

《需求分析与系统设计》

Requirements Analysis and System Design

任课教师： 范 国 祥

电 话： 0451-86418876-811(O)

13199561265(Mobile)

邮 箱： fgx@hit.edu.cn

哈工大计算学部
国家示范性软件学院
软件工程教研室

2023. 09



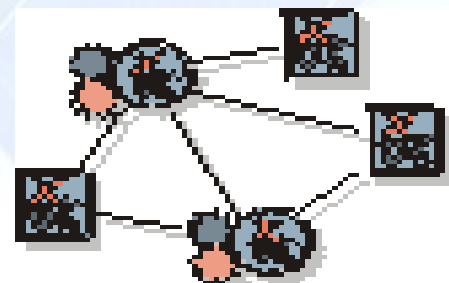
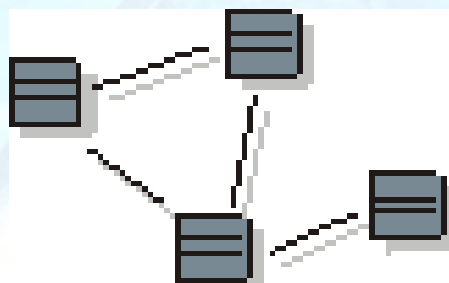
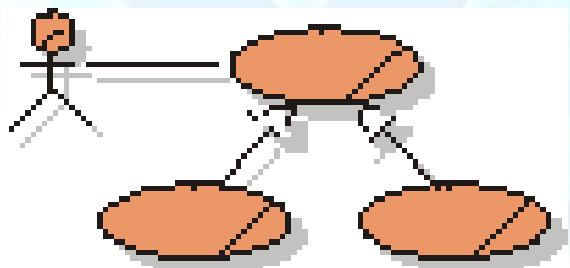
本章主要内容

- 模型和建模
- 事件和系统需求
- 事物和系统需求
- 实体-关系图（ERD）
- 类图



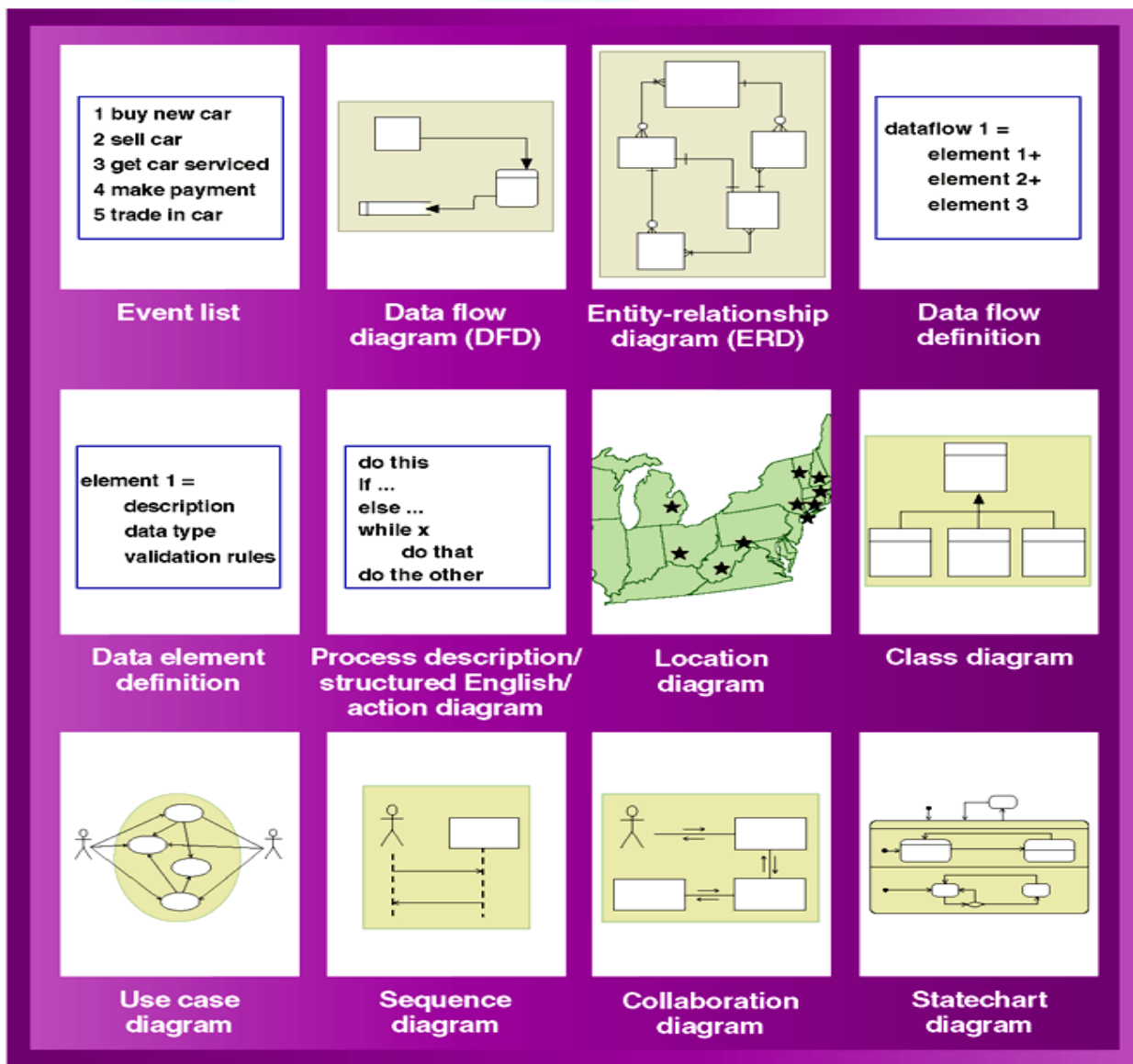
1 模型和建模

- 分析员需要一组模型来充分描述信息系统的需求
- 不同的模型从不同的角度描述需求或解决方案



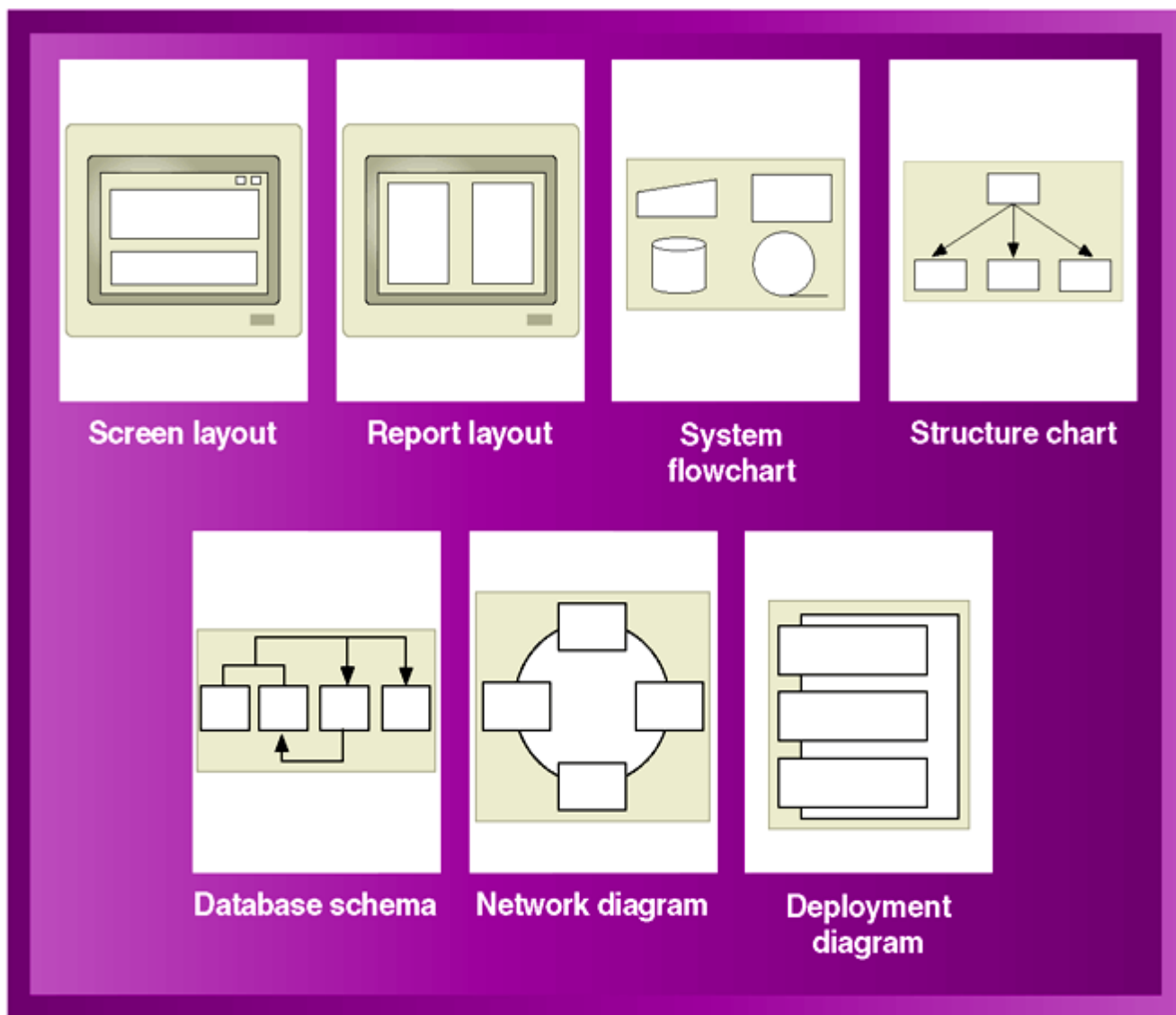


1 模型和建模





1 模型和建模





1.1 模型的作用

- 建立模型的过程能帮助分析员澄清和改良设计
- 系统往往很复杂，模型能很好表示各部分之间的关系
- 模型便于项目组成员之间交流
- 模型有助于和系统用户之间进行交流并促进理解
- 建立的模型可以作为以后的维护和升级时的文档



