



# 实体联系模型



PART 01  
实体-联系模型



PART 02  
约束



PART 03  
实体-联系图



PART 04  
扩展的E-R  
特性



PART 05  
实体-联系  
设计



PART 06  
数据建模的其  
他表示法及工  
具



### 学习目标：

- ✓ 掌握实体-联系模型的实体、联系和属性；
- ✓ 掌握实体-联系模型约束、扩展 E-R 属性；
- ✓ 学会实体-联系图的设计；
- ✓ 熟练使用实体-联系图的设计工具；
- ✓ 了解统一建模语言 UML；
- ✓ 了解数据建模的其他方法。



实体-联系模型需要了解三个基本概念：

- 实体
- 属性
- 联系

## ● 实体

实体是客观世界中描述客观事物的概念，是一个数据对象。实体可以是人，也可以是物或抽象的概念；可以指事物本身，也可以指事物之间的联系，如一个人，一件物品，一个部门等都可以是实体。



## 实体集

- 实体集就是具有相同属性（或性质）的实体集合。例如某个银行的所有客户的集合可以被定义为实体集。实体集可以分为**强实体集**和**弱实体集**。
- **强实体集**指不依赖于其他实体集存在的实体集。强实体集的特点是：每个实例都能被实体集的主键唯一标识。强实体集有时也称为父实体、主实体或者统治实体。
- **弱实体集**指依赖于其他实体集存在的实体集。弱实体集的特点是：每个实例不能用实体集的属性唯一标识。弱实体集也称为子实体、依赖实体或从实体。



### 属性

- 属性指实体具有的某种特性。属性用来描述一个实体。
- 在E-R图中，属性用椭圆形框表示，并用无向边将属性与对应的实体连接起来。实体的主键属性用下划线加以标注。
- 如医生实体可由编号，姓名，年龄，性别，所在病区，技术级别等属性来刻画。在一个实体中，能够唯一标识实体的属性或属性集称为“实体标识符”。但一个实体只有一个标识符，没有候选标识符的概念。实体标识符有时也称为实体的主键或主属性。

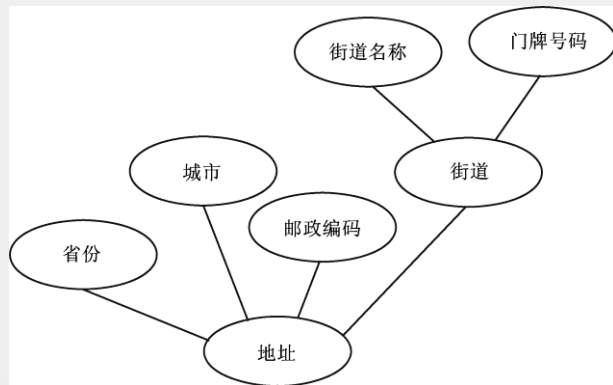


### 属性

属性被分为简单属性、复合属性、单值属性、多值属性、派生属性、空值属性。

- 简单属性和复合属性

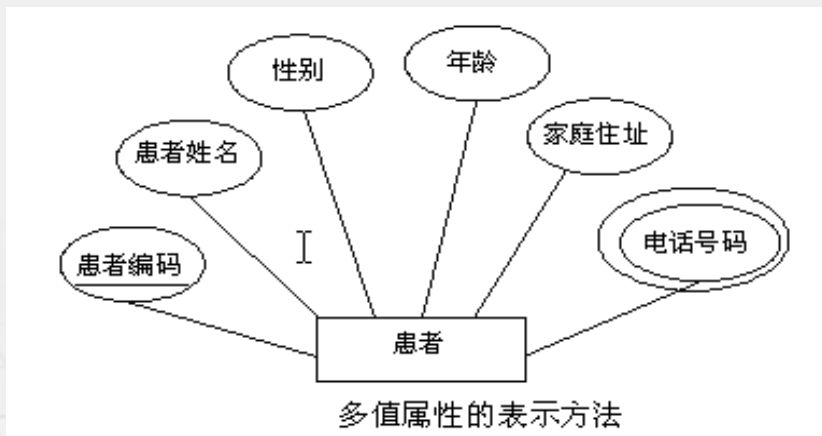
- 根据属性的类别划分。
- 简单属性是不可再分的属性。
- 复合属性是可以再划分为更小的部分（即属性可以嵌套）。
- 例如：患者的地址属性可分解为：省份、城市、街道和邮政编码等4个子属性，而街道又可分为街道名、门牌号两个子属性。



复合属性的层次结构



- 单值属性和多值属性
  - 根据属性的取值特点划分。
  - 单值属性是指同一实体的属性只能取一个值。
  - 多值属性是指同一个实体的某些属性可能对应一组值。
  - 多值属性用**双椭圆形**表示。
  - 例如：具有多个电话号码的患者实体表示，  
在应用中限定数量，如不超过2个。



