CPT203_w5_Context model_interation model

Study Target

一、Context Modeling

上下文模型

二、Interaction Model

交互模型（这节课就俩，一个用例模型一个时序模型）

What is modeling

A model is an abstract view of a system which ignore system details. A complete system model needs to include the System's context, iteraction，structure and behaviour

模型是对系统的抽象简化描述，一个完整的系统模型需要涵盖系统的上下文，交互，结构与行为

Why modeling

Modelling simplify the complex system by abstracting the complex system into appropriate level of details

建模通过将复杂系统抽象为适当的细节级别来简化复杂系统

System modelling helps the analyst to understand the functionality of the system and models are used to communicate with customers.

系统建模有助于分析师了解系统的功能，模型用于与客户沟通。

System perspectives

An external perspective, where you model the context or environment of the system.

外部视角，您可以在其中对系统的上下文或环境进行建模。

An interaction perspective, where you model the interactions between a system and its environment, or between the components of a system.

交互透视图，用于对系统与其环境之间或系统组件之间的交互进行建模。

A structural perspective, where you model the organization of a system or the structure of the data that is processed by the system.

结构透视图，您可以在其中对系统的组织或系统处理的数据的结构进行建模。

A behavioral perspective, where you model the dynamic behavior of the system and how it responds to events.

行为视角，您可以对系统的动态行为及其对事件的响应进行建模。

Context models

Context models are used to illustrate the operational context of a system- they show what lies outside the system boundaries.

上下文模型用于说明系统的操作上下文 - 它们显示系统边界之外的内容。

Social and organisational concerns may affect the decision on where to position system boundaries.

社会和组织问题可能会影响系统边界位置的决定。

System boundaries are established to define what is inside and what is outside the system：

建立系统边界来定义系统内部和系统外部。：

They show other systems that are used or depend on thesystem being developed.

它们显示正在使用或依赖于正在开发的系统的其他系统。

The position of the system boundary has a profound effect on the system requirements.

系统边界的位置对系统需求有深远的影响。

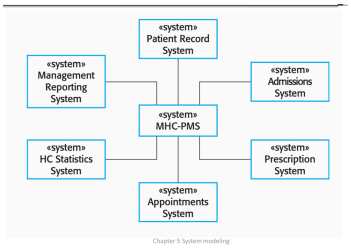
Defining a system boundary is a political judgment：

定义系统边界是一种政治判断（这个翻译感觉有问题，我后面求证一下学长）

There may be pressures to develop system boundaries that increase / decrease the influence or workload of different partsof an organization.

可能存在制定系统边界的压力，这些边界会增加/减少组织不同部分的影响力或工作量

实例图形：



Context models simply show the other systems in the environment, not how the system being developed is used in that environment.

上下文模型只是显示环境中的其他系统，而不是正在开发的系统在该环境中的使用方式。

Used along with other models, such as business process models .

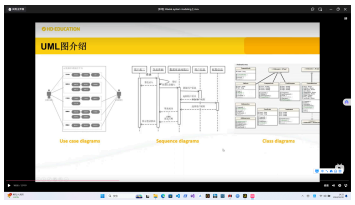
与其他模型一起使用，例如业务流程模型。

UML activity diagrams may be used to define business process models.

UML 活动图可用于定义业务流程模型。

ps：这个图目前来看是暂时没有要求会画，但是后面两个图必须要会画

Interation model



Use Case

Description

A picture describes how actors relate to use cases and use cases relate to one another

一张用来描述actors和用例以及用例彼此之间关系的图片

They do not capture the full information of the actual use cases

它们不会捕获实际用例的完整信息

Purpose

Specify the context of a system

指定系统的上下文

Capture the requirements of a system

捕获系统的要求

Validate a systems architecture

验证系统架构

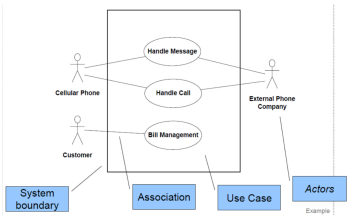
Drive implementation and generate test cases

推动实施并生成测试用例

Developed by analysts and domain experts

由分析师和领域专家开发

示例如图



What is an Actor?

