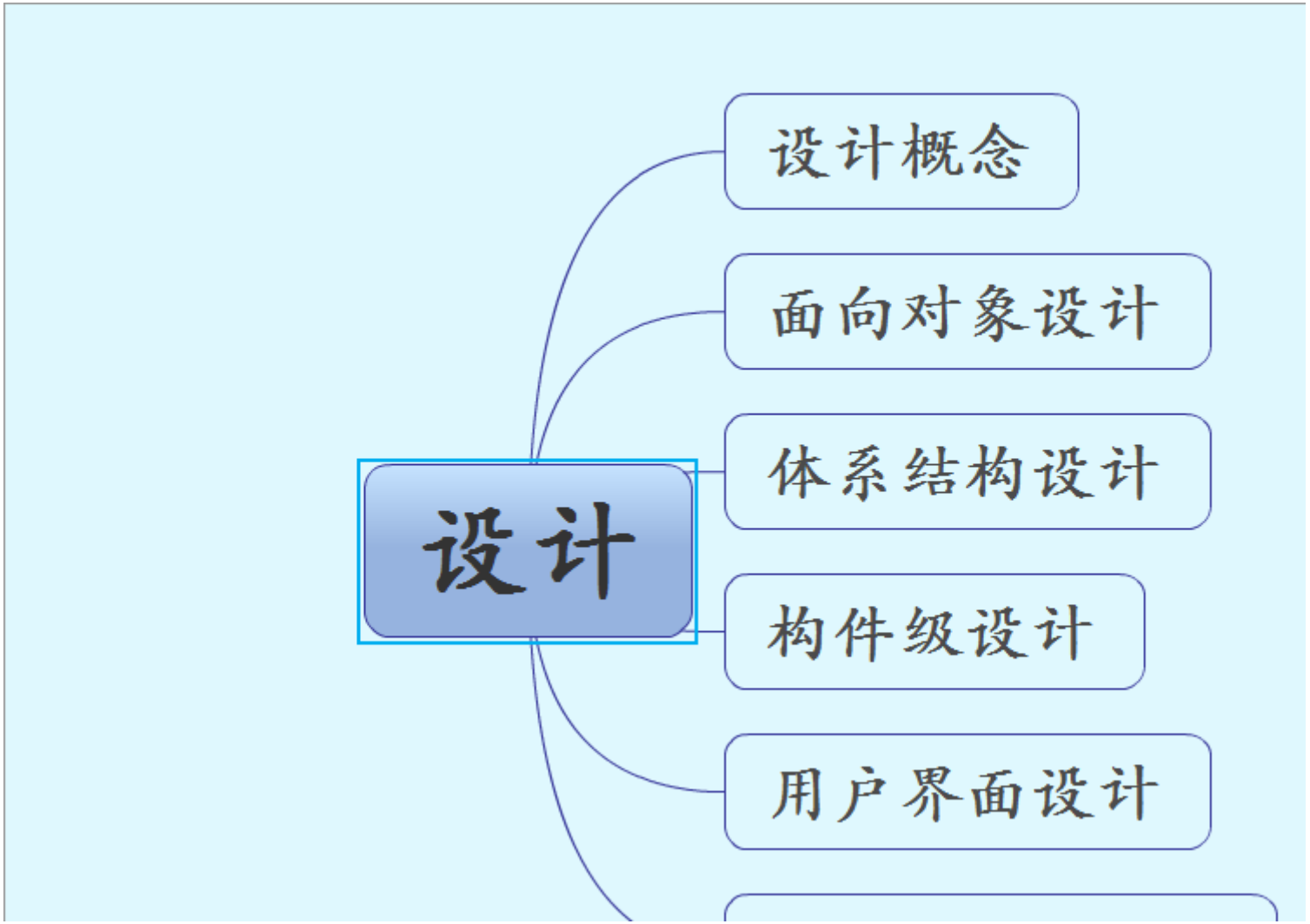
# outline



# 概述

- 从OOA到OOD是一个逐渐扩充模型的过程
  - 对分析模型进行精化
  - 增加实现相关的细节
  - 考虑体系结构、构件和接口
- 在实际的软件开发过程中分析和设计的界限是模糊的。分析和设计活动是一个多次反复迭代的过程。

# OOD准则

## 1、 弱耦合

- 对象之间的两类耦合：
  - 交互耦合：消息连接
    - 使交互耦合尽可能松散的准则：减少消息中包含的参数个数，降低参数的复杂程度，减少消息数。
  - 继承耦合：互为基类和派生类
    - 与交互耦合相反，应该提高继承耦合程度。
    - 继承是一般化类与特殊类之间耦合的一种形式。通过继承关系结合起来的基类和派生类，构成了系统中粒度更大的模块。彼此之间应该越紧密越好。

