College of Computer Science, Chongqing University

Software Engineering

A Practitioner's Approach Seventh Edition

需求建模

zmqmail@cqu.edu.cn 13708390417

重庆大学 计算机学院 COLLEGE OF COMPUTER SCIENCE

需求评审会

召开需求评审会的注意事项

会前准备

- 1、尽可能充分的理解需求
  - 一般在需求评审会之前，产品经理会提前将需求文档发送到各个岗位手里
- 2、整理出自己需要的需求功能列表
  - 通过XMind、脑图等思维导图工具进行分类整理，形成比较直观的功能列表。
- 3、整理成多维度的需求
  - 对功能列表进一步的加工，打破功能纵向上的联系，完成功能横向上的关联。

开发一个点餐系统

步骤1 尽可能充分的理解需求

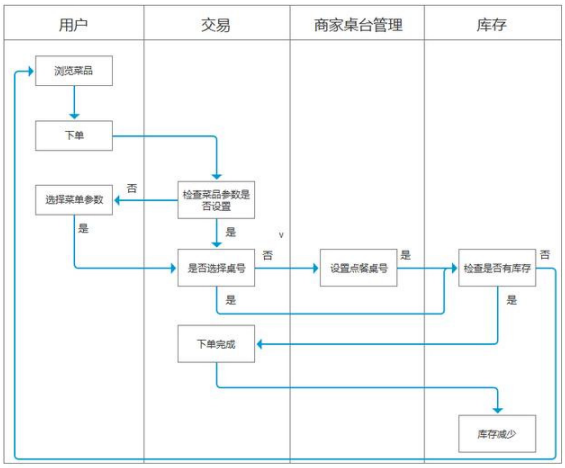
- 充分的去了解需求的背景等相关信息。
- 比如：
  - 餐厅为什么需要打破原来的点餐规则，因为工作效率？
  - 互联网+的噱头？
  - 还是有其他不可抗力的因素？

  - **需求背景**就会**影响**到整个系统的交互**设计风格:**
  - 如果是因为效率，那么你的交互更应该注重效率，把步骤尽量简化
  - 如果是因为蹭互联网热度找一些噱头，那么你可以设计更加酷炫的交互方案，加入一些动效、过场动画等。

步骤2 整理出自己需要的需求功能列表

- 借助一些思维导图工具来帮助自己整理。
  - 根据需求文档对点餐系统的功能进行了整理
  - 整理完后，可以让你页面布局的时候，能够更加直观，不会漏掉该有的功能点。

步骤3 整理成多维度的需求



- 把需求进行多维度的整合
  - 将所有功能点用业务逻辑图进行整理和关联
  - 例如：第二步骤其实还漏掉了一部分内容，那就是系统是否有库存概念，如果没有库存了要怎么办?

会议上

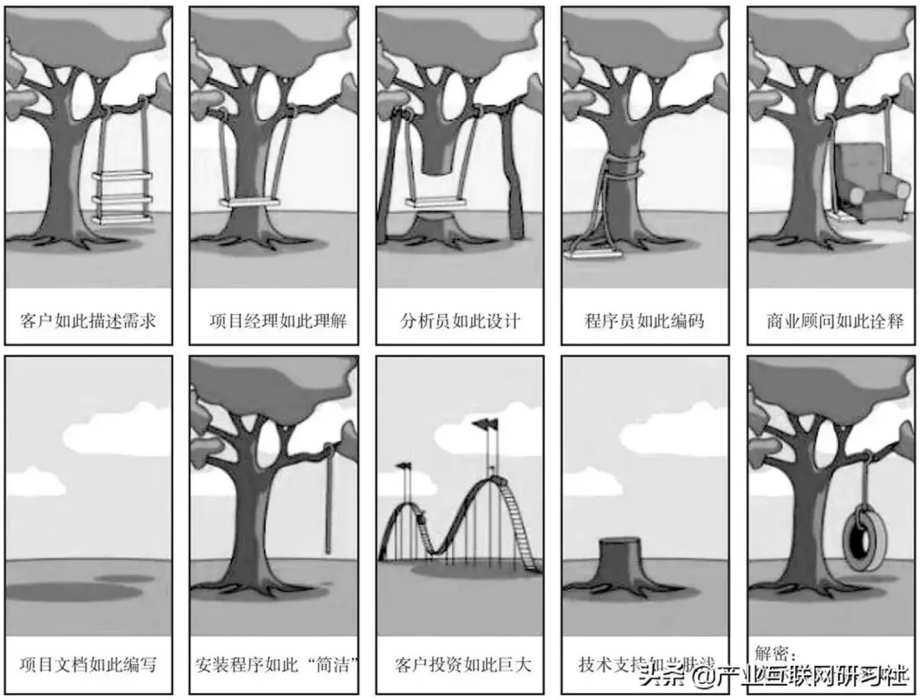
- 1、不要纠结于某个细节点：
  - 会议人员多的时候，容易陷入大家集体纠结于某个细节点
- 2、允许一定程度的发散：
  - 出于业务的差异性，如果讨论的过程中，突然某个人想到了之前的某个业务流的问题并发起了谈论，那么暂时不要阻止，允许他有思考的时间，当节点到了之后，那么拉回来，继续我们刚才没有继续完成的问题，继续讨论。
- 3、切忌陷入开发讨论
  - 绝对还没有涉及到具体开发细节上，这个时候陷入的开发讨论很多时候都仅是开发人员出于下意识的跟产品经理抱怨开发难度过高，时间周期可能会很长。
  - 切忌陷入开发细节的讨论。

评审会后

- 评审会后要预留出一定量的时间对在会上新接收到的新的**需求理解进行消化了整理**。
  - 在充分理解需求之后才开始着手设计。
  - 在设计过程中，如果有遇到一些模棱两可的问题，一定要及时向相关人员再次明确。
  - 沟通很重要！

多多沟通!

