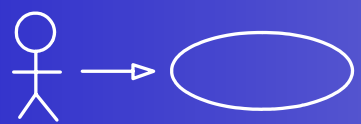
火龙果•整理
uml.org.cn

UML需求工程

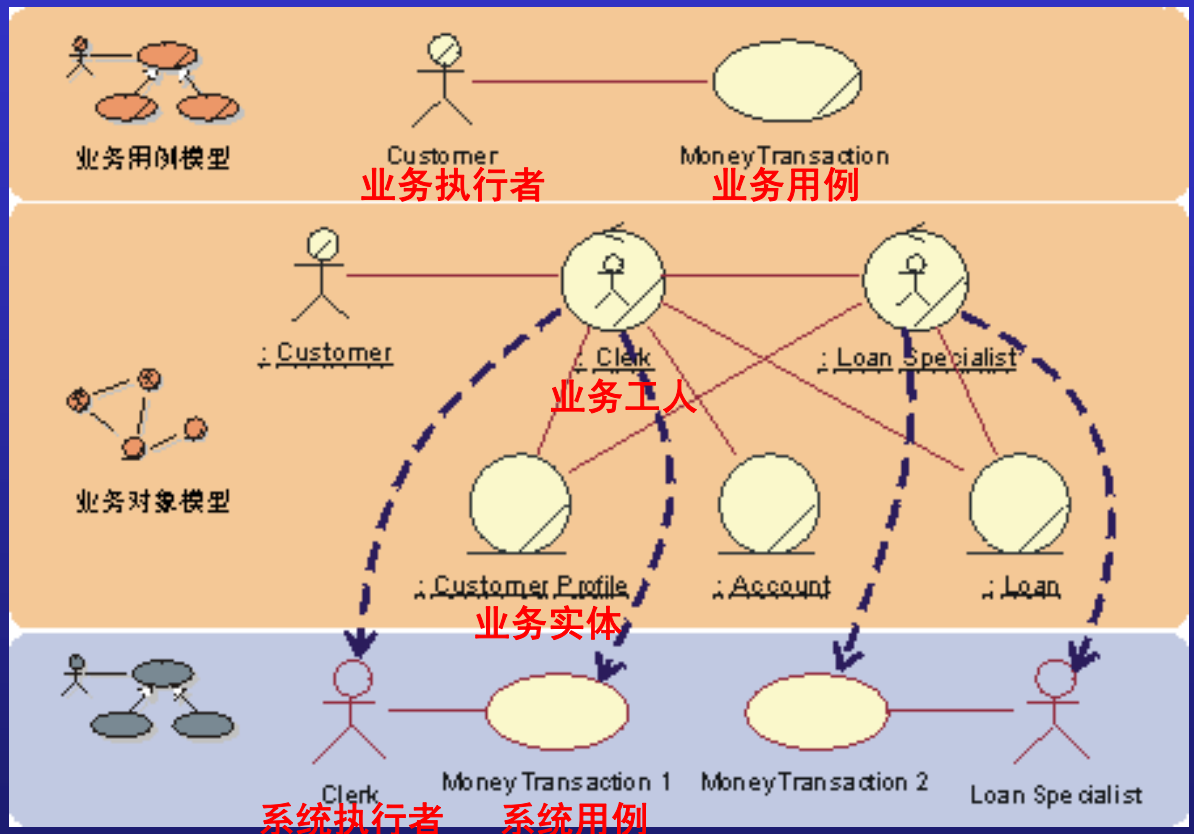
业务建模

Think



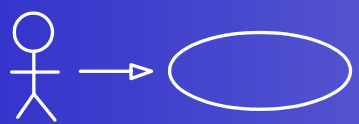
火龙果•整理
uml.org.cn

业务建模



业务建模只是辅助环节
不是所有的项目都需要
也不一定和软件开发相关

对于软件开发的作用：描述现实，帮助发现软件需求



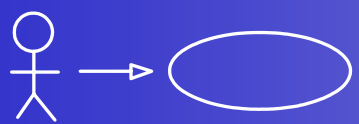
火龙果•整理
uml.org.cn

业务建模开阔眼界

酒店的思考边界？商店的思考边界？

电视机的思考边界？

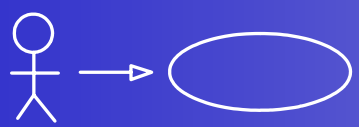
对产品研发，业务建模也很有意义



火龙果•整理
uml.org.cn

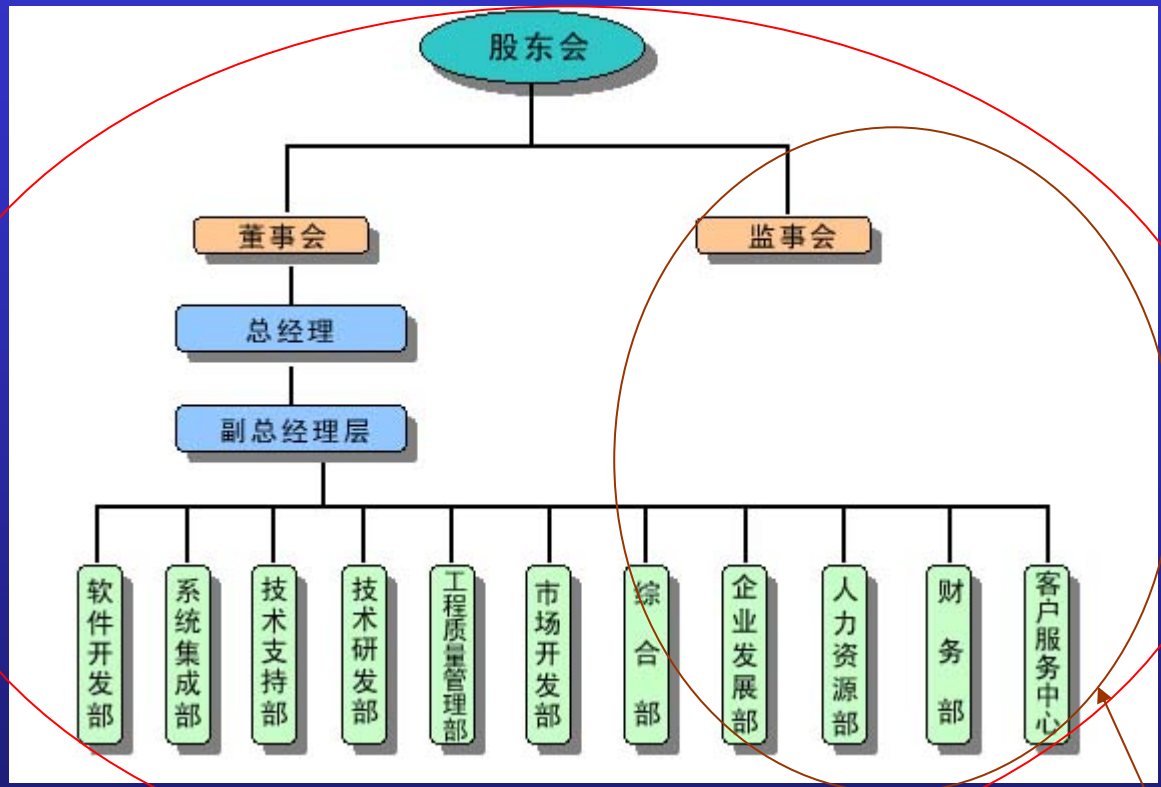
业务建模开阔眼界

——讨论和练习、项目实作



火龙果•整理
uml.org.cn

业务



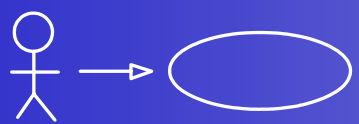
可以具体

也可以抽象

可以是整个组织

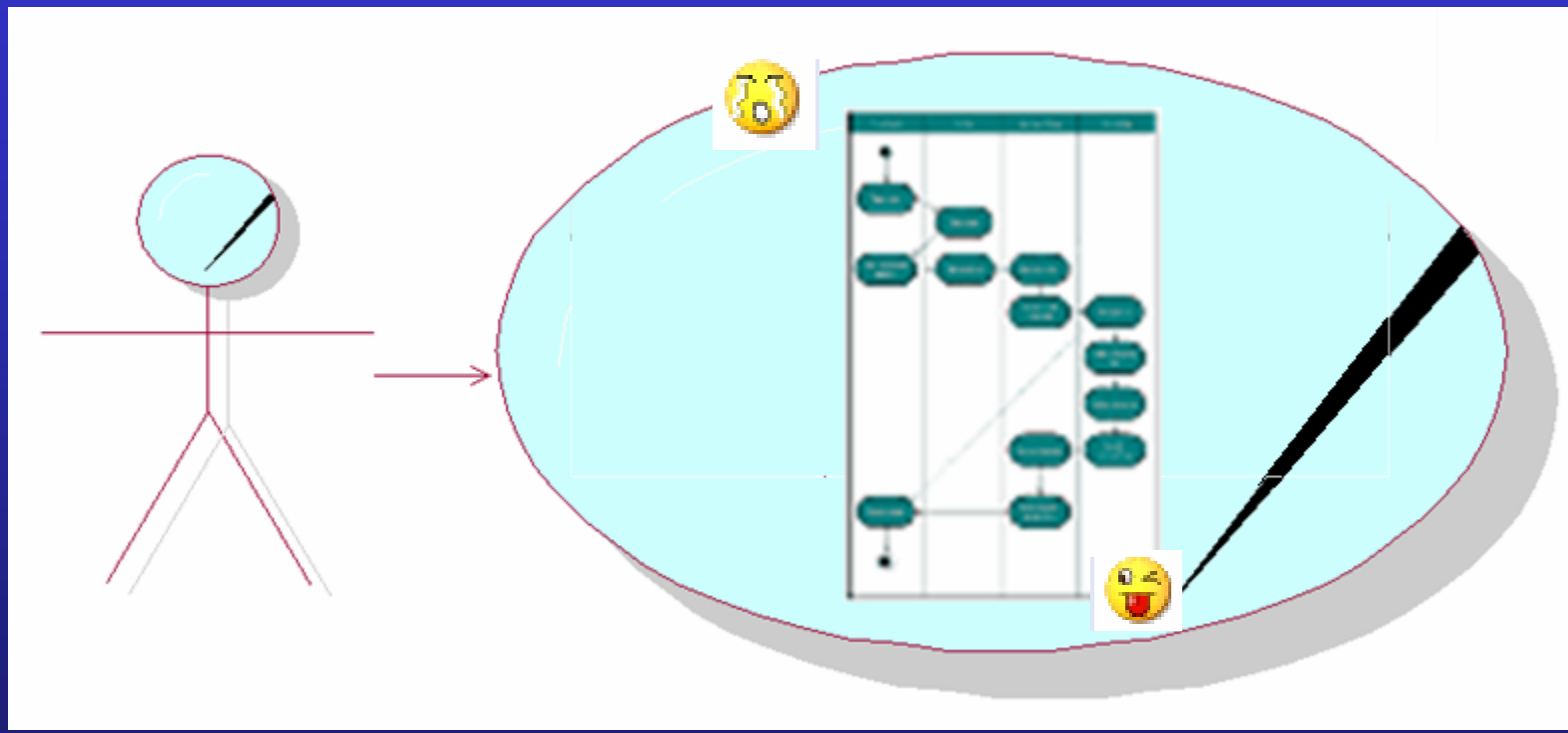
也可以是组织的一部分

需要改进的业务单元

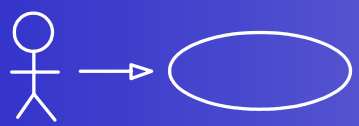


火龙果•整理
uml.org.cn

业务



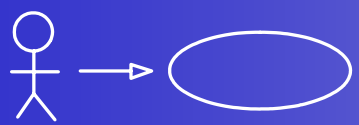
用例观点——把业务看成对外提供价值的价值流



火龙果•整理
uml.org.cn

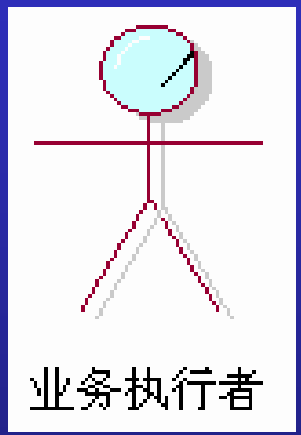
业务建模工作步骤

- 识别业务执行者（*）
- 识别业务用例（*）
- 详述业务用例（*）
- 建立业务对象模型

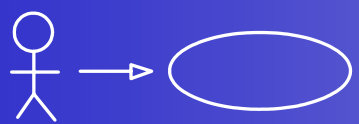


火龙果•整理
uml.org.cn

业务执行者（Business Actor）



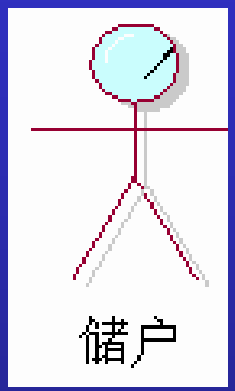
在业务之外和业务交互的人或组织



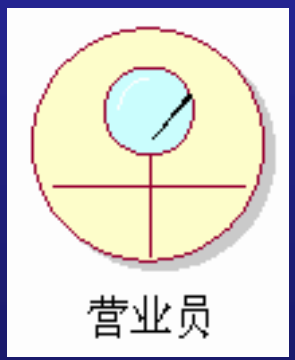
火龙果•整理
uml.org.cn

业务执行者

——注意：和业务工人不同

