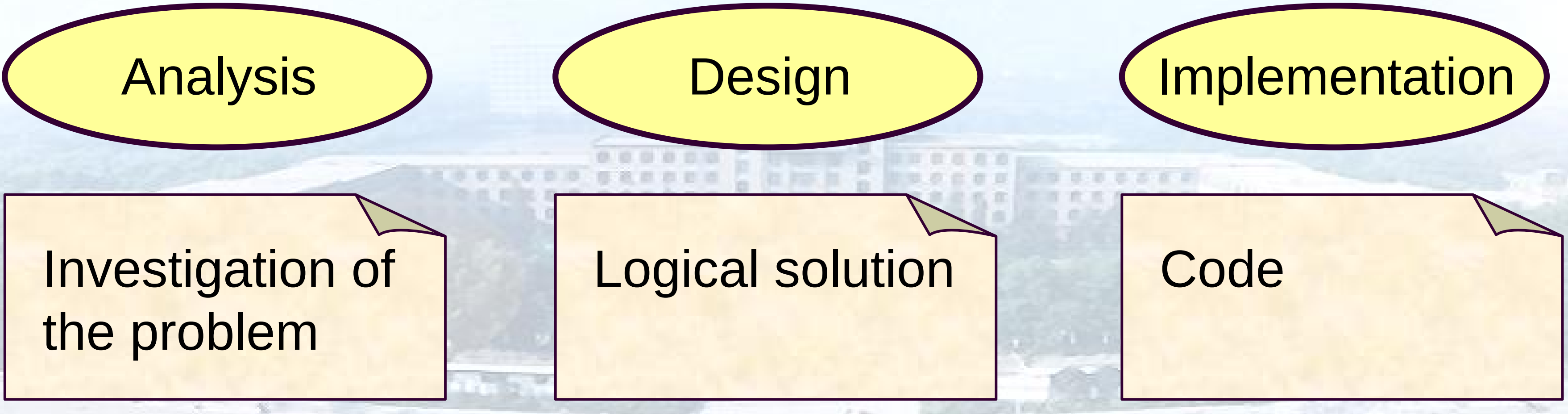
4.12 面向对象系统分析

4.12.1 面向对象分析与设计的基本概念

- Object-oriented analysis
  - Investigation that is object-centric.
- Object-oriented design
  - Solution in terms of interacting software objects
- Object-oriented programming
  - Coding in an object-oriented programming language.



Analysis

Design

Implementation

Investigation of the problem

Logical solution

Code

4.12 面向对象系统分析

4.12.1 面向对象分析与设计的基本概念

- Requirements Analysis:
  - discover and express requirements in use cases. Use Case is a textual description of a business process in the system.
- Domain Analysis:
  - develop a conceptual model of the problem domain. This includes the things, the concepts, as well as the various roles people may take and the relationships between them.
- Design - Assignment of Responsibilities:
  - allocate tasks to software objects as well as roles people take, illustrated in interaction diagrams and logical class diagrams.

4.12 面向对象系统分析

4.12.1 面向对象分析与设计的基本概念

- Model-driven（模型驱动的）Analysis Approaches:
  - Structured Analysis
  - Information Engineering
  - Object-oriented Analysis
- Accelerated（加速的）Analysis Approaches:
  - Discovery Prototyping
  - Rapid Architecture Analysis

4.12 面向对象系统分析

4.12.1 面向对象分析与设计的基本概念

- Model-driven analysis emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems. Ultimately, the system model becomes the blueprint for designing and constructing an improved system.
  - A model is a representation of either reality or vision. Just as “a picture is worth a thousand words,” most models use pictures to represent the reality or vision.

4.12 面向对象系统分析

4.12.1 面向对象分析与设计的基本概念

- A functional requirement（功能性需求）is a description of activities and services a system must provide.
- A nonfunctional requirement（非功能性需求）is a description of other features, characteristics, and constraints that define a satisfactory system.

4.12 面向对象系统分析

4.12.2 功能性需求的建模

- 采用 **UML** 进行系统分析的第一步，是以 **Use case** 图为主要工具，对信息系统的功能性需求进行寻找、描述和整理。
- 功能性需求建模的结果以 **Use Case Model** 的形式表示。

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

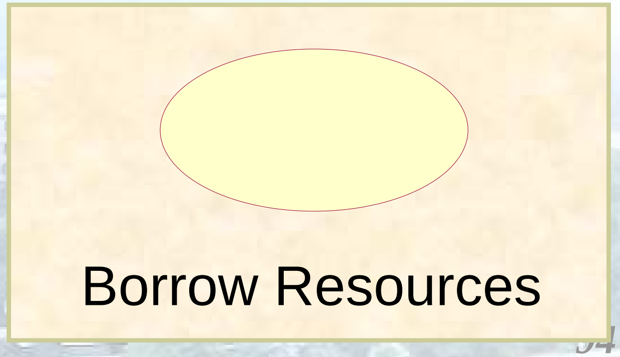
- A library lends books and videos (called resources).
- Different categories of books and videos have different periods of loan.
- System functions (business functions):

Ref #	Function
R 1.1	Record the loan of the resources.
R 1.2	Calculate the due date of loaned resources.
R 1.3	Generate borrowing reports.
R 1.4	Record new resources for loan.
R 1.5	Generate overdue loans reports.
...	...

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Use cases:
  - Describe the sequence of events of an actor using a system to complete a process.
  - Represent business or domain processes.
  - Are a narrative（叙述性的） description of a business process.
  - Are expressed in structured prose（结构化的散文）.
  - Are not directly related to object technology.



4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- A high-level use case:
  - High level use cases briefly describe major processes in the organization.

Name: Borrow Resources

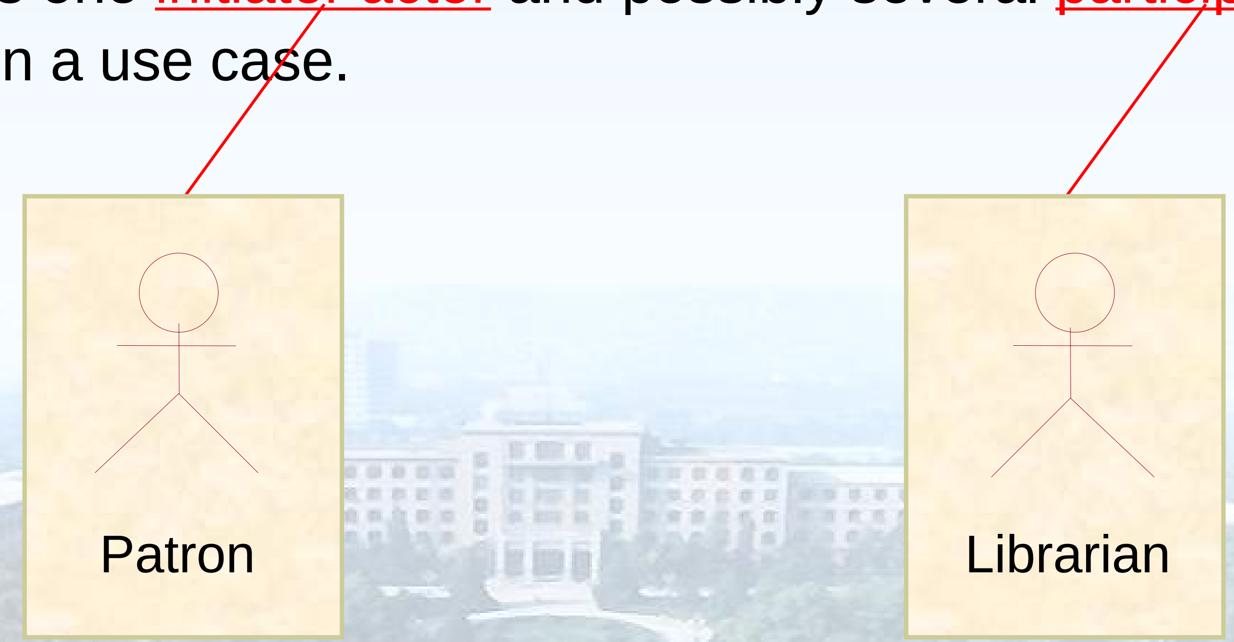
Actors: Patron (initiator（该用例的启动者）), Librarian

Description: The use case begins when the Patron arrives at the check-out with books and videos to borrow and submit them to the Librarian, who records the resources borrowed. The Patron then leaves with the resources.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Actors:
  - An actor in a use case is an external agent that uses or interacts with the system.
  - There is one initiator actor and possibly several participating actors in a use case.



4.12 面向对象系统分析

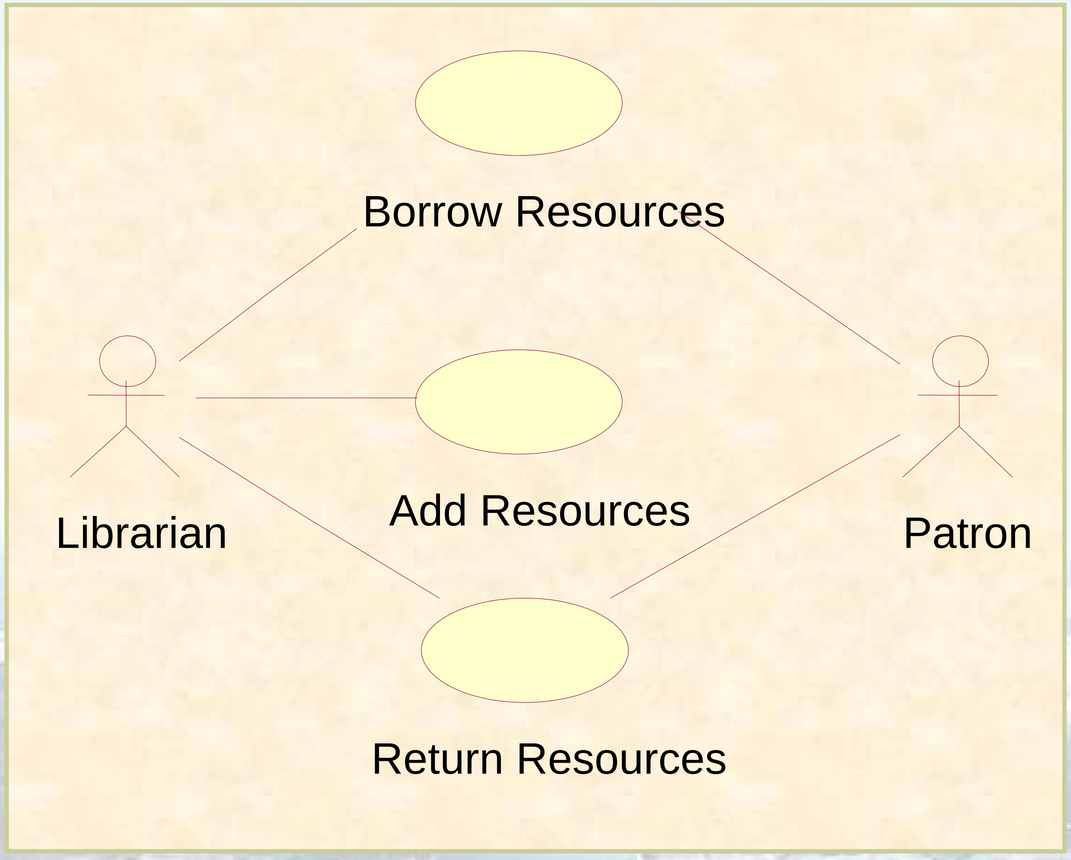
4.12.3 范例：一个图书馆信息系统的需求建模

- Actors can be:
  - Roles of humans.
    - Example: *A Patron.*
  - Other computer systems.
    - Example: *The Visa network.*
  - Inanimate physical objects, such as electro-mechanical devices.
    - Example: *A robot.*

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Use case diagrams:



4.12 面向对象系统分析

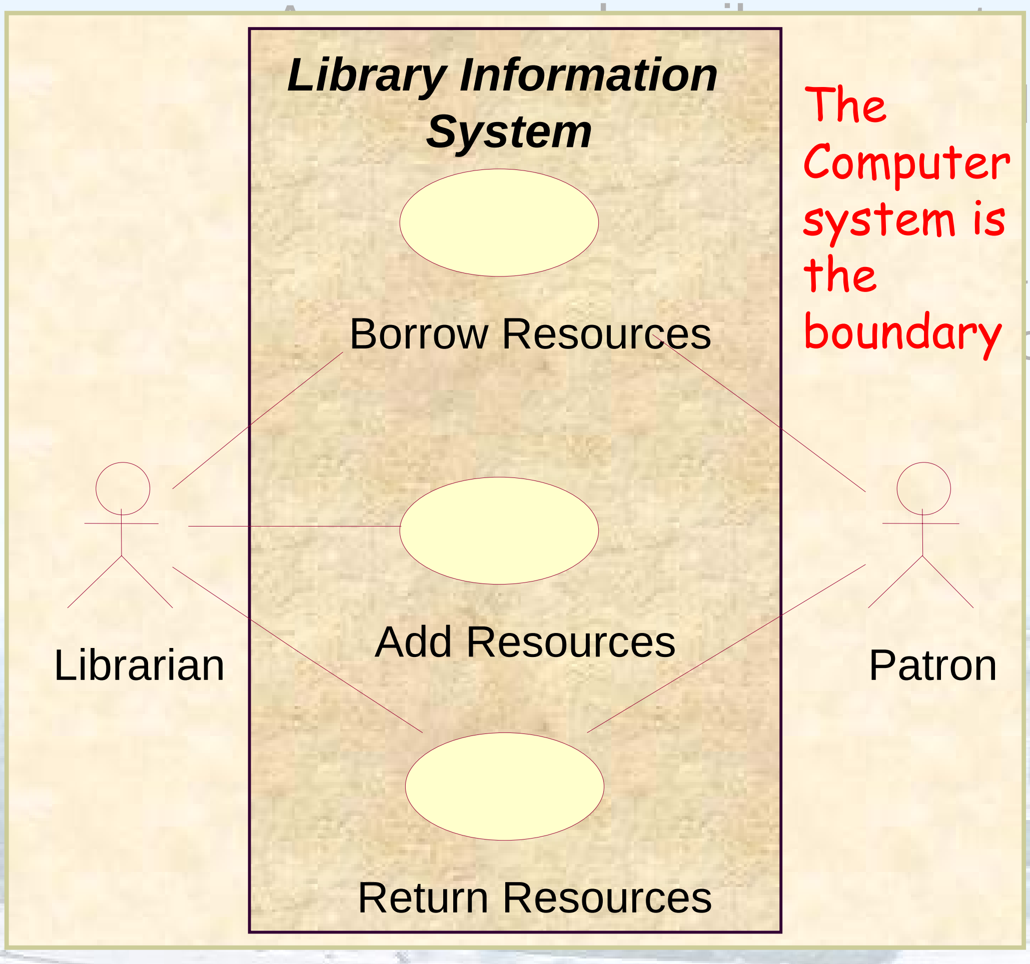
4.12.3 范例：一个图书馆信息系统的需求建模

- Choosing the system boundary:
  - A use case describes events upon a system.
    - This implies a system boundary is chosen.
  - Typically boundaries:
    - Hardware / software boundary of a device or computer system.
    - Department of an organization.
    - Entire organization.

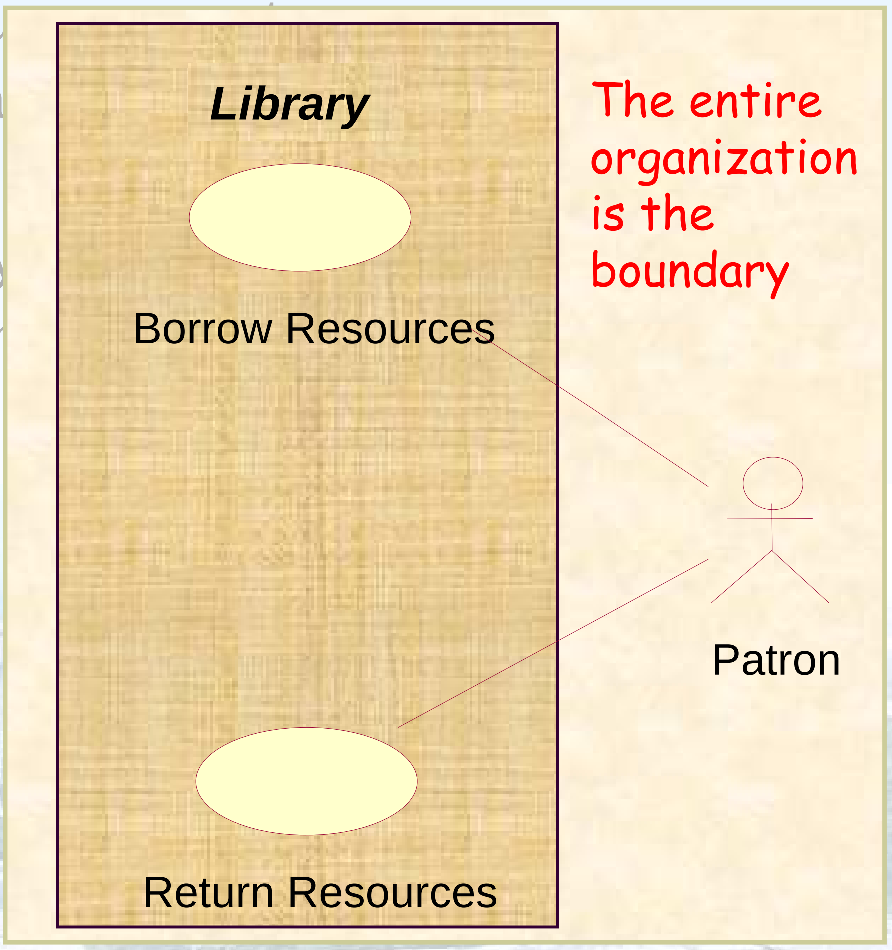
4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Choosing the system boundary:



The Computer system is the boundary



The entire organization is the boundary

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Identifying use cases:
  1. Define the system boundary.
  2. Identify actors.
     - External entities participating in domain processes.
     - Examples: *customers, clerks, managers, technicians, external computer systems*.
  3. For each actor, determine:
     - What fundamentally different processes / actions does it participate in?
     - What initial event does an actor perform to start a process?
  - Each fundamentally different process of each actor is a use case.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

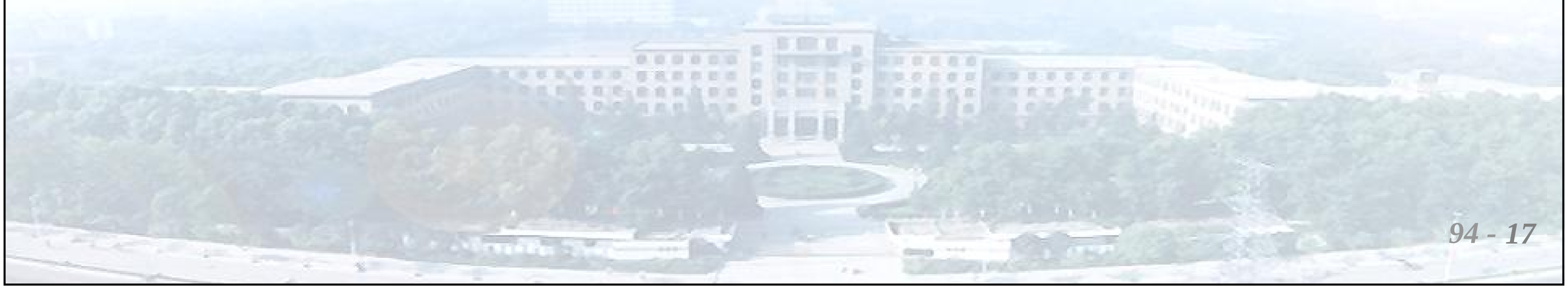
- A common error in identifying use cases is to represent individual steps, operations, or transactions as use cases.
  - A use case is a relatively large end-to-end process that typically includes many steps or transactions; it is not normally an individual step or activity in a process.
  - It is possible to break down portions of a use case into subordinate use cases (called abstract use cases), but this is nor the norm.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- In practice, use cases are categorized in many dimensions:

	CATEGORY	NAME
抽象的程度	Detail level	High level / Expanded
分解的程度	Task level	Super / Sub (Abstract)
重要性	Importance	Primary / Secondary
开发阶段	Abstraction	Essential / Real



4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- In practice, use cases are categorized in many dimensions:

抽象的程度

CATEGORY	NAME
Detail level	High level / Expanded

分解的程

重要

开发阶

HIGH-LEVEL	EXPANDED
General, brief description of processes	Detailed, showing exact order of events and alternatives

4.12 面向对象系统分析

SUPERPROCESS	SUBPROCESS (ABSTRACT)
A broad task use case that is subdivided into subprocesses.	A subtask of a superprocess use case. Also known as an abstract use case （现实中并非单独存在的用例）

分解的程度	Task level	Super / Sub (Abstract)
重要性	Importance	Primary / Secondary
开发阶段	Abstraction	Essential / Real

- The special includes relationship can relate some use cases (not all).
- If A includes B, we can say A is a superprocess and B is a subprocess.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- In practice, use cases are categorized in many dimensions:

	CATEGORY	NAME
抽象的程度	Detail level	High level / Expanded
分解的程度	Task level	Super / Sub (Abstract)
重要性	Importance	Primary / Secondary
开发阶		

PRIMARY	SECONDARY
Describes a major, common process	Describes a rare, unusual, or exceptional process.

