



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

4.4 活动图(activity diagrams)

4.4.1 建模 workflow(Workflow)或操作(Operation)

4.4.2 定义

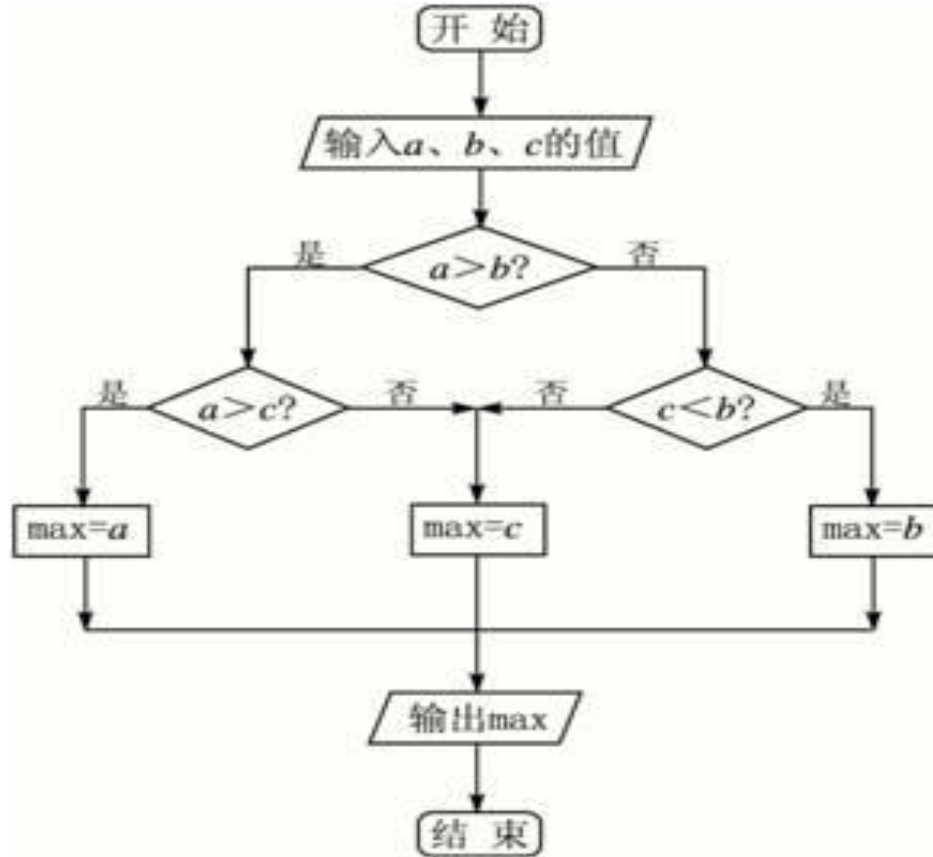
4.4.3 应用场合

4.4.4 建模技术

4.4.5 Case:图书管理系统活动图建模

4.4 活动图(activity diagrams)

例子：输入三个数 a, b, c , 求三个数的最大值。



算法：流程图（有助于理解算法）



4.4 活动图(activity diagrams)

登录系统用例

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。

有助于理解用例的图? 活动图是其中的一种。



4.4 活动图(activity diagrams)

4.4.1 建模 workflow(Workflow)或操作(Operation)

工作流、操作:系统的**动态**方面

■ 工作流: 商业处理过程（业务逻辑）

登录系统用例

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。

4.4.1 建模 workflow(Workflow)或操作(Operation)

■ 操作

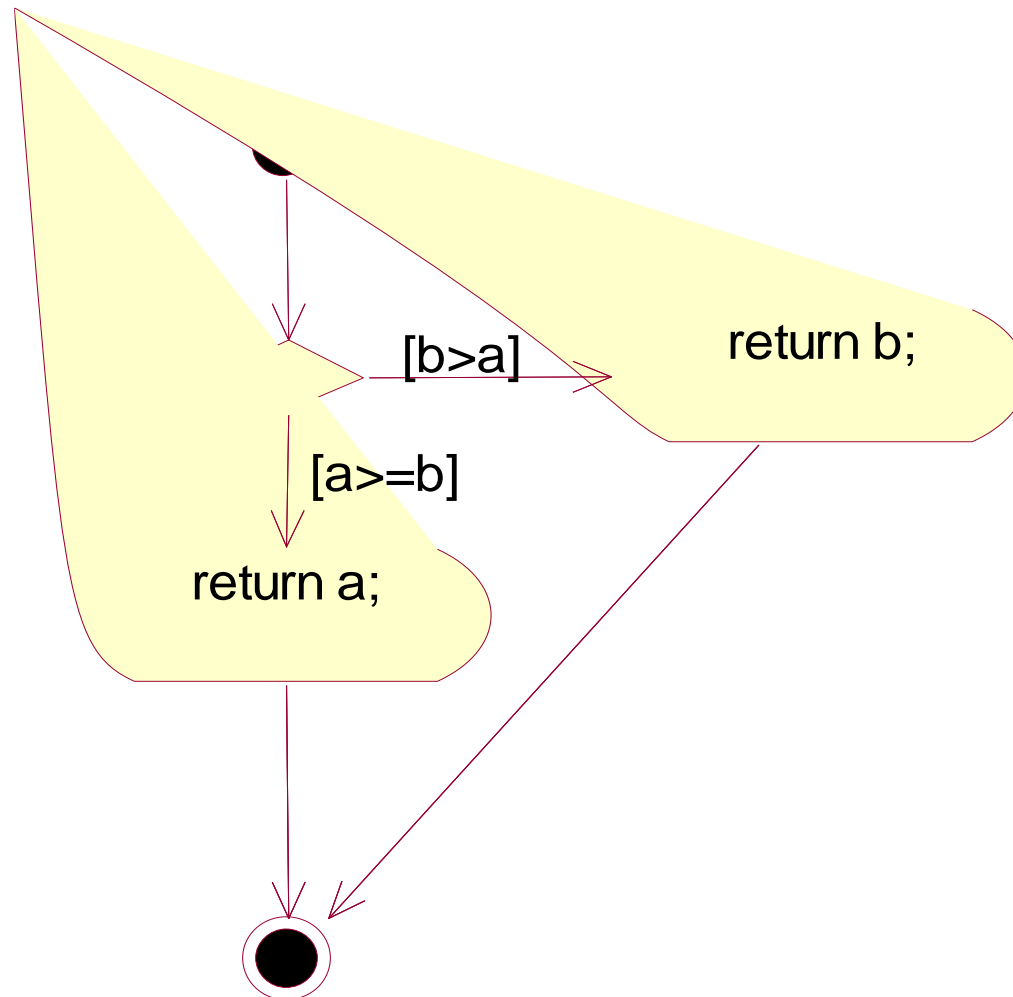


Figure . max(a,b)

4.4 活动图(activity diagrams)

例子

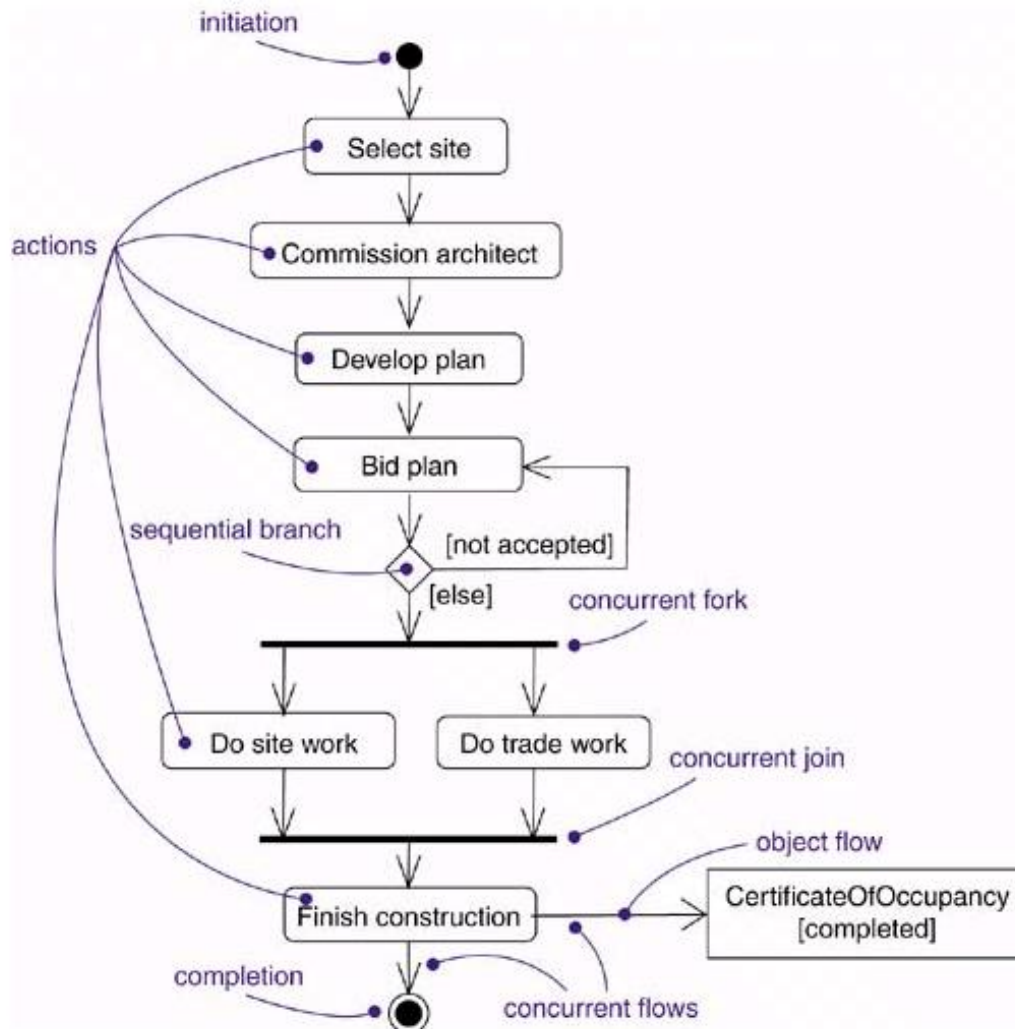


Figure .活动图



说明

- ❖ 活动图实际上是一个**流程图**，它图示了从一个活动到一个活动的流程。
- ❖ 但与传统的流程图不同之处在于，活动图也可表示**并发(concurrency)**
- ❖ 活动图：重点在对象执行的**活动(activities)**上。
它强调的是**活动与活动**之间的**流程**上。

4.4 活动图(activity diagrams)

4.4.2 定义

■ Action

These executable, atomic computations are called actions.

