2

Modeling Requirements
需求建模

Shaoning Zeng

UNIFIED MODELING LANGUAGE™

Review 回顾

UML	Modeling Language 建模语言	Model 模型
Diagram 图	View 视图	4+1 View 4+1视图

Review 回顾

- The Unified Modeling Language (UML) is the **standard modeling language** for software and systems development. UML是标准建模语言。
- To effectively model a system, you **need** one very important thing: **a language** with which the model can be described. (**UML**) 描述模型的语言
- A modeling language can be made up of pseudo-code, actual code, pictures, diagrams, or long passages of description; in fact, it's pretty much anything that helps you describe your system. The elements that make up a modeling language are called its **notation** .

Feedback

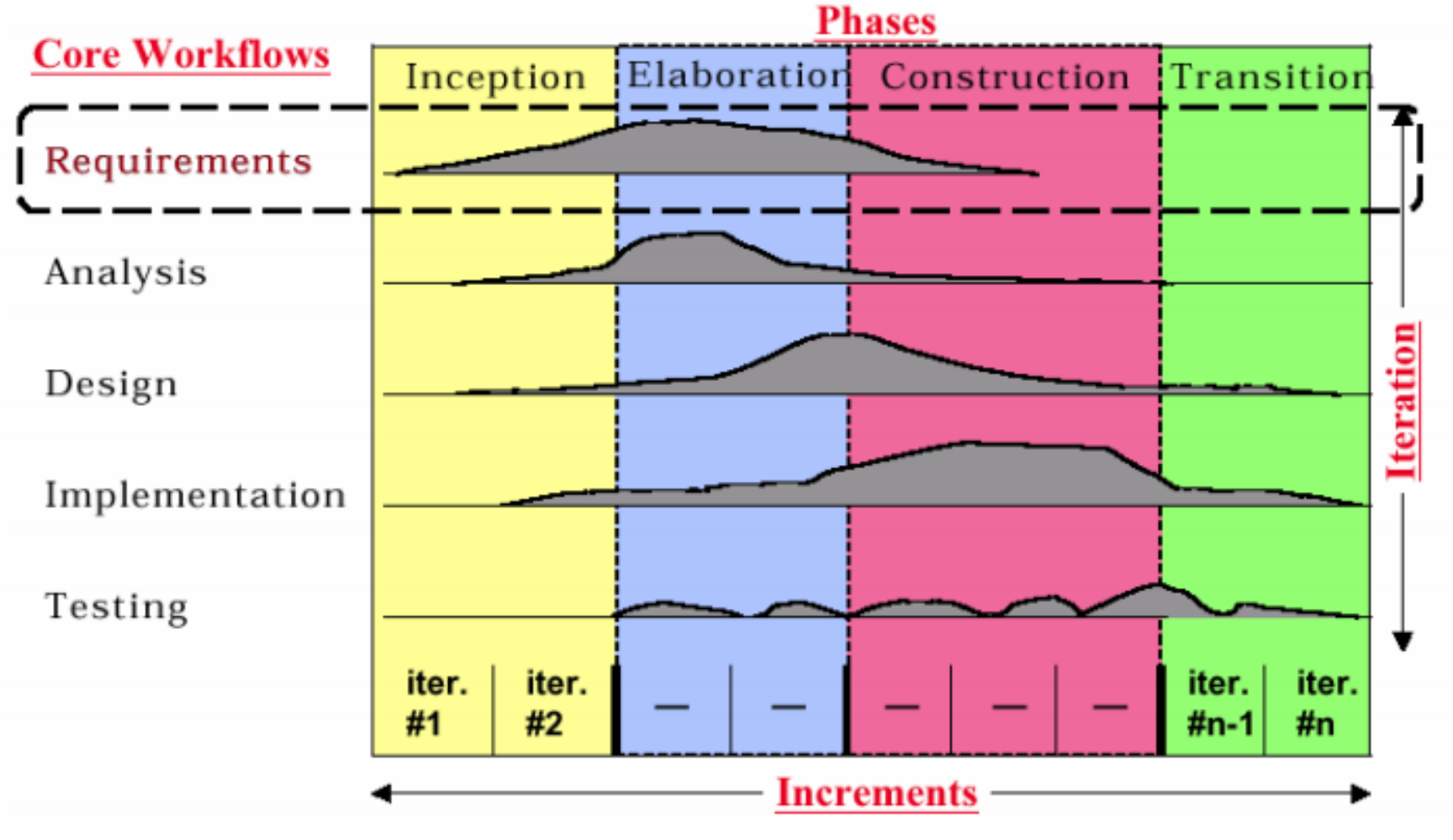
JustQyx：老师讲的内容很明确，但是映射到我们脑子里的思路还不够清晰，内容的划分是很明确了，可以每讲一点，就把那一点写在黑板。这样我们就能快速地把这些点都连起来 (9月5日 15:58)

删除 | 回复

2. Modeling Requirements 需求建模

- 2.1. Capturing a System Requirement
  - 捕捉系统需求 ☆☆☆☆☆
- 2.2. Use Case Relationships
  - 用例的关系 ☆☆☆☆☆
- 2.3. Use Case Overview Diagrams
  - 用例概述图