4.12 面向对象系统分析

4.12.

- In pr :

ESSENTIAL	REAL
The essence of the process. Analysis-oriented. Expresses process (relatively) independent of a hardware / software solution.	Concrete, design-oriented. Expressed in terms of the solution – screen shots of windows, entry into input fields, and so forth.

抽象的程

分解的程

重要

开发阶段	Abstraction	Essential / Real

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Example:

Essential	Real
The Librarian records the Call number.	The Librarian uses the laser wand to scan the bar code for the call number, which is transmitted to the computer.
The AccountHolder identifies himself to the ATM.	The AccountHolder inserts the card into the ATM card reader. He is prompted to enter his PIN (see screen shot 4), which he inputs with a numeric keypad.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Sample high-level primary use cases:

Name: Borrow Resources

Actors: Patron (initiator), Librarian

Description: The use case begins when the Patron arrives at the check-out with books and videos to borrow and submit them to the Librarian, who records the resources borrowed. The Patron then leaves with the resources.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Sample high-level primary use cases:

Name: Add Resources

Actors: Librarian

Description: The use case begins when the Librarian receives new resources (books and videos) to add to the catalog. The title, call number, and other information are recorded. Then the resources are placed on a shelf organized by resource type and call numbers.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Other possible use cases:
  - Return a Resource.
  - Delete a Resource.
  - Notify Overdue Patrons.
  - Collect Fines.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Expanded format use cases:
  - Describe the use case in greater detail.
  - Can be written essential or real.
  - Have the following components:
    - Name
      - *Starts with a verb.*
    - Description
      - *From the high-level use case.*
    - Actors
      - *Initiator and participants from the high-level use case.*
    - Type
      - *If decomposed, then super / sub (abstract).*
      - *Also, primary / secondary, and essential / real.*

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Expanded format use cases:
  - Have the following components *(continued)*:
    - Cross-references
      - *Related use cases and system functions.*
    - Preconditions
      - *Assumptions that must hold true.*
    - Typical course of events
      - *Most important section describes regular flow of events.*
    - Alternatives
      - *Exceptional alternatives that might arise.*

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Expanded use cases – Example:

Name:	Borrow Resources.
Description:	(As in high-level use case)
Actors:	Patron (initiator), Librarian
Type:	Primary, essential
Cross-references:	System functions R1.1, R1.2, R1.3
Preconditions:	Patron has a library card.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Typical course of events:

Actor Action	System Response
1. This use case begins when the Patron submits resources (books and videos) for borrowing.	
2. The Librarian takes the Patron’s library card and resources.	
3. The Librarian checks the card status by recording Patron’s Library ID.	4. Indicates valid membership.
5. The Librarian records the call number of each resources.	6. The due date is calculated and the loan is recorded.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Typical course of events *(continued)* :

Actor Action	System Response
7. On completion of entry, the Librarian requests a borrowing report.	8. A borrowing report is generated.
9. The Librarian Transfers the report to the Patron. 10. The Patron leaves with the resources and report.	

- Alternatives:
  - Alternative at 4: If the Patron card is invalid, do not allow borrowing.
  - Alternative at 5: If the Resource missing a call number, set it aside for repair and do not allow borrowing.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- What to describe in a use case:
  - Emphasize events between actors and the system without decomposing the internal processing of the system.
  - Orient toward things that the actors can perceive（感知）.
  - Discussion of internal processing or design decision is permissible, but should be done with knowledge of the consequences.
    - Example: *Early design commitments*（承诺）.

4.12.3 范例：一个图书馆信息系统的需求建模

- Use case miscellany（杂记）:
  - The first line of a use case course of events should describe the event that starts the use case.
    - Example: *This use case begins when the <actor> <generates an input event>*.
  - Start the use case name with a verb
    - *Purchase …*
    - *Borrow ...*

4.12 面向对象系统分析

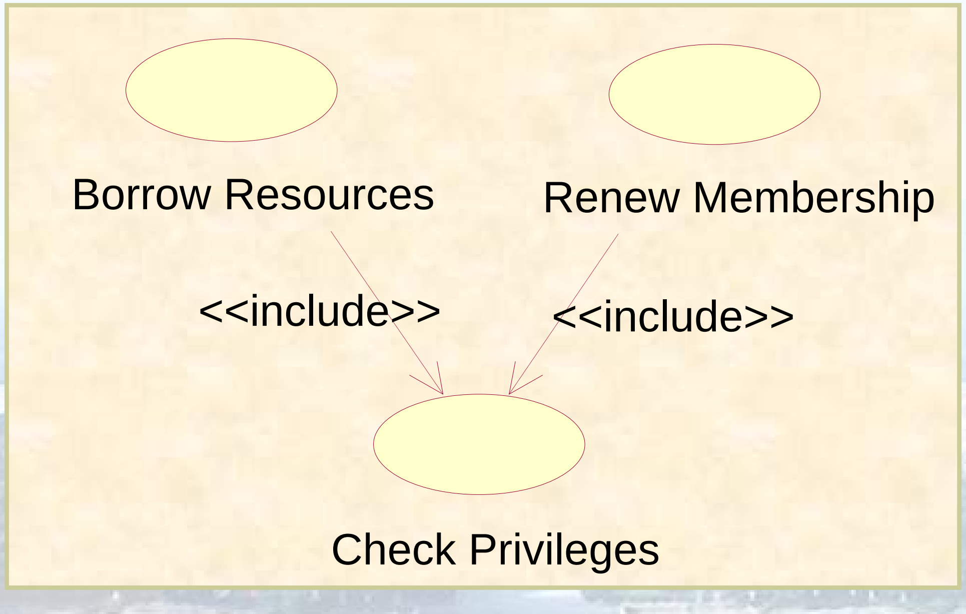
4.12.3 范例：一个图书馆信息系统的需求建模

- Relating use cases:
  - Use cases can be organized and related to simplify their descriptions.
  - Three associations are:
    - Includes.
    - Extends.
    - Generalization-specialization.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Includes:
  - Includes: “The insertion of additional behavior into a base use case that explicitly describes the insertion.”
    - Used to factor out a shared subprocess.



4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Includes *(continued)*:
  - When a super-task use case (base use case) uses a subtask use case (an abstract use case).
    - An abstract use case is a factored-out subtask that does not stand on its own as a complete, separate process.
  - In the use case text, the base use case explicitly initiates the abstract use case.
    - “initiates” is a verb favored by some use case writers.

> *Use Case: Borrow Resources*
>
> *…*
>
> *Step 5. Initiate use case Check Library Privileges*
>
> *…*

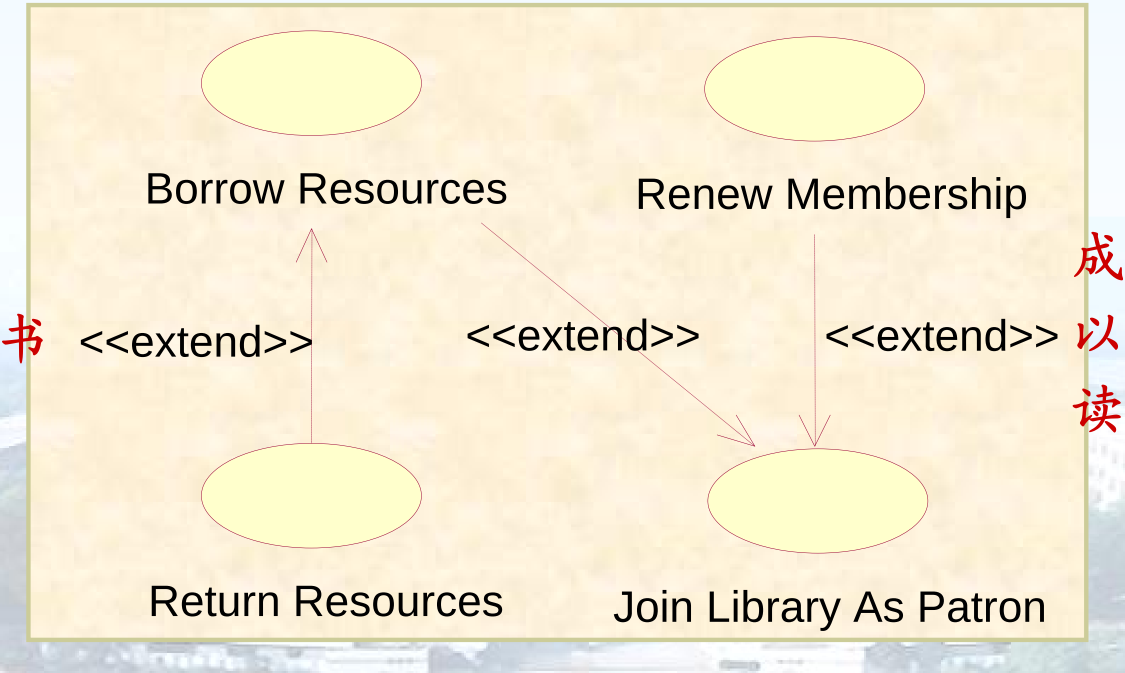
4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Includes *(continued)*:
  - When to use?
    - When several use cases share a common subflow that can be factored out.
    - When a use case is too complex and needs factoring into smaller, understandable chunks.

4.12.3 范例：一个图书馆信息系统的需求建模

- Extends:
  - extends: “The insertion of additional behavior into a base use case that does not know about it.”
  - Used to extend an existing self-complete story.



借书过程中可以还书

成为读者的过程中可以借书，也可以延续读者身份

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Extends *(continued)*:
  - When a second use case extends the story of the first use case.
    - *“Chapter 1 followed by Chapter 2.”*
  - The two use cases are complete on their own.
  - The first use case does not know about or refer to the second use case.

4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Extends *(continued)*:
  - The second use case can extend the first at its end or at any point - the extension point.
  - The second use case might unconditionally extend the first or only under some condition.
  - In the use case text of the second, it notes (in the Cross Reference section) that it extends the first.

Use Case: Return Resources

…

Cross-References: Extends use case Borrow Resources

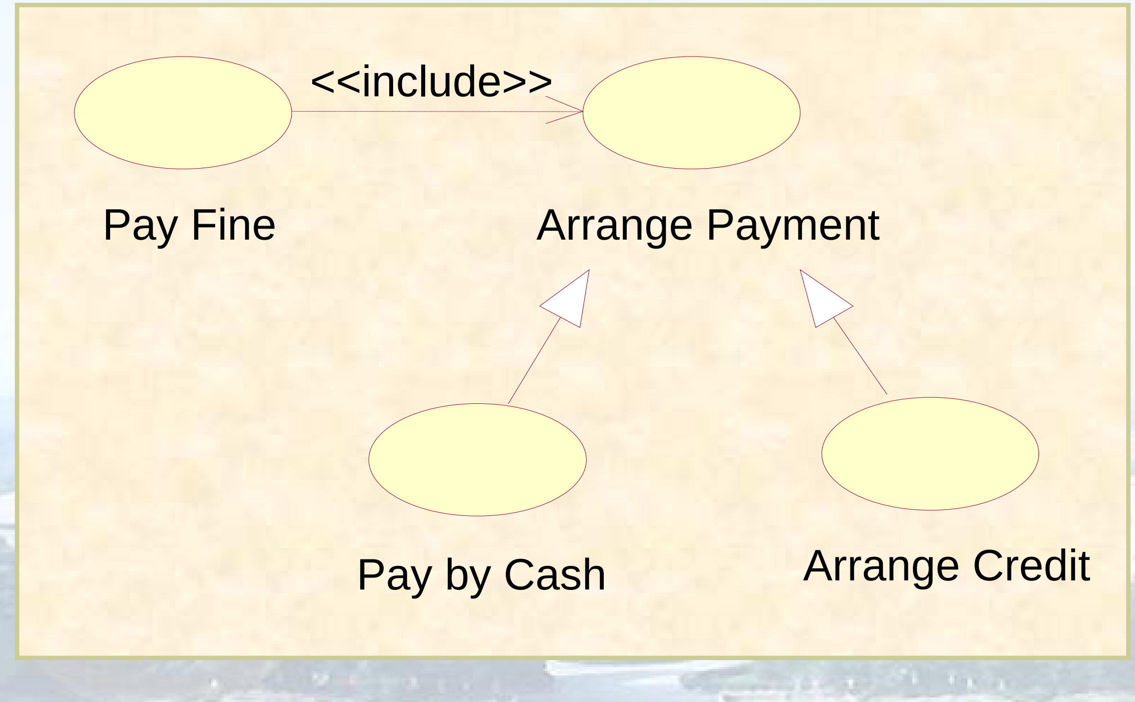
…

4.12.3 范例：一个图书馆信息系统的需求建模

- Extends *(continued)*:
  - When to use?
    - To add new possible postflows to complete use case.
    - To show a conditional or alternate subflow.

4.12.3 范例：一个图书馆信息系统的需求建模

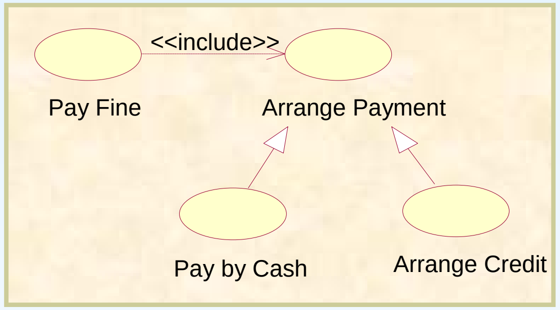
- Generalization-Specialization:
  - “… to add a more specific use case that inherits and adds features to (a more general use case).”



4.12 面向对象系统分析

4.12.3 范例：一个图书馆信息系统的需求建模

- Generalization-Specialization *(continued)*:
  - Example:



Use Case: Pay Fine
…
Step 3. Initiate use case Arrange Payment.
...

Use Case: Arrange Payment
…
Step 1. Collect payment
Step 2. Verify payment
Step 3. Record payment
...

Use Case: Pay by Cash
…
Cross References: Specializes Arrange Payment.
...

Use Case: Arrange Credit
…
Cross References: Specializes Arrange Payment.
...

4.12 面向对象系统分析

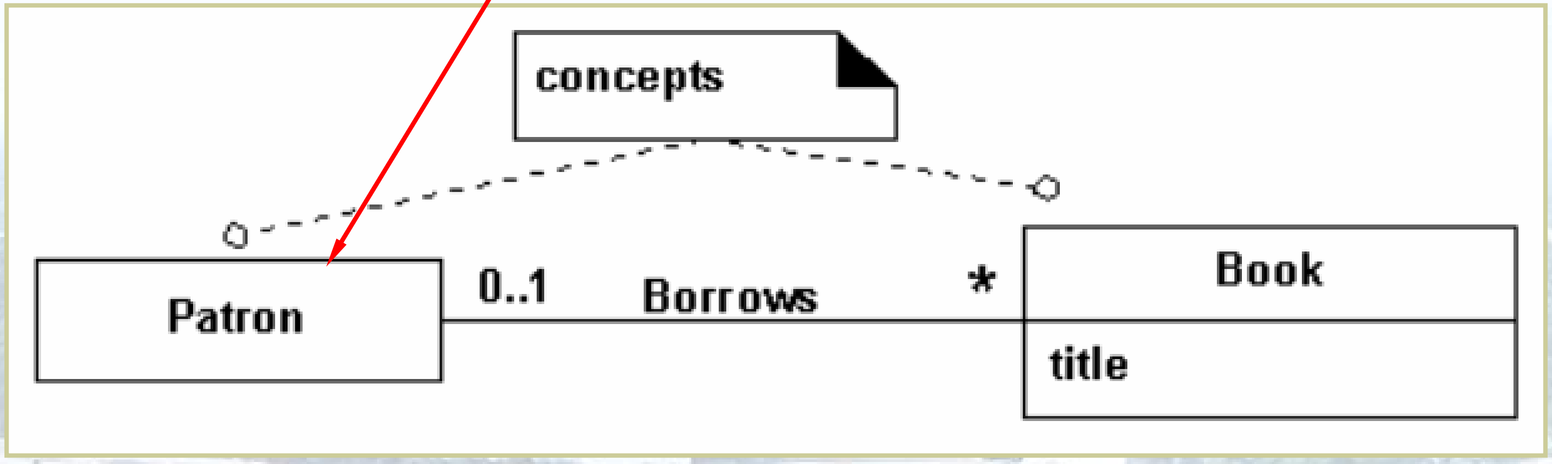
4.12.4 概念建模

- 需求分析是用 **Use case** 进行需求建模，而领域分析则是对问题域（**Problem Domain**）建立概念模型（**Conceptual Model**），即概念建模。
- 概念建模可以在缩小不同开发阶段之间的“语义间隔（**Semantic Gap**）”方面提供相应的支持。

4.12 面向对象系统分析

4.12.4 概念建模

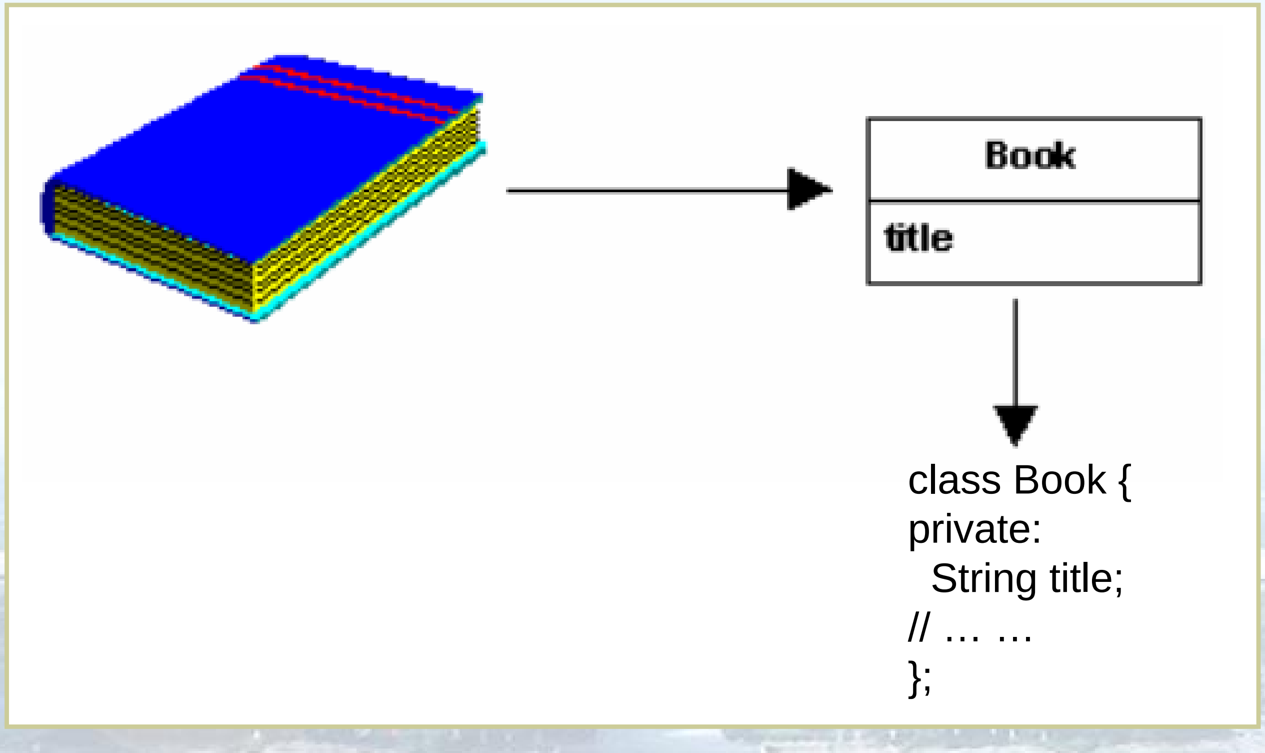
- 概念模型所表现的，是现实世界对应的问题域中的事物（早期的面向对象开发方法中即称之为“对象”，但这与面向对象程序设计中的对象是不同的：处于不同的抽象层次）。
- A concept is a set of examples of like things.
  - Example: *All library patrons*.



