

软件建模

刘 钦

南京大学软件学院

真实世界就是
这么简单？

我们需要一个帮我们计算 $1+2+3...+100$ 的软件？

到底我们的需求是什么？

我们想要一个好用的
计算器软件。

为什么需要这样一个
软件？

到底什么是一个好用的
计算器？

怎么实现这个好的计
算器？

怎么知道实现的这个计算机
器软件是我们需要的那个
？怎么知道他是好用的？

现在的软件已经复杂到不
是一个能够开发出来的

。

所以，如果要求在一定的时
间，一定预算内，领导特定
一群人来求解空间？怎么办
？



How the customer explained it



How the project leader understood it



How the analyst designed it



How the programmer wrote it



What the beta testers received



How the business consultant described it



How the project was documented



What operations installed



How the customer was billed



How it was supported



What marketing advertised

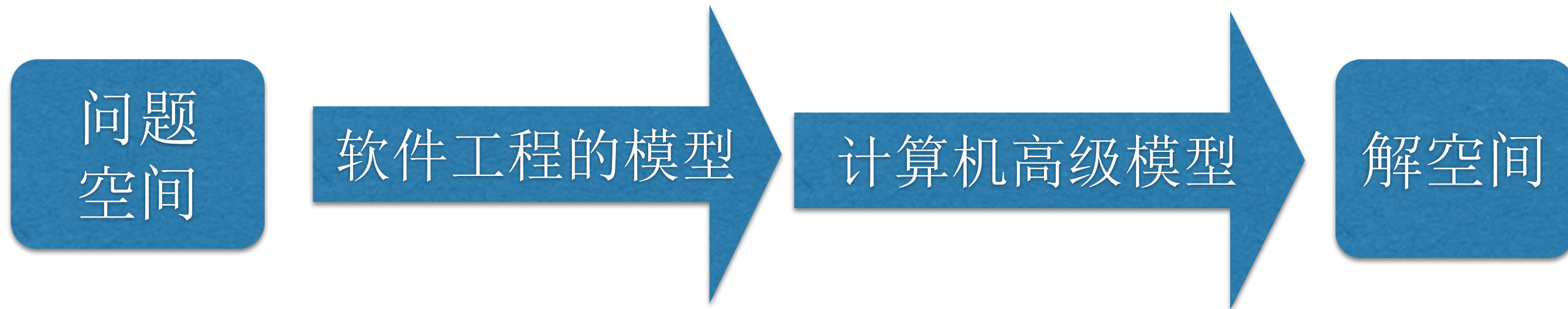


What the customer really needed

What the customer wanted

这时候你需要软件工程！

软件工程的思维



?

满足真实的需求

?

利用过程、方法和工具
管理时间、金钱、人

第零步 - 明确自己的武器

- 软件工程框架
 - 技术
 - 业务模型
 - 分析模型
 - 设计模型
 - 过程
 - 软件开发过程
- 计算机模型的进步
 - 高级语言编译器
 - 高级的模型

