

# 论基于架构的软件设计方法及应用

## 【摘要】

2019年3月，我单位联合某高校研发了《程序在线评测比赛考试系统》。系统以程序代码在线提交自动评测功能为核心，分为题库模块、评测机模块、实验作业模块、考试模块、比赛模块、抄袭判定模块、用户管理模块等，支持对接教务平台。在项目中我担任系统架构师，负责架构设计工作。

本文以该系统为例，主要论述了基于架构的软件设计方法在项目中的具体应用，着重从架构需求、架构设计、架构实现三个阶段展开介绍。在架构需求阶段，通过用户访谈、问卷调查、现场观摩、构造原型的方式全面获取了需求；在架构设计阶段通过UML模型中的4+1视图来对系统的架构进行建模；在架构实现阶段，对系统构件进行了获取、开发和组装。最终系统顺利上线，获得了用户的一致好评。

## 【正文】

笔者在一个专为高校建设计算机专业智能教学一体化平台的单位任职，过往成果有《计算机组成原理仿真实验系统》等。2019年3月，我单位联合某大学研发了《程序在线评测比赛考试系统》项目（以下简称“OJ系统”），以取代原有传统的编程上机考试平台。

系统以程序代码的在线提交自动评测功能为核心，主要分为题库模块、评测机模块、实验作业模块、考试模块、比赛模块、抄袭判定模块、用户管理模块等。题库模块主要负责试题和测试用例的管理，用户根据试题要求编写程序代码提交到系统，系统将测试用例与程序代码发送给评测机模块，由评测机自动编译、执行、判分，并将结果发送给其他相关模块进行统计；实验作业模块用于在线布置作业，从题库中选取试题，设置截止日期等要求；考试模块用于学生在线考试，按教师预先设置的参数自动从题库随机抽题生成试卷，以及向教务平台上传考试成绩；比赛模块主要用于ACM竞赛的培训；抄袭判定模块用于鉴定代码与他人代码雷同率；用户管理模块负责用户信息的管理。在这个项目中，我担任了系统架构师的职务，主要负责系统的架构设计相关工作。

基于架构的软件设计方法（ABSD）包括架构需求、架构设计、架构文档化、架构复审、架构实现和架构演化六个阶段。架构需求阶段明确用户对系统在功能、行为、性能、设计约束等方面期望，包括需求获取、标识构件和架构评审；架构设计阶段根据需求生成并调整架构决策，包括提出架构模型、映射构件、分析构件相互作用、产生架构和评审；架构文档化阶段对架构设计分析与整理，产生架构规格说明书和架构质量说明书；架构复审阶段评价架构能否满足需求与实现质量属性、层次构件划分是否合理，标识潜在的风险，及早发现设计中的缺陷错误；架构实现阶段对架构进行实现，包括架构分析与设计、构件实现、组装和系统测试；架构演化阶段主要解决开发中用户需求变更问题，包括架构演化计划、构件变动、更新构件相互作用、构件组装测试与技术评审。

系统使用基于Spring Cloud的微服务架构开发，本文着重从架构需求、架构设计、架构实现三个阶段，论述基于架构的软件设计方法在项目中的具体应用、遇到的问题和解决方案。

## 1. 架构需求

在此阶段遇到的主要问题是 OJ 平台需要支撑全校编程课程的教学活动，这就需要有效、快速地全面获取需求。在需求的前期阶段，我们采用了用户访谈和调查问卷结合的方式，把需求调研团队分成了几组，分别进行需求收集。通过与课程教学组长的详细沟通，我们对 OJ 系统的主要业务功能、用户角色等有了整体、全面的了解。之后又制作了调查问卷表格，下发给各位任课教师，经过统计整理后，我们获悉了编程课程的教学活动、过程细节。在需求的中期阶段，我们在课程教学组长的协助和安排下，跟随任课教师，前往学校的计算机实验室，对目前编程课程的实验教学现场进行了观摩，了解了在传统实验教学方式中，学生和教师们的具体操作流程。在需求的后期阶段，我们基本上已经完成了大部分业务需求的收集，通过快速原型法构造出了一个仿真的 OJ 系统，供用户试用与反馈，让用户也参与到设计中，提供了工作流程方面、业务领域方面不可或缺的经验，也为以后项目通过验收提供了有力支持。