1.2 模型的类型

三种常见模型：数学模型、描述模型、图形模型

- **数学模型**：描述系统技术方面的一系列数学公式
- **描述模型**：描述系统某些方面的叙述性的备忘录、报表、列表、文字等
- **图形模型**：描述系统的图表或系统某些方面的示意性表示



1.3 用于分析和设计的模型

分析阶段创建的模型

- 事件列表
- 事物列表

结构化方法

- 系统流程图
- 决策树/决策表
- 实体关系图
- 数据流图
- 数据字典
- 伪代码

基于UML的OO方法

- 用例图
- 用例场景描述
- 分析类图
- 活动图
- 状态图
- 时序图/协作图

设计阶段创建的模型

- 体系结构图
- 界面布局图

- 系统结构图
- 程序流程图

-
- 设计类图
 - 时序图
 - 包图
 - 组件图
 - 网络图
 - 部署图



2 事件和系统需求

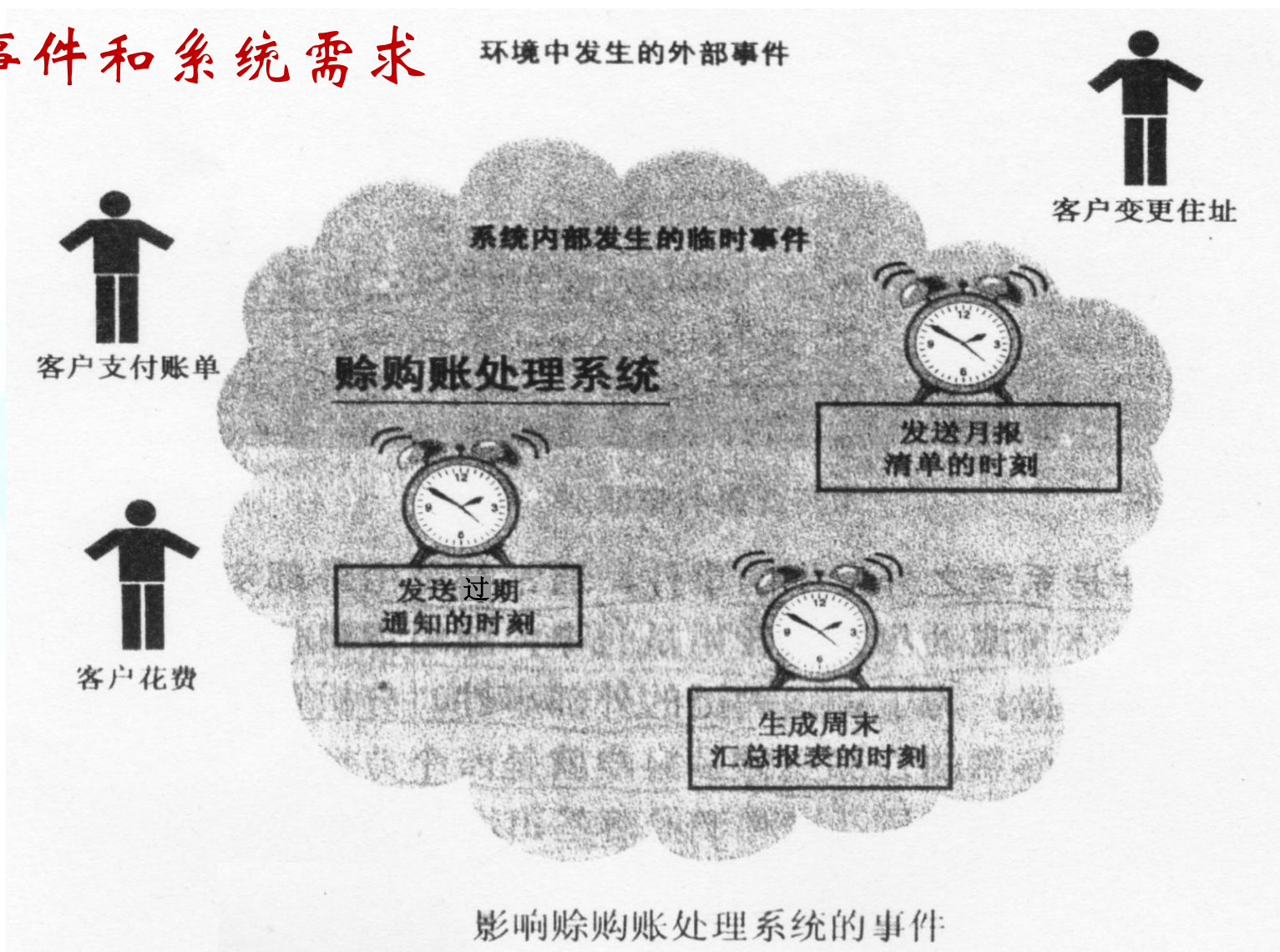
事件：发生在某一特定的时间和地点、可描述并且系统应该记录下来事情

事件和系统需求的关系：事件发生时需要系统做出响应，能列出所有这样的事件就可以搞清楚用户对系统的需求

- 信息系统的所有处理过程都是由事件驱动/触发的
- 当定义系统需求时将**所有事件**罗列出来并加以分析是非常有意义的



2 事件和系统需求



赊购账务处理系统：外部三个事件，内部三个事件



2.1 事件的类型

事件分三大类：外部事件、临时事件、状态事件

- **外部事件**：系统之外发生的事件，通常是由外部实体或系统参与者触发的
- **临时事件**：由于到达某一时刻所发生的事件
- **状态事件**：当系统内部发生了需要处理的情况时所引发的事件



2.1 事件的类型

外部事件实例：

- ◆ 外部实体的需要触发一个事务处理，如客户下订单
- ◆ 外部实体想获得某些信息，如客户查询帐户信息
- ◆ 数据发生改变需要更新，如客户地址变更
- ◆ 管理部门想获取某些信息，如管理者需要销售情况分析报告

临时事件实例：

- ◆ 所需的内部输出结果，如定期报表
- ◆ 所需的外部输出结果，如定期结算单、帐单等
- ◆ 突发的情况，如系统掉电、内存溢出、读写错误等

状态事件实例：

- ◆ 所有外部事件和临时事件都要考虑是否有状态事件发生，如销售引起库存变化、签收引起订单状态变化等



2.2 事件定义

区分事件和随事件发生的一部分交互行为



客户想买一件新衬衫



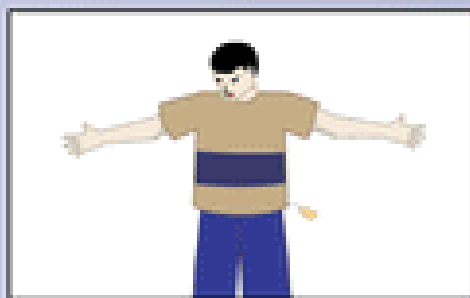
客户开车来到购物中心



客户在Sears试穿衬衫



客户走进沃尔玛



客户在沃尔玛试穿衬衫



客户购买一件衬衫
(该事件直接影响系统)

