



第4章 面向对象的软件分析与设计

4.1 4+1模型及UML语言实现

4.2 面向对象的软件工程

4.3 用例图

4.4 活动图

4.5 用户界面设计

4.6 类图

4.7 交互图

4.8 包图

4.9 系统与子系统

4.10 部署图



第4章 面向对象的软件分析与设计

4.7交互图(Interaction Diagrams)

4.7.1 建交互图的原因

4.7.2 概念

4.7.3 顺序图(Sequence Diagrams)

4.7.4 通信图(Communication Diagrams)

4.7.5 应用场合

4.7.6 Case-图书管理系统交互图建模

4.7.7 完善类图

4.7交互图(Interaction Diagrams)

4.7.1建交互图的原因

■ 例：用户登录用例

❖ 需求分析(用户提供)

1. 用户登录用例界面：



2. 用户登录用例描述：

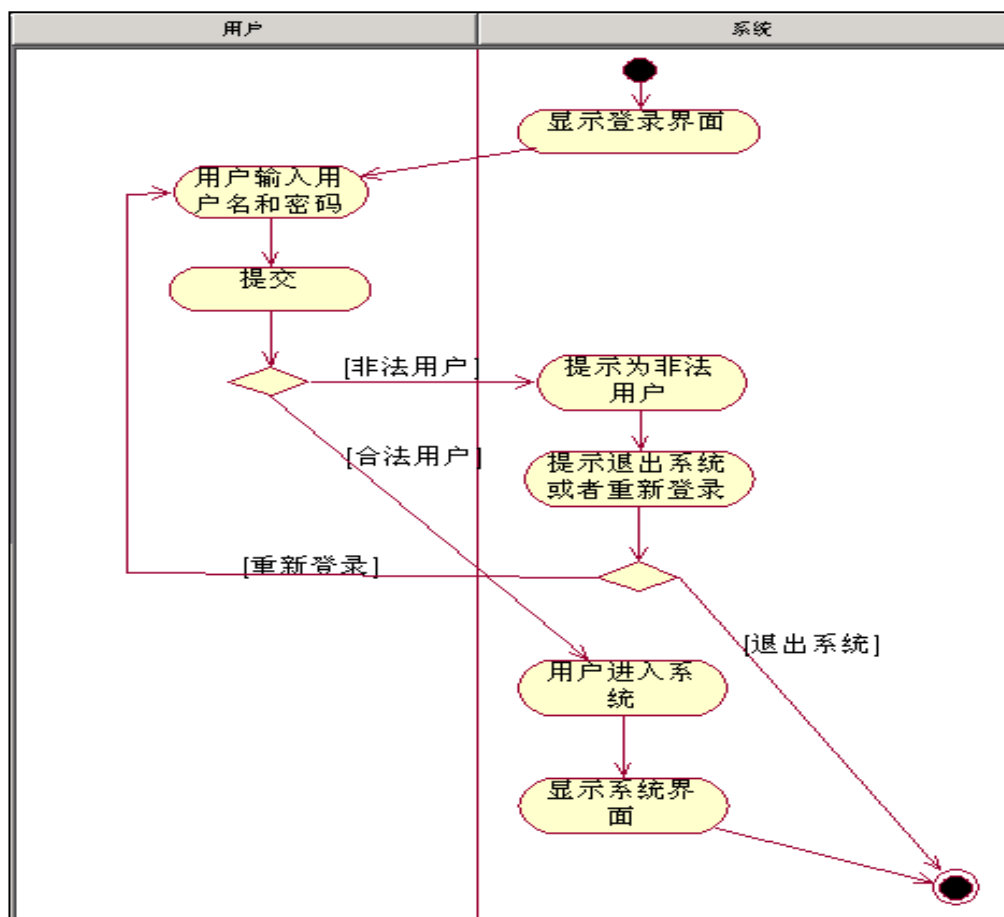
- (1) 用户在登录界面输入用户名、密码。
- (2) 如果登录失败，则显示：用户名或密码错误。
- (3) 如果登录成功，则进入系统界面。

4.7.1 建交互图的原因

■ 例：用户登录用例

❖ 需求分析(用户提供)

3. 用户登录用例活动图：



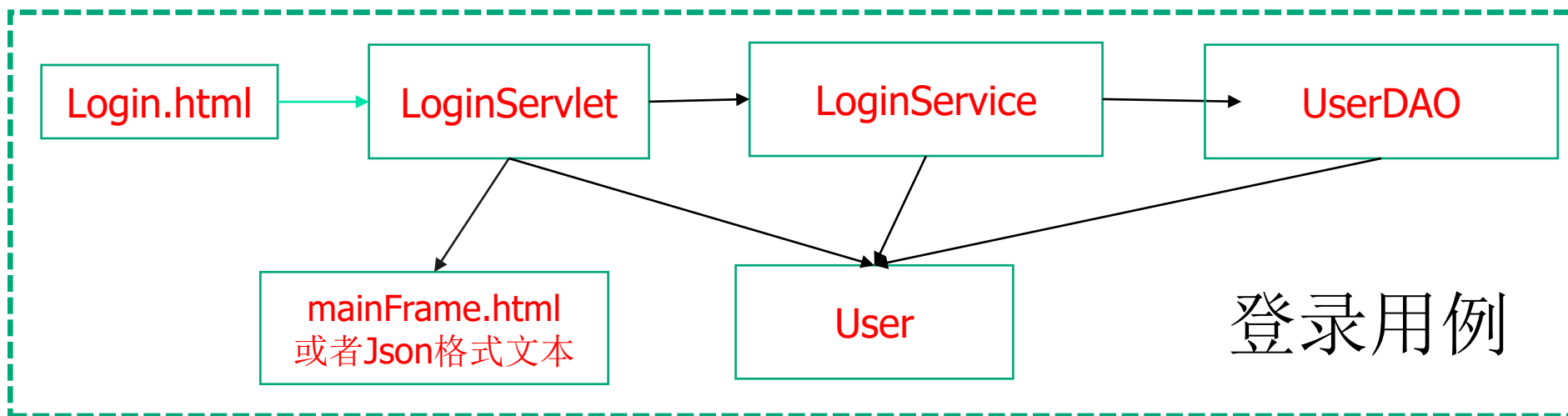
4.7.1 建交互图的原因

■ 例：用户登录用例

❖ 软件设计

网页登录用例(login)领域建模

JavaEE Web软件体系结构风格





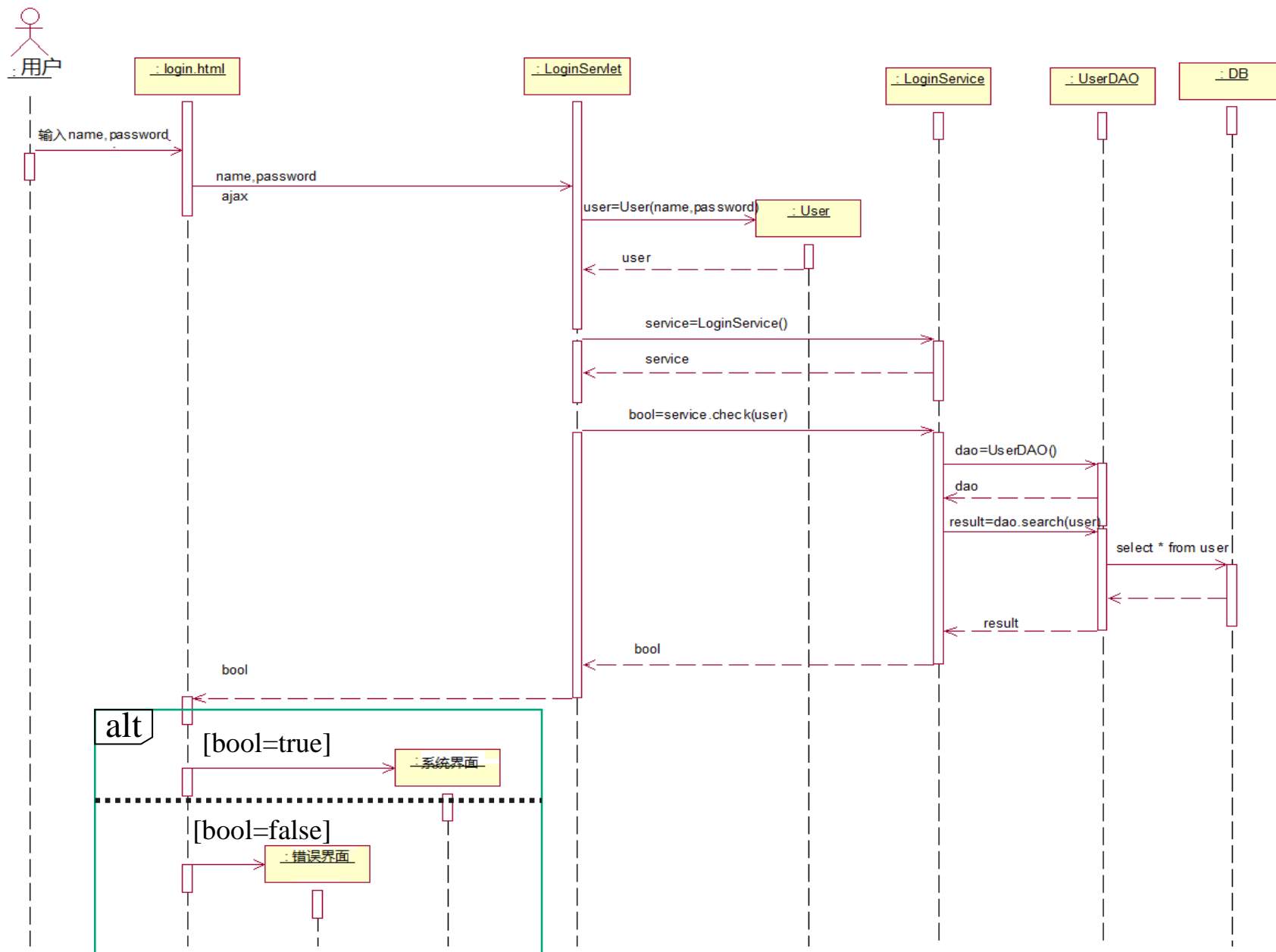
4.7交互图(Interaction Diagrams)

4.7.1建交互图的原因

■ 例： 用户登录用例

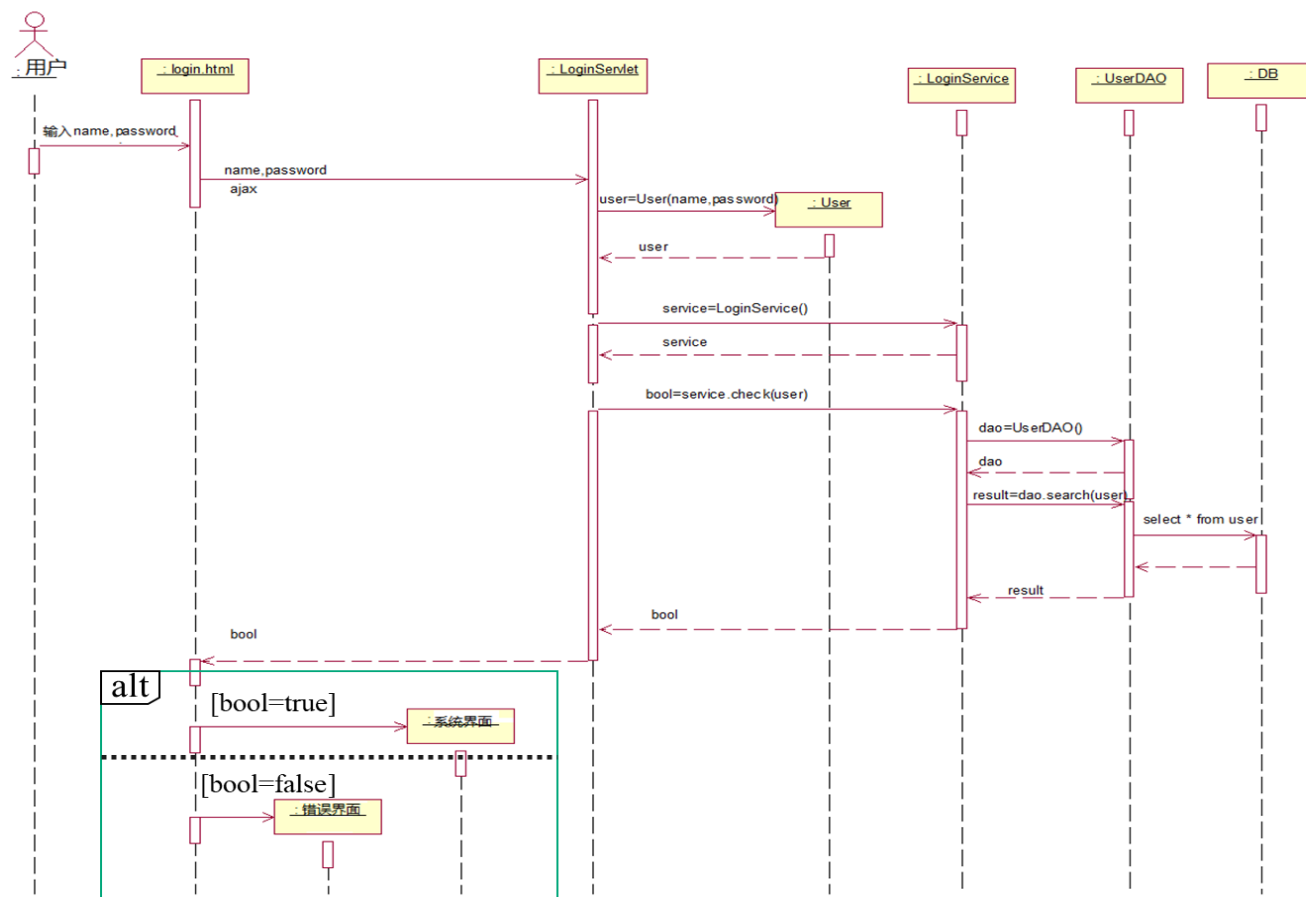
❖ 上述类图中的类如何**协作**完成用例（场景）的执行过程？

● 交互图



4.7.1 建交互图的原因

- 用例（场景）的执行过程
- 系统的动态部分



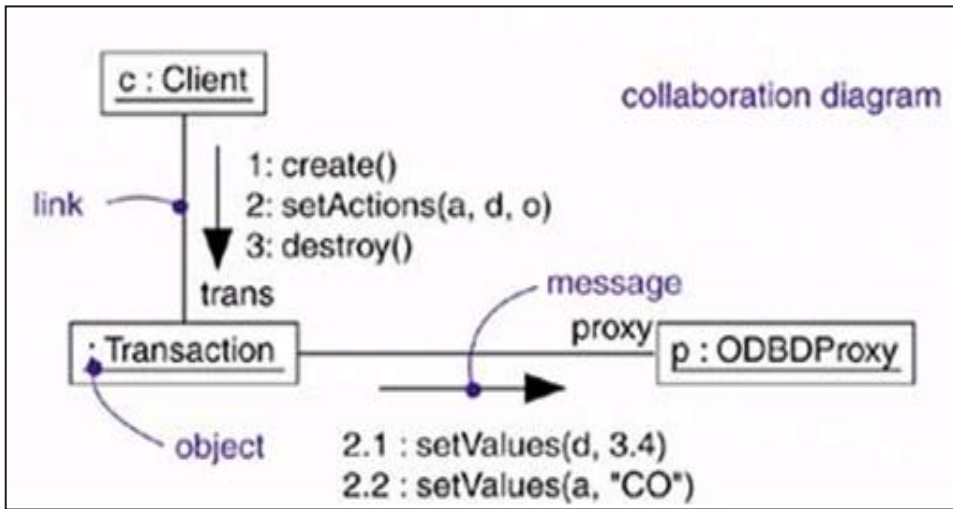
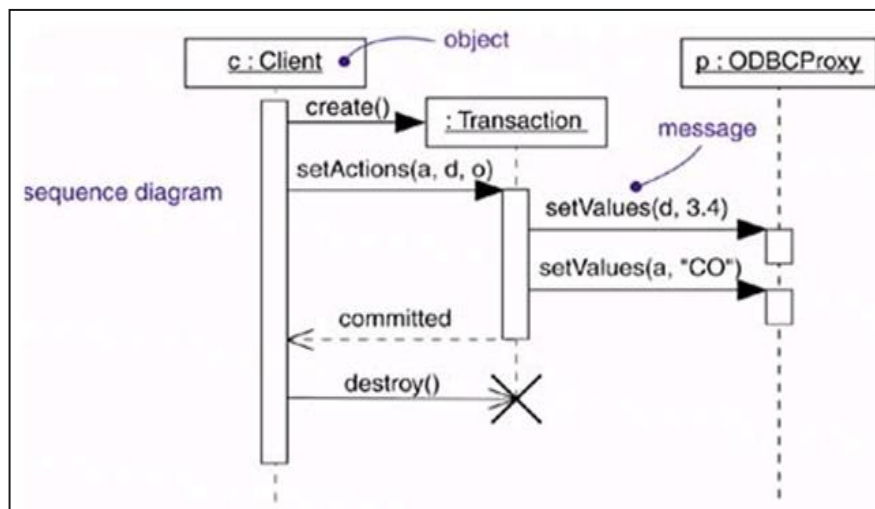
4.7.1 建交互图的原因

■ 交互图分类

❖ 顺序图(sequence diagram) : 消息的先后次序

❖ 通信图(communication diagram) :