原子性

- ❖ 调用方法
- ❖ 发一个信号
- ❖ 创建或者删除一个对象
- ❖ 计算表达式.

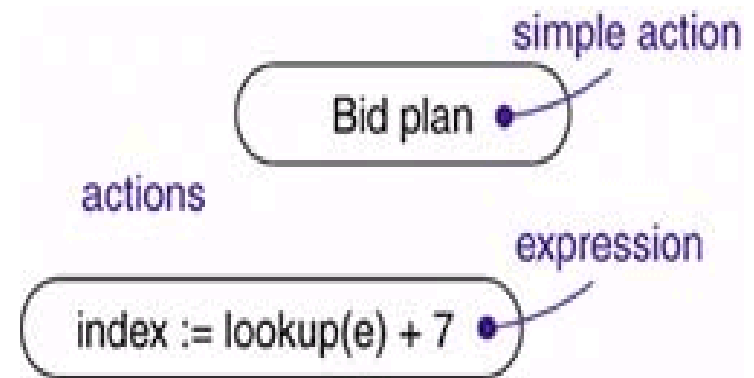
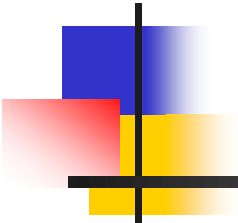


Figure . Actions



系统查询用户借还书记录

1 连接数据库

2 从数据库中查询
用户借还书记录

3 关闭数据库连接

Action

4.4.2定义

Activity

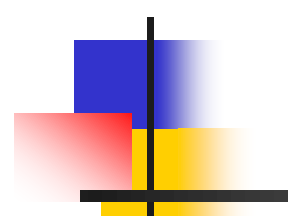
An activity is an ongoing nonatomic execution.

❖ an activity 是由多个actions执行来完成的.

- 一次 action 是一次 activity 的特例。
- 一个activity 结点是由其它的 activity 结点和 actions结点构成的。



Figure . Activity Nodes



系统查询用户借还书记录

Activity

1 连接数据库

2 从数据库中查询
用户借还书记录

3 关闭数据库连接

4.4.2定义

■ 控制流

从一个one action 或者activity 结点到下一个 action 或者activity 结点

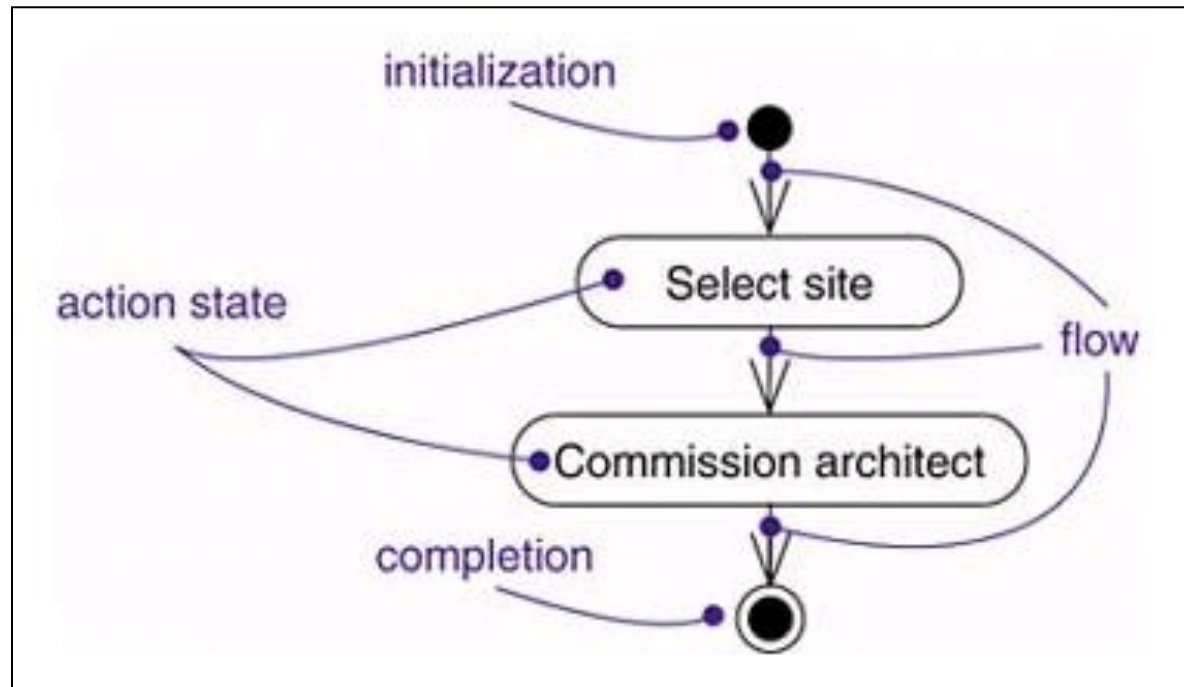
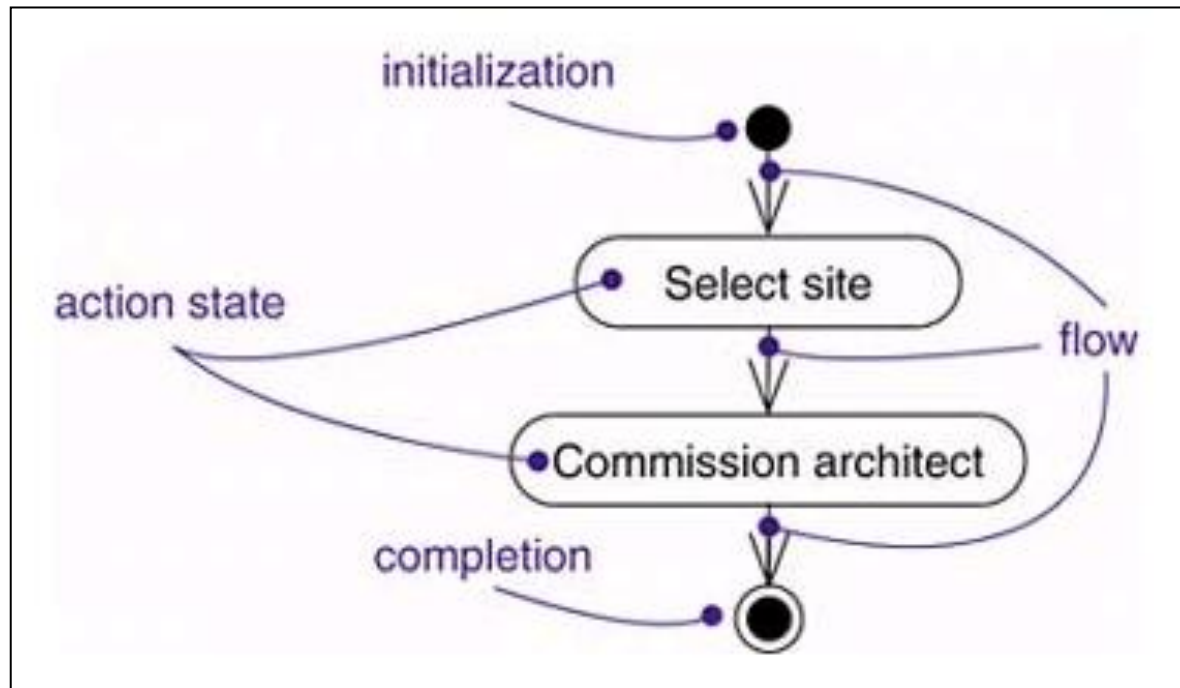


Figure . Control Flows

4.4.2定义

■ 活动图（activity diagram）

控制流和数据流从一个活动(activity)到另一个活动(activity)