导致影响系统的一个事件的一系列行为，不能都算作“事件”！



2.2 事件定义

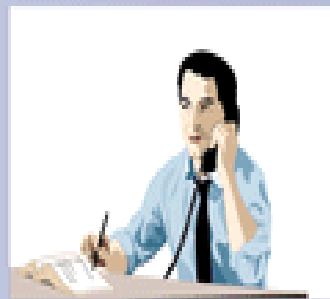
跟踪事务处理生命周期可以得到一系列事件



用户想要一本
商品目录



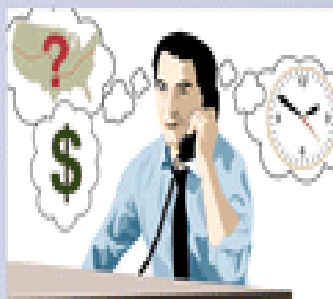
客户想确定商品
信息的有效性



客户下订单



客户修改或取消
订单



客户查询订单状态



客户修改帐户信息



客户退货

导致许多事件的某一特定客户的“事务”处理序列，都算作“事件”！



2.2 事件定义

特别细节的事件在系统分析阶段不予考虑



用户登录系统



用户需要更改密码



系统要改变系统
性能设置



系统崩溃，需要
恢复数据库



该备份数据库了



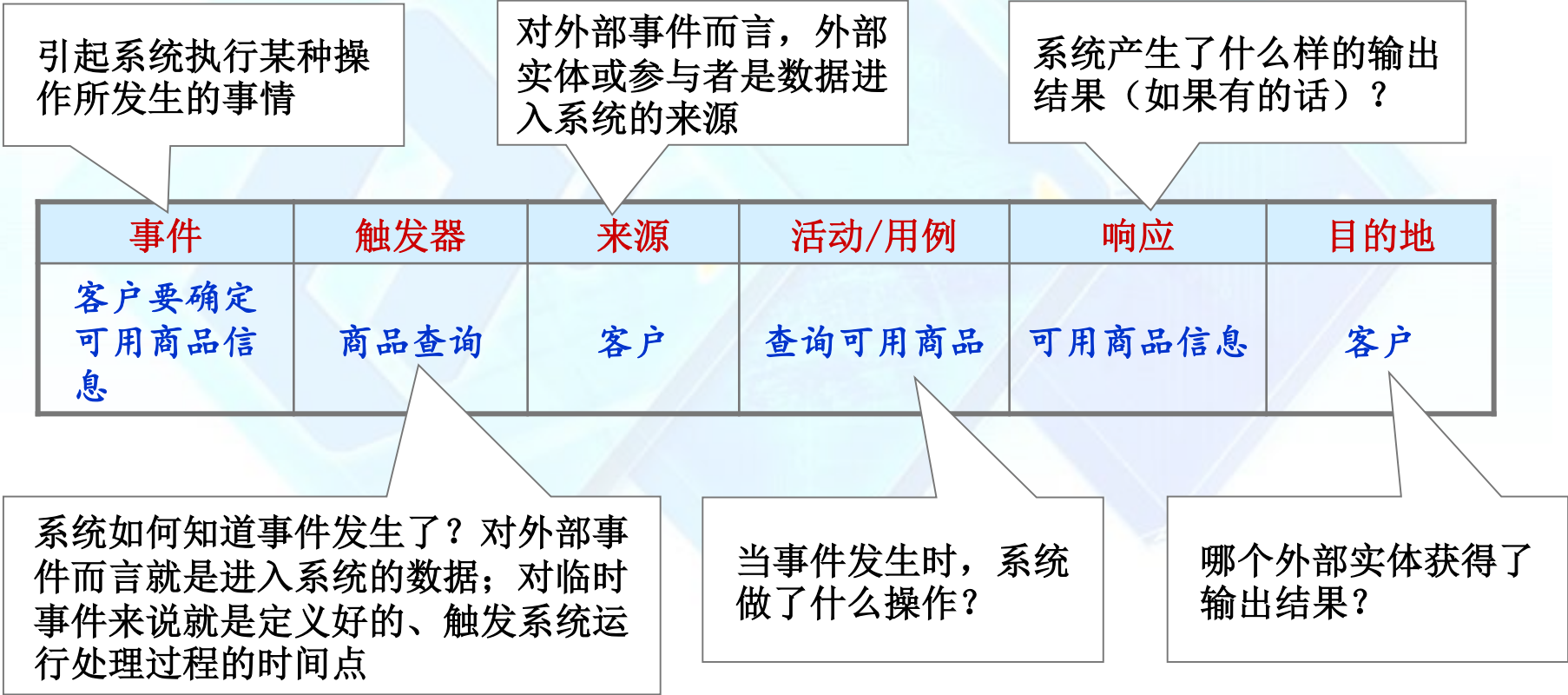
用户该修改密码了

是“事件”，但在设计阶段之前可以忽略的事件



2.3 系统需求建模：事件列表

事件列表属于“描述性模型”





2.3 系统需求建模：事件列表

落基山运动用品商店（RMO）客户支持系统的完整事件表

客户支持系统事件表					
事件	触发器	来源	活动/用例	响应	目的地
1. 客户想确定商品信息的有效性	商品查询	客户	查询可用商品	有效商品信息细节	客户
2. 客户发送订单	新订单	客户	生成新订单	实时连接	信用卡部门
				订单确认	客户
				订单细节	发货部门
				交易处理	银行
3. 客户修改或取消订单	订单修改请求	客户	修改订单	修改确认	客户
				订单修改细节	发货部门
				事务处理	银行
4. 生成订单汇总报表的时刻	周末、月末、季度末、年末		生成订单汇总报表	订单汇总报表	管理部门
5. 生成交易汇总报表	每天下班时		生成交易汇总报表	交易汇总报表	会计
6. 客户或管理人员想查询订单状态	订单状态查询	客户或管理人员	查询订单状态	订单状态细节	客户或管理人员
7. 发货部门按订单发货	订单完成通知	发货部门	记录订单完成情况		



2.3 系统需求建模：事件列表

落基山运动用品商店（RMO）客户支持系统的完整事件表

客户支持系统事件表					
事件	触发器	来源	活动/用例	响应	目的地
8. 发货部确认延期订 货单	延期订货单通 知	发货部门	延 期 订 货 单 通 知	延期订货单通知	客户
9. 客户退货	订单退回通知	客户	生成订单	退货确认	客户
			退回记录	事务处理	银行
10. 生成完成情况汇总 报表的时刻	周末、月末、 季度末、年末		生成完成 情况 汇总报表	完成情况汇总报 表	管理部门
11. 潜在客户索取商 品目录	索取商品目录 的请求	潜在客户	提供商品目录 信息	商品目录	潜在客户
12. 生成潜在的客户信 息报表的时刻	月末		生成潜 在 的 客 户活动报表	潜在的客户活动 报表	市场部门
13. 客户修改账户信 息	客户账户信息 修改通知	客户	更新客户账 户 信息		
14. 市场部门想给客户 发送促销品	促销品细节	市场部门	分发促销品包	促销品包	客户和潜在 客户
15. 管理人员调整对 客户的收费	客户收费的调 整	管理人员	生成客户收 费 调整记录	收费调整通知	客户
				事务处理	银行



