

画图不要解释不要过程，只要结果

第一个作图题：用例图

用例是主要参与者与系统之间的交互（一系列动作），来完成他的目标。

用例：谁，想干什么事。

要给系统定义一个边界范围。

可能还需要一些外部的系统和设备来支持，在右边画出支持性参与者。

会给一个表格，说明参与者是谁，目标是什么，这个目标有什么子功能，有哪些场景，需要的外部设备有哪些。只需要按照这个表去做就行了。

举例：课程网站上面那份酒店预订的文档，给你附上链接吧：

[http://ss.sysu.edu.cn/~pml/se305/2014spring/assignment/Asg\\_RH.pdf](http://ss.sysu.edu.cn/~pml/se305/2014spring/assignment/Asg_RH.pdf)

1、明确是什么系统（系统边界的标题），大概是从文档第一段找吧

2、为了获得目标，有经验的作者都是把目标写在段落的第一行或者最后一行，告诉你使用者的意图是什么，找到这一句的动词和名词即可，如：

The Search button allows to submit the request to the system to **find** available **hotels** on the dates specified.

这个功能的目标就是为了 **find hotel**，**hotel** 不要加 **s**，也不要保留形容词，做需求的时候要简洁，要把冠词、复数等去掉，只留下动词和名词

如果有特殊情况，遇到这种特殊情况，系统会增加一个功能，这个是属于 **extend** 的（目标同样是查找酒店，只是查找的方式不同而已）：

If the City/Town location does not exist in the database, the application offers the traveler to **find** the **location** by browsing through the selections, as shown in Screenshot 2.

后面就没有什么跟 **find hotel** 有关的了，所以这个功能到此提取结束。

3、接下来讲找到酒店之后，给了一个工作流告诉你怎样去下一个订单：

workflow that informs that **making a reservation** consists of four steps: (1) search, (2) choose hotel, (3) choose room type, and (4) confirm reservation

这个功能是 **make reservation**。

这里是一个功能包含多个子功能，第一个(search)我们前面已经做了，所以这里的 **功能** 包含了后面 3 个子功能，是 **include**，要注意 **include** 与 **extend** 的区别。

（老师在黑板上写的 3 个子功能分别是：choose hotel, choose room, confirm reservation，大概原则是尽可能简洁吧）

虽然说这 3 个子功能可能可以继续细分，但是考试时间有限，老师说最多考三层，正常情况下写两层你就可以得到满分了。（老师特别提醒女生不要太追求细节，太追求细节不但不加分，反而要扣分，然后他就开始继续拿起他那独特的吹水功夫了...）

4、他说然后剩下的就是 **check out** 功能了，于是乎我在文档里面搜了一下，这个 **check out** 好像不像他说的那么好找啊...于是我仔细听一下他是怎么找出来的：“下面就要做 **check out**，**check out** 干嘛呢，这你一看 **check out hotel**，所以说这个东西呢，就是你的直觉也会告诉你的，这样划分为 3 个用例是非常合理的”

好的...直觉，恩，记住了！请练好你的直觉...

但是他最后画出来的图 **check out** 是放在 **make payment** 里面的...所以最后我画图是按照四个功能来画的，这个 **check out** 放到 **pay** 里面去了

5、接下来还有管理 **basket**、**pay**，老师说漏掉不会扣分，不要太追求细节。

整体地从参与者的角度去看这个系统，而不是从系统功能的角度去研究系统，我为

什么要用它，用它来干嘛。

注意 pay 这里用到了支持性参与者，如信用卡等

扣分标准（一个图才 10 分啊!）:

1、在图里面有“什么什么界面”“什么什么菜单”“什么什么功能”“什么什么按钮”之类的字眼，扣 10 分，哦，后来又改成一分都不给了

2、画得像流程图，不给分

其他情况下，画得像流程图就有分。

3、没有系统边界，扣 1 分

4、没有系统名称，扣 1 分

5、只出现名词却没有出现动词，扣 1 分

6、一定要有一个 include 或者一个 extend，不需要两种都有，搞不清楚二者的区别全都写 include 就好了。如果是 extend，一定是虚线箭头指向父用例，代表从父用例扩展出来，如果是 include，一定是虚线箭头指向子用例，代表包含子用例。

如果不满足上面条件，扣分，扣多少就不知道了...