4.12 面向对象系统分析

4.12.4 概念建模

- 然而，我们不能期望在客观事物与概念模型中的概念之间总是有下面这样的一对一映射。

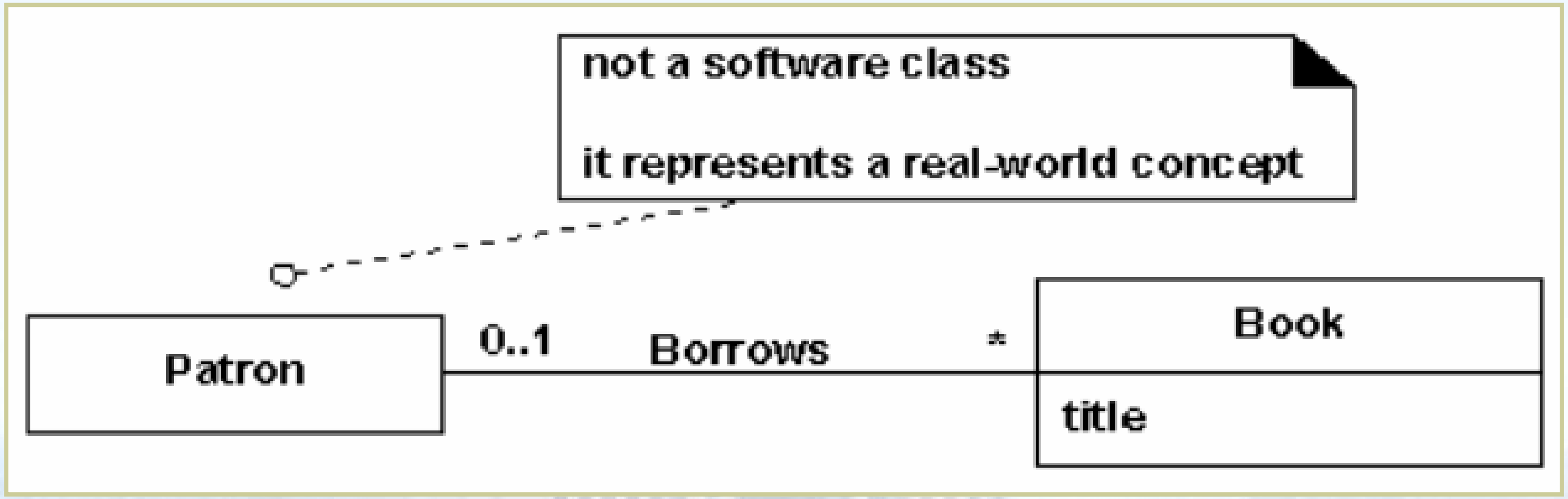


```
class Book {
private:
  String title;
// … …
};
```

4.12 面向对象系统分析

4.12.4 概念建模

- 概念模型不是由软件构件或面向对象程序设计语言中的类构成的图，而是在例示（illustrate）现实世界中的概念。



not a software class

it represents a real-world concept

4.12 面向对象系统分析

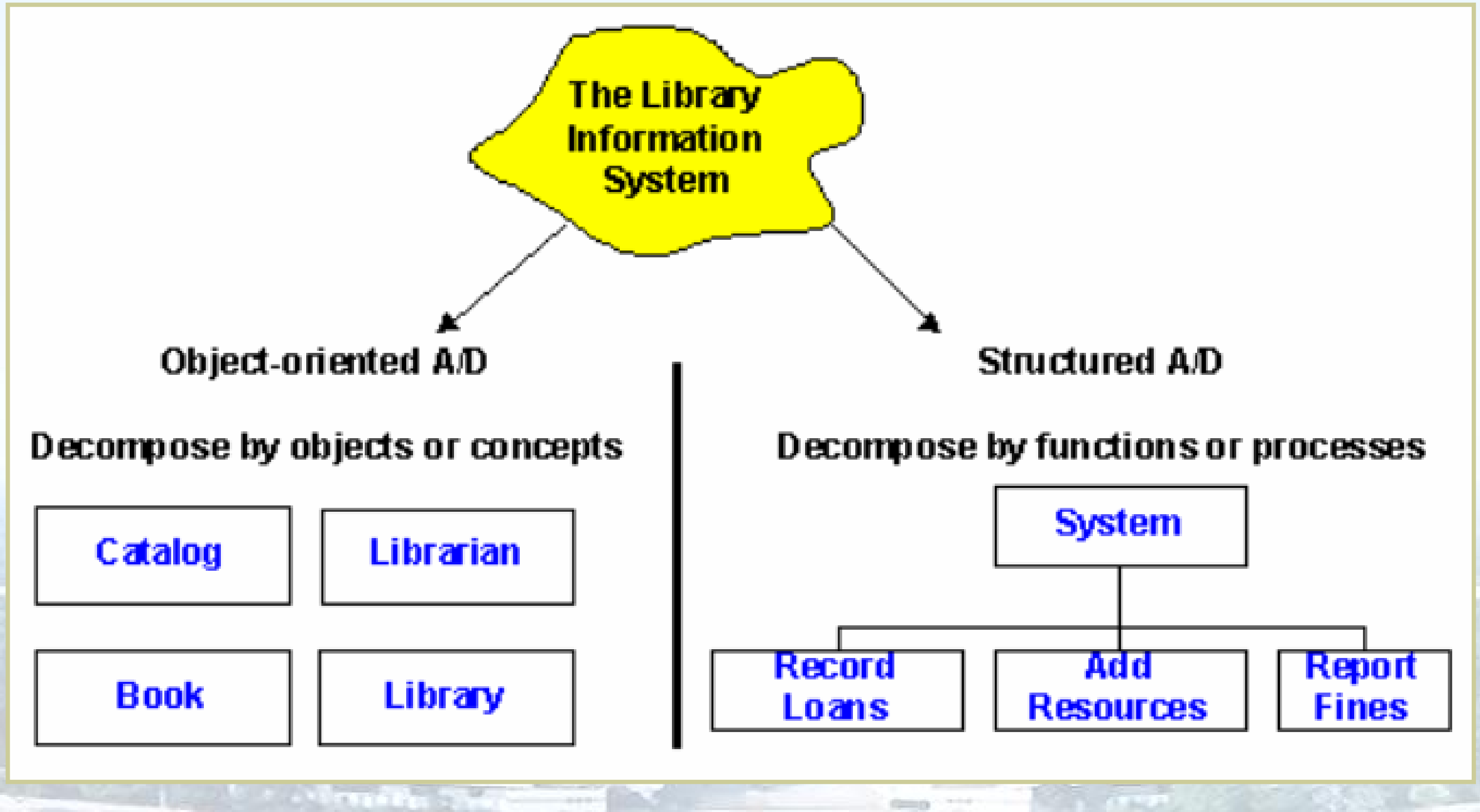
4.12.4 概念建模

- 概念建模的精神是：“The map-maker ”
  - Make a class model in the spirit of how a map-maker works.
  - A conceptual model is a map（示意图）of concepts in a domain.
  - Use the familiar names of the territory（地域）.
  - Do not add things that are not there.
  - Exclude information from a map if not relevant to the purpose of the map.

4.12 面向对象系统分析

4.12.4 概念建模

- 概念建模强调的是按事物而不是按功能来对系统进行分解。



4.12 面向对象系统分析

4.12.4 概念建模

- 概念建模的步骤:
  - 识别问题域中的概念;
  - 识别概念之间重要的关联;
  - 确定概念中的属性;
  - 产生系统序列图;
  - 产生操作合约（**Operation Contracts**）。

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- 建立概念模型，第一步是识别出问题域中的概念。
- 识别概念的三种策略:
  - Use the common concepts list.
  - Identify nouns and noun phrases (from use cases).
    - Example: *"computer department", "student information system"*
  - Study and apply models from others.
    - Example: Martin Fowler, *Analysis Patterns: Reusable Object Models*, Addison Wesley, 1997（中国电力出版社2003年6月影印版）

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Create a list of candidate concepts from this common concepts list.

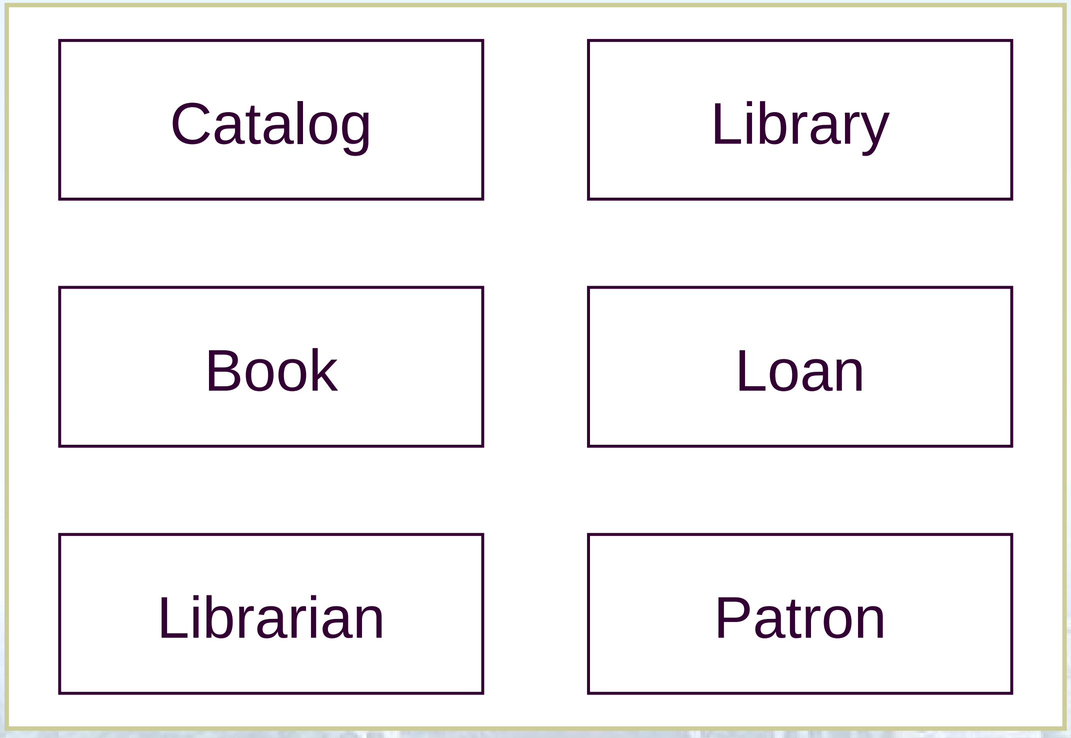
4.12 面向对象系统分析

Concept Category	Example
Tangible or physical objects	*Book*
Places	*Library*
Transactions	*ResourceLoan, Sale*
Transaction line items	*SalesLineItem*
Roles of people	*Librarian, Patron*
Specifications or descriptions	*ProductSpecification*
Scheduled events	*Appointment, Flight*
Containers	*Library, Catalog, Shelf*
Items in a container	*Book*
External systems	*StudentInformationSystem*
Organizations	*Library, ReceivingDept*
Policies	*LoanPolicy*

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model



Catalog

Library

Book

Loan

Librarian

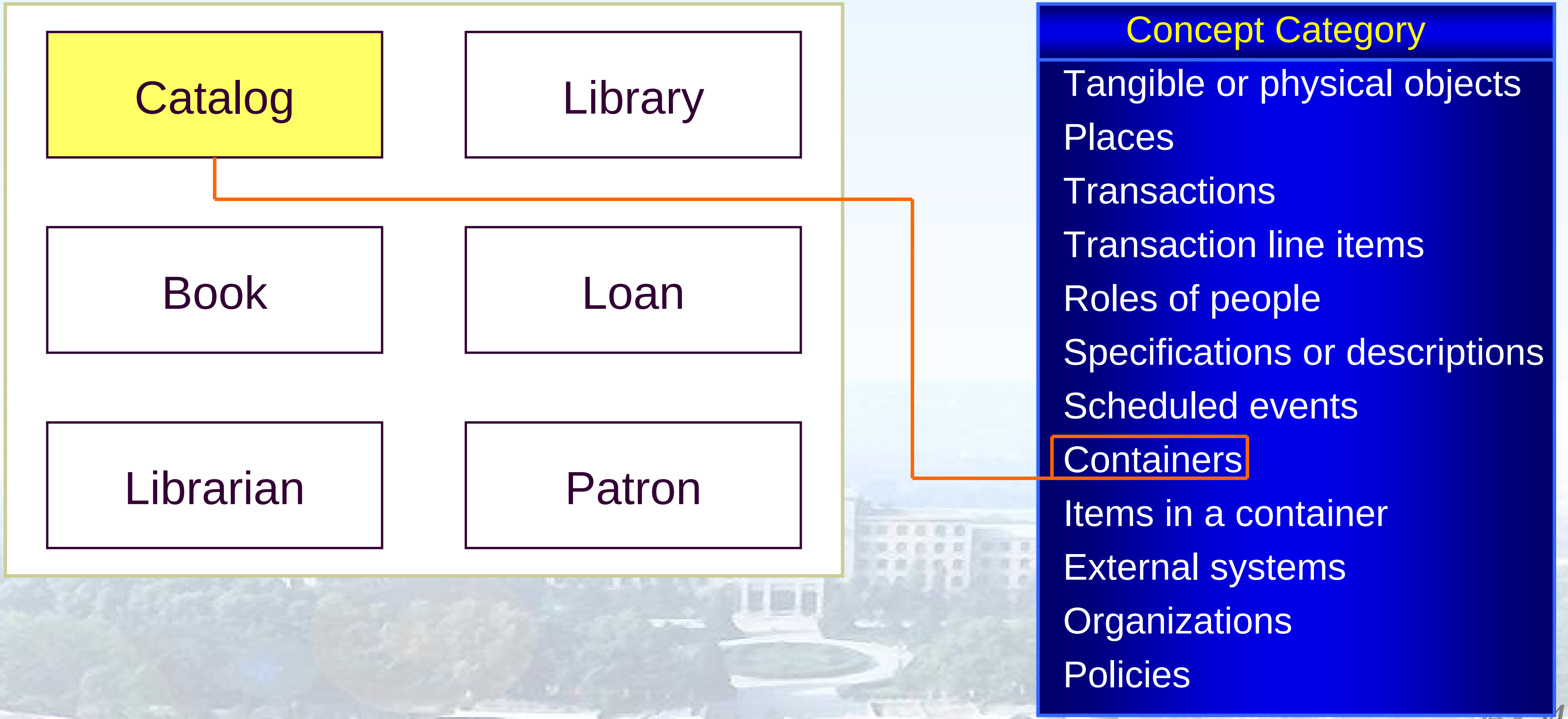
Patron

Concept Category
Tangible or physical objects
Places
Transactions
Transaction line items
Roles of people
Specifications or descriptions
Scheduled events
Containers
Items in a container
External systems
Organizations
Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model