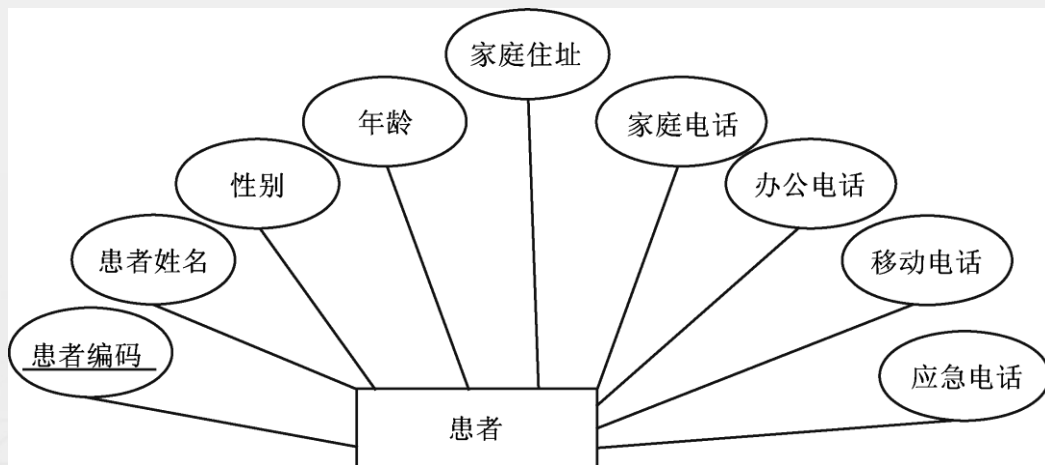
多值属性的表示方法  
多值属性的表示方法



- 多值属性的变换通常有两种方法

方法一：将原来的多值属性用几个新的单值属性来表示。

例如：患者的联系电话可以用家庭电话、办公电话、移动电话等进行分解。



多值属性的变换表示（新属性方法）



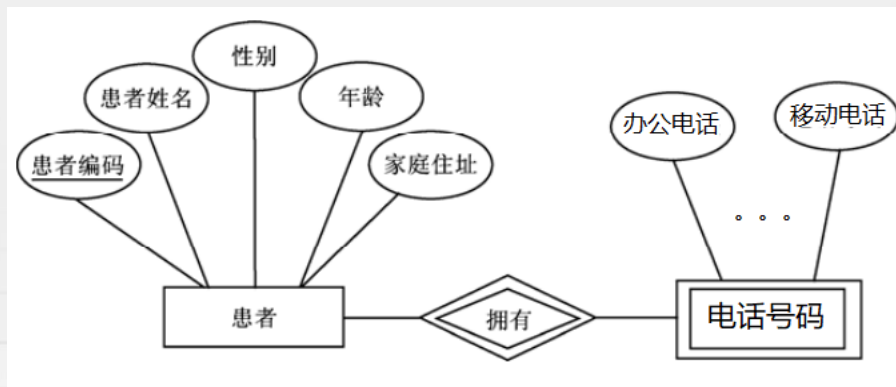


- 多值属性的变换通常有两种方法

方法二：将原来的多值属性用一个新的实体类型表示。

新的实体类型和原来的实体类型之间是1:N联系，新的实体依赖于原来的实体而存在，因此称新的实体为**弱实体**。在E-R模型中，弱实体用**双线矩形框**表示，与弱实体相关的联系用**双菱形框**表示。

例如：在医院管理信息数据库中可以增加一个电话号码弱实体，患者实体与该电话号码弱实体之间具有“拥有”联系。

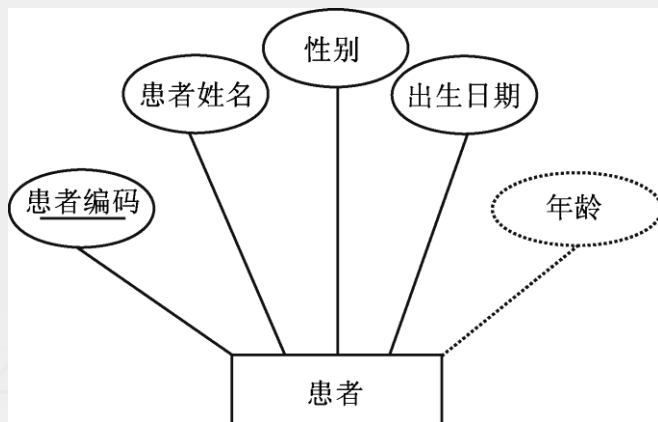


多值属性的变换表示（弱实体方法）



## ● 派生属性

- 通过具有相互依赖的属性推导出来的属性称为派生属性（Derived Attribute）。
- 派生属性用虚线椭圆形与实体相连。
- 例如：在患者实体中，患者年龄可以由患者的出生日期与当前就诊日期推导出来。



派生属性的表示



- 空值属性
  - 当实体在某个属性上没有值时应该使用空值（Null Value）。
  - 例如，新应聘到医院的医生尚未分配岗位，则该医生的所属科室的属性值应该为空值Null，表示未知或无意义。



## 联系集

- 联系表示一个或多个实体之间的关联关系。
- 在信息世界中, 事物之间的联系可分为两类: 一是实体内部的联系, 如组成实体的各属性之间的关系; 二是实体之间的联系。这里我们主要讨论实体之间的联系。
- 联系是实体之间的一种行为, 一般用**动词**来命名联系, 如“**就诊**”, “**交费**”, “**发药**”等。



### 映射基数

又称为基数比率，指的是一个实体通过一个联系能同时与多少个实体相关联。现有实体集A和B，映射的基数分为四种情况：

- 一对一
- 一对多
- 多对一
- 多对多



### 参与约束

如果实体集A中的每一个实体都参与到联系集R的至少一个联系中，称实体集A全部参与到联系集R中。如果实体集A中部分实体参与到联系集R的联系中，我们称实体集A部分参与联系集R。



### 实体

- 在 E-R 模型中，实体用**方框**表示，方框内注明实体的名称。实体名通常用大写字母开头的具有特定含义的英文名词表示。但建议实体名在需求分析阶段使用中文表示，而在设计阶段再根据需要转换成英文形式，这样有利于分析人员和用户之间的交流。
- 例如：医生实体表示。