Requirements Analysis 需求分析

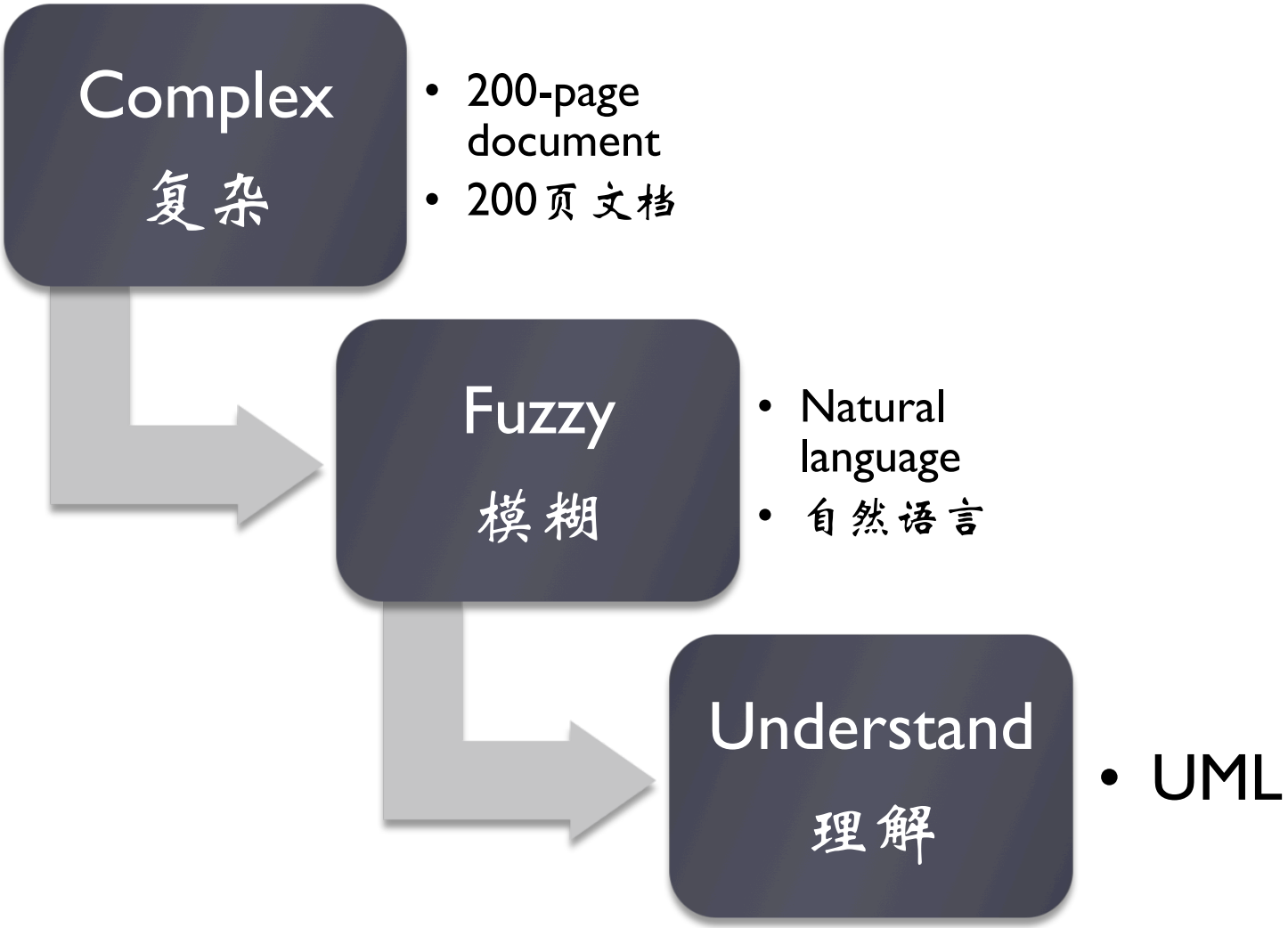


参考资料

Requirements Analysis 需求分析

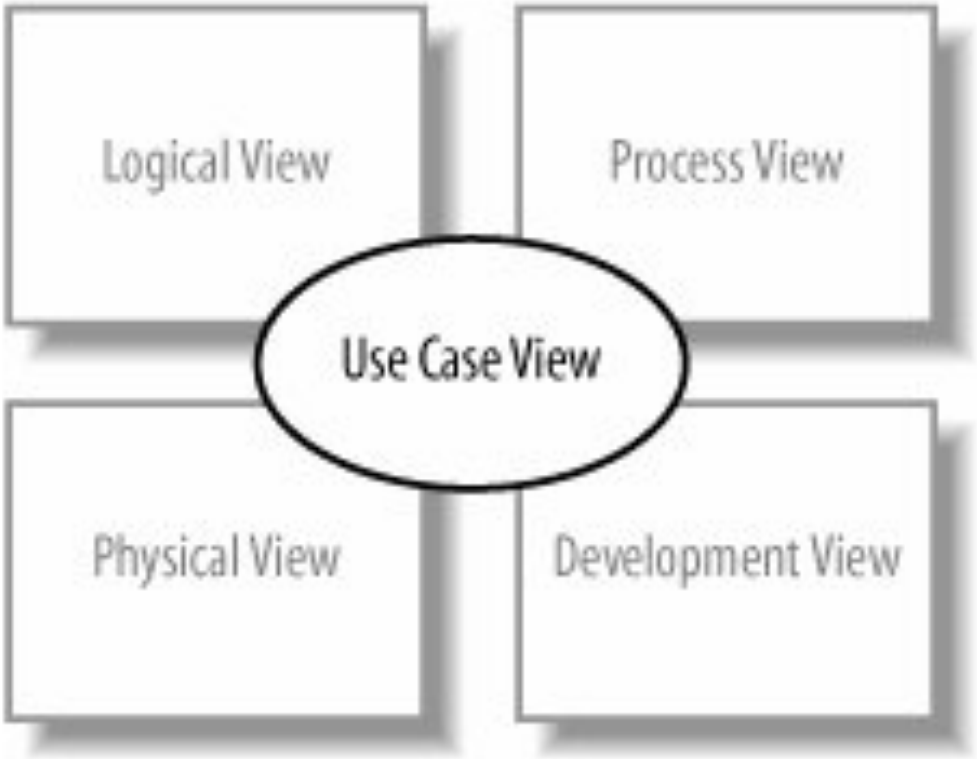
- http://en.wikipedia.org/wiki/Requirements_analysis

Real-world Requirements 需求描述



Use case 用例

- A use case is a case (or situation) where your system is used to fulfill one or more of your user's requirements;
  - 系统解决一个或多个用户需求的场景/情况/案例
- A use case captures a piece of functionality that the system provides.
  - 系统所提供的一个功能
- Use cases are at the heart of your model, since they affect and guide all of the other elements within your system's design.
  - 用例是模型的核心，因为它影响系统设计的其他元素



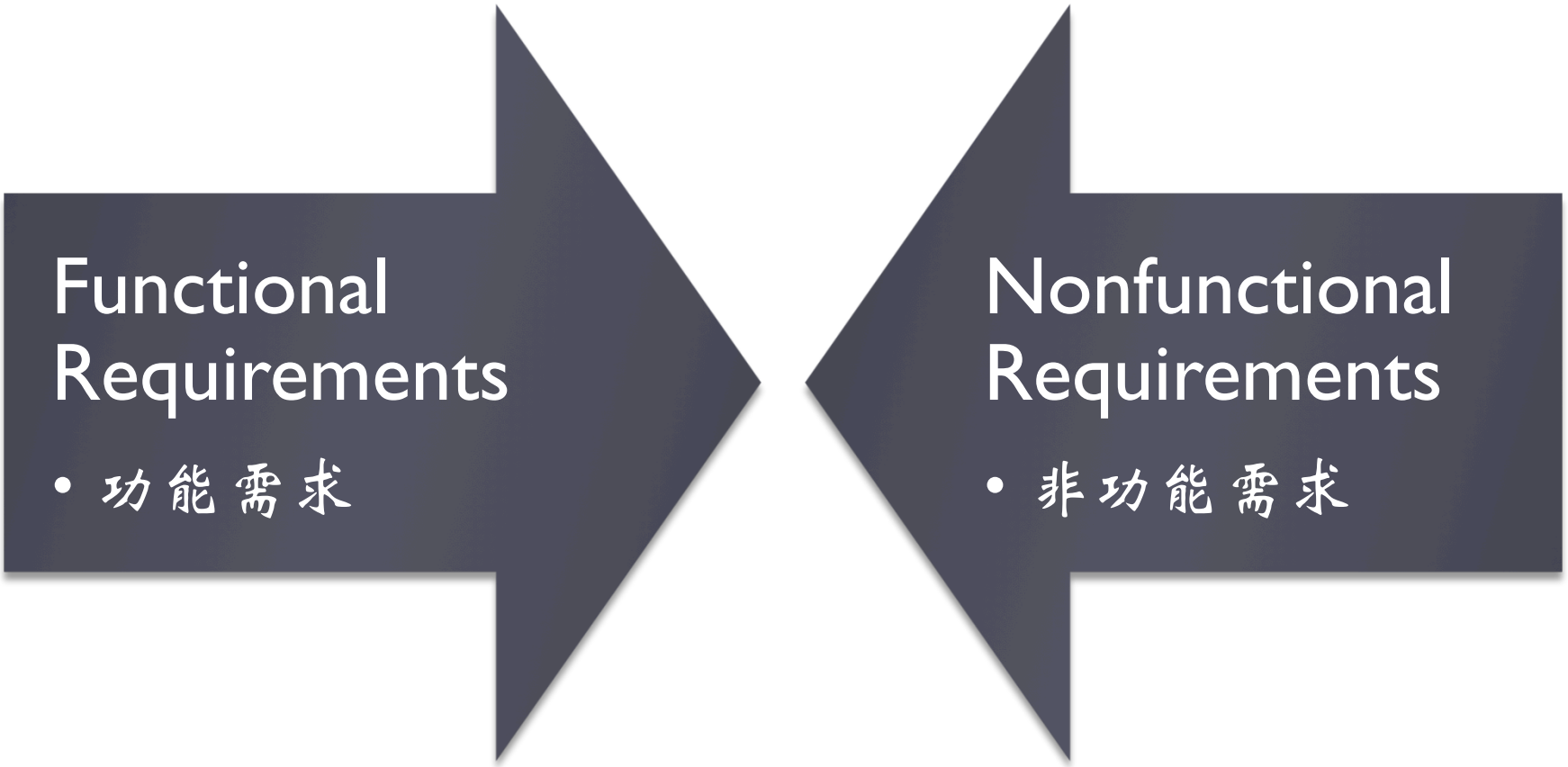
Use cases affect every other facet of your system's design; they capture what is required and the other views on your model, then show how those requirements are met

用例影响系统设计的各个方面

Use case

- Use cases are an excellent starting point for just about every facet of object-oriented system development, design, testing, and documentation. 起点
  - 用例是面向对象系统开发、设计、测试和文档等各个方面的起点
- They describe a system's requirements strictly from the outside looking in; they specify the value that the system delivers to users. 外部视图
  - 反映系统交付给用户的价值
- Because use cases are your system's functional requirements, they should be the first serious output from your model after a project is started. 功能需求
  - 用例是指功能需求，是项目启动后模型的第一个输出