7、所有的用例都必须和 actor 关联，如果没有关联线或者关联线有箭头，扣分，关联线一定是没有箭头的。

8、actor 和 actor 之间作为角色可以存在泛化，但是不能存在管理，因为整个文档没有出现“管理”这个字样。

9、如果是手机的 APP，右边的支持性参与者常用的有 device 或传感器，GPS，加速传感器，支持性参与者不能写在左边，写在左边要扣分

10、画用例是画有用的用例，不能出现“登陆”或者“login”这种对系统没有贡献的字眼

11、一个用例就是一个基本业务流程：做事过程中间是不能停顿的。比如在下订单的过程中，是不能中断的，如果下一半就关掉系统则没有成功下订单。

如果你把所有功能堆积到一个用例里面，说明你认为所有功能必须一次性完成，不能中断，这个是错误的，直接是 0 分

12、size 原则：用例不是操作，不要把每一个操作都拿出来做一个单独的用例，比如输入地点等等。

如何判断是否符合 size 原则：这个功能是否有一个单独的界面

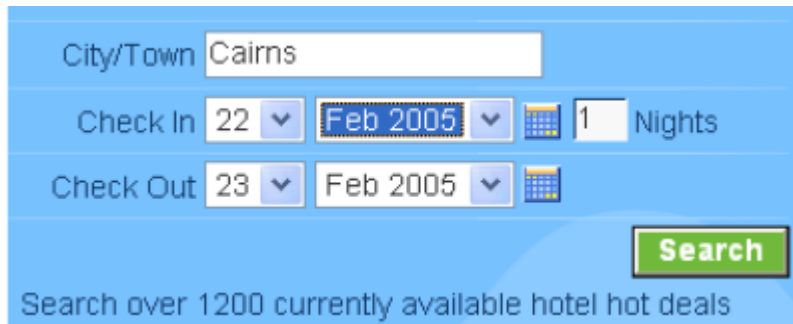
这意味着每一个用例都有一个单独的交互过程

第二个图：活动图

1、首先画好起点（一定要有起点终点，起点只有一个，终点可以有多个）

2、画活动

填写以下表单的活动：



City/Town	Cairns
Check In	22 Feb 2005 1 Nights
Check Out	23 Feb 2005
<input type="button" value="Search"/>	
Search over 1200 currently available hotel hot deals	

- 1) 输入地点
- 2) 输入 check in 日期和天数
- 3) 输入 check out 日期
- 4) 点击 search 按钮

以上写得很详细，写得详细不扣分，但是考试时间有限啊!!!

遇到这种情况，考试时就写成下面这样就好了：

- 1) 输入查询条件

至于查询条件是什么，那一般是在系统说明里面详细说的。

- 2) 提交查询表单

活动图一般是画一个用例或者一个子用例，一定要看清楚它要你画哪个用例。

假设这里是要画 find hotel 的活动图，那么就是：

- 1) 输入查询条件
- 2) 提交查询表单
- 3) 如果 city 存在，则这个用例就结束了
- 4) 如果 city 不存在，则调用 find location 子用例，不用研究怎么 find 的，文字细节以后进一步改进，这样的图一看就很清晰了

注意事项：

有箭头的线，如果有循环一定有归并节点，如果有条件的话，一定要写 guard(写在 guard 里面会自动加上左右中括号[]的，guard 在 constraint 选项卡里面)。

可能会考什么是活动状态 Activity：

活动状态用于表达状态机中的非原子的运行，特点如下：

- 1) 活动状态可以分解成其他子活动或者动作状态
- 2) 活动状态的内部活动可以用另一个活动图来表示
- 3) 和动作状态不同，活动状态可以有入口动作和出口动作，也可以有内部转移
- 4) 动作状态(Action)是活动状态的一个特例，如果某个活动状态只包括一个动作，那么它就是一个动作状态。