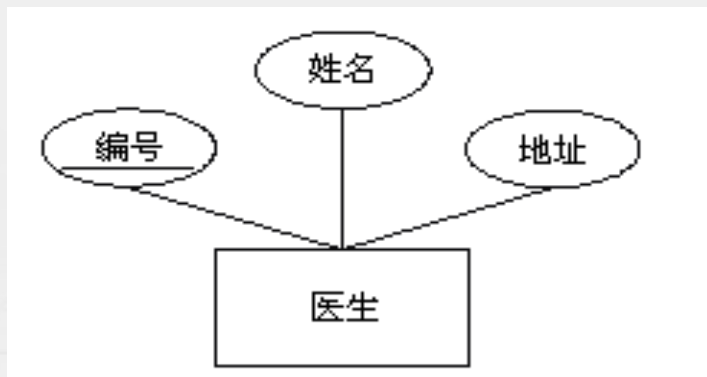
医生

医生实体表示示意图



### 属性

- 在E-R图中，属性用**椭圆形框**表示，并用无向边将属性与对应的实体连接起来。  
实体的主键用下划线加以标注。
- 例如：医生实体可有编号、姓名、地址等属性。

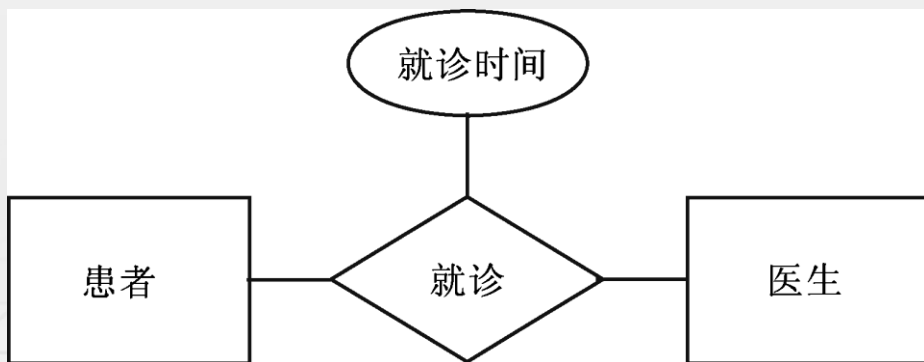


医生实体的属性表示示意图



### 联系

- 在E-R图中，联系用**菱形框**表示，并用无向边将其与相关的实体连接起来。
- 联系也可能会有属性，用于描述联系的特征，如就诊时间、发药数量等，但联系本身没有标识符。
- 例如：患者与医生之间的关系表示。



患者医生之间的关系表示示意图





采用E-R模型能够建立一些基本的数据库应用系统，如进销存管理，客户关系管理等系统。然而，自20世纪80年代以来，许多新型的数据库应用系统得到快速的发展，它们比传统的应用系统提出了更多的数据库需求。这些数据库应用系统包括计算机辅助设计(CAD)，计算机辅助制造(CAM)，计算机辅助软件工程(CASE)，办公信息系统(OIS)和地理信息系统(GIS)等。

由于只使用E-R模型的基本概念，已经无法充分地表示这些新的复杂应用系统。基于以上这些情况，促使新的语义建模概念的发展。支持附加语义概念的E-R模型也称为(Enhanced Entity-Relationship, EER)模型。下面将学习扩展的E-R特性。



### 特殊化/概化

特殊化/概化的概念与一些特殊实体类型及属性继承联系在一起，这些特殊的实体类型称为超类（Superclass）和子类（Subclass）。

- 超类和子类的概念

- 一个实体类型表示一些同类型实体的集合。
- 这一实体类型可能包含一些子集，子集中的实体在某些方面区别于实体集中的其他实体，也可以将实体类型组织成包含超类和子类的分层结构。
- 当较低层上实体类型表达了与之联系的较高层上的实体类型的特殊情况时，就称较高层上实体类型为**超类型**，较低层上实体类型为**子类型**。



### 属性继承

- 一个子类也是一类实体，因而子类也可以有一个或多个自己的子类。
- 同样，子类的子类也可以有自己的子类，以此类推。
- 当一个子类不止有一个超类时，这个子类则为共享子类。
- 共享子类的成员必须是所有相关超类的成员。
- 因此，超类的属性都被共享子类所继承，同时，共享子类还可以有子集的附加属性，这种继承称为多重继承。
- 例：“社保患者”子类的成员继承了超类“患者”的所有属性，包括患者编码、姓名、性别、年龄等，同时还具有“社保患者”子类特有的属性“社保编号”和“社保类型”。



### 特殊化过程

- 通过标识实体成员的差异特征使成员间的差异最大化的过程。
- 特殊化是一种**自上而下**的方法。这种方法定义一系列的超类和它们相关的子类，而子类的定义是建立在超类中实体之间差异特征的基础之上。



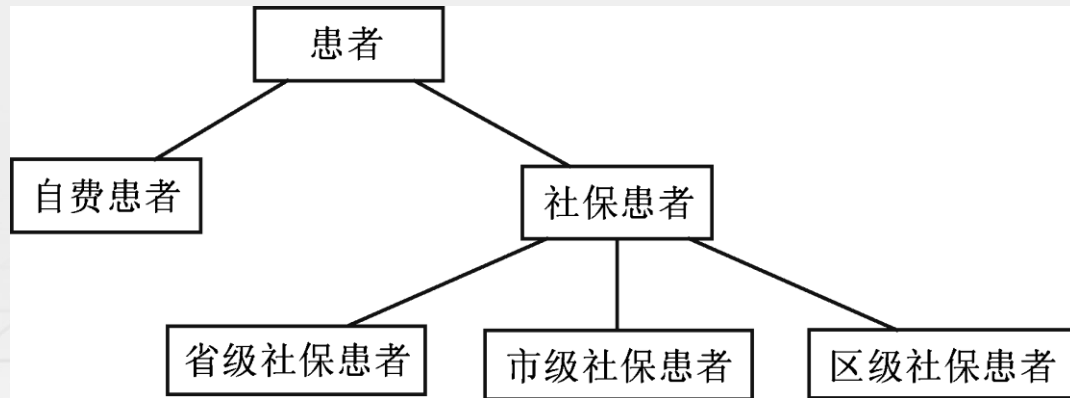
### 特殊化过程

- 当为一个实体类型确定子类时，将属性和特殊子类关联起来，并确定子类之间的联系，以及子类和其他实体类型或子类之间的联系。例如，对“患者”实体进行特殊化过程时，需要确定该实体成员之间的差异，如成员的独特属性和联系。
- 具有不同医疗费用报销处理方法的患者类型，如“自费患者”和“社保患者”，它们拥有各自特有的属性，所以就将“患者”确定为超类，“自费患者”和“社保患者”确定为“患者”超类的子类。



