|  |  |  |  |
| --- | --- | --- | --- |
| 质量属性 | 属性求精 | | 场景 |
| 可维护性 |  | A6：与系统交互的银行系统、身份验证系统等外部系统提供的接口发生更改时，相关模块的修改在0.5个人月内完成（M,M） | |
| A7：当增加新的作业文件格式、音频格式时，维护相应代码的时间应不超过2小时，且受影响的代码量小于1% （H，M） | |
| A8：当系统部署环境的新特性影响系统原有功能的实现时，系统应能在0.5个人月内完成更新，且成本不超过总开发成本的5% (M,M) | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A7** | 用户上传作业文件或音频文件 | | | |
| **质量属性** | 可维护性 | | | |
| **环境** | 系统正常运行 | | | |
| **刺激** | 用户希望系统支持更多的文件和音频格式 | | | |
| **响应** | 运维人员升级系统并发布更新通知 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| 划分模块 | S1 |  |  |  |
| 抽象接口 | S2 | T1 |  |  |
| **理由说明** | 进行模块划分，实现低耦合高内聚，可以有效地将修改和影响控制在一定的范围内，减少系统维护时的代码量，增强系统的可修改性和可维护性。  目前互联网上的文件、音频格式诸多，且仍处在变化之中，想要统一文件格式当前无法实现，除了支持常见的格式外，用户很可能希望系统能支持一些其他的格式。因此需要对具体的实现进行抽象接口的设计，以增强灵活性和可修改性。 | | | |
| **相关架构图** |  | | | |

**敏感点**

|  |  |  |
| --- | --- | --- |
| # | 架构决策 | 理由 |
| S1 | 模块划分 | 将系统模块化，可以在需要修改时有效地减少修改和影响的范围，是可修改性和可维护性的敏感点 |
| S2 | 抽象接口 | 对系统进行接口的抽象，可以在需要修改时只对必要的模块进行修改而不影响其他模块，是可修改性和可维护性的敏感点 |

**权衡点**

|  |  |  |
| --- | --- | --- |
| # | 架构决策 | 理由 |
| T1 | 抽象接口 | 提高了可维护性但是也增加了设计成本 |