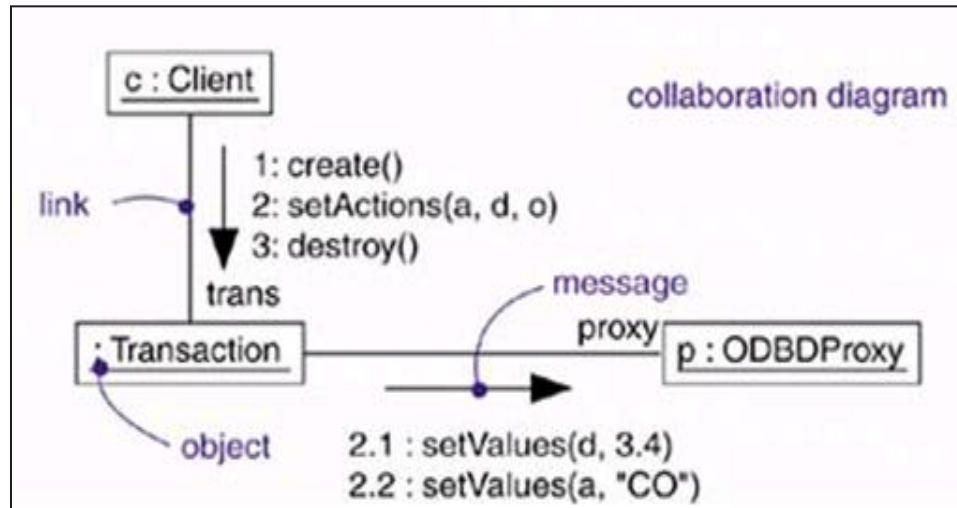
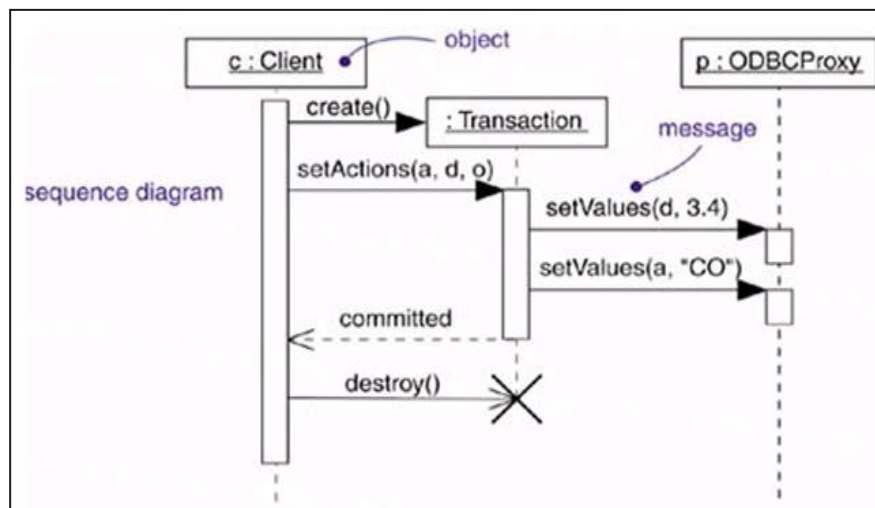
- 又称协作图collaboration diagram
- 发送消息的对象间的结构关系.



交互图

4.7.1 建交互图的原因

- 顺序图与通信图:语义一致
可以相互转换



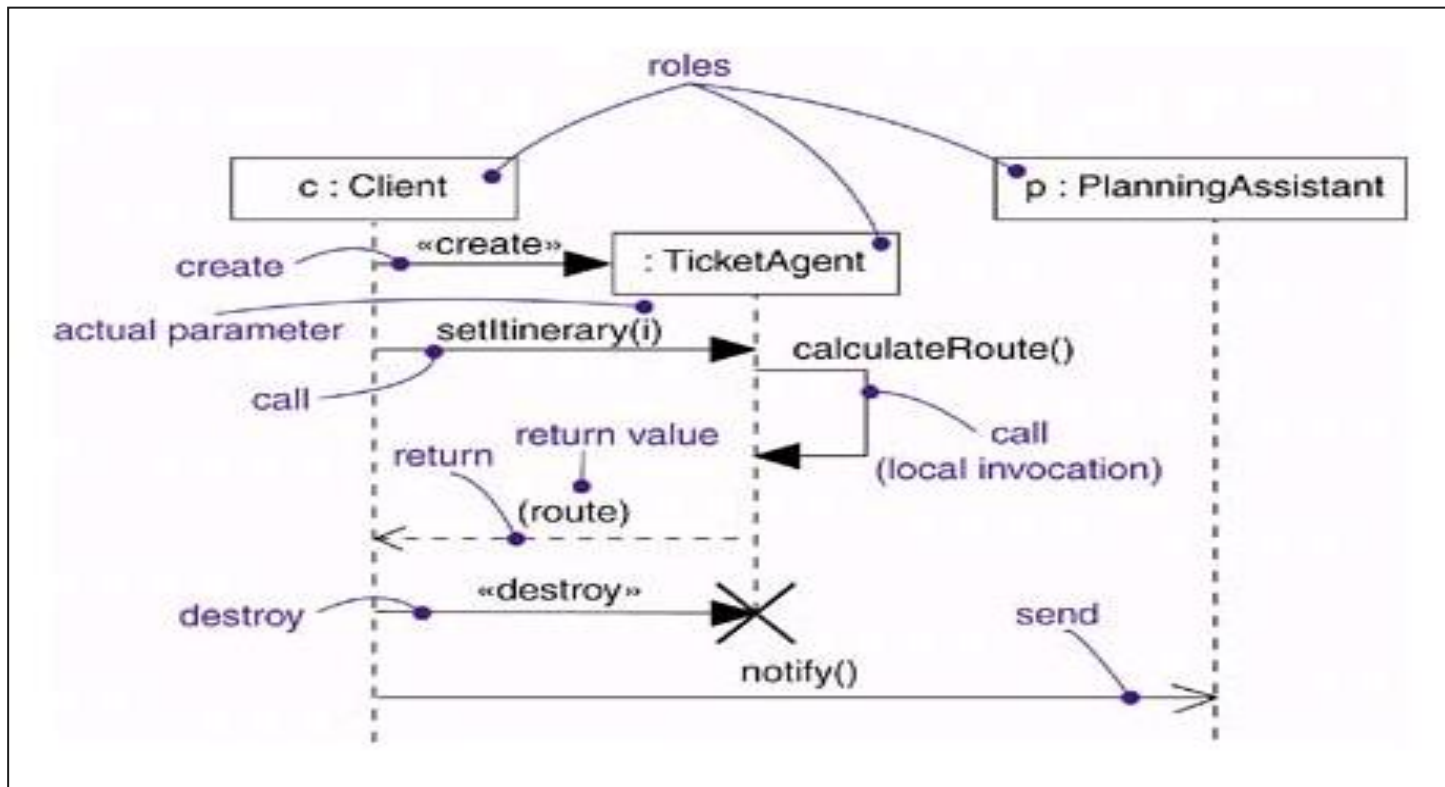
- 视图上的区别.
 - ❖ 顺序图: 建模对象的生命线(lifeline)。
 - ❖ 通信图: 建模对象间的结构关系。

4.7交互图(Interaction Diagrams)

4.7.2概念

■ 交互(interaction):

在一个上下文(context)中，**对象**为了完成某项任务而相互发送消息，这个**发送和接收消息的过程**就是交互(interaction)。

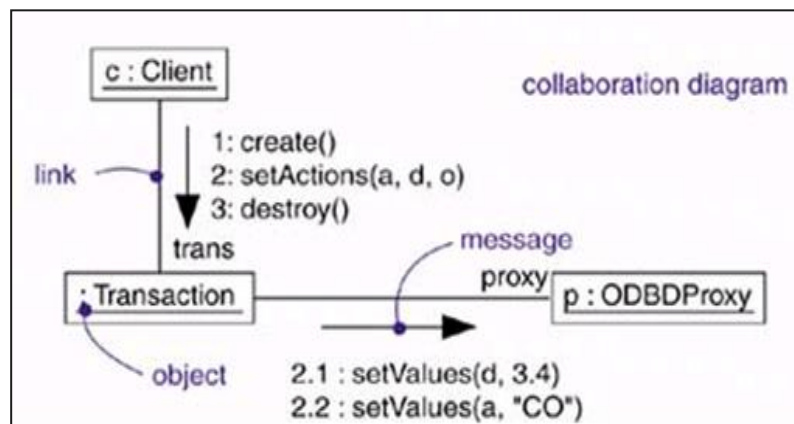
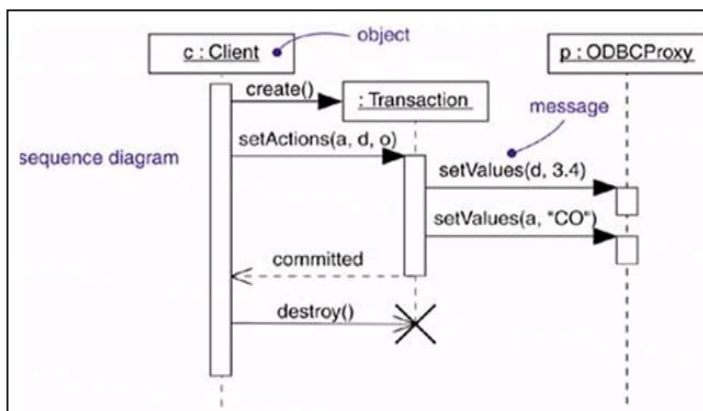


4.7.2概念

交互(interaction)

❖ 交互可能发生于

- 系统 或子系统的上下文背景下的**对象间的合作**。



- 某项操作(operation)的上下文背景下。

❖ 交互可能存在于

某个类(class)的上下文背景下。

4.7.2概念

■ 对象(Objects)与角色 (Roles)