### 概化过程

- 通过标识实体成员间的共同特征使成员间的差异最小化的过程。概化是一种**自下而上**的方法，最终的结果是从一些最初的实体类型中概化出一个超类。
- 例如：患者实体的特殊化和概化。



特殊化/概化示例



E-R模型的设计分为**局部设计**与**全局设计**两种方式。

### ● 设计局部E-R模式

主要工作是要确定出实体和联系的定义、属性的分配，以及根据系统的实际情况，恰当地划分出各个分系统的局部结构范围。

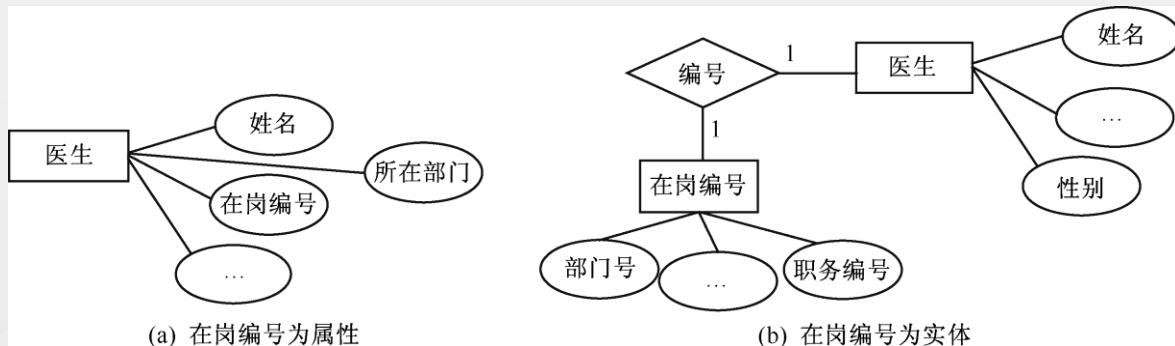
数据抽象后得到的实体和属性需要做适当的调整，而实体和属性是相对的概念，在调整的过程中应遵循两个基本原则：

- “属性”是不可分的最小数据项，不能再具有需要描述的性质。
- “属性”不能与其他的实体具有联系，联系只能发生在实体之间。



### ● 设计局部E-R模式

和实体相比较而言，属性更加简单，不具备独立的使用特征，因而不能与其他实体具有联系。如果某个实体对象对于具体应用来说没有进一步划分的必要，并且仅作为对某类实体特征的描述，该实体对象可被划分为属性。下图给出了在岗编号分别作为实体和属性两种类别的情况。



在岗编号两种不同的抽象方式

图(a)中，用户不关心在岗编号的应用，只作为表征医生的一种属性存在。在图(b)中，实体医生没有标记部门的属性，而是通过与之联系的在岗编号实体的属性来决定医生所在的部门。





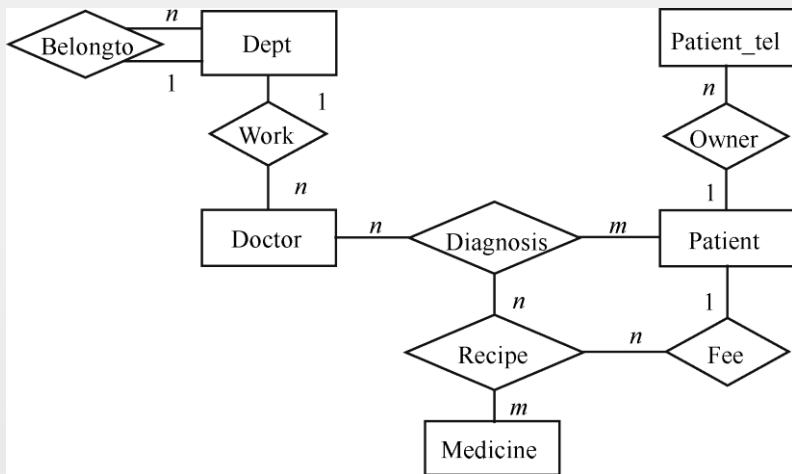
### ●设计局部E-R模式

- 一般情况下，只要能够作为属性对待的数据对象，应尽量设计为属性，以达到简化E-R图的目的。随后更多的设计工作将放在对局部子模式结构的范围确定上。
- 采用模块化的设计方法，可把大系统划分为多个相对独立的子系统。对局部子模式的划分主要需要处理好子模块的内聚性和各个子模块之间的接口情况。