第三个图：状态图（最好自己百度怎么画）

状态图的对象是什么，可能是一个 system，也可能是一个用例，也可能是一个对象，一定要看清楚题目要求画什么东西的状态图。

画系统和用例的状态图一般是画它的过程，而画对象的状态图是画它的生命周期。

然后就是找状态了，一定要知道状态变量是什么，一定要能枚举状态有哪些情况。

接下来是操作。

- 1、识别状态图的对象

- 2、识别状态

考试是有组合状态的。

- 3、转换边

格式：触发事件 [监护条件] / 动作

触发事件：触发转换的事件，包括调用、触发信号、时间等

监护条件 (guard)：决定是否转换的条件，监护条件为 true 才能转换

动作：转换被激活时会发生的操作

事件一般用被动语态写出来（用被动语态写出来的叫做触发，如 onKeyPressed，一般是与代码对应的函数名相同的）。

如果程序显式告诉你，“如果什么，怎么样”，则一定要写 guard，否则要扣分

动作可以不写，写错了要扣分

对例子进行状态建模：

- 1) 起始状态
- 2) Choose hotel 状态
- 3) 经历 Hotel selected 事件（不要写成 continue 事件）之后进入 submit 状态
- 4) Confirm 状态
- 5) 终止状态

#### 第四个图：领域模型

名词法：找一堆名词，然后把这堆名词之间的关系给建立起来

名词里面有属性。要判断名词是不是概念类，是不是属性。

考试的时候是针对一个用例来画领域模型，一定要看清楚是要对哪个用例建模，没有那么多时间对整个系统建模。

- 1、先找到所有名词，判断它是类还是属性

找名词的原则（下面不要的名词标红）：

- 1) 跟 UI 相关的名词不要
  - 2) 跟 database 相关的名词不要
  - 3) 跟业务流程没有关系的名词不要，如技术相关的术语，如下面的 workflow, list
  - 4) 任何计算出来的结果，不参与业务运算，不要，如果留下了这个会扣分
  - 5) 模糊的术语一定要过滤掉
- 2、如果出现动词，扣分
  - 3、没有名词，扣分
  - 4、多重性（关联的一对多，一对一等）没有，扣分
  - 5、漏掉一两个类，不扣分
  - 6、属性，假如每一个类有七八个属性，只写一两个典型的代表即可，考试没有那么多时间
  - 7、领域模型的类不能有操作（也就是类的函数），如果写出来要扣分。
  - 8、如果有描述类，一定要画出来。

描述类是包含其他事物的信息的类。命名方式：被描述类名 Description

被描述的事物存在，并且描述独立于事物的实例

比如酒店的每一个同类型的房间价格都是一样的，它并不随着房间号的变化而变化，所以把房间描述独立出来会比较好

对下面这段话（make reservation 用例）进行领域模型建模：

Screenshot 3 shows a web page resulting from a successful search. The page shows:

workflow that informs that making a reservation consists of four steps: (1) search, (2) choose hotel, (3) choose room type, and (4) confirm reservation

summary that displays the original search criteria

search again option, which returns the customer to Screenshot 1

option to sort results by: (1) our favourites, (2) lowest price, (3) highest star rating, and (4) alphabetical

search results listing the hotels and basic information about them, as well as the continue option to proceed with the reservation

- 1) 首先看它是什么用例：**make reservation**，就先把 **reservation** 提取出来，这一定是重要的，把它放在最中间，其他的都围着它转，因为大部分会与它有关系
- 2) 看描述文档里面的单词，哪些单词与它联系比较紧密，出现次数比较多
- 3) **search criteria** 是前面输入的搜索条件，是已经存在内存里面的，表示方法为在前面加个 **m**，表示为 **mSearch Criteria**，要注意以 **m** 开头的东西，一般不要与其他类发生关联，除了 **mBasket**（购物车）可能要与 **item** 发生关联，其他都不要
- 4) 找到所有的名词后，建立它们的关联  
**include, own, has** 等等