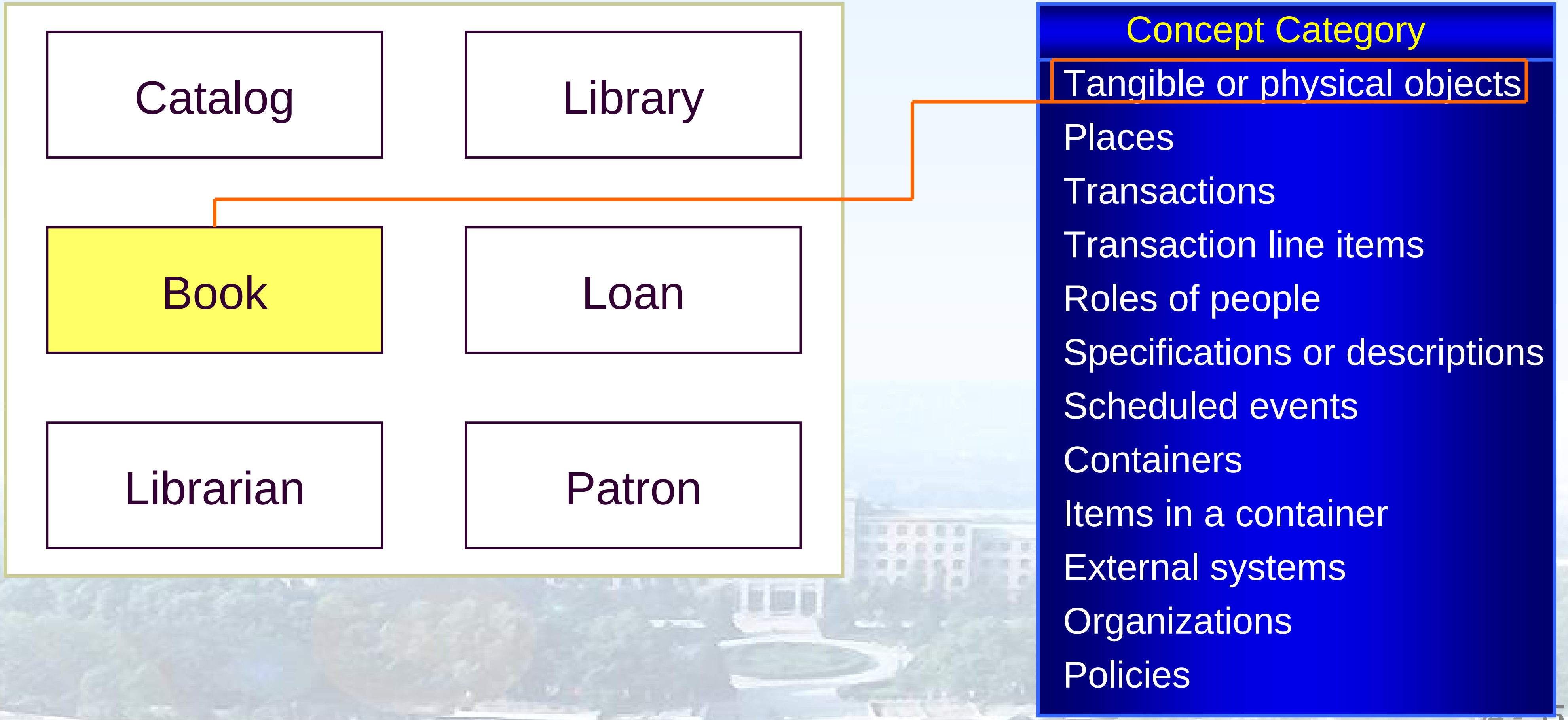
Concept Category

Tangible or physical objects
Places
Transactions
Transaction line items
Roles of people
Specifications or descriptions
Scheduled events
Containers
Items in a container
External systems
Organizations
Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model



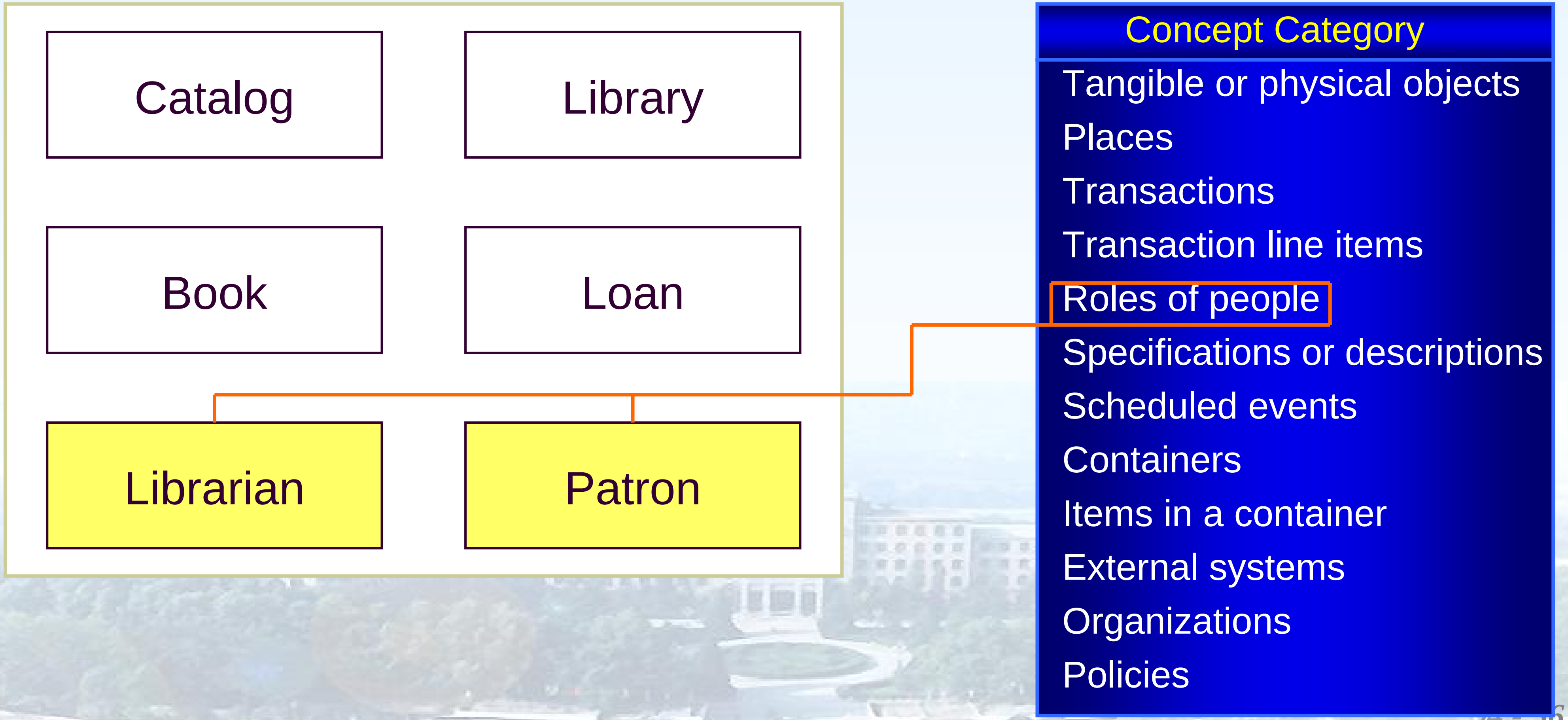
Concept Category

Tangible or physical objects
Places
Transactions
Transaction line items
Roles of people
Specifications or descriptions
Scheduled events
Containers
Items in a container
External systems
Organizations
Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model



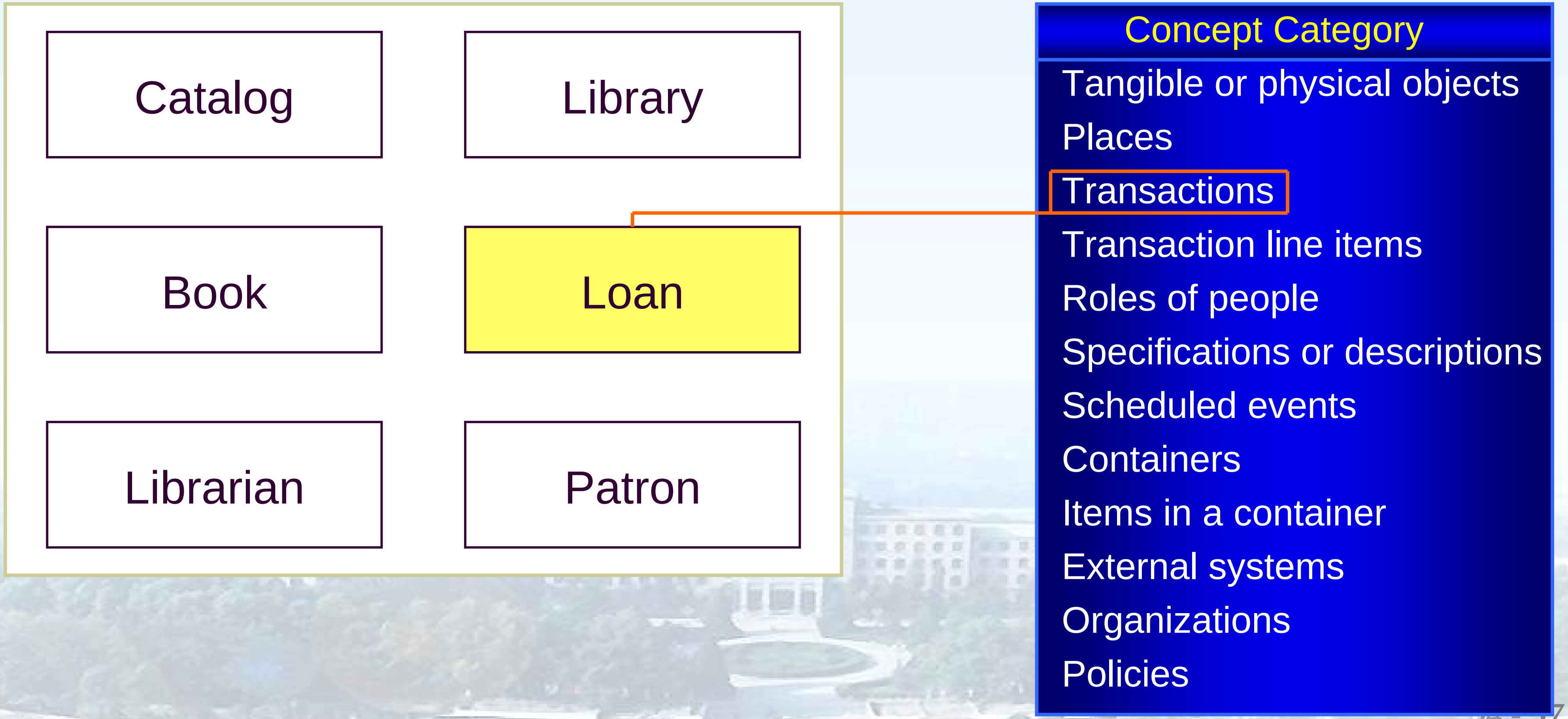
Concept Category

Tangible or physical objects
Places
Transactions
Transaction line items
Roles of people
Specifications or descriptions
Scheduled events
Containers
Items in a container
External systems
Organizations
Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model



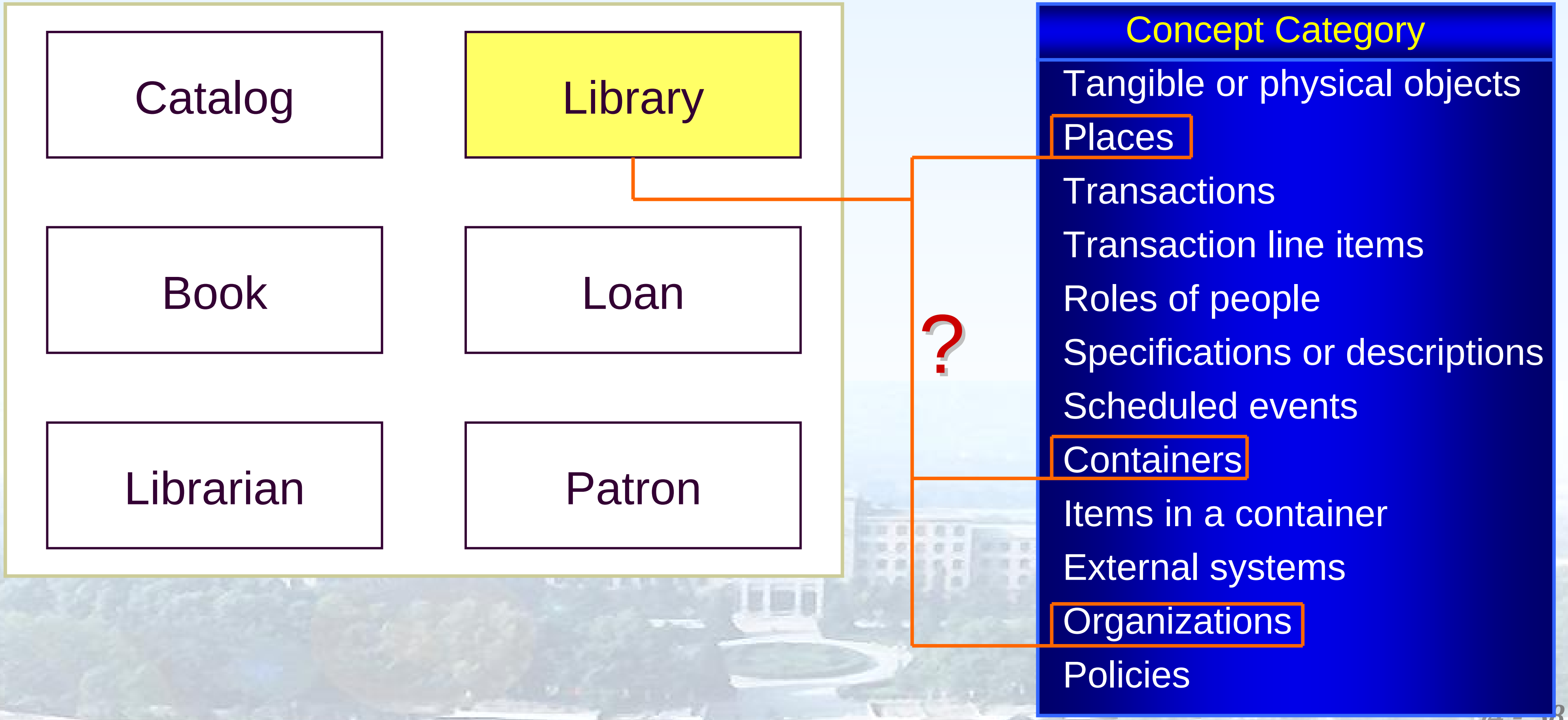
Concept Category

- Tangible or physical objects
- Places
- Transactions
- Transaction line items
- Roles of people
- Specifications or descriptions
- Scheduled events
- Containers
- Items in a container
- External systems
- Organizations
- Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model



Catalog

Library

Book

Loan

Librarian

Patron

?

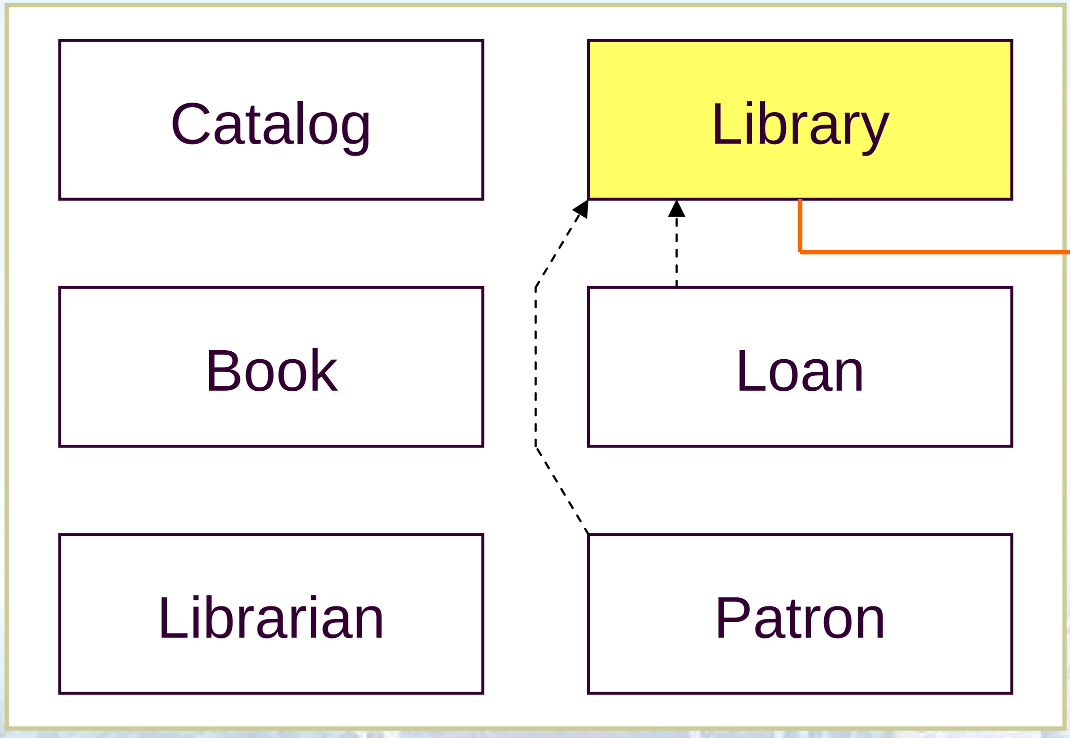
Concept Category

- Tangible or physical objects
- Places
- Transactions
- Transaction line items
- Roles of people
- Specifications or descriptions
- Scheduled events
- Containers
- Items in a container
- External systems
- Organizations
- Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model

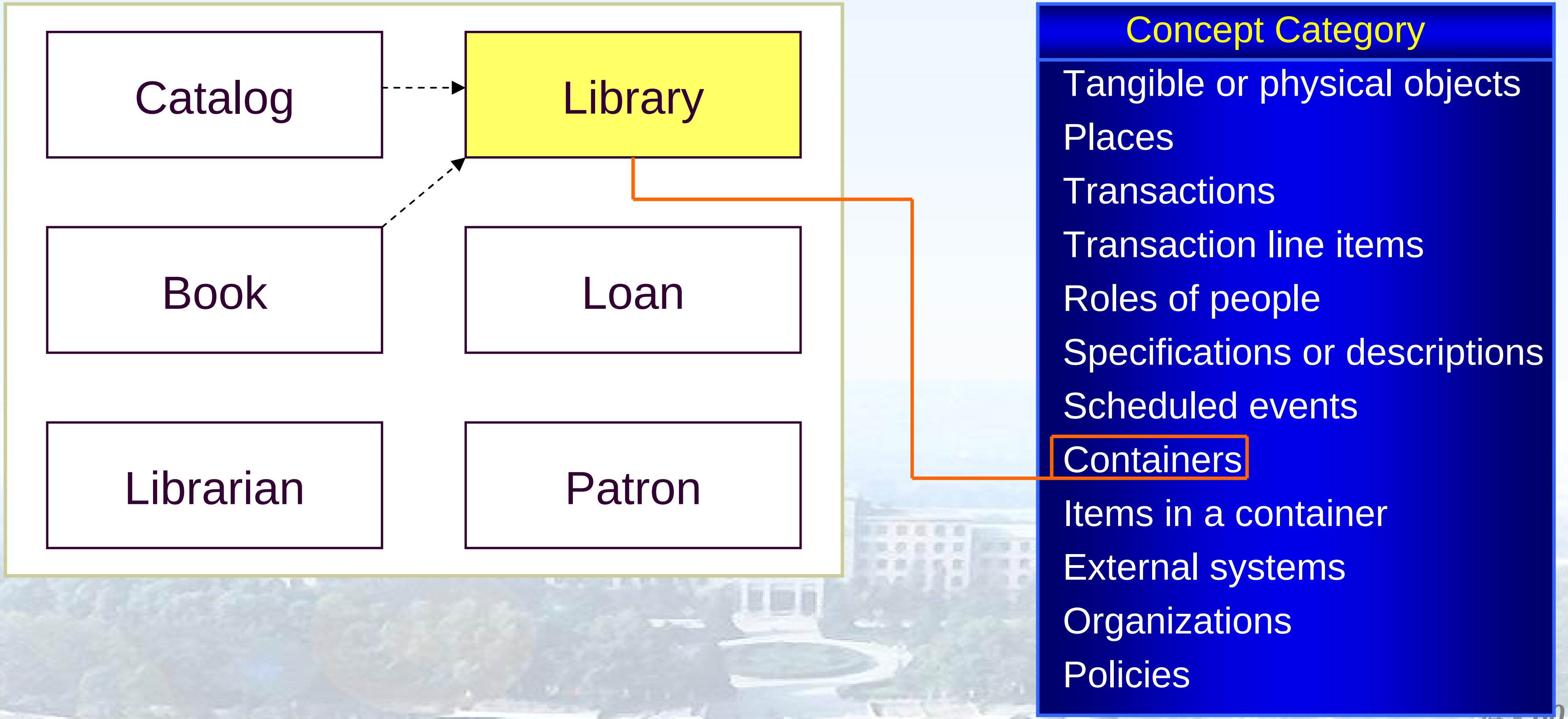


Concept Category
Tangible or physical objects
Places
Transactions
Transaction line items
Roles of people
Specifications or descriptions
Scheduled events
Containers
Items in a container
External systems
Organizations
Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model