第一步 - 需求分析（审题）

- 做什么？

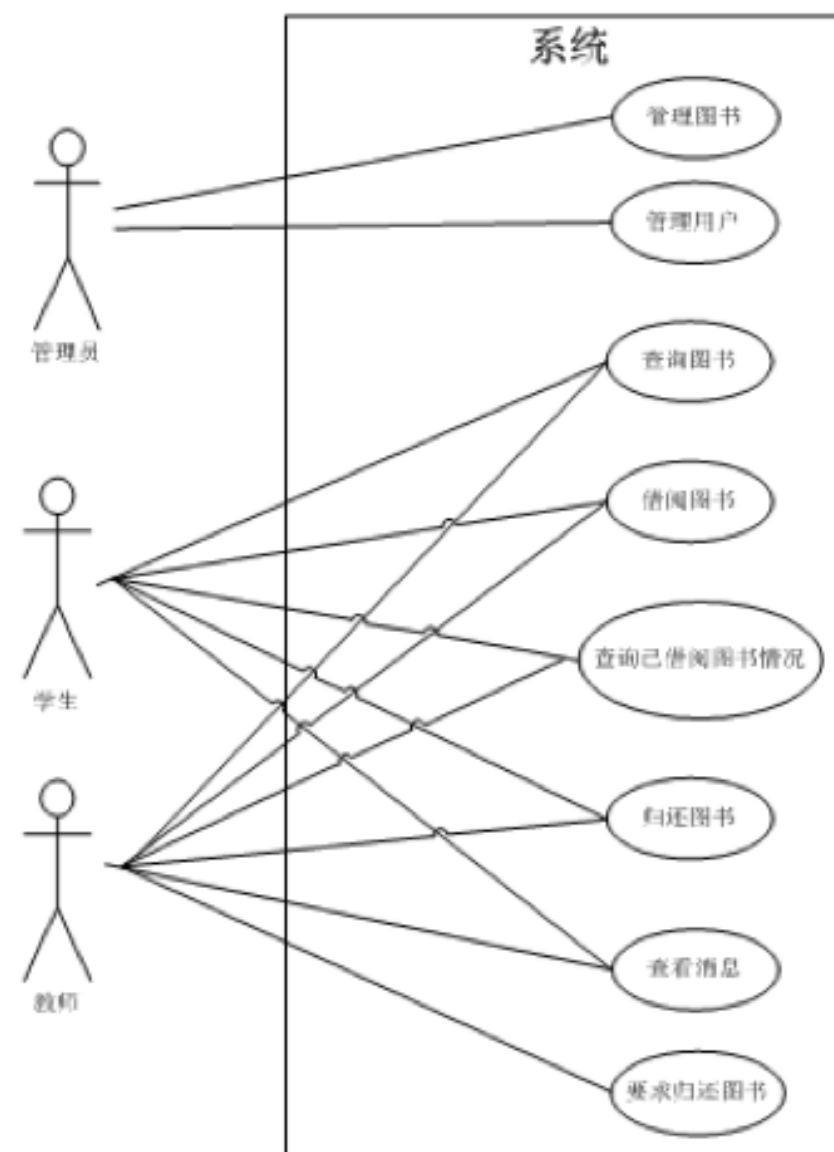
分析模型 - 业务描述

- 系统管理员
 - 身份：所有系统管理员采用相同的身份和权限。
 - 操作：
 - 管理系统管理员：可以查询、添加、修改和删除系统管理员信息。
 - 管理借阅人：可以查询、添加、修改和删除借阅人信息。
 - 管理图书：可以查询、添加、修改和删除图书信息。
- 借阅人
 - 身份：分为本科生、研究生和教师三种身份，各种身份具有不同的权限。
 - 操作：
 - 查询图书：根据图书基本信息对图书进行查询。
 - 借阅图书：对选定的图书进行借阅。其中，本科生只可借阅普通图书，最多可同时借阅**5**本；研究生可以借阅普通图书和珍本图书，最多可同时借阅**10**本；教师可以借阅普通图书和珍本图书，最多可同时借阅**20**本。
 - 请求图书：当教师希望借阅的某种图书被借空时，可以请求图书，系统将自动通知借阅该书时间最长的本科生或研究生在**7**天内归还图书。
 - 查看已借图书：查看本人当前借阅图书的情况。
 - 续借图书：图书每次借阅时间为**30**天，本科生和研究生可以续借**1**次，教师可以续借**2**次。超期的图书和被教师请求的图书不得续借。
 - 归还图书：归还本人借阅的图书。
 - 查看消息：查看图书到期提醒、提前还书通知（本科生或研究生所借图书被请求归还时）、请求图书到馆通知（教师所请求的图书归还时）等。