# OOD准则

## 2、强内聚

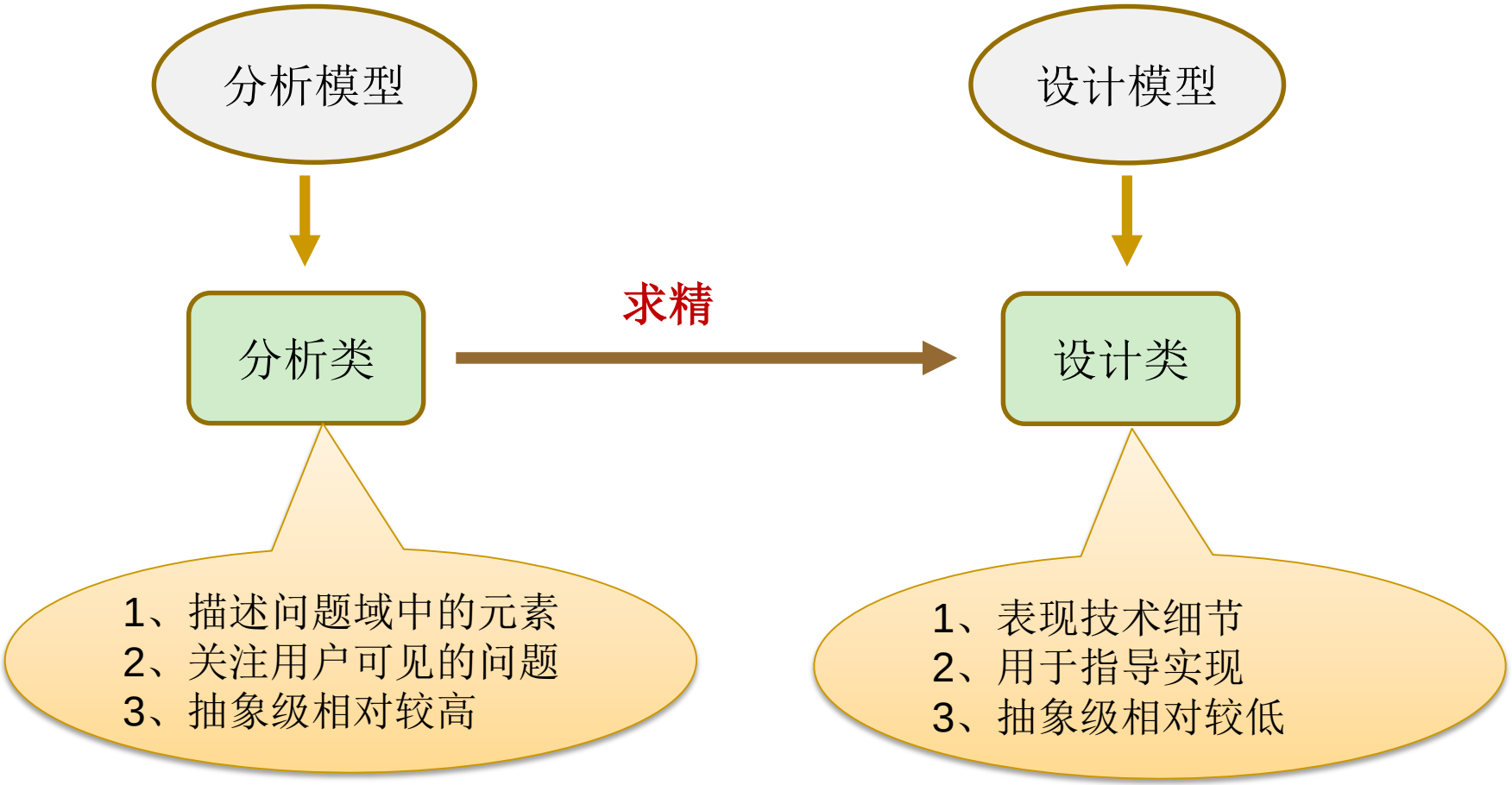
- 在面向对象设计中存在下述3种内聚：
  - 服务内聚。一个服务应该完成一个且仅完成一个功能。
  - 类内聚。一个类应该只有一个用途，它的属性和服务应该是高内聚的。如果某个类有多个用途，通常应该把它分解成多个专用的类。
  - 一般-特殊内聚。设计出的一般-特殊结构，应该符合多数人的概念，是对相应领域知识的正确抽取。