activity diagram

4.4.2定义

分支(Branching) if

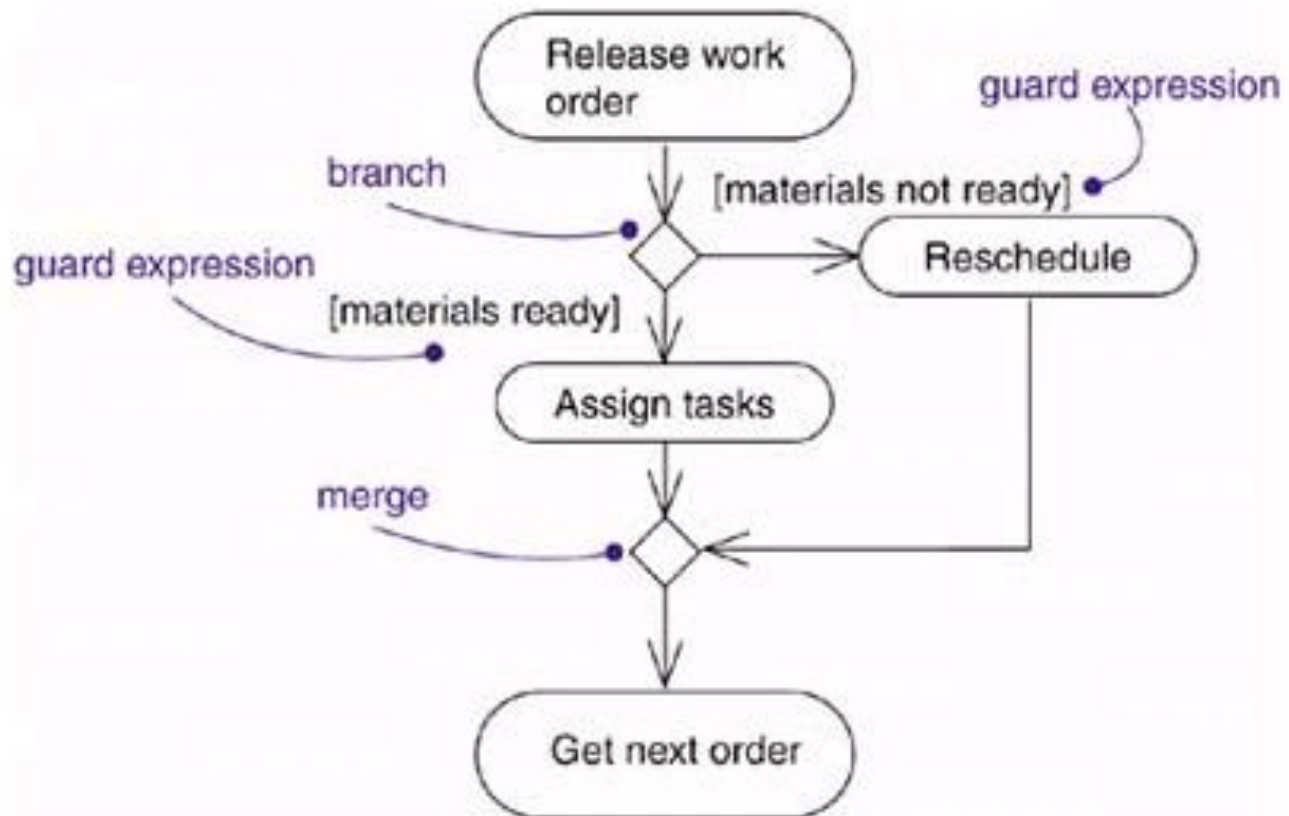


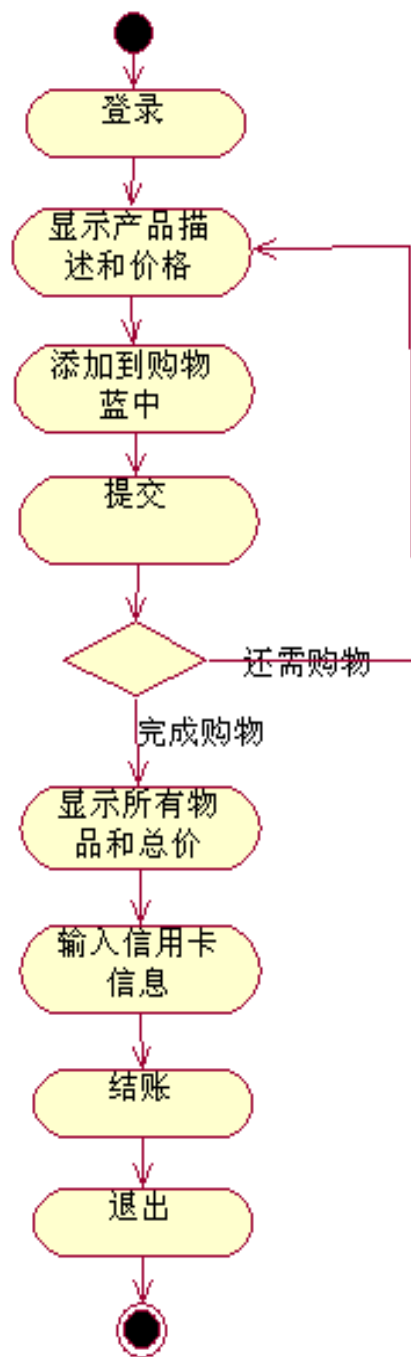
Figure . Branching

4.4.2定义

循环(iteration)

- ❖ 首先设置迭代器(iterator)的初始值。
- ❖ 增加迭代器(iterator)的值。
- ❖ 分支语句判断迭代过程是否结束。

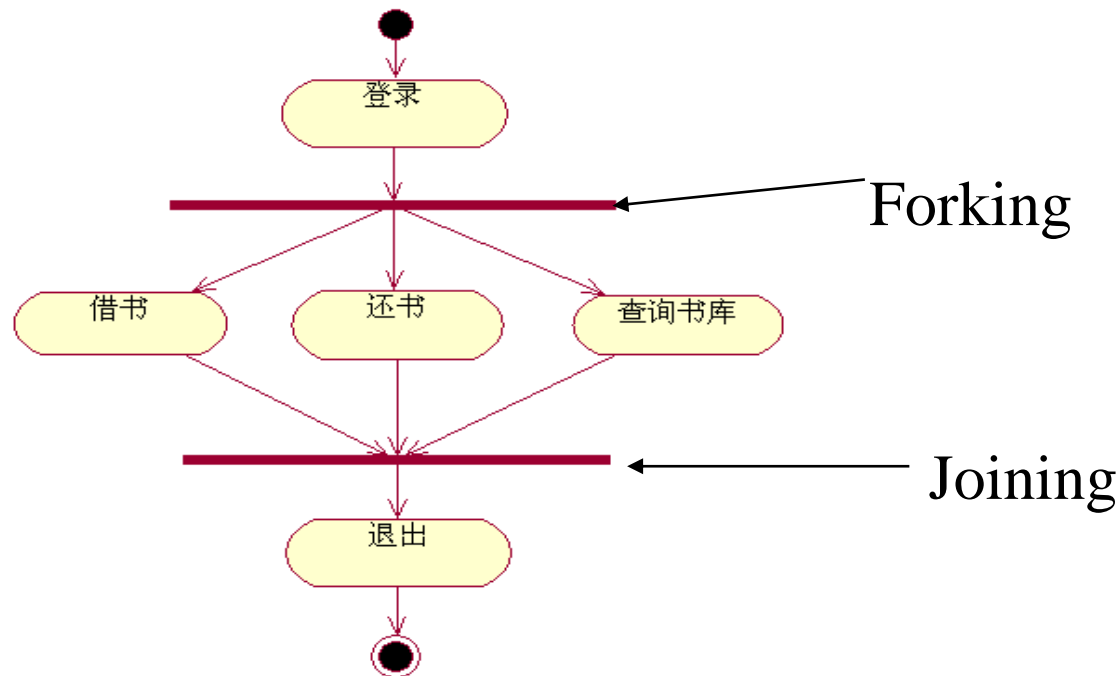
网上购物活动图



4.4.2定义

分岔(Forking) 与联结(Joining)

- ❖ 建模**并发**concurrent的工作流。
- ❖ 在UML中，使用同步条(synchronization bar)来说明并发流中的**forking** 和 **joining** 。



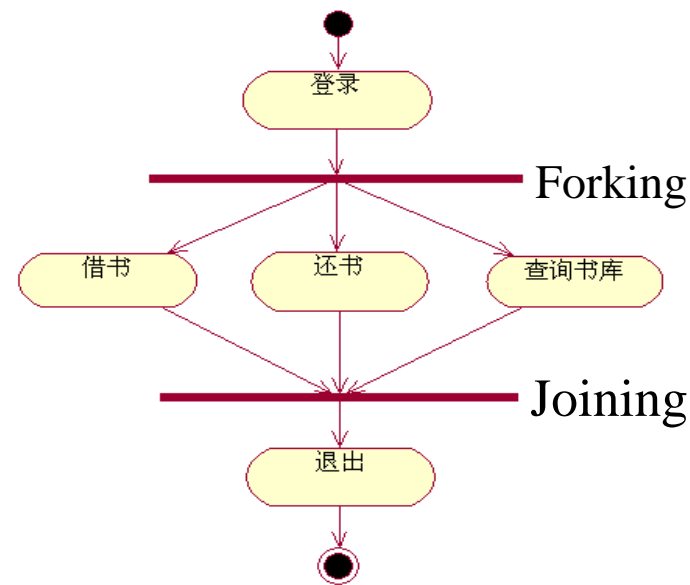
Forking and Joining

4.4.2定义

■ 分岔(Forking) 与联结(Joining)

❖ Forking

- 一个fork 代表将一条控制语句分成若干条并发的语句。
- 一般来说，这些语句是parallel执行的。
- 但在实际运行的系统中：
 - concurrent: 当系统布署在多个结点上。
 - sequential yet interleaved :当系统布署在单个结点上



4.4.2定义

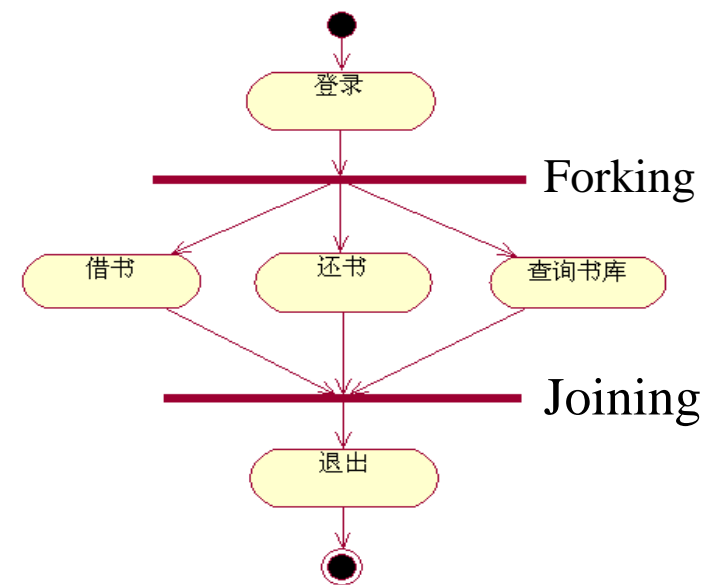
■ 分岔(Forking) 与联结(Joining)