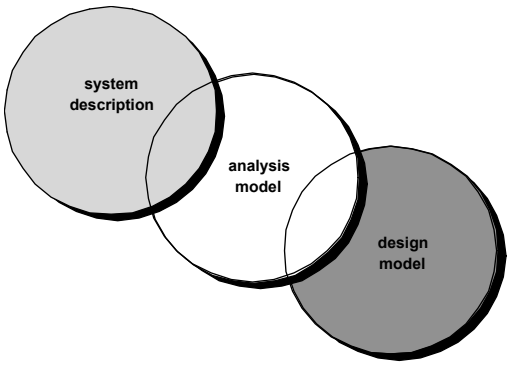
沟通很重要。



需求模型在系统描述和设计模型之间建立桥梁

Bridge

系统描述和设计模型的桥梁



懂-》表达?

变成程序员能够“准确”理解的语言或形式

不直观!

很多基于文本的需求建模情景，甚至和用例一样简单，不能简明扼要地传递信息.

文字》符号

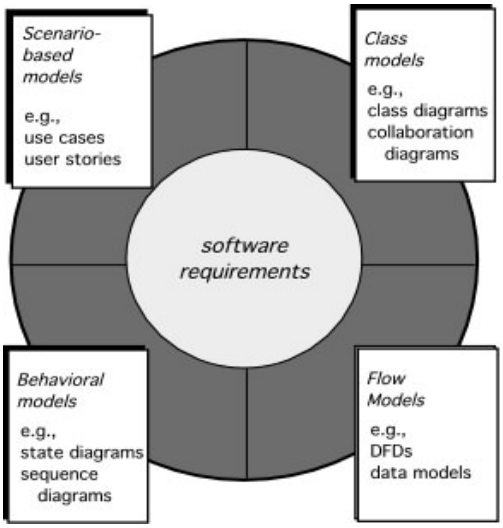
表达形式的变化

Elements of Requirements Analysis

需求模型的元素

基于场景的模型

用例
用户故事



类模型

类图
协作图

行为模型

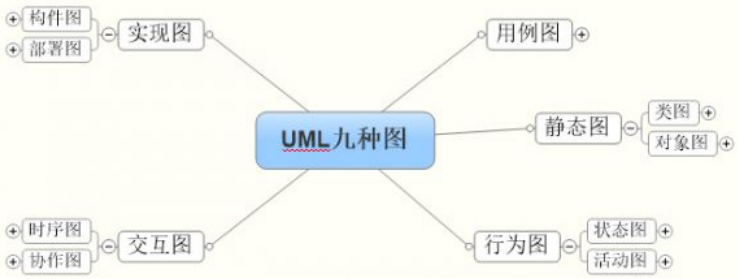
状态图
顺序图

流模型

数据流图
数据模型

UML

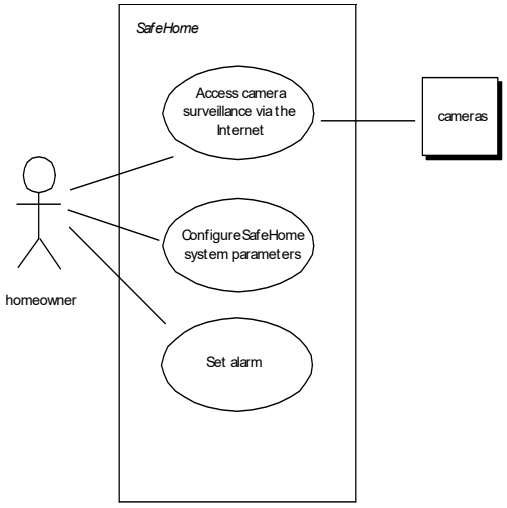
UML建模语言



用例图 (Use-Case Diagram)

Use-Case Diagram

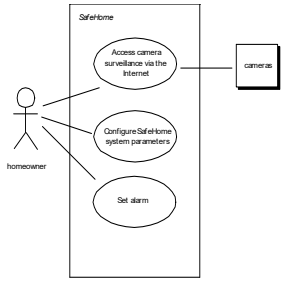
用例图，每一个用例用一个椭圆表示。



SafeHome系统初步用例图

初始用例 → 不够用? → **编写正式用例**

用非正式的描述性风格编写
(常常够用的)



具有大量异常处理的负责步骤

要点:

1 情景目标：确定了用例的范围

2 前提条件

3 触发器：确定“用例开始”的事件或条件

4 场景

5 异常处理

每种建模注释方法都有局限

用例关注功能和行为需求，一般不适用于非功能性需求。

补充用例的UML模型

不能简明扼要地传递信息——补充大量的UML图形模型

开发活动图 (Activity Diagram)

(类似于流程图)

两端为半圆形的矩形表示一个特定的系统功能

判定菱形表示判定分支(标记从菱形发出的每个箭头)