业务描述案例

分析模型

用例图



分析模型

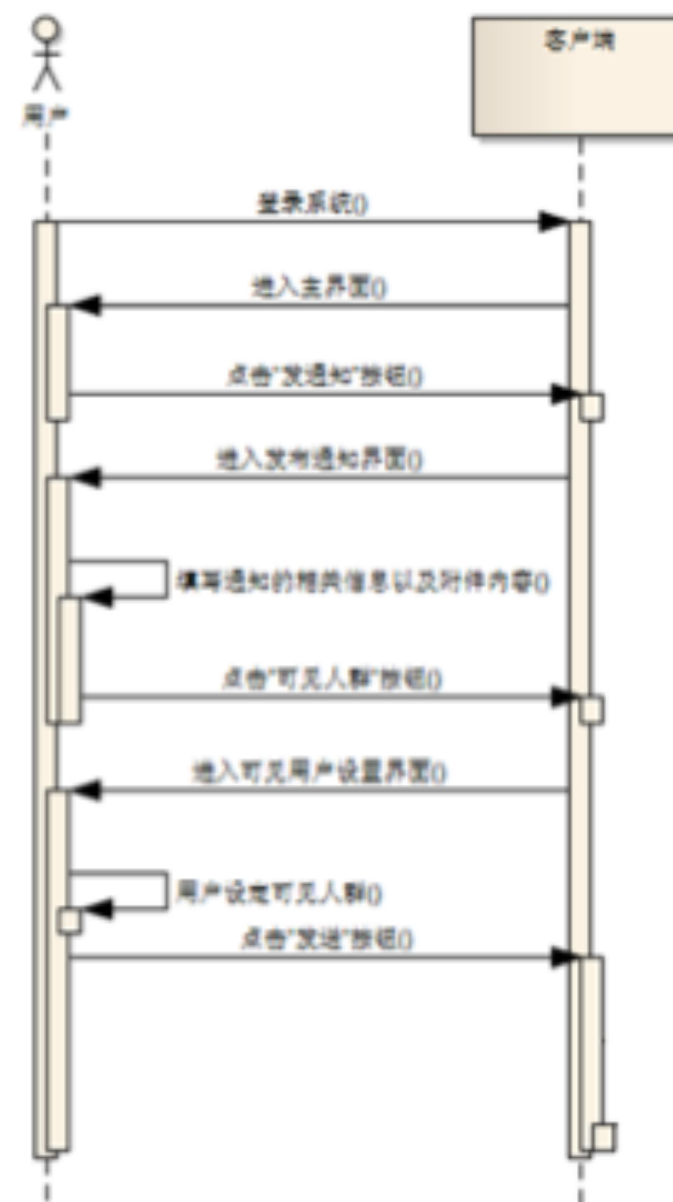
分析类图

图书类别
-图书信息列表 -图书计数
+获取图书数量() +获取单本图书() +根据 ISBN 搜索图书() +根据基本信息搜索图书() +添加图书() +删除图书()