对象：实体

角色： A prototypical object

例：

p $\xrightarrow{\text{类 Person 的具体实例}}$ 某个具体的Person

p $\xrightarrow{\text{原型对象}}$ Person的任何实例



4.7.2概念

■ 消息

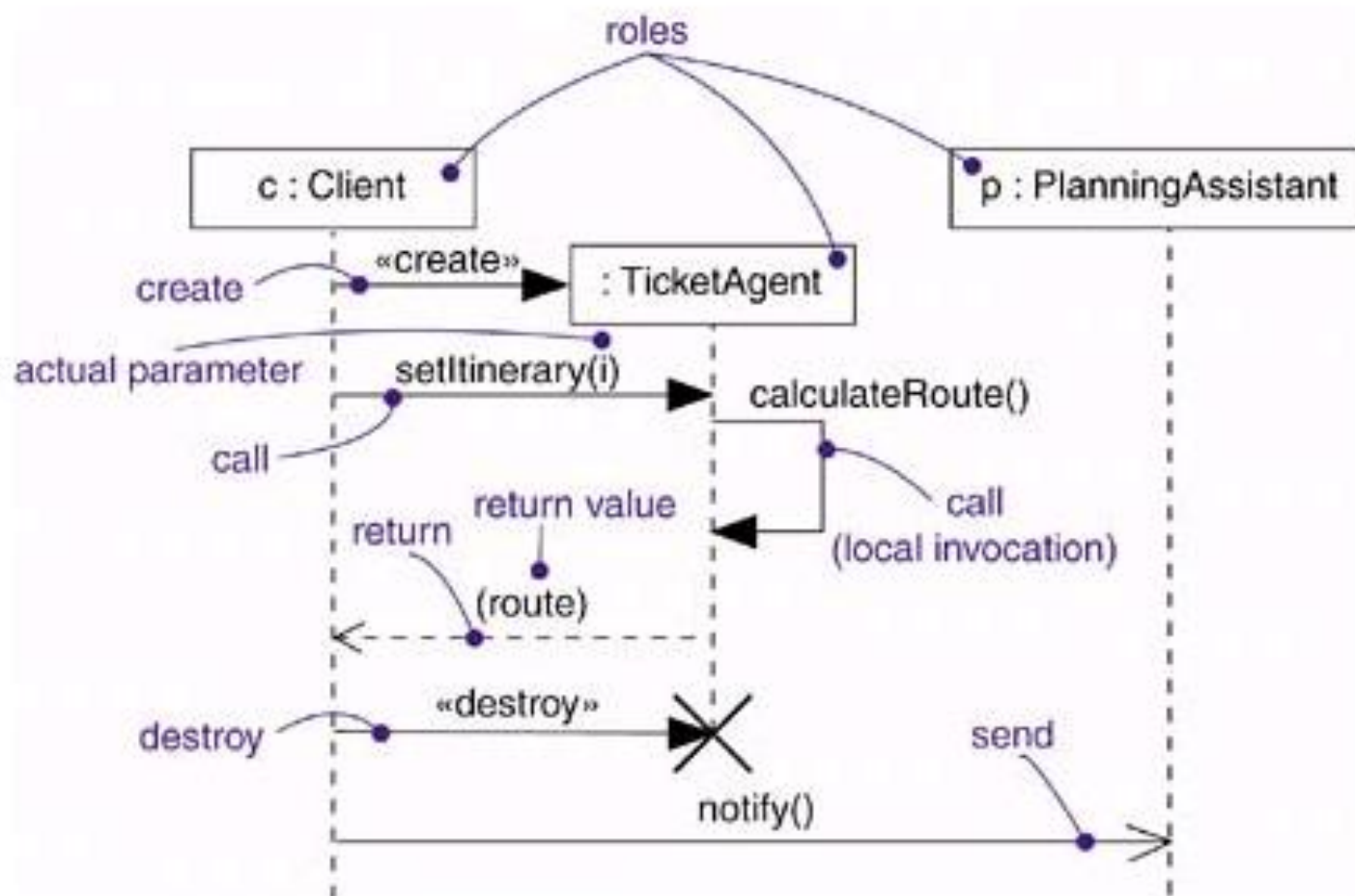
❖消息是对象间的通信规范。

❖UML中的消息种类

Call	调一个对象的操作。
Return	返回一个值给调用者
Send	给一个对象发送信号
Create	创建一个对象。
Destroy	删除一个对象

4.7.2概念

消息



4.7.2概念

■ 序号

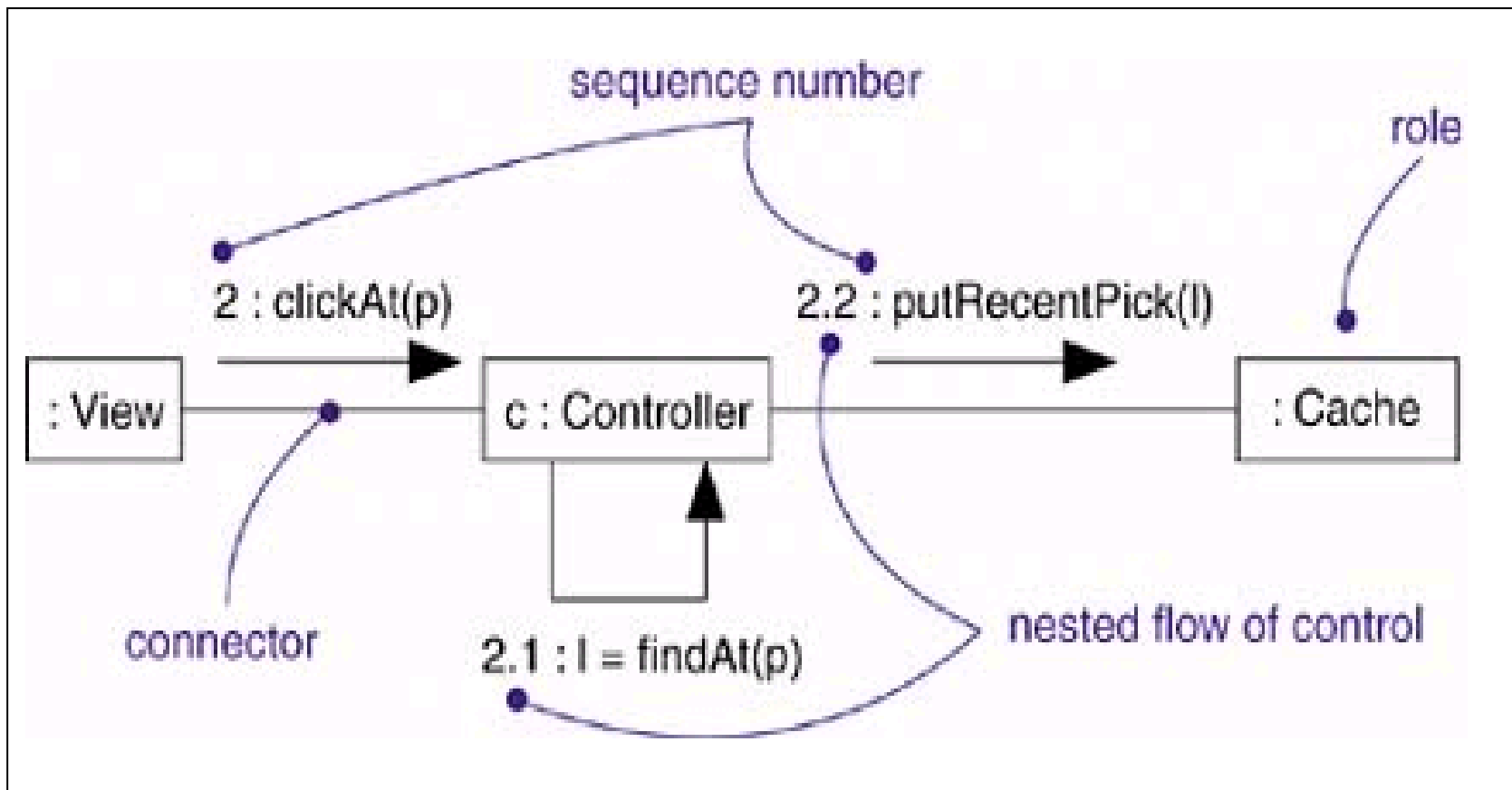
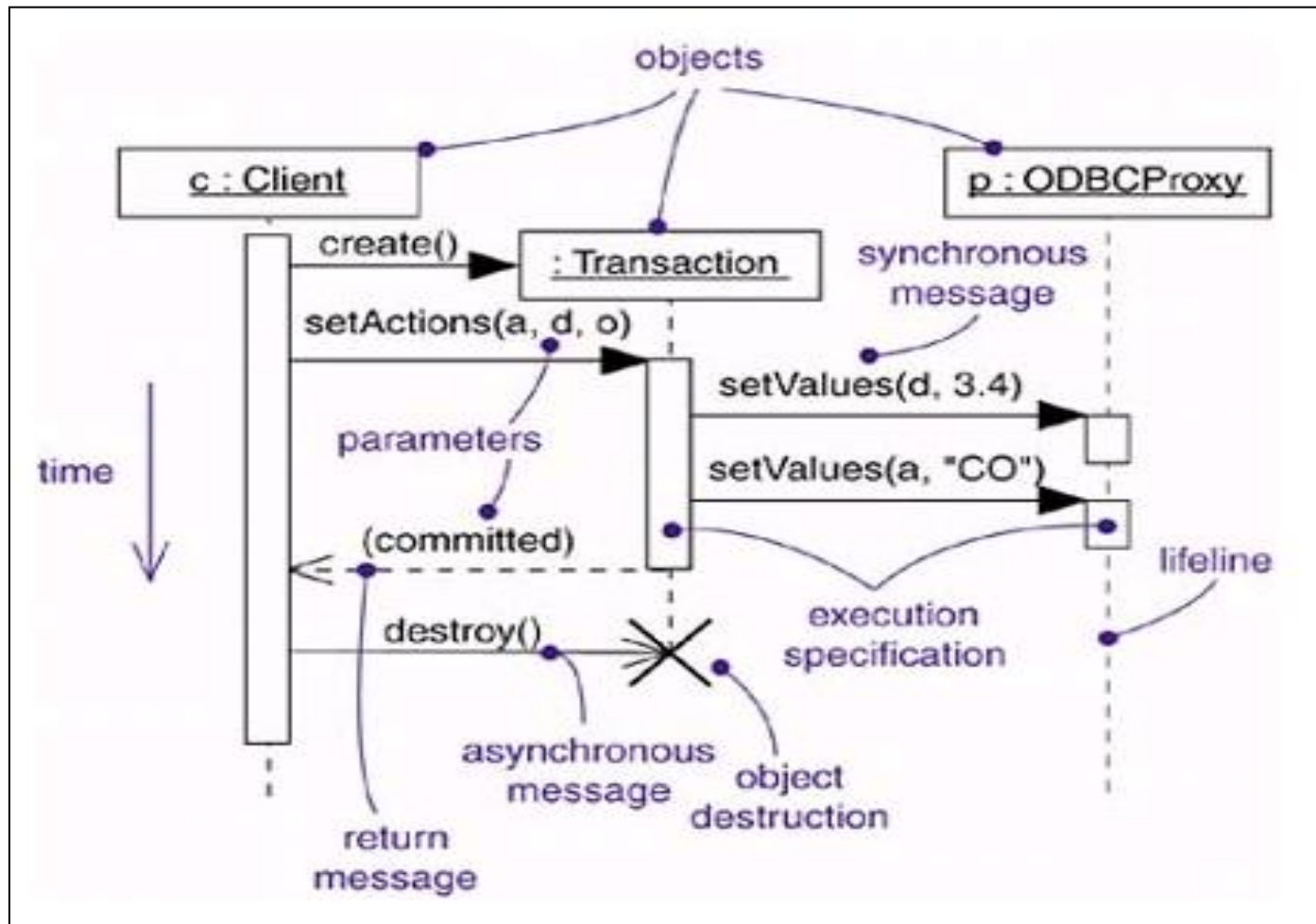


Figure . Sequence

4.7交互图(Interaction Diagrams)

4.7.3 顺序图(Sequence Diagrams)

顺序图重点在消息的先后次序。



顺序图

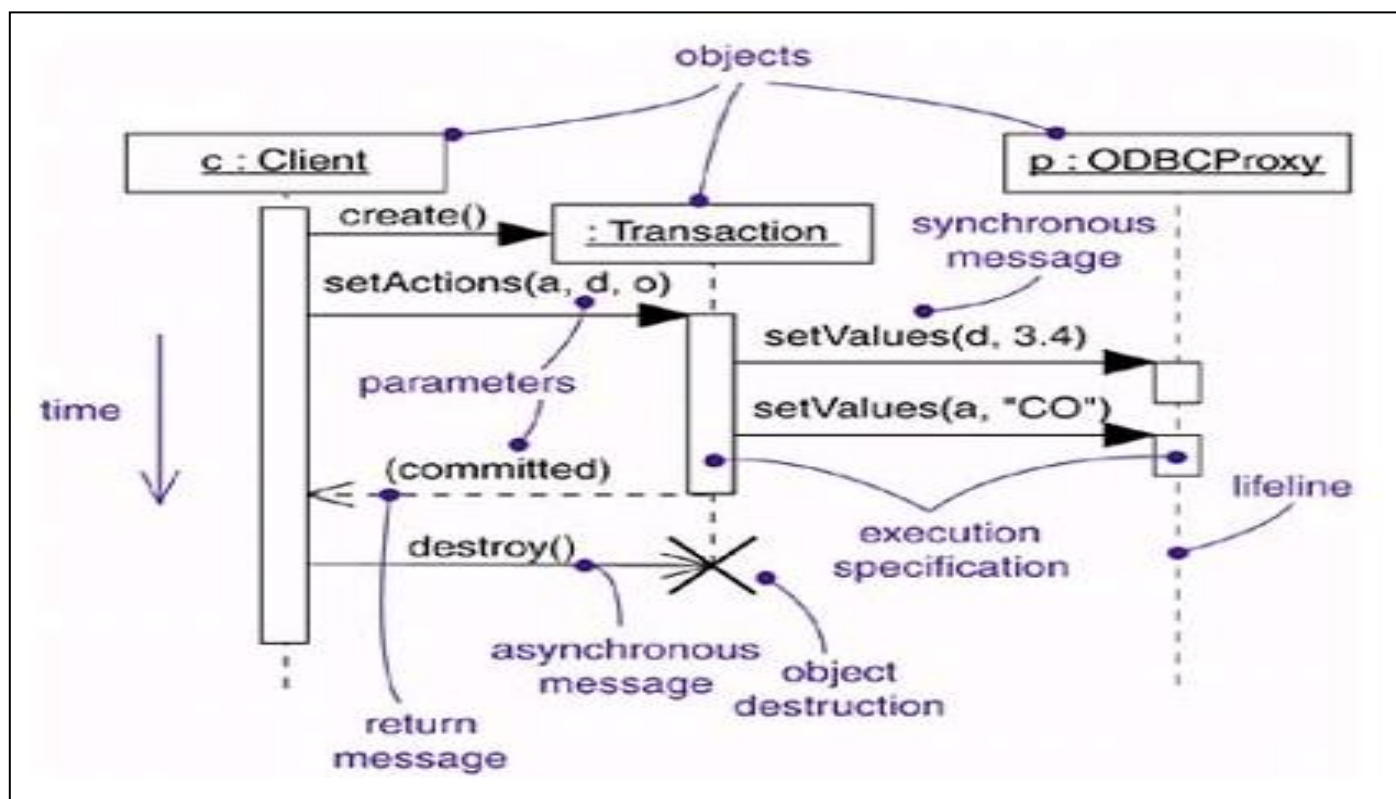
4.7.3 顺序图(Sequence Diagrams)

顺序图主要内容: messages.

❖ 异步消息(asynchronous message):

❖ 同步消息synchronous message (a call):

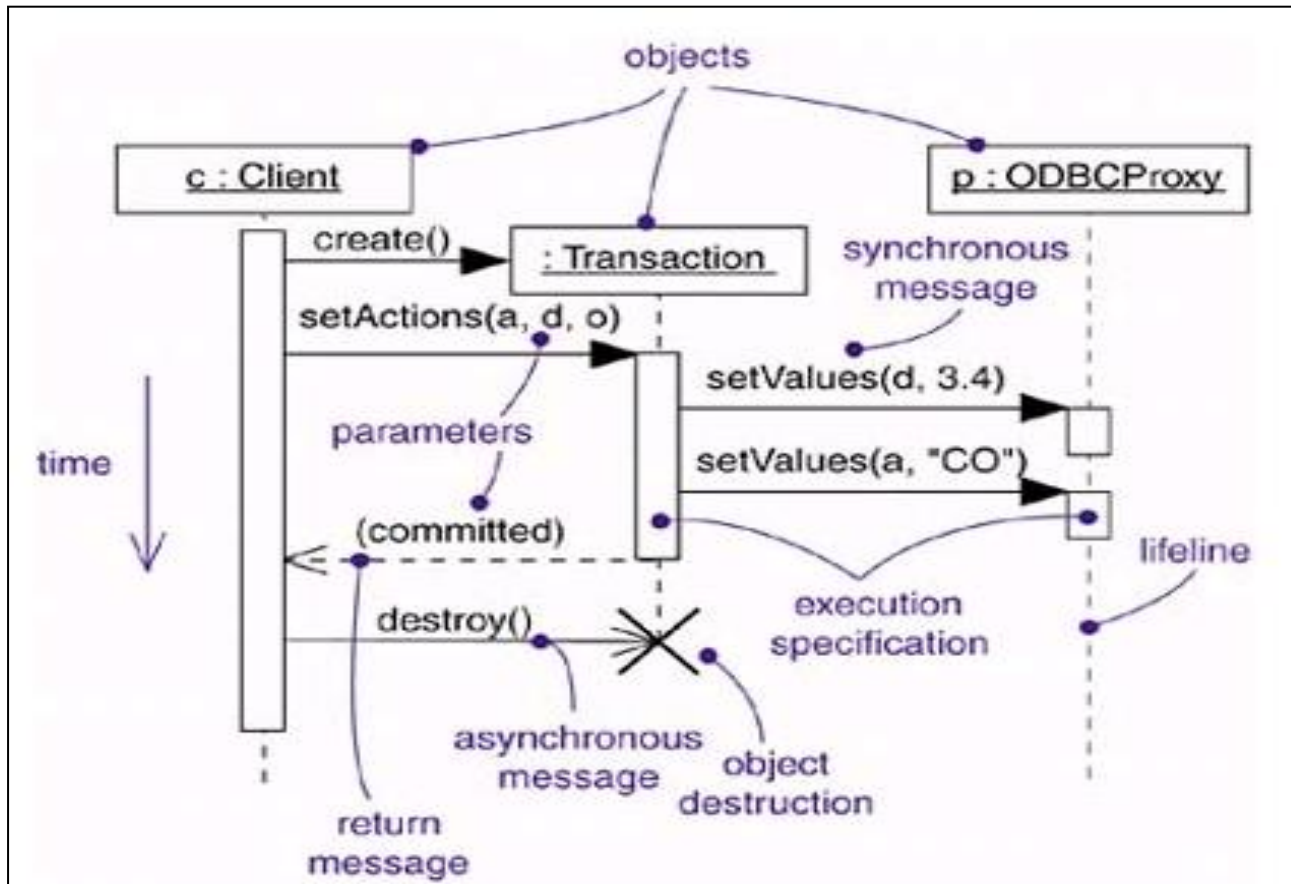
● 返回消息: 同步消息的返回信息



4.7.3 顺序图(Sequence Diagrams)

顺序图与通信图的不同处

- ❖ the lifeline.
- ❖ the focus of control.

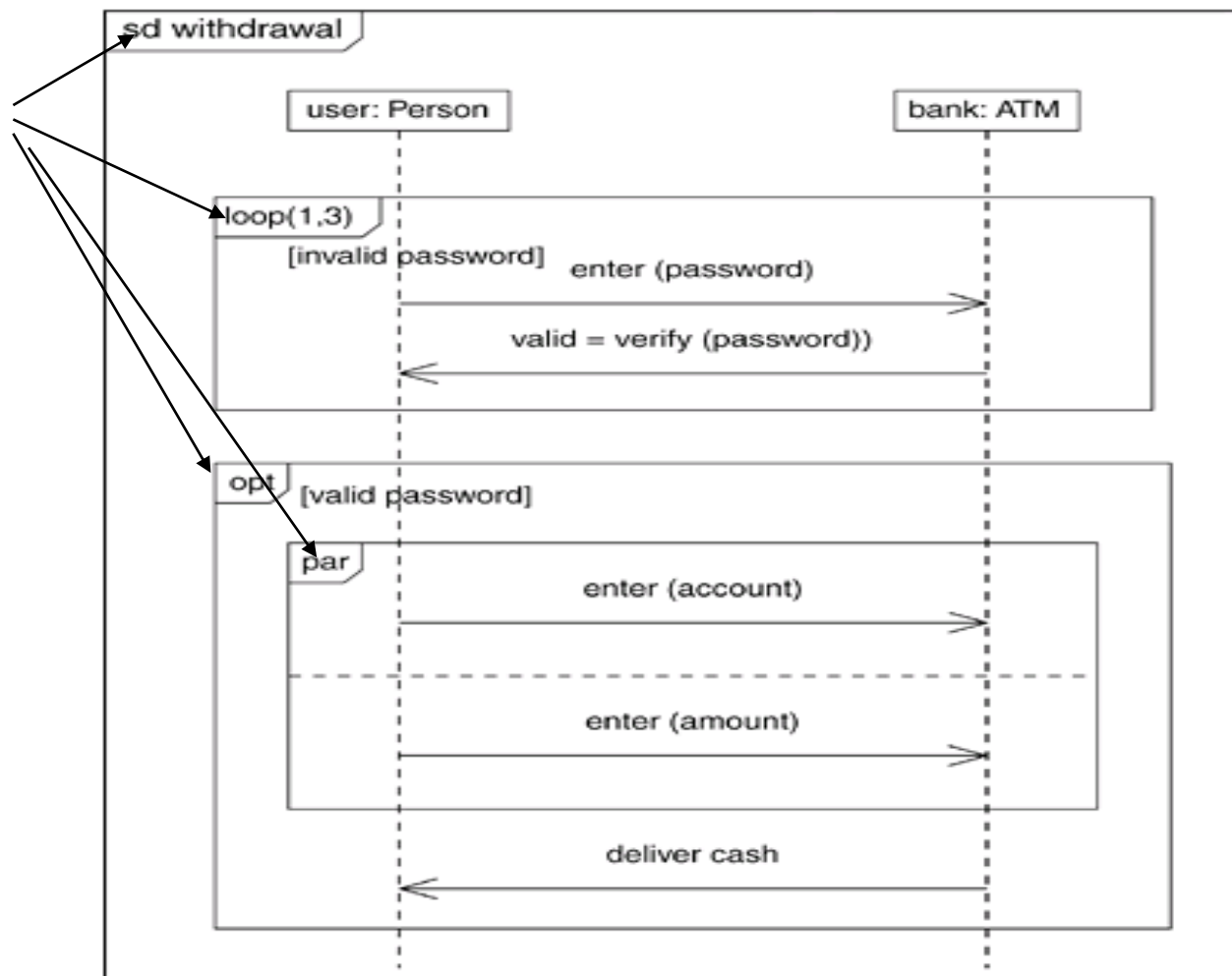


4.7.3 顺序图(Sequence Diagrams)

■ 顺序图中的结构化语句

❖ 条件与循环

operator

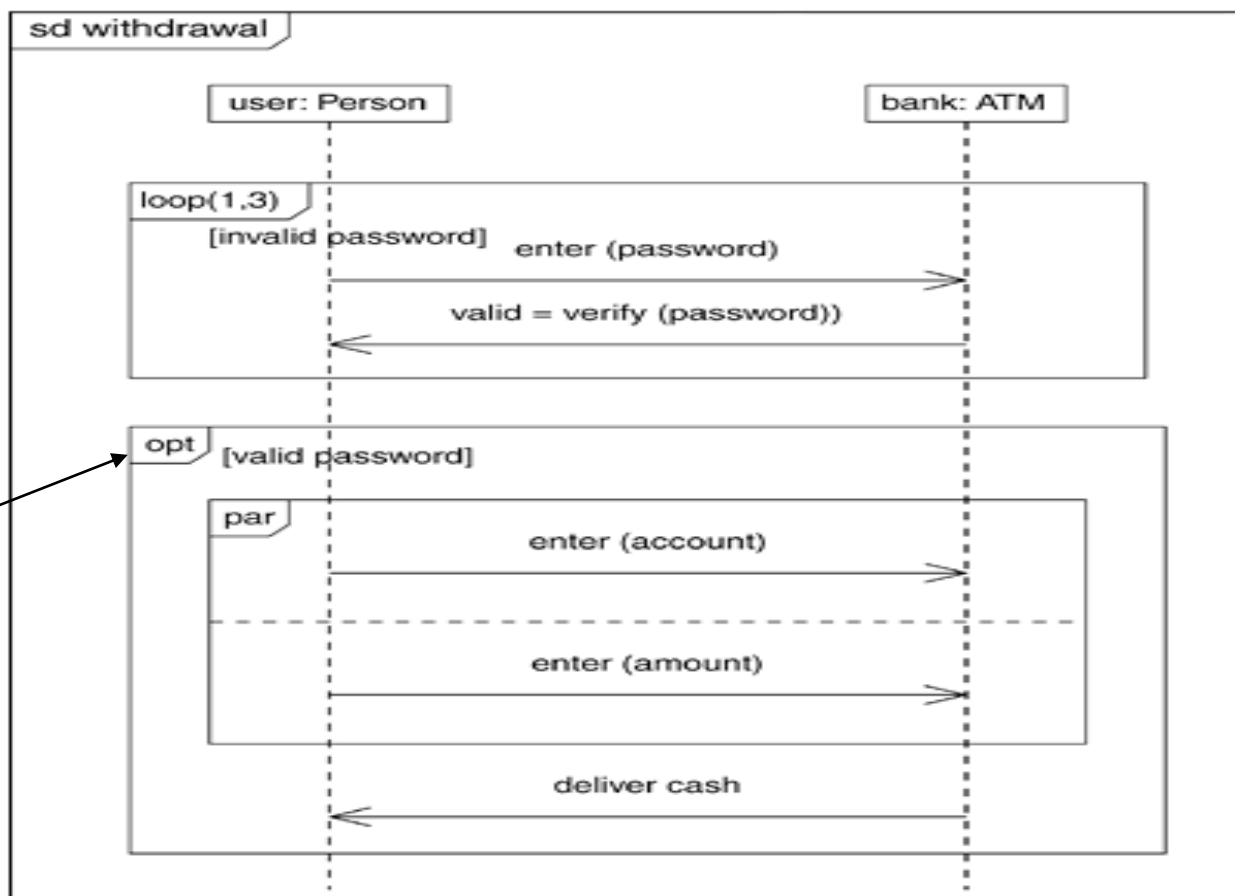


4.7.3 顺序图(Sequence Diagrams)