3 事物和系统需求

事物：在传统的开发方法中，事物就是构成系统存储信息的相关数据

在面向对象的开发方法中，事物就是在系统中相互交互的对象

事件和事物的区别与联系：

事件 — 发生在瞬间，有一定的随机性

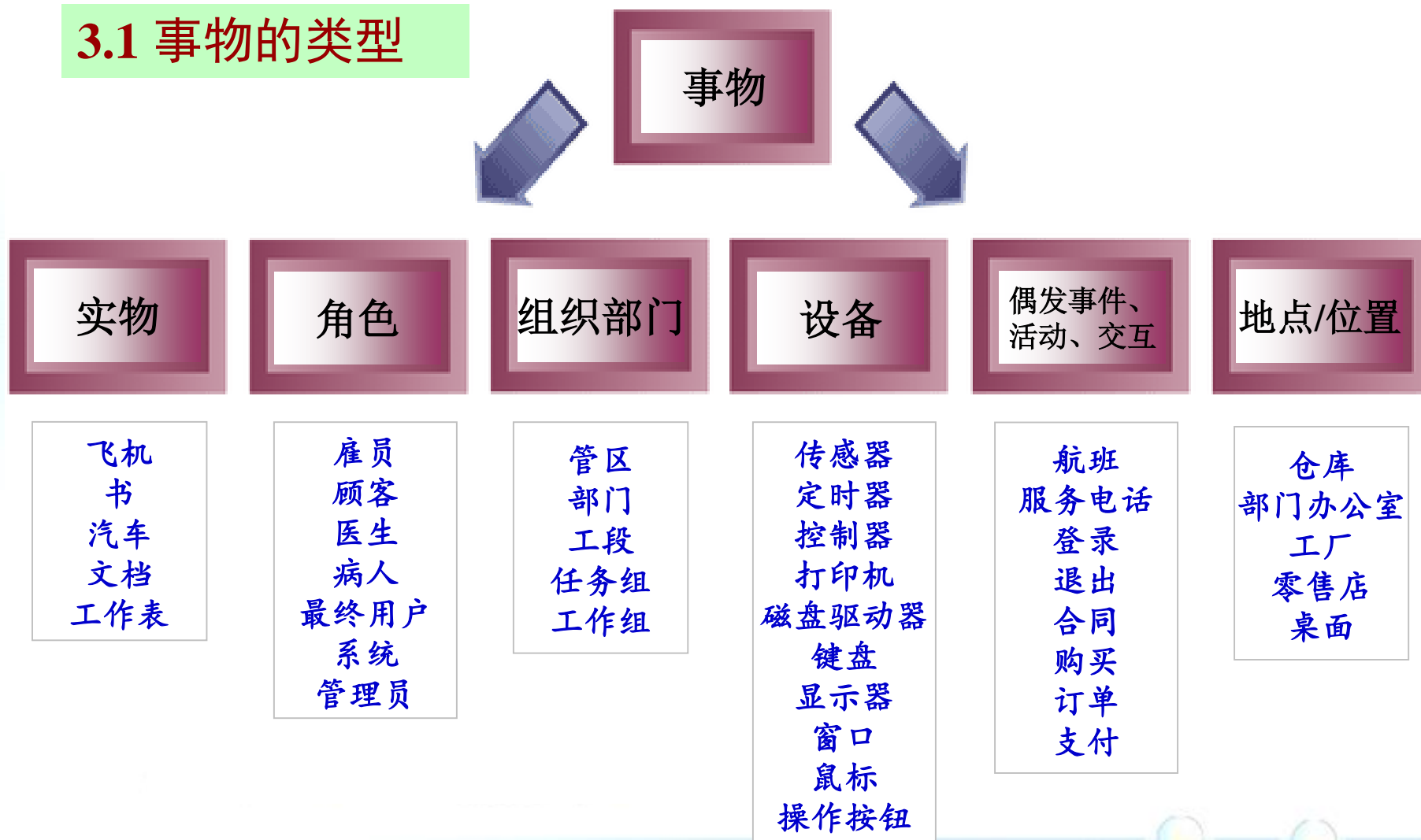
事物 — 客观存在，不以主观意志为转移

联系 — 事件的发生可以改变事物的状态或产生新的事物



3 事物和系统需求

3.1 事物的类型





3 事物和系统需求

3.2 建立事物列表的过程

第一步：收集可能得到的名词

方法1：与客户谈话中获得

方法2：使用事件表及关于每一事件的信息获得

方法3：通过其他需求获取过程得到

第二步：将名词列表精简并确认

(1) 确定名词是否应包含在列表中

(2) 确定每个名词是否应该从列表中排除

(3) 对每个名词提出问题确定是否需进一步研究



3 事物和系统需求

RMO 系统中基于“名词”的事物的部分列表

确定的名词	将该名词作为事物来存储的一些注释
财务人员	我们知道都有哪些人，不需存储
延期订单	一个具体的订单类型，还是订单状态值？需要考虑
延期订单确认	从其他信息产生的输出
银行	仅有一个，不必存储
目录	需要记住，不同季节和年份的目录。应该保留
目录活动报表	可以从其他信息产生的输出。不必存储
目录细节	与目录类似吗？或者与这个目录中产品项类似？需要研究
改变请求	将导致对订单改变进行记录的输入
收费调整	导致一次交易的输入
颜色	关于产品项的一条信息
确认	从其他信息产生的一个输出。不需存储
信用卡信息	订单的一部分？还是客户信息的一部分？需要考虑
客户	是一个关键事物，带有很多细节。应该保留
客户账号	如果包含 RMO 支付计划，则是必要的。需要研究
完成报表	从运输信息产生的输出。不需存储
库存数量	关于产品项的一条信息。需要研究
产品项	RMO 包含在目录中用于销售的。应该保留



3 事物和系统需求

RMO 系统中基于“名词”的事物的一部分列表

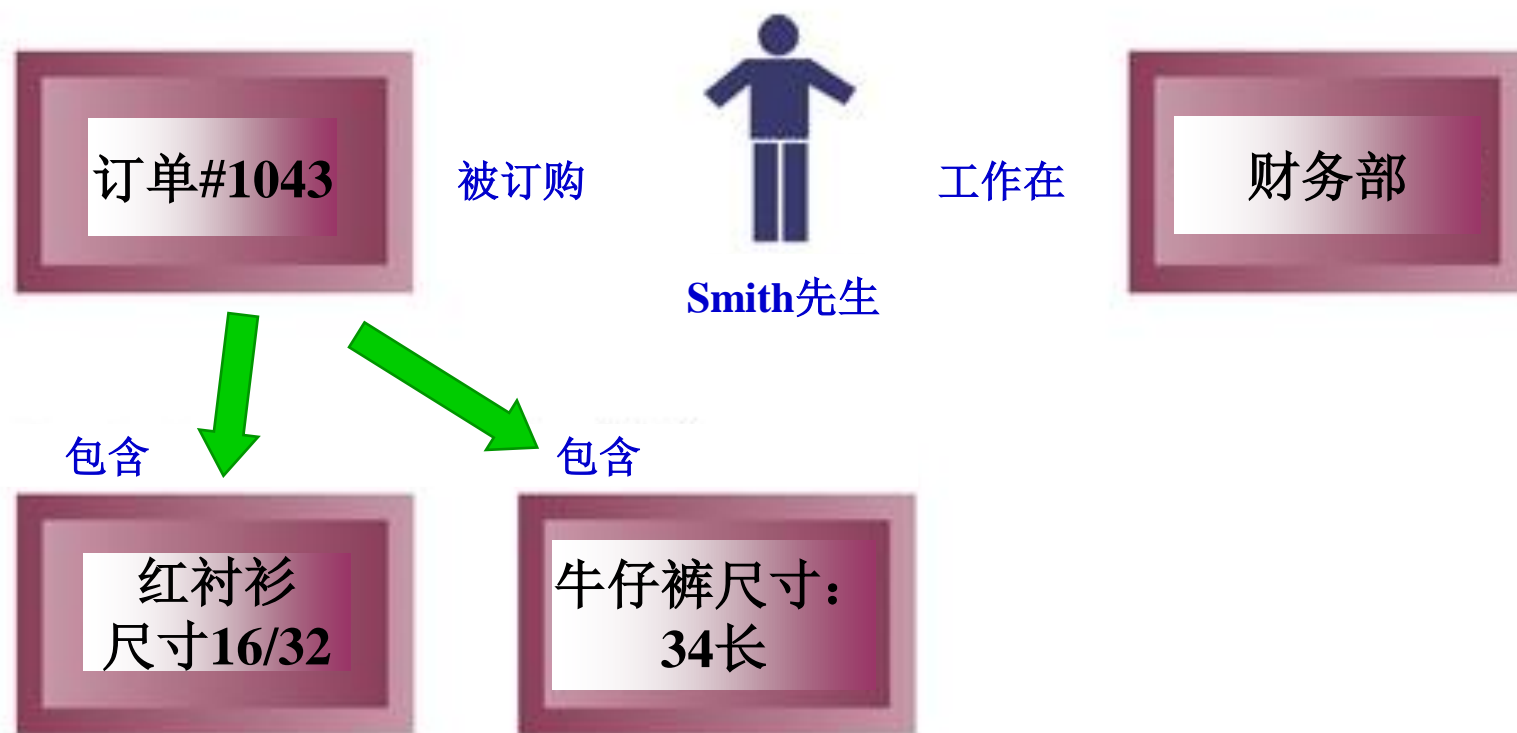
确定的名词	将该名词作为事物来存储的一些注释
管理人员	我们知道都有哪些人，不需存储
市场人员	我们知道都有哪些人，不需存储
销售人员	我们知道都有哪些人，不需存储
订单	关键的系统响应。应该保留
支付方式	订单的一部分。需要研究
价格	产品项的一部分。需要研究
促销资料	一个输出？还是存储在范围之外的文档？需要研究
潜在客户	可能与客户类似。需要研究
退货	订单的反面。应该保留
退货确认	退货信息产生的输出。不必存储
RMO	只有一个。不必存储
季节	目录的一部分？还是还有更多的？需要研究
运输	用于跟踪的关键事物。应该保留
运输人员	他们常有变动且我们需要跟踪订单。应该保留
运输部门	我们的部门，不需存储
运输部门地址	客户的一部分？还是订单的？还是运输的？需要研究



3 事物和系统需求

3.3 事物间的关系

关系：某些事物间自然发生的联系，比如客户发送订单、雇员在某部门工作等



事物之间自然存在的关系



3 事物和系统需求

3.3 事物间的关系

根据每件事物的关联数目来理解每种关系的本质非常重要

基数（重数）： 发生在事物间关联的数目

例如： 一个客户发送量多个订单

一个雇员在一个部门工作等

基数可以： 0, 1, 0..n, 1..n, m..n, 0..*, **

二元关系： 两种不同类型事物之间的关系，如Teacher与Student

一元(回归)关系： 同一类型的两个事物间的关系，例如一个人和另一个人的婚姻关系

***三元关系：** 三种不同类型事物之间的关系，
如Course、Teacher和Student

***n元关系：** n (n为任意数) 种不同类型事物之间的关系



3 事物和系统需求

3.4 事物的属性

属性：有关事物的一条特定信息

标示符（关键字）：能唯一标志事物的一个属性

复合属性：包括了许多相关属性的属性，如客户全名：名+姓

所有客户具有如下属性	每个客户的每个属性都有一个值		
客户编号	101	102	103
名	John	Mary	Bill
姓	Smith	Jones	Casper
住宅电话	555-9182	423-1298	874-1297
公司电话	555-3425	423-3419	874-8546