❖Joining

join 表示多条并发的控制流同步(synchronization)。

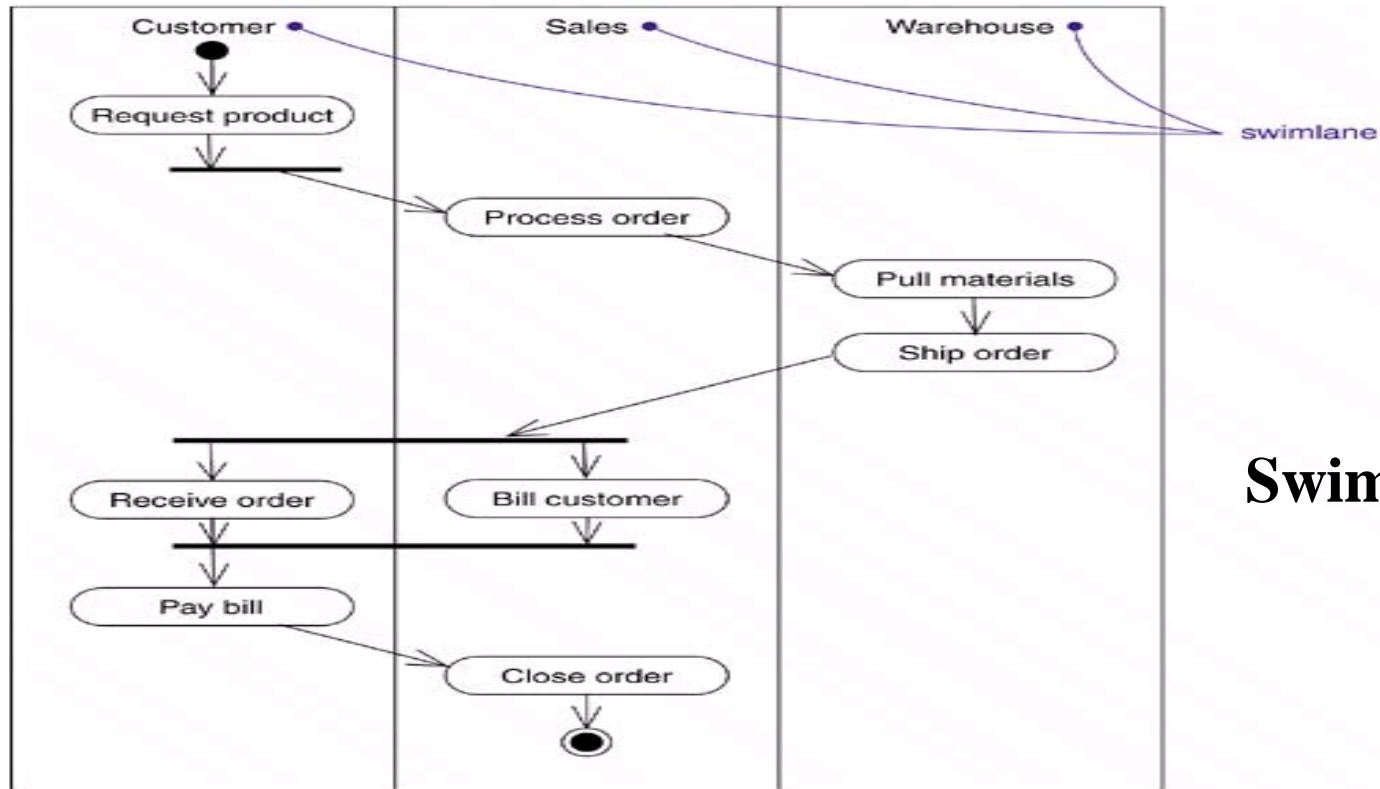
说明

- Join处有多条入的activities和一条出的activity。
- 在join上面，多条入的activities 是**并发(parallel)**的。
- 在Join处，这些并发的流要同步，这意味着先到的流要等待其它还没有到的流。



4.4.2定义

泳道(Swimlanes)

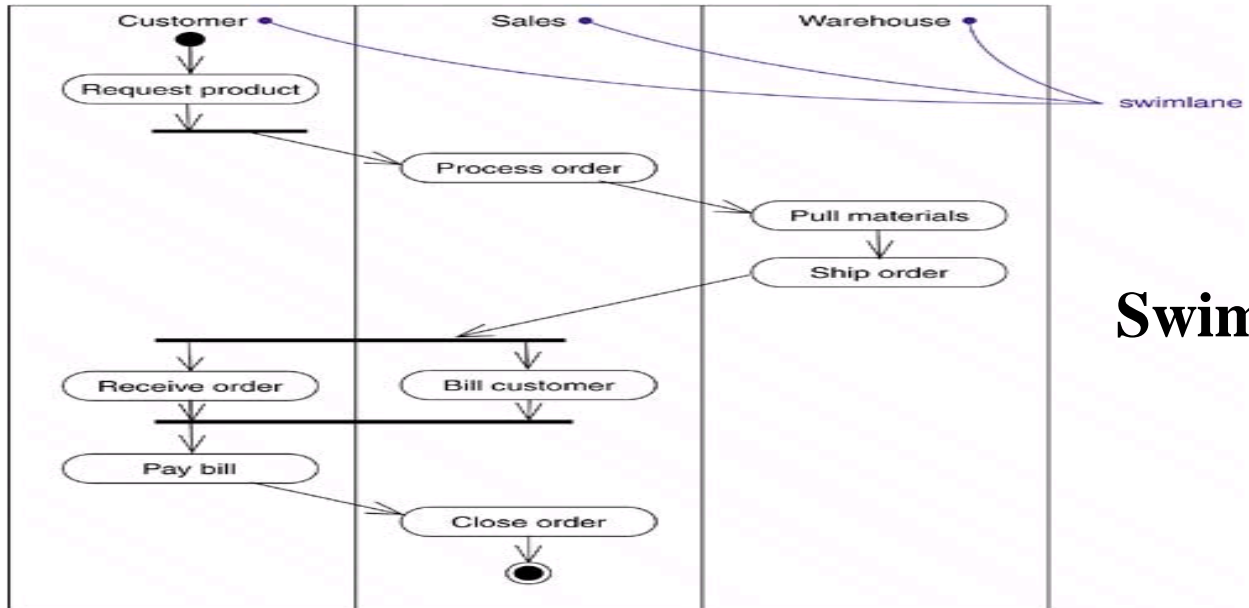


Swimlanes

- ❖ 把活动(activity)分组，**每组代表一个商业组织(business organization)**，商业组织负责这些活动的执行。
- ❖ 在UML中，这些组称为**泳道(Swimlanes)**。

4.4.2定义

泳道(Swimlanes)



Swimlanes

说明:

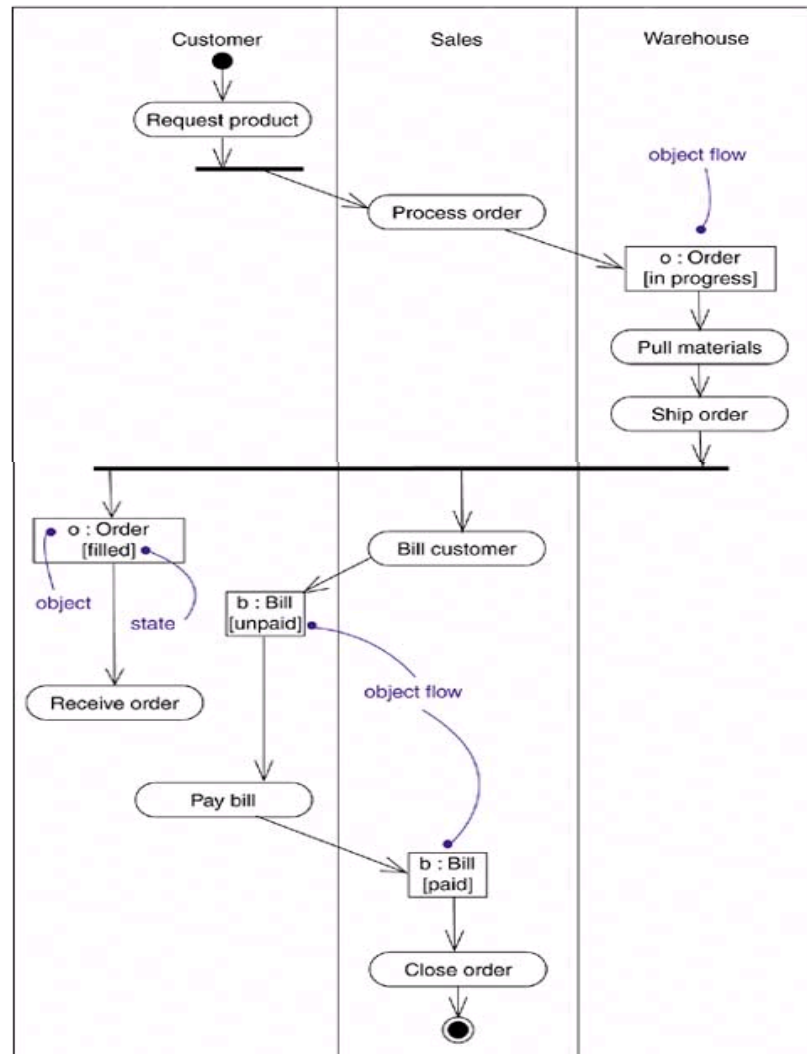
- ❖ 泳道没有很深的语义，仅仅代表现实世界中的组织机构。
- ❖ 每一个泳道可能由一个或者多个类实现。
- ❖ 每一个活动仅仅属于一个泳道，但活动与活动间的转移是穿过泳道的。

4.4.2定义

■ 对象流(Object Flow)

❖ 活动图中可能会涉及到对象 (Objects)

❖ 对象是由某些活动产生，并由其它的活动使用和修改。



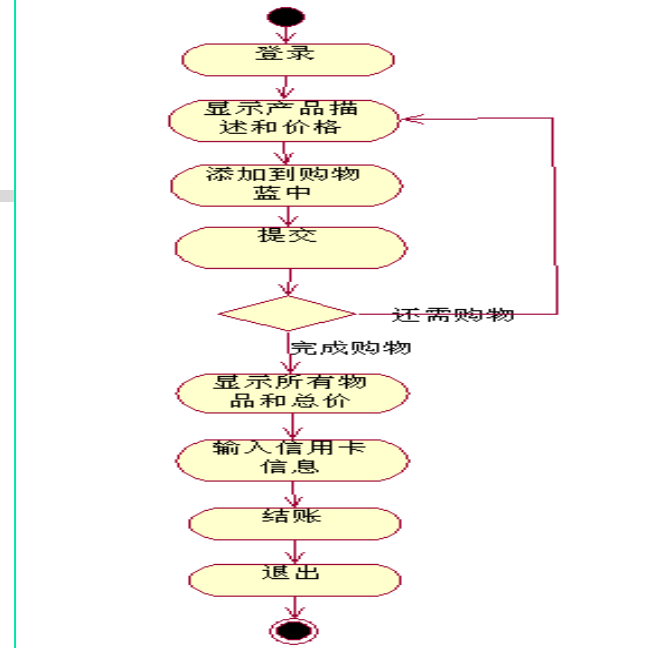
❖ 之所以称为对象流，原因在于它代表了**对象的值**从一个活动到另一个活动会有变化。

4.4.2定义

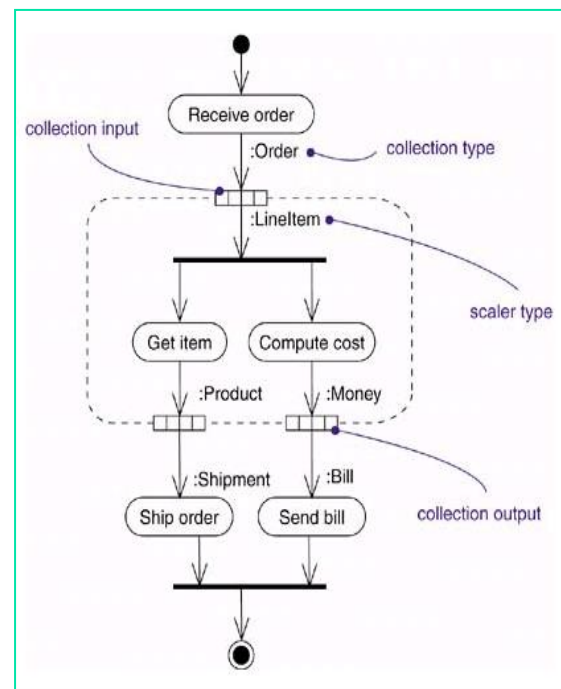
■ 扩展区域(Expansion Regions)