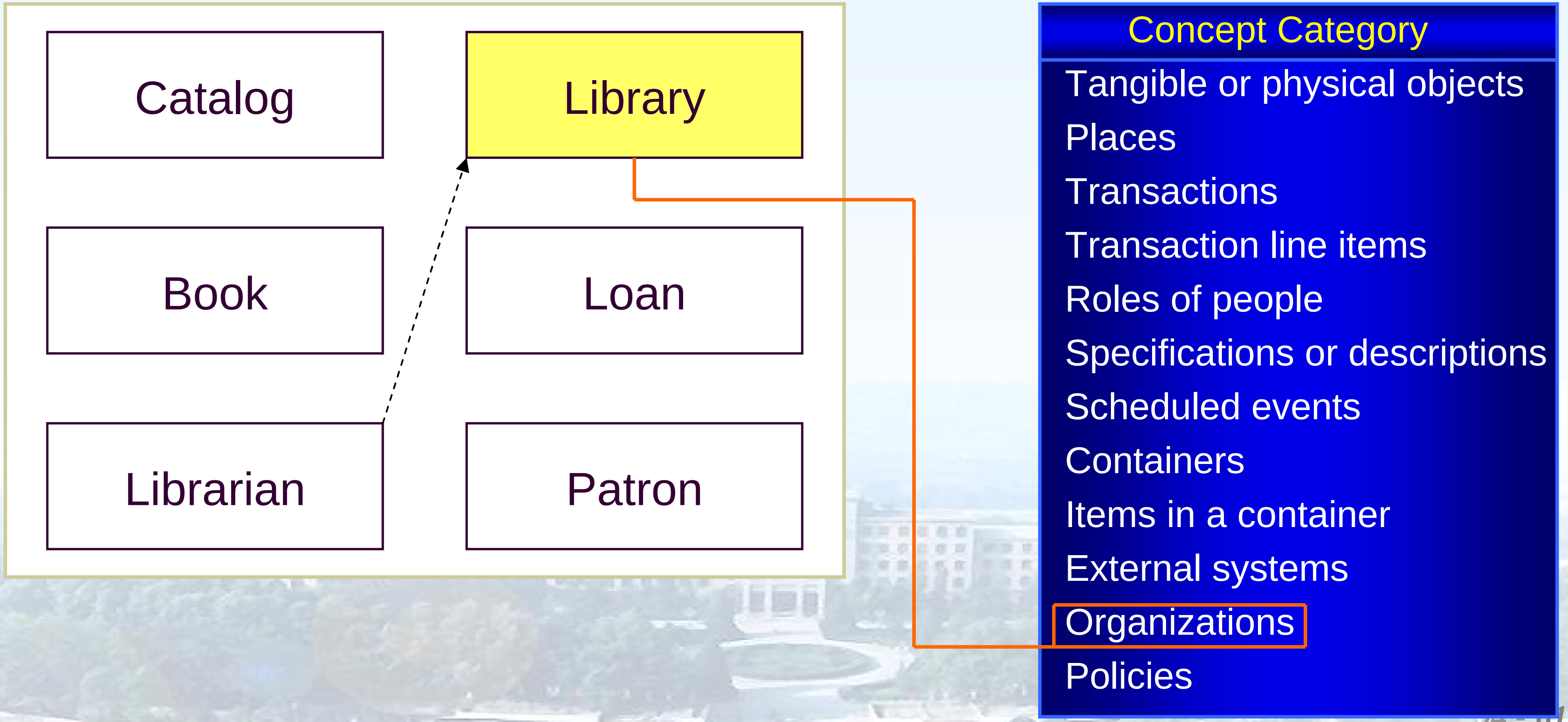
Concept Category

- Tangible or physical objects
- Places
- Transactions
- Transaction line items
- Roles of people
- Specifications or descriptions
- Scheduled events
- Containers
- Items in a container
- External systems
- Organizations
- Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Initial conceptual model



Catalog

Library

Book

Loan

Librarian

Patron

Concept Category

Tangible or physical objects

Places

Transactions

Transaction line items

Roles of people

Specifications or descriptions

Scheduled events

Containers

Items in a container

External systems

Organizations

Policies

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- 建立概念模型，第二步是识别出概念之间重要的关联。
- Use common association categories.

4.12 面向对象系统分析

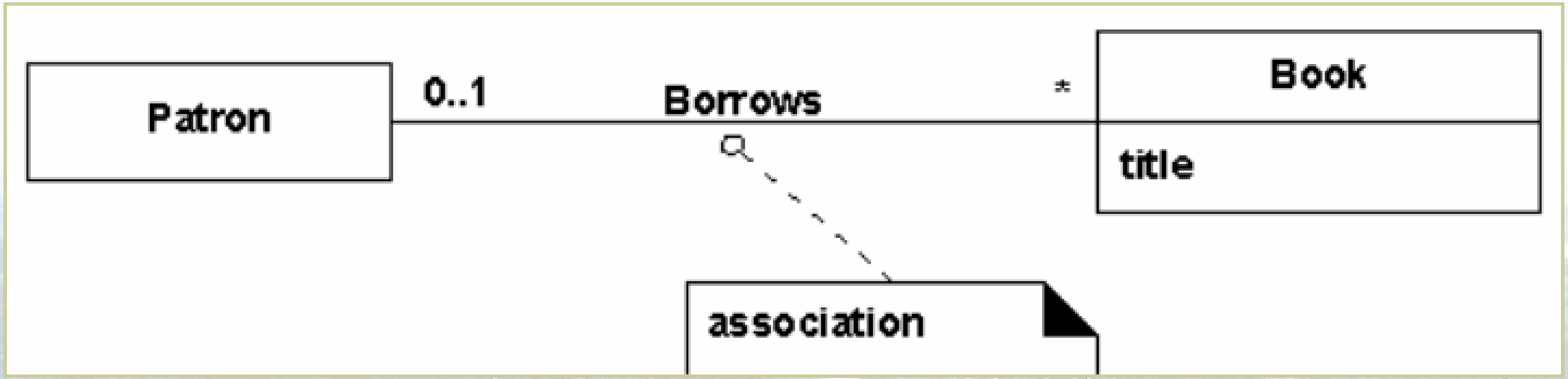
4.12.5 范例：一个图书馆信息系统的概念建模

Association Category	Examples
A is part of B	*Sentence – Paragraph*
A is a line item of B	*SalesLineItem – Sale*
A is contained with in B	*Book – Library*
A is a member of B	*Librarian – Library*
A is an organizational subunit of B	*Department – Company*
A is a policy related to B	*LoanPolicy – Book*
A is recorded in B	*Book – Catalog, Loan – Library*
A uses or manages B	*Patron – Library*
A is related to transaction of B	*Patron – ResouceLoan*
A communicates with B	*Patron – Librarian*
A is a transaction related to another transaction B	*Loan – Return*
A is next to B	*ChessSquare – ChessSquare*
A is related to B via a transaction	*Patron – Librarian, Husband – Wife*

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

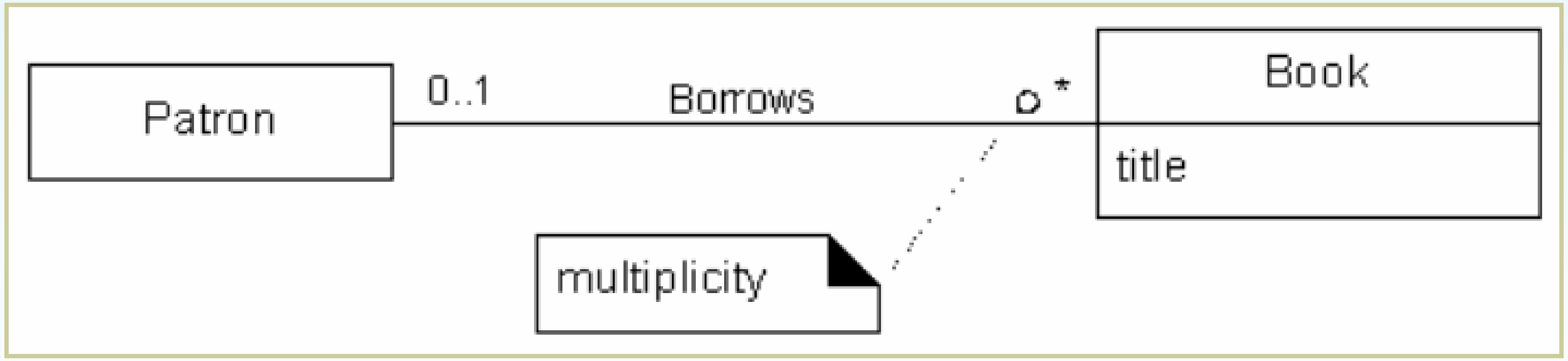
- Associations:
  - Are noteworthy relationships between concepts.
  - Are bidirectional.
  - Should have a meaningful name.
    - Form: <Noun><Verb-phrase><Noun>



4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Multiplicity（重复度）

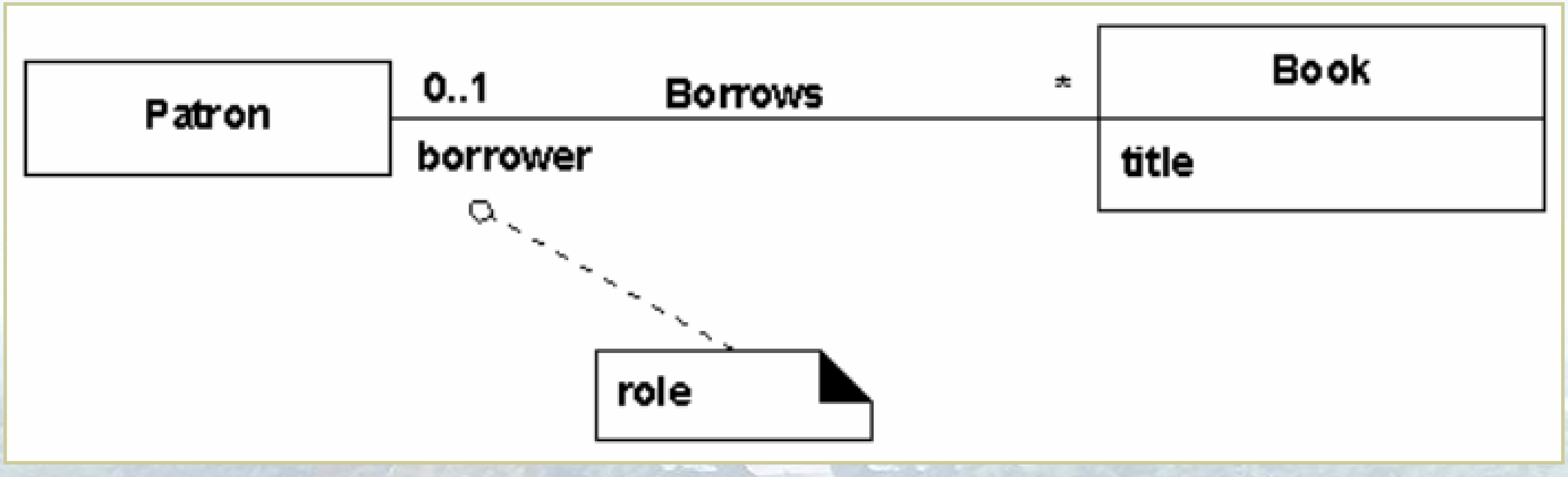


* = (Zero or more)	1 .. * = (One to many)
0 .. * = (Zero or more)	40 = (Exactly specified)
0 .. 2 = (Zero to two)	1, 3, 5 = (One or three or five)

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

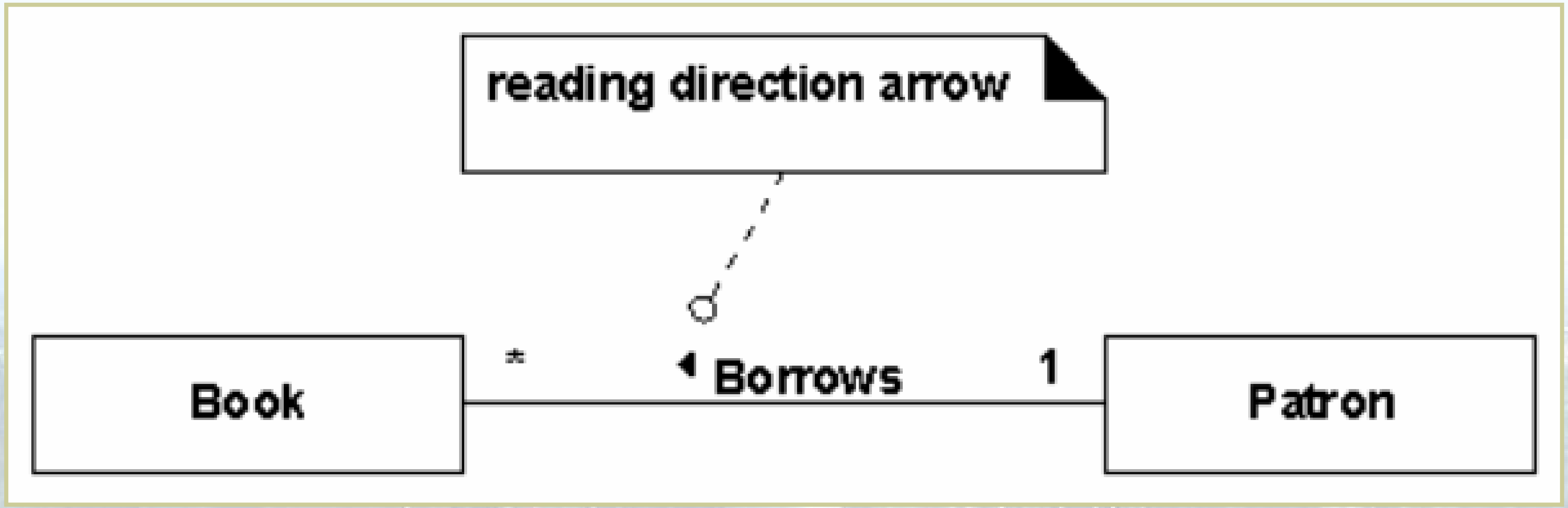
- Each end of an association is a role（角色）having:
  - Multiplicity.
  - An optional name.



4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

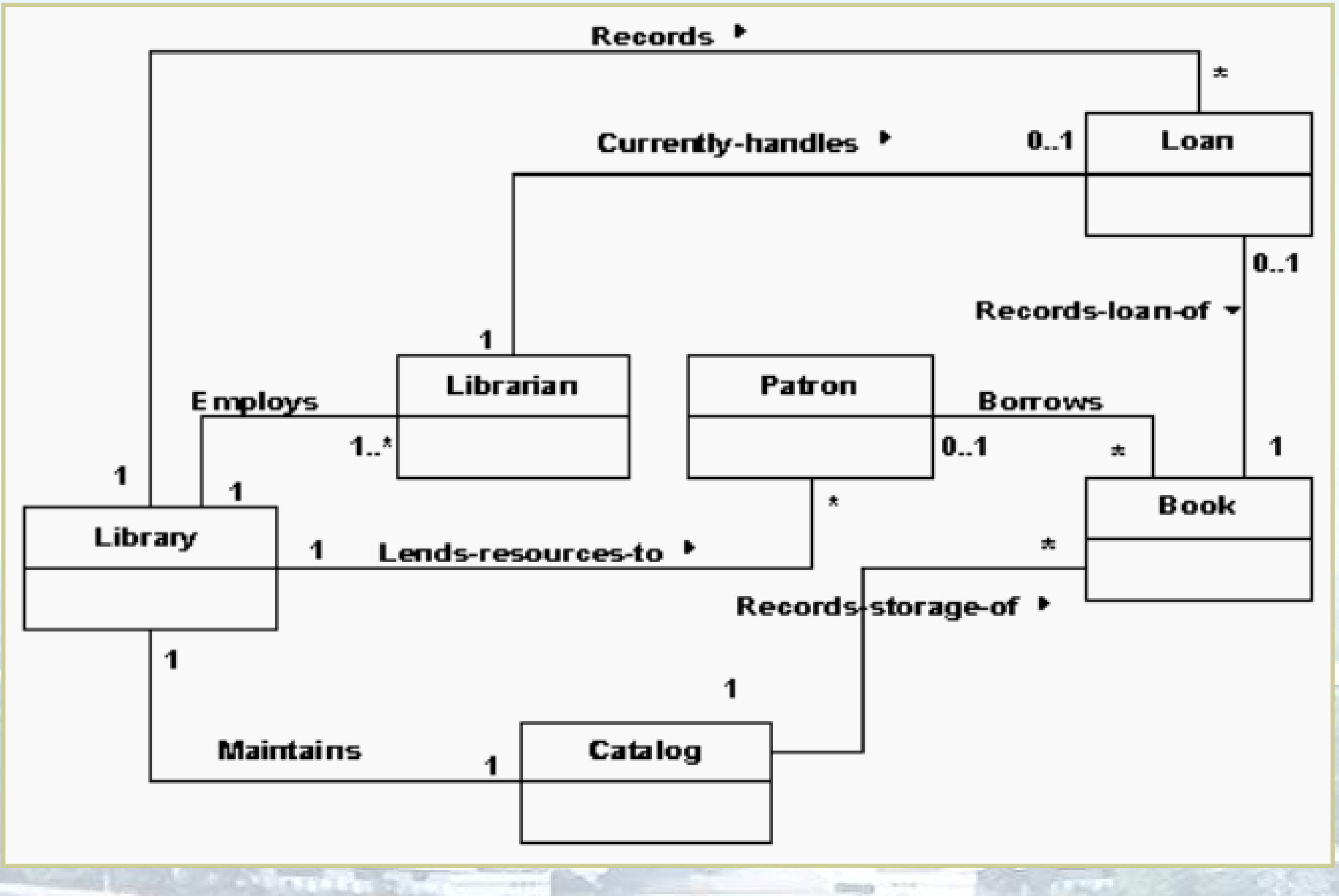
- Reading direction arrow:
  - An association can have an optional reading direction arrow.
  - Aid in reading only.
  - Has no semantic meaning.