Question


Functional Requirements
• 功能需求
Nonfunctional Requirements
• 非功能需求

Functional v.s. Nonfunctional

- Use cases specify only what your system is supposed to do, i.e., the system's functional requirements.
  - 用例只规定了系统的功能需求
- They do not specify what the system shall not do, i.e., the system's nonfunctional requirements.
  - 用例不能规定系统的非功能需求
- Nonfunctional requirements often include performance targets and programming languages, etc.
  - 非功能需求包括性能、编程语言、平台选择等

For different roles 针对不同角色

- Use cases are a means to bring those gaps in the user's requirements to the forefront at the beginning of a project. 用户之间的沟通桥梁
- This is a real bonus for the system designer since a gap or lack of understanding identified early on in a project's development will cost far less in both time and money than a problem that is not found until the end of a project. 帮助设计人员及早发现问题
- Once a gap has been identified, go back to the system's stakeholders, the customers and users, so they can provide the missing information.
  - 项目干系人、客户和用户等

For project management 针对项目管理

- Once priority and risk are assigned to a use case, it can help manage a project's workload.
  - 通过用例（优先级和风险级别）来管理项目工作量
- Your use cases can be assigned to teams or individuals to be implemented and, since a use case represents tangible(有形) user value, you can track the progress of the project by use cases delivered. 跟踪进度
  - 通过用例完成情况的跟踪项目进度
- If and when a project gets into schedule trouble, use cases can be jettisoned(放弃) or delayed to deliver the highest value soonest. 项目控制
  - 通过调整用例开发来控制项目

For testing 针对测试

- Last but not least, use cases also help construct tests for your system. 用例对于系统测试的作用
- Use cases provide an excellent starting point for building your test cases and procedures because they precisely capture a user's requirements and success criteria.
  - 用例是测试用例的基础
- What better way to test your system than by using the use cases that originally captured what the user wanted in the first place?
  - 用例是反映用户诉求的最佳载体

2.1. Capturing a System Requirement

- 2.1.1. Outside Your System: Actors 参与者
- 2.1.2. Use Cases 用例
- 2.1.3. Communication Lines
- 2.1.4. System Boundaries 系统边界
- 2.1.5. Use Case Descriptions 用例描述

Example

Requirement A.1

- The content management system shall allow an administrator to create a new blog account, provided the personal details of the new blogger are verified using the author credentials database.
  - 内容管理系统
  - 创建一个新的博客帐号
- There's actually no specific "best way" to start analyzing Requirement A.1, but one useful first step is to look at the things that interact with your system.
  - 第1步是寻找与系统交互的“东西”
- In use cases, these external things are called actors .

2.1.1. 系统之外：参与者 Actors

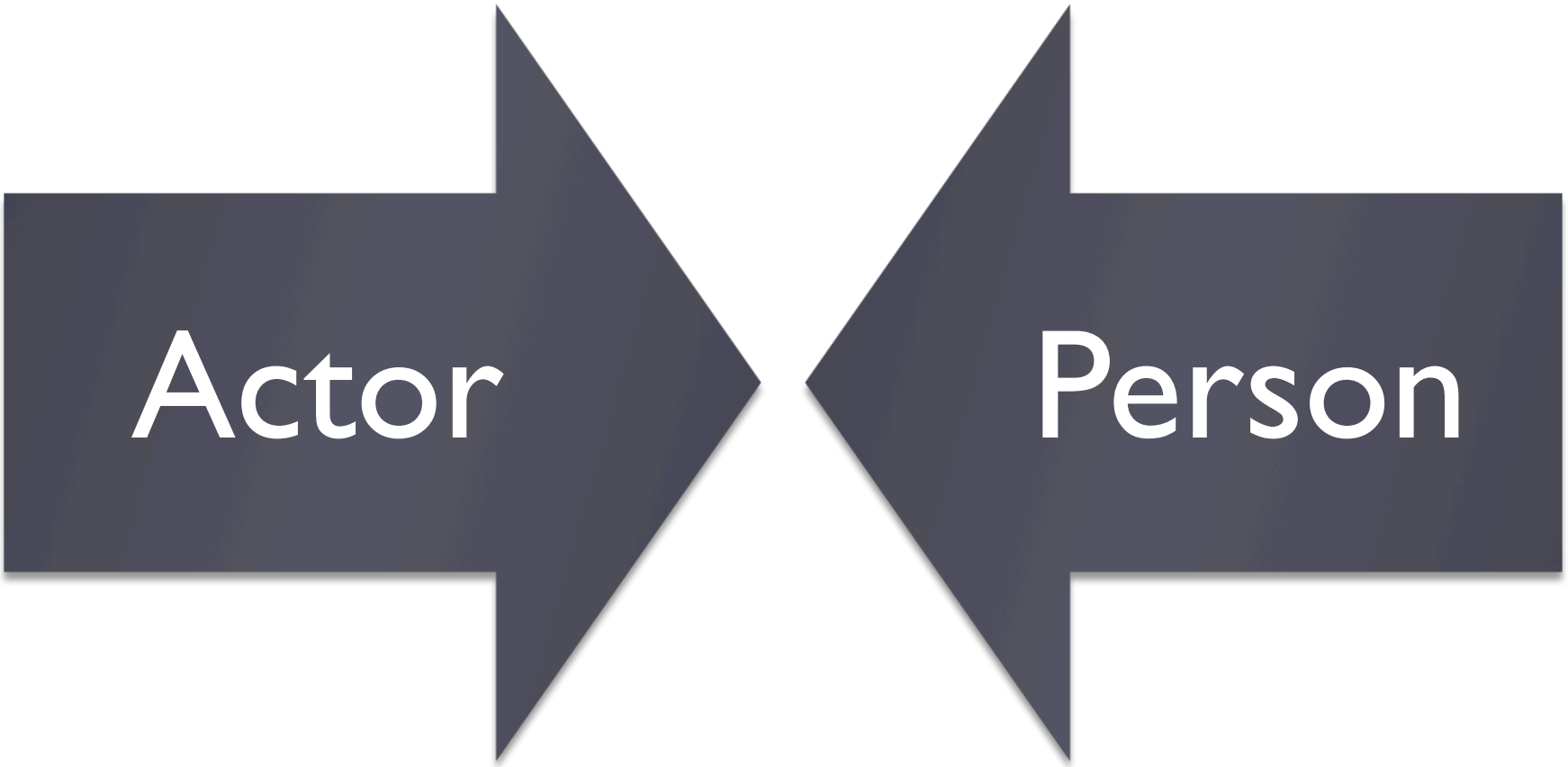
Requirement A.1 contains an Administrator actor that interacts with the system to create a blog account

需求A.1 管理员与系统交互，创建一个博客帐号

Naming actor 参与者的命名

- The best approach is to use a name that can be understood by both your customer and your system designers.
  - 使用不同的角色都能理解的词汇
- Wherever possible, use the original term for the actor as identified within your customer's requirements; that way, at least your use cases will be familiar to your customers.
  - 使用与业务相关的词汇
- This approach also lets system designers get comfortable with the system's unique context.
  - 使系统设计人员也能够明白系统的特殊上下文