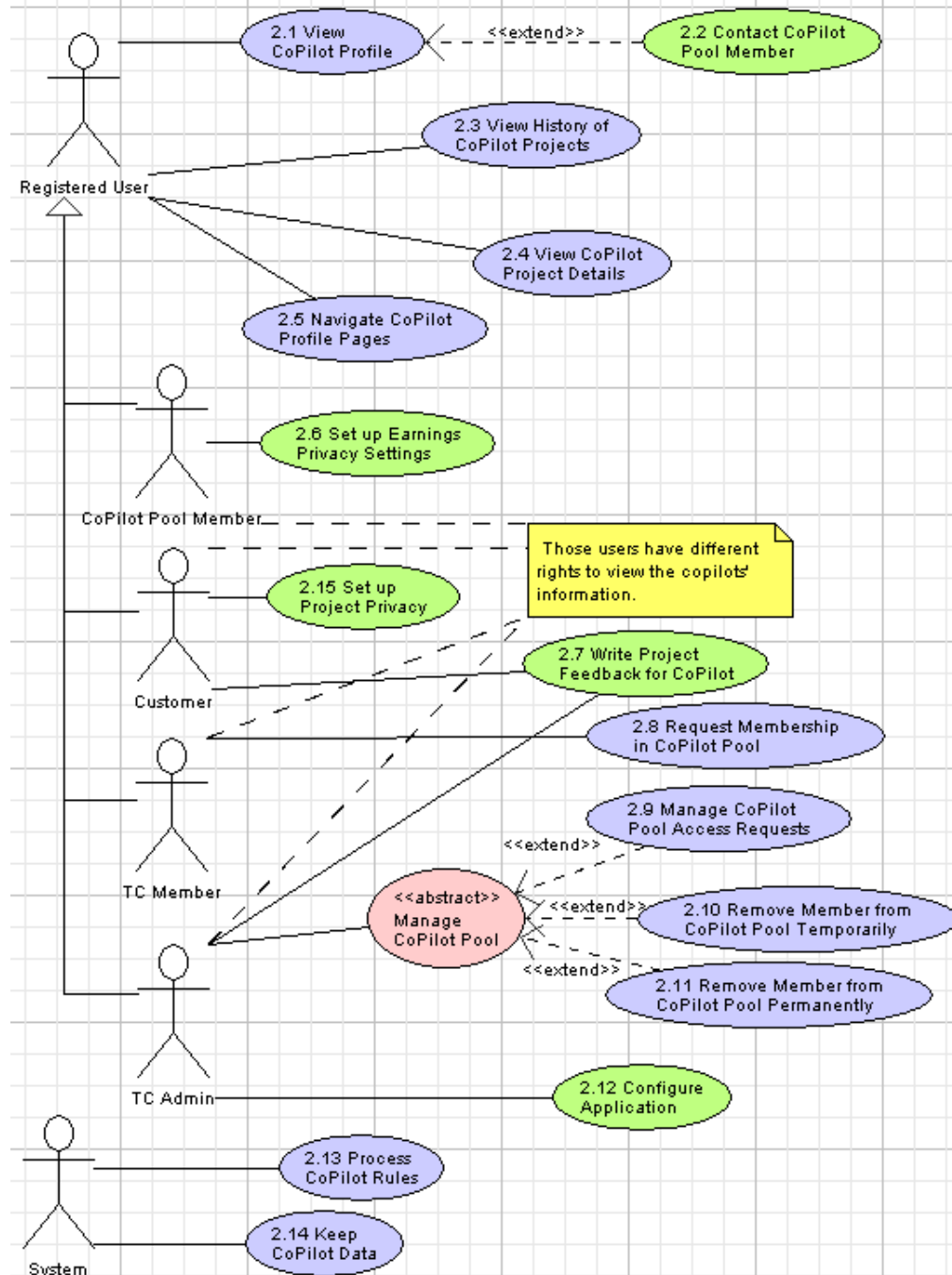
借阅人
-编号 -姓名 -密码 -借阅列表 -借阅数量 -借阅数量限制
+获取个人属性信息() +修改个人信息() +列举所借图书() +列举图书信息() +查询图书信息() +借阅图书() +归还图书() +检查所借图书状态()