4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

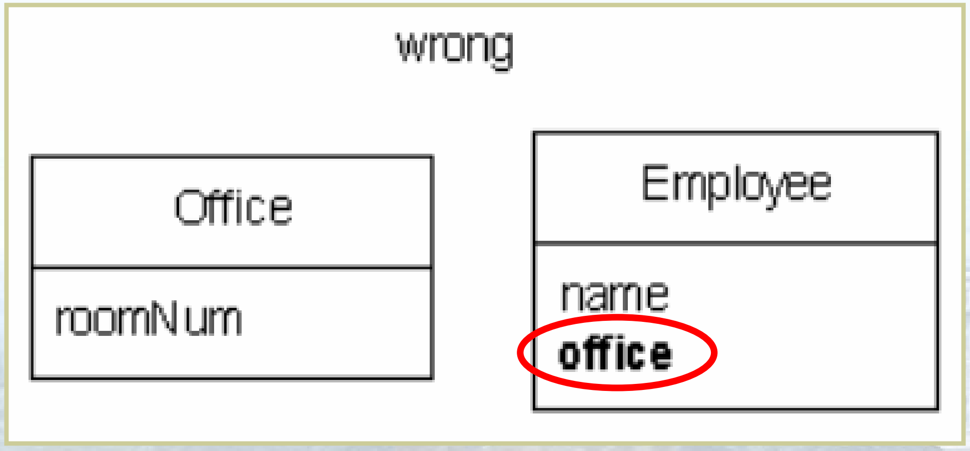
- Conceptual model example:

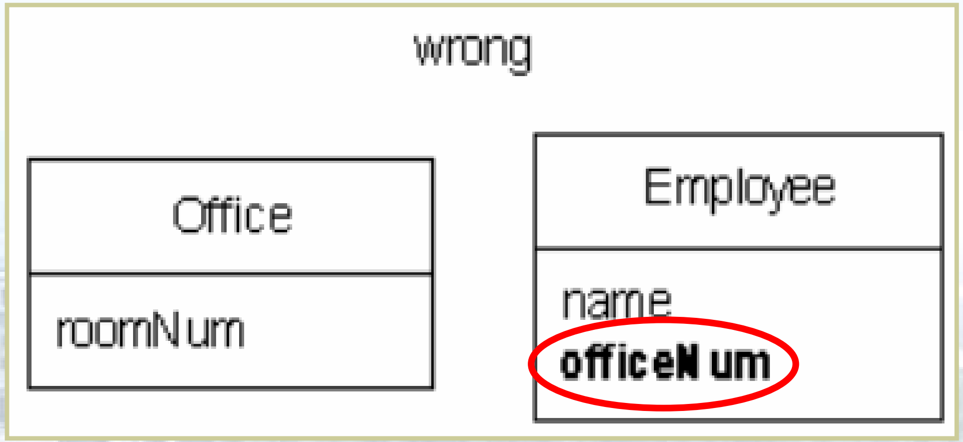


4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- 建立概念模型，第三步是确定概念中的属性。
- Only simple attributes.
    - Examples: *number, string, boolean, date, time.*
- No foreign keys（不要涉及实现时所需的属性）.

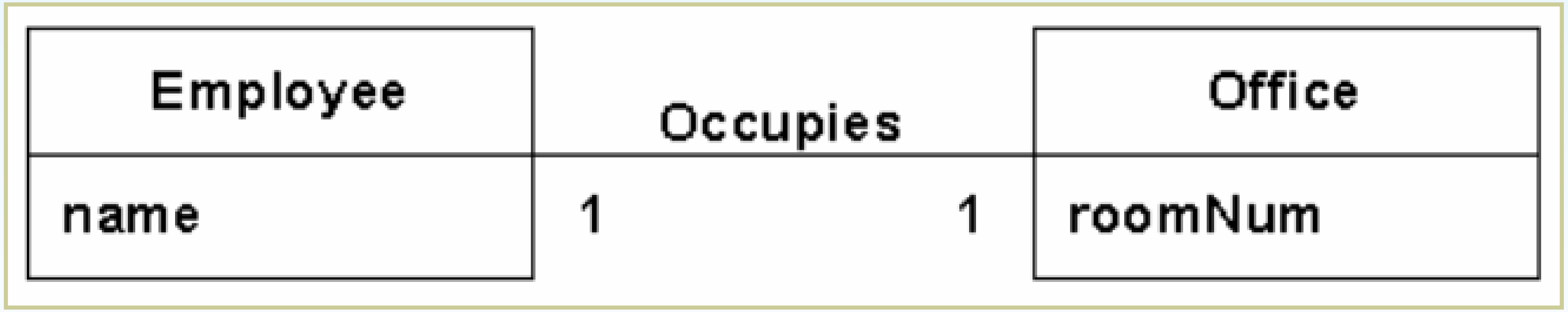




4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Relate concepts with an association, not an attribute.

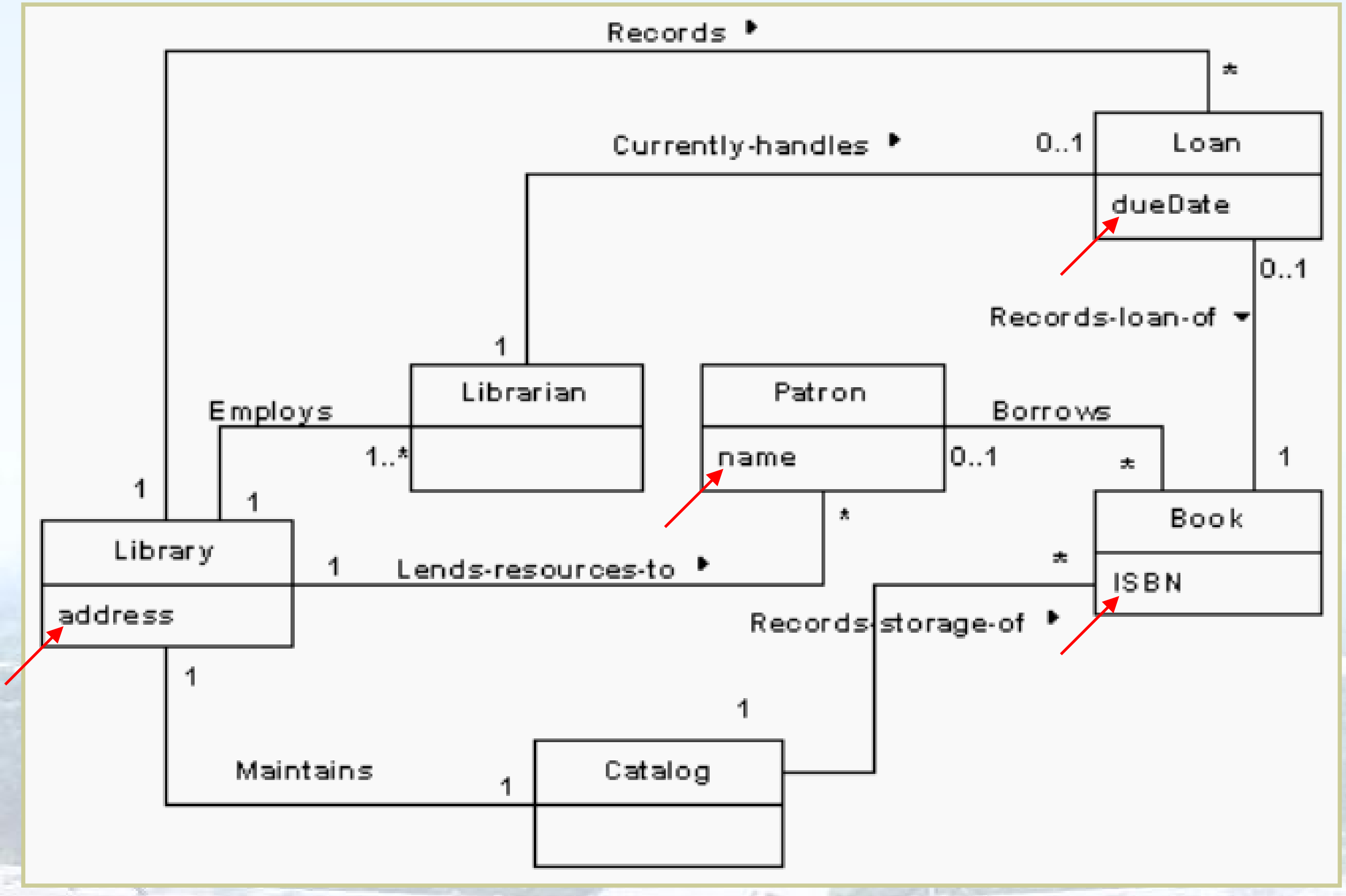


- Add attributes to fulfill the information needs of the requirements.

4.12 面向对象系统分析

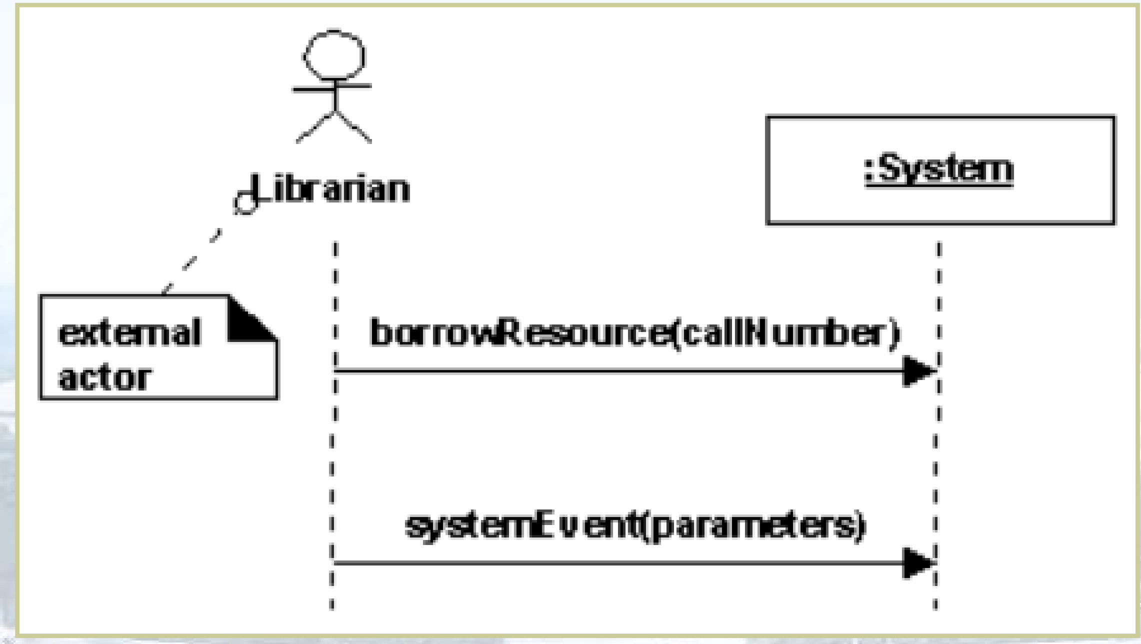
4.12.5 范例：一个图书馆信息系统的概念建模

- Sample attributes:



4.12.5 范例：一个图书馆信息系统的概念建模

- 建立概念模型，第四步是产生系统序列图。
- A system sequence diagram:
  - Illustrates time sequence of system events from actors.
  - Treats the system as a black box.



4.12 面向对象系统分析

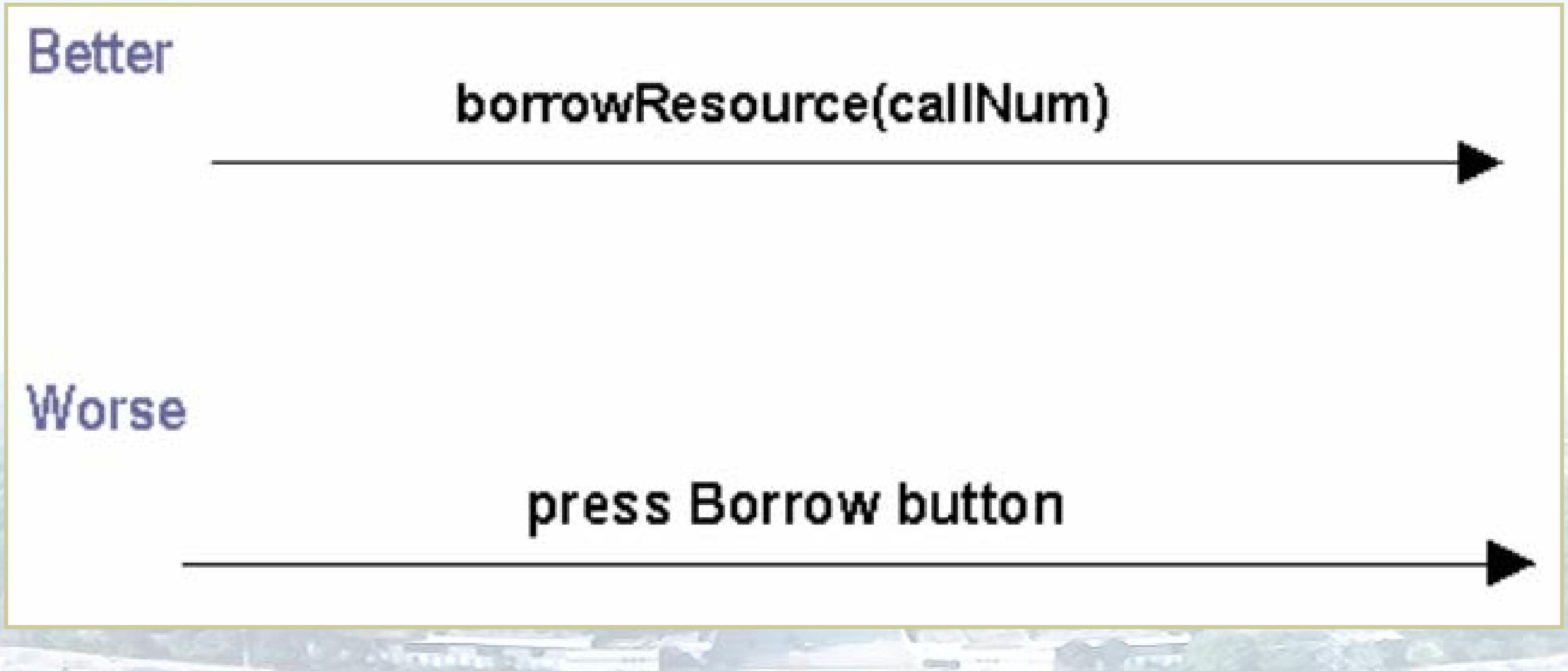
4.12.5 范例：一个图书馆信息系统的概念建模

- System event:
  - External input event generated by an external actor.
  - Indicates an operation should be performed.
- Every system has at least a *start()* system event.

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

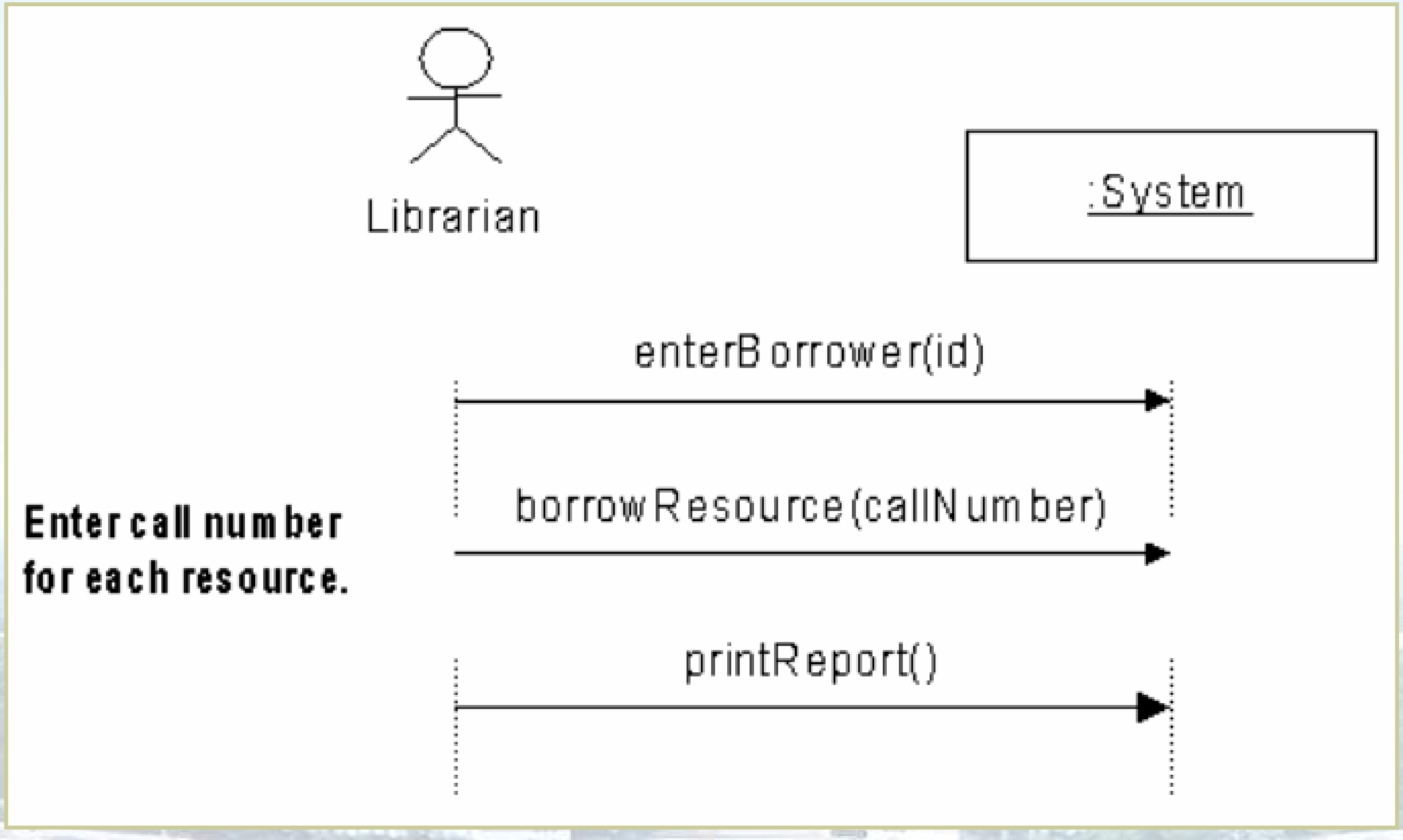
- Naming systems events:
    - Name system events at a high, abstract level of intent – not in terms of input technology.