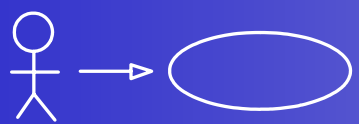


业务执行者在业务外面



业务工人在业务里面

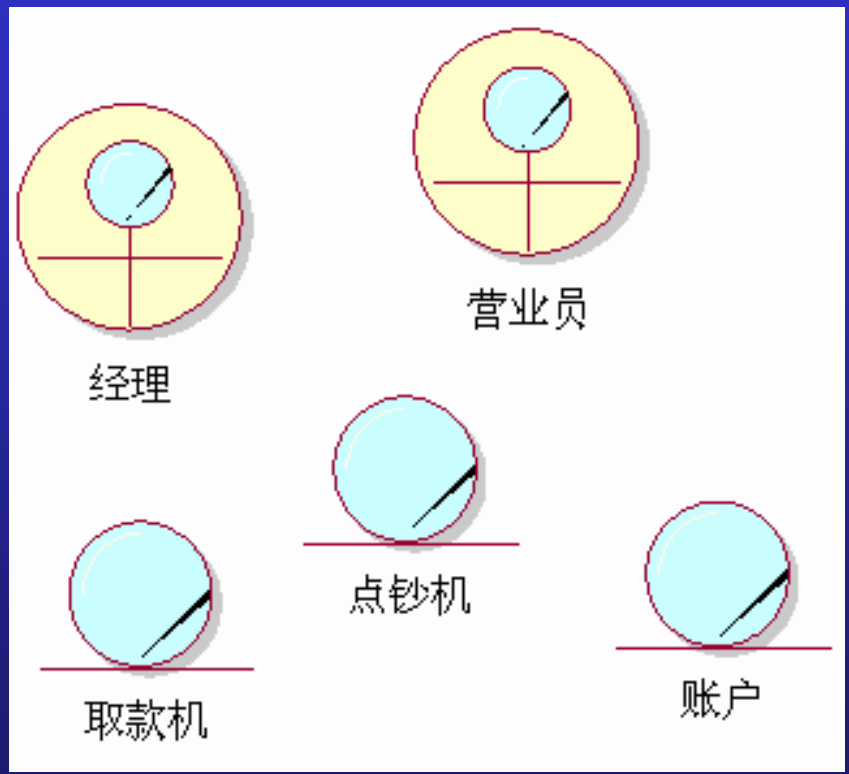
（Business Worker）



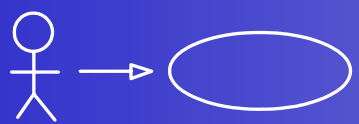
火龙果•整理
uml.org.cn

业务执行者

——业务工人、业务实体



可以相互取代责任

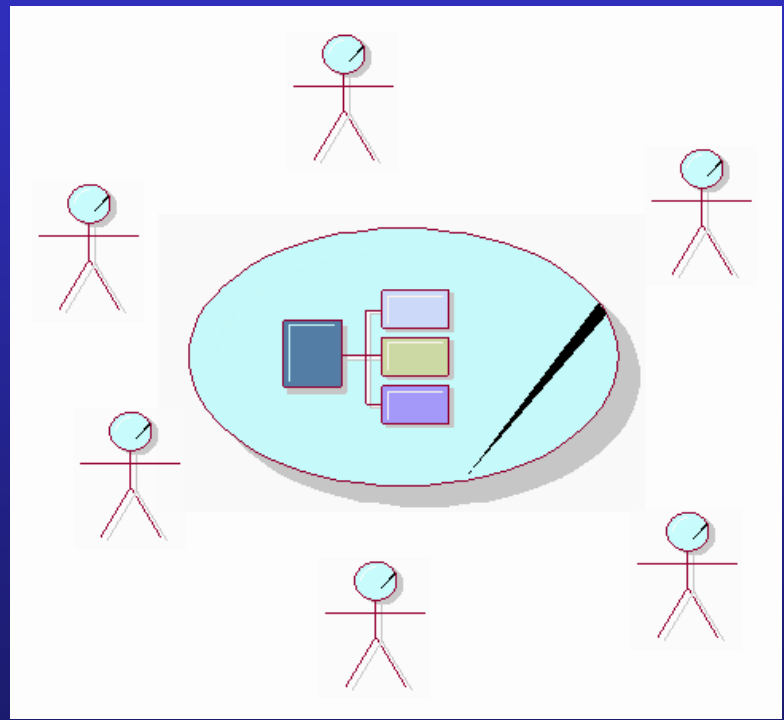


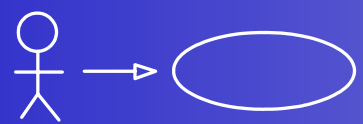
火龙果•整理
uml.org.cn

业务执行者

——查找业务执行者

- 客户
- 供应商
- 合作伙伴
- 潜在客户（市场）
- 政府
- 组织中未建模部分
- ……

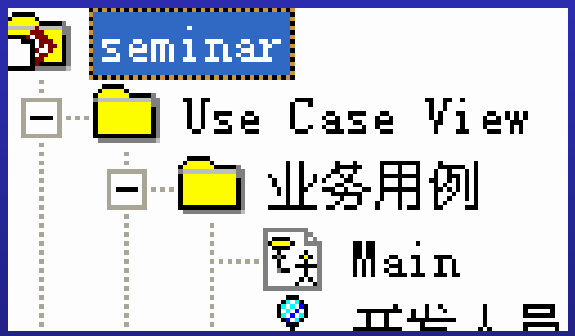


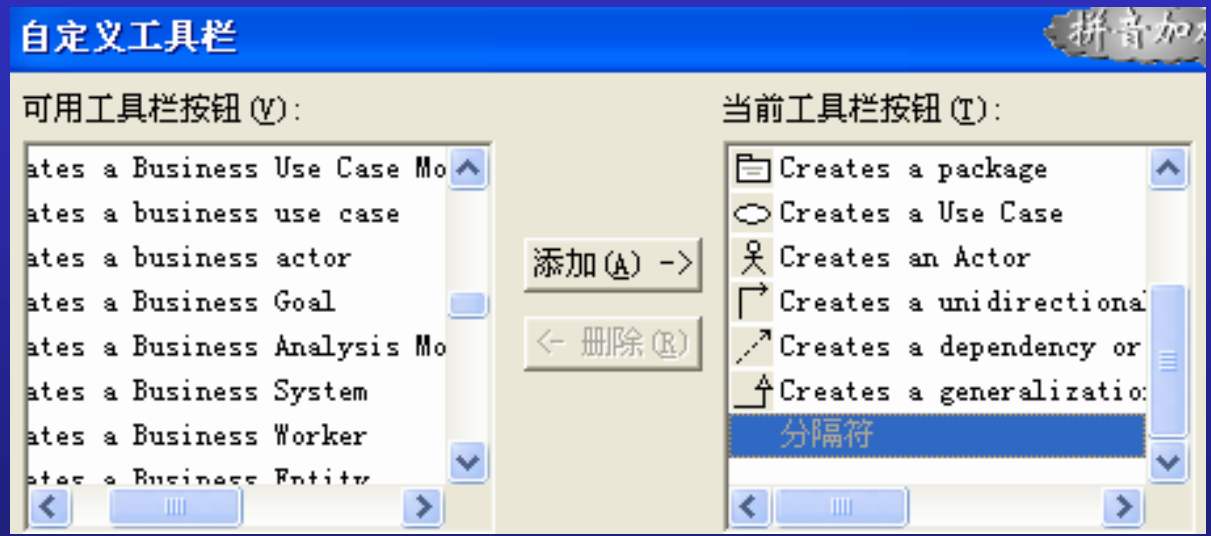


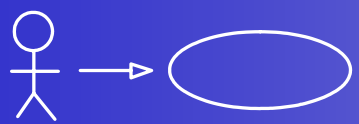
火龙果•整理
uml.org.cn

业务用例模型

——工具指南



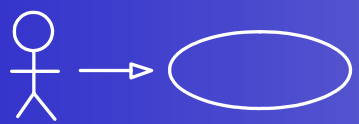




火龙果•整理
uml.org.cn

业务执行者

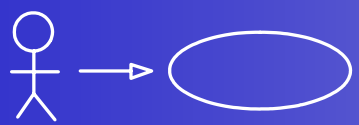
——讨论和练习、项目实作



火龙果•整理
uml.org.cn

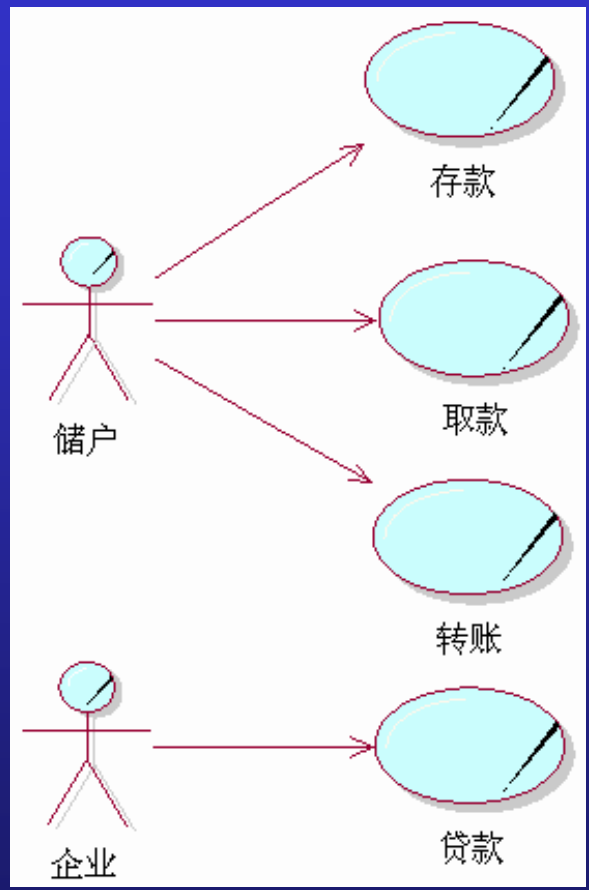
业务建模工作步骤

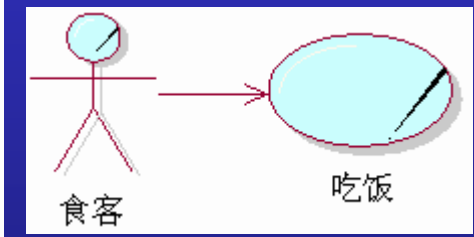
- 识别业务执行者（*）
- 识别业务用例（*）
- 详述业务用例（*）
- 建立业务对象模型



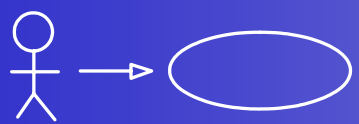
火龙果•整理
uml.org.cn

业务用例





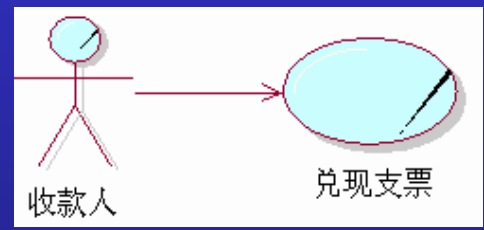
业务为业务执行者提供哪些价值？



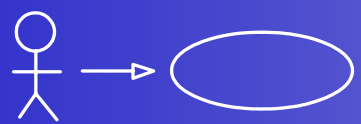
火龙果•整理
uml.org.cn

业务用例

1. 收款人在支票背后签名，写上身份证件号码，把支票和身份证件交给营业员
2. 营业员核对印章正确及证件有效
3. 营业员操作营业受理系统，办理支票兑现手续
4. 营业员把现金和证件交给收款人