❖ 循环机制模糊了操作的实际意义。

❖ 一个扩展区域代表了对表(list)或者集合(set)操作的活动块。



网上购物活动图

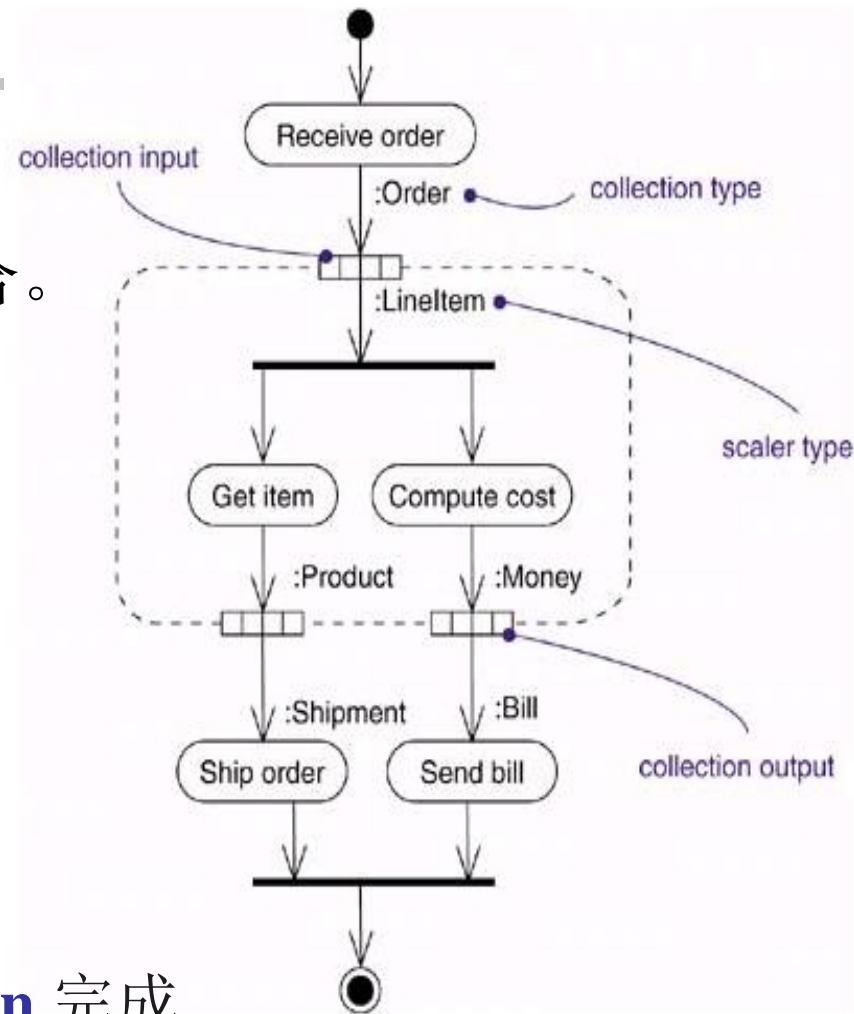


4.4.2定义

■ 扩展区域

- ❖ 扩展区域的输入和输出都是值集合。
- ❖ 对于数组中的**每一个元素**在此区域中都会**执行到一次**。
- ❖ 当**expansion region** 中的每一次运算都完成后，它的输出值都放入到**输出数组**中，次序与**expansion region** 的**输入数组**一致。
- ❖ 换句话说，一个**expansion region** 完成的是“**for all**”的操作，操作对象是输入数组中的每一个元素，结果都会放到一个新的输出数组中。

Figure . Expansion region





4.4 活动图(activity diagrams)

4.4.3 应用场合

当建模一个动态系统时，一般在如下两方面要用到**活动图**。

- 建模工作流。
 - ❖ 重点在活动上。
 - ❖ 在这种应用中，对象流显得尤其重要。

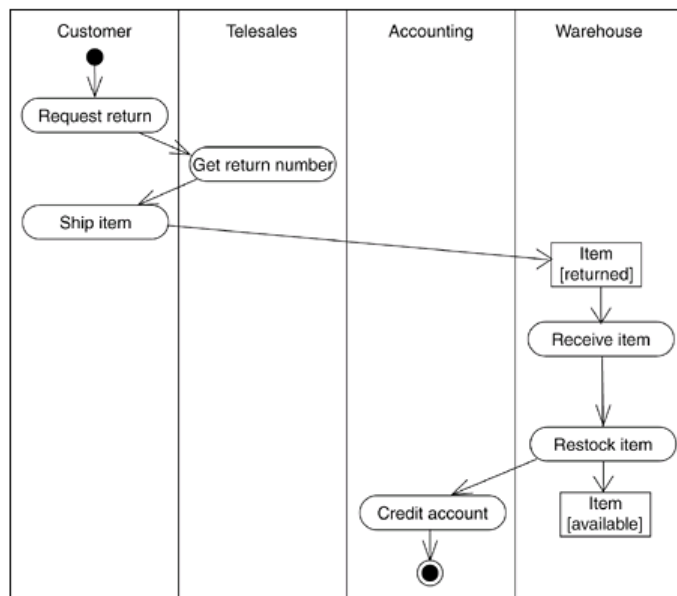
- 建模一个操作(operation)
 - ❖ 活动图：流程图

4.4 活动图

4.4.4 建模技术

■ workflow 建模

- ❖ 选择好 workflow 中涉及到的**商业组织**。
- ❖ 为每一个商业组织创建一个**泳道**。
- ❖ 标记出 workflow 的**前置条件**和**后置条件**。
- ❖ 从 workflow 的初始状态开始，对 workflow 中**每一次活动**按顺序建模。
- ❖ 如果在 workflow 中涉及到重要的对象，最好把这些**对象**及**值**在活动图中描述出来。





■ 说明

- ❖ 一个用例的**Activity** 图解释活动者如何与系统交互来达到该用例的目标。
- ❖ 工作流级的**Activity** 图表示用例的执行顺序和执行条件。



4.4 活动图

4.4.4 建模技术

■ 对操作建模

- ❖ 标记出操作中初始状态的前置条件和结束状态的后置条件。
- ❖ 从初始状态开始，描述出操作中的活动。
- ❖ 在建模过程中，如有必要，使用分支和迭代。
- ❖ 当且仅当该操作属于某一个类时，才使用**forking**和**joining**来表示并发流。

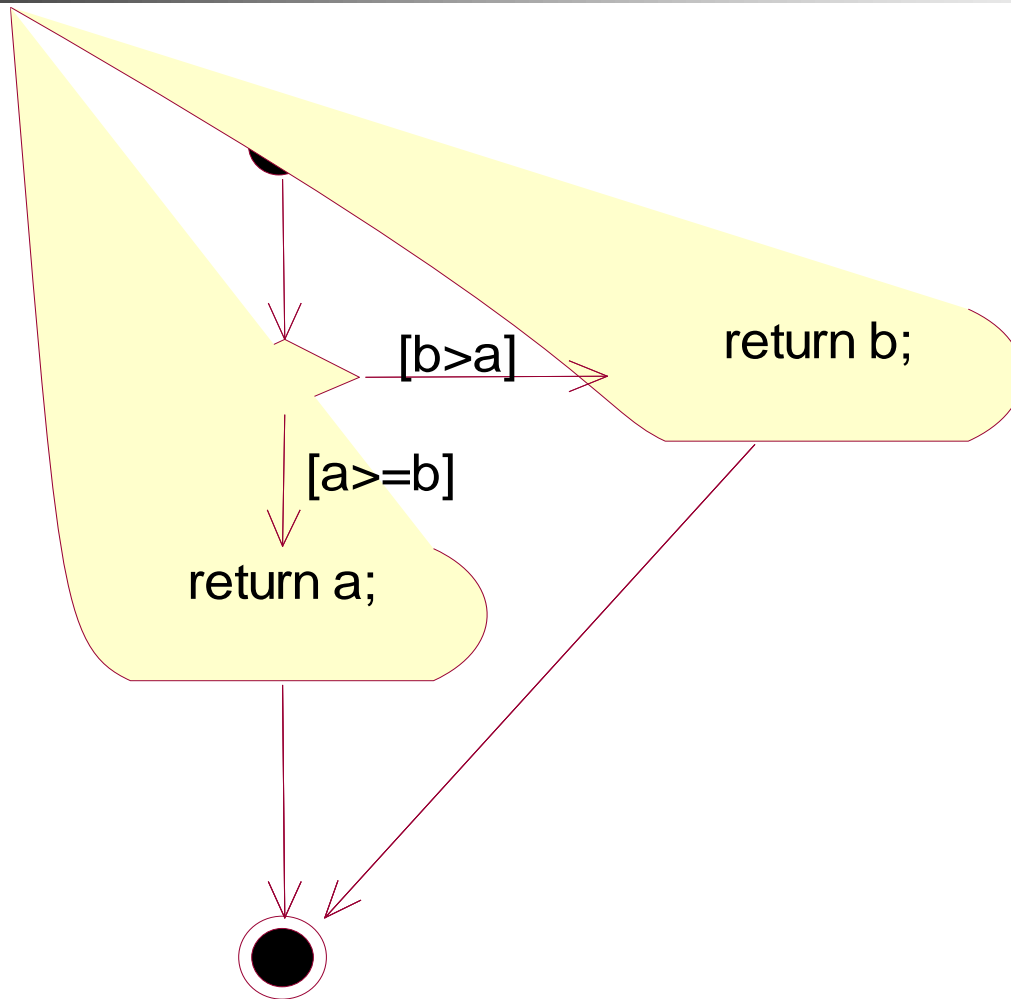


Figure . max(a,b)

$\text{fib}(n)$
 $\left\{ \begin{array}{ll} 1 & n=1 \\ 2 & n=2 \\ \text{fib}(n-1)+\text{fib}(n-2) & n>2 \end{array} \right.$