#### 第五个图：SSD 操作协议（系统顺序图）

什么是 SSD？

通常用这个图来画一个用例场景（例如主场景或复杂的常用的场景）。

- 1、首先要画一个 **system**，前面要加个冒号，不写 **system**，扣全部分，不写冒号扣 1 分，位置放错扣 1 分。  
因为要画的是系统事件，没有系统还画什么
- 2、最左边是 **actor**（前面也要加冒号），然后是 **system**，然后就是用例的外部实体
- 3、主场景是按照最理想的情况把事情做完就可以了，不需要考虑细节
- 4、系统顺序图通常只有 3-5 个事件，如果某个事件操作很多，直接忽略后面那些细节
- 5、

**make reservation** 的系统事件：

事件不能写什么 **continue** 之类的，要写具体的意图，对应代码里面的函数名

- 1) **findHotel(mSearchForm)**
- 2) **chooseHotel(hotel)**
- 3) **chooseRoomType(room)**
- 4) **confirm(reservation)**
- 5) **addBasket()**

后置条件只能写这 3 句话中的一句或几句：

创建什么对象或删除什么对象，修改什么属性，生成什么关联

这是整个画图考试唯一需要文字的地方

#### 第六个图：包图

我想实现这一个场景，请使用 **MVC** 模式生成一个层次架构，请你用一个包图来表示这个层次架构，并把场景里面的元素填到包里面去。

有 3 个包，一个叫 **M**(模型)，一个叫 **V**（视图），一个叫 **C**（控制器）。

把每个包里面涉及到的东西添加到包里面。

**findHotel** 领域模型里面包括：**city, hotel**，这个是在 **M** 里面的，包图的 **M** 里面的元素全都来自领域模型里面。

**findHotelAction, mSearchCriteria** 属于 **C**，临时变量都属于控制层，动作的命名规则是在动作后面加个 **Action** 或者 **Controller**，一个用例就一个控制器。

**searchForm** 属于 **V**

对一个用例画包图就是把这个用例的领域模型里面的元素填到 **M** 和 **C** 包里面，然后给界面起个名字，然后写在 **V** 包里面即可。（老师没说要画包之间的关系，他自己画的图也没有画关系，所以干脆还是不要画了吧）

### 第七个图：顺序图

表示主场景是怎么实现的

BCE 方法：boundary、control、entity

注意名称都是以冒号开头，冒号不写要扣分，画下划线的是静态对象

最左边的方法是把 SSD 里面的方法 copy 过来，方法不能多也不能少，顺序图是研究系统事件是如何实现的，所以必须和 SSD 一样的事件。只是把系统的职责转移到控制器中来实现它。

findHotel 方法：

1) 根据 cityId 找到 city 名称

2) 根据 city 名称找到 hotel 的列表

不要讲找不到 hotel 的情况，因为我们研究的是主场景

3) create 页面

记住一定要简洁，遇到并行的就不管了，随便画，把意思表达出来就可以了（老师原话）。

### 第八个图：设计类图 DCD

首先把顺序图里面的类抄过来。

然后查看领域模型，把领域模型里面对应这里的类的属性 copy 过来，然后把领域模型里面的关联到这里变成实现。

补充类的方法。

控制器类方法的数量只能与前面的系统事件数量一样，系统事件有多少个就只能写多少个方法。

1) 前面的 has, own 这种关联不能留下来，如果留下来，扣分，有箭头没文字

2) 多重性要保持

3) 不要写什么 set 方法、get 方法，这是编程的问题，不是这里的问题

### 附加题：部署图

会用文字描述好整个部署。

有哪些数据库，数据库里面用什么数据，有哪些 server，server 叫什么名字，有哪些 client，client 里面包含哪些组件，它们之间是怎么通讯的。

就只要把这些翻译成部署图就可以了。

少一个组件扣一分，描述错一个扣一分。

就拿单项练习 10 来开刀吧：

对一个大学生选课系统从客户端到数据库全程部署

主要有：客户端 APP（浏览器），web 服务器，数据库服务器

额，没写怎么精细，到时候题目会给得比较细致的吧...