取
- 架构推荐：**根据需求分析结果，推荐至少 3 种候选体系结构风格（需包含主流架构如分层、微服务、事件驱动等）
- 决策支持：**通过多维度对比分析，给出最终推荐架构风格及其优缺点评估报告
- 知识进化（进阶）：**支持架构知识库的持续扩展与案例学习能力

二、技术要求

1. 核心架构约束

- 建议采用**微服务架构**实现系统主体
- 至少包含 3 类智能体（如需求解析 Agent、架构匹配 Agent、评估生成 Agent）
- 需集成至少一个大语言模型（如 DeepSeek、通义千问等）作为底层大模型支持

2. 功能模块要求

模块名称	技术要求
需求理解模块	实现非结构化文本的特征提取（如并发需求、数据流特征、部署约束识别等）

知识库模块	构建可扩展的架构风格知识图谱（需包含≥10 种架构风格的属性及适用场景）
推理决策模块	实现基于规则引擎与 LLM 协同的混合推理机制
可视化模块	提供架构风格对比矩阵可视化界面（需支持架构拓扑图渲染）

三、交付要求

1. 开发文档

- 需求规格说明书（含 AI 系统特有需求分析）
- 架构设计文档（重点说明微服务划分、智能体协作机制、LLM 集成方案）
- 系统测试报告（需包含典型场景测试案例）

2. 系统实现

- 可运行的演示系统（需提供 Web API 接口）
- 核心代码（需包含智能体协作逻辑与 LLM 调用实现）
- 测试数据集（≥20 个典型需求场景案例）

3. 答辩材料

- 5 分钟系统演示（需展示从需求输入到架构推荐的完整流程）
- 15 分钟架构设计专题讲解（建议使用 C4 模型或 UML 进行架构表达）
- 5 分钟问答（重点回答技术选型决策依据）

四、评分标准

考核维度	权重	评分要点
需求分析	15%	对 AI 系统不确定性的处理方案、非功能需求定义合理性
架构设计	30%	微服务划分合理性、智能体协作机制创新性、LLM 集成方案的工程化实现

系统实现	25%	核心功能完成度、代码质量、异常处理机制
测试验证	15%	测试案例覆盖度、决策结果可解释性验证
答辩表现	15%	架构设计讲解深度、系统演示流畅度、团队协作体现

五、时间地点安排

上机环节进行验收与演示工作。

六、技术建议

- 1. 架构设计提示：
 - 推荐采用 LLM+知识图谱双驱动架构
 - 可参考 LangChain 框架实现智能体协作
 - 建议使用 Neo4j 进行架构知识图谱存储
- 2. 典型挑战应对：
 - 处理需求描述的模糊性：可结合 Few-shot Prompt Engineering
 - 保证架构推荐的可靠性：建议采用规则引擎进行结果校验
 - 提高系统响应速度：考虑 LLM 结果缓存机制
- 3. 创新方向建议：
 - 增加架构决策溯源功能
 - 实现架构模式组合推荐
 - 开发架构重构建议模块

七、参考案例

输入需求: "开发一个跨平台的即时通讯系统，要求支持万人同时在线，需要保证消息的实时性和可靠性，后期可能需要快速扩展视频通话功能"

预期输出:

- 1. 推荐架构: 事件驱动架构（核心推荐）、微服务架构（备选）
- 2. 推荐理由:
 - 高并发场景下的事件异步处理能力

- 松耦合特性便于扩展视频通话模块

3. 优缺点分析：

- √ 优点：高吞吐量、模块解耦、扩展性强
 - × 缺点：事件溯源实现复杂度高、调试困难
-

本作业设计强调 **AI 时代的架构设计思维转变**，重点培养学生在以下方面的能力：

1. 处理非确定性需求的架构决策方法
2. 传统架构模式与 AI 技术的融合设计
3. 复杂智能系统的工程化实现能力

建议在项目启动前了解 **Compound AI System** 文献，并学习大模型智能体相关知识点。