Actor is all user roles that interact with the system

所有与系统存在交互的角色都叫actor

Include system components only if they responsible for initiating/triggering a use case.

仅当系统组件负责启动/触发用例时，才包含系统组件。

 For example, a timer that triggers sending of an e-mail reminder

例如，触发发送电子邮件提醒的计时器

primary - a user whose goals are fulfilled by the system

primary 系统实现其目标的用户

importance: define user goals

重要性： 定义用户目标

supporting - provides a service (e.g., info) to the system

supporting - 为系统提供服务（例如，信息）

importance: clarify external interfaces and protocols

重要性：阐明外部接口和协议

offstage - has an interest in the behavior but is not primary or supporting, e.g., government

offstage - 对行为感兴趣，但不是主要或支持者，例如政府

importance: ensure all interests (even subtle) are identified and satisfied

重要性：确保所有利益（即使是细微的）都得到识别和满足

How to find actors?

Ask the following questions:

Who are the system's primary users?

谁是系统的主要用户?

Who requires system support for daily tasks?

谁的日常工作需要系统支持

Who are the system's secondary users?

谁是系统的二级使用者

What hardware does the system handle?

系统需要处理哪些硬件

Which other (if any) systems interact with the system in question?

其他的什么系统会如何与该系统交互

Do any entities interacting with the system perform multiple roles as actors?

是否有任何与系统交互的实体作为参与者扮演多个角色?

Which other entities (human or otherwise) might have an interest in the system's output?

哪些其他实体（人类或其他实体）可能对系统的输出感兴趣?

Elements of use case diagram 用例图的要素

use case

System function (process – automated or manual).

系统功能（过程 - 自动或手动）。

Named by verb.Do something

用动词命名。做点什么

Each Actor must be linked to a use case, while some use cases may not be linked to actors.

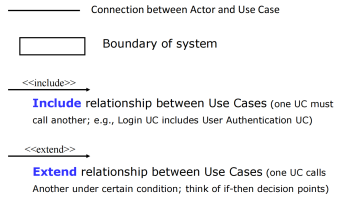
每个 Actor 都必须链接到一个用例，而某些用例可能未链接到 Actor

USER/ACTOR	USER GOAL = Use Case
Order clerk	Look up item availability Create new order Update order
Shipping clerk	Record order fulfillment Record back order
Merchandising manager	Create special promotion Produce catalog activity report

以下是表格内容的中文翻译：

用户/角色	用户目标 = 用例
订单员	- 查询商品库存
	- 创建新订单
	- 更新订单
发货员	- 记录订单履行情况
	- 记录缺货订单
商品经理	- 创建特别促销活动
	- 生成目录活动报告

Other details其他细节



Linking Use Cases

Association relationships

关联 关系

Generalization relationships：

泛化 关系：

One element (child) "is based on" another element (parent)

一个元素（子元素）“基于”另一个元素（父元素）

Include relationships

包括 关系

One use case (base) includes the functionality of another (inclusion case)

一个用例 （base） 包含另一个 （包含案例） 的功能

Supports re-use of functionality

支持功能重用

Extend relationships

扩展 关系

One use case (extension) extends the behavior of another (base)

一个用例（扩展）扩展了另一个用例（基本）的行为

Generalization 归纳

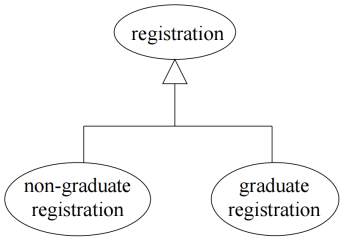
The child use case inherits the behavior and meaning of the parent use case.

子用例继承了父用例的行为和含义。

The child may add to or override the behavior of its parent.

子项可以添加或覆盖其父项的行为。

具体例子可以参照PPT的原图



Include 包含

The base use case explicitly incorporates the behavior of another use case at a location specified in the base.