箭头表示通过系统的流

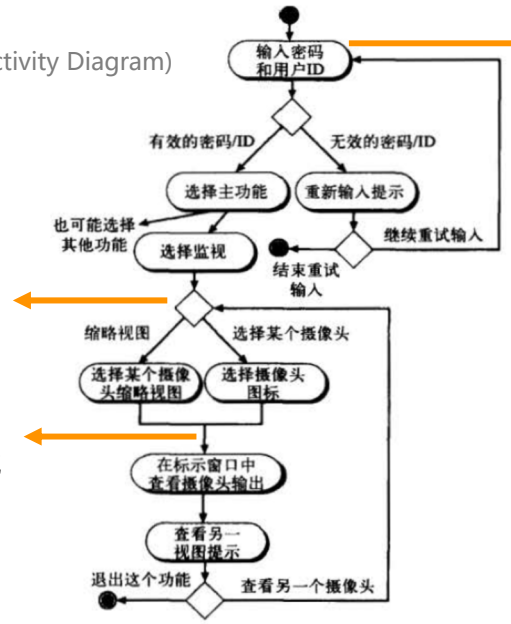
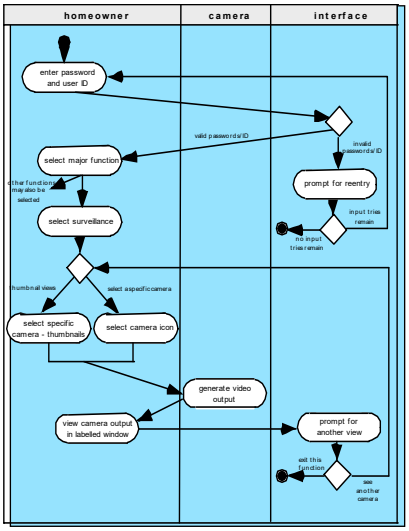


图6-5 通过互联网进入摄像机监视并显示摄像机视图功能的活动图

泳道图 (Swimlane Diagrams)

Swim lane Diagrams

泳道图：**活动图的一种有用变形**。

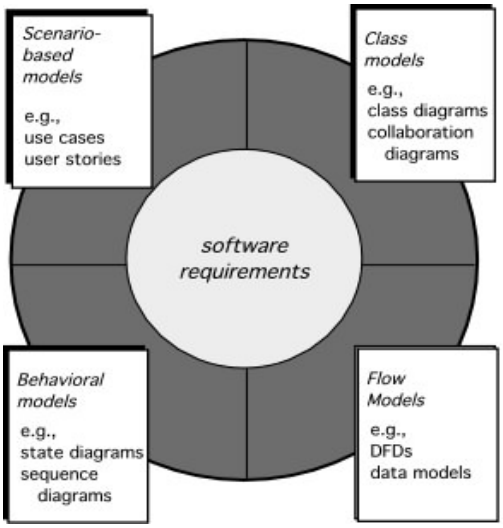


Elements of Requirements Analysis

需求模型的元素

基于场景的模型

用例
用户故事



类模型

类图
协作图

行为模型

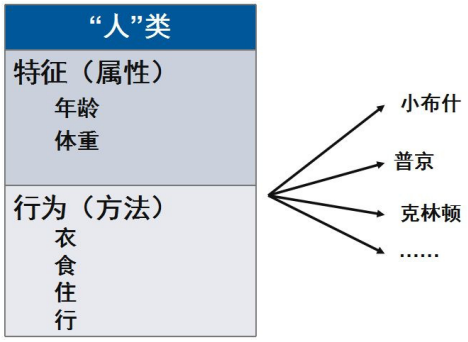
状态图
顺序图

流模型

数据流图
数据模型

基于类的建模基本思路

- 1 识别分析类 名词?
  - 分析出有哪些类
- 2 描述属性
  - 对应将来编写代码定义类的属性
- 3 定义操作 动词?
  - 对应将来编代码定义方法
- 4 CRC建模
  - 解决类的功能是什么，协作关系如何
- 5 关联与依赖
  - 解决类之间有什么关系



类或对象的分类问题

- 实体类
  - 模型或者业务类，从问题说明中直接提取。例如，Sensor等。
- 边界类
  - 创建用户可见的和在使用软件时交互的接口。如交互屏幕或打印的报表。
- 控制类：通常，直到设计开始才开始考虑控制类。
  - 自始至终管理“工作单元”
    - 1 实体类的创建或更新；
    - 2 边界类的实例化；
    - 3 对象集合之间的通信；
    - 4 对象间和用户和应用系统间交互数据的确认。

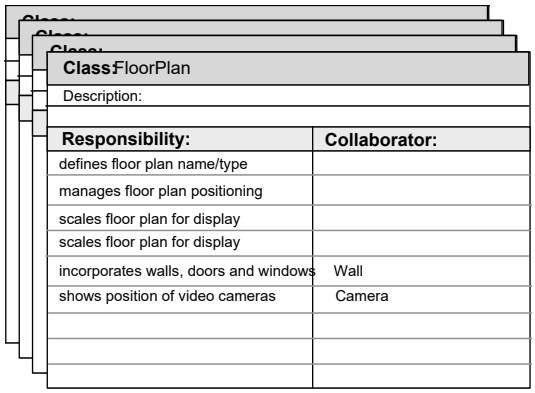
Class-Responsibility-Collaborator

Entity classes
实体类
Boundary classes
边界类
Controller classes
控制类

Attributes and Operations
属性和操作

Fulfill their responsibilities
实现

CRC Modeling 建模结果就是一堆卡片?