4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

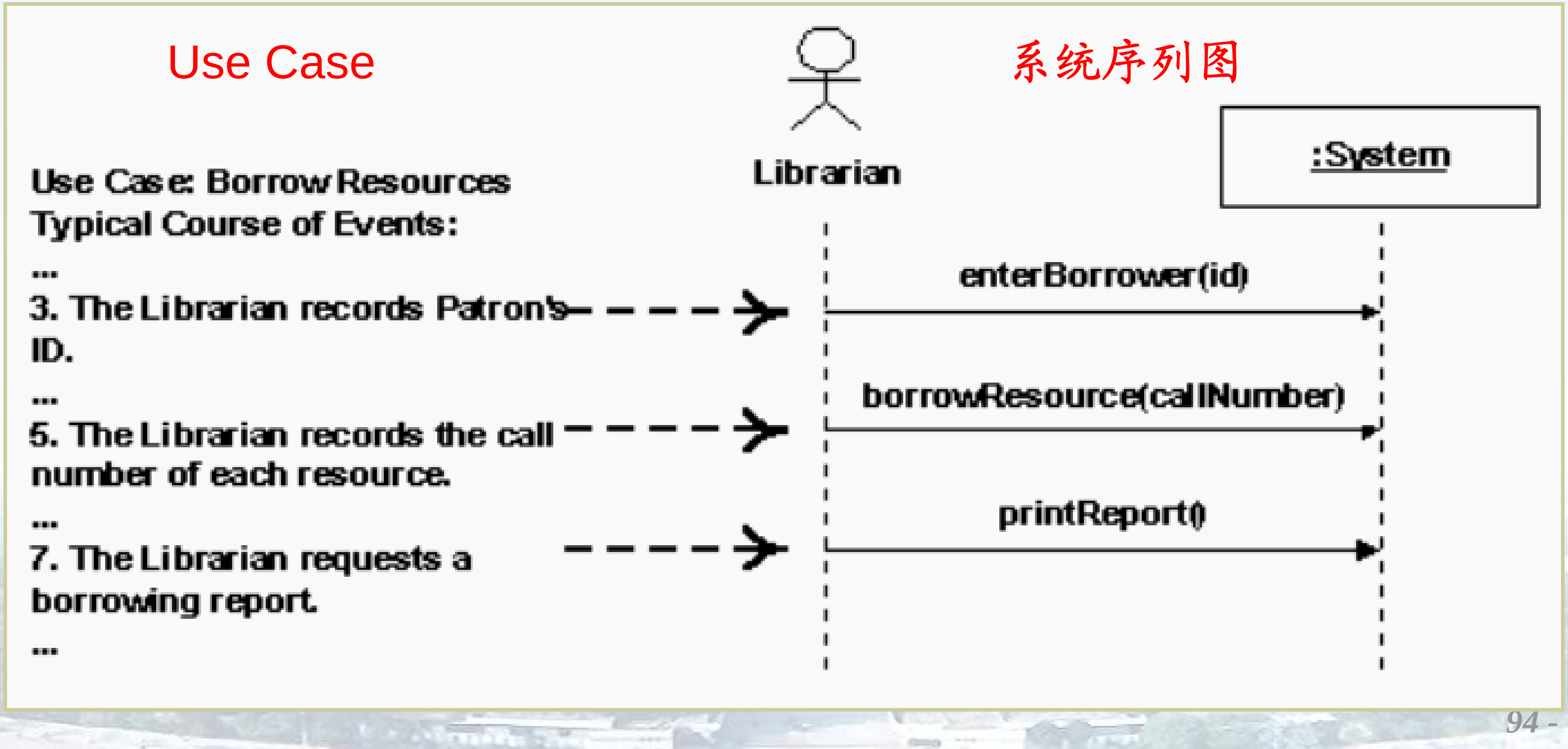
- Example:


Librarian
:System
enterBorrower(id)
borrowResource(callNumber)
printReport()
Enter call number for each resource.

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Artifact relationships:



Use Case

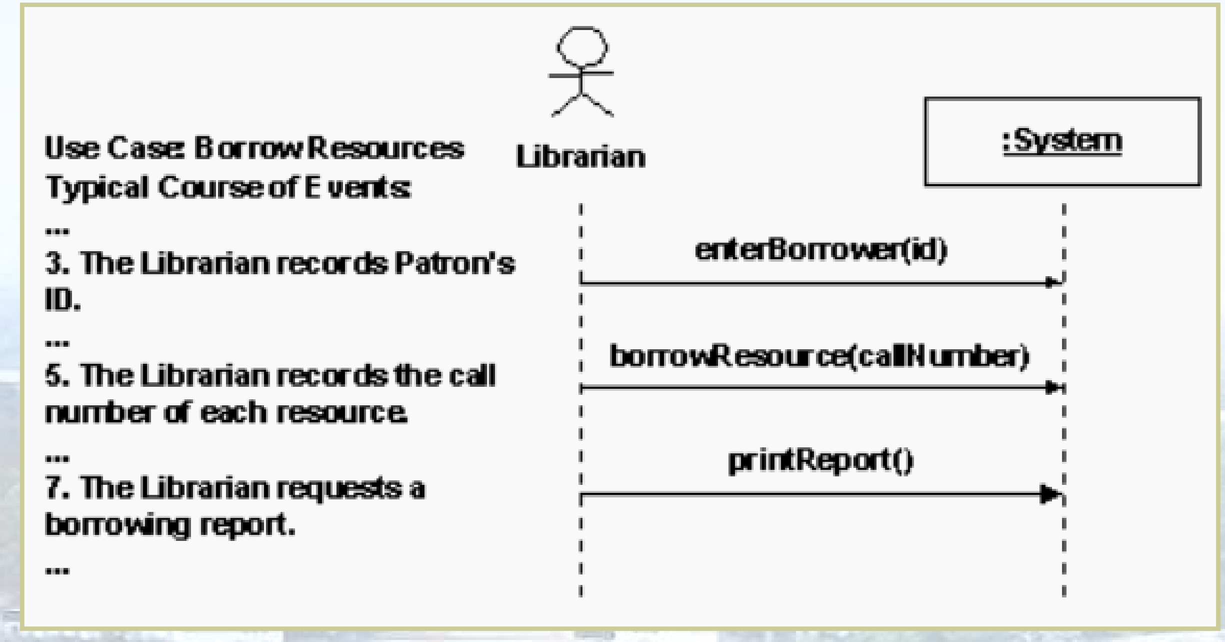
系统序列图

Use Case: Borrow Resources
Typical Course of Events:
...
3. The Librarian records Patron's ID.
...
5. The Librarian records the call number of each resource.
...
7. The Librarian requests a borrowing report.
...

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Use case text + sequence diagram:
  - It is legal in the UML to place use case text for the sequence diagram to the left.
  - The sentences should line up horizontally with the associated events.



Use Case: Borrow Resources
Typical Course of Events

...

3. The Librarian records Patron's ID.

...

5. The Librarian records the call number of each resource.

...

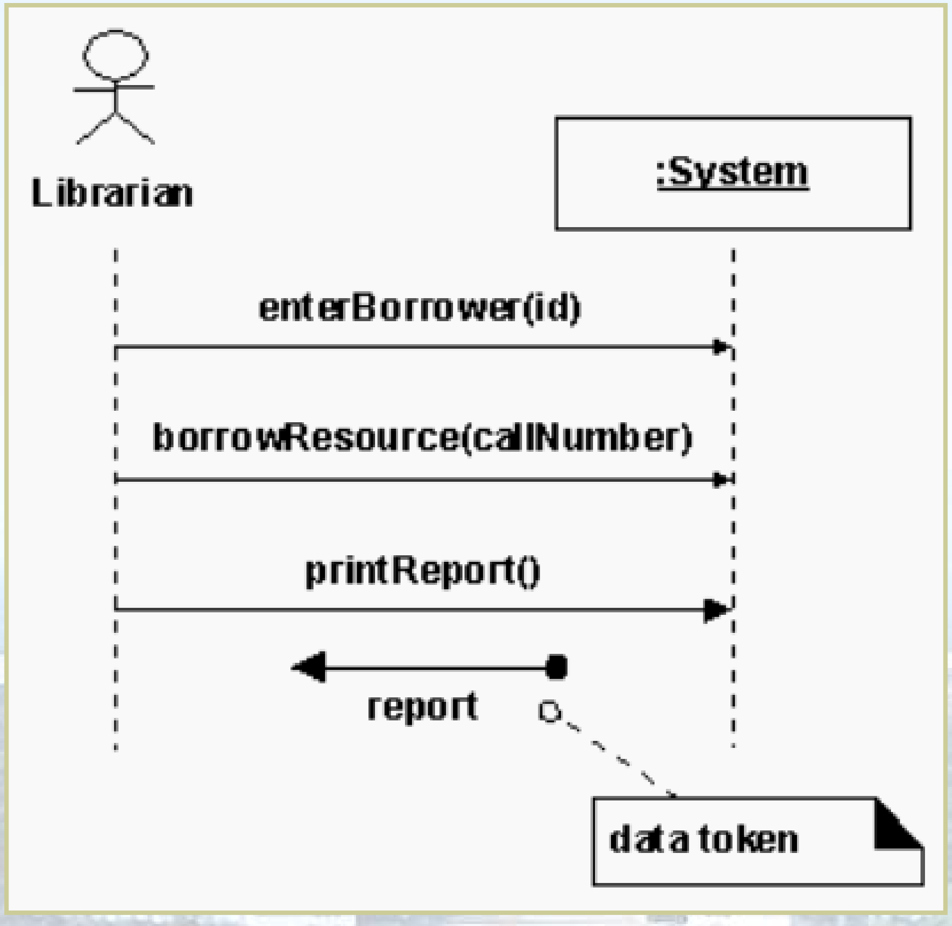
7. The Librarian requests a borrowing report.

...

4.12 面向对象系统分析

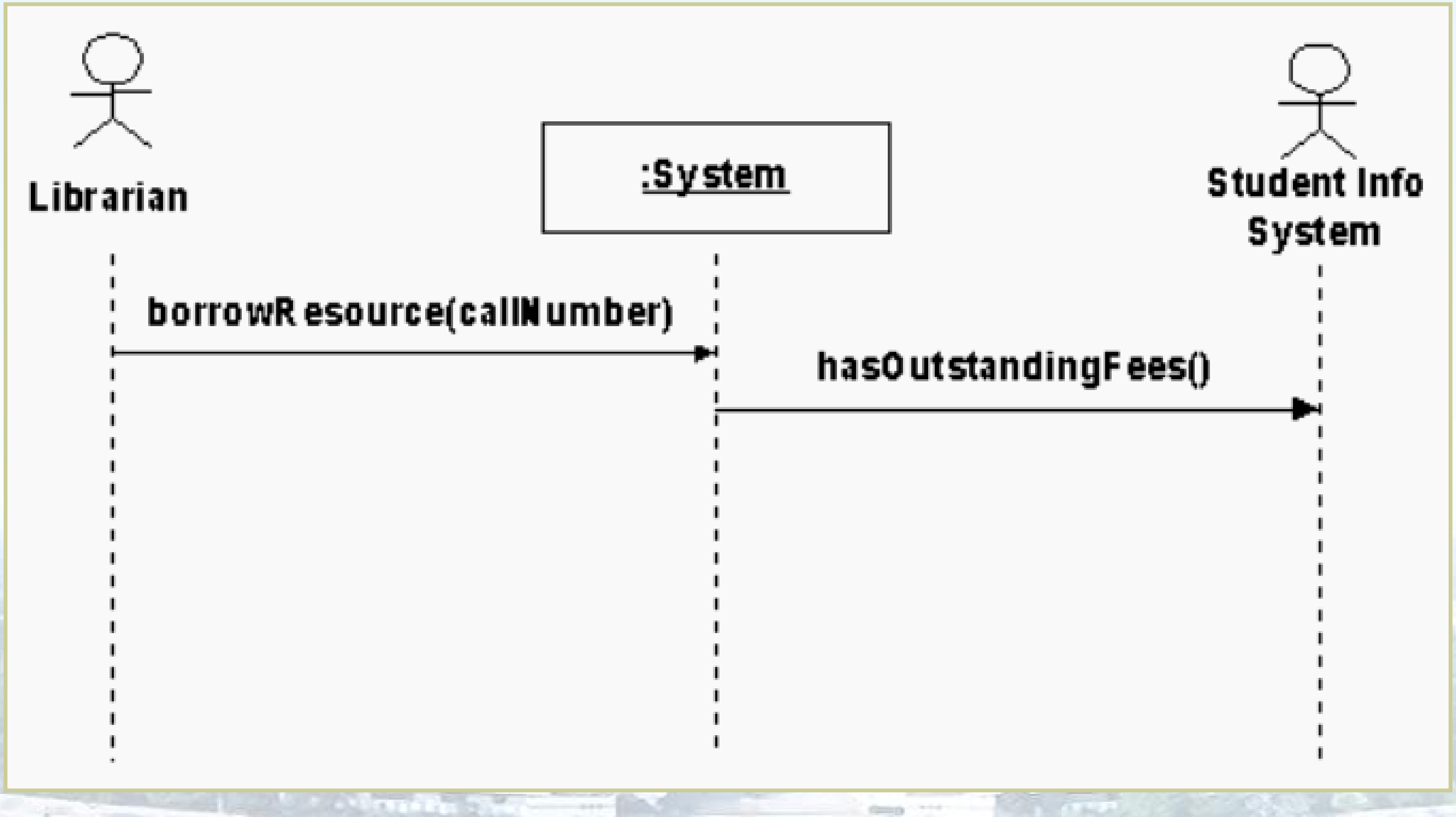
4.12.5 范例：一个图书馆信息系统的概念建模

- One approach to showing output:



4.12.5 范例：一个图书馆信息系统的概念建模

- Inter-system events:



4.12 面向对象系统分析

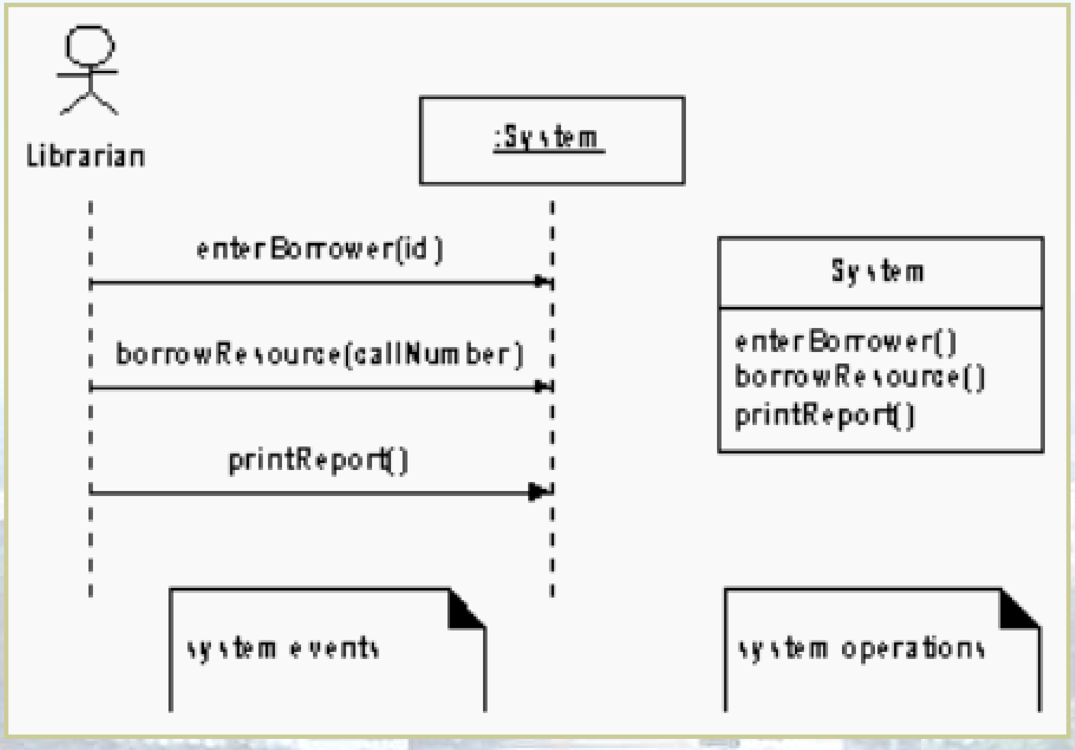
4.12.5 范例：一个图书馆信息系统的概念建模

- 建立概念模型，第五步是产生操作合约（**Operation Contracts**）。

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

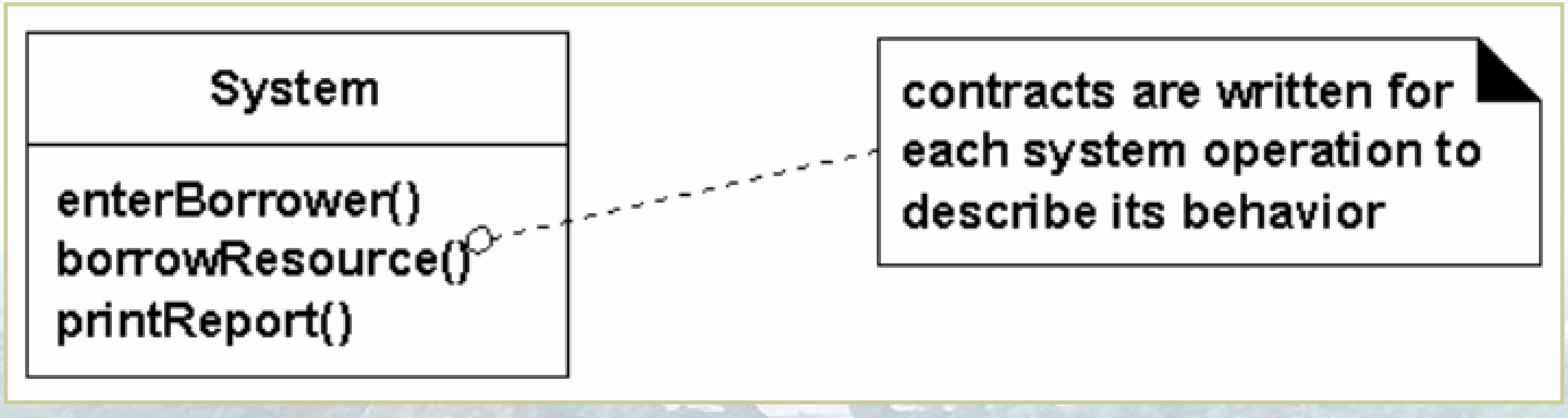
- System operations:
  - system event → system operation
    - analogous to: message → method



4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Operation contracts:
  - What does a system operation do?
  - An operation contract describes the changes in the state of the system when a system operation is invoked.


System
enterBorrower()
borrowResource()
printReport()
contracts are written for each system operation to describe its behavior

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Contract example:

Name:	borrowResource (callNum)
Responsibilities:	Record the borrowing of a library resources.
Cross References:	System Functions: R1.1, R1.2, R2.1 Use Case: Borrow Resources
Notes:	Use super-fast database lookup.
Preconditions:	The resources has a loan policy.
Postconditions:	
(instance creation)	• A loan, L, was created.
(association formed)	• L was associated with the Resource, res, whose callNumber = callNum.
(attribute modification)	• L.dueDate was set to today + loanPolicy.loanLength.
(association formed)	• L was associated with the Patron and Library.
(attribute modification)	• res.onLean was set to true.

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Postconditions（后置条件）：
  - Postconditions describe the state of a system - not actions.
- To encourage the right “declarative” frame of mind, express postconditions in the past tense.
  - (better) *A Loan was created.*
  - (worse) *Create a Loan.*

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Metaphor（比喻）: The system and its objects are presented on a theater stage（戏剧舞台）.
  1. Before the operation, take a picture of the stage.
  2. Close the curtains（幕布）on the stage and apply the system operation.
  3. Open the curtains and take a second picture.
  4. Compare the before and after pictures and express the changes as postconditions in the state of the stage.

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- Postcondition state changes:
  - Limit postconditions to the following categories:
    - Instance creation.
    - Associations formed.
    - Associations broken.
    - Attributes changed.