顺序图中的结构化语句

❖ 选择:opt

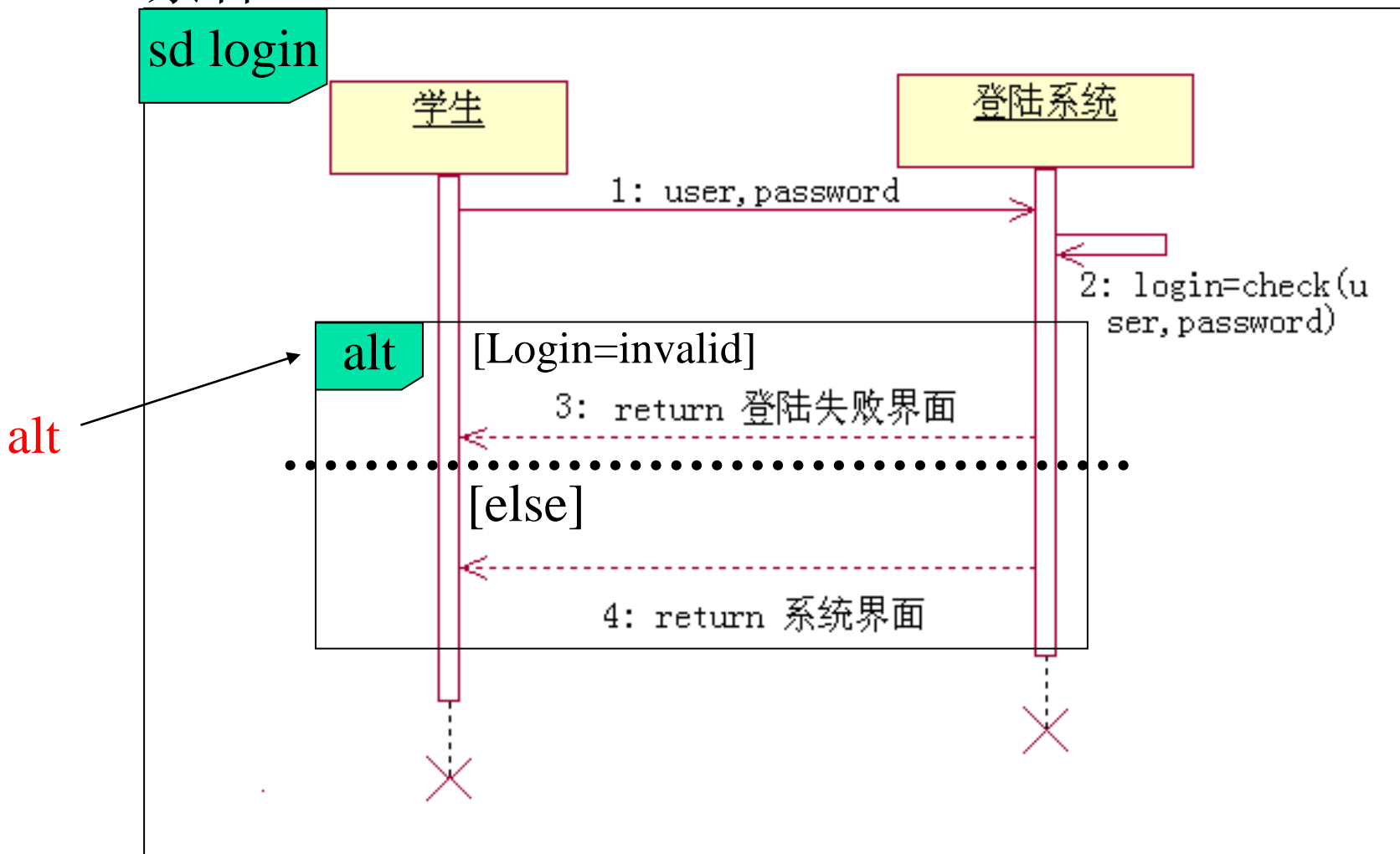
如果 **guard condition** 为真，则执行相应的语句。



4.7.3 顺序图(Sequence Diagrams)

顺序图中的结构化语句

❖ 条件:alt.

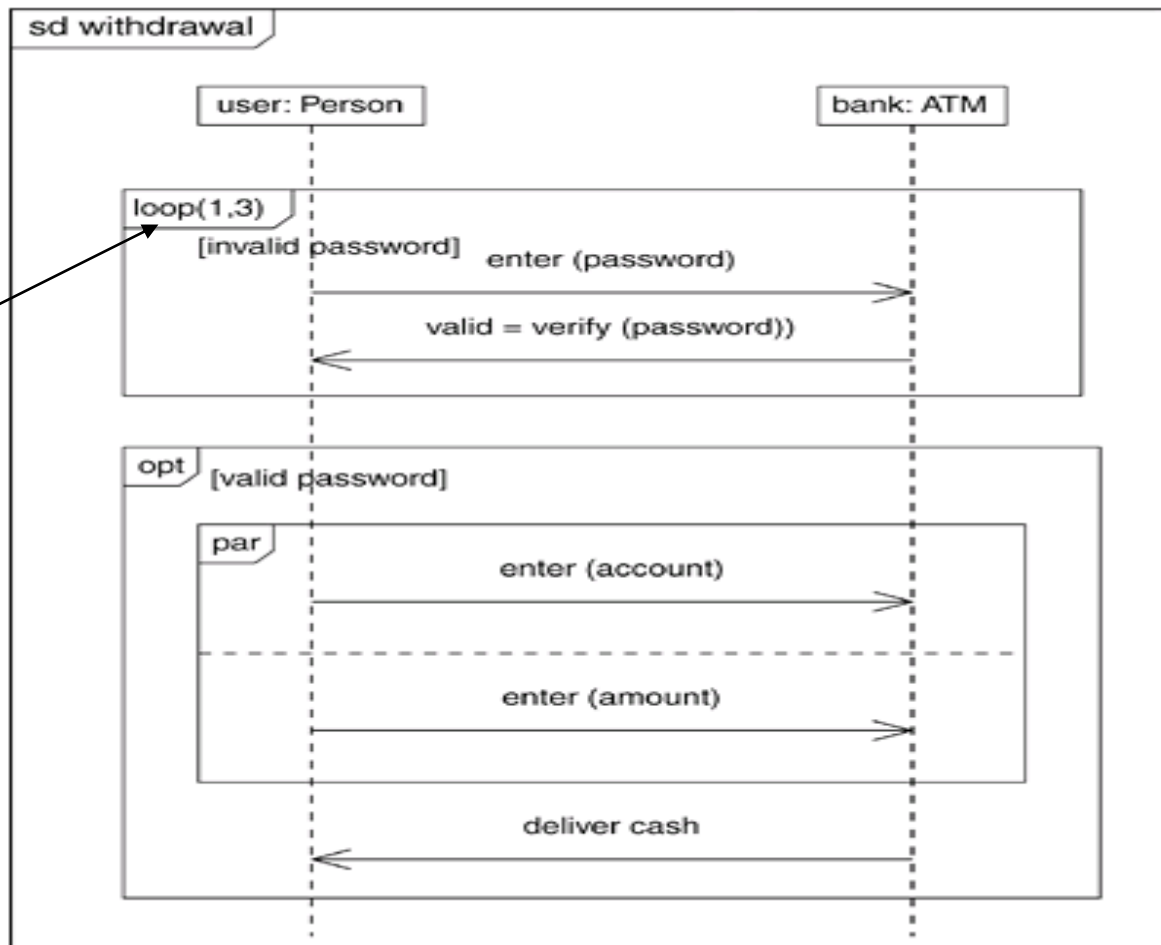


4.7.3 顺序图(Sequence Diagrams)

■ 顺序图中的结构化语句

❖ 循环:loop

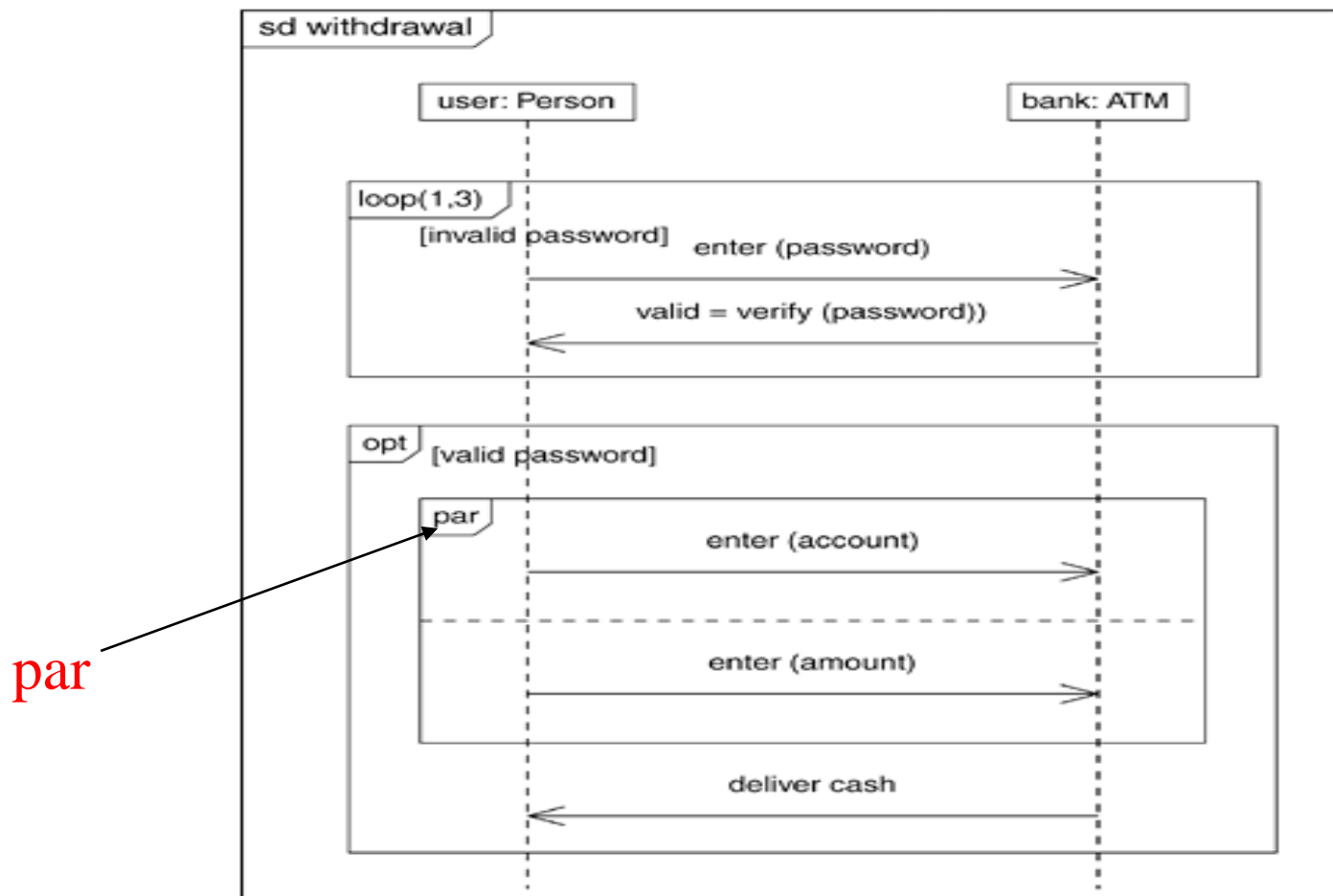
loop



4.7.3 顺序图(Sequence Diagrams)

■ 顺序图中的结构化语句

❖ 并发:par



4.7.3 顺序图(Sequence Diagrams)

■ 顺序图中的结构化语句

❖ 引用:ref

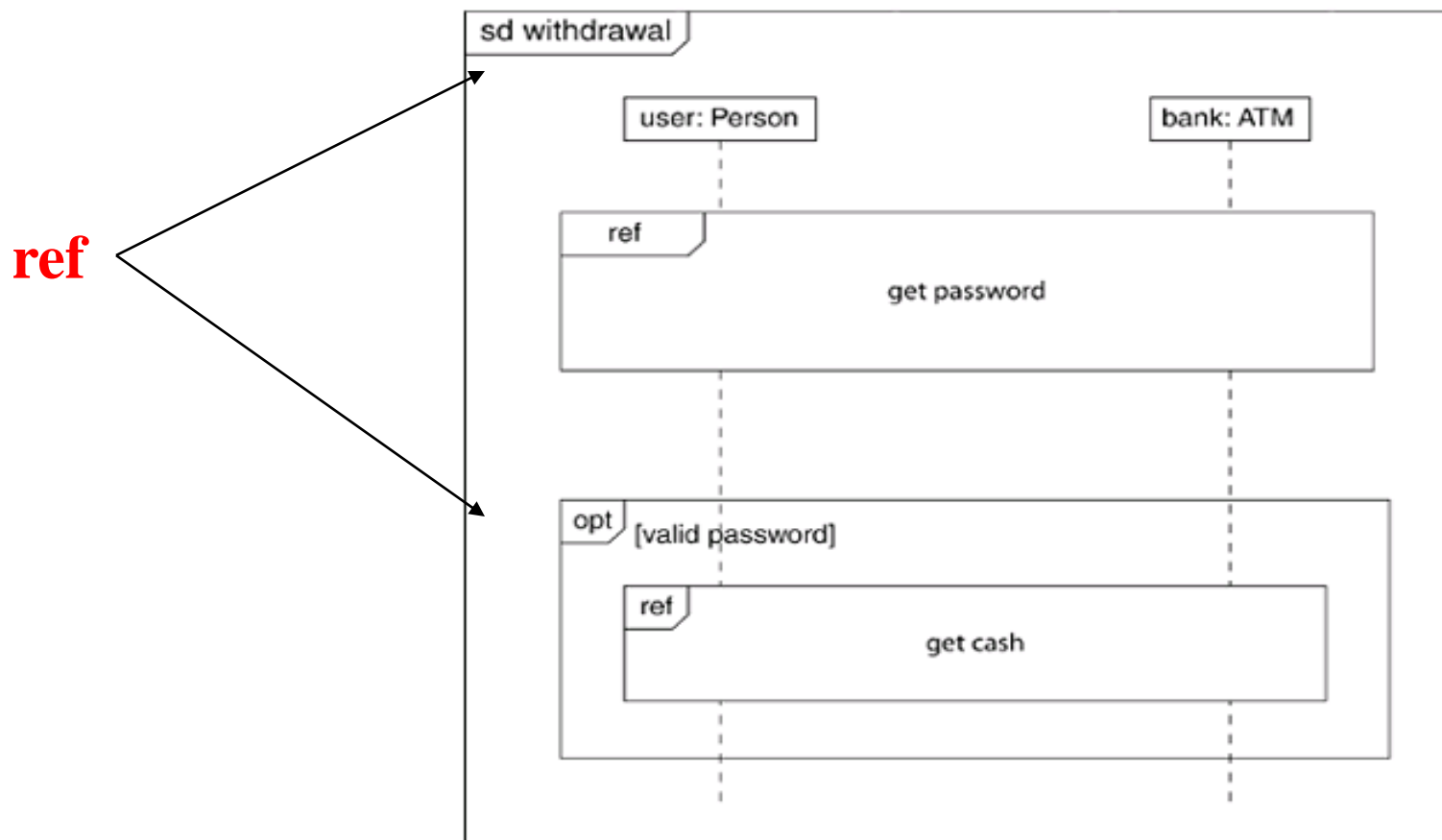
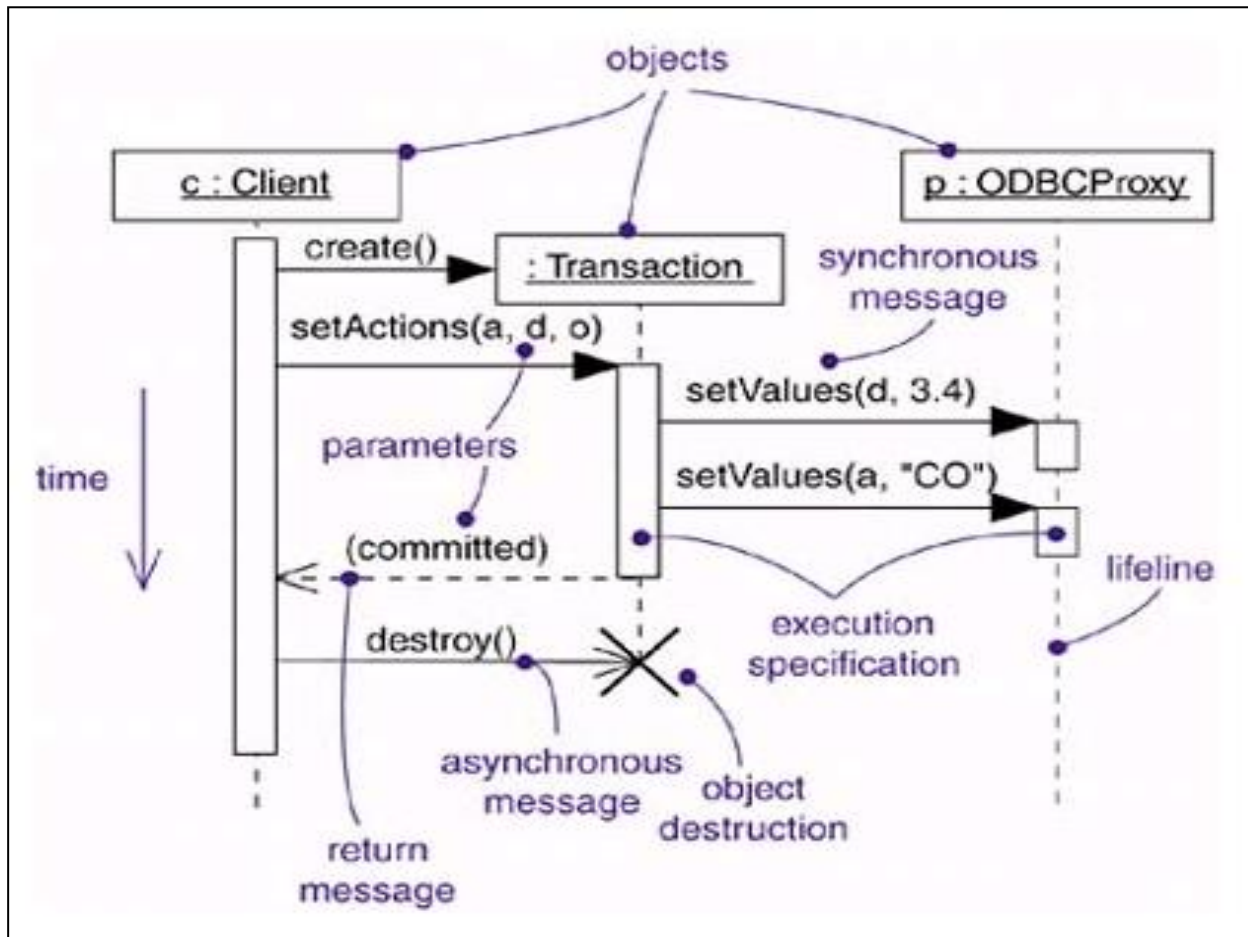