基本用例在基本用例中指定的位置显式合并了另一个用例的行为。

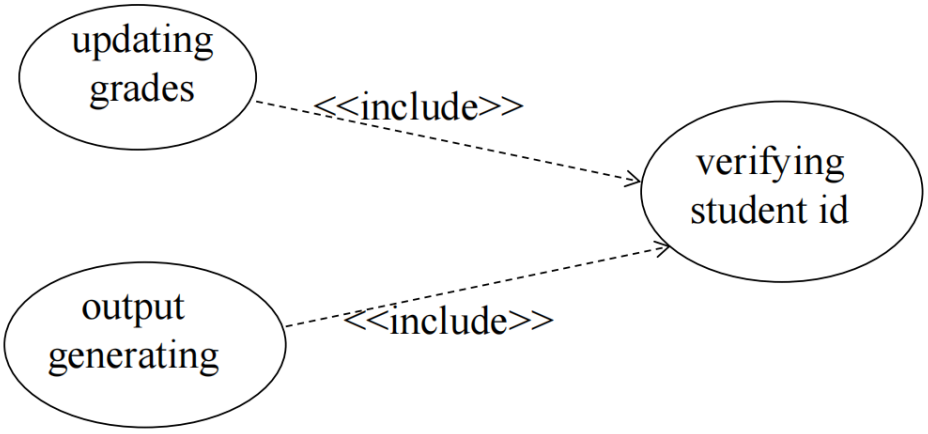
The included use case never stands alone. It only occurs as a part of some larger base that includes it.

包含的使用案例从来都不是孤立的。它仅作为包含它的某个更大基础的一部分出现。

More about Include:

Enables us to avoid describing the same flow of events several times by putting the common behavior in a use case of its own.

通过将常见行为放在其自己的用例中，使我们能够避免多次描述相同的事件流



个人认为这个比较抽象，详细的可以参照一下这个链接的内容

https://www.bilibili.com/video/BV1Vp4y1D7Sh/?share_source=copy_web

假设你是一家水果店的系统设计师，正在为订单处理流程绘制UML的用例图，其中包含“购买西瓜”的主要用例。我们可以将“购买西瓜”的过程分解为几个步骤，用例图中的 **include** 关系就是用于表示某个用例在执行时总是会包含其他用例。

举例：

1. 主要用例：购买西瓜
   - 这是用户的主要目标，就是在系统中购买一个西瓜。
2. 包含的用例：选择西瓜
   - 在用户购买西瓜时，他们必须先选择一个西瓜，这个动作是“购买西瓜”的一部分。
3. 包含的用例：结账
   - 在“购买西瓜”过程中，用户总是需要完成结账操作，因此这个步骤也是被包括进“购买西瓜”中的。

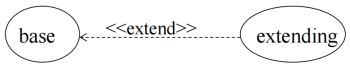
在这种情况下，include 关系用于表示“购买西瓜”用例始终包含“选择西瓜”和“结账”两个子用例。这些步骤是购买西瓜过程中不可或缺的部分，因此通过include关系来明确它们是被主用例调用的。

UML图中的include关系：

- 购买西瓜—— include ——> 选择西瓜
- 购买西瓜—— include ——> 结账

这就意味着每当用户执行“购买西瓜”时，系统必须先执行“选择西瓜”和“结账”用例，二者是购买过程中的子操作。

Extend

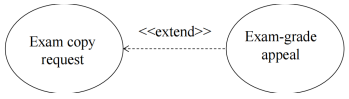


The base use case implicitly incorporates the behavior of another use case at certain points called extension points.

基本用例隐式地合并了另一个用例在某些点（称为扩展点）的行为。

The base use case may stand alone, but under certain conditions its behavior may be extended by the behavior of another use case.

基本用例可以独立存在，但在某些情况下，其行为可能会因另一个用例的行为而扩展。



More about Extend

Enables to model optional behavior or branching under conditions.

启用以对条件下的可选行为或分支进行建模。

Extend Relationship

Extend relationship – linking an optional use case to a standard use case.

扩展 关系 – 将 可选 使用案例链接到标准使用案例。Example: Register Course (standard use case) may have Register for Special Class (extend use case) – class for non-standard students, in unusual time, with special topics, requiring extra fees…).

示例：注册课程（标准用例）可能有 注册特殊课程（扩展用例）——非标准学生的课程，在不寻常的时间，有特殊主题，需要额外费用.....

The optional UC extends the standard UC

可选的 UC 扩展了标准 UC

Standard use case can execute without the extend case loose coupling.