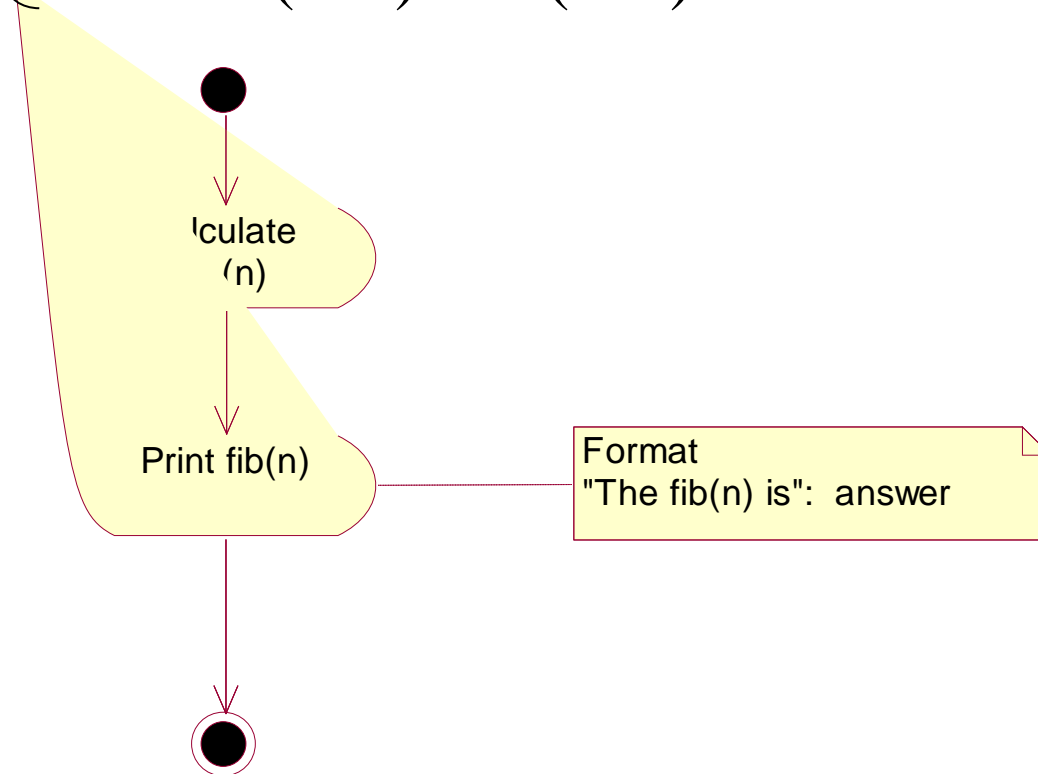


Figure . fib(n)

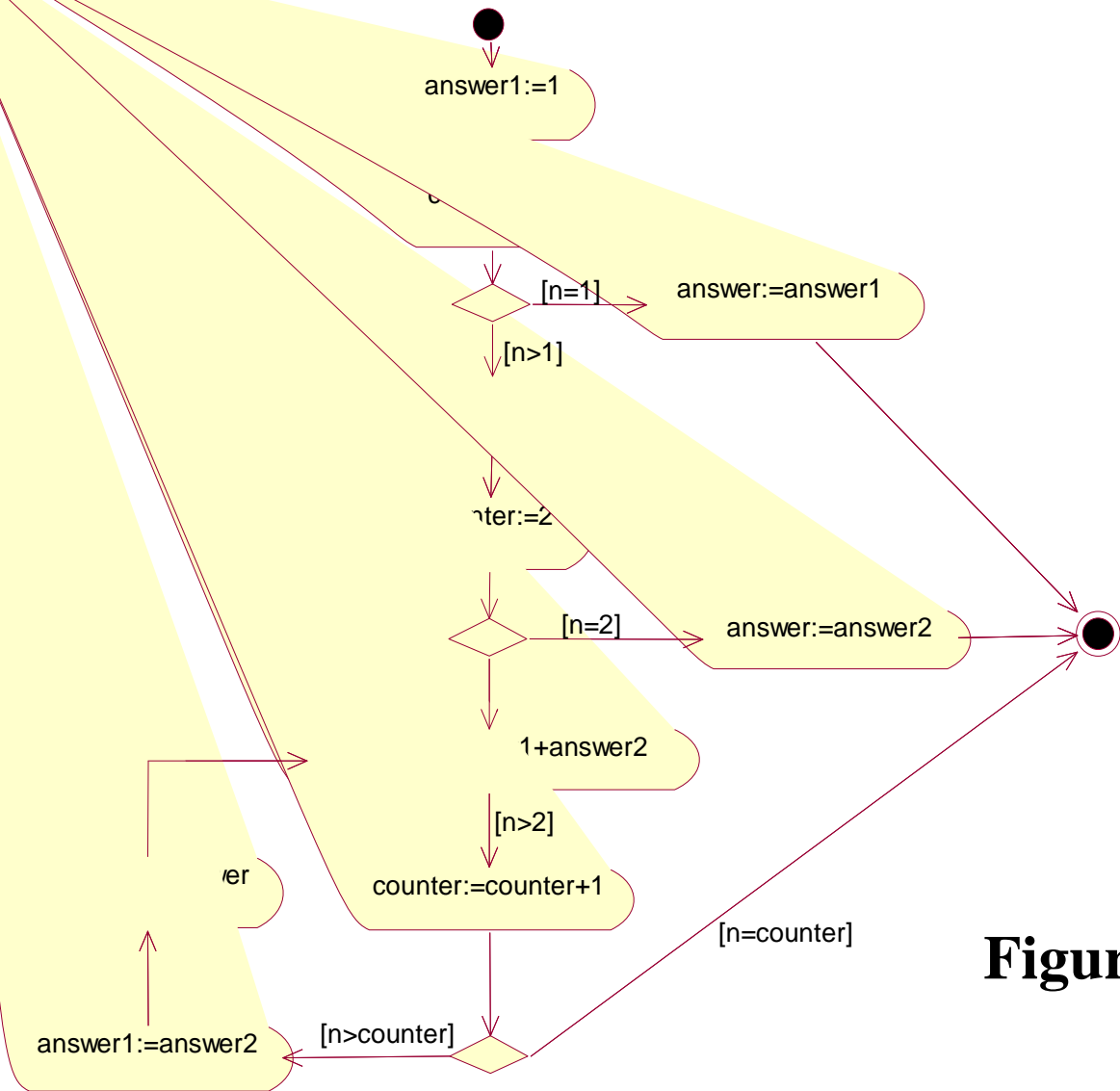
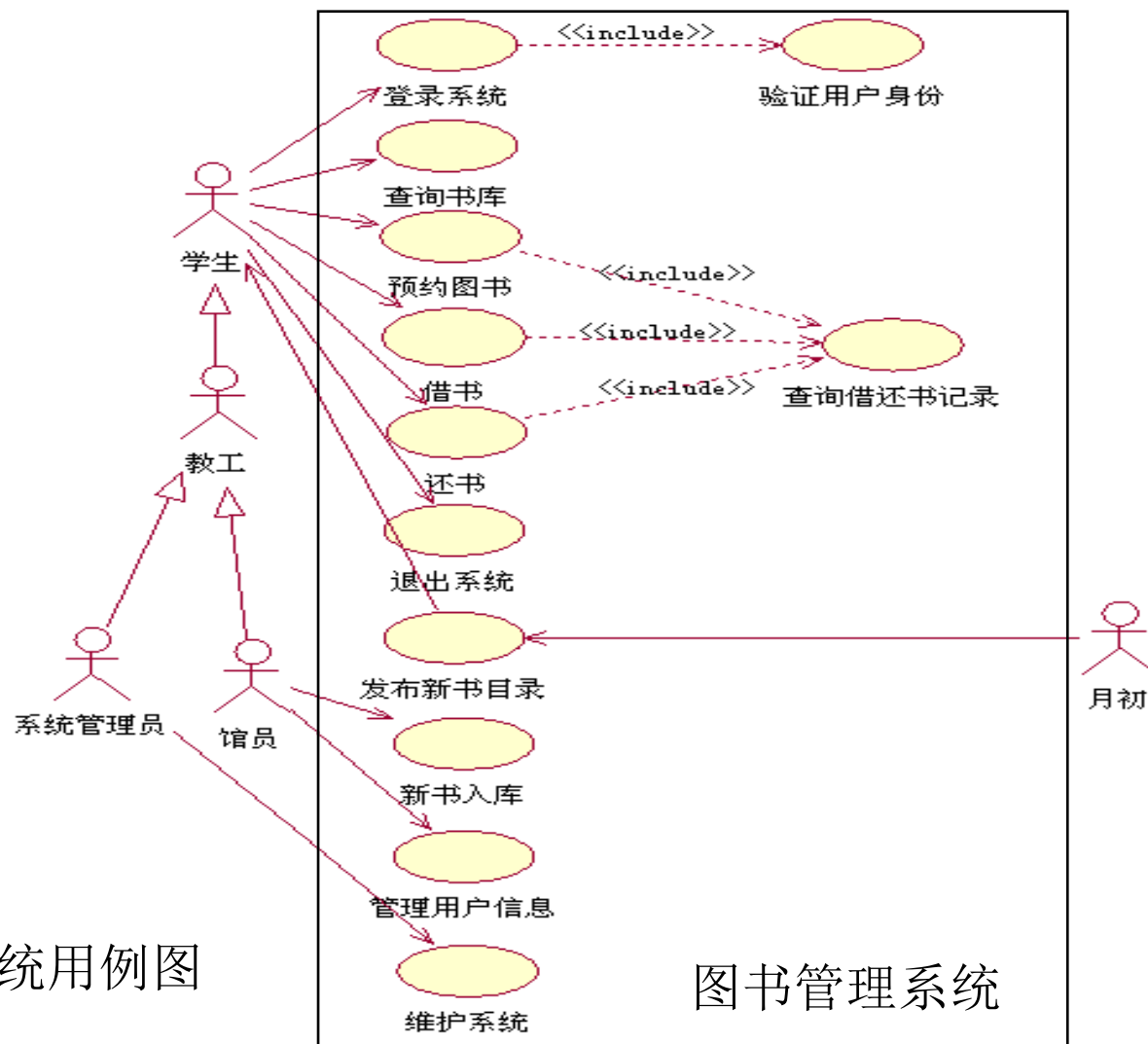


Figure . fib(n)