# OOD准则

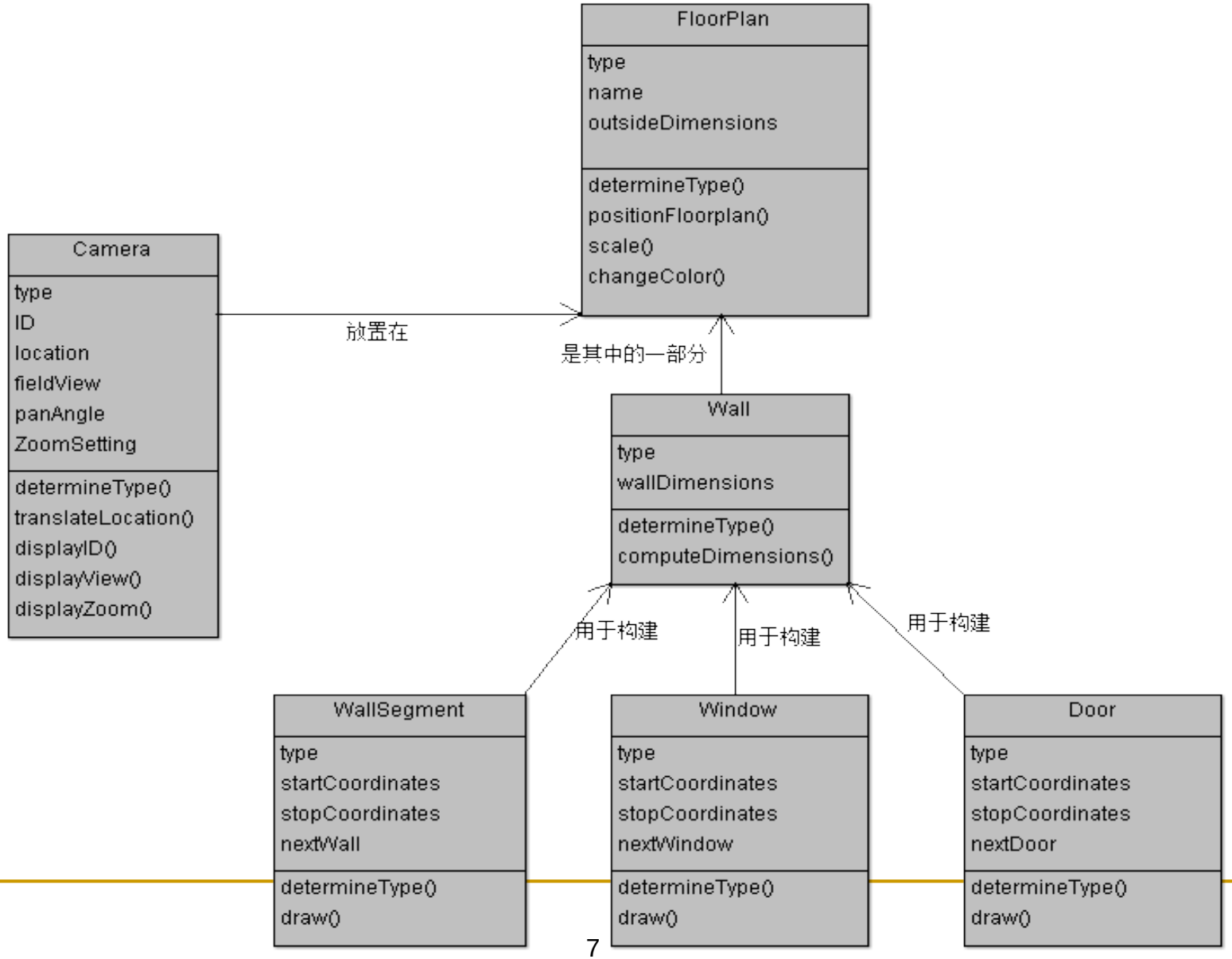
## 3、可重用

- 重用基本上从设计阶段开始
- 重用有两方面的含义：
  - **with reuse**：尽量使用已有的类（包括开发环境提供的类库，及以往开发类似系统时创建的类）。
  - **for reuse**：如果确实需要创建新类，则在设计这些新类的协议时，应该考虑将来的可重复使用性。