分析模型

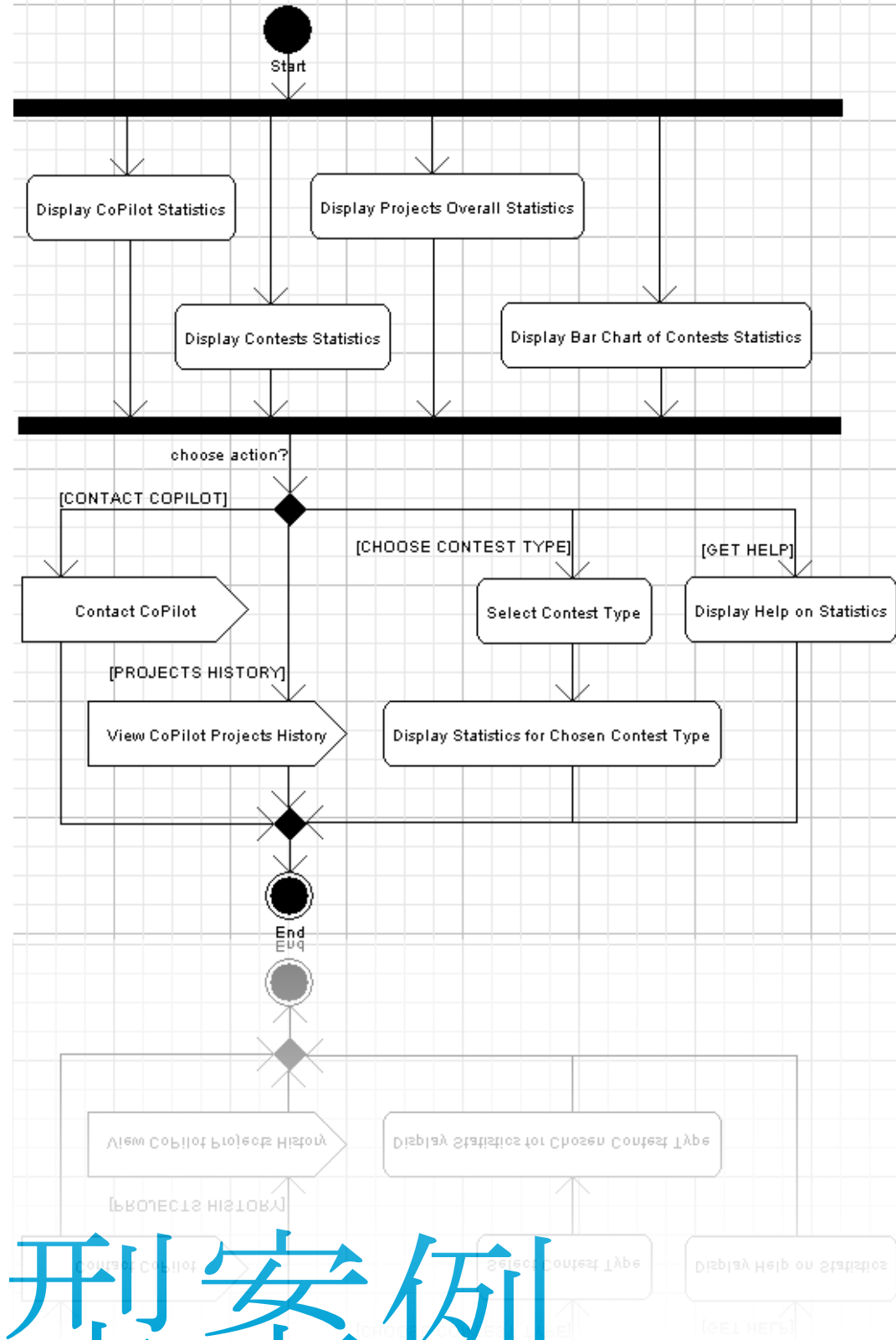
系统顺序图



Copilot Pool and Profile Pages 1.0 - Use Case Diagram



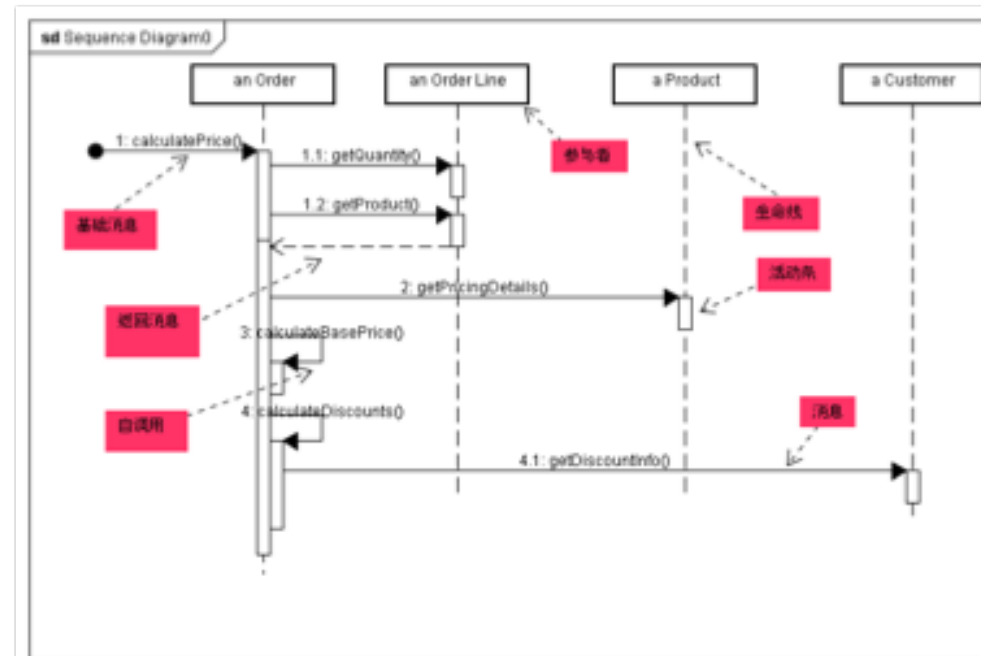