Figure . Nested Interaction diagram

4.7交互图(Interaction Diagrams)

4.7.4通信图(Communication Diagrams)



4.7.4通信图(Communication Diagrams)

- 着重的是建模对象间的组织结构关系。

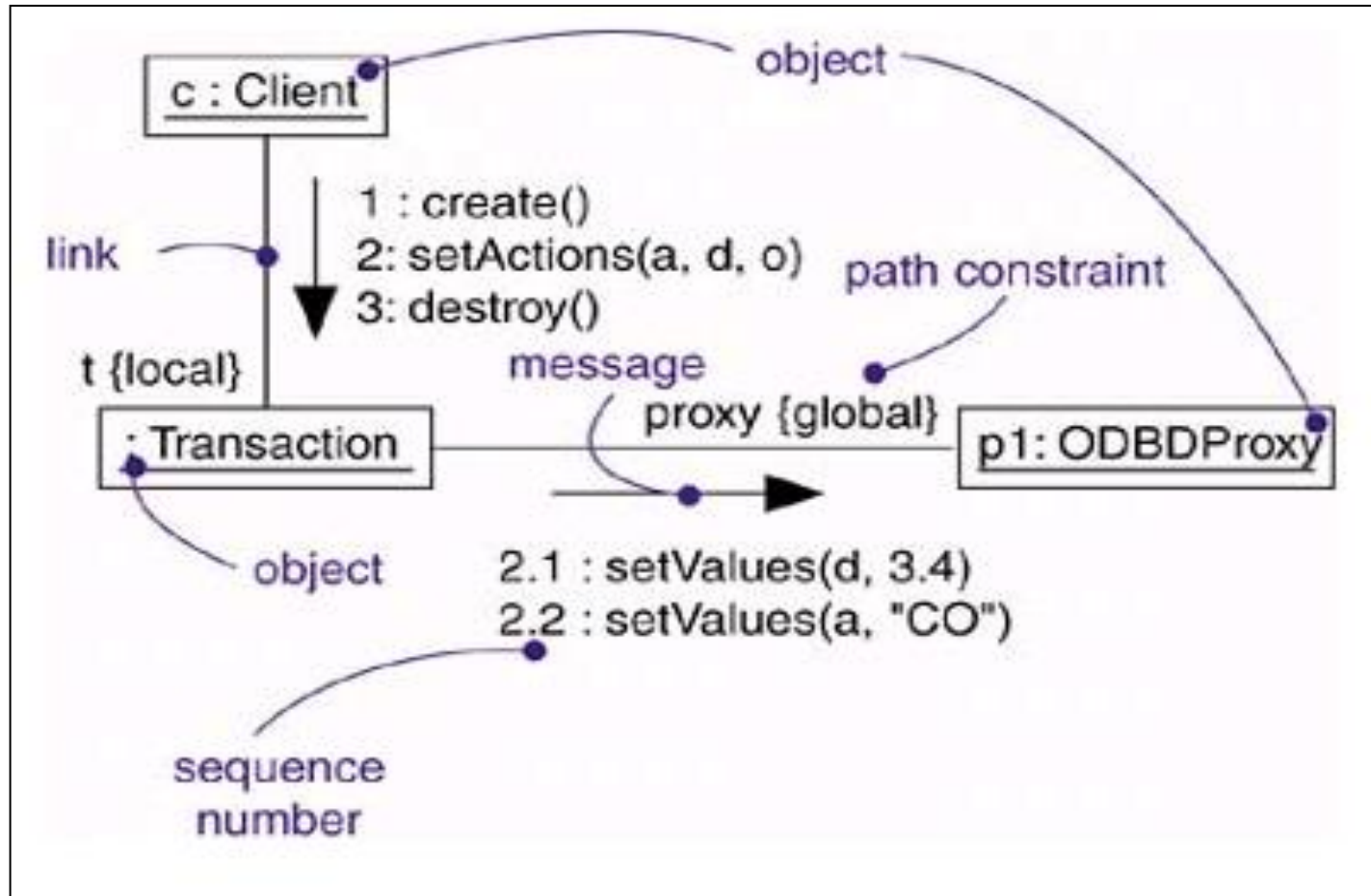


Figure . Communication Diagram

4.7.4通信图(Communication Diagrams)

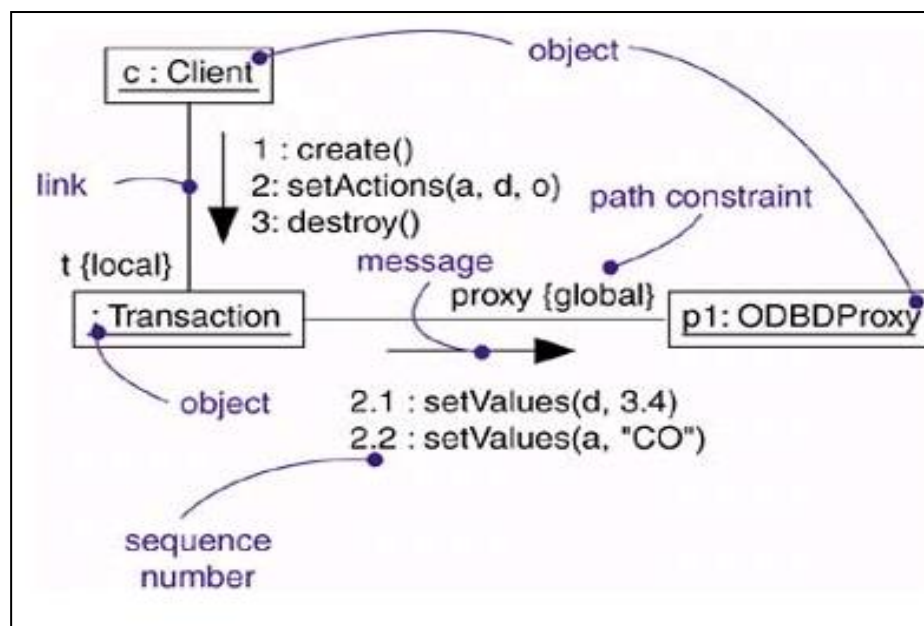
说明

❖ 区别于顺序图的二点:

- 存在 **path**.
- 有顺序号

❖ 语义相同

- 通信图表明的是对象间如何相连。
- 顺序图表明的是消息的传递。



4.7交互图(Interaction Diagrams)

4.7.5应用场合

- To model flows of control by time ordering

❖ 顺序图: use case scenario

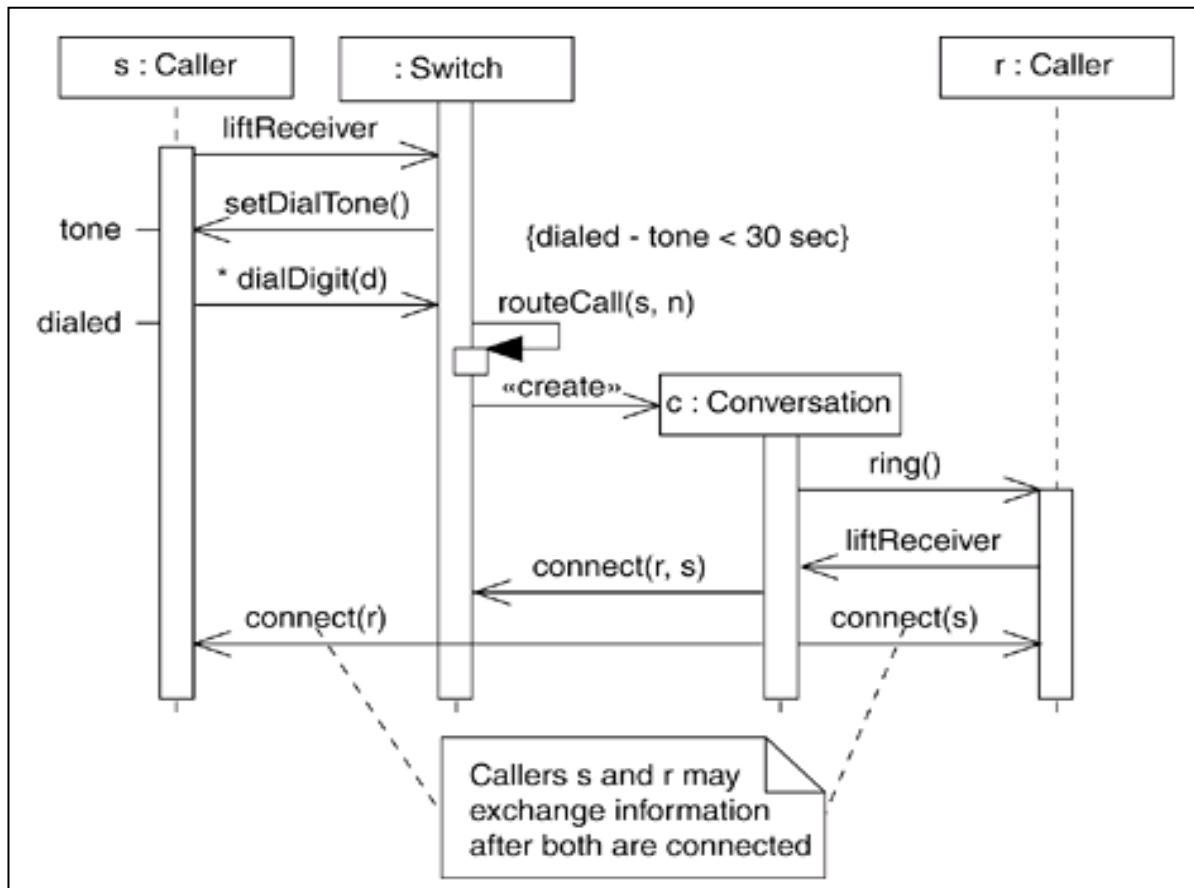
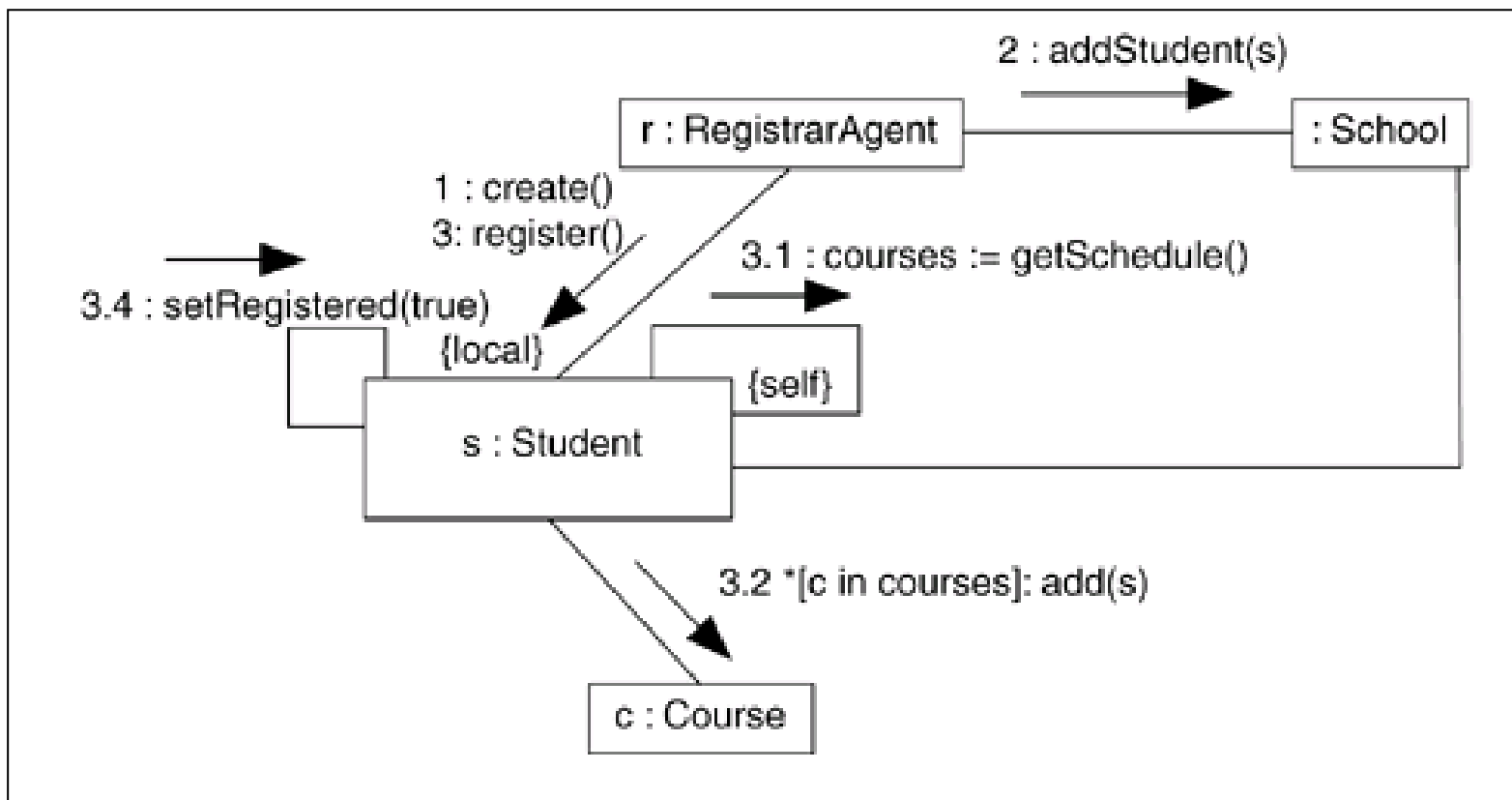


Figure .
initiating a
simple, two-party
phone call.