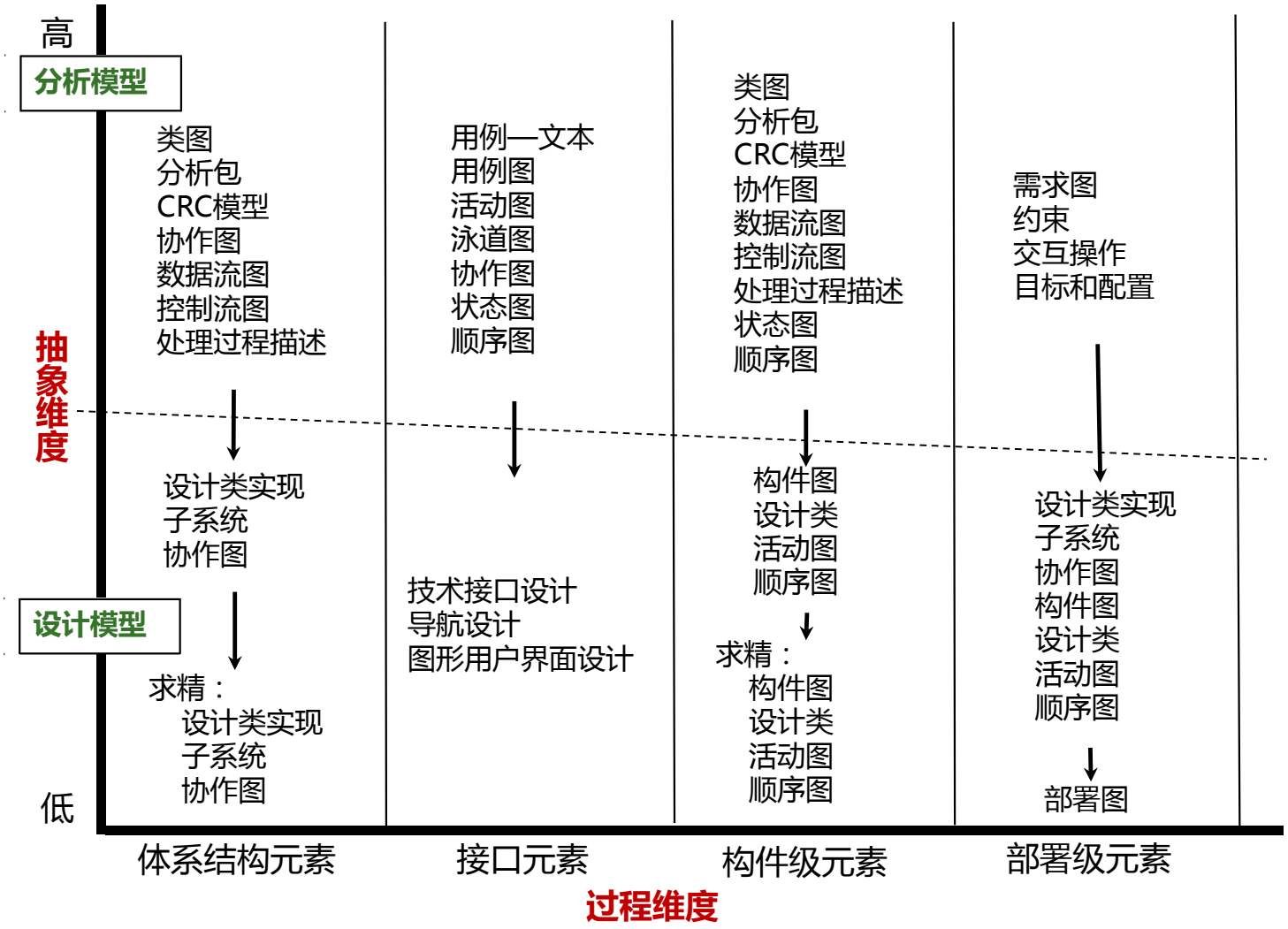
# 设计类


分析模型
设计模型
分析类
求精
设计类
1、描述问题域中的元素
2、关注用户可见的问题
3、抽象级相对较高
1、表现技术细节
2、用于指导实现
3、抽象级相对较低

# ACS-DCV平面设计图分析类



# 设计模型


高
分析模型
类图
分析包
CRC模型
协作图
数据流图
控制流图
处理过程描述
抽象维度
设计类实现
子系统
协作图
设计模型
求精：
设计类实现
子系统
协作图
低
用例—文本
用例图
活动图
泳道图
协作图
状态图
顺序图
技术接口设计
导航设计
图形用户界面设计
类图
分析包
CRC模型
协作图
数据流图
控制流图
处理过程描述
状态图
顺序图
构件图
设计类
活动图
顺序图
求精：
构件图
设计类
活动图
顺序图
需求图
约束
交互操作
目标和配置
设计类实现
子系统
协作图
构件图
设计类
活动图
顺序图
部署图
体系结构元素
接口元素
构件级元素
部署级元素
过程维度

# 设计模型元素

- 数据设计元素
- 体系结构设计元素
- 接口设计元素
- 构件级设计元素
- 部署级设计元素

# 数据设计元素

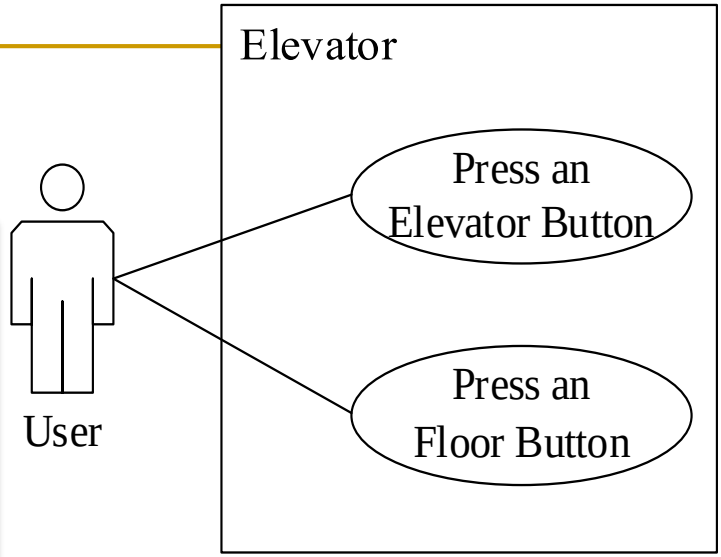
- 完成类图
  - 确定属性的格式
    - 通常由需求分析得到
    - 实际情况下，oo是迭代过程，越晚为类增加属性越好
    - 例如日期格式，美国为mm/dd/yyyy，欧洲为dd/mm/yyyy，中国为yyyy/mm/dd，长度为10个字符
  - 确定类的方法
    - 基于交互模型确定所有的方法
    - 确定方法属于哪个类
      - 信息隐藏：类的属性是私有的，属性上的操作在该类中
      - 职责驱动设计：如果一个客户端向一个对象发消息，则该对象就要为执行客户端的请求负责。