4.12 面向对象系统分析

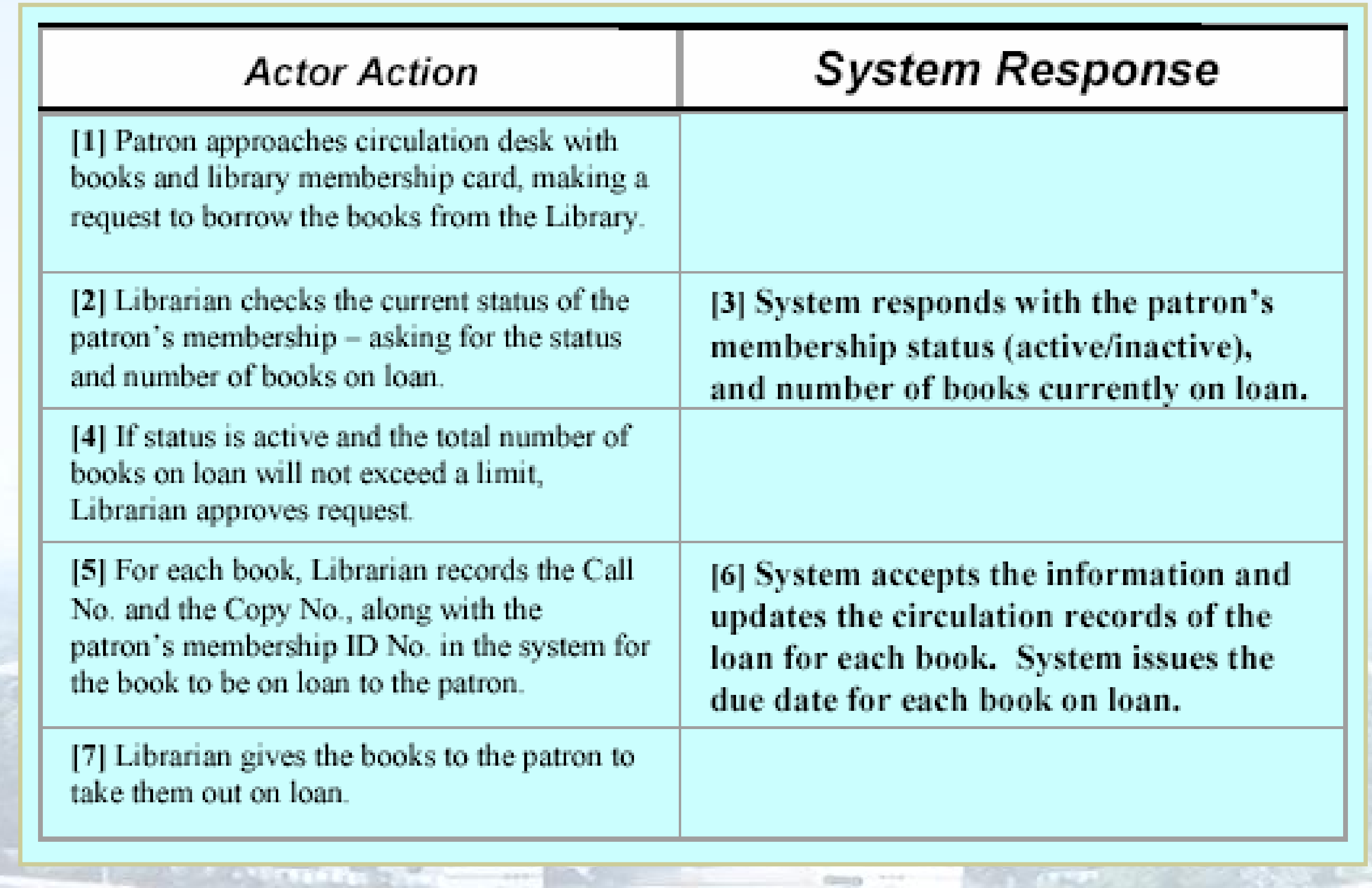
4.12.5 范例：一个图书馆信息系统的概念建模

- Ruminations（反刍）on contracts:
  - Describe system behavior along with use cases and system sequence diagrams.
  - Describe what an operation does in terms of state changes.
  - Defers（推迟） how it does it.

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- 例（借书业务的用例 – Borrow Books）：

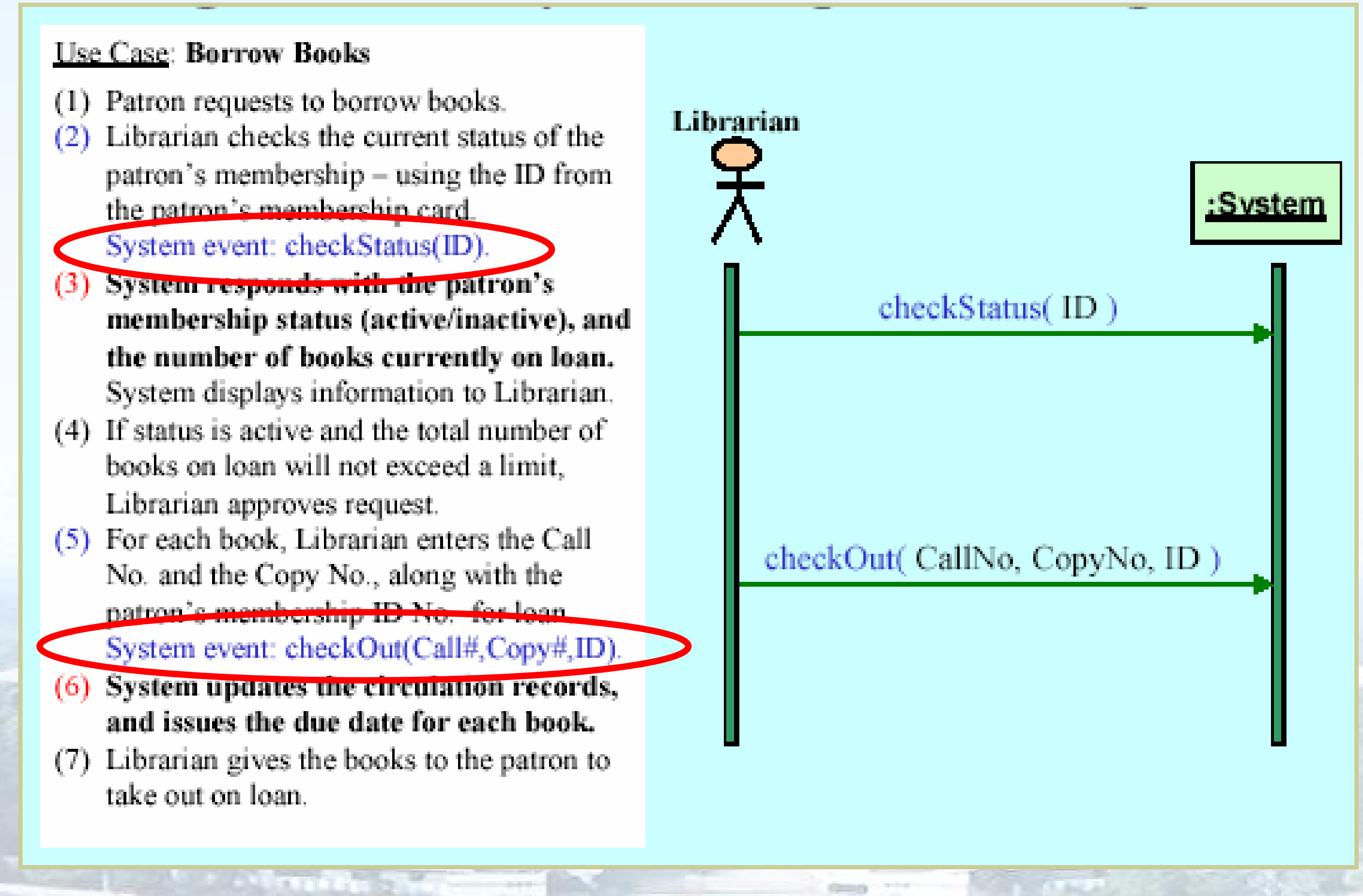


Actor Action	System Response
[1] Patron approaches circulation desk with books and library membership card, making a request to borrow the books from the Library.	
[2] Librarian checks the current status of the patron's membership – asking for the status and number of books on loan.	[3] System responds with the patron's membership status (active/inactive), and number of books currently on loan.
[4] If status is active and the total number of books on loan will not exceed a limit, Librarian approves request.	
[5] For each book, Librarian records the Call No. and the Copy No., along with the patron's membership ID No. in the system for the book to be on loan to the patron.	[6] System accepts the information and updates the circulation records of the loan for each book. System issues the due date for each book on loan.
[7] Librarian gives the books to the patron to take them out on loan.	

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- 例（Borrow Books 对应的系统序列图）：



Use Case: **Borrow Books**

(1) Patron requests to borrow books.

(2) Librarian checks the current status of the patron's membership – using the ID from the patron's membership card.
System event: checkStatus(ID).

(3) **System responds with the patron's membership status (active/inactive), and the number of books currently on loan.** System displays information to Librarian.

(4) If status is active and the total number of books on loan will not exceed a limit, Librarian approves request.

(5) For each book, Librarian enters the Call No. and the Copy No., along with the patron's membership ID No. for loan.
System event: checkOut(Call#,Copy#,ID).

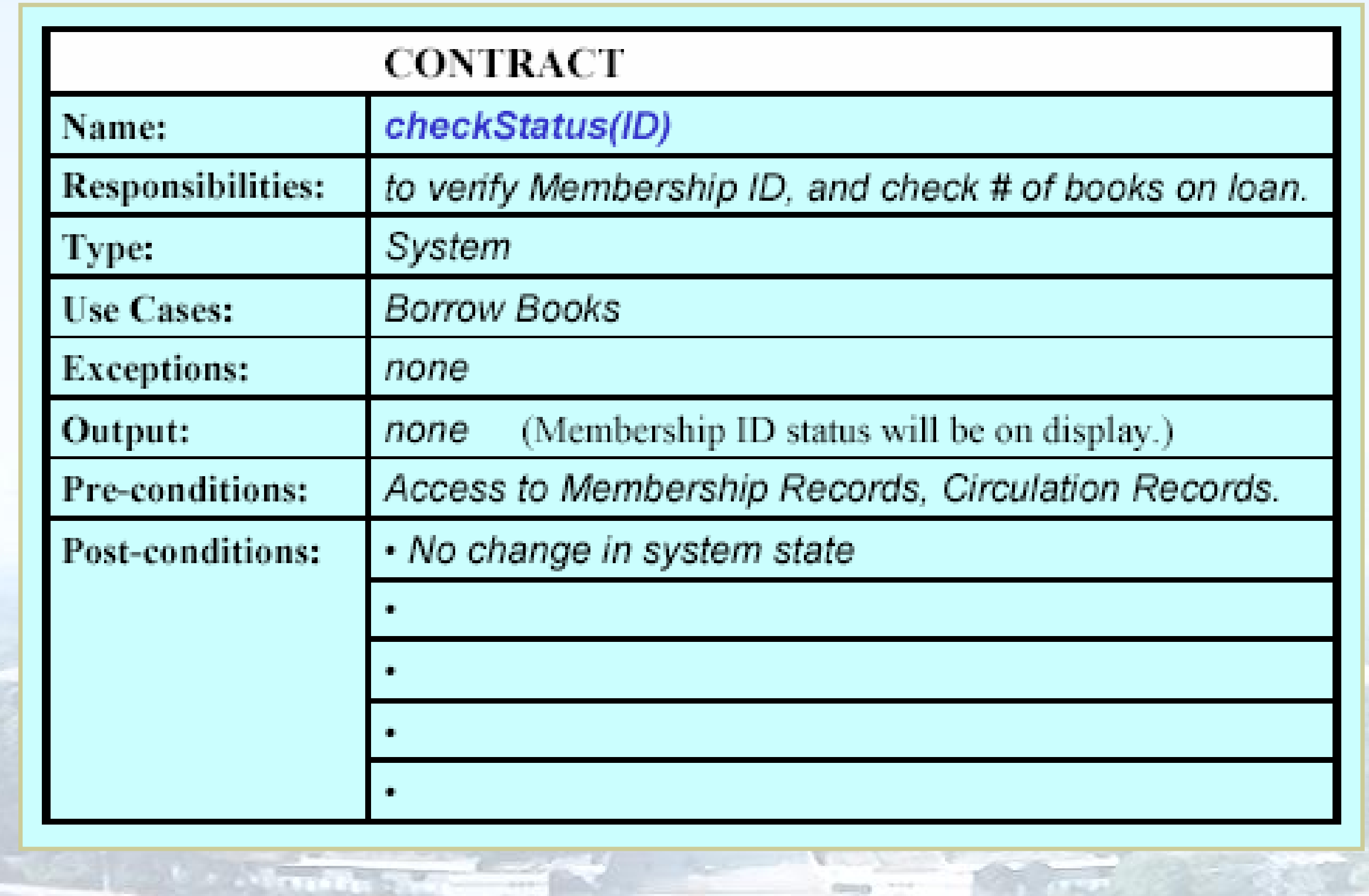
(6) **System updates the circulation records, and issues the due date for each book.**

(7) Librarian gives the books to the patron to take out on loan.

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- 例（系统操作 checkStatus(ID) 的操作合约）：

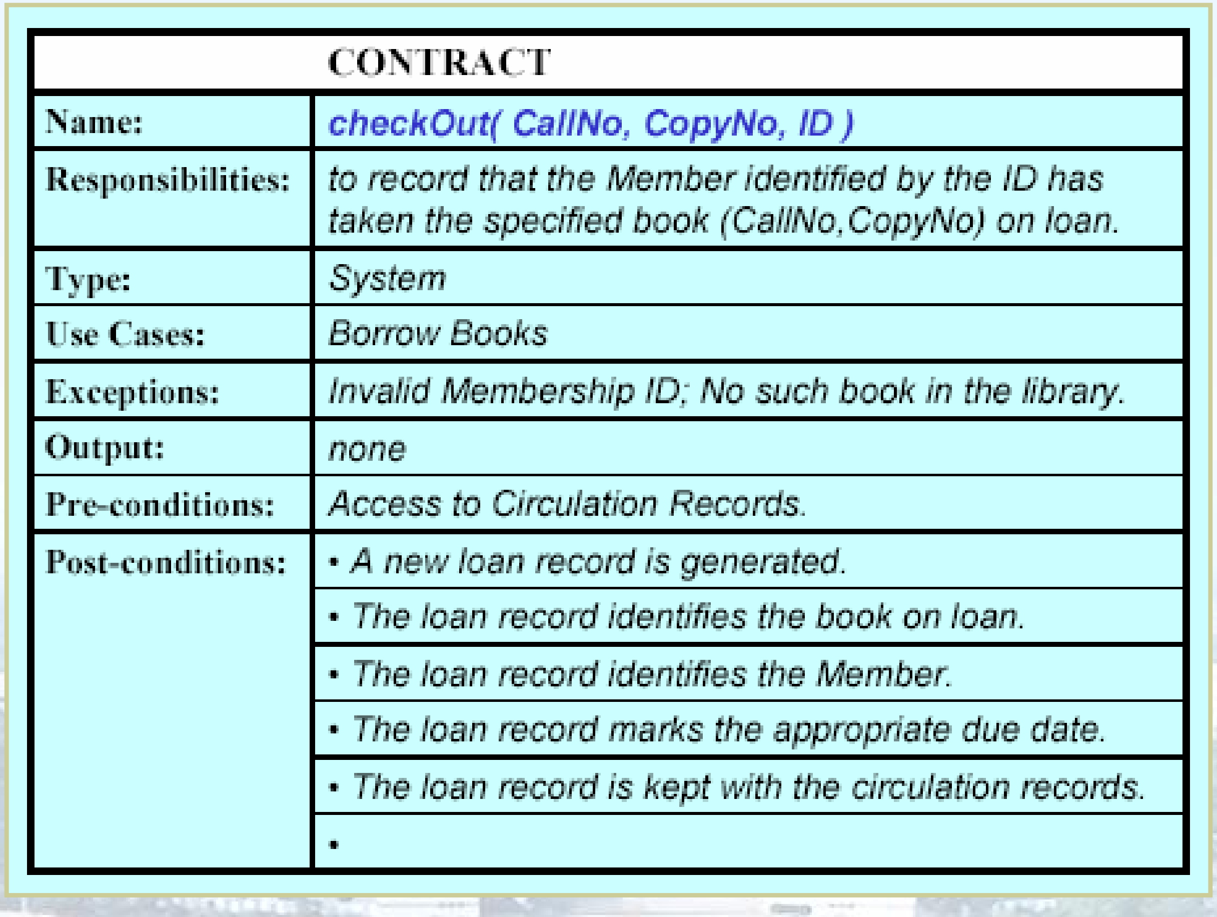


CONTRACT	
Name:	*checkStatus(ID)*
Responsibilities:	*to verify Membership ID, and check # of books on loan.*
Type:	*System*
Use Cases:	*Borrow Books*
Exceptions:	*none*
Output:	*none* (Membership ID status will be on display.)
Pre-conditions:	*Access to Membership Records, Circulation Records.*
Post-conditions:	• *No change in system state*
	•
	•
	•
	•

4.12 面向对象系统分析

4.12.5 范例：一个图书馆信息系统的概念建模

- 例（系统操作 checkOut(CallNo, CopyNo, ID) 的操作合约）：



CONTRACT	
Name:	*checkOut(CallNo, CopyNo, ID)*
Responsibilities:	*to record that the Member identified by the ID has taken the specified book (CallNo,CopyNo) on loan.*
Type:	*System*
Use Cases:	*Borrow Books*
Exceptions:	*Invalid Membership ID; No such book in the library.*
Output:	*none*
Pre-conditions:	*Access to Circulation Records.*
Post-conditions:	• *A new loan record is generated.*
	• *The loan record identifies the book on loan.*
	• *The loan record identifies the Member.*
	• *The loan record marks the appropriate due date.*
	• *The loan record is kept with the circulation records.*
	•

要点与引伸

- 信息系统必须实现用户所需的业务功能。
- 利用 Use case 进行功能性需求建模，首先要完整地表达出用户需求，其次还要帮助用户对需求进行归纳、抽象和整理，后者体现在对 Use case 的适度分解与相互关联定义上。
- 利用 Use case 进行功能性需求建模，是从本质功能到功能细节渐进的发现和认识过程，重要的是抓住系统功能的本质。
- 识别 Actor 和 Use cases、阅读和书写高层次的和详细格式的 Use cases、区分本质的 Use cases 和 现实的 Use cases、利用三种关联组织好 Use cases 等，这些技能是利用 Use case 进行功能性需求建模的关键。

要点与引伸（续）

- 概念建模所坚持的是以数据为中心的理念，因为首先识别出的是概念，而概念的实现将是类而不是函数。
  - 概念建模的前三步（识别概念、识别关联、确定属性），是在充分借鉴了信息建模方法的基础上加以扩充后进行的（请大家自己进行与信息建模方法的对比）；
  - 第四步（产生系统序列图）是将需求建模结果反映到概念建模结果的过程;
  - 第五步（产生操作合约）则是将需求建模结果中的系统事件，落实到概念建模中的系统操作上，并标识出对系统状态的作用。
- 请大家自己确认，到目前为止所涉及到的是 **Use Case** 各范畴中的什么层次。

下一次课的内容

- 面向对象系统分析与设计 – 面向对象系统分析
  - 非功能需求概览
- 面向对象系统分析与设计 – 面向对象系统设计
  - 系统设计的必要性
  - 面向对象系统设计的主要活动
  - 必须重视软件体系结构
  - 特定域软件体系结构（DSSA）
  - 典型信息系统的软件体系结构
  - 典型的分布、并发控制结构
  - 模式
    - 体系结构模式
    - 设计模式