职责(属性+操作)：5个指导原则

- 1 智能系统因分布在所有类中以求最大程度地满足问题的需求。
  - 本质：高内聚，避免超类
- 2 每个职责的说明应尽可能具有普遍性。
- 3 信息和与之相关的行为应该放在同一类中。
  - 封装
- 4 某个事物的信息应局限于一个类中而不要分布在多个类中。
- 5 适合时，职责应由相关类共享。
  - 抽象，接口实现。

设计模式中详细分析

3种不同的通用关系

- 1 is-part-of(是……一部分)
- 2 has-knowledge-of(有……的知识)
- 3 depends-upon(依赖关系)

UML 不同！！！

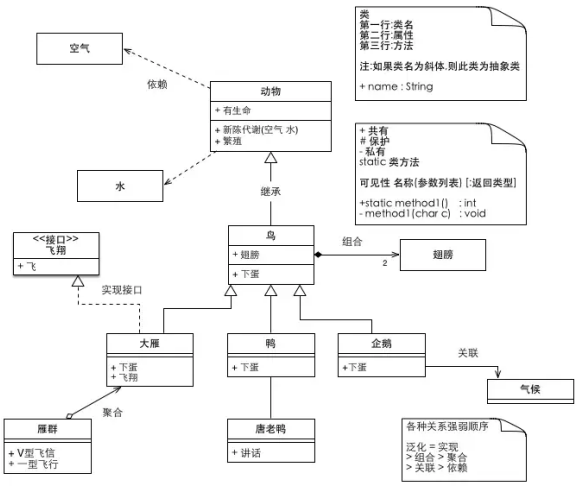
结果 → **渊源**

依赖 — 指向依赖对象

泛化(继承)
细化 — 指向父类(接口)

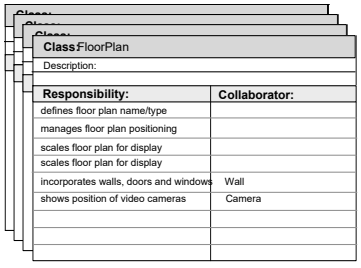
组合
不可分割
聚合
可分割 — 指向细节(被)

关联



CRC

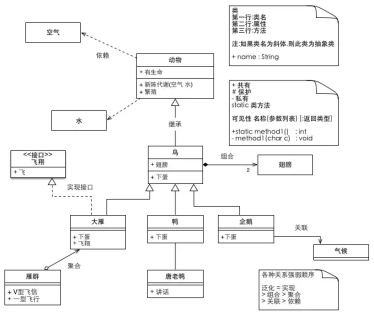
结果



Class: FloorPlan	
Description:	
Responsibility:	**Collaborator:**
defines floor plan name/type	
manages floor plan positioning	
scales floor plan for display	
scales floor plan for display	
incorporates walls, doors and windows	Wall
shows position of video cameras	Camera

UML

结果

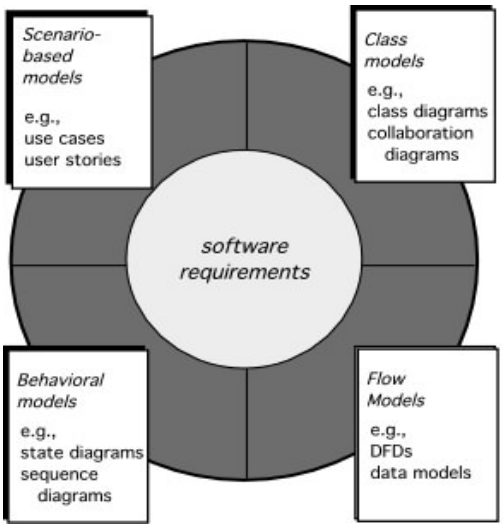


Elements of Requirements Analysis

需求模型的元素

基于场景的模型

用例
用户故事



类模型

类图
协作图

行为模型

状态图
顺序图

流模型

数据流图
数据模型

Data Objects

数据对象

- **Entity** (e.g., Anything that produces or consumes information)
- **Thing** (e.g., A report or a display)
- **Occurrence** (e.g., A telephone call) or **event** (e.g., An alarm)
- **Role** (e.g., Salesperson)
- **Organizational unit** (e.g., Accounting department),
- **Place** (e.g., A warehouse)
- **Structure** (e.g., A file).

E-R图

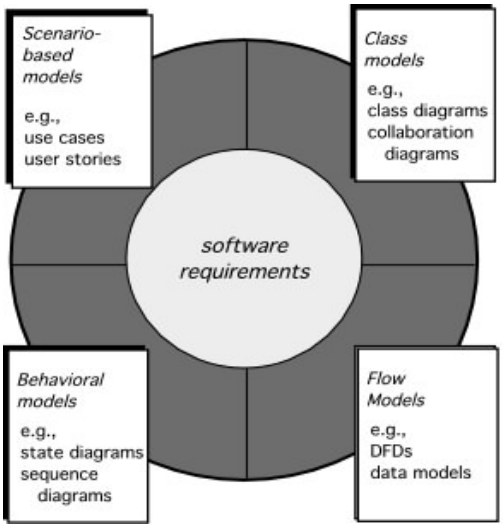
数据库

Elements of Requirements Analysis

需求模型的元素

基于场景的模型

用例
用户故事



类模型

类图
协作图

行为模型

状态图
顺序图

流模型

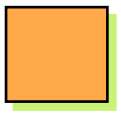
数据流图
数据模型

DFD

Data Flow Diagram
数据流图

Flow Modeling Notation

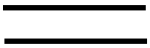
数据流图 符号



external entity
外部实体

data flow
数据流

process
处理过程

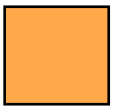


data store
数据存储

External Entity

外部实体

Examples: *a person, a device, a sensor computer-based system*



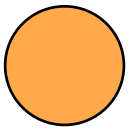
A producer or consumer of data

Data must always originate somewhere and must always be sent to something

Process

过程(软件或系统)

Examples: *compute taxes, determine area, format report, display graph*



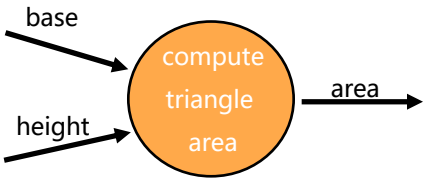
A data transformer (changes input to output)

Data must always be processed in some way to achieve system function

Data Flow

数据或控制对象

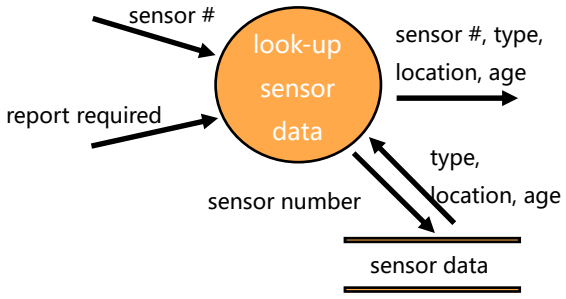
Data flows through a system, beginning as input and transformed into output.