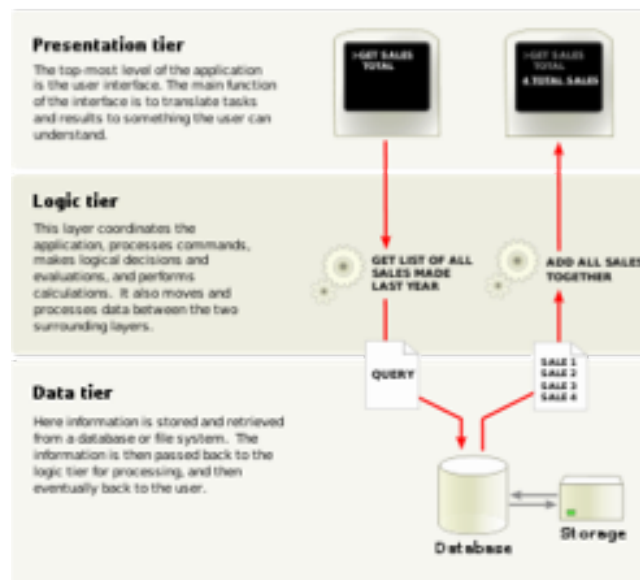
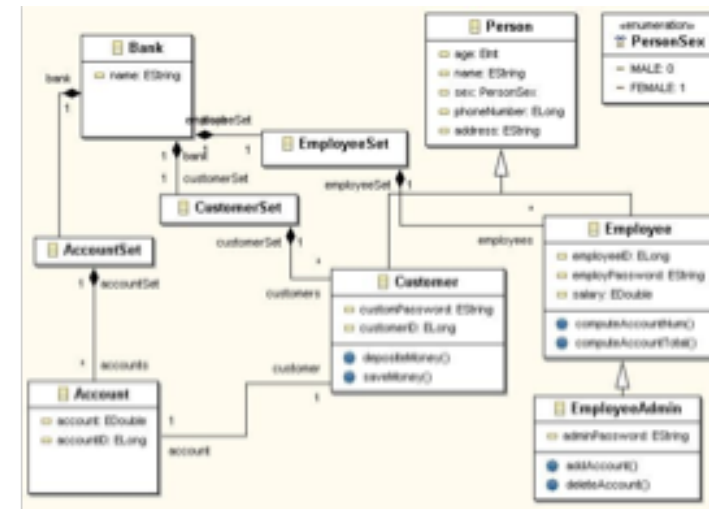
2.01.1 View CoPilot Profile Activity