## 2. 架构设计

在此阶段遇到的主要问题是如何合理设计与描述软件架构。我们以 UML 模型中的 4+1 视图来对架构建模。场景视图使用 UML 模型中的用例图来进行建模，结合用户需求，在系统中划定了在校学生、任课教师、系统管理员、校外人员四类用户角色，并分别为这些角色标识了相应用例。逻辑视图使用 UML 模型中的包图来进行建模，我们经过分析，决定采用微服务架构风格开发，将系统分为前端 Web 服务、平台保障服务、业务服务三部分。前端 Web 服务由负载均衡与服务器集群结合；平台保障服务分为 API 网关、服务注册中心、监控平台，用以实现基础服务框架；业务服务分为多个微服务，实现具体业务功能。物理视图使用 UML 模型中的部署图来进行建模，系统微服务采用分布式的部署方法，每个微服务根据实际情况可在同一台物理机器或多台物理机器上同时部署多个实例。集群通过负载均衡做统一访问，部署为路由模式，系统内部网络与外部网络分属于不同的逻辑网络。在负载均衡算法的选择上，使用了最小连接法。

## 3. 架构实现

在此阶段遇到的主要问题是系统构件如何实现和组装。构件实现分为两种方式：获取现有构件、开发新构件。OJ 系统需要与教务管理和 OA 系统对接，我们使用开发商提供的 SDK 实现；微服务架构的基础框架，如 Spring Cloud、Eureka，以及评测机调用的编译器等，直接集成第三方软件实现。对于常见信息系统共同具备的用户管理、角色权限管理、日志记录、内容维护、消息中心等基本功能，通过取用单位过往项目开发中所积累的相应构件来实现。对于 OJ 系统专属功能构件，需要专门开发，我们使用了多种设计模式，例如通过装饰器模式来实现同一试题在实验作业、比赛、考试等多种使用场景下的扩展功能，通过策略模式来实现评测机对 C 语言、Java、Python 等多种编程语言的不同编译方法。构件实现完成后，我们根据不同业务类型，采用了不同的构件组装方式。例如考试服务从题库获取试题信息，采用同步消息方式；程序评测服务属于耗时操作，采用异步消息方式；涉及审批流程的业务，采用基于工作流的组装方式。

## 【总结】

系统自 2019 年 10 月正式上线已运行一年有余，在学校的日常教学考试和竞赛培训中投入使用，截至目前已有 3000 名以上的学用户、评测了 70000 份以上的程序代码，获得了单位同事领导和学校教师们的一致好评。在使用中系统也

出现了一些问题，比如评测机服务需要执行用户提交的代码，部分用户短时间内提交了大量不安全代码，导致评测机集群中所有实例全部宕机。这是由于早期在可靠性方面的需求分析不够周全，系统没有进行充分的容错设计。我们引入心跳机制、快照回滚机制，以及基于机器学习技术的预判断机制，使评测服务宕机时能够在 10 秒内自动重置恢复运行，最终解决了该问题。

实践证明，OJ 系统项目能够顺利上线，并且稳定运行，与系统采用了合适的设计方法密不可分。经过这次基于架构的软件设计方法和实施的效果后，我也看到了自己身上的不足之处，在未来还会不断地更新知识，完善本系统在各方面的设计，使整个系统能够更加好用，更有效地服务于高校师生。

---

版权声明：本文为 CSDN 博主「苏若薛」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：[https://blog.csdn.net/sinat\\_31152963/article/details/111387700](https://blog.csdn.net/sinat_31152963/article/details/111387700)