4.7.5 应用场合

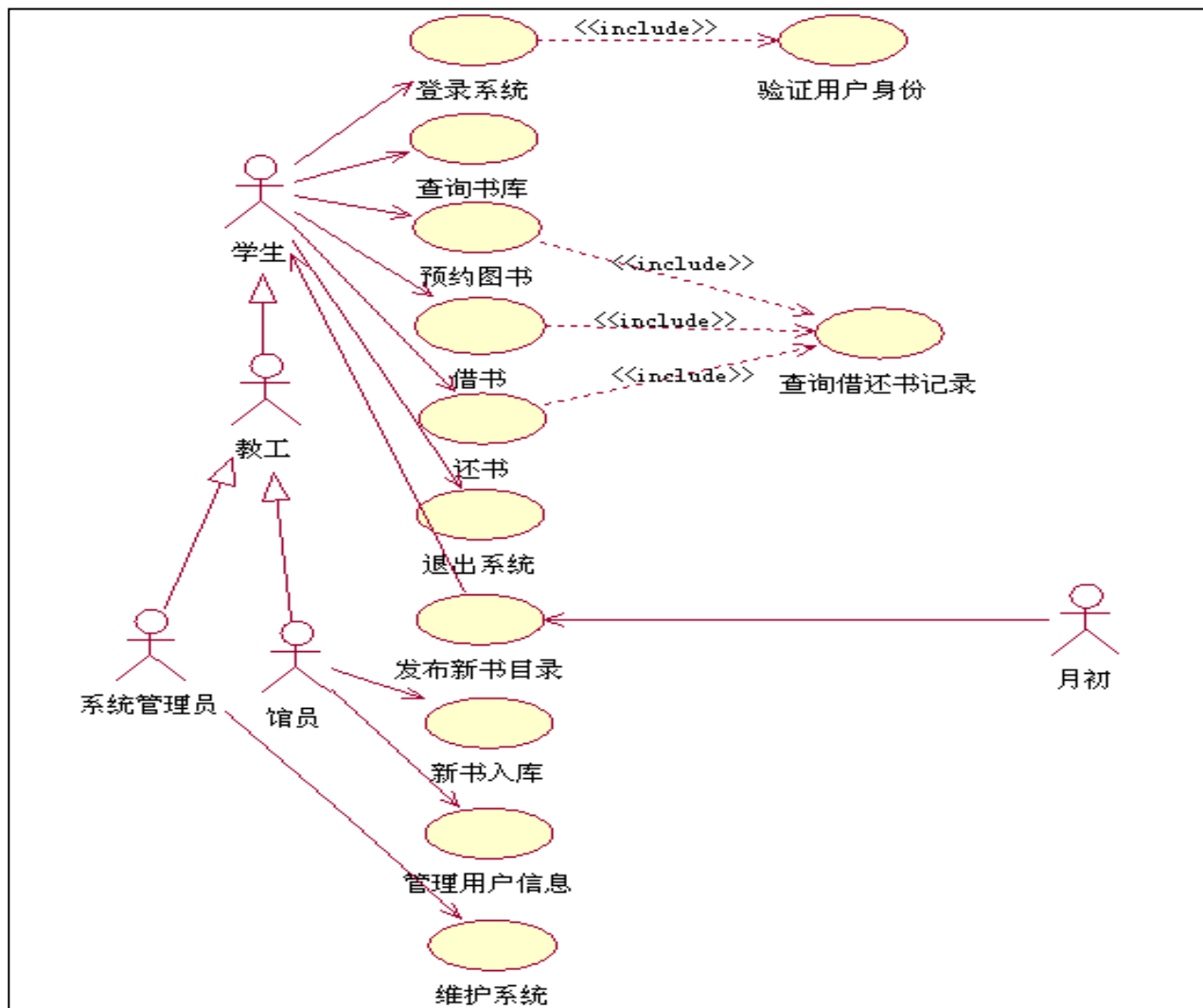
- To model flows of control by organization
- ❖ communication diagrams



学生入校注册

4.7交互图(Interaction Diagrams)

4.7.6 case-图书管理系统交互图建模





4.7.6 case-图书管理系统交互图建模

■ 对登录系统用例建交互图

- ❖ 用例的描述表现为执行者和系统之间的消息。
- ❖ 用例中的行为通常以执行者和系统对象之间的消息的形式表现。

4.7.6 case-图书管理系统交互图建模

对登录系统用例建交互图

前置条件:

事件流:

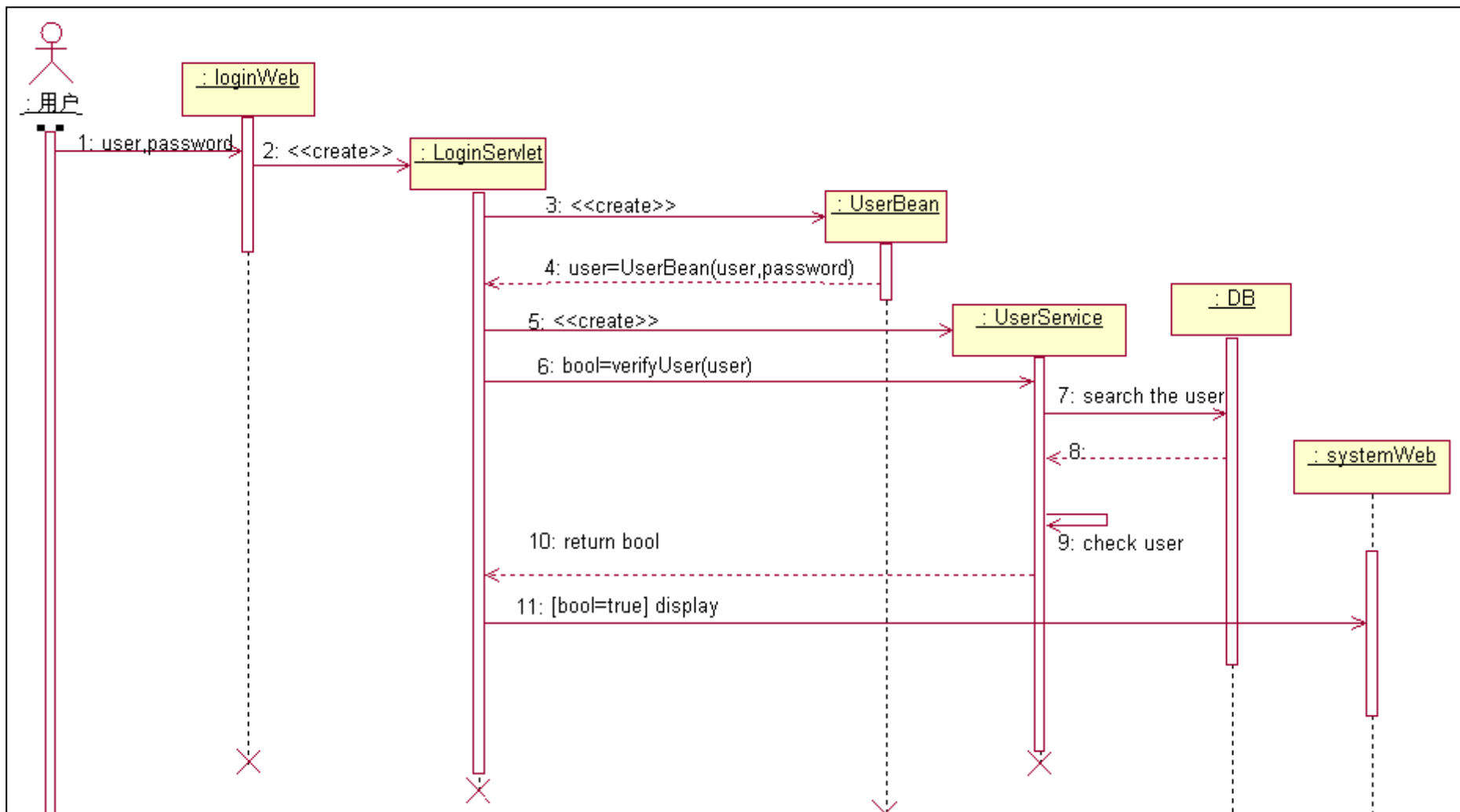
1. 当用户通过浏览器输入图书管理系统地址: library.ecust.edu.cn,用例开始。
2. 系统显示登录界面。
3. 用户输入用户名和密码。
4. 用户选择提交。
5. 系统检测用户信息,如是非法用户(**if---**),系统提示用户为非法用户,同时提示用户选择退出系统或者重新登录。
 - a) 如果选择重新登录,就回到第2步;
 - b) 如果选择退出系统,用例结束。
6. 系统检测用户信息,如是合法用户(**if---**),
 - a) 系统设置访问权限,并显示主界面,用例结束。

后置条件: 用户进入系统或者不能进入系统。

登录系统用例

4.7.6 case-图书管理系统交互图建模

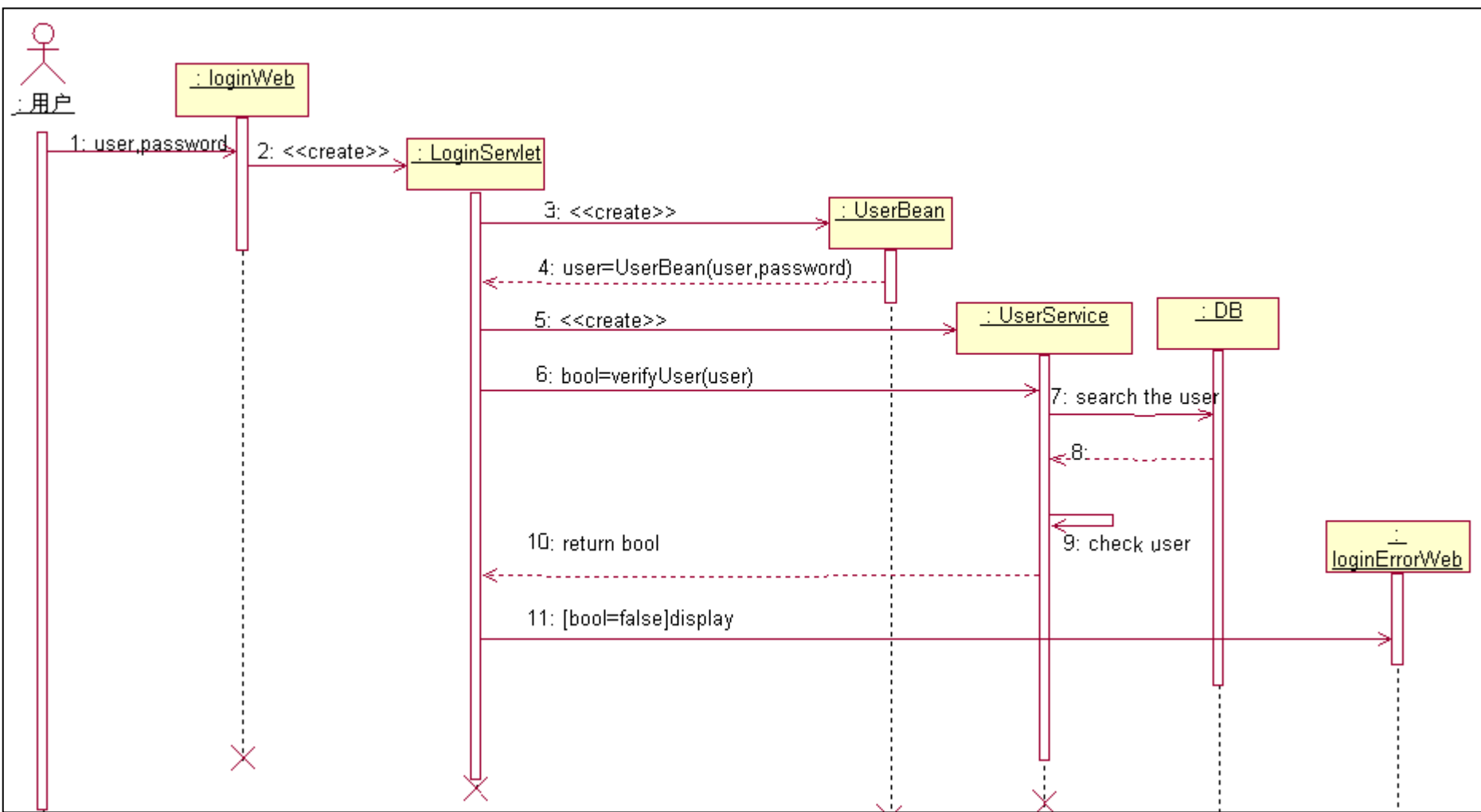
对登录系统用例建交互图



登录系统用例的场景1[成功]

4.7.6 case-图书管理系统交互图建模

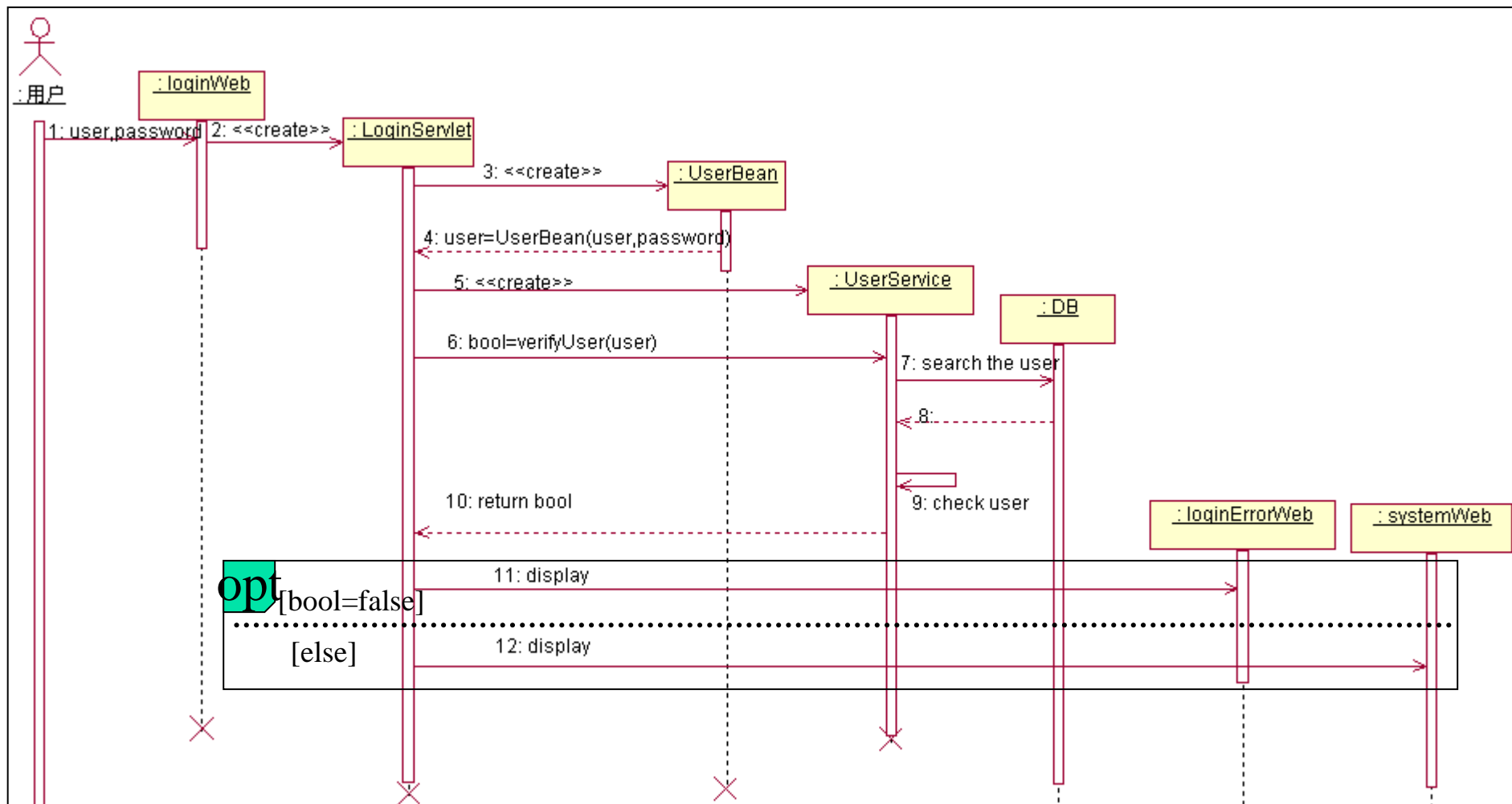
对登录系统用例建交互图



登录系统用例的场景2[失败]