# 数据设计元素

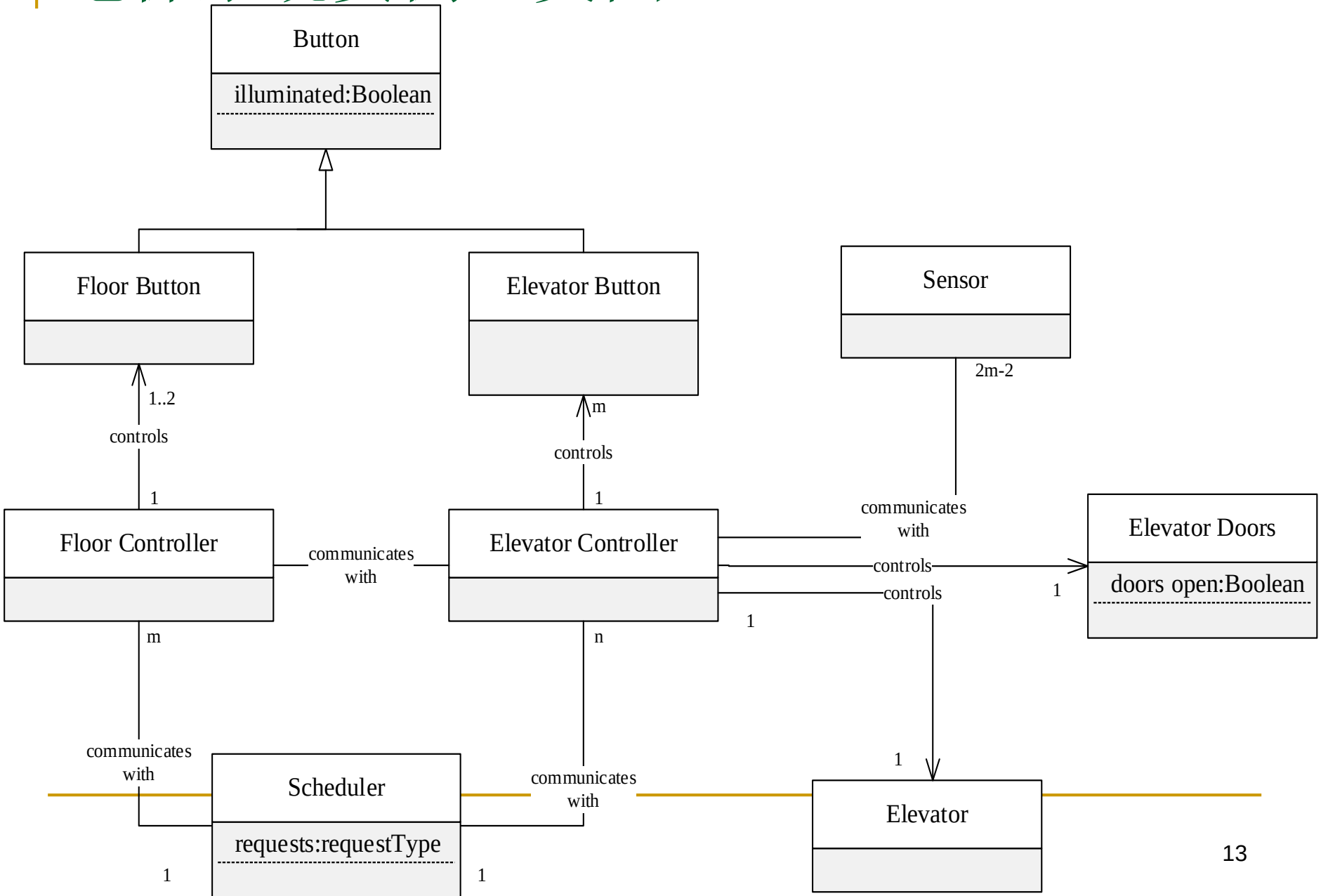
- **完成类的详细设计**
- **数据设计**
  - 在体系结构层上，设计数据库
  - 在构件层上，设计数据结构及其算法

# 电梯系统实例

1、用户A在3楼按了向上的楼层按钮，他想去7楼。
2、向上的楼层按钮灯亮了。
3、一部电梯到达3楼，用户B在电梯里，他从1楼进电梯，想去9楼。
4、电梯门开了。
5、计时器开始计时。
用户A进入电梯
6、用户A按下7楼的电梯按钮。
7、7楼的电梯按钮灯亮了。
8、计时时间到，电梯门关上了。
9、向上的楼层按钮灯灭了。
10、电梯上升至7楼。
11、 7楼的电梯按钮灯灭了。
12、电梯门开了。
13、计时器开始计时。
用户A走出电梯
14、计时时间到，电梯门关上了。
15、电梯载着用户B继续向9楼行进。