4.4 活动图

4.4.5 Case:图书管理系统活动图建模



图书管理系统用例图

图书管理系统

4.4.5 Case:图书管理系统活动图建模

对登录系统用例建活动图

❖思路：先对基本路径建模，再对可选路径建模。

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

4.4.5 Case:图书管理系统活动图建模

对登录系统用例建活动图

❖ 思路：先对基本路径建模，再对可选路径建模。

- 对登录系统的基本路径建活动图

前置条件：

事件流：

基本路径

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. 系统显示主界面，用例结束。

后置条件：用户进入系统，系统显示主界面

登录系统用例的基本路径

4.4.5 Case:图书管理系统活动图建模

对登录系统用例建活动图

❖ 思路：先对基本路径建模，再对可选路径建模。

- 对登录系统的基本路径建活动图

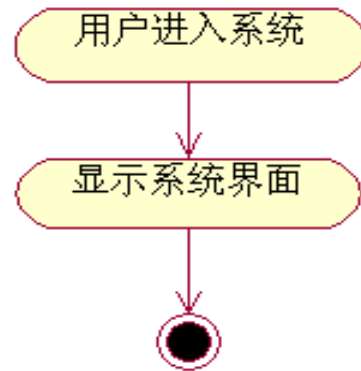
方法：遵循用例的陈述。

步骤：

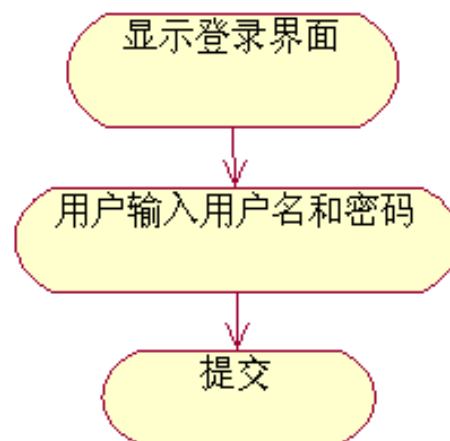
- 1 先识别前置条件。
- 2 接下来标识后置条件。
- 3 再接下来对其余陈述部分建模。
- 4 最后组织前置条件、后置条件、其余部分。

1 先识别前置条件:无

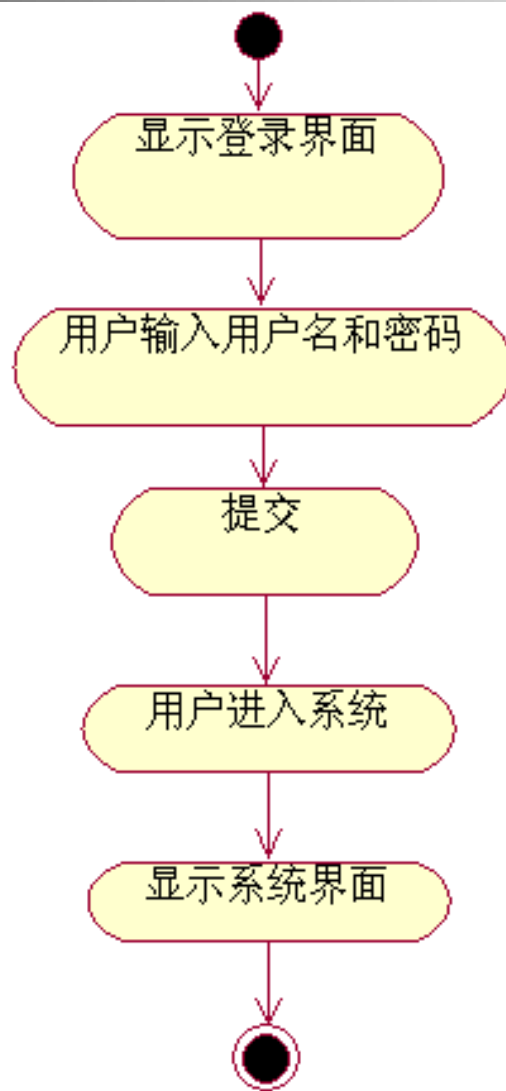
2 接下来标识后置条件:



3 再接下来对其余陈述部分建模。



4 最后组织前置条件、后置条件、其余部分。



登录系统的活动图

4.4.5 Case:图书管理系统活动图建模

对登录系统用例建活动图

❖思路：先对基本路径建模，再对可选路径建模。

● 登录系统的可选路径建模

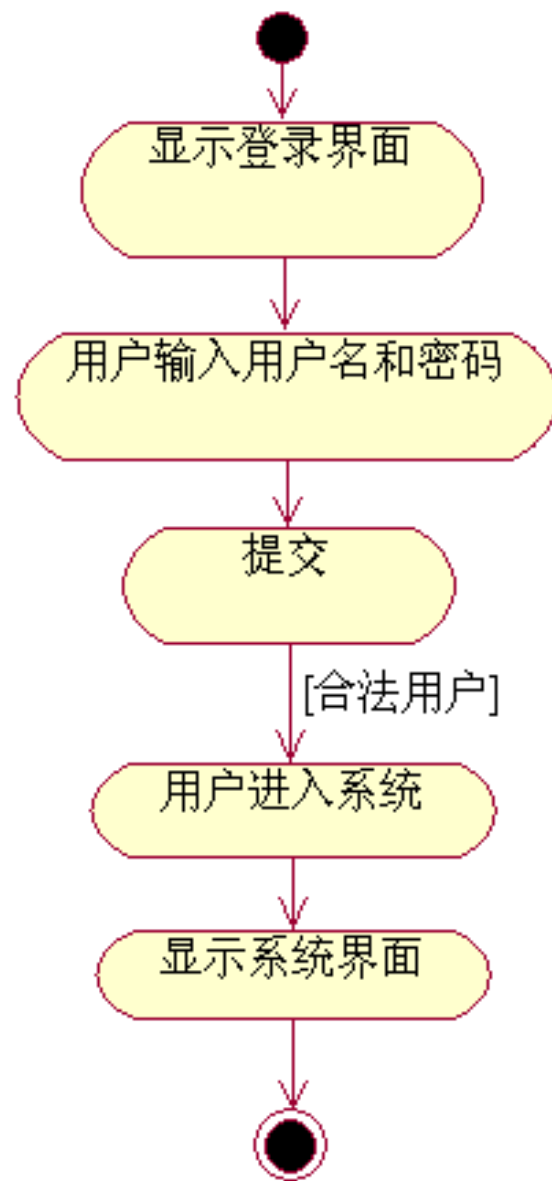
前置条件：

事件流：

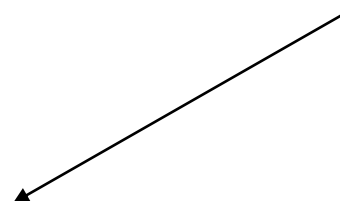
1. 当用户通过浏览器输入图书管理系统地址：library.ecust.edu.cn,用例开始。
2. 系统显示登录界面。
3. 用户输入用户名和密码。
4. 用户选择提交。
5. 系统检测用户信息，如是非法用户(if---)，系统提示用户为非法用户，同时提示用户选择退出系统或者重新登录。
 - a) 如果选择重新登录，就回到第2步；
 - b) 如果选择退出系统，用例结束。
6. 系统检测用户信息，如是合法用户(if---)，
 - a) 系统设置访问权限，并显示主界面，用例结束。