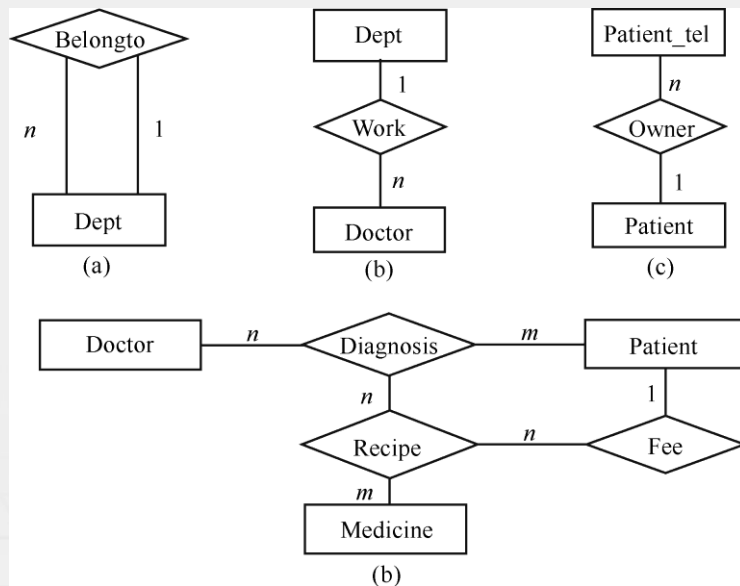


### ●设计局部E-R模式

图a给出了对HIS系统的E-R全局图；图b是针对该全局模式的局部划分。



HIS中的E-R全局图（不含属性）

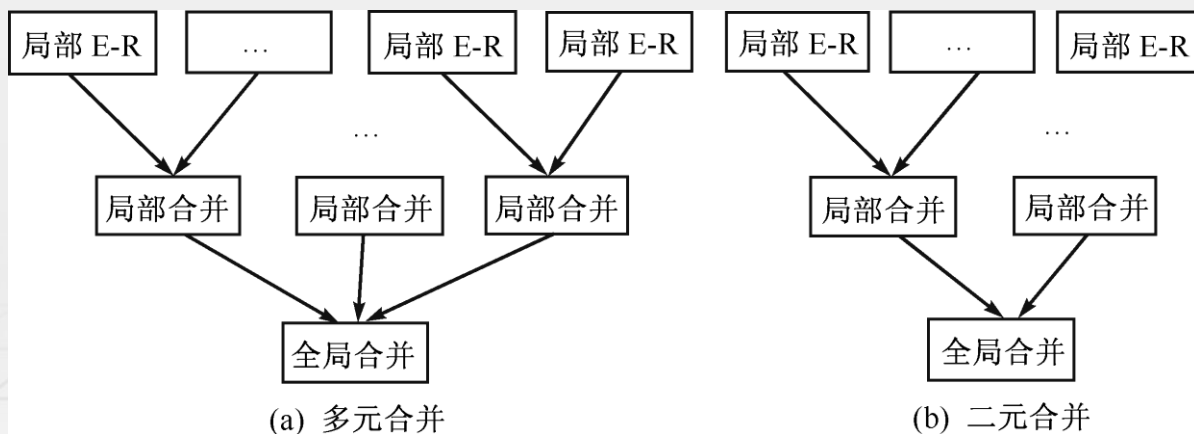


HIS中的局部E-R图（不含属性）



### ● 全局E-R视图集成

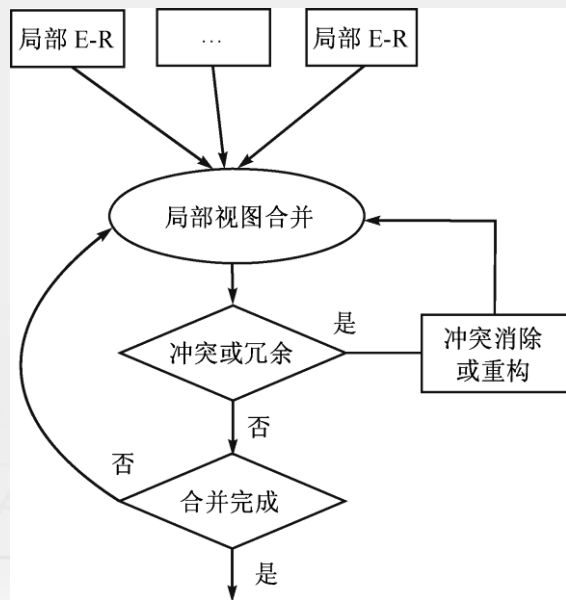
- 在局部E-R模式设计完成后，需要集成这些局部模式，形成全局的E-R模型。
- 常用的集成方法有多元集成法和二元集成法。
- 多元集成 是一次性把多个局部E-R模式集成为一个全局E-R视图；二元集成则把局部模式两两合并，然后再逐层向上合并成全局视图。
- 例如：局部模式合并成全局E-R模式



局部视图的合并示意图



- 具体选择哪种合并方法应主要根据实际系统的复杂情况来考虑，但不论用哪种合并方式，都需要考虑合并过程中可能存在的冲突情况。
- 例如：全局合并时对冲突的消除与重构过程。



视图集成冲突消除与重构



## 冲突的消除

- 属性冲突

这类冲突最容易产生于在不同局部模式中**使用同一属性时采用了不一致的标记**。在实际的应用中，属性冲突的问题可以通过部门协商方式解决，也可以根据实际应用需求考虑是否将属性统一或分离表示。

- 命名冲突

局部模式中使用的数据对象名字与其他模块产生冲突。同名异义和异名同义是常见的命名冲突情况。可以考虑协商统一命名方式解决。

- 结构冲突

结构冲突常见的一种情况是同一实体在不同的E-R局部视图所包含的属性个数不完全相同。实际情况中可考虑取属性的合集来解决这一问题。



## 消除冗余与优化

- 冗余存在于实体数据和实体间的冗余联系。
- 冗余数据是指可由基本数据导出得到的数据。
- 冗余的联系则是指可由其他关系联合导出的联系。
- 冗余也是相对的，有时考虑到性能和效率等综合因素，一些冗余的存在还是可以被接受的。



## 用实体集还是用属性

举例：现在有一个关系，职员（职员姓名，职员ID，电话），但是电话有好几种，可以有家庭联系电话和手机联系电话，应该怎么办？

解：需要考虑到是使用实体集还是用属性。显然电话可以有“家庭”、“办公室”、“移动电话”，因此要考虑把电话作为一个新的实体，不再是职员实体中的一个属性，重新定义如下：

实体集：职员（职员姓名，职员ID）

实体集：电话（电话号码，电话类型）

联系集：职员-电话（职员ID，电话号码）



### 用实体集还是用属性

举例：将电话处理成为职员实体下的一个属性，表示对于每个员工，正好有一个电话号码与之相联系；通过将电话定义为一个多值属性来允许一个员工有多个电话。这两种定义模式主要有什么差别？