标准用例可以在没有扩展案例松散耦合的情况下执行。

Example

1. 主要用例：购买西瓜
    - 用户的主要目标是购买西瓜。
2. 扩展用例：切西瓜
    - 在某些情况下，水果店可能提供附加服务，比如用户可以选择让店员帮忙把西瓜切开。
    - 但是，“切西瓜”这个步骤不是购买西瓜的必要流程，只有用户要求时才会触发这个额外操作。

在这种情况下，extend 关系用于表示“切西瓜”用例是对“购买西瓜”用例的一个扩展。用户并不是每次买西瓜都要进行这个操作，只有当用户选择了这项附加服务时，才会执行“切西瓜”的功能。

UML图中的extend关系：

- 购买西瓜 —— extend ——> 切西瓜 （条件：用户要求切西瓜）

这表示“购买西瓜”用例在某些特定条件下（用户请求）会扩展为“切西瓜”，而这并不是购买过程的必需步骤。

Sequence Diagrams

时序图

Describe the flow of messages, events, actions between objects

描述对象之间的消息、事件、操作流

Show concurrent processes and activations

显示并发进程和激活

Show time sequences that are not easily depicted in other diagrams显示其他图表中不易描述的时间串行

Typically used during analysis and design to document and understand the logical flow of

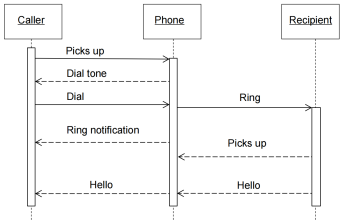
your system

通常在分析和设计期间用于记录和了解系统的逻辑流程

Emphasis on time ordering!

强调时间排序

Sequence Diagram (make a phone call)



Representing Objects 表示对象

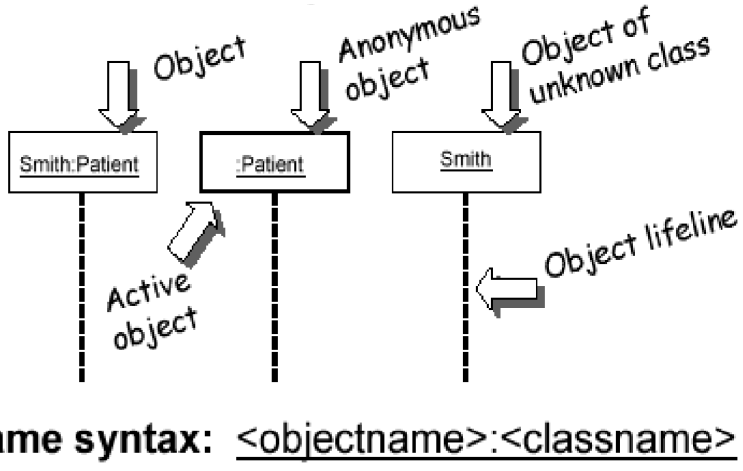
Squares with object type, optionally preceded by "name :"具有对象类型的正方形，可选择以“name： ”开头