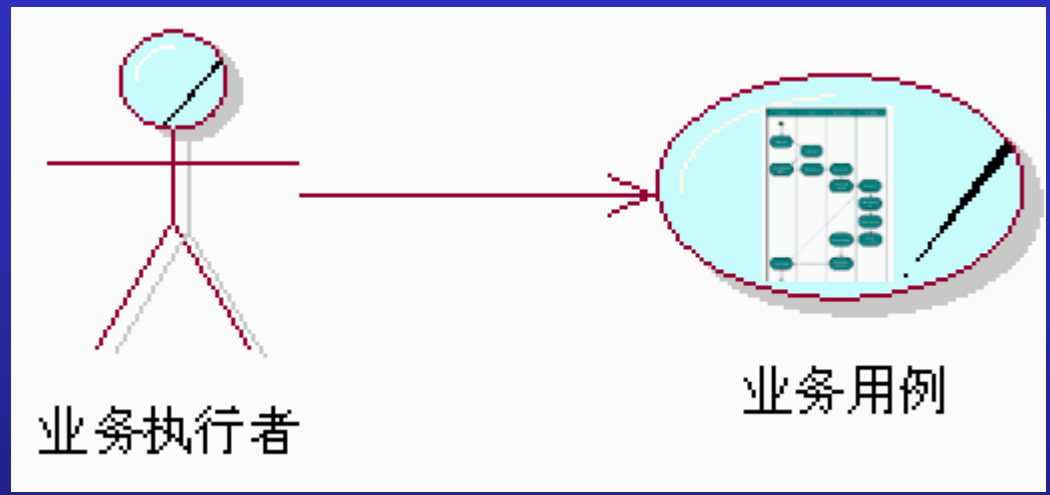
- 业务流程就是业务用例
- 业务里发生的一切都是为业务执行者提供价值



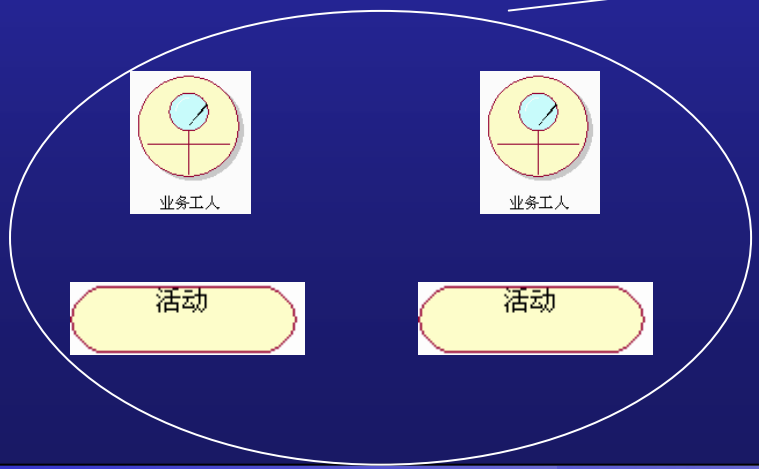
火龙果•整理
uml.org.cn

业务用例

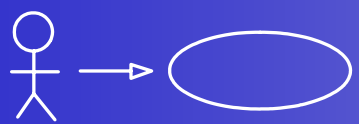
直接得出



拼装



两条启发路线



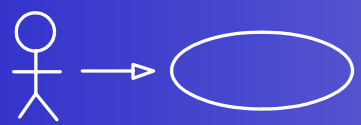
火龙果•整理
uml.org.cn

业务用例

——识别业务用例

- 支持性事件：
  - 人员的发展与维护
  - 业务内部 IT 的开发与维护
  - 办公室的设立与维护
  - 安全性
  - 法律活动

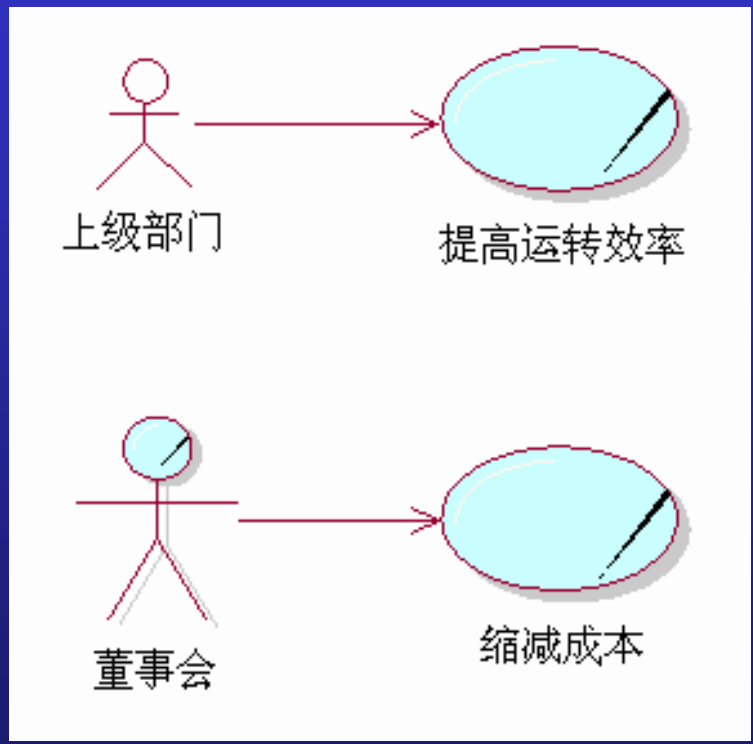
不要遗漏那些支持性的业务事件



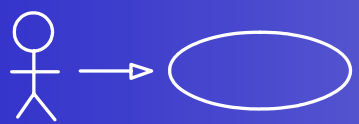
火龙果•整理
uml.org.cn

业务用例

——识别业务用例



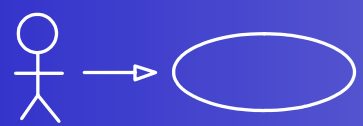
支撑性业务流程背后的业务用例



火龙果•整理
uml.org.cn

业务用例

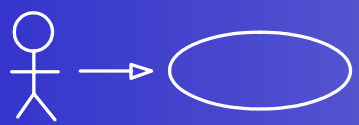
——讨论和练习、项目实作



火龙果•整理
uml.org.cn

业务建模工作步骤

- 识别业务执行者（*）
- 识别业务用例（*）
- 详述业务用例（*）
- 建立业务对象模型



火龙果•整理
uml.org.cn

详述业务用例

——可选技术

UC1：注册

主执行者

潜在会员

前置条件

潜在会员访问系统

后置条件

系统记录注册信息

涉众利益

潜在会员——希望注册尽量简单

商店——希望获得尽可能多的本来客户信息，特别是联系方法

基本路径

1. 潜在会员请求注册。
2. 系统显示注册界面。
3. 潜在会员提交注册信息。
4. 系统验证注册信息充分。
5. 系统生成用户名和密码，保存注册信息。
6. 系统显示“注册成功，等待开放账户”。

扩展

4a. 潜在会员提供的信息不充分：

4a1. 系统提示输入剩余信息

4a2. 返回 3

字段列表

3. 注册信息包括：公司名、联系人、电话、传真、Email，以及若干个联系地址。联系地址包含以下信息：州、城市、街道、邮编。

业务规则

4. 公司名、联系人、电话是必需的。

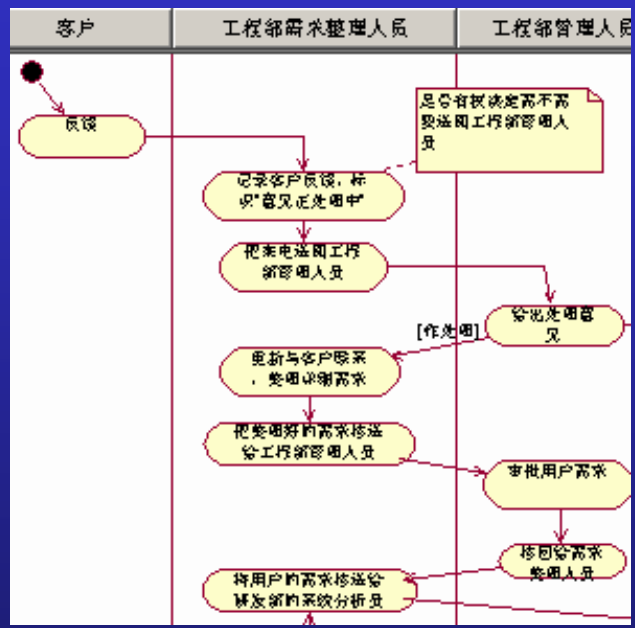
非功能需求

设计约束

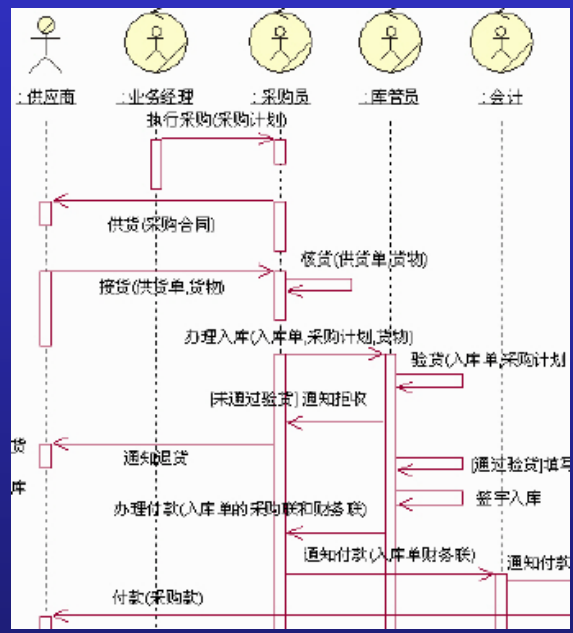
待解决问题

5. 用户名和密码生成规则待定

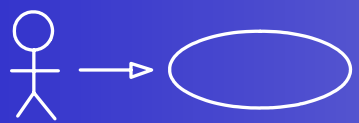
文字



活动图



顺序图

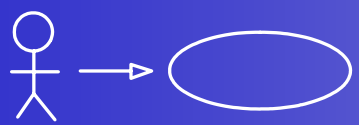


火龙果•整理
uml.org.cn

详述业务用例

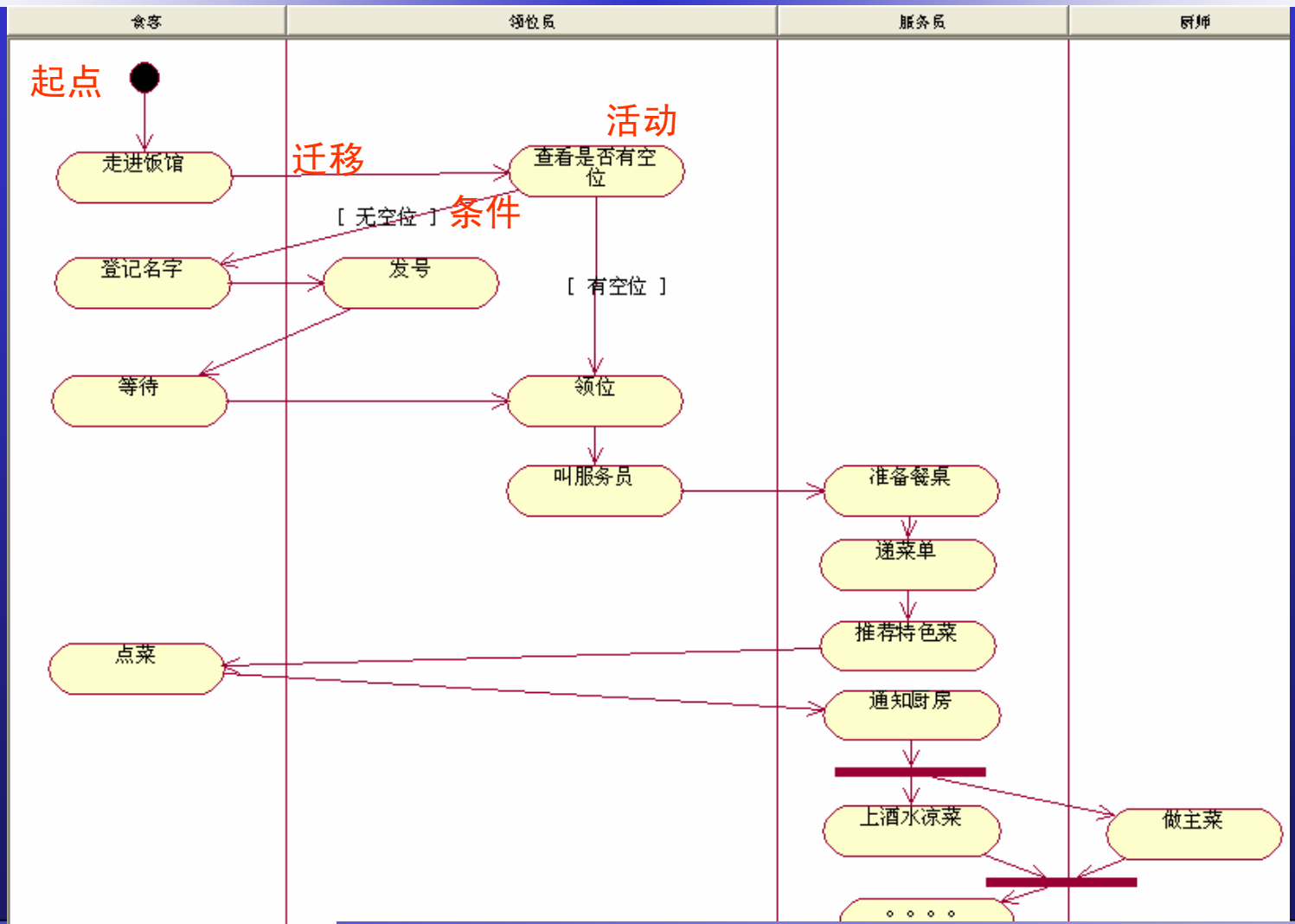
——可选技术讨论

- 只有文字——不生动，不便和客户交流
- 只有活动图——难以表达所有细节
- 用例文档中插入活动图
- 活动图中插入文字（+注释+标注基本路径）
- 顺序图，讲完后面部分再讲。



火龙果•整理
uml.org.cn

活动图解说

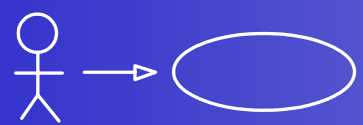


火龙果•整理
uml.org.cn

活动图

——起点终点

- 活动的一种特殊形式，各自只有一个
- 起点  ：画在左上角，只有离开的迁移
- 终点  ：画在右下角，只有进入的迁移
- 对每一项活动，都存在从起点出发，经过它到终点的路径。