# 电梯系统实例（类图）


Button
illuminated:Boolean
Floor Button
Elevator Button
Sensor
1..2
controls
1
m
controls
1
2m-2
Floor Controller
communicates
with
Elevator Controller
communicates
with
controls
controls
Elevator Doors
doors open:Boolean
1
1
m
n
communicates
with
Scheduler
requests:requestType
communicates
with
1
Elevator
1
1

# 电梯系统实例（分配职责）

Class

**Elevator Controller**

Responsibility

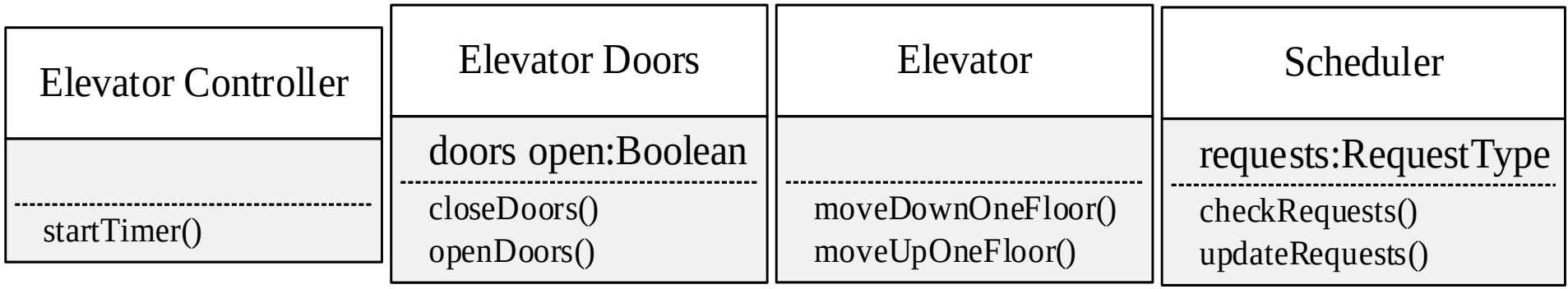
1. Send message to **Elevator Button** to check if it is turned on
2. Send message to **Elevator Button** to turn itself on
3. Send message to **Elevator Button** to turn itself off
4. Send message to **Elevator Doors** to open themselves
5. Start timer
6. Send message to **Elevator Doors** to close themselves after time out
7. Send message to **Elevator** to move itself up one floor
8. Send message to **Elevator** to move itself down one floor
9. Send message to **Scheduler** that a request has been made
10. Send message to **Scheduler** that a request has been satisfied
11. Send message to **Scheduler** to check if the elevator is to stop at the next floor
12. Send message to **Floor Controller** that elevator has left floor

# 电梯系统实例（增加方法）

- 职责驱动设计
  - 事件5表示**Elevator Controller**有启动计时器的职责，因此，**Elevator Controller**具有该方法
  - 其余11个事件
    - “Send a message to another class to tell it to do something.”
    - 为相应的类增加方法
- 所有12个事件都遵循信息隐藏原则

# 电梯系统实例（增加方法）

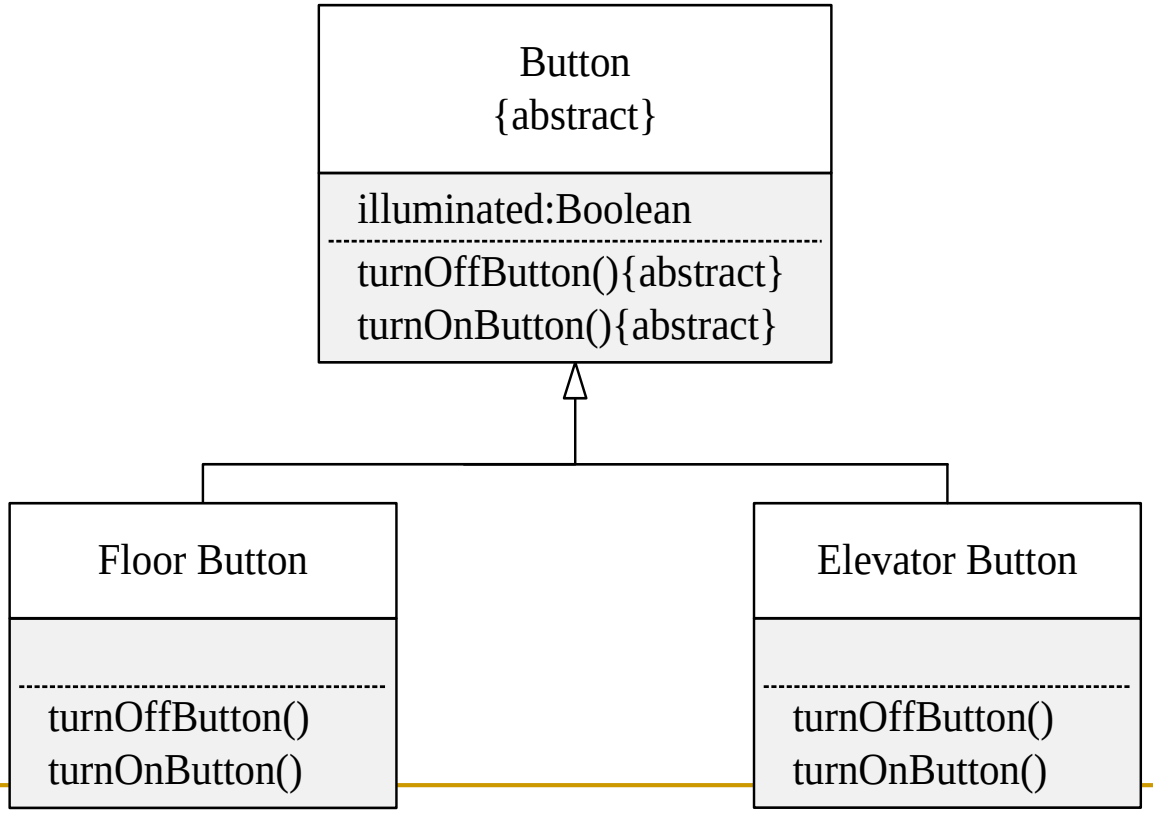
1. **Elevator Controller** starts timer.
2. **Elevator Controller** sends message to **Elevator Doors** to close or open themselves.
3. **Elevator Controller** sends message to **Elevator** to move itself down or up one floor.
4. **Elevator Controller** sends message to **Scheduler** to check if the elevator is to stop at the next floor.