4.7.6 case-图书管理系统交互图建模

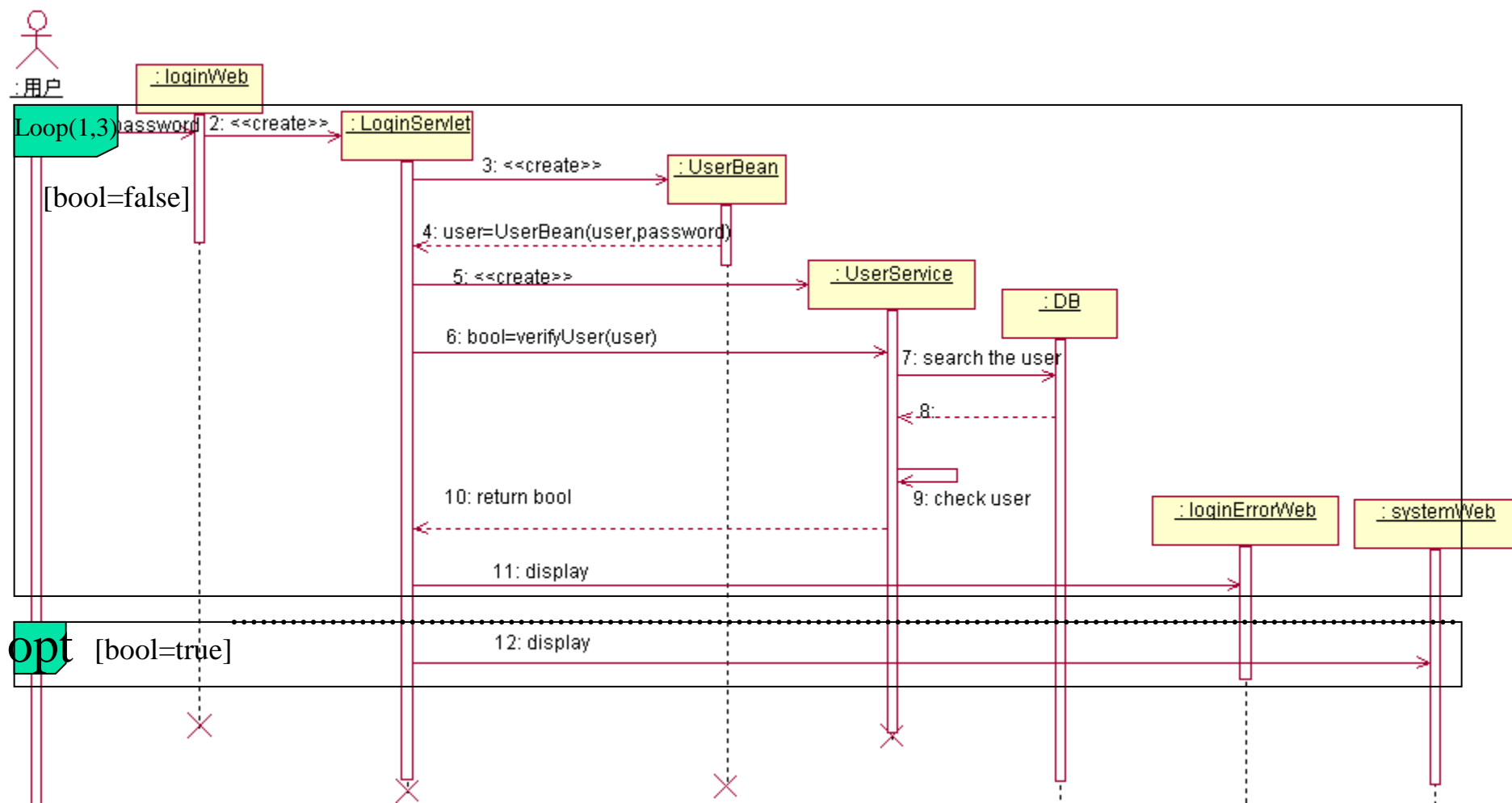
对登录系统用例建交互图



登录系统用例的场景3

4.7.6 case-图书管理系统交互图建模

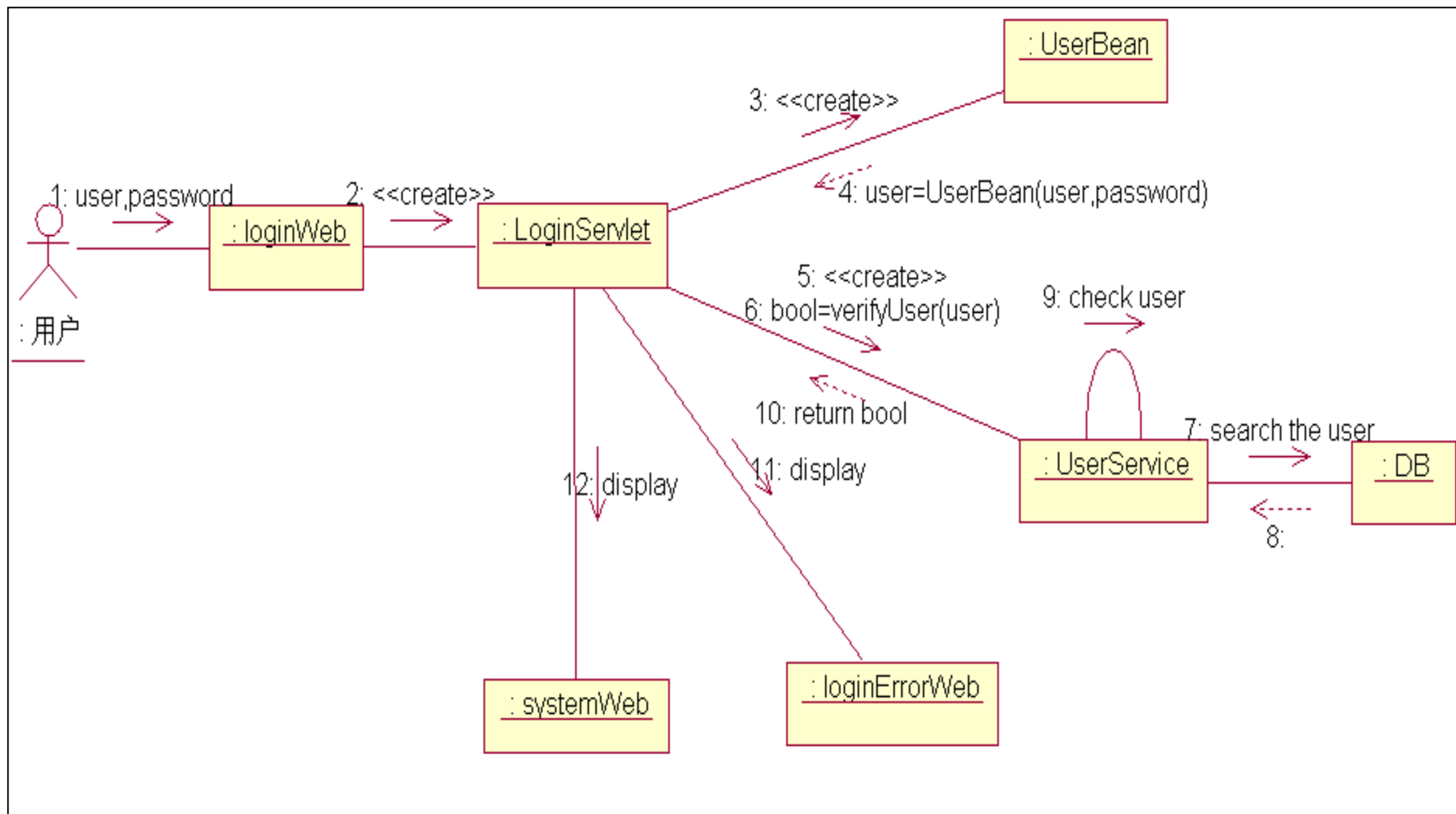
对登录系统用例建交互图



登录系统用例的场景4

4.7.6 case-图书管理系统交互图建模

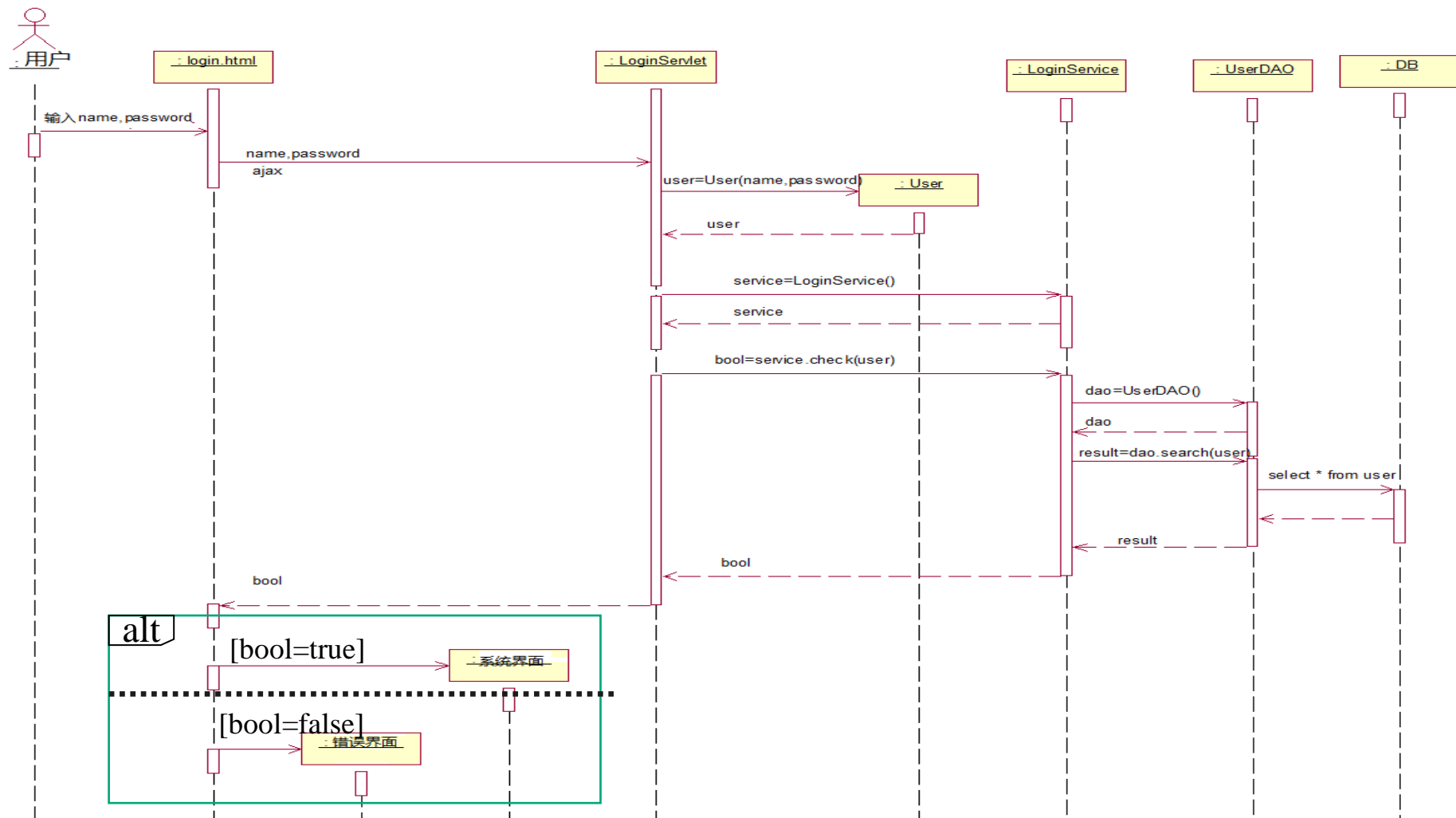
对登录系统用例建交互图



登录系统用例场景3的通信图

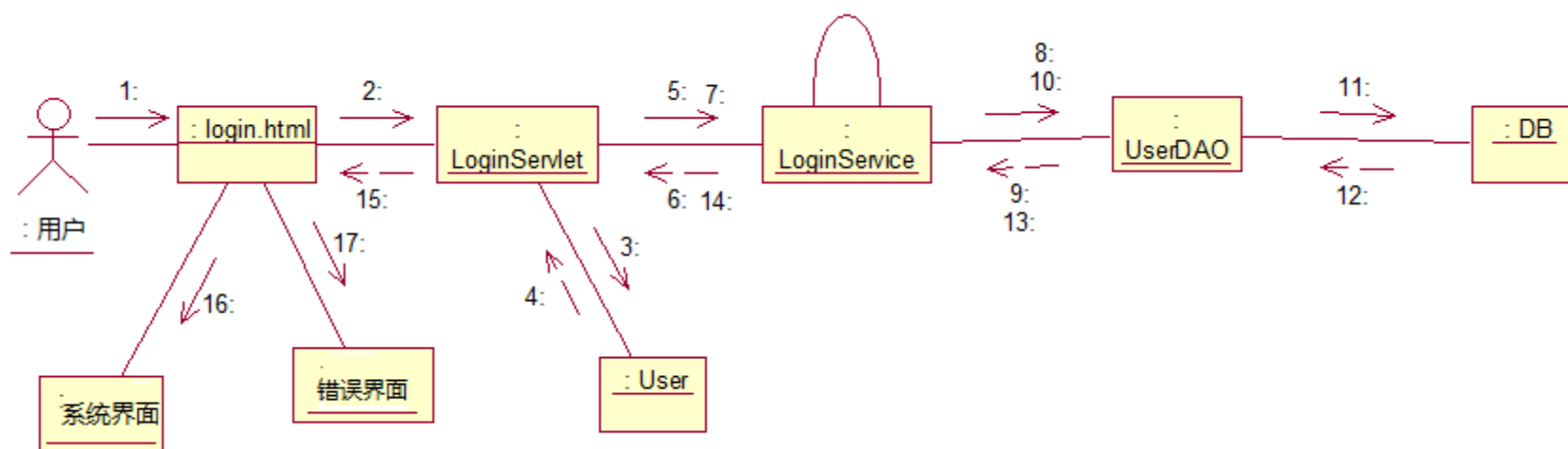
4.7.6 case-图书管理系统交互图建模

对登录系统用例建交互图



4.7.6 case-图书管理系统交互图建模

对登录系统用例建交互图





4.7.6 case-图书管理系统交互图建模

■ 对借书用例建交互图

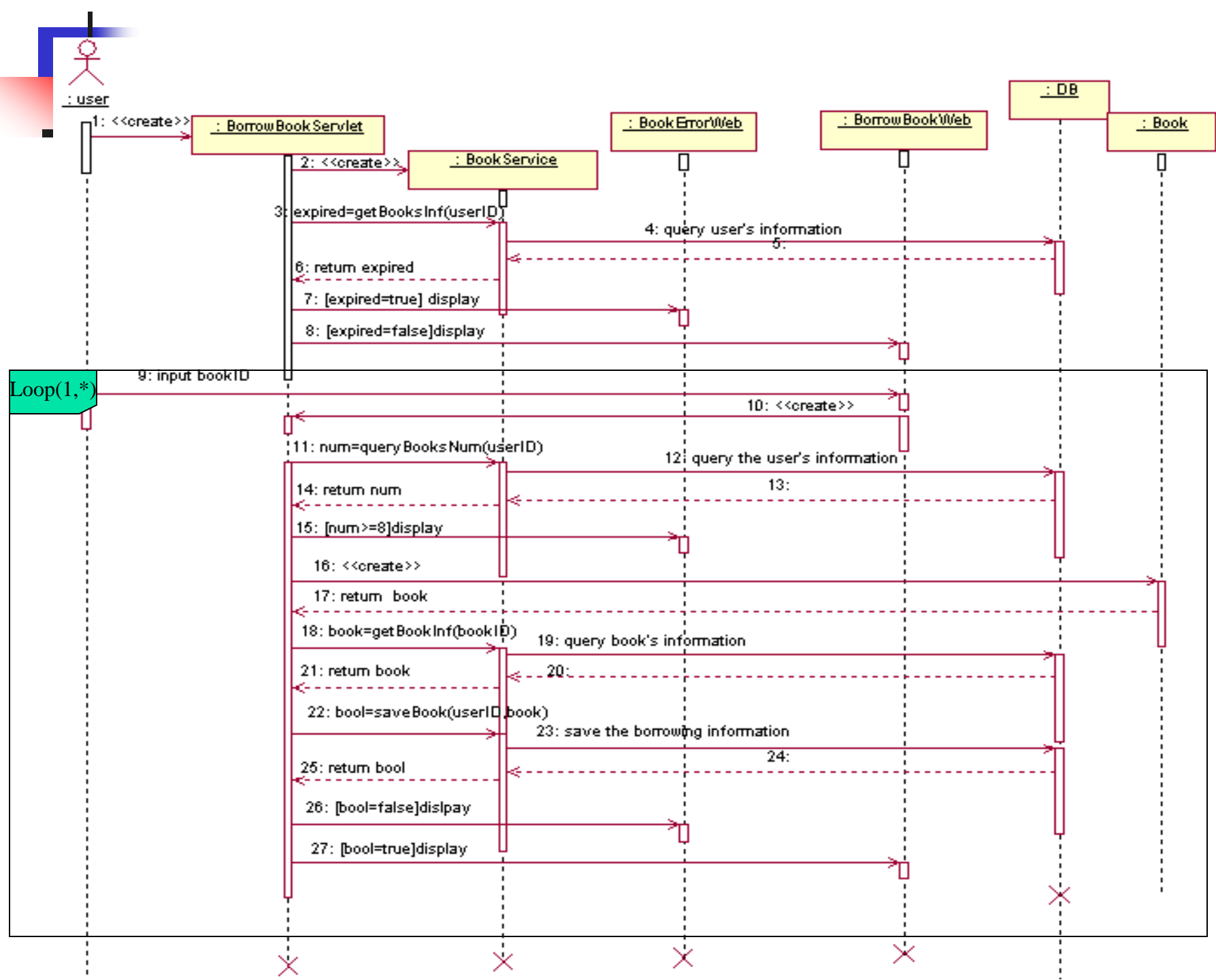
前置条件：一个合法的用户已经登录系统

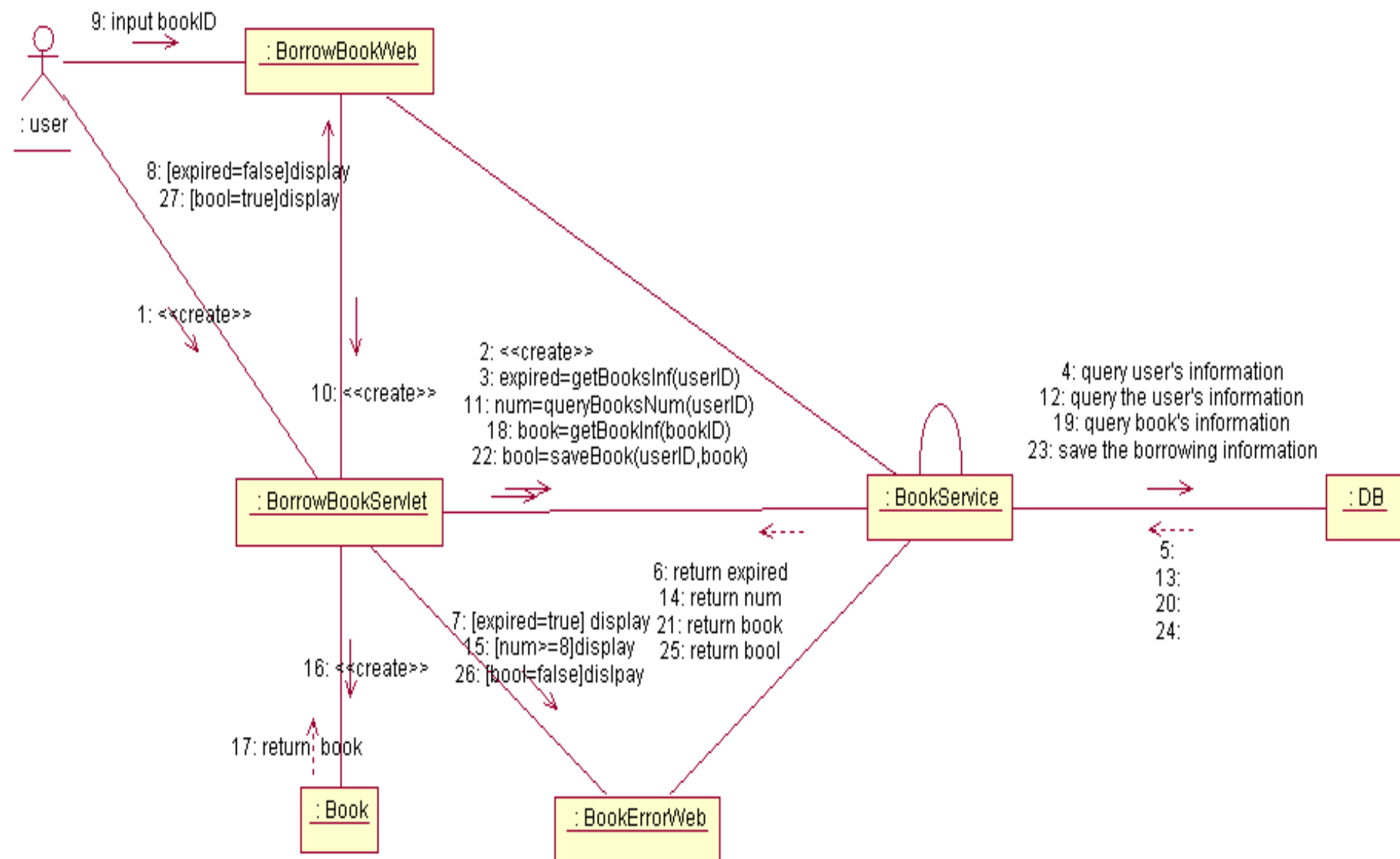
事件流：

1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 - a) 如有超期图书，用例结束。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) 系统查询用户的借书记录，如超出所借图书数目(8本)，用例结束。
 - b) 系统给出书籍的具体信息：书名、作者、出版社、日期。
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书
 - e) 用户选择退出，用例结束。
- 循环结束。