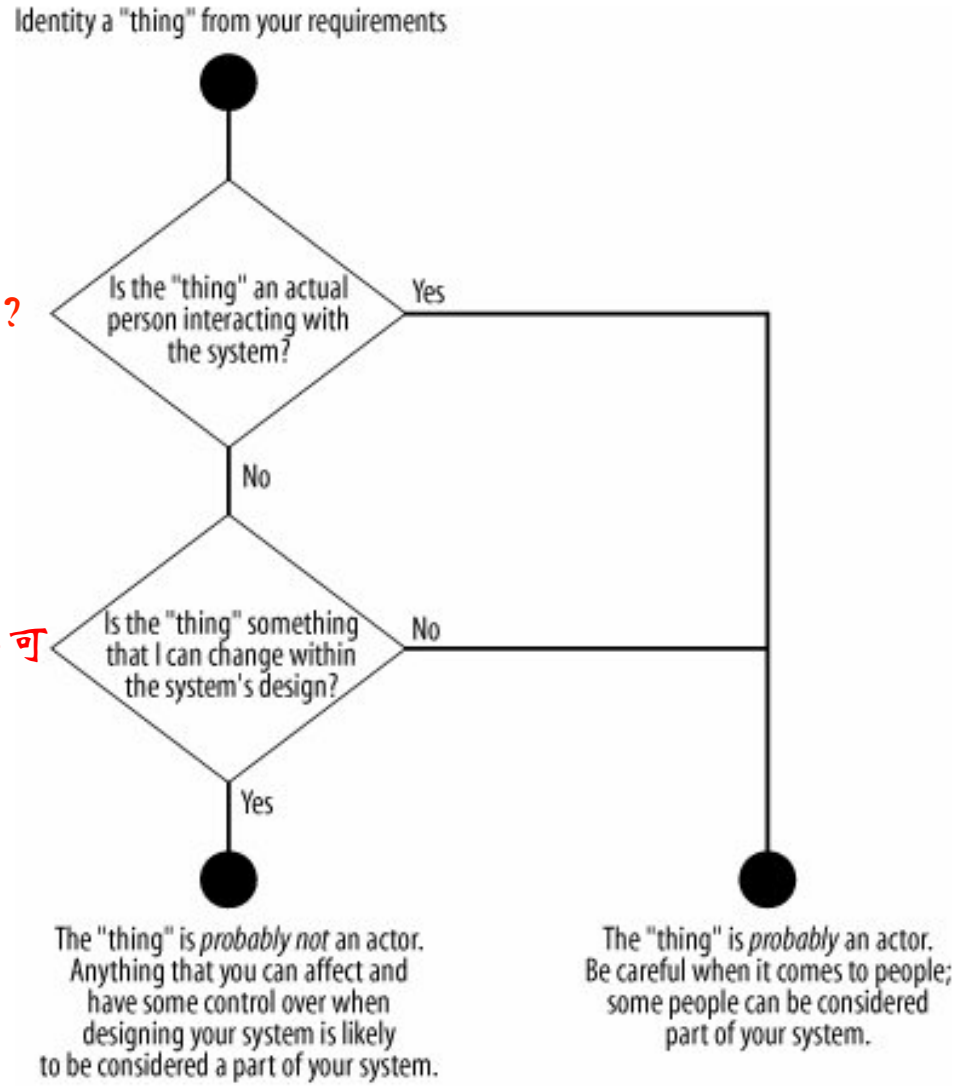
Question ?



Answer

- Actors don't have to be actual people.
  - 参与者不一定是“人”
- While an actor might be a person, it could also be a third party's system, such as in a business-to-business (B2B) application.
  - 除了人，参与者还可以是第三方系统
- Think of an actor as a black box: you cannot change an actor and you are not interested in how it works, but it must interact with your system.
  - 参与者可看作是一个黑盒

2.1.1.1. Tricky actors



是否与系统有交互？

在系统设计时是否可以修改？

在设计过程中可以控制的，很可能属于系统组成部分。

可能是参与者！

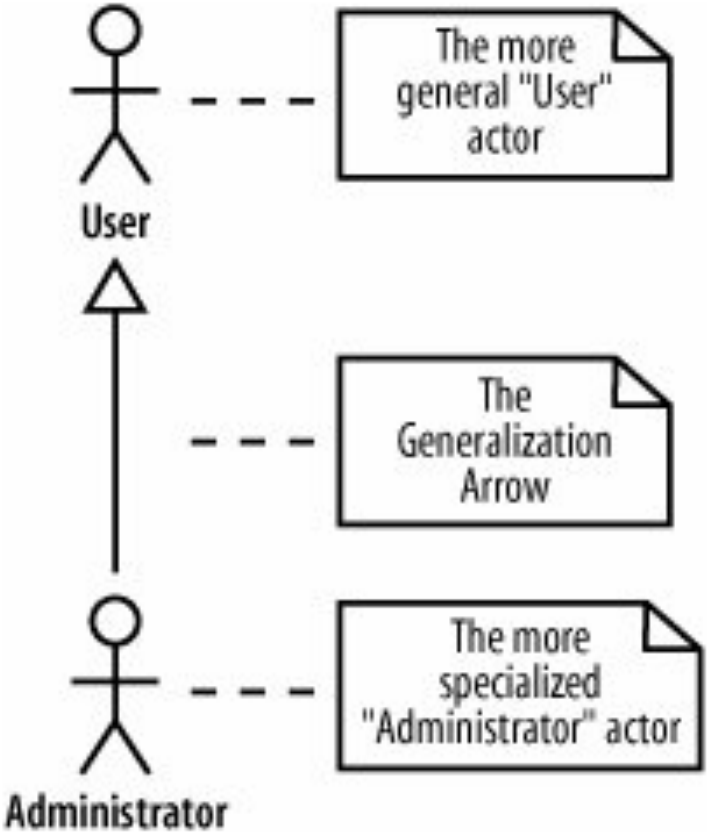
system clock 系统时钟

- The system clock comes into play when it invokes some behavior within your system. 时钟影响系统内部行为
  - It is hard to determine whether the system clock is an actor because the clock is not clearly outside of your system.
- As it turns out, the system clock is often best described as an actor because it is not something that you can influence. 通常可以将系统时钟看作参考者
  - Additionally, describing the clock as an actor will help when demonstrating that your system needs to perform a task based on the current time. 任务是基于当前时间来执行的

People other than users 非用户的人

- It is also tempting to focus on just the users of your systems as the actors in your model, but don't forget about other people, such as auditors, installers, maintainers, upgraders, and so on.
  - 审计人员、安装人员、维护人员、升级人员
- Those actors may have a veto（否决权）("We can't certify this system without proof that the data has not been tampered with") or they may have to enforce important nonfunctional requirements, such as an upgrade in a 10-minute system downtime window and an upgrade without shutting the system down, etc.
  - 一些重要角色提出的非功能需求

2.1.1.2. Refining actors 优化参与者



Showing that an administrator is a special kind of user

为管理员抽象出一个特殊的用户类

2.1.2. Use Cases 用例

- Use cases can be identified from your user's requirements.
  - 用例源于用户需求
- Remember, if use cases are truly requirements, then they must have very clear pass/fail criteria.
  - 必须有清晰的通过/失败条件
- The developer, the tester, the technical writer, and the user must explicitly know whether the system fulfills the use case or not.
  - 所有角色都需要理解用例

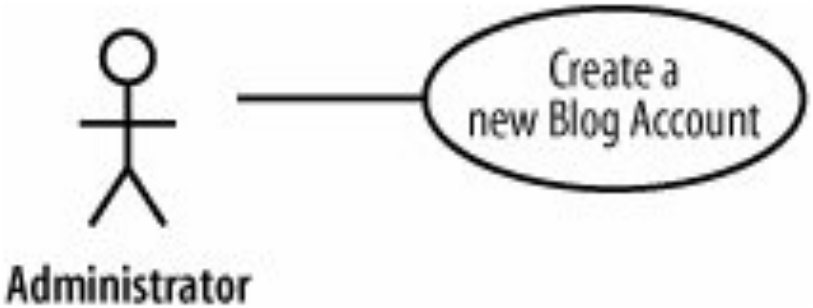
A use case in UML is drawn as an oval（橢圓） with a name that describes the interaction that it represents

What Makes a Good Use Case? 好的用例

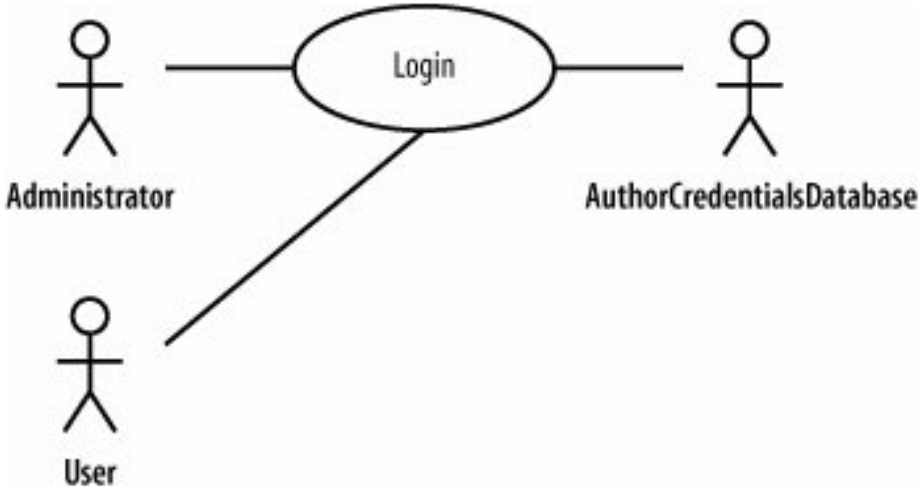
- Experience will help you determine when you have a good use case, but there is a rule of thumb that can be used to specify a use case: 经验法则
  - A use case is something that provides some measurable result to the user or an external system.
    - 一定有确定的结果
  - Any piece of system behavior that meets this simple test is likely to be a good candidate for a use case.

