陈维烨个人陈述部分

·个人感受

首先总结一下使用ADD时感觉到的优点：

在进行Interaction Service元素的分解迭代时，我并不需要关注这个元素之外的其他元素对该元素本身有什么影响，也不需要考虑采用了某个Pattern之后该元素本身会对其之外的其他元素有什么影响，简而言之就是设计的关注点能够比较集中，这一好处简化了迭代的过程和设计决策的考虑。此外，使用Attribute Driven Design的另一个优势就是能够在具体实现功能性需求之前将非功能性需求考虑到系统设计中，而不需要等到在设计具体功能性需求时同时考虑非功能性需求该达到什么程度，甚至在系统功能性需求完成之后再去考虑非功能性需求如何实现，而这个时候的非功能性需求实现所需要遵循的约束就会非常繁杂，很难设计。使用了Attribute Driven Design则能够抛开选出来的重要的非功能性需求以外的限制或需求，并且能够参照各个Quality Attributes的tactics和strategies等信息来对设计进行权衡选择。简单来说使用Attribute Driven Design能够让设计系统满足ASR的大方向非常清晰，而需要考虑和协同甚至权衡的点也相对比较少，设计的思路能够比较开放。

其次是使用ADD时遇到的困难：

首先选择的待分解元素必须是个人比较熟悉的，这样才能明确的知道元素中各个组件对于Quality Attributes的要求是什么，而这样设计出来的针对该元素的Design也更可能去避免潜在的冲突。其次，不论选择了什么元素，设计者都必须对Quality Attribute的Scenario，Tactics和Strategies以及它们之间的联系、优劣等有比较全面的理解，所需要的信息量比较庞大。

·个人工作

我主要负责阅读C4系统文档，对C4系统进行一个初步的，更加细化的分解，然后找出初步的Functional Requirements，Non-functional Requirements以及Constraints，并列出初步的Scenario，根据它们涉及到的QA作简单的初步分类。

此外是第一次迭代的工作。我选择了Interaction Service元素并对其进行了第一次迭代，描述了初步的Model View，C&C View和建立了Software Elements View框架。