

论信息系统建模方法

【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，主要论述了软件系统建模方法在项目中的具体应用。系统采用面向对象建模方法，基于UML中的4+1视图建模，着重从场景视图、逻辑视图与物理视图三个方面介绍。场景视图以用例图分析主要用户角色与用例；逻辑视图通过包图对系统的前端Web服务、平台保障服务、业务服务功能建模；物理视图使用部署图描述微服务在硬件环境的具体部署方法。最终系统顺利上线，获得用户一致好评。

【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（以下简称为“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。

常用的软件系统建模方法有结构化建模方法、信息工程建模方法、面向对象建模方法三种。结构化建模方法以过程为中心，用于分析一个现有的系统以及定义新系统的业务需求，创建的模型为数据流图（DFD），适合流程较为稳定的系统；信息工程建模方法以数据为中心，但过程敏感，强调在分析和研究过程需求之前，首先研究和分析数据需求，创建的模型为实体联系图（ERD），主要用于数据建模；面向对象建模方法将数据和过程集成到对象中，创建的模型为对象模型，通过统一建模语言（UML）描述，定义了几种不同类型的模型图，以对象的形式共建一个信息系统或应用系统。

OJ系统使用微服务架构开发，基于面向对象建模方法中的4+1视图建模，建模工具为Rational Rose，描述语言为UML，这里着重从场景视图、逻辑视图、物理视图三个方面展开介绍。

1. 场景视图

场景视图使用UML模型中的用例图来进行建模。OJ系统功能主要面向高校

学生程序设计语言的在线学习、考试、比赛，我们经过分析，结合用户的需求，在系统中划定了四类用户角色，这些角色分别为：在校学生、任课教师、系统管理员、校外人员。在校学生用户是 OJ 系统学生端的主要使用者，学生涉及的主要用例有：登录系统、提交代码、自由练习、参加考试、提交实验作业、参加比赛、信息维护、查看系统帮助、交流讨论等；任课教师用户属于 OJ 系统的管理者，教师涉及的主要用例有：登录系统、信息维护、班级管理、助教管理、学生管理、考试管理、题库管理、实验作业管理、比赛管理、代码查重、论坛管理等；系统管理员拥有 OJ 系统最高的系统权限，在具备学生与教师所有用例的基础上，还增加了教师管理、评测机管理、系统管理、服务器管理等用例。校外人员用户主要面向社会以及其他高校的编程学习爱好者，仅具备登录系统、提交代码、自由练习、参加比赛、查看系统帮助五种用例。

2. 逻辑视图

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，解决前台界面并发问题；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，所有业务服务都注册到服务注册中心；业务服务分为多个微服务，实现具体业务功能，解决协同问题。当用户通过网络访问系统时，首先会访问到前置的负载均衡服务器，负载均衡服务器会将请求转发到前端网站的集群，前端网站通过发起 http 请求来和后端交互。API 网关收到前端的请求，会从服务注册中心根据当前请求，来获取对应的服务配置，随后通过服务配置再调用已注册的服务。当后端微服务存在多个实例时，将采取负载均衡的方式调用。后端微服务通过同步消息、异步消息、工作流三种方式协同工作，完成具体业务功能。服务监控平台注册到服务注册中心，获取所有后端业务服务的运行状态信息。

3. 物理视图

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，每个微服务单独编译、打包、部署，基于 Docker 容器连同运行环境一起封装，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来提高系统的性能。服务启动后会将自身信息注册到已部署好的分布式服务注册中心，所有客户端请求一律进入路由网关，路由网关根据请求通过服务注册中心来进行服务发现，然后通过反向代理的方式调用具体服务。在分布式部署多个服务的基础上，前端服务器集群通过 Nginx 负载均衡服务器做统一代理访问，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，以实现系统内部与外部网络的隔离。在负载均衡算法的选择上，使用了最小连接法，每当客户端的请求来临时，任务分发单元会将任务平滑分配给最小连接数的微服务节点，这样的部署方法以廉价且透明的方式扩展了服务器和网络的带宽，可以大大提升系统的并发量，同时保证系统整体的稳定性和可靠性。

【总结】

通过 4+1 视图的场景视图、逻辑视图和物理视图等建模方式，对系统进行了详细的分析，为系统的设计和接下来的项目开发提供了有力的支持。

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 名以上的学生用户、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。不可避免的，我们在设计过程中，也存在一些问题和不足，不少开发人员在实现过程中有时还是

习惯于原有的结构化设计方法，对 4+1 视图模型的使用有些抵触。而且，这些视图在应用过程中，往往不是单独使用，需要多个视图综合运用。这方面，我们还缺少相关的经验。

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软件系统建模方法密不可分。经过这次 4+1 视图的建模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「苏若蘼」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/111387604

樊樊的资料库，仅供个人学习

樊樊的资料库，仅供个人学习