2.1.3. Communication Lines 会话线

- A communication line connects an actor and a use case to show the actor participating in the use case.
  - 连接参与者与用例
- Sometimes UML diagrams will have communication lines with navigability; 连接方向
  - for example, a diagram with an arrow at one end will show the flow of information between the actor and the use case, or show who starts the use case.
  - Although this notation is not really a crime in UML terms, it's not a very good use of communication lines
  - 最好不要加方向



A communication line joins the Administrator actor to the "Create a new Blog Account" use case; the Administrator is involved in the interaction that the use case represents.

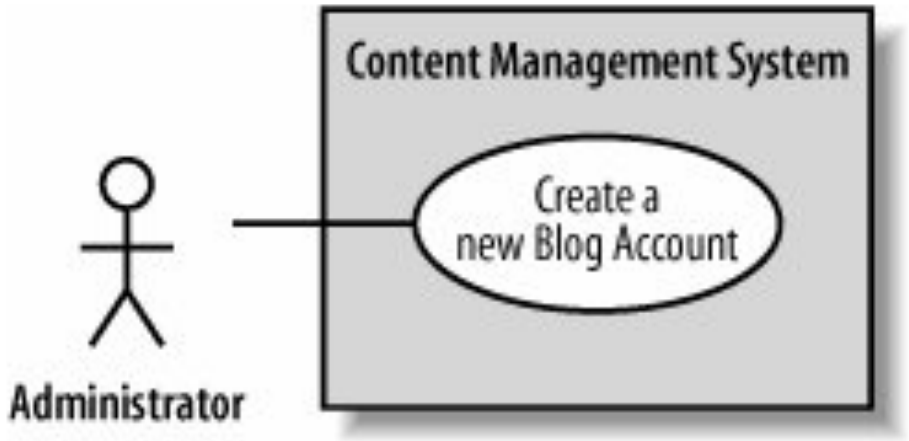


The login use case interacts with three actors during its execution

Use navigation or not 是否使用方向

- The purpose of a communication line is to show that an actor is simply involved in a use case, not to imply an information exchange in any particular direction or that the actor starts the use case.
  - 参与者调用用例，不涉及交换的信息
- That type of information is contained within a use case's detailed description, therefore it doesn't make sense to apply navigation to communication lines.
  - 用例描述包含了所有信息

2.1.4. System Boundaries 系统边界



The Administrator actor is located outside of the CMS, explicitly showing that the system boundary box use cases must fall within the system boundary box, since it doesn't make sense to have a use case outside of your system's boundary

2.1.5. Use Case Descriptions 用例描述

- A diagram showing your use cases and actors may be a nice starting point, but it does not provide enough detail for your system designers to actually understand exactly how the system's concerns will be met. 不够详细
  - How can a system designer understand who the most important actor is from the use case notation alone?
  - What steps are involved in the use case? 步骤
- The best way to express this important information is in the form of a text-based description, every use case should be accompanied by one.
  - 一般需要添加文字性描述

Use case specification 用例规约

Related Requirements	关联需求
Goal In Context	场景目标
Preconditions	前置条件
Successful End Condition	正常结束条件
Failed End Condition	异常结束条件
Primary Actors	主参与者
Secondary Actors	次参与者
Trigger	触发器
Main Flow	基本流程
Extensions	扩展流程

A complete use case description for the "Create a new Blog Account" use case

Use case name	**Create a new Blog Account**	
Related Requirements	Requirement A.1.	
Goal In Context	A new or existing author requests a new blog account from the `Administrator`.	
Preconditions	The system is limited to recognized authors and so the author needs to have appropriate proof of identity.	
Successful End Condition	A new blog account is created for the author.	
Failed End Condition	The application for a new blog account is rejected.	
Primary Actors	Administrator.	
	Secondary Actors	Author Credentials Database.
	Trigger	The `Administrator` asks the CMS to create `a new blog account`.
Main Flow	Step	Action
	1	The `Administrator` asks the system to create a new blog account.
	2	The `Administrator` selects an account type.
	3	The `Administrator` enters the author's details.
	4	The author's details are verified using the Author Credentials Database.
	5	The new blog account is created.
	6	A summary of the new blog account's details are emailed to the author.
Extensions	Step	Branching Action
	4.1	The Author Credentials `Database` does not verify the author's details.
	4.2	The author's new blog account application is rejected.

Question ?

- How Many Use Cases Should Your Model Have?

Answer

- There is no set rule for the number of use cases that your use case model should contain for a given system.
- The number of use cases depends on the of the jobs that your system has to do according to the requirements.
  - 取决于需求
- It is more important that you have the right use cases, rather than worrying about the amount you have.
  - 关键在于获取正确的需求
- As with most things in system modeling, the best way to get your use cases right is to get used to applying them; experience will teach you what is right for your own systems.
  - 通过建模经验来选择用例数量和粒度

2.2. Use Case Relationships 用例关系

2.2.1. The <<include>> Relationship 包含

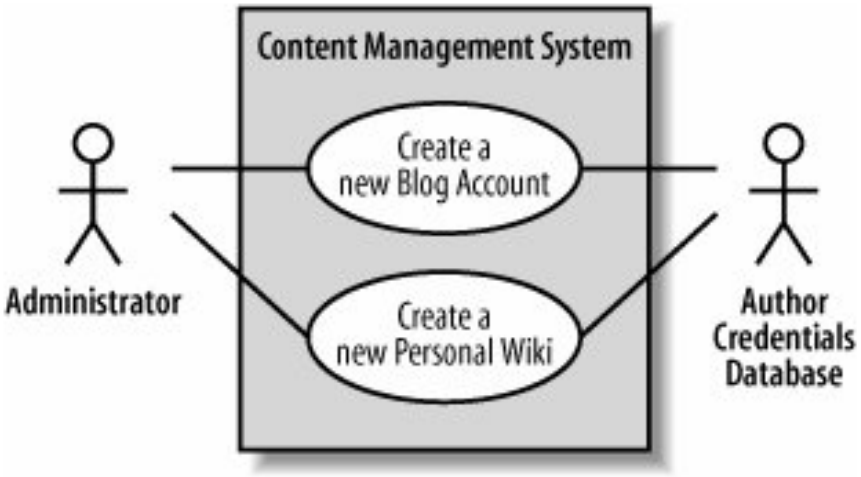
- Relationships between use cases are more about breaking your system's behavior into manageable chunks than adding anything new to your system.
  - 关系可用于划分系统行为
- The purpose of use case relationships is to provide your system's designers with some architectural guidance so they can efficiently break down the system's concerns into manageable pieces within the detailed system design.
  - 目的是指导系统设计人员一些架构线索
  - 将系统划分为一定的模块

Example: Wiki of Blog

- In addition to blogs, a CMS can have any number of means for working with its content. One popular mechanism for maintaining documents is by creating a Wiki. Wikis allow online authors to create, edit, and link together web pages to create a web of related content, or a Wiki-web. A great example of a Wiki is available at http://www.Wikipedia.org. 创建维基帐号

Requirement A.2

- The content management system shall allow an administrator to create a new personal Wiki, provided the personal details of the applying author are verified using the Author Credentials Database. 需求2：创建Wiki帐号