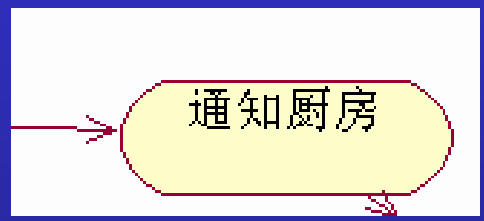


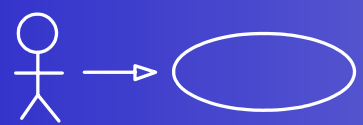
火龙果•整理
uml.org.cn

活动图

——活动

- 有进有出
- 命名：动宾结构
- 可以简单，可以复杂（见后）



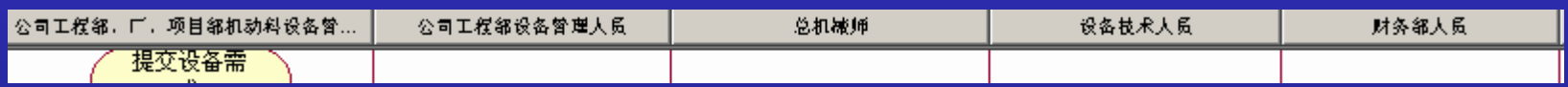


火龙果•整理
uml.org.cn

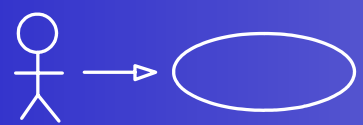
活动图

——泳道

- 活动的负责者



- 泳道可以多维

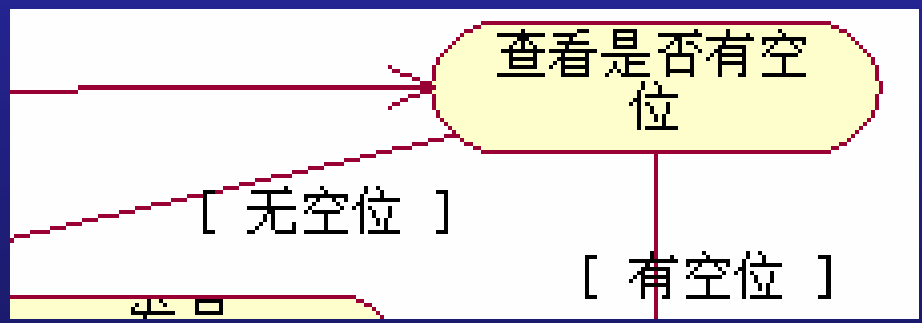


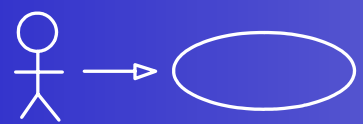
火龙果•整理
uml.org.cn

活动图

——迁移和迁移条件

- 向外迁移的条件之和必须是完备集（“其他”）
- 向外迁移的条件之间不能重叠



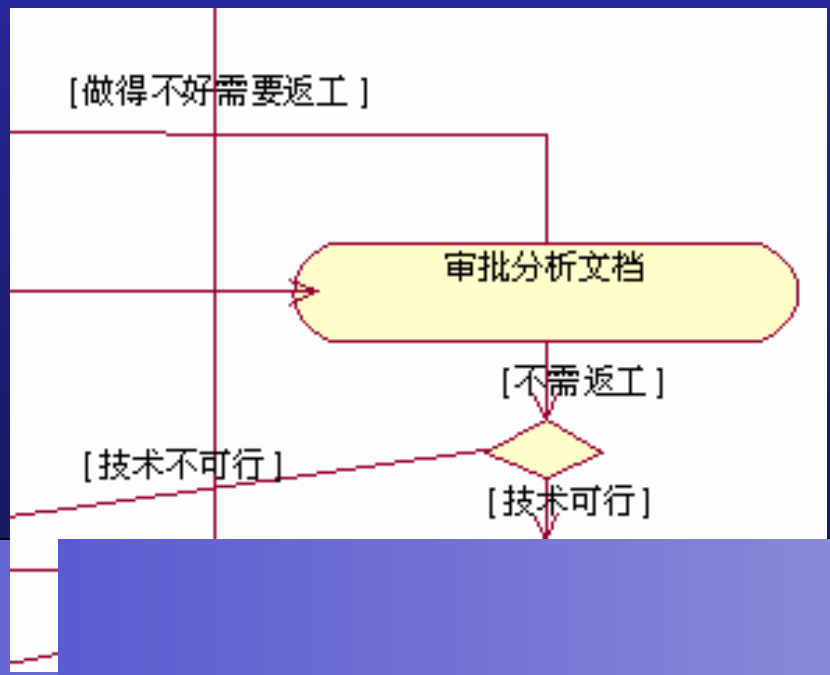


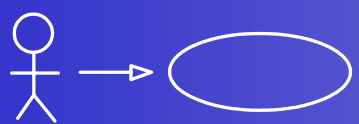
火龙果•整理
uml.org.cn

活动图

——判定

- 和流程图里的有区别（空的，判定内容在前面活动中或者由泳道直接选择）
- 第一个判断不用加判定
- 谨慎使用（误把活动当判定）



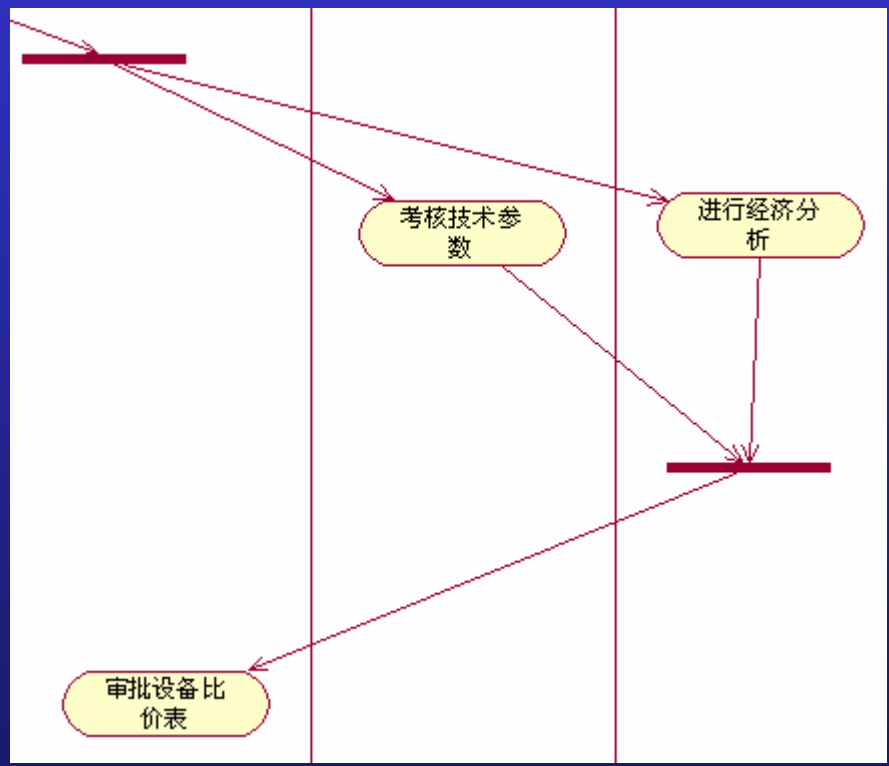


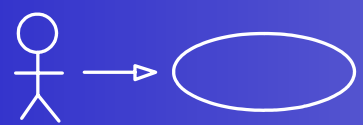
火龙果•整理
uml.org.cn

活动图

——并行（分叉与合并）

- 有分必有合
- 有分必有进
- 有合必有出
- 并行！＝同时

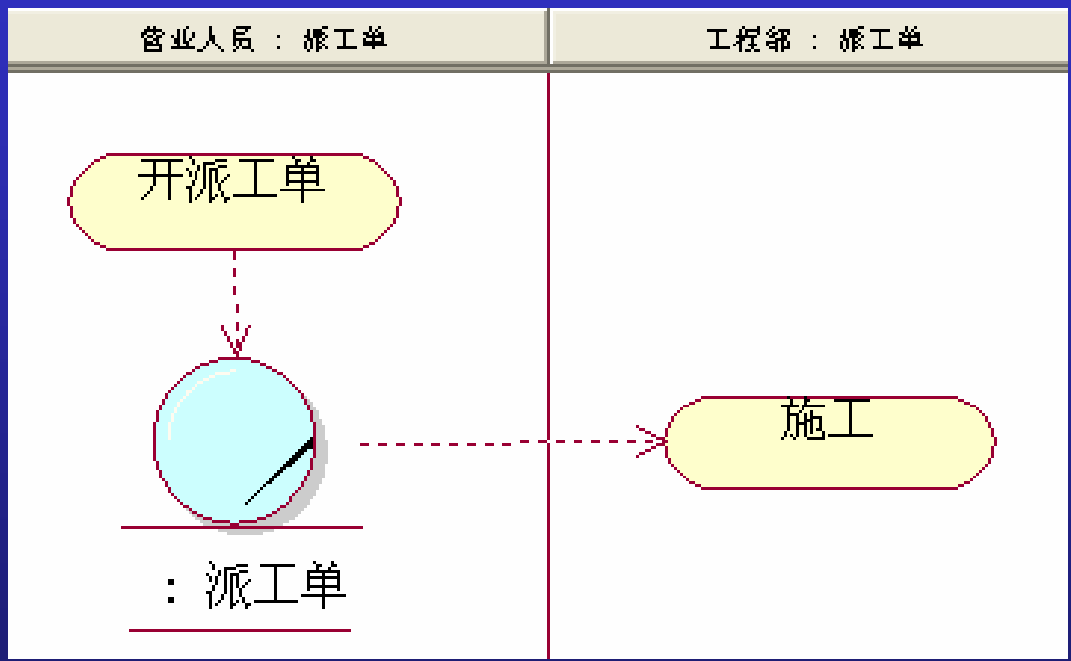




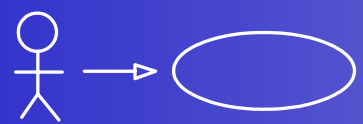
火龙果•整理
uml.org.cn

活动图

