民航售票系统架构评审文档

**组长：袁谨**

**组员：王晶、蒙国儒、周魁龙、吴波**

**2016.10.20**

目录

[民航售票系统架构评审文档 1](#_Toc464809399)

[0.引言 2](#_Toc464809400)

[编写目的： 2](#_Toc464809401)

[背景： 2](#_Toc464809402)

[定义： 2](#_Toc464809403)

[三层架构软件设计（MVC） 2](#_Toc464809404)

[ATAM架构评审模式 2](#_Toc464809405)

[参考资料： 3](#_Toc464809406)

[第0阶段：合作关系及准备工作 3](#_Toc464809407)

[第1阶段：评估阶段 3](#_Toc464809408)

[项目产品立项表述： 4](#_Toc464809409)

[架构表述： 4](#_Toc464809410)

[架构方法分类： 4](#_Toc464809412)

[生成质量属性效用树： 5](#_Toc464809413)

[初步分析架构方法： 6](#_Toc464809414)

[第2阶段：评估阶段 6](#_Toc464809415)

[集体讨论并确定场景的优先级 6](#_Toc464809416)

[分析架构方法 7](#_Toc464809417)

[结果表述 7](#_Toc464809418)

[第3阶段：后续阶段 7](#_Toc464809419)

[附录 8](#_Toc464809420)

[拟采用架构评审方法中的ATAM方法 8](#_Toc464809421)

# 0.引言

## 编写目的：

本文档的编写目的是对民航售票系统架构设计进行简略的评审，为后继的详细项目设计等工作提供参考和依据，本文档主要描述的内容有：

* 表述
* 调查和分析
* 测试
* 形成报告

本文档的预期读者为：系统设计人员、测试人员、用户及其它有权限查阅本文档的相关人员。

## 背景：

* 系统名称：民航售票系统
* 任务提出者：袁谨、王晶、吴波、蒙国儒、周奎龙
* 开发者（承接单位）：开发小组
* 用户：网上订购机票的人

## 定义：

### 三层架构软件设计（MVC）

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

### ATAM架构评审模式

Architecture Tradeoff Analysis Method(构架权衡分析方法）。他是评价软件构架的一种综合全面的方法。这种方法不仅可以揭示出构架满足特定质量目标的情况，而且（因为它认识到了构架决策会影响多个质量属性）可以使我们更清楚地认识到质量目标之间的联系——即如何权衡诸多质量目标。

## 参考资料：

Software Architecture in Practical（第三版）

# 第0阶段：合作关系及准备工作

此次对项目的评估方法经小组协商讨论是采用ATAM架构评估综合方法。待评估的项目系统为民航售票系统。这是一个基于B/S的体系的常见应用，能够网络的优势在多个平台进行浏览管理。对其进行架构评估主要有如下几个原因：

1. 在架构搭建的过程中一定会碰见许多一致或者未知的问题和困难，如果在核心功能模块或者架构本身的设计根本上出现缺陷，尽早的发现对于晚发现，甚至完成项目后才发现的综合成本要低得多；
2. 由于该架构面向多个用户多平台，因此要有足够的健壮性，稳定性，可拓展性以及可修改性；

3.由于该系统借助了网络的传播性，可以随时随地的对系统进行管理和维护，但是网络的泛滥使得网络环境总是充斥着有意无意的攻击，为了避免系统所部属的服务器沦为肉鸡的下场，或者内部数据被恶意破坏造成重大损失，所以系统应保证相对的安全性，使得入侵者所花费的入侵成本>入侵系统的获利成本或客户损失。

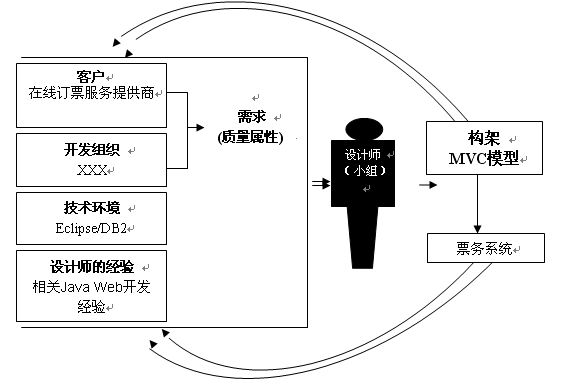
主要目标协商完成后，我们组成了一个虚拟的一人多角色的评估小组，成员列表如下：

|  |  |
| --- | --- |
| 成员 | 角色 |
| 王晶 | 评估小组负责人、提问者 |
| 袁谨 | 评估负责人、提问者、架构设计师 |
| 吴波 | 场景书记员、进展书记员 |
| 蒙国儒 | 数据收集人、提问者 |
| 周魁龙 | 领域专家、资料员 |
| 周魁龙 | 评估总结者、时间管理者 |