分析模型案例

第二步 - 设计（建立计算机模型）

- 怎么做



设计模型

Component Diagram

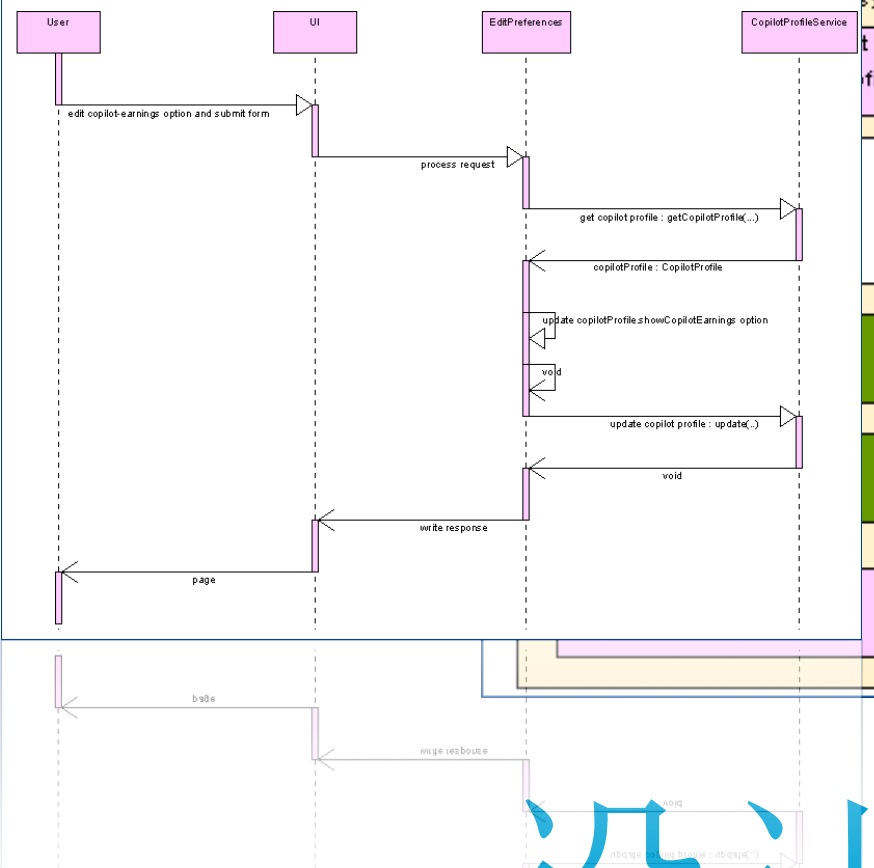
This Module

<< Component >>
Copilot Pool and Profile Request Process

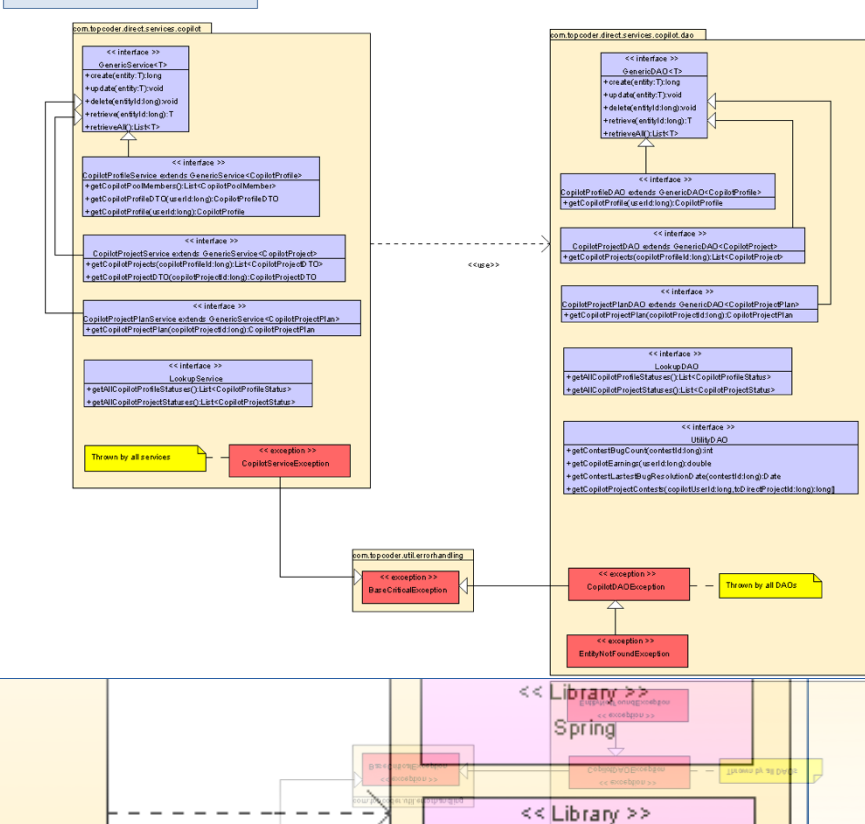
<< Component >>
Copilot Pool and Profile Services