——对象流



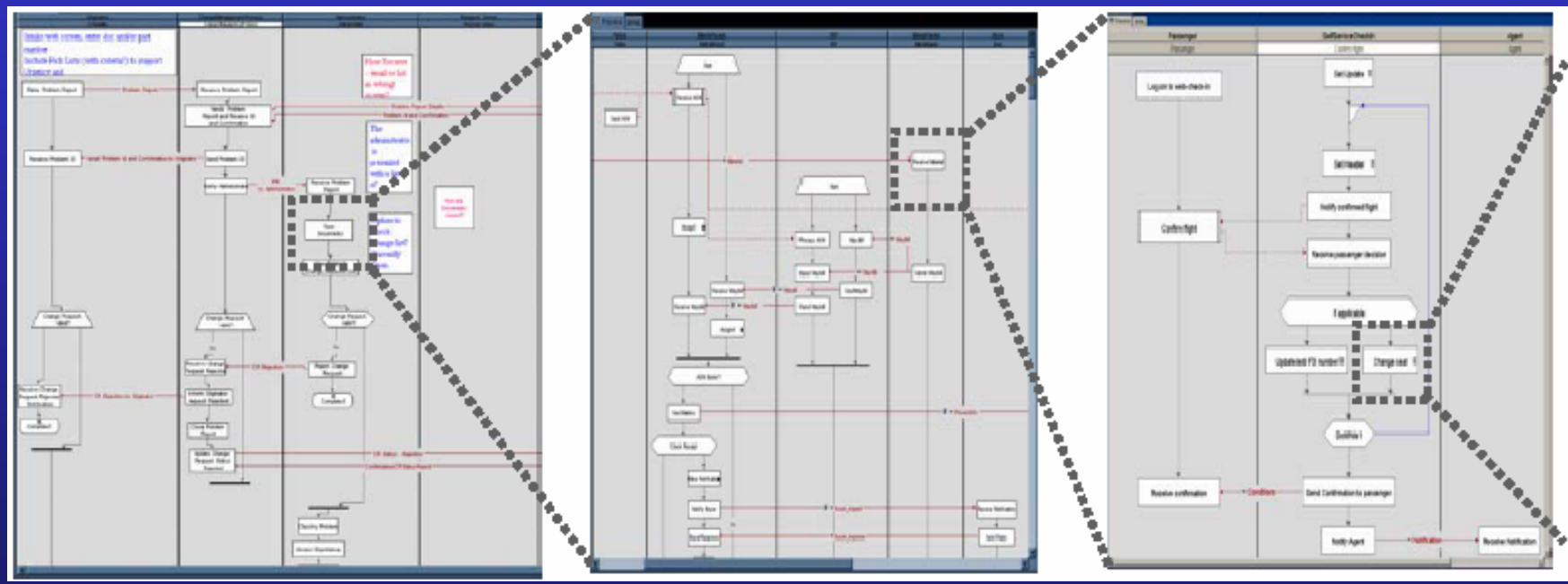
特地指出对某些业务实体的操作，类似数据流图



火龙果•整理
uml.org.cn

活动图

——分层（1）

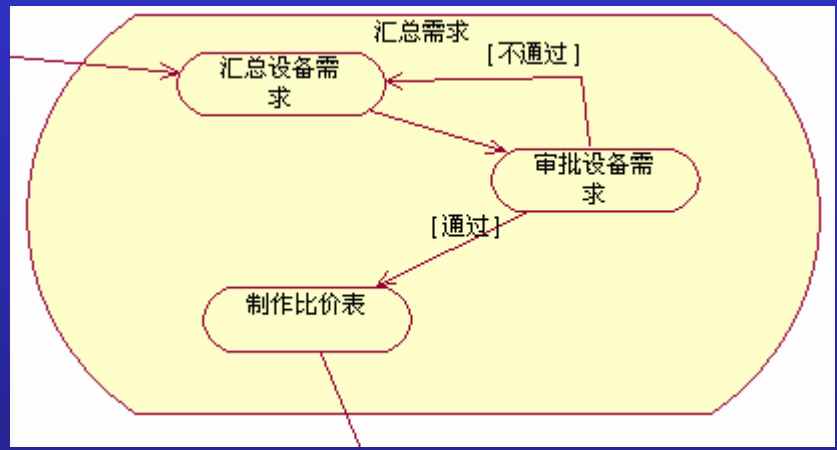


火龙果•整理
uml.org.cn

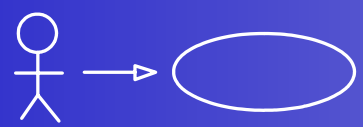
活动图

——分层（2）

- Rose



- PD

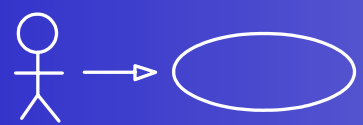


火龙果•整理
uml.org.cn

活动图

——分层（3）

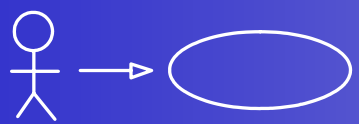
- 出入平衡
- 顶层有起点终点，下层可以没有
- 展开和隐藏



火龙果•整理
uml.org.cn

详述业务用例

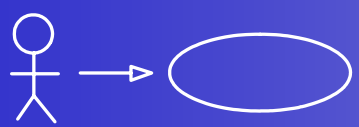
——讨论和练习、项目实作



火龙果•整理
uml.org.cn

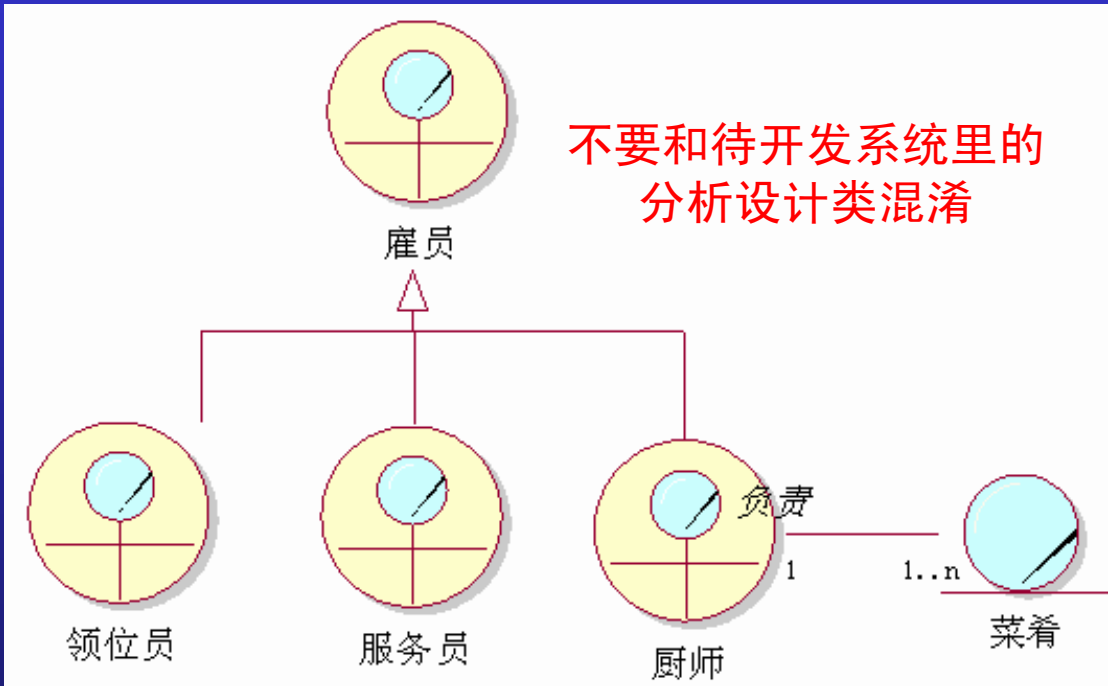
业务建模工作步骤

- 识别业务执行者（*）
- 识别业务用例（*）
- 详述业务用例（*）
- 建立业务对象模型

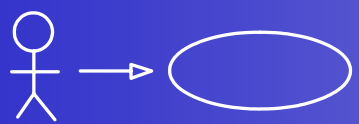


火龙果•整理
uml.org.cn

业务对象模型

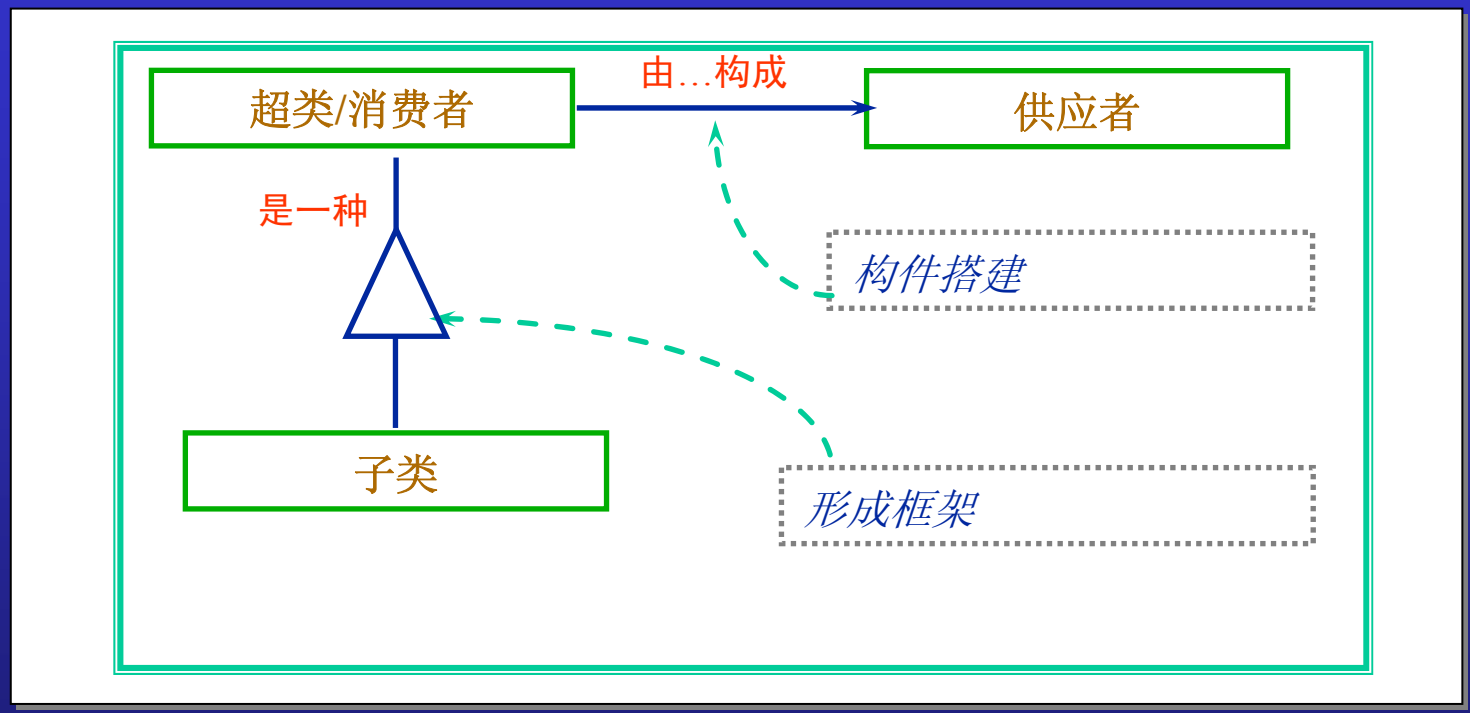


用类图勾勒出现实中的人、事物、关系

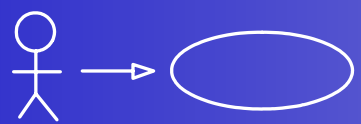


火龙果•整理
uml.org.cn

类的关系



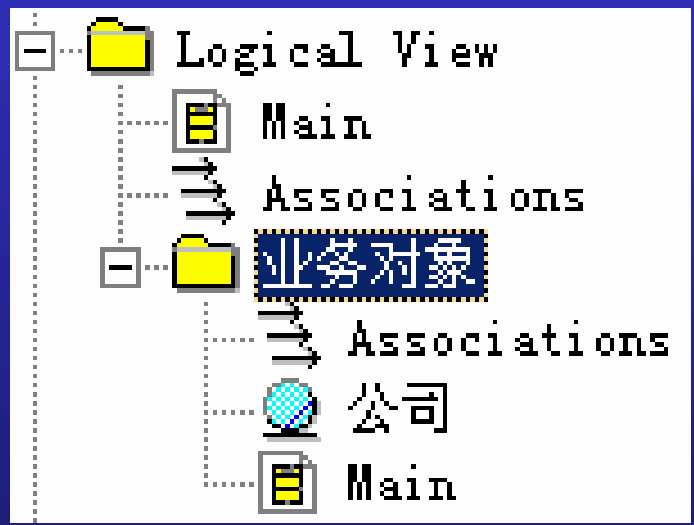
- 泛化：子类通过继承拥有超类的特征（集合关系）
- 关联：对象通过组装拥有其他对象的特征（个体关系）

