**考察各种图的功能**

SSD：

for场景，actor和系统的交互（与系统相关的input和output）

for one particular scenario of a use case , the events that external actors generate , their order , and inter-system events.

交互图：对象如何在单个用例中**细化** , for finish a task

showing how several objects collaborate in single use case

顺序图：强调对象之间消息传递的**时间**先后

通信图：强调有特定**结构**的对象群之间的消息传递

状态机图：①描述在一个用例下，被系统操作者组织和使用的**外部系统事件**

describe external system events that are recognized and handled by system operators in the context of a use case

②显示一个**对象**在其**生命周期**内的行为

活动图：分析用例、理解涉及**多个用例**的**工作流**、处理多线程应用。

分配职责的原则：最小化依赖

**依赖：**

描述对象之间的**变量**、**静态方法**的依赖

A依赖B

eg.A接收B类对象作为参数

eg.A调用B类的静态方法

* 例题：

Dependency association between the 2 classes described in java below.

public class A{

private ArrayList \_Bs=new ArrayList();

public A**(\_Bs：B)**{

this.\_Bs.add(\_Bs);

}

}

public class B{

…;

}

**组合composition（强·整体-部分关系）：**

16.13 三准则

（人话版：finger不能同时属于两只手、不存在游离的finger、手负责finger的创建和删除）

**聚合aggregation（has a）：**

记住栗子，车有4个轮子是聚合，即可。课本都说了聚合是没有明确语义的。

**继承inheritance**：is a 或者is a kind of

用例：功能性需求

补充规约[supplementary specification]：非功能性需求