解：需要具体问题具体分析。区分它们主要依赖与被建模的实际企业的结构，以及被讨论的属性的相关语义。

举例：究竟怎么区分实体和属性？

解：需要具体问题具体分析。区分它们主要依赖与被建模的实际企业的结构，以及被讨论的属性的相关语义。





## 用实体集还是用联系集

举例：现在有客户和运营商两个实体：客户（客户ID，客户姓名，所在单位）；运营商（运营商ID，公司名称，法人代表），现在，客户A需要向运营商中国移动申请手机号码，客户和运营商产生联系，是否可以重新定义客户呢？

解：（客户ID，客户姓名，所在单位，手机号码，运营商ID）

举例：客户A想再拥有一个电信手机号码，移动的公用，电信的私用，这时客户关系怎么存储2个手机号码呢？

解：（客户ID，客户姓名，所在单位，手机号码1，运营商1ID，手机号码2，运营商2ID）



## 二元还是n元联系集

举例：现在有三个实体：卖家（卖家ID, 卖家名称, 开店时间）；买家（买家ID, 买家名称, 账户余额）；商品（商品ID, 商品名称, 商品类别）。

三个实体通过卖家出售商品给买家建立一个三元联系订单（订单编号, 卖家ID, 买家ID, 商品ID, 数量, 金额），这种三元结构有什么优劣呢？

解：好处是可以清晰地表示几个实体参与到一个联系集中，但是三元联系一致性难以维护。如果订单表中有100万条数据，其中商品ID = 1001的记录有1万条，如果有一天，需要将1001更新成1002，可能会出现不一致的结果。

**数据库中联系通常都是二元的。**一些看来非二元的联系实际上可以用多个二元关系更好地表示。例如上面的联系可以拆分为（订单编号, 卖家ID, 买家ID）和（订单编号, 买家ID, 商品ID, 数量, 金额）。

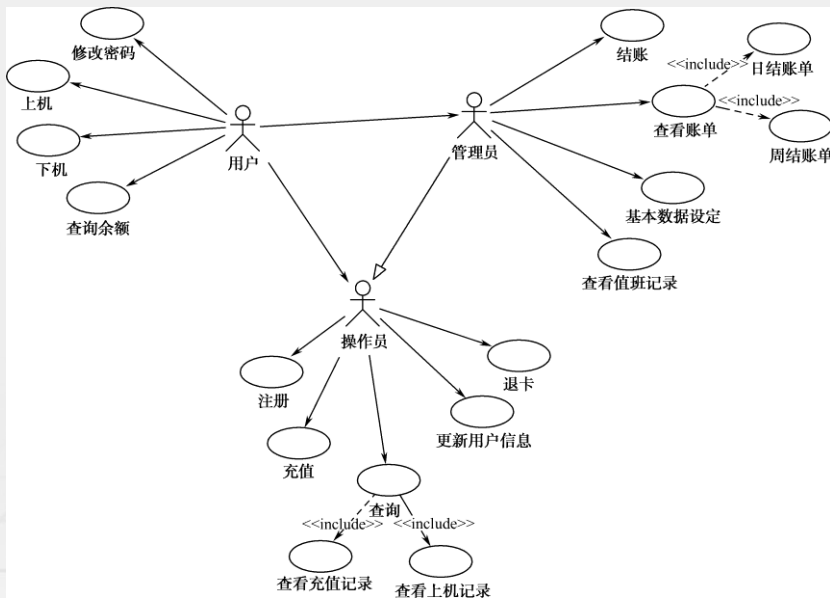


- 统一建模语言UML
- 统一建模语言 (UML Unified Modeling Language) 是非专利的第三代建模和规约语言。UML是一种开放的方法。用于说明、可视化、构建和编写一个正在开发的、面向对象的、软件密集系统的制品的开放方法。
- 在UML系统开发中有三个主要的模型：
  - 功能模型：从用户的角度展示系统的功能，包括用例图。
  - 对象模型：采用对象、属性、操作、关联等概念展示系统的结构和基础，包括类别图、对象图。
  - 动态模型：展现系统的内部行为。包括序列图、活动图、状态图。



## ● 用例图

用例图描述了系统提供的一个功能单元。用例图一般表示出用例的组织关系，用例角色用椭圆形或人型表示，角色和用例之间的关系使用简单的线段来描述。

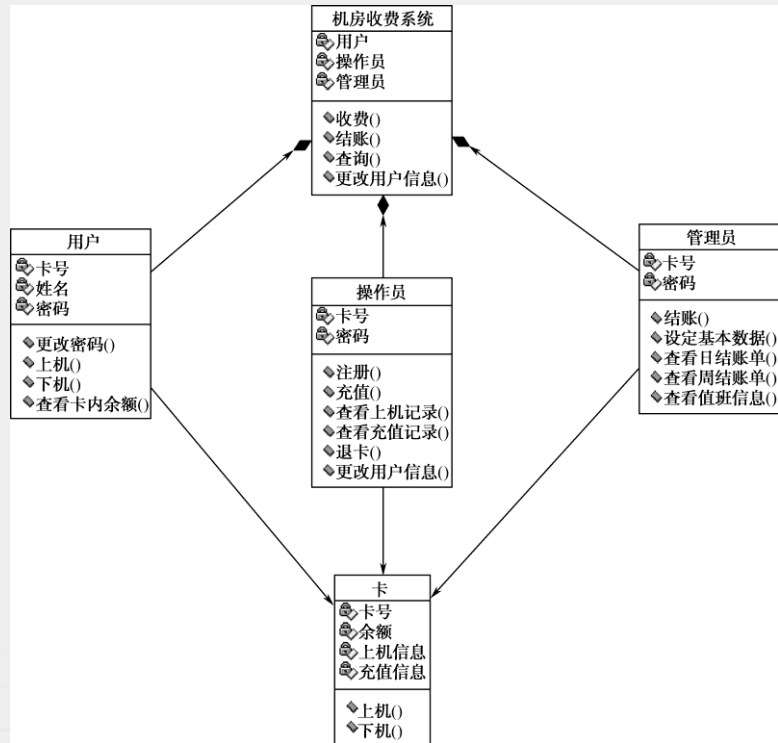


用例图示例



### ● 类图

类图表示不同的实体（人、事物和数据）如何彼此相关。类在类图上使用包含三个部分的矩形来描述，如下图所示。最上面的部分显示类的名称，中间部分包含类的属性，最下面的部分包含类的操作（或者说“方法”）。

