**模式评分：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 属性 | 优先级 | 黑板模式 | 分层模式 | 管道-过滤器模式 | 浏览器-服务器端模式 | 模型-视图-控制器 |
| 好的性能 | 5 | 3 | 3 | 4 | 5 | 4 |
| 易于修改 | 3 | 1 | 5 | 4 | 3 | 4 |
| 好的安全性 | 5 | 3 | 3 | 3 | 4 | 3 |
| 易于复用 | 2 | 5 | 4 | 3 | 3 | 4 |
| 有效的数据表示 | 3 | 3 | 3 | 2 | 4 | 5 |
| 总分 |  | 52 | 62 | 59 | 72 | 70 |

因此，我们的架构选择分层模式、浏览器-服务器端模式、模型-视图-控制器三种设计模式的混合模式，融合三种模式的优点，提高系统的性能和安全性两个重点特征。

**模式：**

1）黑板模式：

修改数据空间的结构，所有程序都被影响

可用于非确定性问题求解，启发式解决过程，具有可维护性，可重用性。但是不能确保期望结果，效率低下，回退，不支持并行，共享空间的访问需要同步。

2）分层模式：

用法：一般桌面应用程序、电子商务Web应用程序

【分层的优点】优势：重用性好、标准化支持、局部依赖、可替换性

【分层的缺点】影响多层次的修改；降低了效率；多层传递，重复的工作；层的粒度难以把握。

3）管道-过滤器模式

【管道/过滤器模式的优点】：

1、高内聚、低耦合；

2、多过滤器简单合成；

3、功能模块重用；

4、便于维护；

5、支持特定分析；

6、支持并行操作

【管道/过滤器模式的缺点】：

1、导致系统成批操作；

2、需协调数据流；

3、性能下降，实现复杂

4）模型-视图-控制器模式  
这种模式也称为MVC模式，将交互式应用程序分为三部分，

1. 模型 - 包含核心功能和数据
2. 视图 - 将信息显示给用户（可以定义多个视图）
3. 控制器 - 处理来自用户的输入

这样做是为了将信息的内部表示与信息呈现给用户并从用户接受的方式分开。 它将组件分离并允许有效的代码重用。