表1：评估小组成员列表

# 第1阶段：评估阶段

## 项目产品立项表述：



## 架构表述：

(1) 与构架商业周期的关系

## 

## 架构方法分类：

进行了架构表述后，评估小组列出他们曾听到的架构方法，以及那些在对文档进行评估前的评审中所了解到的方法。

## 生成质量属性效用树：

下表给出了在对民航售票系统评估期间生成的质量属性效用树，有几个质量属性求精没有与之相关的场景。这种情况经常出现，这并不是问题，对于某个质量属性，人们有时能够想出一个合理的求精，但当让他们在自己的系统的上下文中对该质量属性聚义用例进行说明时，却发现该求精实际上并不适用。

表 2：对民航售票系统进行ATAM评估的效用树表格

|  |  |  |
| --- | --- | --- |
| **质量属性** | **属性求精** | **场景** |
| 性能 | 最大负载  响应时间  吞吐量 | 最多10000~20000人同时在线（M,L）  在系统处于峰期负载时，对信息发送做出响应，进行SERVLET并发处理，信息处理响应要在2秒以内完成（H,M）  在峰期负载下，系统每秒能处理1000次界面申请（峰值人流10000~20000同时在线，平均2秒跳转一次界面）(M,L) |
| 易用性 | 熟练度操作  模糊搜索复杂度  正常操作 | 在网页人员具有一定人机交互设计的经验的情况下，设计出目的明确的界面，使得才接触此系统的用户能够不用学习或者少量的辅助信息指导下完成查票、购票等相应操作（H,L）  买机票的用户只需通过查询功能即可查找到相应的航班，并进行购买（H,H）  用户的注册，登陆以及修改个人信息方式简洁明确（M,L） |
| 可配置性 | 多浏览器支持 | 不仅支持PC端的IE,CHROME,FIREFOX,还要支持移动端的常用浏览器，比如UC浏览器，QQ浏览器（M,M） |
| 可维护性 | 管理员功能  维护管理  系统自动报错 | 提供给管理员最高的权限，使得管理员在接管系统以后能够按个人要求设置改变和操作系统（M,M）  维护人员遇到了搜索和响应时间缺陷，修复该bug，并分配bug修复（H,M）  当出现系统问题时提示相应错误以及可能的解决方案，使得管理员能够自己解决大部分问题以及跟维护人员沟通后解决（M,H） |
| 可扩展性 | 添加新功能 | 由于航班变更的时间问题，要有可拓展的功能轻松实现（L,M） |
| 可扩充性 | 功能业务的子模块 | 管理系统每个功能模块都是独立的（L,L） |
| 模块性 | 功能子集  可更换性（灵活性） | 构建一个可以与核心功能自治地运行的平台系统（H,H）  更换操作系统（H,H）  更换数据库可移植性层（H,H） |
| 可互操作性 |  | 构建一个与其他站点互相搜索的机制，给用户更完整的体验（L,L）  给页面开发者提供相应的API接口，从而满足一部分用户希望自己定制页面的方式（M,M） |

表中的场景给出了决策者所分配的优先级。在每一对有序的字母中，第一个代表能力的重要性，第二个代表设计师对实现该质量属性的难度的估计。

我们需要注意到，一些场景已经很完备了，具备了刺激、环境和响应三个部分；一些场景没有刺激，还有一些场景么有响应。在这个阶段，只要涉众能够理解场景的含义，不明确的场景说明是允许的。如果所选择的场景用于进行分析，那么该场景中的刺激和响应必须得到足够的明确。

## 初步分析架构方法：

评估小组首先分析最重要而且最难实现的场景，每次分析一个最高优先级的场景，同时我们的设计师详细地解释了构架如何支持每个场景。小组成员探查设计师用来实现场景的架构方法，把相关架构决策编成文档，一共确定了5个敏感点，3个权衡点，5个风险点以及2个机遇点。分别如下：