Data Stores

数据存储

Data is often stored for later use.



数据流图 指南

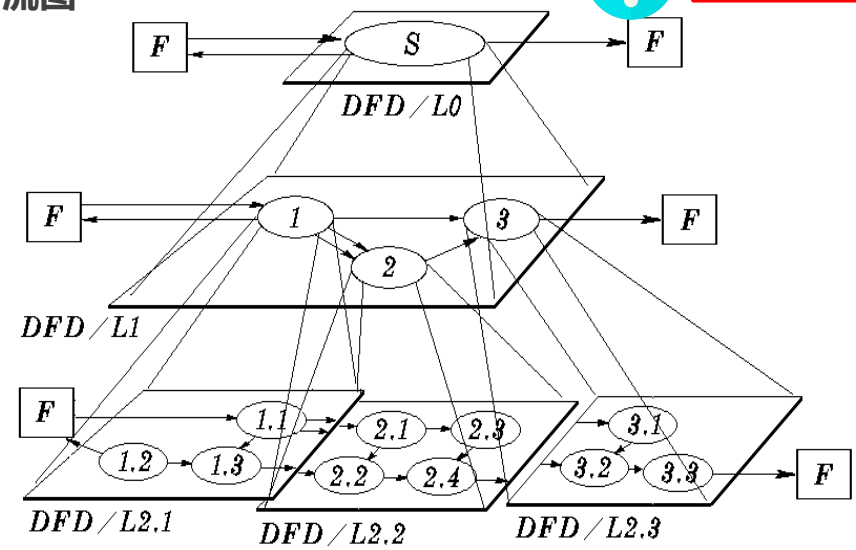
Guidelines

指南

- **有意义的名字**
  - All icons must be labeled with meaningful names
- **分层**
  - The DFD evolves through a number of levels of detail
  - Always begin with a context level diagram (also called level 0)
  - Always show external entities at level 0
- **标注箭头**
  - Always label data flow arrows
- **不展示处理逻辑**
  - Do not represent procedural logic

分层的数据流图

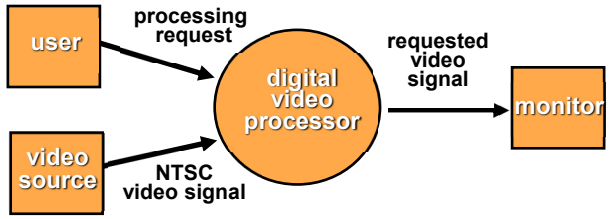
外部实体*L0*层中必须出现



构建DFD—I

- Create a level 0 DFD
  - **Operations**
    - Review user scenarios and/or the data model to isolate data objects and use a grammatical parse to determine operations
  - **External entities**
    - Determine external entities (producers and consumers of data)

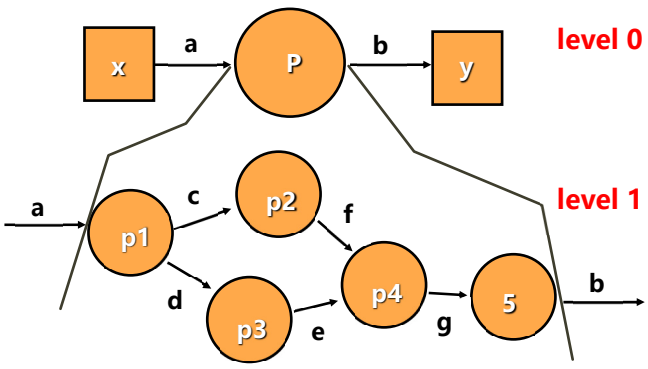
Level 0 DFD Example



构建DFD—II

- Develop a level 1 DFD
  - Write a narrative describing the transform
  - Parse to determine next level transforms
  - "Balance" the flow to maintain data flow continuity
- Use a **1:5** (approx.) expansion ratio

The Data Flow Hierarchy



考务处理系统的功能

实例考务处理系统的功能

- 1) 对考生送来的报名单进行检查;
- 2) 对合格的报名单编好准考证号后将准考证送给考生，并将汇总后的考生名单送给阅卷站;
- 3) 对阅卷站送来的成绩单进行检查，并根据考试中心制定的合格标准审定合格者;
- 4) 制作考生通知单 (含成绩及合格/不合格标志) 送给考生;
- 5) 按地区进行成绩分类统计和试题难度分析，产生统计分析表。

实例考务处理系统的功能

外部实体*L0*层中必须出现

- 1) 对**考生**送来的报名单进行检查;
- 2) 对合格的报名单编好准考证号后将准考证送给**考生**，并将汇总后的**考生**名单送给**阅卷站**;
- 3) 对**阅卷站**送来的成绩单进行检查，并根据**考试中心**制定的合格标准审定合格者;
- 4) 制作**考生**通知单 (含成绩及合格/不合格标志) 送给**考生**;
- 5) 按地区进行成绩分类统计和试题难度分析，产生统计分析