3 事物和系统需求

3.5 数据实体和对象

数据实体：在传统的系统开发方法中，事物被称为数据实体

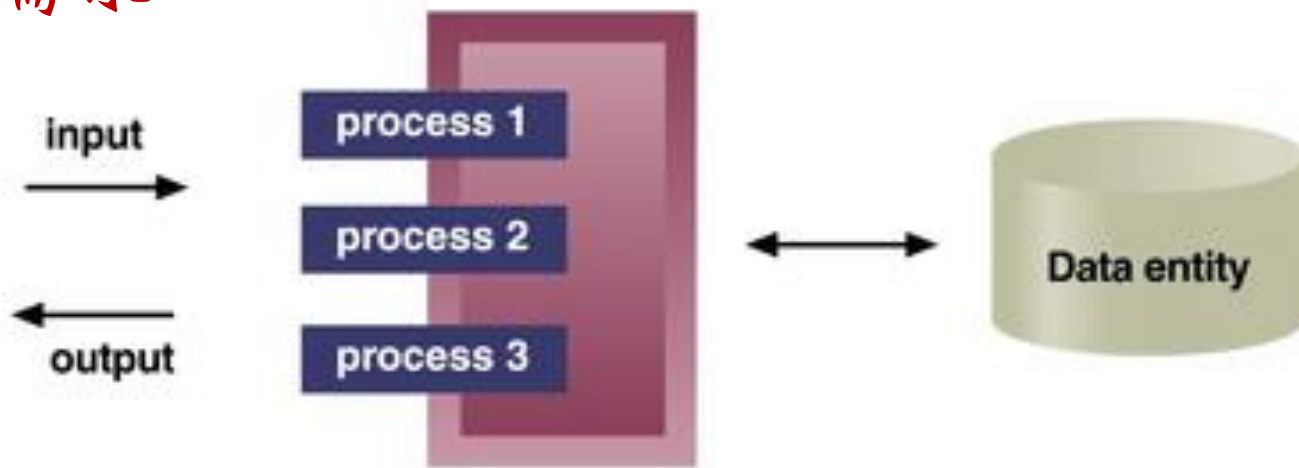
对象：在面向对象的系统开发方法中，将事物称为对象



3 事物和系统需求

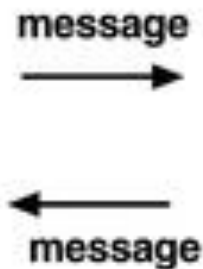
数据实体与过程是分离的

3.5 数据实体



对象将数据和处理数据的方法封装成一个单元

要让某个对象执行某种操作，可以让另外一个对象给该对象发送一个消息

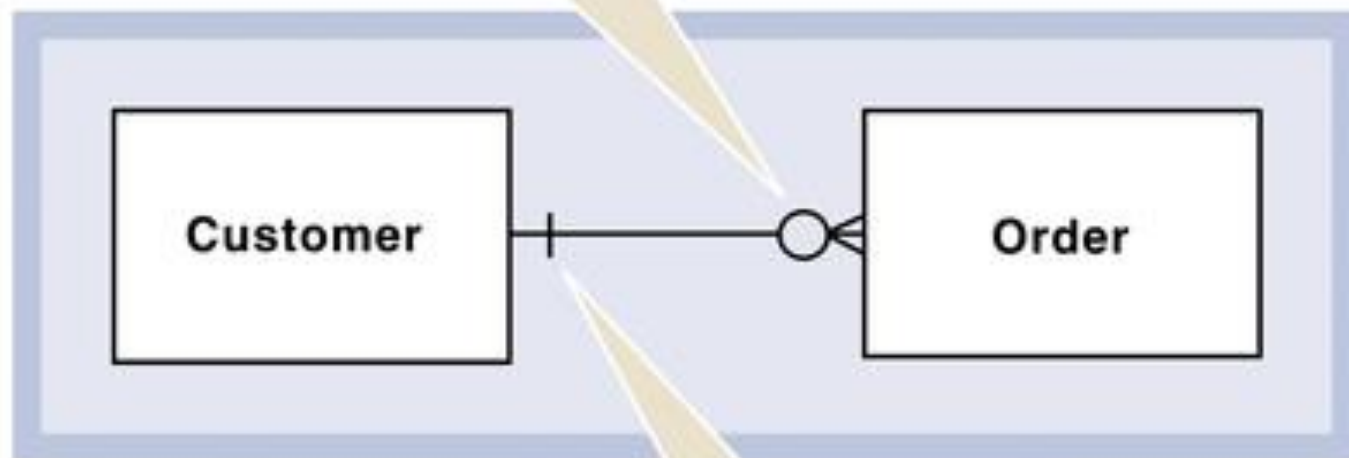




4 实体-关系图

- 传统的系统开发方法都将重点集中在数据存储需求上
- 数据存储需求包括数据实体、数据实体的属性以及它们之间的关系

4.1 ERD概念的实例



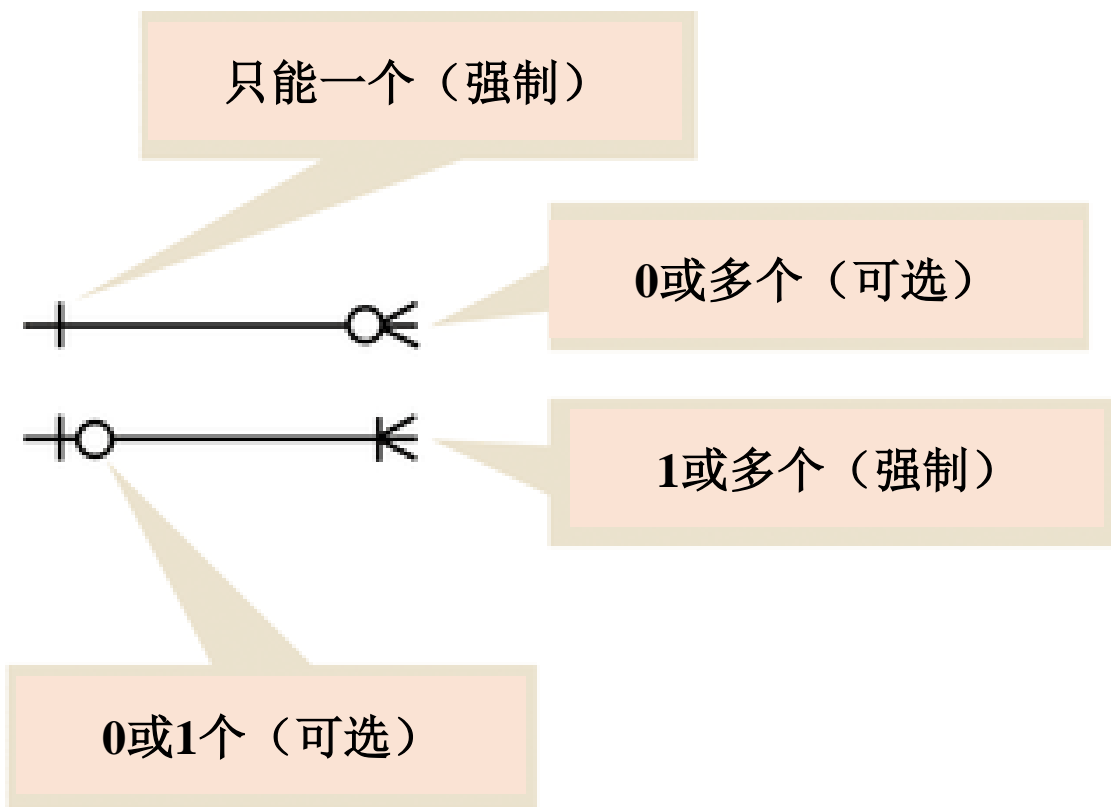
一个客户可以发送
0个或多个订单

一个订单必须对应一个客户



4 实体-关系图

4.1 ERD概念的实例

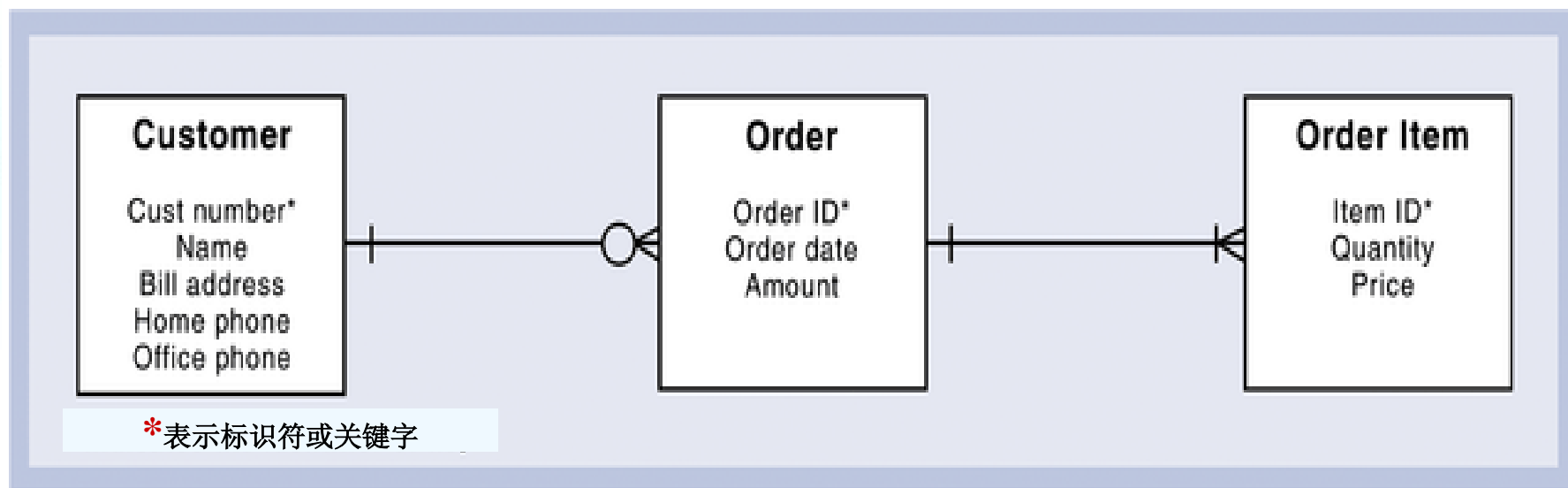


实体之间关系的基数符号