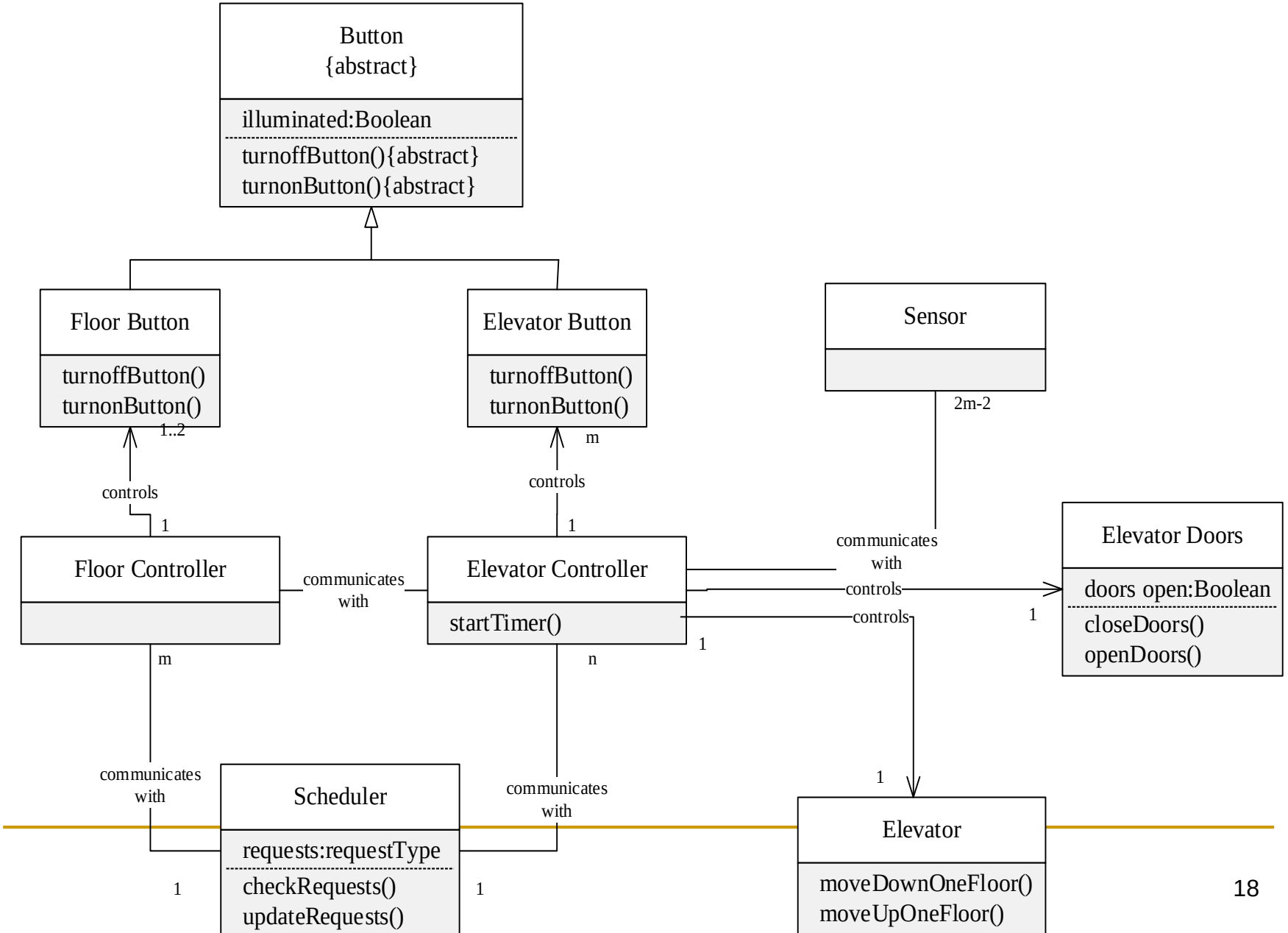
# 电梯系统实例（增加方法）

1. **Elevator Controller** sends message to **Elevator Button** to turn itself on or off.