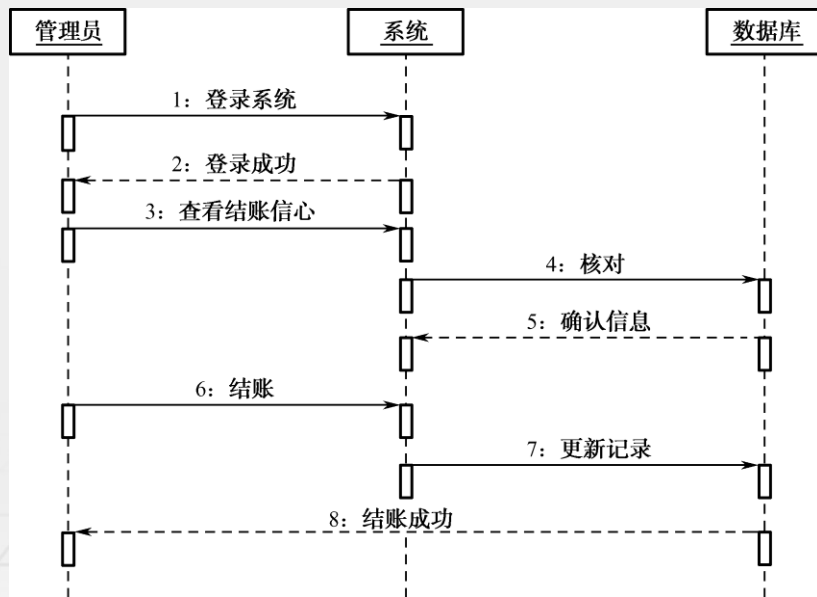


类图示例



## ● 序列图

序列图显示具体用例(或者是用例的一部分)的详细流程。它几乎是自描述的,并且显示了流程中不同对象之间的调用关系,同时还可以很详细地显示对不同对象的不同调用。

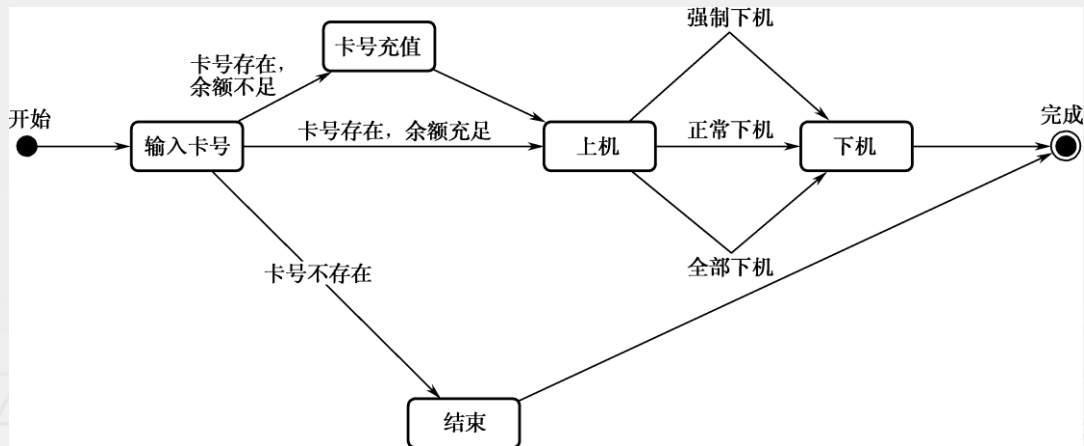


序列图示例



### ● 状态图

状态图表示某个类所处的不同状态和该类的状态转换信息。如下图所示，状态图的符号集包括5个基本元素：初始起点，它使用实心圆来绘制；状态之间的转换，它使用具有开箭头的线段来绘制；状态，它使用圆角矩形来绘制；判断点，它使用空心圆来绘制；以及一个或者多个终止点，它们使用内部包含实心圆的圆来绘制。绘制状态图，首先绘制起点和一条指向该类的初始状态的转换线段。然后只需使用状态转换线条将它们连接起来。

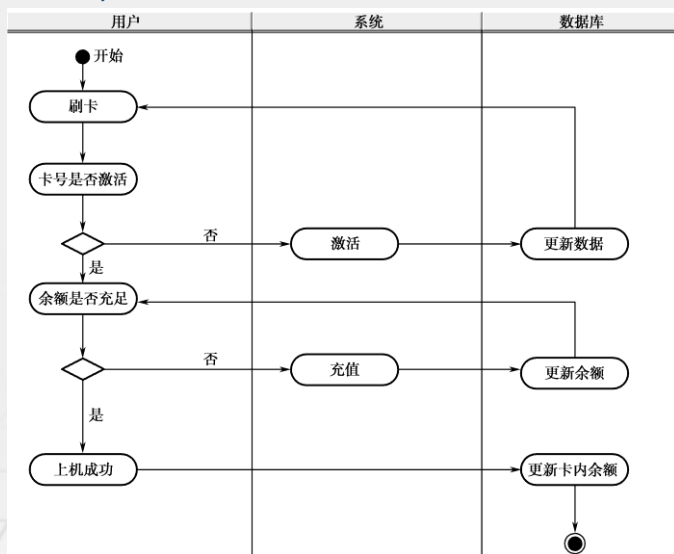


状态图示例



## ● 活动图

活动图表示在处理某个活动时，两个或者更多类对象之间的过程控制流。活动图是从一个连接到初始活动的实心圆开始。活动是通过一个圆角矩形来表示的。活动可以通过转换线段连接到其他活动，或者连接到判断。结束过程的活动连接到一个终止点。如下图所示。