4 实体-关系图

4.1 ERD概念的实例

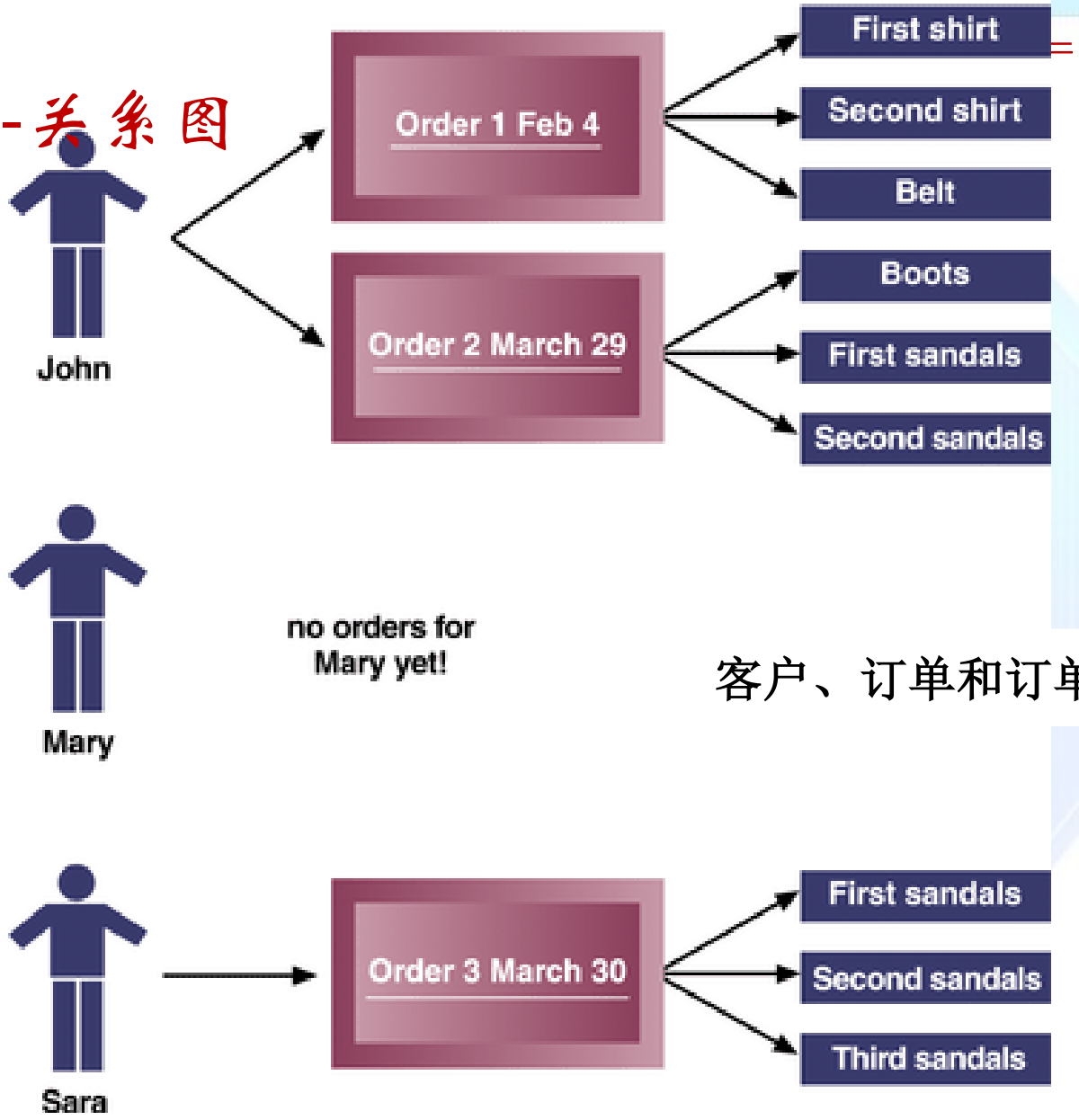


带有属性的扩展ERD图



4 实体-关系图

4.1 E



客户、订单和订单中的商品

该图显示了实际的数据在事务处理中呈现的形式



4 实体-关系图

4.1 ERD概念的实例

- ◆ 分析员在建模的过程中，常常对ERD进行细化的工作就是处理多对多的关系
- ◆ 由于关系数据库中不能直接实现多对多的关系，因此必须建立一个单独的表，来消除多对多的关系
- ◆ **关联实体** – 解决上述问题的人为增加的数据实体，它一定包含两端数据实体的关键字



5 类图

- ◆ 面向对象的方法也强调对系统中所包含事物的理解
- ◆ 面向对象的方法给事物建立的模型即是“类图”
- ◆ “类”和“实体”是明显区别的



5 类图

5.1 有关对象类的更复杂的问题

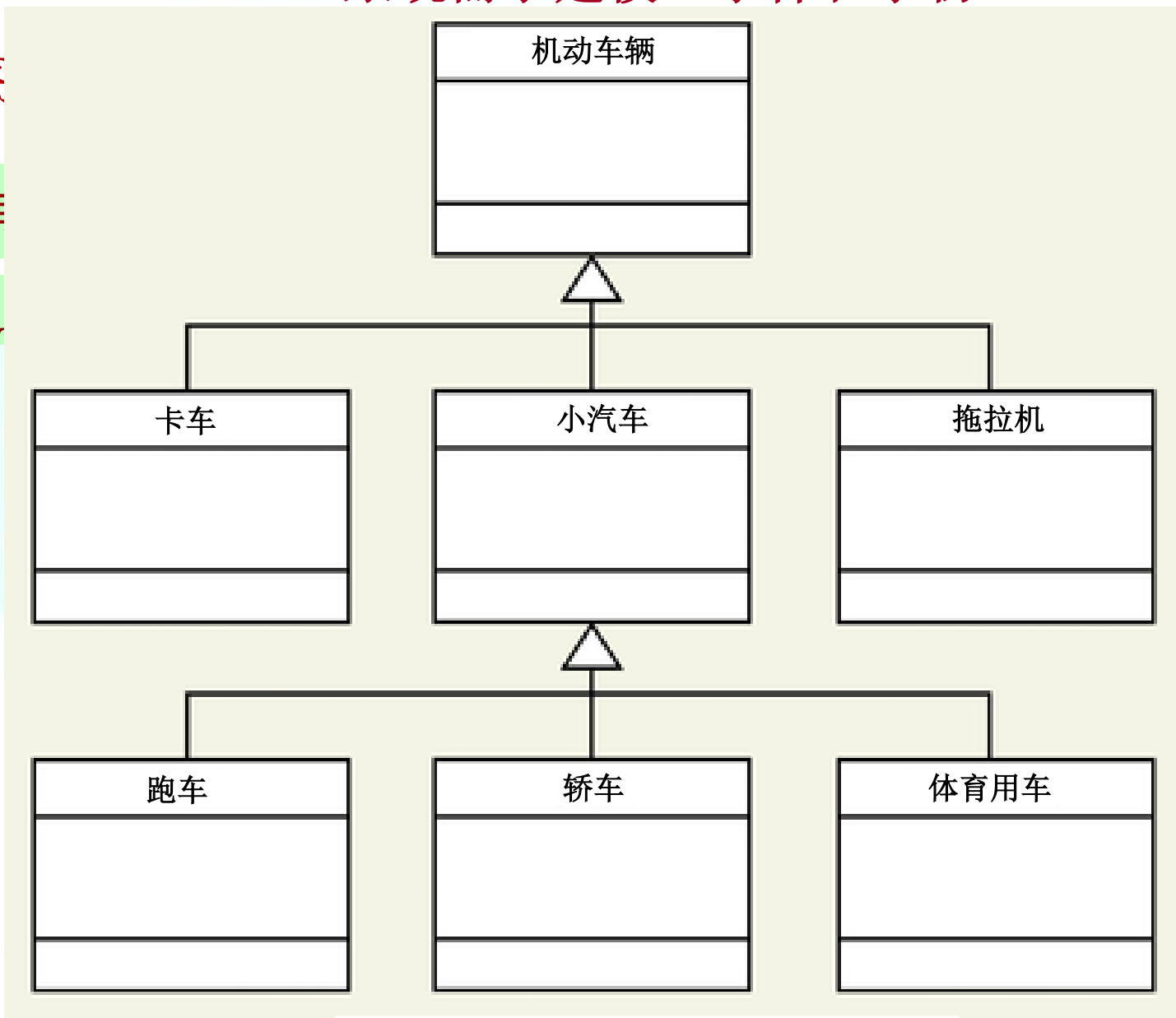
- ◆ **泛化/具体层次图** - 把类按照从最概括的父类到最具体的子类的顺序进行排列的层次图，有时也被称作继承层次图
- ◆ **继承** - 允许子类共享其父类的所有的特征