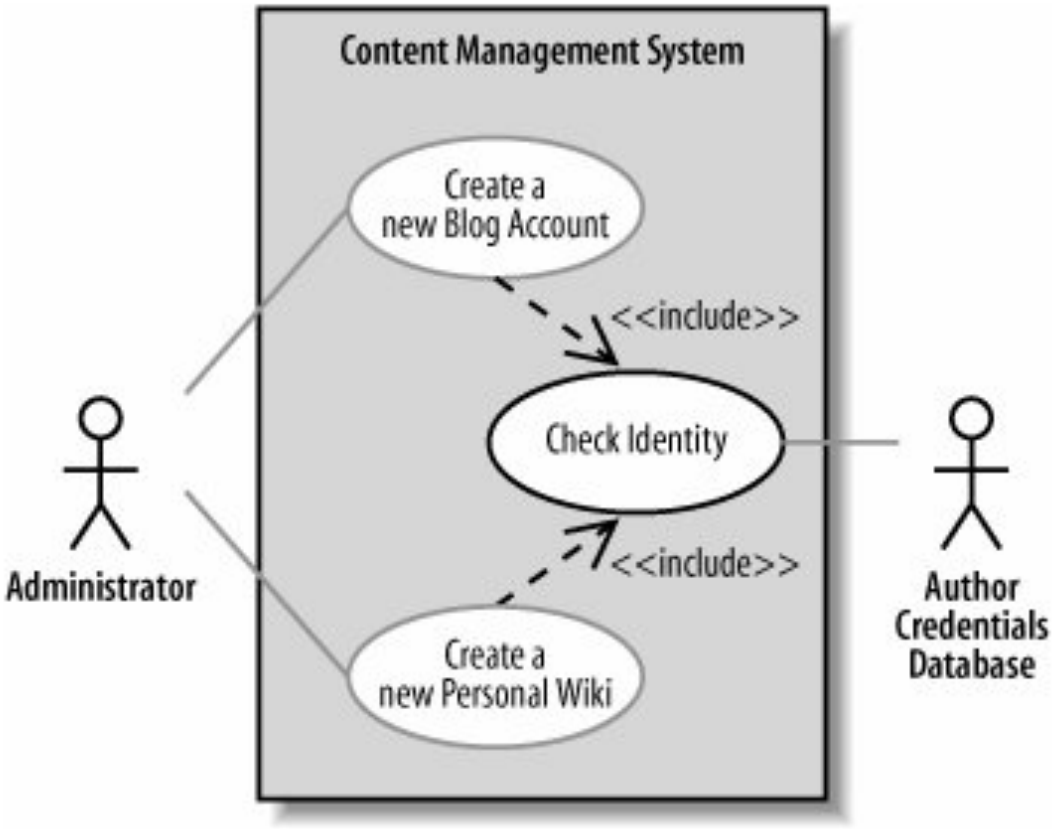
A new requirement can often mean a new use case for the system, although it's not always a one-to-one mapping

The detailed description for the "Create a new Personal Wiki" use case

Use case name	**Create a new Personal Wiki**	
Related Requirements	Requirement A.2.	
Goal In Context	A new or existing author requests a new personal Wiki from the Administrator.	
Preconditions	The author has appropriate proof of identity.	
Successful End Condition	A new personal Wiki is created for the author.	
Failed End Condition	The application for a new personal Wiki is rejected.	
Primary Actors	Administrator.	
Secondary Actors	Author Credentials Database.	
Trigger	The Administrator asks the CMS to create a new personal Wiki.	
Main Flow	Step	Action
	1	The Administrator asks the system to create a new personal Wiki.
	2	The Administrator enters the author's details.
	3	The author's details are verified using the Author Credentials Database.
	4	The new personal Wiki is created.
	5	A summary of the new personal Wiki's details are emailed to the author.
Extensions	Step	Branching Action
	3.1	The Author Credentials Database does not verify the author's details.
	3.2	The author's new personal Wiki application is rejected.

Problem


Both check the applicant's credentials
两个用例都检查帐号
Redundancy
冗余
重叠



The <<include>> relationship supports reuse between use cases.

包含关系可以实现用例的重用

Showing <<include>> in a use case description using Included Cases and include::<use case name>

Use case name	**Create a new Blog Account**	
Related Requirements	Requirement A.1.	
Goal In Context	A new or existing author requests a new blog account from the Administrator.	
Preconditions	The author has appropriate proof of identity.	
Successful End Condition	A new blog account is created for the author.	
Failed End Condition	The application for a new blog account is rejected.	
Primary Actors	Administrator	
Secondary Actors	**None**	
Trigger	The Administrator asks the CMS to create a new blog account.	
Included Cases	**Check Identity**	
Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects an account type.
	3	The Administrator enters the author's details.
	4 **include::Check Identity**	The author's details are checked.
	5	The new account is created.
	6	A summary of the new blog account's details are emailed to the author.

The Create a new Personal Wiki use case description also gets a makeover

Use case name	**Create a new Personal Wiki**	
Related Requirements	Requirement A.2	
Goal In Context	A new or existing author requests a new personal Wiki from the Administrator.	
Preconditions	The author has appropriate proof of identity.	
Successful End Condition	A new personal Wiki is created for the author.	
Failed End Condition	The application for a new personal Wiki is rejected.	
Primary Actors	Administrator	
Secondary Actors	**None**	
Trigger	The Administrator asks the CMS to create a new personal Wiki.	
Included Cases	**Check Identity**	
Main Flow	Step	Action
	1	The Administrator asks the system to create a new personal Wiki.
	2	The Administrator enters the author's details.
	3 **include::Check Identity**	The author's details are checked.
	5	The new personal Wiki is created.
	6	A summary of the new personal Wiki's details are emailed to the author.

The Check Identity use case description contains the reusable steps

Use case name	**Check Identity**	
Related Requirements	Requirement A.1, Requirement A.2.	
Goal In Context	An author's details need to be checked and verified as accurate.	
Preconditions	The author being checked has appropriate proof of identity.	
Successful End Condition	The details are verified.	
Failed End Condition	The details are not verified.	
Primary Actors	Author Credentials Database.	
Secondary Actors	None.	
Trigger	An author's credentials are provided to the system for verification.	
Main Flow	Step	Action
	1	The details are provided to the system.
	2	The Author Credentials Database verifies the details.
	3	The details are returned as verified by the Author Credentials Database.
Extensions	Step	Branching Action
	2.1	The Author Credentials Database does not verify the details.
	2.2	The details are returned as unverified.