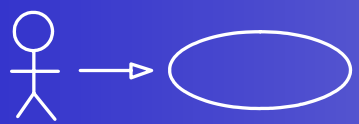


火龙果•整理
uml.org.cn

业务对象模型

——工具指南

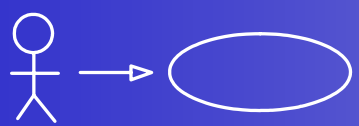




火龙果•整理
uml.org.cn

业务对象模型

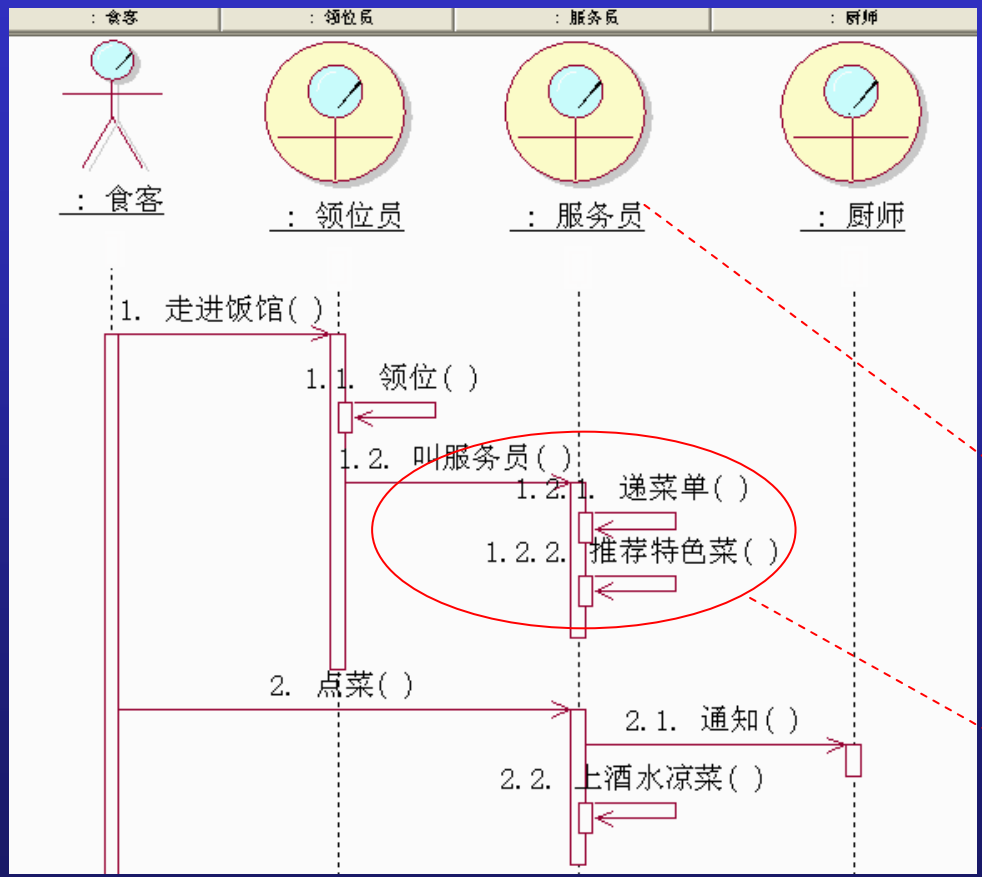
——讨论和练习、项目实作



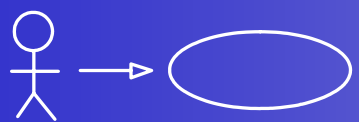
火龙果•整理
uml.org.cn

业务对象模型

——通过顺序图详述业务用例

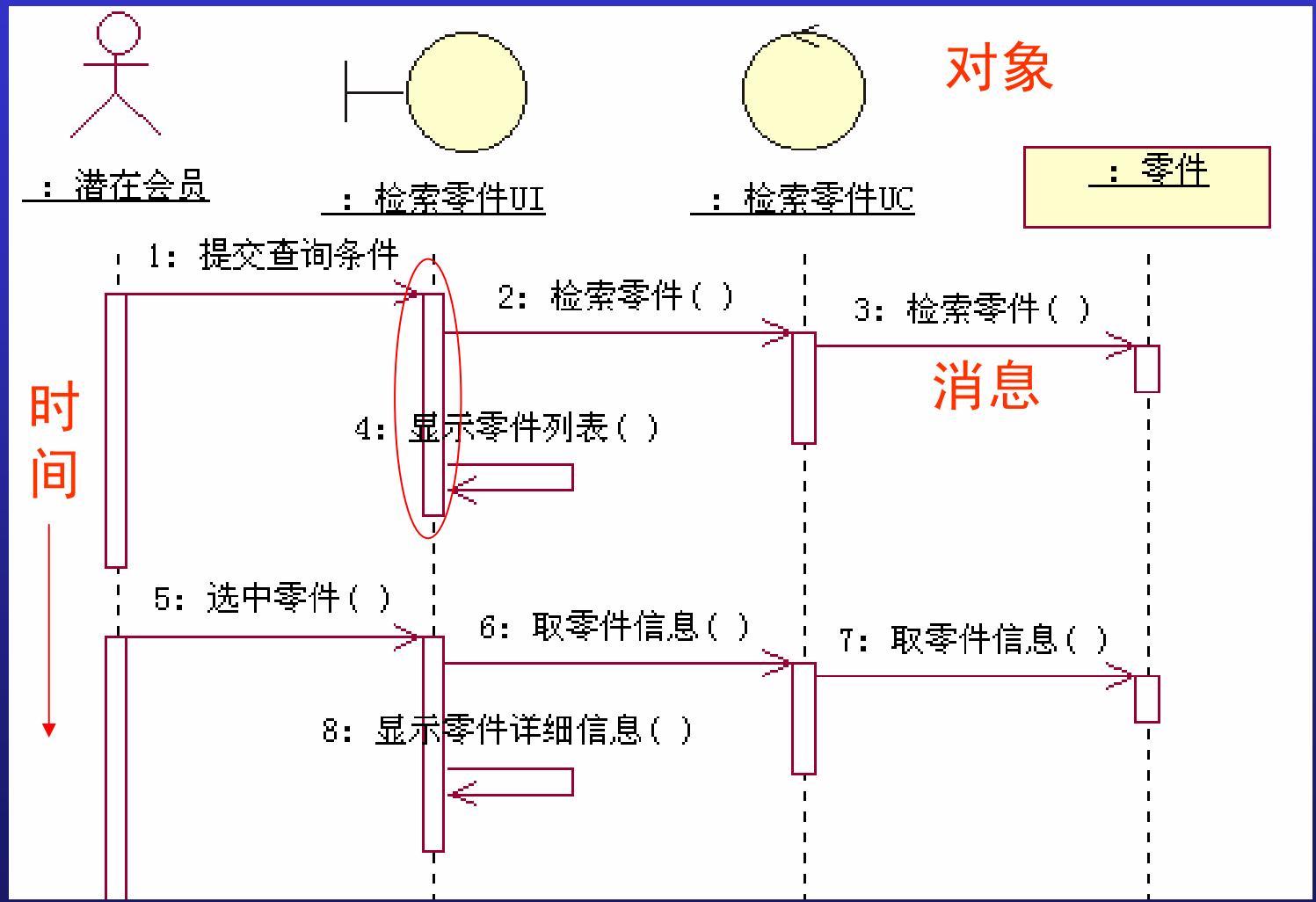


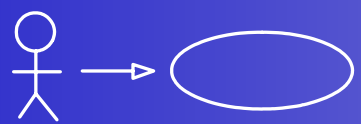
相对活动图，
能看到一个能力单元的总体责任



火龙果•整理
uml.org.cn

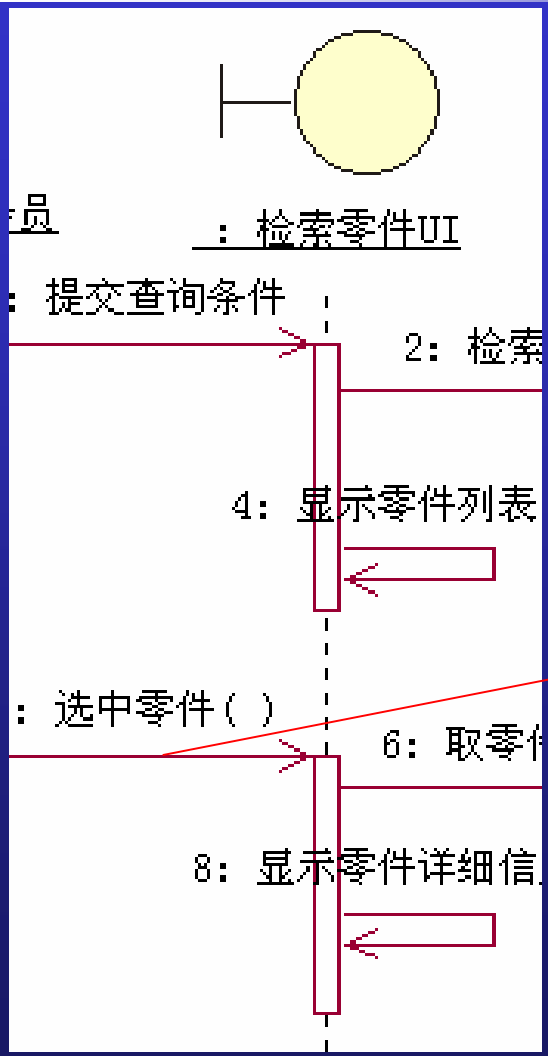
顺序图解说

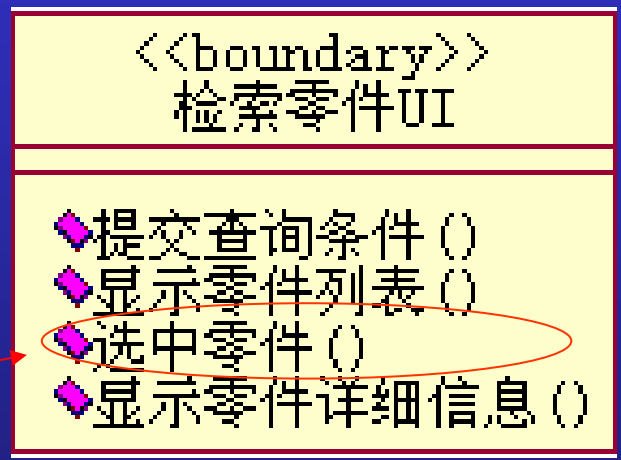




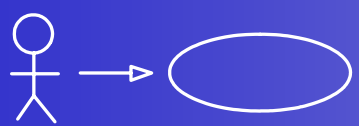
火龙果•整理
uml.org.cn

顺序图和类图的映射



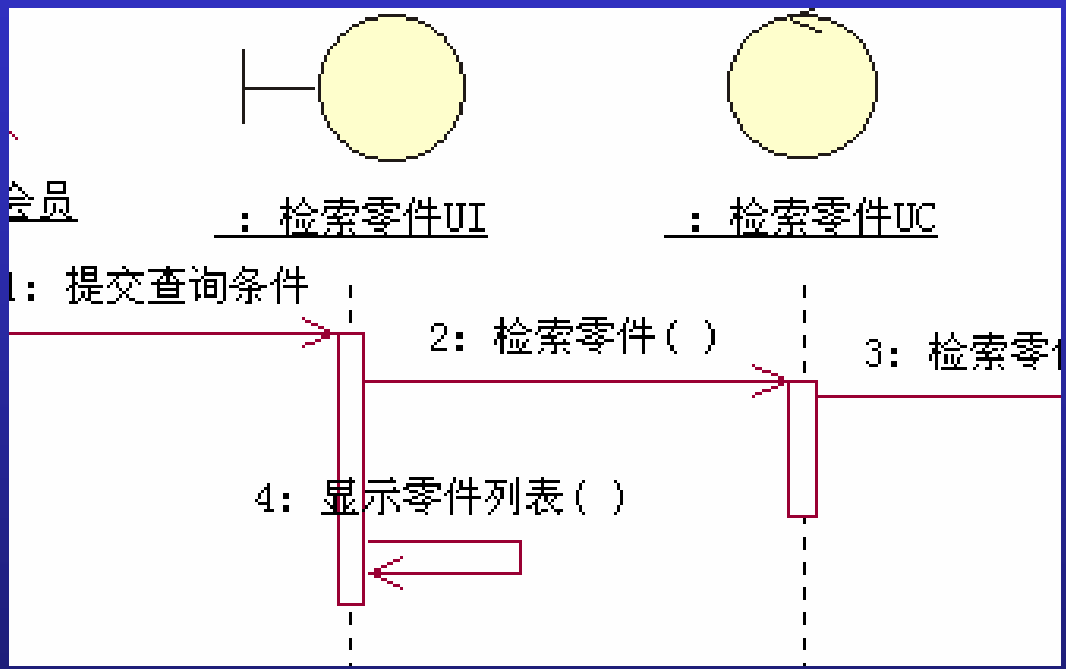


消息的传入：类对象所具有的操作——责任



火龙果•整理
uml.org.cn

顺序图和类图的映射



检索零件UI.提交查询条件()

{

…

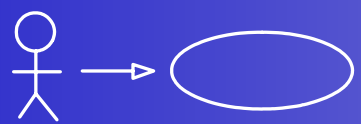
检索零件UC.检索零件()

…

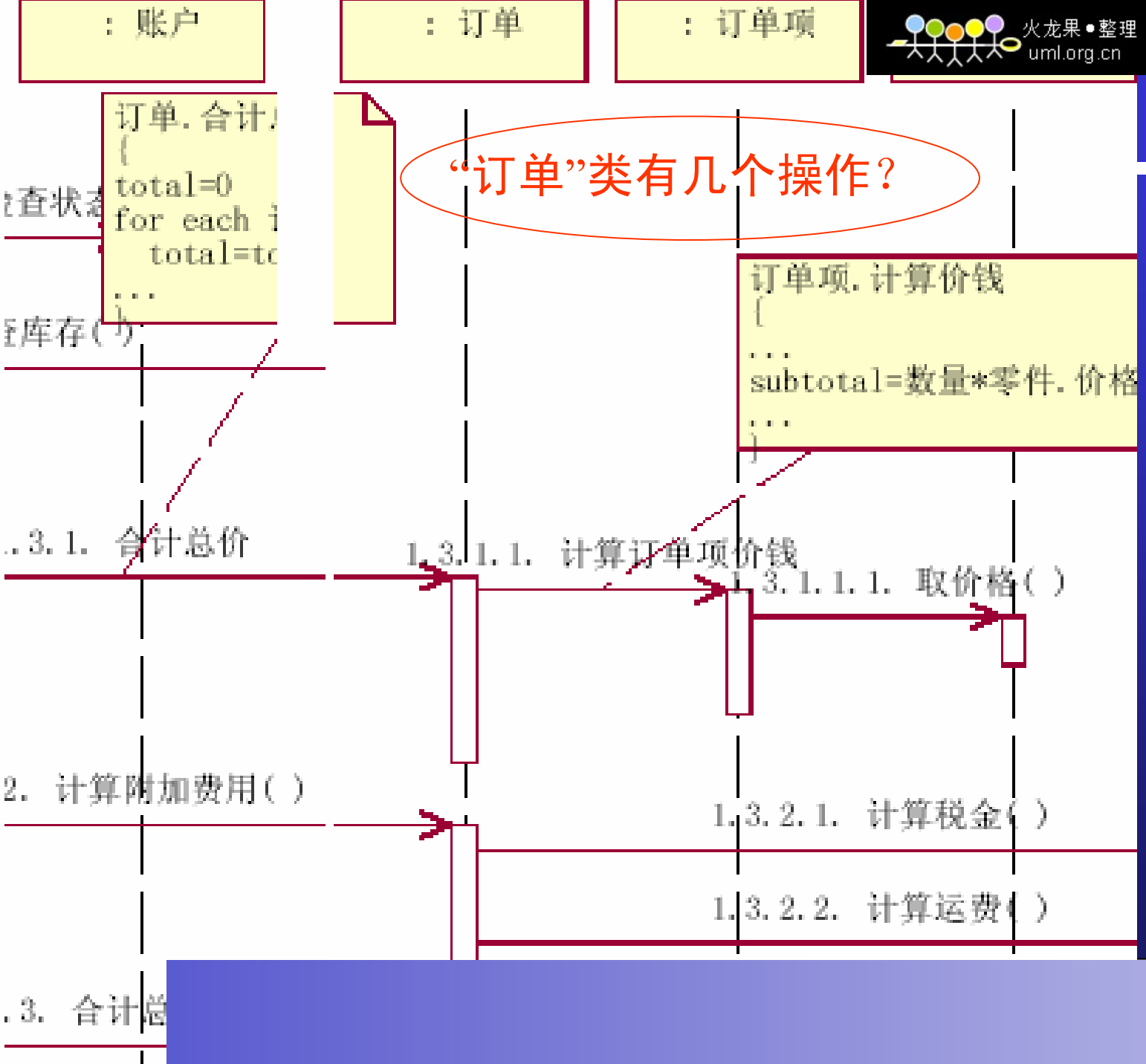
(检索零件UI.)显示零件列表()

}

消息的传出：类对象完成操作所需合作——协作



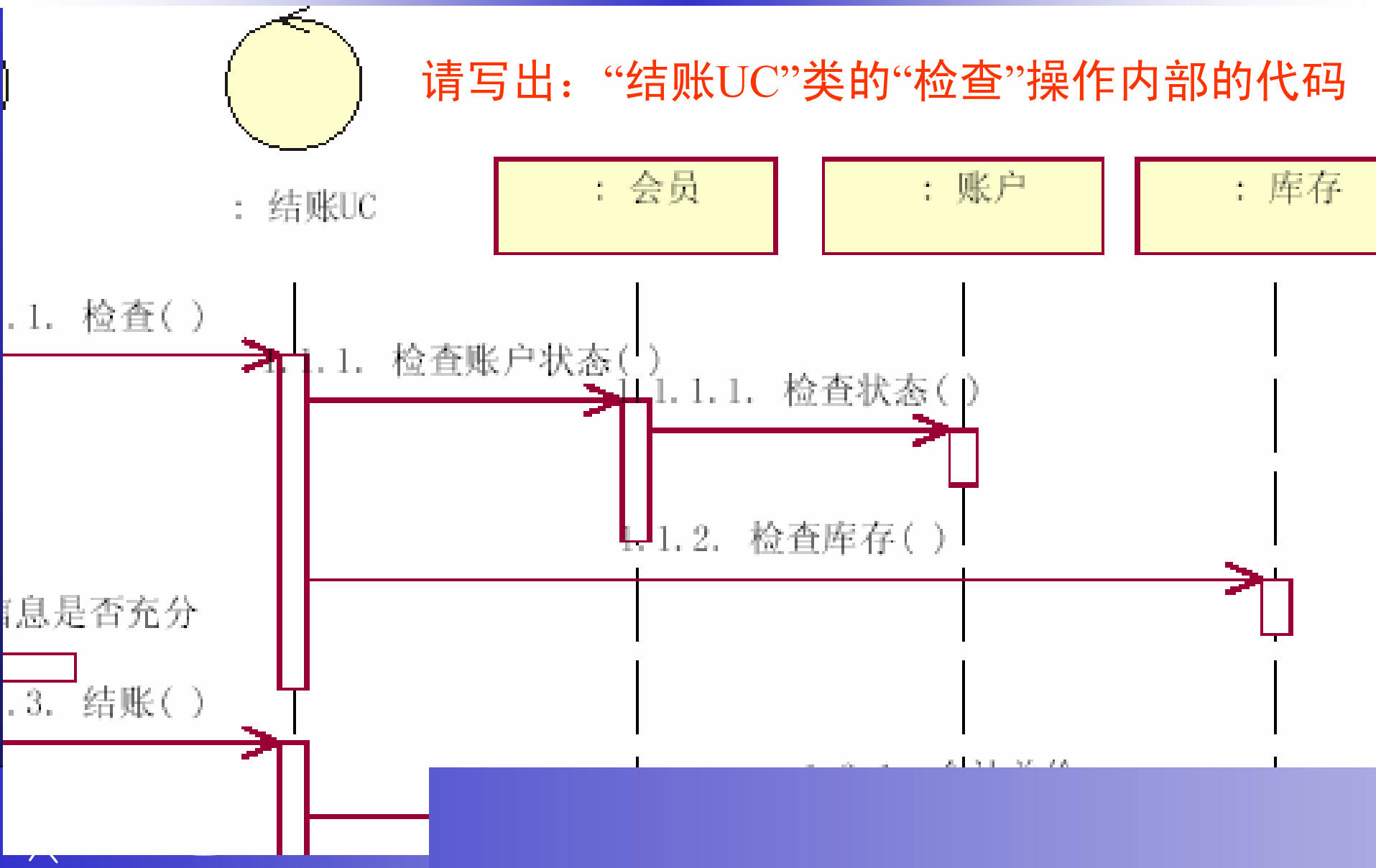
练习


：账户
：订单
：订单项
火龙果•整理
uml.org.cn
订单.合计
{
total=0
for each
total=tc
...
}
“订单”类有几个操作？
订单项.计算价钱
{
...
subtotal=数量*零件.价格
...
}
.3.1. 合计总价
1.3.1.1. 计算订单项价钱
1.3.1.1.1. 取价格( )
2. 计算附加费用( )
1.3.2.1. 计算税金( )
1.3.2.2. 计算运费( )
.3. 合计总