5 类图

5.1 有

◆ 泛



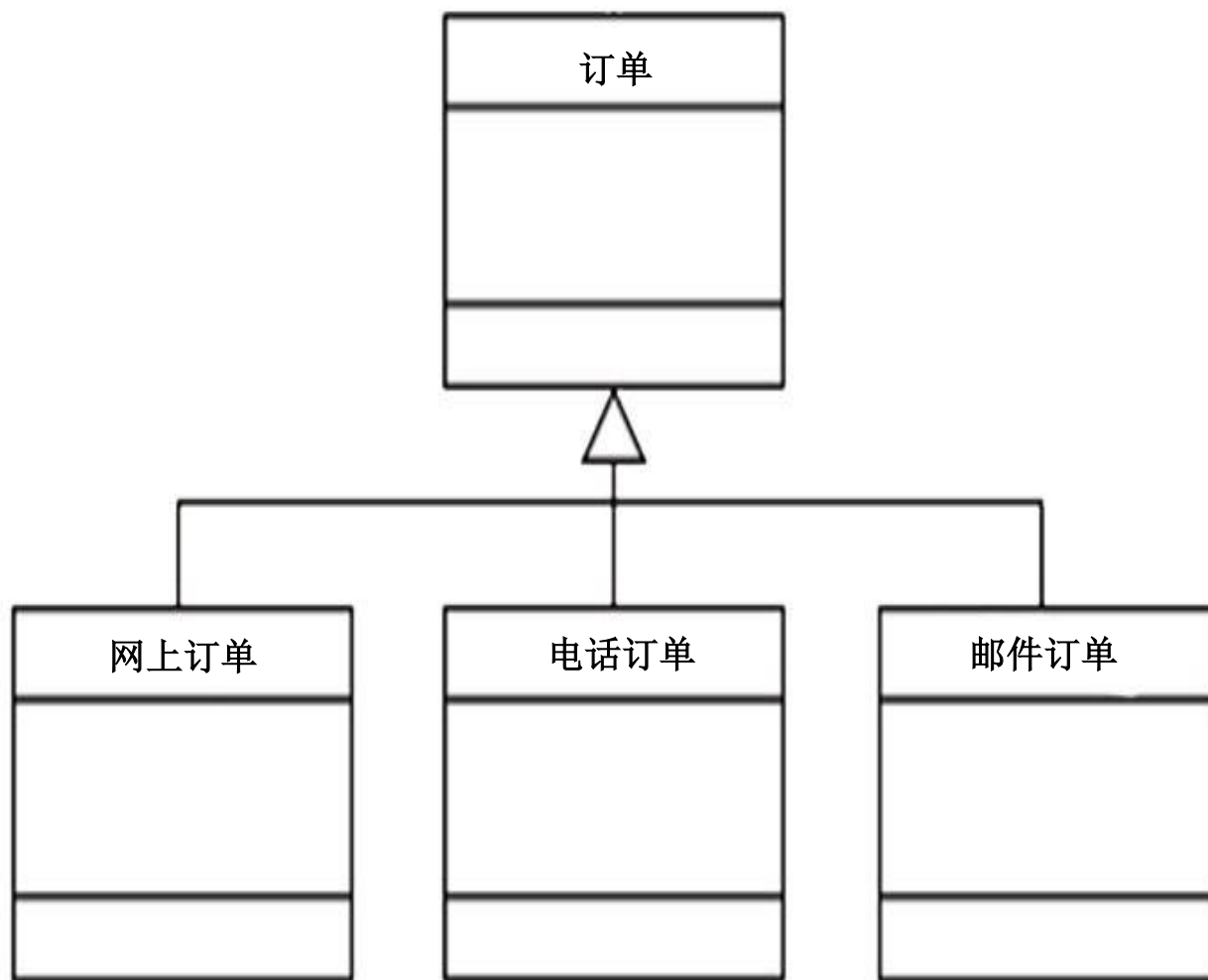
机动车辆的泛化/具体层次图（继承关系）



5 类图

5.1 有关对象

◆ 泛化/具体



订单的泛化/具体层次图（继承关系）



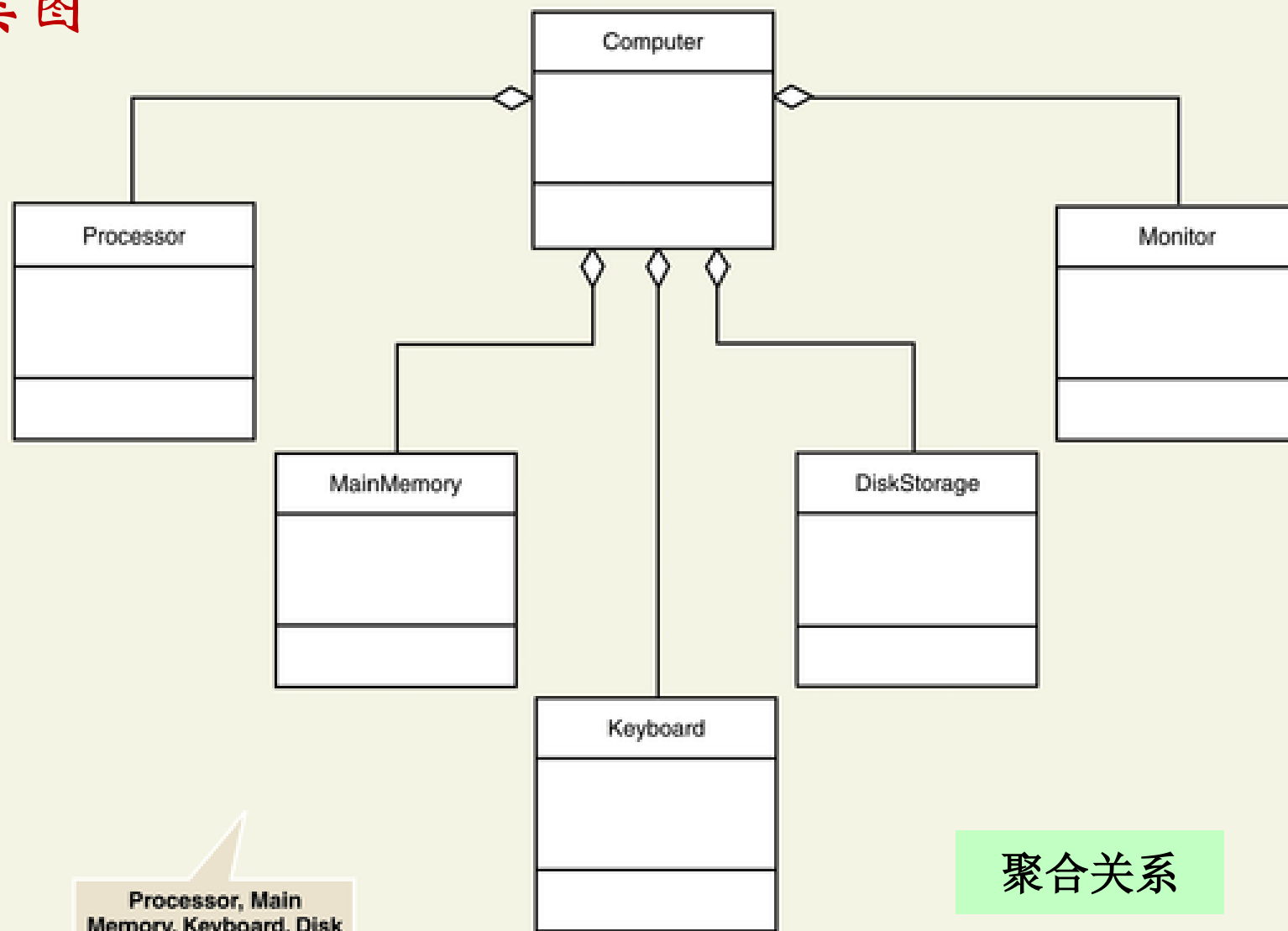
5 类图

5.1 有关对象类的更复杂的问题

- ◆ **整体/局部层次图：** 按照类之间的关联组件将类进行结构化分解的层次图
- ◆ **聚合：** 对象及其各个部分之间的一种整体-局部关系
- ◆ **组成：** 对象及其与它不可分割的各个部分之间的一种整体-局部关系



5 类图



Processor, Main Memory, Keyboard, Disk Storage, and Monitor are parts of a computer