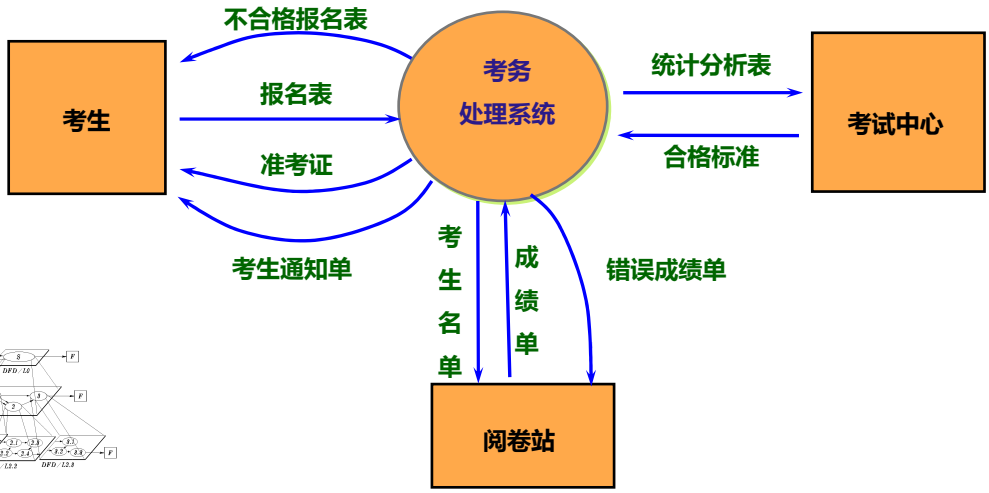
分析实体!

实例考务处理系统的功能

- 1) 对考生送来的**报名单**进行检查;
- 2) 对合格的报名单编好准考证号后将准考证送给考生，并将汇总后的**考生名单**送给阅卷站;
- 3) 对阅卷站送来的**成绩单**进行检查，并根据考试中心制定的合格标准审定合格者;
- 4) 制作考生**通知单** (含成绩及合格/不合格标志) 送给考生;
- 5) 按地区进行**成绩分类统计**和试题难度分析，产生**统计分析**

分析数据流!

顶层数据流图



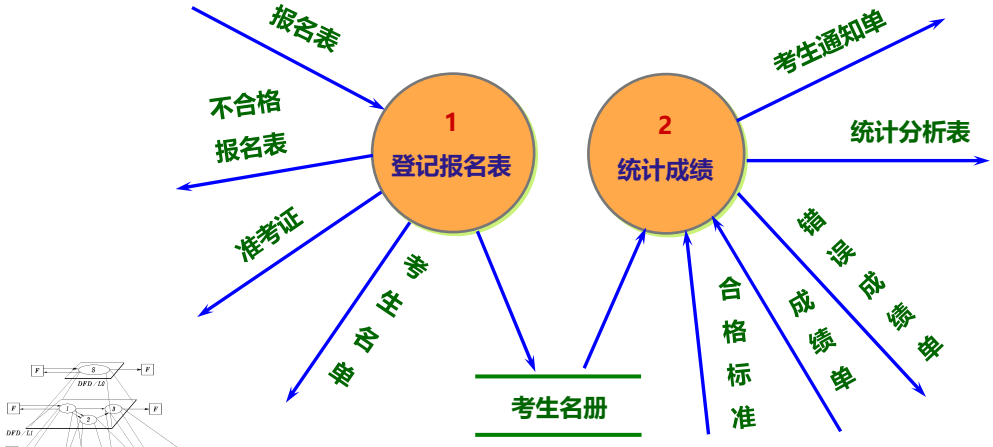
业务分析

逐步细化!

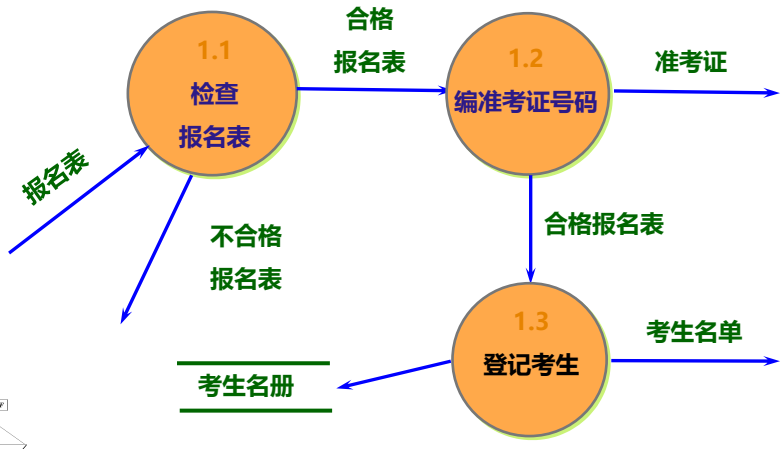
- 根据考务处理业务，画出顶层数据流图，以**反映最主要业务处理流程及系统与外界的关系。**

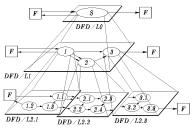
- **考务业务处理**的主要功能应当有**登记报名单**、**统计成绩**两个主要**数据流**。
  - 输入的源点和输出终点是考生、考试中心和阅卷站。
  - 然后从输入端开始，根据考务业务工作流，画出数据流流经的各加工框，逐步画到输出端，得到第 0 层数据流图。