火龙果•整理
uml.org.cn

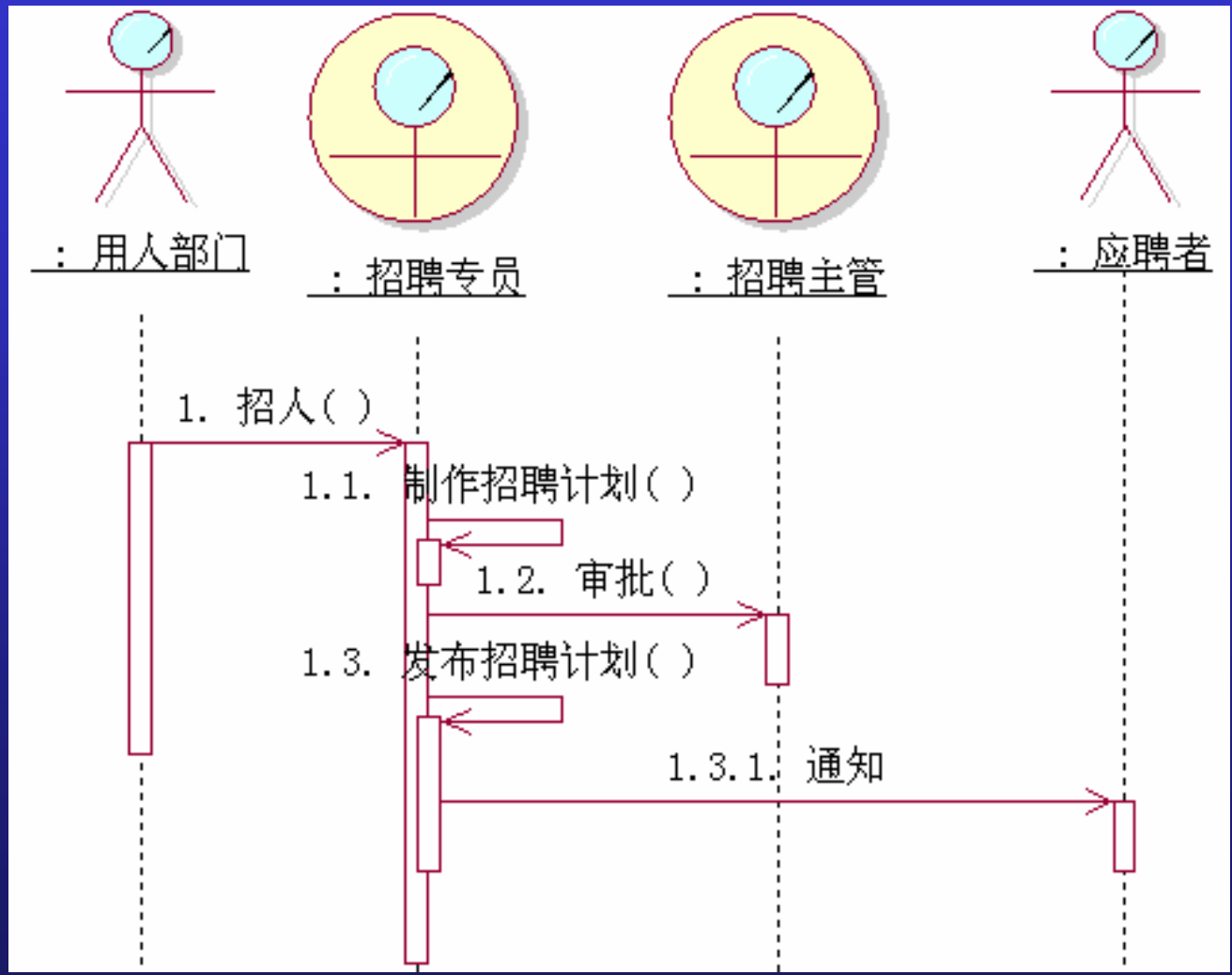
练习

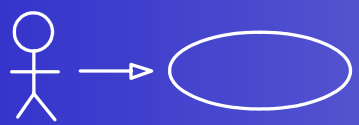
请写出：“结账UC”类的“检查”操作内部的代码



火龙果•整理
uml.org.cn

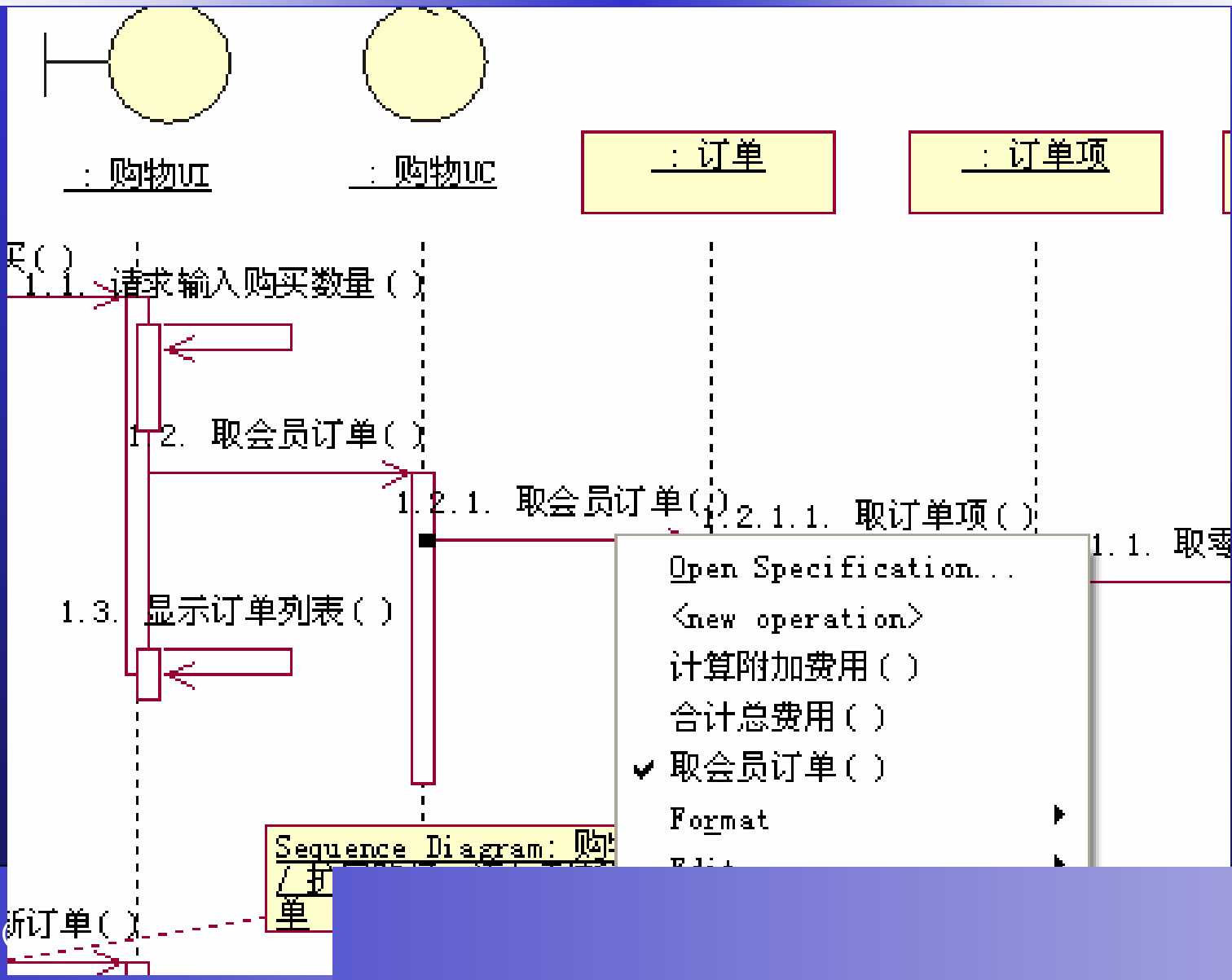
业务顺序图也一样





火龙果•整理
uml.org.cn

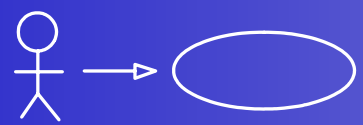
建模工具允许自动映射类图



火龙果•整理
uml.org.cn

责任分配原则

- 原则1. 专家（ Expert ）原则
- 原则2. 老板（Boss）原则
- 原则3：可视（Visibility）原则



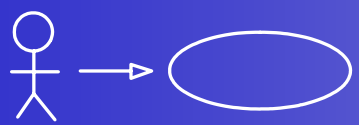
火龙果•整理
uml.org.cn

责任分配原则（1）

——专家（ Expert ）原则

- 把责任分配给专家

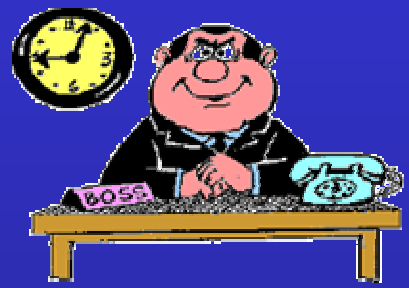
资源决定责任——各尽其才，各施其能



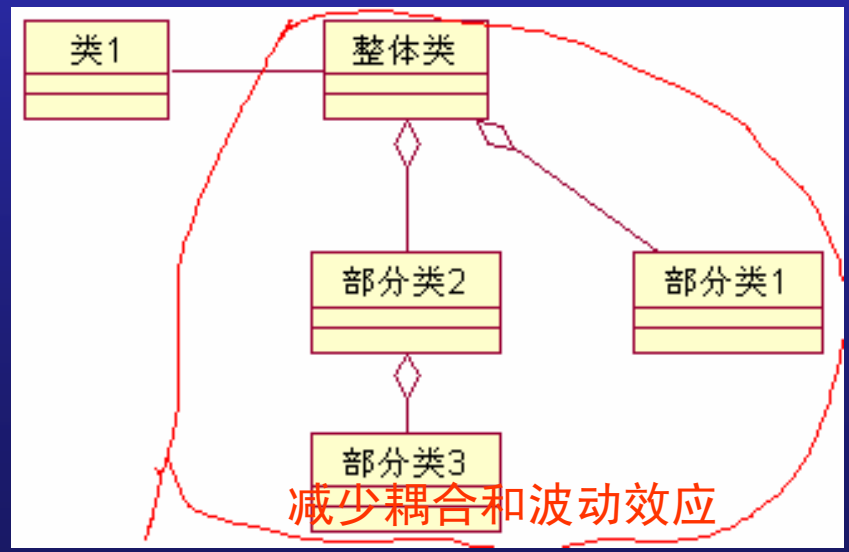
火龙果•整理
uml.org.cn

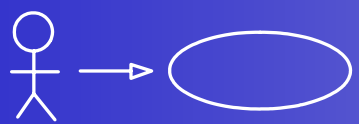
责任分配原则（2）

——老板（Boss）原则



- 聚合/组合结构的消息传递
- 当出现以下情况时，发给A的消息先通过B处理和中转
  - B聚合A（Aggregation）
  - B组合A（ Composition ）