后置条件：系统保存所借图书或者超出所借图书数目，不能借书。

借书用例





4.7 交互图(Interaction Diagrams)

4.7.7 完善类图

■ 例：用户登录用例

❖ 需求分析(用户提供)

1. 用户登录用例界面：



2. 用户登录用例描述：

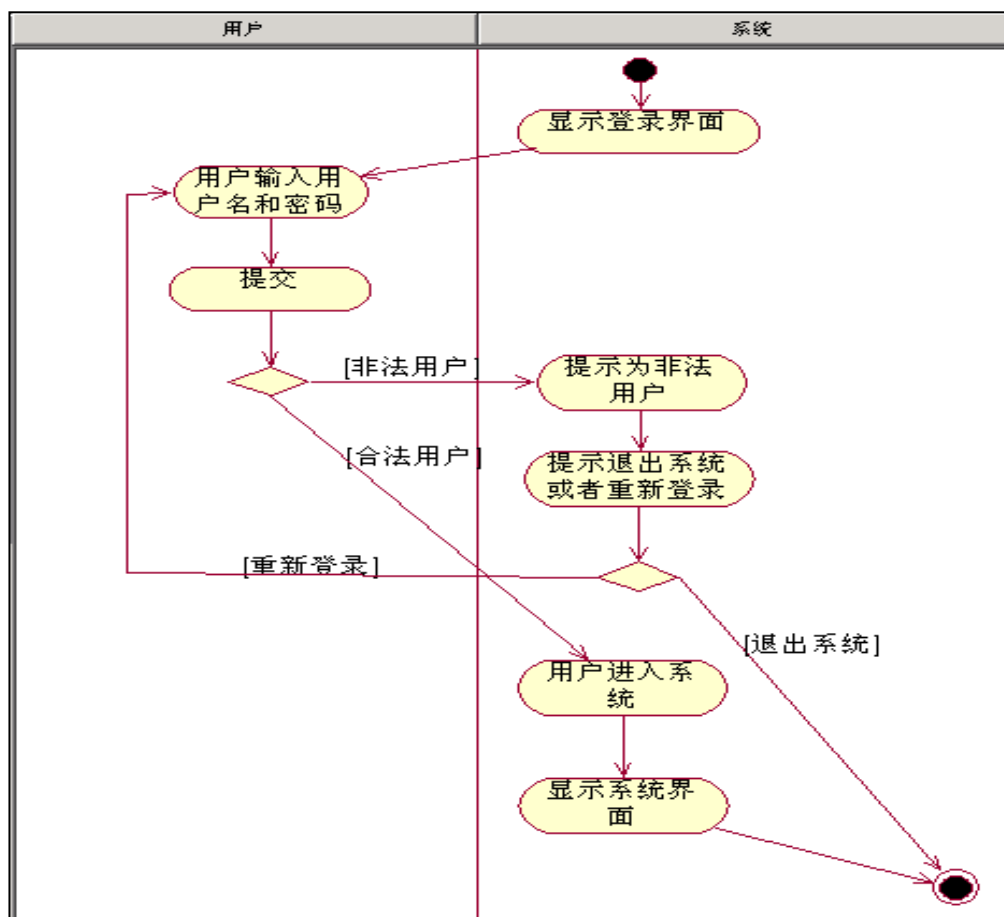
- (1) 用户在登录界面输入用户名、密码。
- (2) 如果登录失败，则显示：用户名或密码错误。
- (3) 如果登录成功，则进入系统界面。

4.7.7 完善类图

■ 例：用户登录用例

❖ 需求分析(用户提供)

3. 用户登录用例活动图：



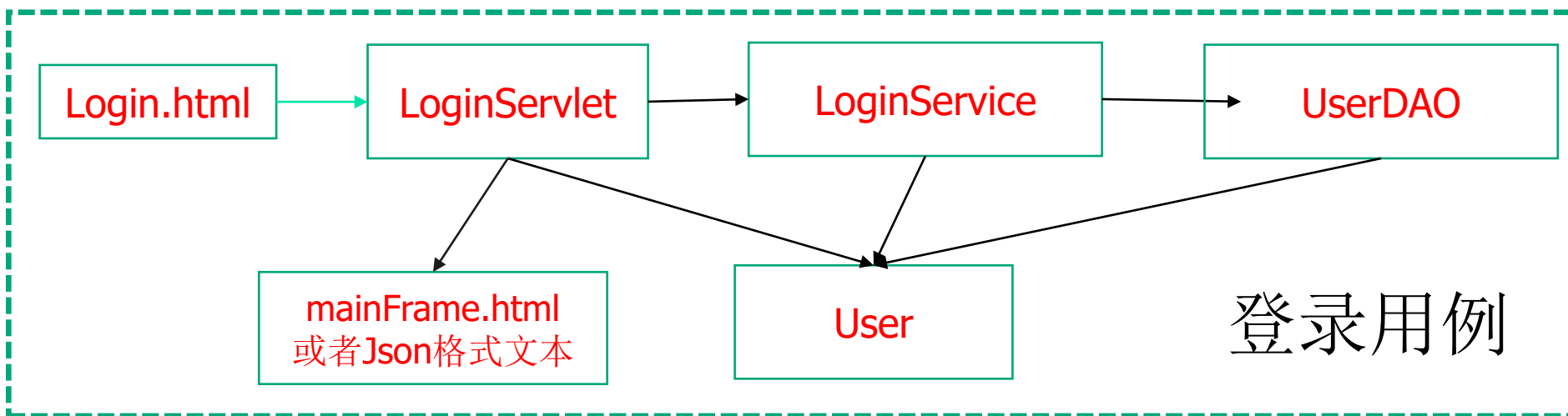
4.7.7 完善类图

■ 例：用户登录用例

❖ 软件设计

网页登录用例(login)领域建模

JavaEE Web软件体系结构风格





4.7.7 完善类图

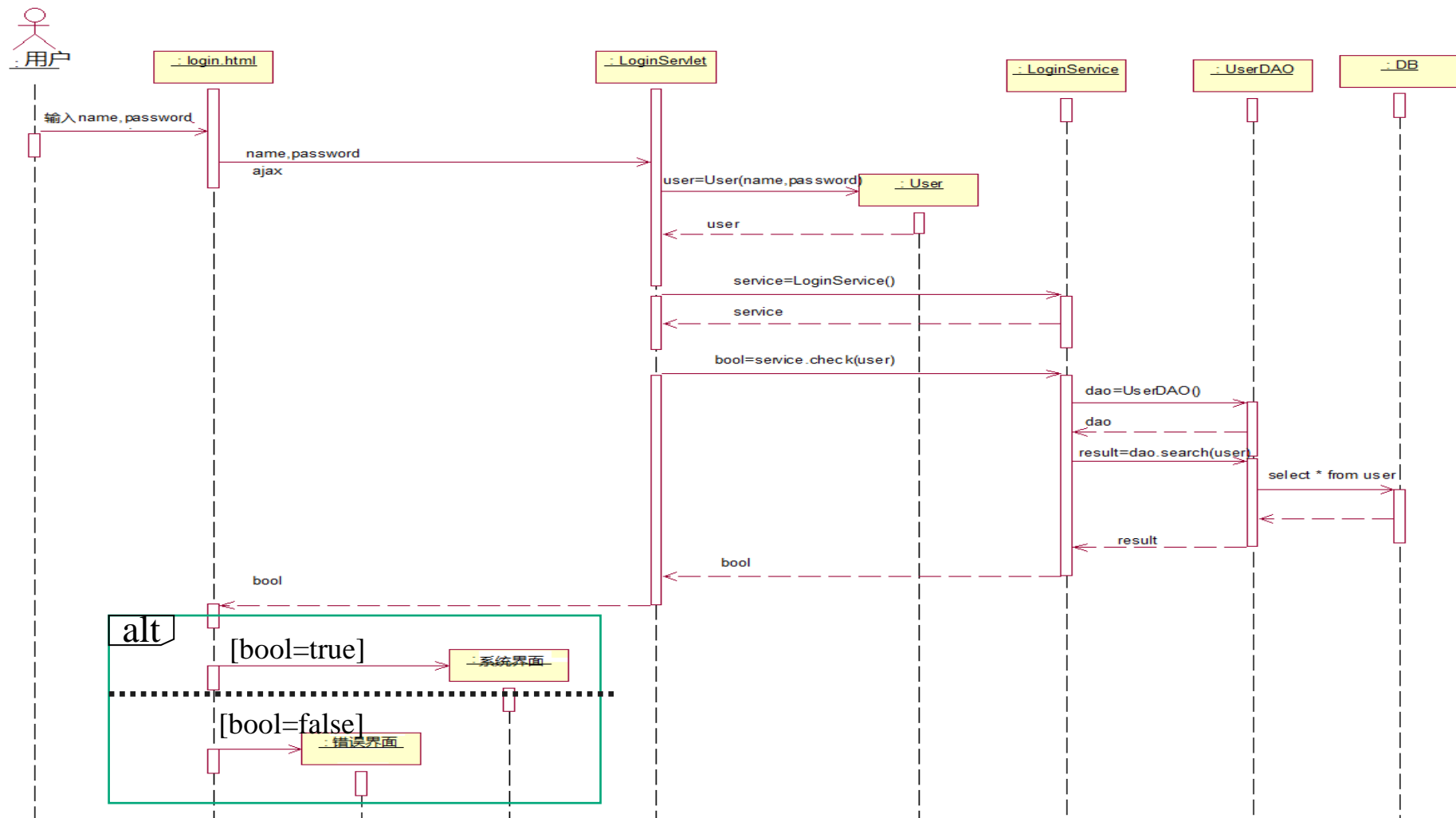
■ 例：用户登录用例

❖ 上述类图中的类并不完善：主要是类的方法没有完善。

● 交互图

4.7.7 完善类图

用户登录用例交互图

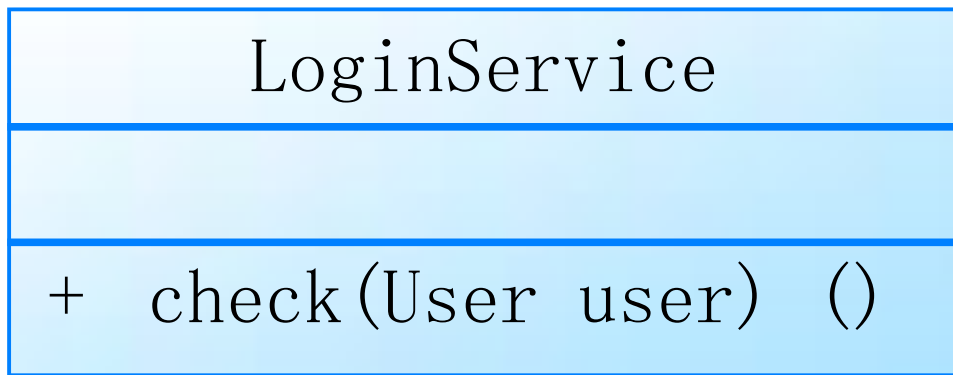
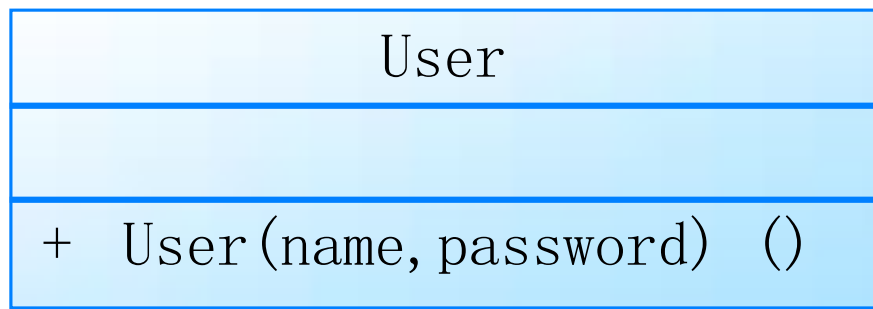


4.7.7 完善类图

■ 完善用户登录用例的类图

❖ 方法

- 顺序图中的消息
- Service类、DAO类添加常见的增、删、改、查询、分页查询方法。





UserService

```
+ UserService ()  
+ add (User user)      : void  
+ delete (User user)   : void  
+ update (User user)   : void  
+ select (User user)   : ArrayList<User>  
+ selectPage (User user) : ArrayList<User>
```

UserDAO

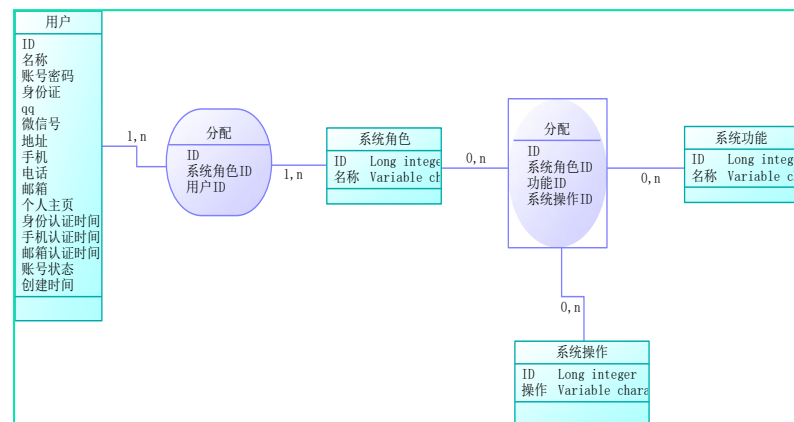
```
+ UserDAO ()  
+ add (User user)      : void  
+ delete (User user)   : void  
+ update (User user)   : void  
+ select (User user)   : ArrayList<User>  
+ selectPage (User user) : ArrayList<User>
```

4.7.7 完善类图

完善用户登录用例的类图

❖ 属性

- ER图的字段
- 界面
- 用例文本描述中的名词。
- 针对实体类（表对应的类）
 - ✓ ID: 每个实体加一个ID（自增: 主键）
 - ✓ int version属性: 防止并发时出错。
 - ✓ DateTime updateTime属性: 更新后可以马上看到效果
 - ✓ int updateUserID
- 针对每个属性添加:get/set方法。



控制面板

权限管理 3

用户维护

角色维护

许可维护

业务审核 3

业务管理 7

参数管理

首页 / 数据列表 / 新增

表单数据

登陆账号

请输入登陆账号

用户名称

请输入用户名称

邮箱地址

请输入邮箱地址

请输入合法的邮箱地址, 格式为: xxxx@xxx.com

+ 新增

重置

前置条件：一个合法的需求分析人员已经登录系统

事件流：

1. 当需求分析人员在用户维护主界面选择新增按钮时,用例开始。
2. 系统显示新增用户界面。
3. 需求分析人员输入该用户的用户名、邮箱
4. 点击保存按钮。

后置条件：系统保存新增的用户（密码默认888888）。



User

- ID : int
- name : String
- password : String
- address : String
- email : String
- version : int
- updateTime : DateTime
- UpdateUserID : int

- + User (String name, String password)
- + getName () : String
- + setName (String name) : void



thanks