第0层数据流图

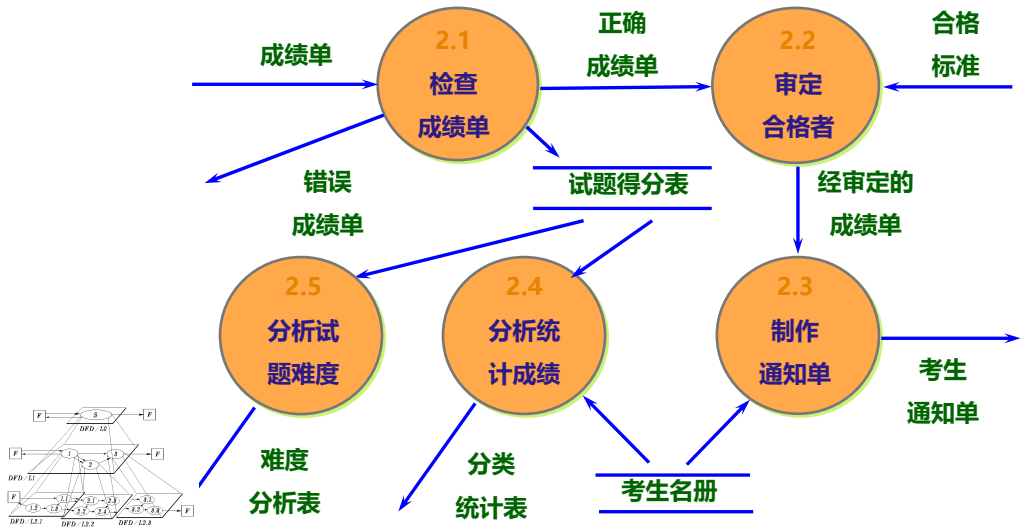


第1层数据流图 (a)





第1层数据流图 (b)



数据流图注意事项

- 一个“泡泡”只做一件事情
  - Each bubble is refined until it does just one thing
- 3-7层
  - Most systems require between 3 and 7 levels for an adequate flow model
- 膨胀率随着层数的增加而减小
  - The expansion ratio decreases as the number of levels increase
- 单个数据流项（箭头）可以随着级别的增加而扩展（数据字典提供信息）
  - A single data flow item (arrow) may be expanded as levels increase (data dictionary provides information)

控制流

Control Flow Modeling

事件或者控制项可以实现为布尔值（例如真假）或者条件的离散列表（空、拥挤、满）

11.3 控制流建模(Control Flow Modeling)

- Suggested **Guidelines**(I):
  - Listing all sensors that are "read" by the software.
    - 列出所有被软件“读”的传感器
  - Listing all interrupt conditions.
    - 列出所有的中断条件
  - Listing all "switches" that are actuated by an operator.
    - 列出操作人员能够启动的所有开关
  - Listing all data conditions.
    - 列出所有的数据条件

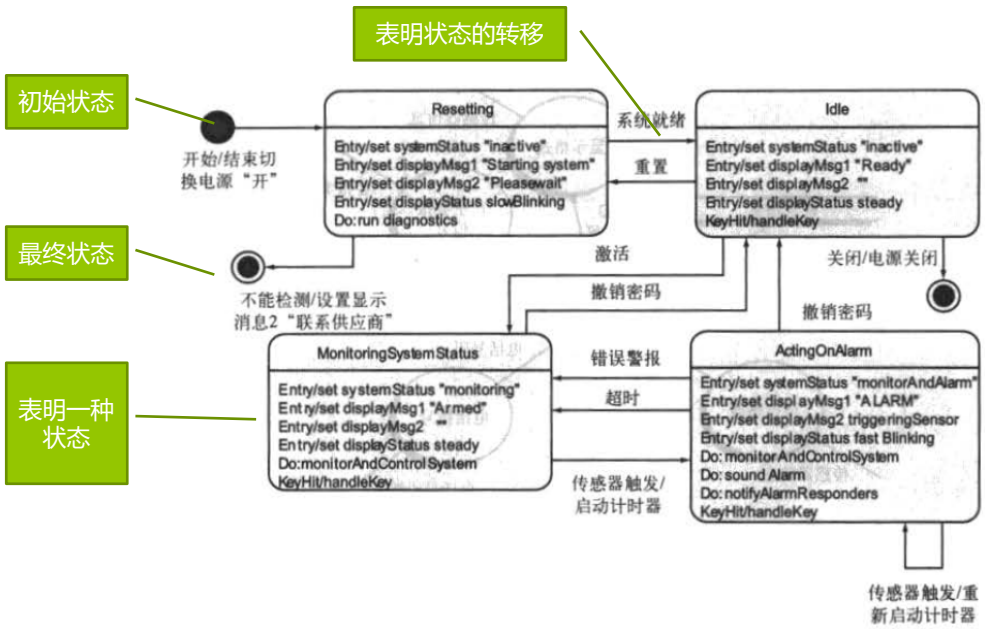


图7-4 SafeHome安全功能的状态图

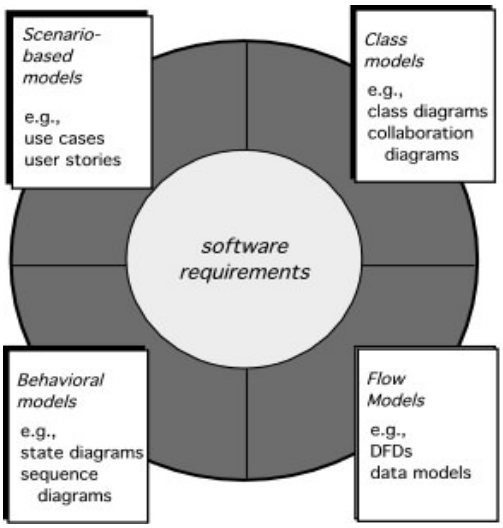
input events						
sensor event	0	0	0	0	1	0
blink flag	0	0	1	1	0	0
start stop switch	0	1	0	0	0	0
display action status complete	0	0	0	1	0	0
in-progress	0	0	1	0	0	0
time out	0	0	0	0	0	1
output						
alarm signal	0	0	0	0	1	0
process activation						
monitor and control system	0	1	0	0	1	1
activate/deactivate system	0	1	0	0	0	0
display messages and status	1	0	1	1	1	1
interact with user	1	0	0	1	0	1