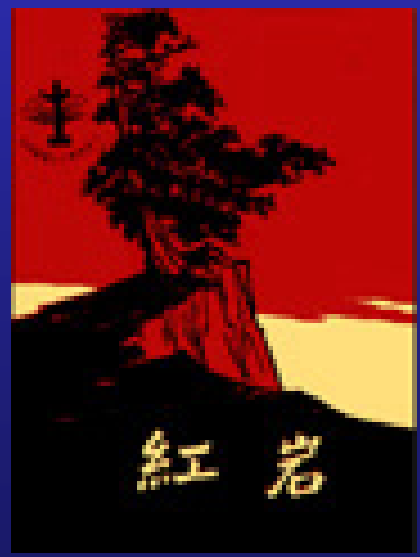
火龙果•整理
uml.org.cn

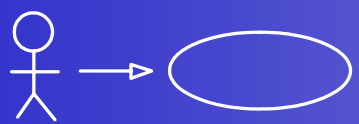
责任分配原则（3）

——可视（Visibility）原则

- 两个对象之间有消息传递，相应类应有关联

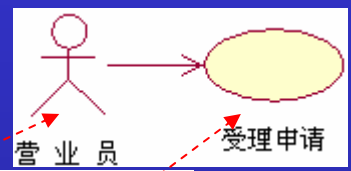
- 不要与陌生人说话



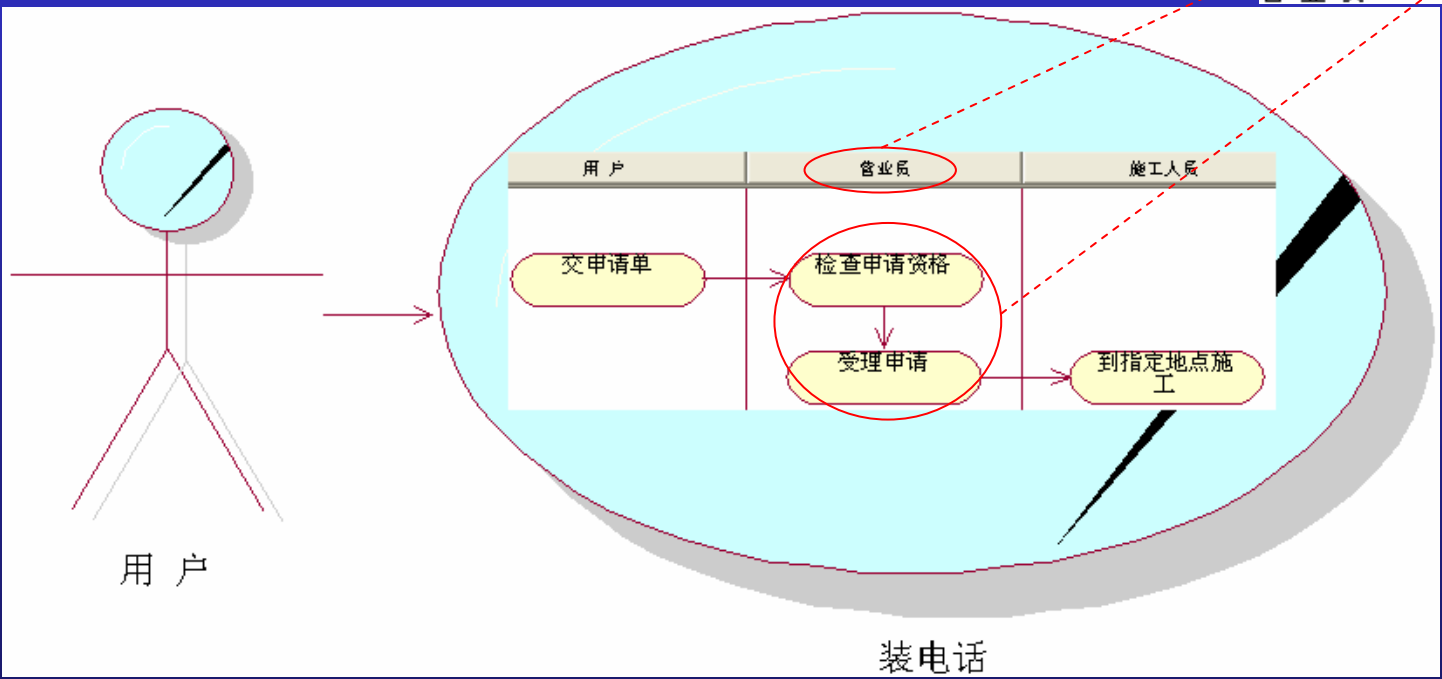


火龙果•整理
uml.org.cn

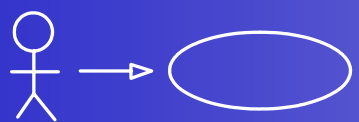
从业务用例到系统用例



研究对象：系统



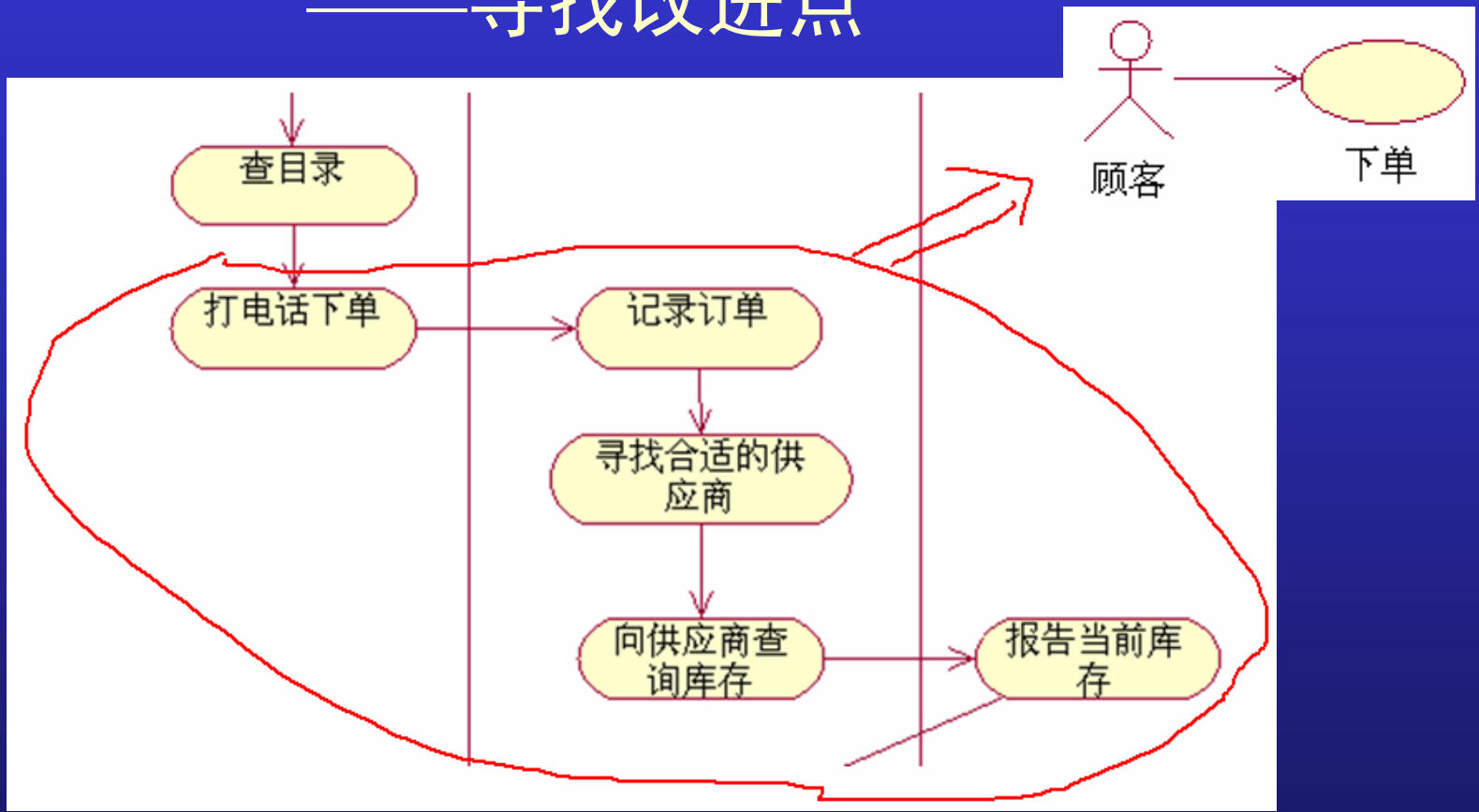
研究对象：组织



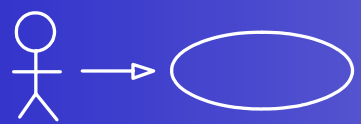
火龙果•整理
uml.org.cn

从业务用例到系统用例

——寻找改进点



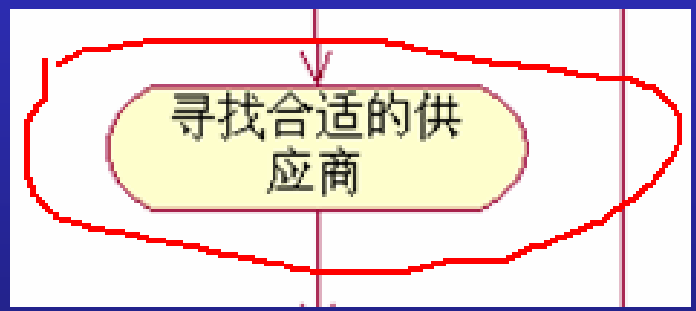
信息自动流转

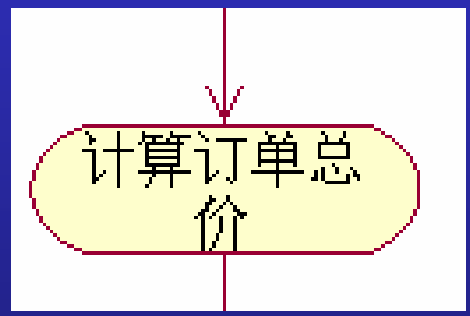


火龙果•整理
uml.org.cn

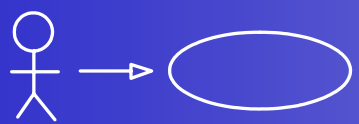
从业务用例到系统用例

——寻找改进点





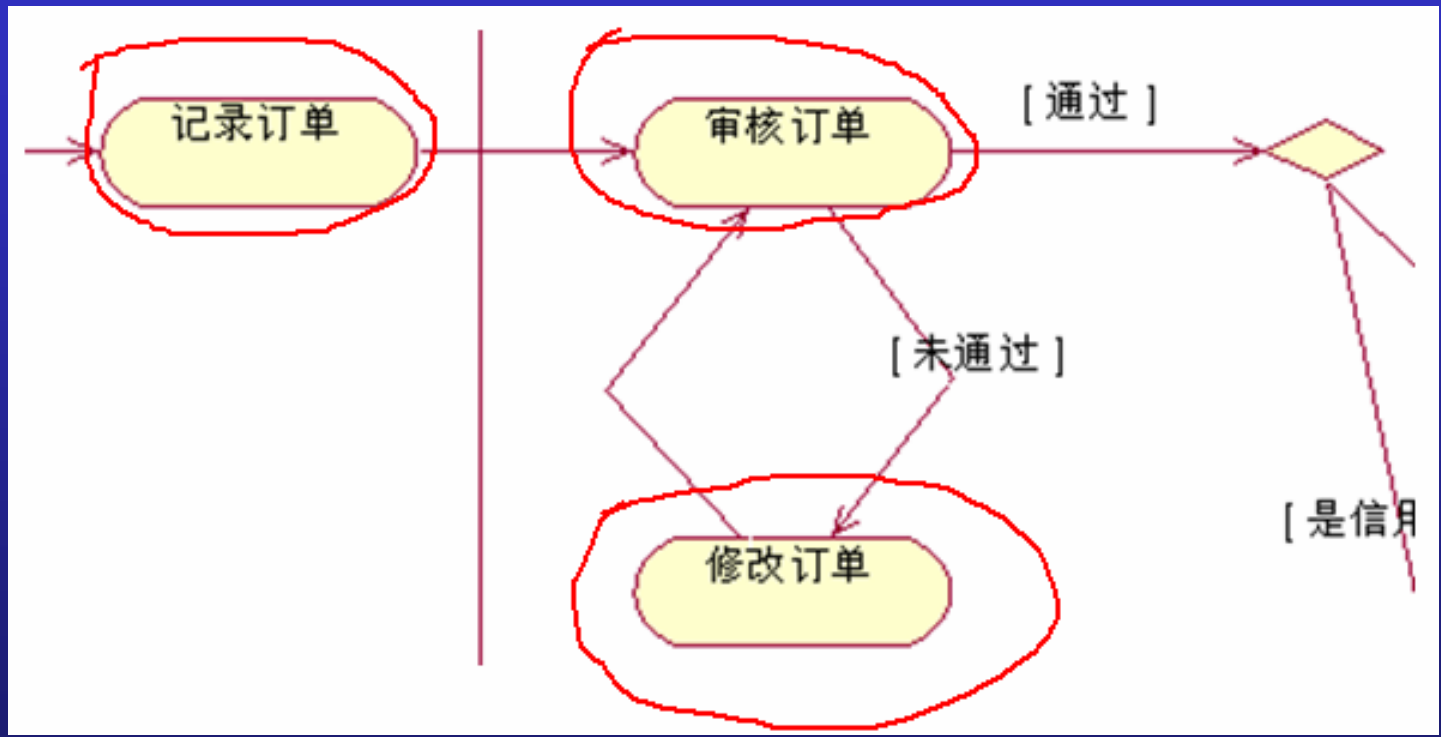
演绎复杂业务逻辑



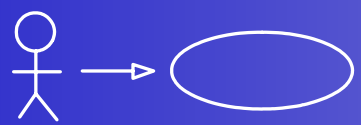
火龙果•整理
uml.org.cn

从业务用例到系统用例

——寻找改进点



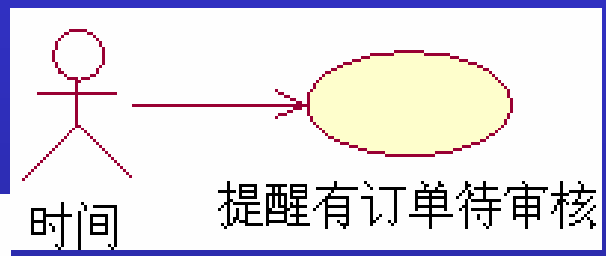
访问和操作业务实体

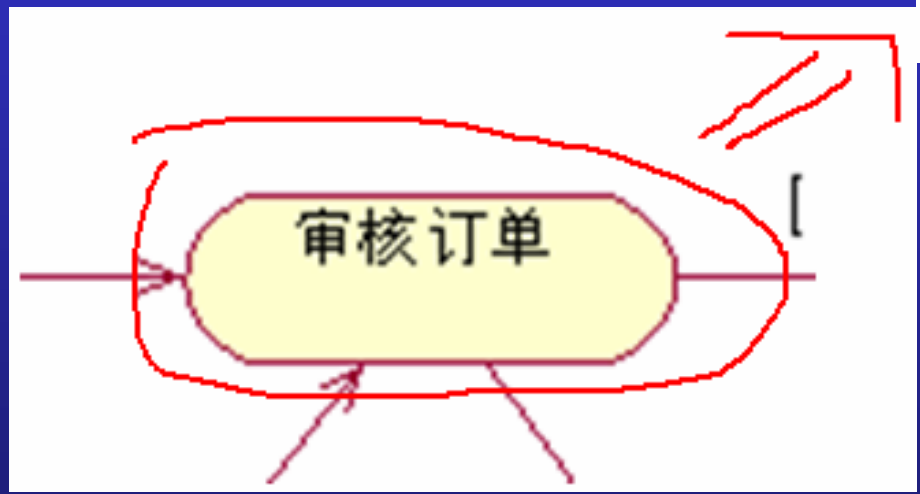


火龙果•整理
uml.org.cn

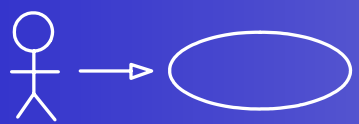
从业务用例到系统用例

——寻找改进点





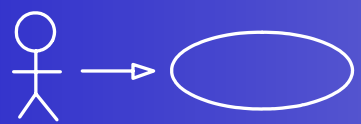
自动工作—时间执行者



火龙果•整理
uml.org.cn

从业务用例到系统用例

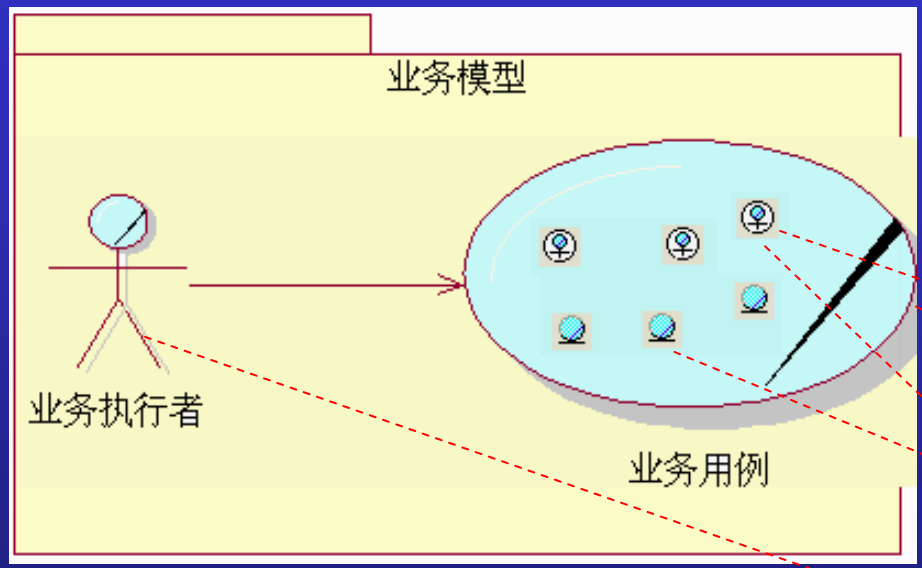
——讨论和练习、项目实作



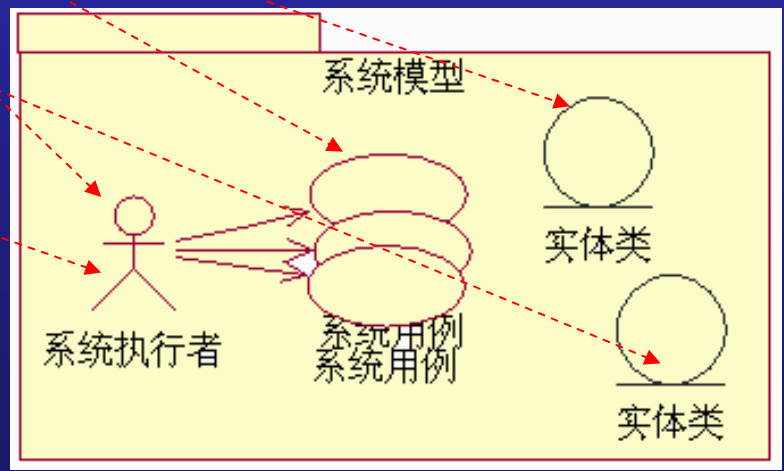
火龙果•整理
uml.org.cn

从业务模型到系统模型

——可能的对应



- 业务用例→系统（子系统）
- 业务执行者→系统执行者
- 业务工人→系统执行者
- 活动→系统用例
- 业务实体、业务工人、业务执行者→实体类



并没有标准答案
——依赖于愿景！