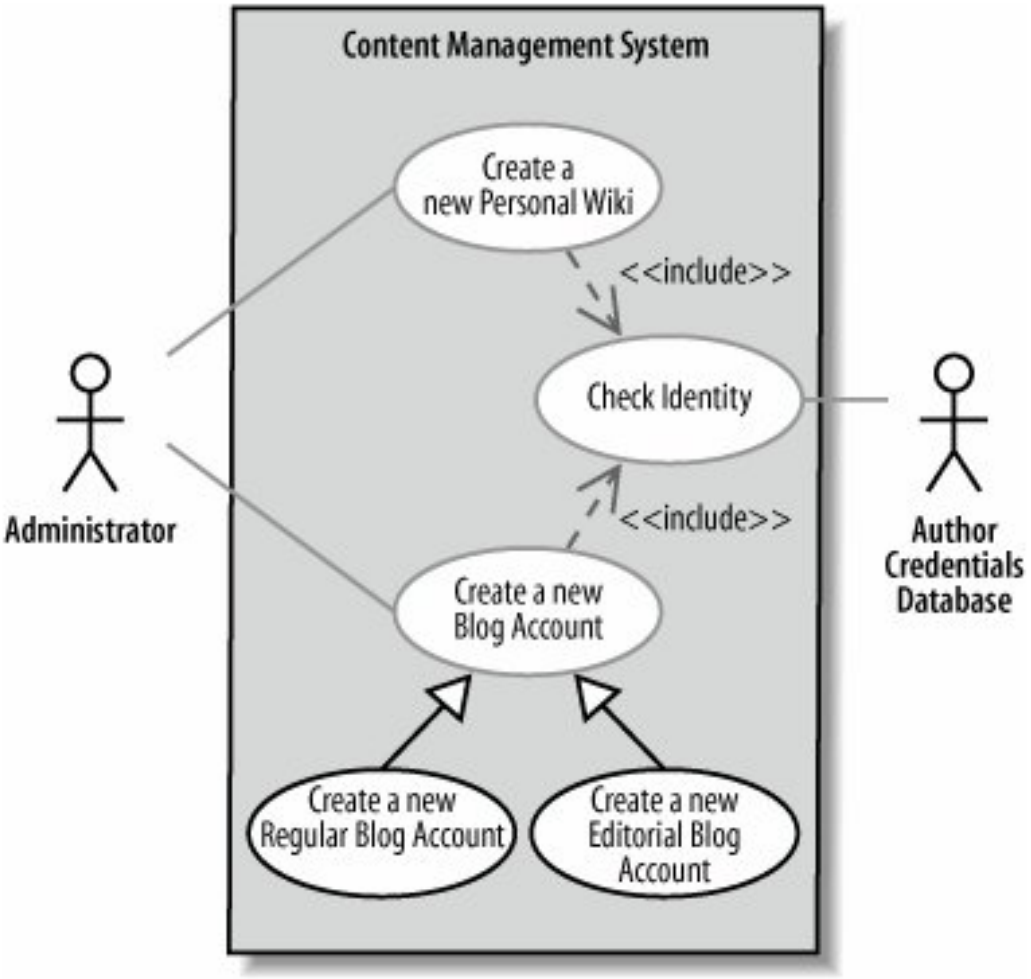
Benefits 好处

- Reuse using <<include>> removes the need for tedious cut-and-paste operations between use case descriptions, since updates are made in only one place instead of every use case.
  - 重用避免用例描述的复制
- The <<include>> relationship gives you a good indication at system design time that the implementation of Check Identity will need to be a reusable part of your system.
  - 形成一个可重用的系统组件

2.2.2. Special Cases 特殊情况

- Sometimes you'll come across a use case whose behavior, when you start to analyze it more carefully, can be applied to several different cases, but with small changes.
  - 只需要小小的修改就可以应用到另一种特殊情况中
- Unlike the <<include>> relationship, which allows you to reuse a small subset of behavior, this is applying a use case with small changes for a collection of specific situations.
  - 包含是重用一小部分，继承由重复大部分的用例
- In object-oriented terms, you potentially have a number of specialized cases of a generalized use case.
  - 用例泛化

Two types of blog account, regular and editorial, can be created by the Management System



You can show that a use case is a special case of a more general use case within the detailed description using the Base Use Cases field

Use case name	**Create a new Editorial Blog Account**	
Related Requirements	Requirement A.1.	
Goal In Context	A new or existing author requests a new **editorial** blog account from the Administrator .	
Preconditions	The author has appropriate proof of identity.	
Successful End Condition	A new **editorial** blog account is created for the author.	
Failed End Condition	The application for a new **editorial** blog account is rejected.	
Primary Actors	Administrator.	
Secondary Actors	**None**.	
Trigger	The Administrator asks the CMS to create a new **editorial** account that will allow an author to edit entries in a set of blogs.	
Base Use Cases	**Create a new Blog Account**	
Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	**The Administrator selects the editorial account type**.
	3	The Administrator enters the author's details.
	4	**The Administrator selects the blogs that the account is to have editorial rights over**.
	5 include::Check Identity	The author's details are checked.
	6	The new editorial account is created.
	7	A summary of the new editorial account's details are emailed to the author.
Extensions	Step	Branching Action
	5.1	**The author is not allowed to edit the indicated blogs**.
	5.2	**The editorial blog account application is rejected**.
	5.3	**The application rejection is recorded as part of the author's history**.

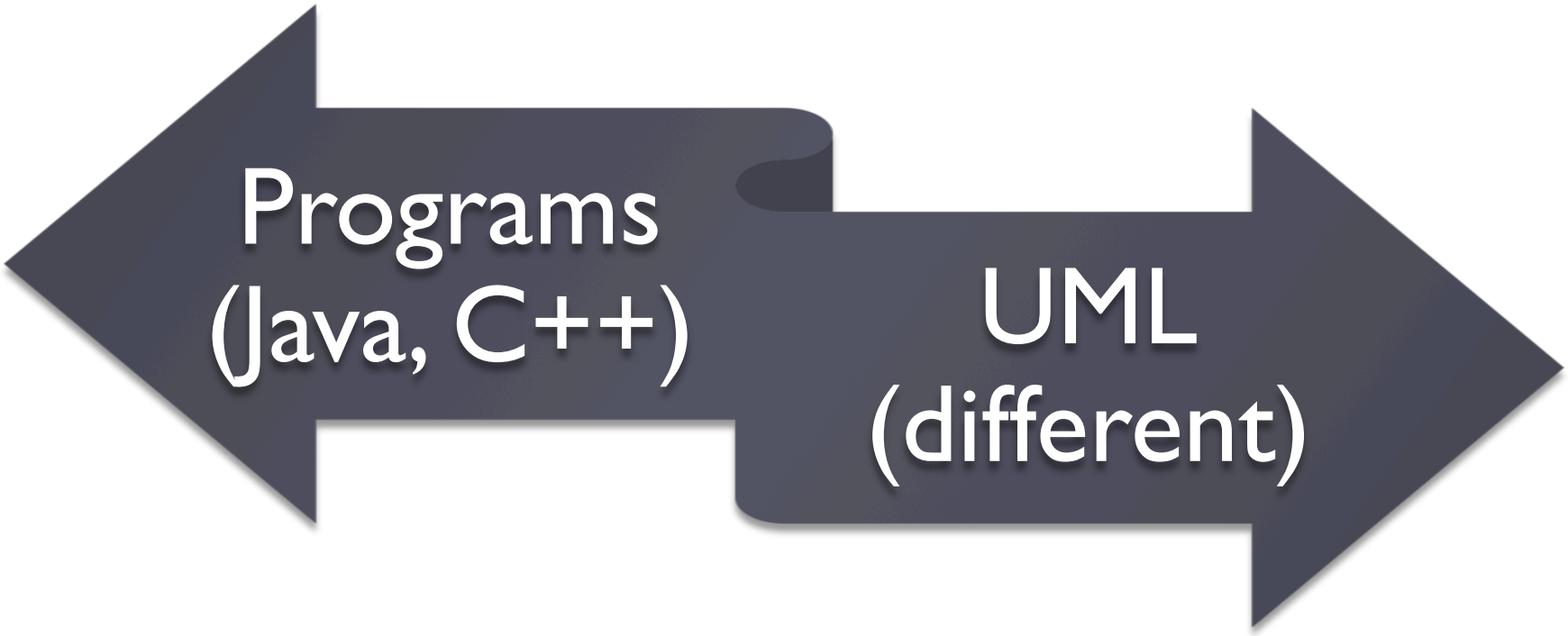
Note

- But be carefully using inheritance, you are effectively saying that every step in the general use case must occur in the specialized use cases. 重用包含所有的步骤
- Also, every relationship that the general use case has with external actors or use cases must also make sense in the more specialized use cases. 一般用例的行为一定会发生
- If you really don't want your more specific use case to do everything that the general use case describes, then don't use generalization. 大部分时候避免使用
  - Instead, you might want to consider using either the <<include>> relationship shown in the previous section or the <<extend>> relationship coming up in the next section.