Process activation table for SafeHome security function

处理激活表(Process Activation Table, PAT)

Elements of Requirements Analysis
需求模型的元素

基于场景的模型
用例
用户故事



类模型
类图
协作图

行为模型
状态图
顺序图

流模型
数据流图
数据模型

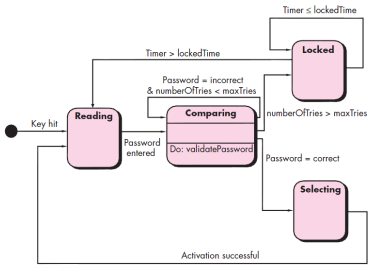
行为建模 (Behavioral Modeling)

- **行为建模基本方法**
- **Make a list** of the different states of a system (How does the system behave?)
- Indicate how the system makes a transition from one state to another (How does the system change state?)
  - indicate **event**
  - indicate **action**
- draw a state diagram or a sequence diagram

11.4 行为建模（Behavioral Modeling）

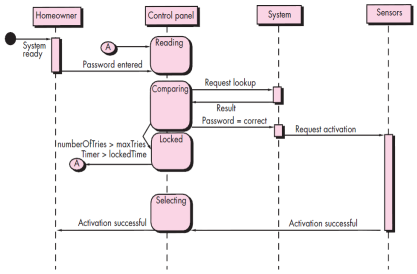
State Diagram

状态图



Sequence Diagram

顺序图/时序图

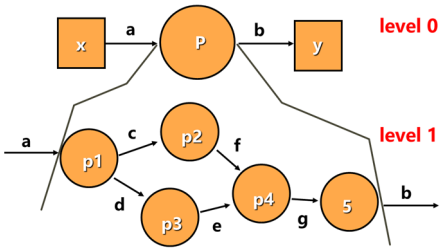


UML图(Diagrams)

- UML语言**5种**类型，**9种**不同的**图**，把它们有机的结合起来就可以描述系统的所有视图。
  - **用例图**(Use case diagram) 从用户角度描述系统功能,并指出各功能的操作者。
  - **静态图**(Static diagram),表示系统的静态结构。
    - 包括**类图**、**对象图**、**包图**。
  - **行为图**(Behavior diagram)，描述系统的动态模型和组成对象间的交互关系。
    - 包括**状态图**、**活动图(—>泳道图)**。
  - **交互图**(Interactive diagram), 描述对象间的交互关系。
    - 包括**顺序图(时序图)**、**合作图**。
  - **实现图**(Implementation diagram) 用于描述系统的物理实现。
    - 包括**构件图**、**部件图**。

Data Flow Diagram

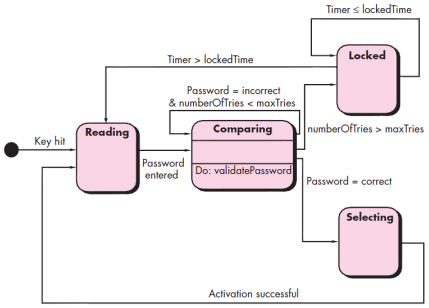
DFD/数据流图



只有三(四)种元素

State Diagram

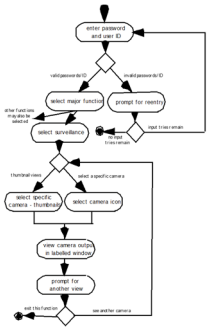
状态图



状态图重点在于描述对象的状态及其状态之间的转移

Activity Diagram

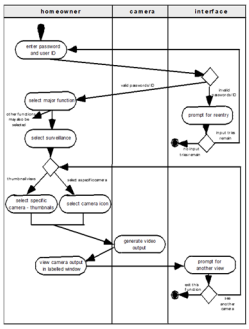
活动图：通过提供迭代流的图形化，类似流程图



活动图描述的是对象(多个)活动的顺序关系

Swim lane Diagrams

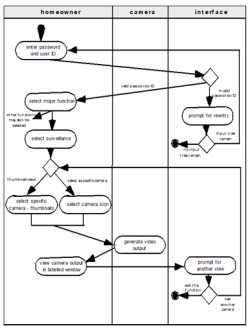
泳道图：活动图的一种有用变形



体现操作者的相互关系，没有时序要求

Swim lane Diagrams

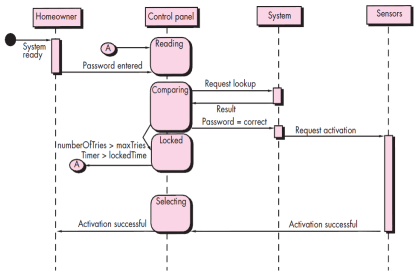
泳道图：活动图的一种有用变形



状态在泳道中间
体现操作者的相互关系，没有时序要求

Sequence Diagram

顺序图/时序图



状态在生命线上
体现时序的前后关系，（UML交互图）

去实践吧

人最宝贵的是生命，这生命属于每个人只有一次。

人的一生应当这样度过：当他回首往事的时候，不因虚度年华而悔恨，也不因碌碌无为而羞愧。

——奥斯托洛夫斯基

College of Computer Science, Chongqing University

Software Engineering

A Practitioner's Approach Seventh Edition

需求建模

zmqmail@cqu.edu.cn 13708390417

重庆大学 计算机学院 COLLEGE OF COMPUTER SCIENCE