write object's name if it clarifies the diagram

write object's name（如果它阐明了图表）

object's "life line" represented by dashed vert. line

对象的“生命线”由虚线表示



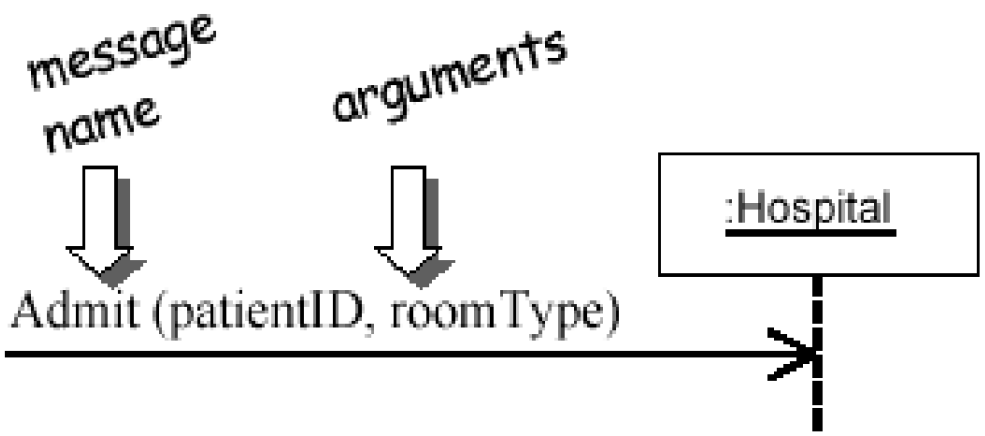
Messages Between Objects 对象之间的消息

messages (method calls) indicated by arrow to other object

messages （方法调用） 由箭头指示到其他对象

write message name and arguments above arrow

将消息名称和参数写在箭头上方



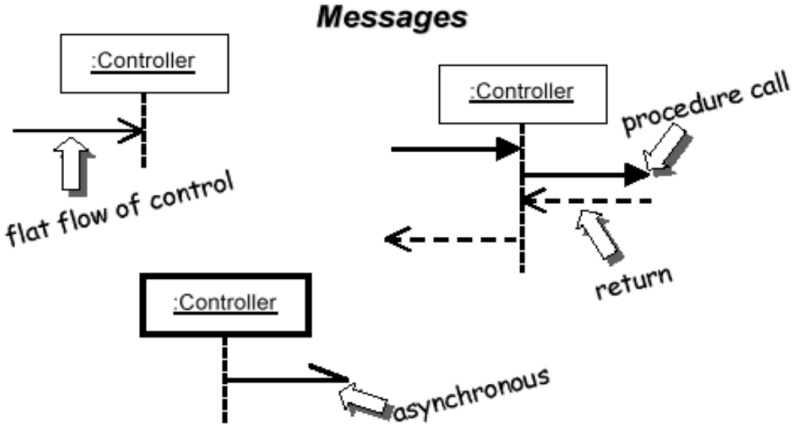
messages (method calls) indicated by arrow to other object

messages （方法调用） 由箭头指示到其他对象

dashed arrow back indicates return

虚线箭头向后表示返回different arrowheads for normal / concurrent (asynchronous) calls

正常/并发（异步）调用的不同箭头



Indicating method calls 指示方法调用

activation: thick box over object's life line; drawn when object's method is on the stack

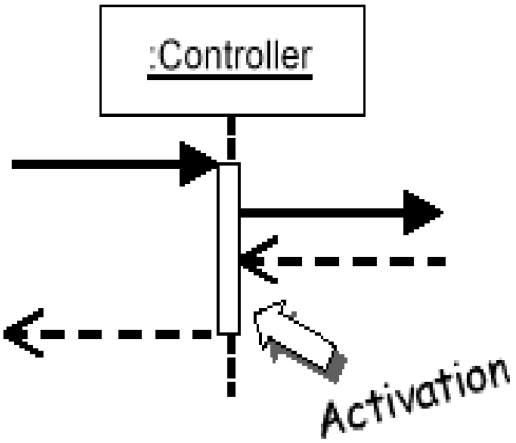
激活 （Activation）： 对象生命线上的厚框;当 Object 的方法位于堆栈上时绘制

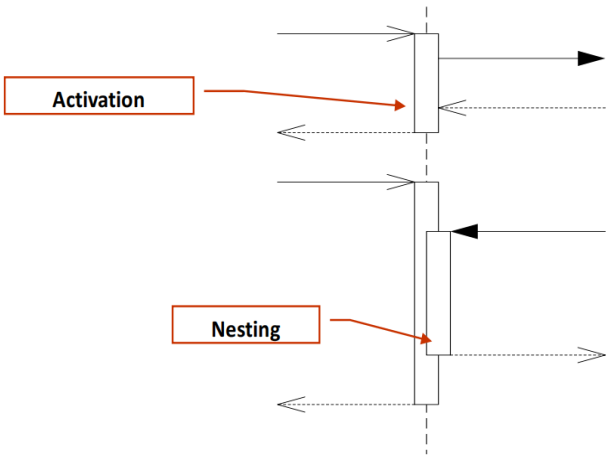
either that object is running its code,or it is on the stack waiting for another object's method to finish

该对象正在运行其代码，或者它正在堆栈上等待另一个对象的方法完成

nest activations to indicate recursion

嵌套激活以指示递归





（未完待续，loop和递归感觉没说明白还得再看看）