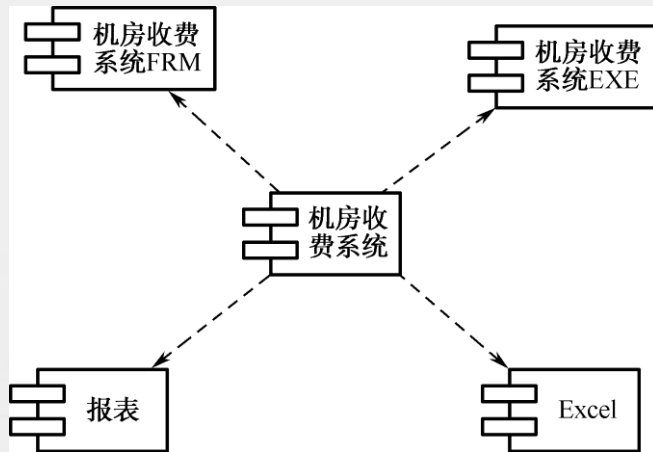
活动图示例





### ● 组件图

组件图提供系统的物理视图。它的用途是显示系统中的软件对其他软件组件(例如,库函数)的依赖关系。组件图可以在一个非常高的层次上显示,从而仅显示粗粒度的组件,也可以在组件包层次上显示。

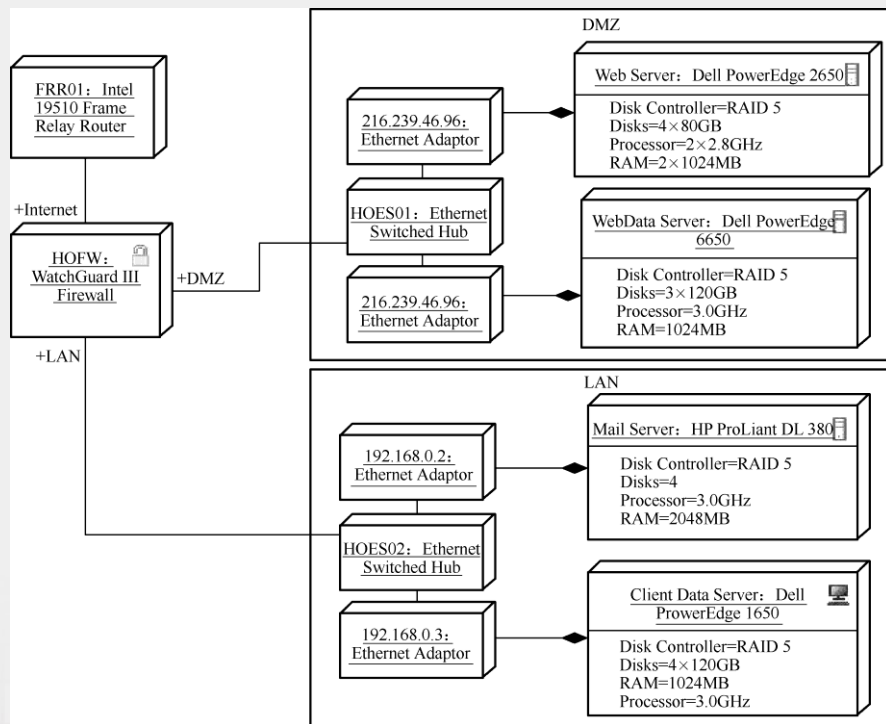


组件图示例



### ● 部署图

部署图表示该软件系统如何部署到硬件环境中。它的用途是显示该系统不同的组件将在何处物理地运行，以及它们将如何彼此通信。



部署图示例



### ● IDEF1X

- IDEF的含义是集成计算机辅助制造(ICAM)DEFinition。
- 根据用途，可以把IDEF族方法分成两类：
  - 第一类IDEF方法的作用是沟通系统集成人员之间的信息交流。主要有：IDEF0、IDEF1、IDEF3、IDEF5。
  - 第二类IDEF方法的重点是系统开发过程中的设计部分。目前有两种IDEF设计方法：IDEF1X和IDEF4。



### ● IDEF1X

IDEF1X 模型的基本结构和E-R模型基本类似，主要有以下元素：

- 实体(如人，地点，概念，事件等)用矩形方框表示；
- 实体之间的关系(联系)，用方框之间的连线表示；
- 实体的属性，用方框内的属性名称来表示。



1、实体是客观世界中描述客观事物的概念，是一个数据对象。实体集就是具有相同属性（或性质）的实体集合。实体集可以分为强实体集和弱实体集。

### 2、属性的分类：

简单属性、复合属性、单值属性、多值属性、派生属性、空值属性。

### 3、联系表示一个或多个实体之间的关联关系。可分为两类：

实体内部的联系、实体之间的联系。

### 4、映射基数指的是一个实体通过一个联系能同时与多少个实体相关联，可分为：

一对一、一对多、多对一和多对多。

### 5、E-R图的设计时，需要了解三个基本概念：

- 实体用方框表示，方框内注明实体的名称；
- 属性用椭圆形框表示，并用无向边将属性与对应的实体连接起来；
- 联系用菱形框表示，并用无向边将其与相关的实体连接起来。

### 6、E-R图的设计分为：

局部设计、全局设计、消除冲突。