2.2.3. The <<extend>> Relationship

The <<extend>> use case relationship looks a bit like the <<include>> relationship, but that's where the similarities end

Understanding extend 不同于Java



Extend in UML

- The designers of UML 2.0 took a very different view as to the meaning of <<extend>> between use cases.
  - UML的扩展有不同的含义
- They wanted a means for you to show that a use case might completely reuse another use case's behavior, similar to the <<include>> relationship, but that this reuse was optional and dependent either on a runtime or system implementation decision.
  - 完全重用所扩展用例的行为，但是并非一定发生

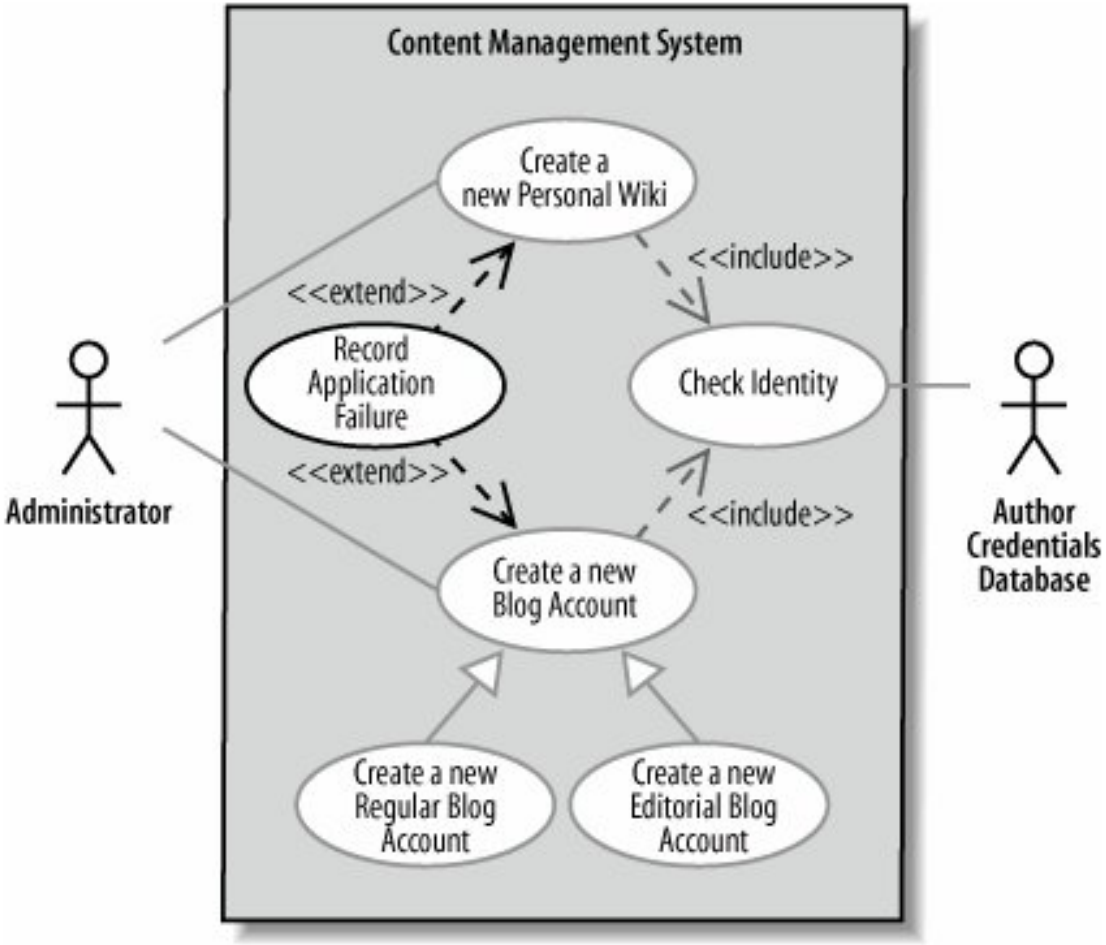
Example

- From the CMS example, the Create a new Blog Account use case might want to record that a new author applied for an account and was rejected, adding this information to the author's application history.
  - 创建帐号时需要记录创建失败的次数
- Extra steps can be added to the Create a new Blog Account use case's description to show this optional behavior.

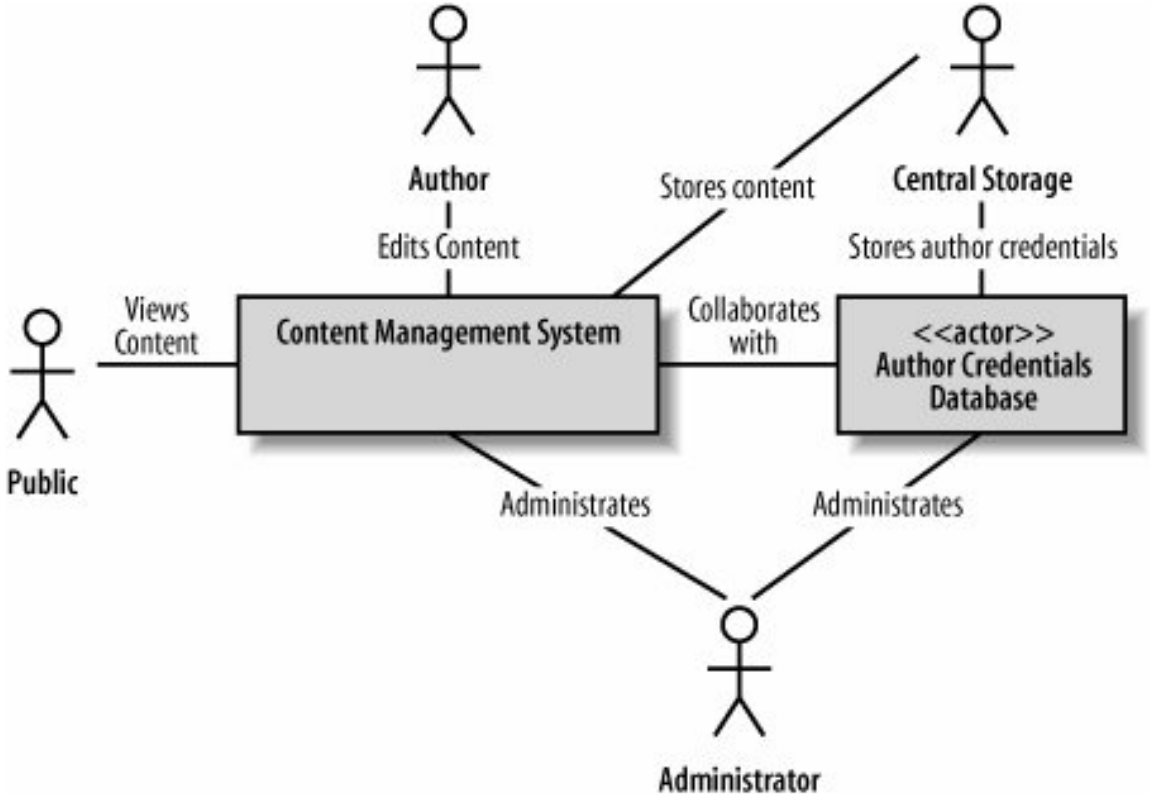
Behavior that is a candidate for <<extend>> relationship reuse can usually be found in the Extensions section of a use case description

Use case name	**Create a new Blog Account**	
Related Requirements	Requirement A.1.	
Goal In Context	A new or existing author requests a new blog account from the Administrator.	
Preconditions	The author has appropriate proof of identity.	
Successful End Condition	A new blog account is created for the author.	
Failed End Condition	The application for a new blog account is rejected.	
Primary Actors	Administrator.	
Secondary Actors	None.	
Trigger	The Administrator asks the CMS to create a new blog account.	
Included Cases	Check Identity	
Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects an account type.
	3	The Administrator enters the author's details.
	4 include::Check Identity	The author's details are checked.
	5	The new account is created.
	6	A summary of the new blog account's details are emailed to the author.
Extensions	Step	Branching Action
	4.1	The author is not allowed to create a new blog.
	4.2	The blog account application is rejected.
	4.3	**The application rejection is recorded as part of the author's history.**

The <<extend>> relationship comes into play to show that both the "Create a new Personal Wiki" and "Create a new Blog Account" use cases might occasionally share the application rejection recording behavior



2.3. Use Case Overview Diagrams



The CMS's context as shown on a Use Case Overview diagram

Summary

- 2. Modeling requirements
  - 2.1. Capturing a System Requirement
    - actor ☆☆☆☆☆
    - use case ☆☆☆☆☆
  - 2.2. Use Case Relationships
    - <<include>> ☆☆☆☆☆
    - <<extend>> ☆☆☆☆☆
  - 2.3. Use Case Overview Diagrams

Next

- 3. Modeling System Workflows: Activity Diagrams
  - 3.1. Activity Diagram Essentials
  - 3.2. Activities and Actions
  - 3.3. Decisions and Merges
  - 3.4. Doing Multiple Tasks at the Same Time
  - 3.5. Time Events
  - 3.6. Calling Other Activities
  - 3.7. Objects
  - 3.8. Sending and Receiving Signals
  - 3.9. Starting an Activity
  - 3.10. Ending Activities and Flows
  - 3.11. Partitions (or Swimlanes)
  - 3.12. Managing Complex Activity Diagrams

See you …