2. **Floor Controller** sends message to **Floor Button** to turn itself on or off.



# 电梯系统实例


Button
{abstract}
illuminated:Boolean
turnoffButton(){abstract}
turnonButton(){abstract}
Floor Button
turnoffButton()
turnonButton()
Elevator Button
turnoffButton()
turnonButton()
Sensor
1..2
m
2m-2
controls
1
controls
1
Floor Controller
communicates
with
Elevator Controller
startTimer()
communicates
with
controls
controls
Elevator Doors
doors open:Boolean
closeDoors()
openDoors()
1
1
m
n
1
communicates
with
communicates
with
1
Scheduler
requests:requestType
checkRequests()
updateRequests()
1
Elevator
moveDownOneFloor()
moveUpOneFloor()

# 体系结构设计元素

- 类似于房屋的建筑平面图
  - 描绘房间的总体布局：大小、形状、位置关系、门、窗
  - 软件体系结构给出软件的全貌图
- 体系结构模型
  - 源于软件应用领域的信息
  - 源于需求模型：用例、类及其之间的关系
  - 源于已有的体系结构模式

# 接口设计元素

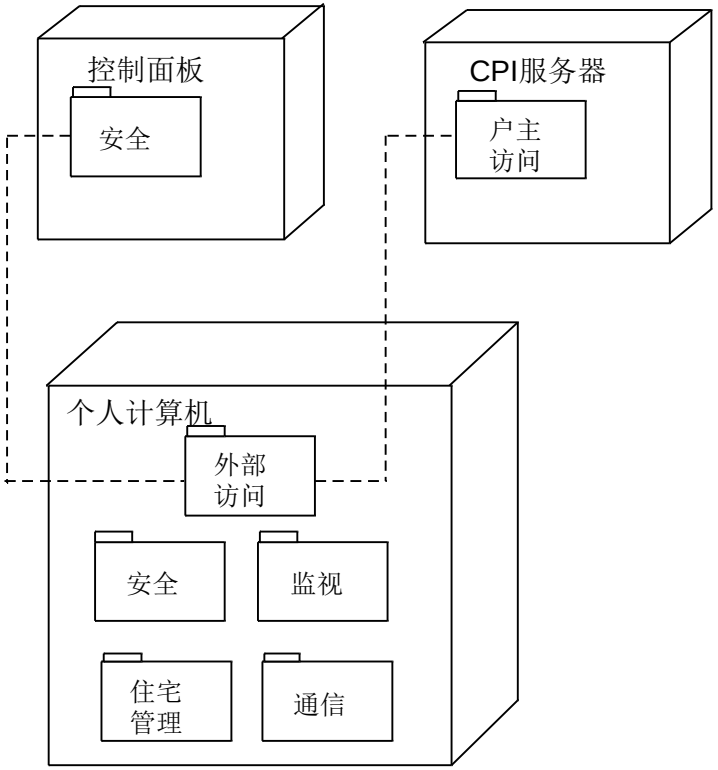
- 类似于房屋的门窗及外部设施的详细绘图（说明）
  - 描绘物体是如何进出房屋的
  - 软件接口描绘信息流是如何输入输出系统的，以及构件间是如何通信的
- 接口设计包括三个方面
  - 用户界面（UI）
    - 审美元素：布局、颜色、图表和交互机制
    - 人体工程学元素：信息布局、隐喻（metaphor）、UI导航
    - 技术元素：UI模式、可复用构件
  - 与外部系统的接口
    - 在需求工程阶段确定
  - 内部不同构件间的接口
    - 在构件设计阶段确定

# 构件级设计元素

- 类似于房屋每个房间的详细绘图（说明）
  - 描绘房间的布线、管道，以及电插座、开关面板、水龙头、水槽、淋浴、浴缸、地漏、橱柜、衣帽间的位置等
  - 完整的描述每个构件的**内部细节**
- 构件设计包括三个方面
  - 数据结构
    - 伪代码
  - 算法
    - 伪代码
    - 图表工具（流程图、盒图、判定表等）
  - 访问所有服务的接口

# 部署级设计元素

- 指明软件功能和子系统将如何在支持软件的物理计算环境内分布
- UML部署图建模
- *描述符形式*的部署图显示了计算环境，但并没有明确地说明配置细节
- *实例形式*的部署图在后面阶段的设计中明确硬件配置细节



# 小结