聚合关系



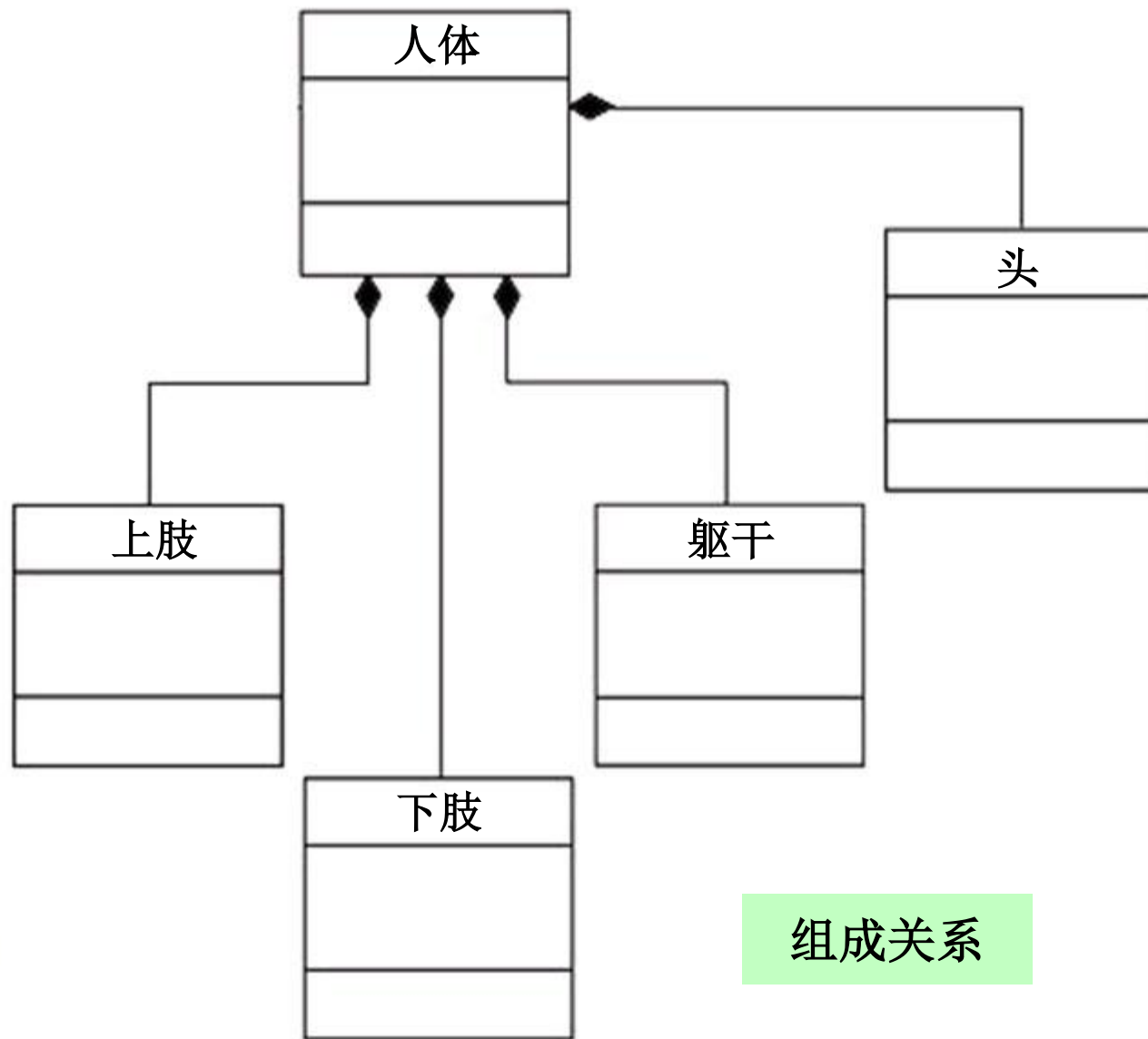
5 类图

5.1 有关对象

◆ 整体/局部

◆ 聚合：对

◆ 组成：对
整



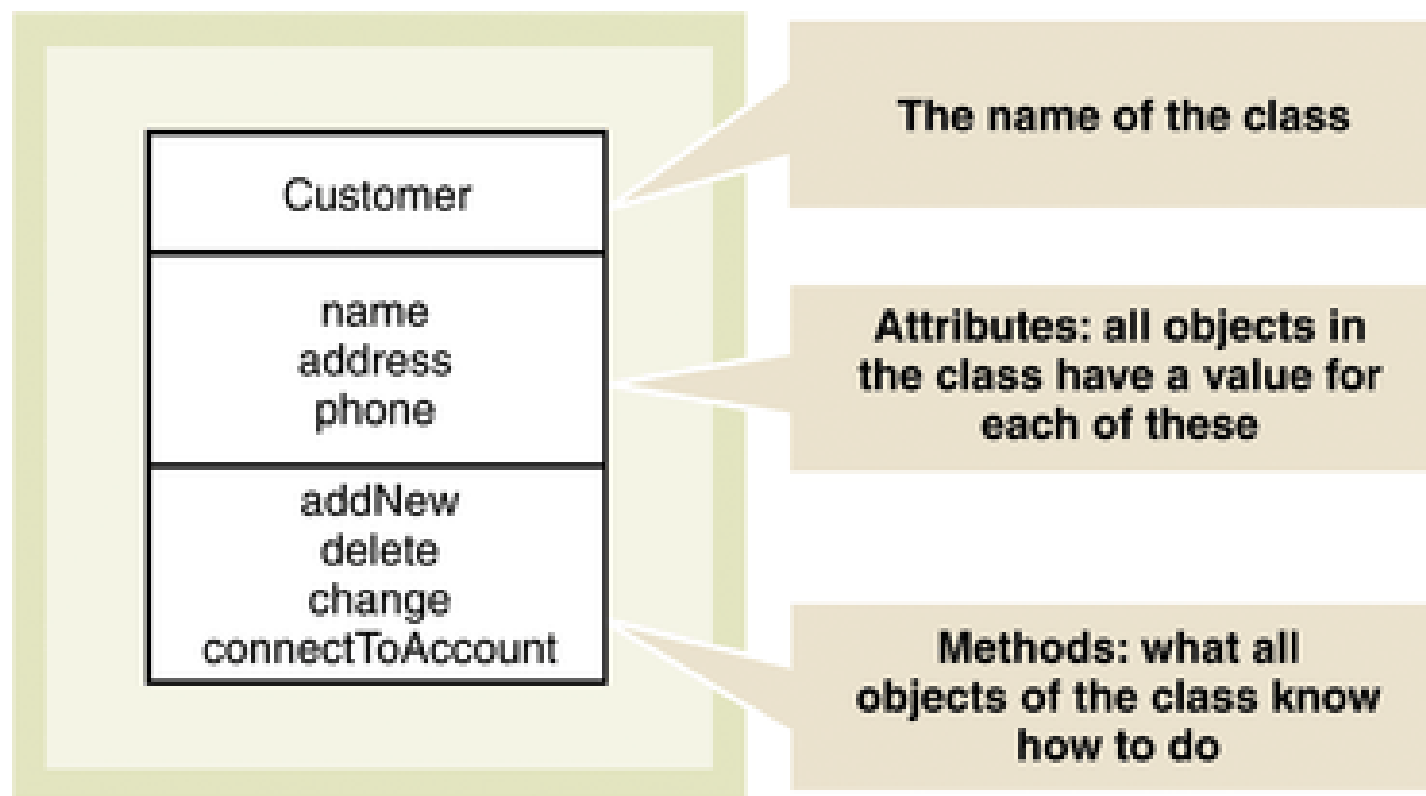
组成关系



5 类图

5.1 有关对象类的更复杂的问题

◆ UML类图符号





5 类图

5.2 类图符号的实例

- ◆ **抽象类：**一种不能被实例化（即不能创建对象）的类，仅为了使其子类能够继承他的属性与方法
- ◆ **具体类：**能够被实例化的类
- ◆ **关联类：**一种表示两种不同类之间多对多的关系的类



事件、事物抽取实例

WaitersOnCall送餐管理系统需求分析过程的对话资料

- Sue Brickford和Tom Brickford在上大学期间都曾在餐馆打工，他们一直梦想着开一家自己的餐馆，但资金有限未能如愿。
- Brickford兄弟发现：许多餐馆都提供外卖，而且一些餐馆（例如比萨饼店）还提供送货上门的服务。
- Sue和Tom认为电话订餐是最佳的选择。于是他们开了一个名为WaitersOnCall的送餐店，和全城各种知名的餐馆联系，接受客户订单并负责将全部饭菜送货上门。当饭店准备好了预定的饭菜后，按批发价交给他们，而饭菜送到后，客户按零售价支付，并付给他们服务费和小费。
- WaitersOnCall刚起步时规模很小，仅包括两家餐馆和一个配送司机。随着生意越做越大，Brickford兄弟意识到：必须有一套专门的计算机系统来支持业务运转。
- 于是他们聘请了信息系统顾问Sam来进行技术咨询。下面是顾问Sam同兄弟俩的关于建立送餐管理信息系统的对话。



事件、事物抽取实例

WaitersOnCall送餐管理系统需求分析过程的对话资料

- **Sam**: “在你们打理生意时, 哪些事情促使你们决定采用计算机管理? 告诉我通常这些业务是如何进行的。”
- **Sue**: “是这样, 当客户打电话订餐时, 我需要把它记下来, 然后通知给相应的餐馆。我需要决定派哪一个司机去送货, 因此要司机打电话告诉我他们什么时间有空。有时, 客户会又打电话更改订单内容, 因此我必须找到原始订单, 然后通知餐馆更改。”
- **Sam**: “好的, 那你们又怎么管理现金呢?”
- **Tom**: “司机取饭菜时会从餐馆直接拿到账单的副本, 账单和我们的计算应该是一致的, 然后司机送货时收取相应的现金并加上服务费。在下班时, 司机报账, 我们把司机收到的现金汇总起来, 和我们的记录进行比较, 所有的司机都交完账后, 我们需要开张银行存款单, 存入当天的现金总收入。每周末, 我们按提前约定的批发价来计算欠餐馆多少钱, 把结算单和支票寄给他们。”



事件、事物抽取实例

WaitersOnCall送餐管理系统需求分析过程的对话资料

- **Sam**: “那你们还想从这个系统中获取别的信息吗?”
- **Sue**: “如果每周末能统计出每个餐馆有多少订单、城里每个区有多少订单等信息就更好了。这能帮助我们制定广告策略及与餐馆的合同,而且我们还需要每月财务状况统计结果。”
- 在**Sue**及**Tom**说话时, **Sam**记下了几个要点,画了几张草图。之后,他花了一些时间仔细考虑,总结出了**WaitersOnCall**的需求状况。



事件、事物抽取实例

Waiters on Call—餐馆送货系统—事件与事物的抽取

1.在发生如下**事件**时，系统
可以进行处理：

- 客户打电话下订餐订单
- 客户打电话修改订单
- 送货司机汇报工作情况
- 送货司机上交一天的收入

2.在特定时间内系统产生所
需信息（**临时事件**）：

- 生成日结算存款单
- 生成周末餐馆支付帐单
- 生成周销售报表
- 生成月财务报表

3.系统还需要处理以下**事件**：

- 新餐馆加盟
- 餐馆更改了菜单
- 雇佣了新的送货司机（送餐员）
- 送货司机（送餐员）离职

4.需要存储的信息（**事物**）：

- 餐馆
- 菜单
- 客户
- 订单
- 订单支付帐单
- 司机（送餐员）