敏感点：各种人员对数据库的操作等级，，网上支付的安全性衔接，系统安全性，查询便捷性多样性，用户支付的便捷程度

权衡点：查询结果的响应与查询的便捷性多样性，系统的安全性与管理员的监管难度，网上支付的安全性衔接与用户支付的便捷程度

风险点：网上支付的安全性衔接，系统安全性，查询结果的响应时间

机遇点：查询便捷性，系统安全性

# 第2阶段：评估阶段

## 集体讨论并确定场景的优先级

下表给出了在本步骤中提出的某些部分感兴趣场景进行重点分析。按重点次序罗列，由于篇幅有限，有些细微场景没有列出，只列出了认为重要场景。

表 3：集体讨论确定的场景

|  |  |
| --- | --- |
| **场景号** | **场景** |
| 1 | 由于架构是采用的Oracle关系型数据库，所以需要谨防网络上常见的SQL注入攻击，比如“=1；有害SQL语句；……”以及URL地址后的参数，还有一些较为高级的注入攻击从而阻绝大多数一般的网络扫描攻击；对于特定的攻击，除了管理员加强数据库日志的维护，维护系统的安全外，采用正则表达式或者预编译语句等多种方式限制SQL攻击。 |
| 2 | 用户账户与付款的绑定，以及安全性的要求。 |
| 3 | 突然激增的流量导致服务器处理缓慢，甚至崩溃异常，要求对有害信息进行过滤。 |
| 4 | 用户访问的相关航班查询，买票，下订单，付款的流程体验。 |
| 5 | 支持新航班的录入。 |
| 6 | 如果存在多个管理员时怎么并发管理系统。 |

## 分析架构方法

在集中的讨论中，我们重点讨论了上述问题的解决方案，考虑到时间和空间的限制，我们就第一二个详细展开说明，以及分析和权衡。

## 结果表述

除了上述的解决方案，我们确立了个风险题：

1. 由于开发人员对架构缺乏一定的熟练度，我们决定边学边做，碰见一些解决不了的问题采用在文档中总结报告，记录下来，优先完成功能模块的实现环节；
2. 平台开发入门槛较低，产品容易被模仿，需要及时更新设计以摆脱竞争对手，所以应该预留系统API接口，为不管是以后管理方设计更改界面还是可以由用户自定义开发界面都能起到良好的促进作用。

## 第3阶段：后续阶段

ATAM评估的一个具体结果就是生成了最终报告，该报告包括一个有风险决策、误风险决策、敏感点和权衡点的列表。还包括一个涉及如下内容的目录：所使用的架构方法、效用树、经过集体讨论确定的场景以及所选择的每个场景的分析记录。最后，最终报告还包括由该评估小组所确定的风险主题的集合，并指出了每个风险主题所危及的商业动机。

# 附录

## 拟采用架构评审方法中的ATAM方法

对ATAM模型方法的简略描述：

软件构架的评估方法：SAAM和ATAM。这里只详细说明ATAM方法。

ATAM一种进行构架评估的综合方法，ATAM是评估软件构架的一个健壮的方法。在该方法中，项目决策者和涉众要清晰地阐述一个准确的质量属性需求列表（以场景的方式），并说明与实现每个高优先场景相关的构架决策。然后，把这些决策确定为有风险决策或无风险决策，以找到构架中任何存在问题的地方。

ATAM不是需求评估。ATAM不是代码评估。ATAM不包括实际的系统测试。ATAM不是一个准确的手段，但它识别了构架中可能存在风险的区域。这些风险包含在敏感点和权衡中。

ATAM活动的4个阶段：

在第0阶段（合作关系和准备）确定细节：人员名单，时间，地点；评估小组获取资料并进行初步了解分析。

第1阶段，评估阶段，决策者参与，小组开始信息收集与分析；耗时约1周。1～2周中断期，评估小组进一步以非正式方式了解构架。

第2阶段，评估阶段，涉众参与，分析继续；约2天。

第3阶段，后续阶段，生成最终报告，进行评估活动总结；1周。

评估阶段的步骤：

第1步：ATAM方法的表述。评估负责人向决策者表述ATAM方法，使大家理解其过程，了解角色布局。

第2步：商业动机的表述。决策者介绍系统商业动机、重要功能、各种限制（任何相关的技术、管理、经济和政治限制）、商业目标和上下文、主要的涉众、驱动因素等。

第3步：构架的表述。首席设计师或架构小组介绍构架，技术限制、所用模式等。

第4步：对构架方法进行分类。评估小组利用所有已知信息对构架方法进行分类。

第5步：生成质量属性效用树。生成质量属性效用树，捕获详细的需求信息，为每个场景分配一个级别，如（高，中），前者为重要度，后者为实现难易度，重点放在（高，高）的场景；此处场景具备刺激、环境、响应三要素就可以了。

第6步：分析构架方法。评估小组分析所有重要场景，设计师解释如何支持该场景，检查所用构架方法，分析风险点、权衡点、敏感点。

经过一段中断期，第2阶段开始，此时涉众开始参与；首先仍然需要一个对ATAM方法的介绍，并使涉众了解已有的成果。

第7步：集体讨论并确定场景的优先级。集体讨论并分析场景的优先级，以了解更广泛的涉众的想法；该过程可能产生新的场景；使用“有限票数法”投票确定每个场景的优先级——此处不考虑实现难度。

第8步：分析构架方法。分析新的高优先级的场景，构架师解释构架是怎么满足各场景的。

第9步：结果表述。总结评估结果，评估负责人展示该结果。