Edit Copilot Earnings Privacy Sequence Diagram

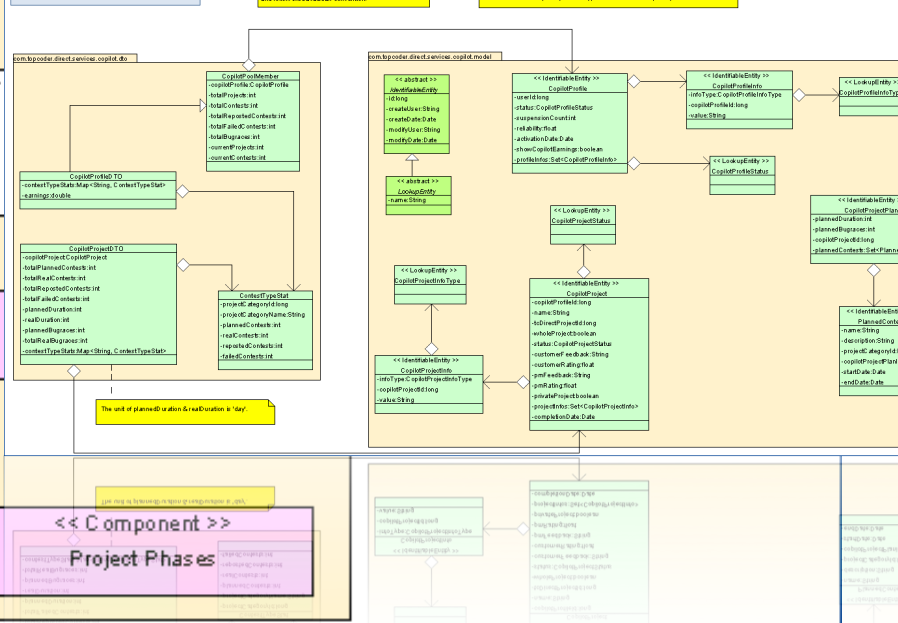
Only changed portions are shown.



Copilot Interface Class Diagram



Copilot Model Class Diagram



<< Component >>
Project Phases

设计模型案例

第三步 - 构造（制订解决方案）

- **Do it**

构造模型 - 高级语言语法

- Java
- C++
- Haskell

构造结果 - 程序本身

- `class HelloWorldApp {`
- `public static void main(String[] args) {`
- `System.out.println("Hello World!"); //`
`Display the string.`
- `}`
- `}`

第四步 - 软件测试（检验）

- **Verification** - 检查解决方案的有效性
 - whether do it right?
- **Validation** - 检查是否解决了问题？
 - whether do the right thing?

第五步 - 移交和演化（实施）

- 在真实环境中运行
- 演化新的版本

步骤

- 需求
- 设计
- 实现
- 测试
- 部署