后置条件：用户进入系统或者不能进入系统。

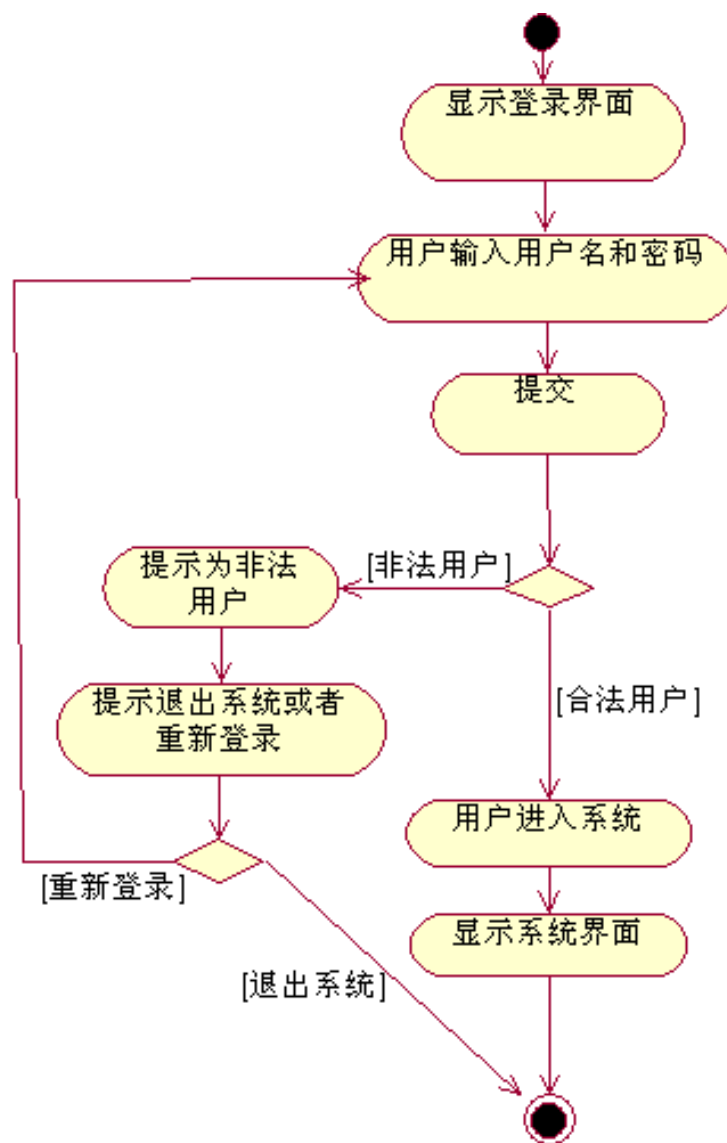
登录系统用例



Guard expression



登录系统的活动图



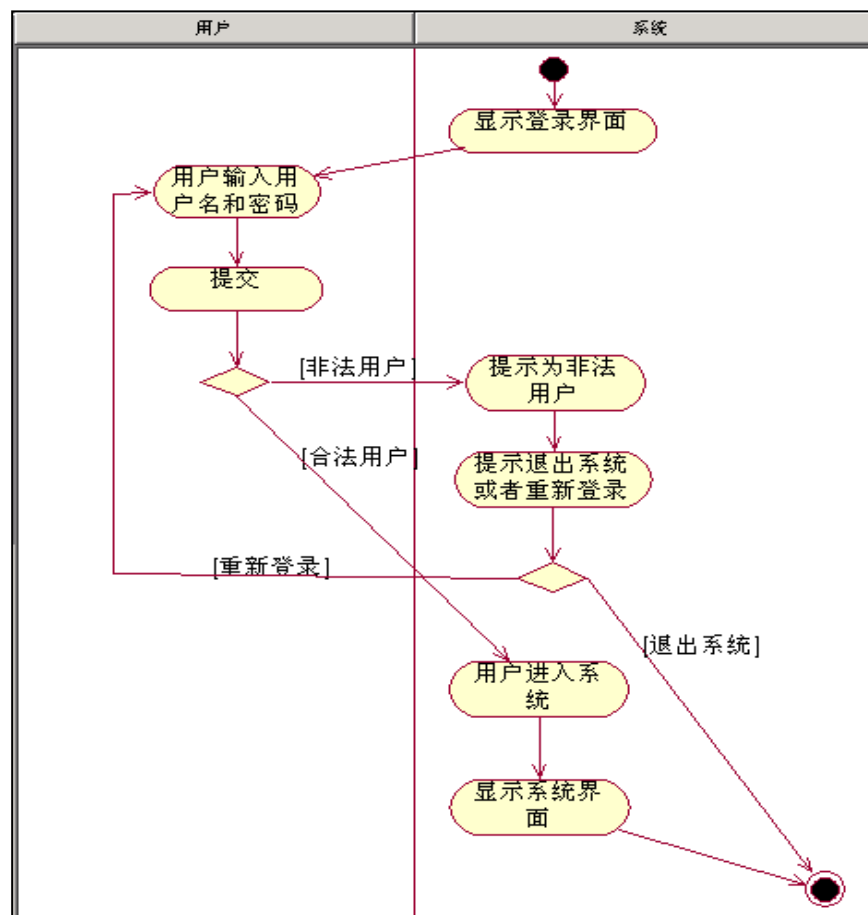
对可选路径建模后的活动图

4.4.5 Case:图书管理系统活动图建模

对登录系统用例建活动图

❖思路：先对基本用例建模，再对可选路径建模。

- 增加泳道(swimlane)后登录用例的活动图



4.4.5 Case:图书管理系统活动图建模

对借书用例建活动图

❖思路：先对基本路径建模，再对可选路径建模。

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

4.4.5 Case:图书管理系统活动图建模

对借书用例建活动图

❖ 先对基本路径建模，再对可选路径建模。

● 对借书用例的基本路径建活动图

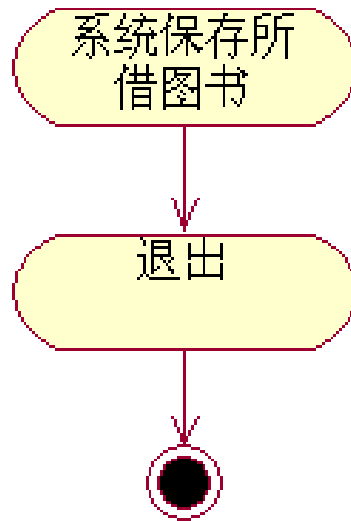
事件流：

基本路径

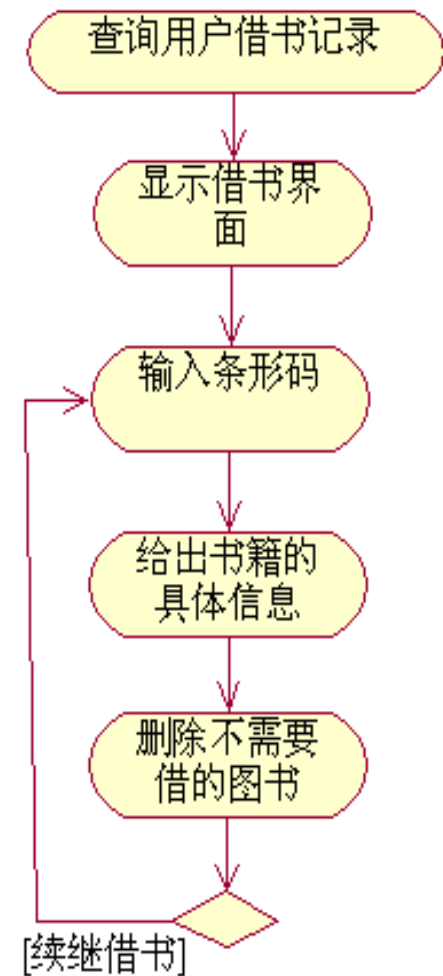
1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) 系统给出书籍的具体信息：书名、作者、出版社、日期
 - b) 用户可以删除不需要借的图书。
 - c) 系统保存所借图书。
 - d) 用户选择退出，用例结束。
- 循环结束。

1 先识别前置条件：
合法的用户

2 接下来标识后置条件：

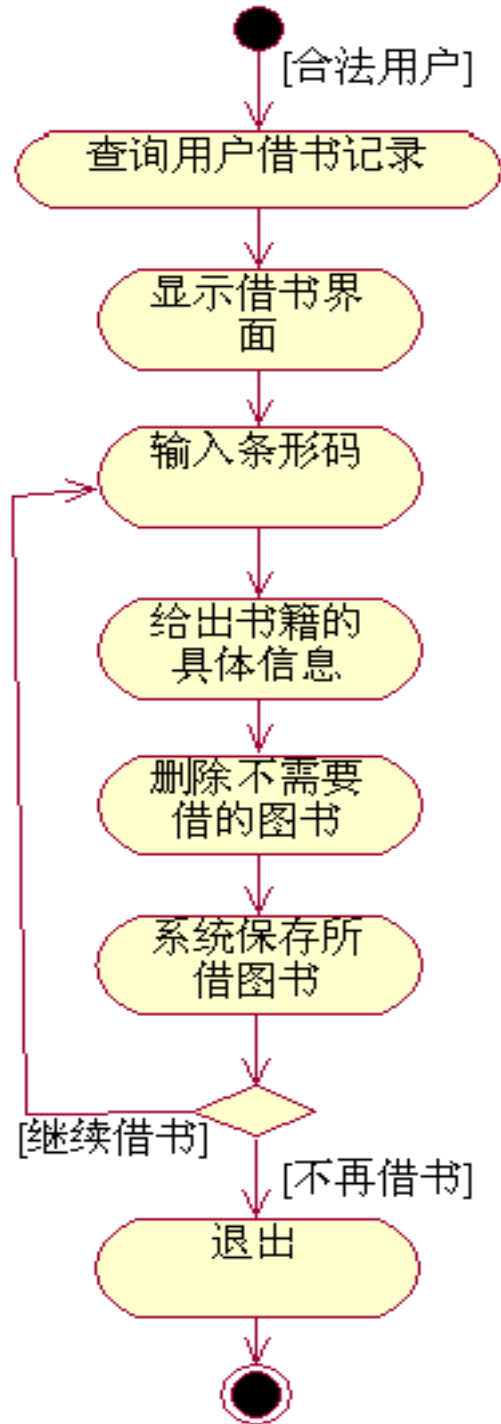


3 再接下来对其余陈述部分建模。





4 最后组织前置条件、后置条件、其余部分。



4.4.5 Case:图书管理系统活动图建模

■ 对借书用例建活动图

❖ 先对基本路径建模，再对可选路径建模。

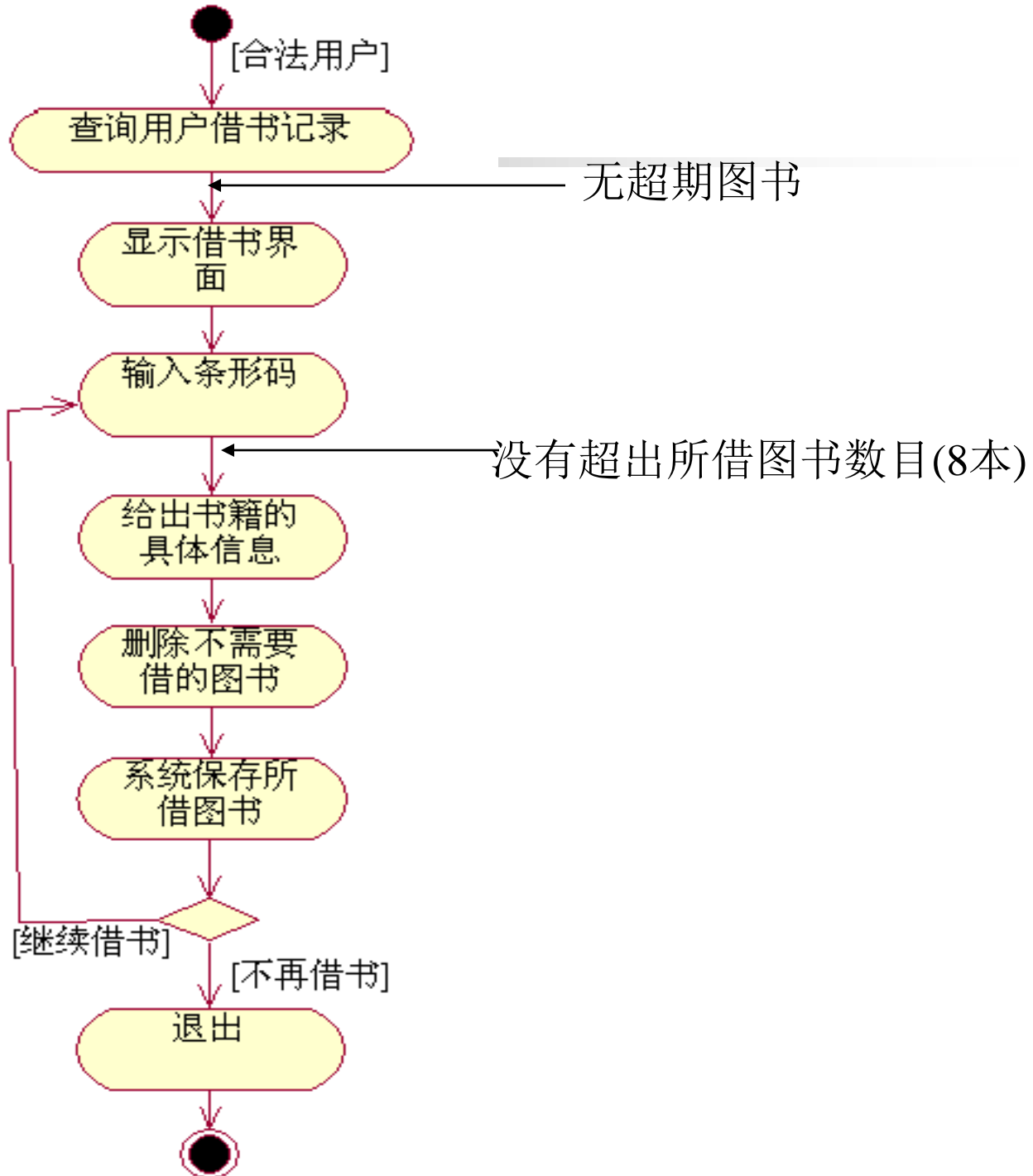
● 对借书用例的可选路径建活动图

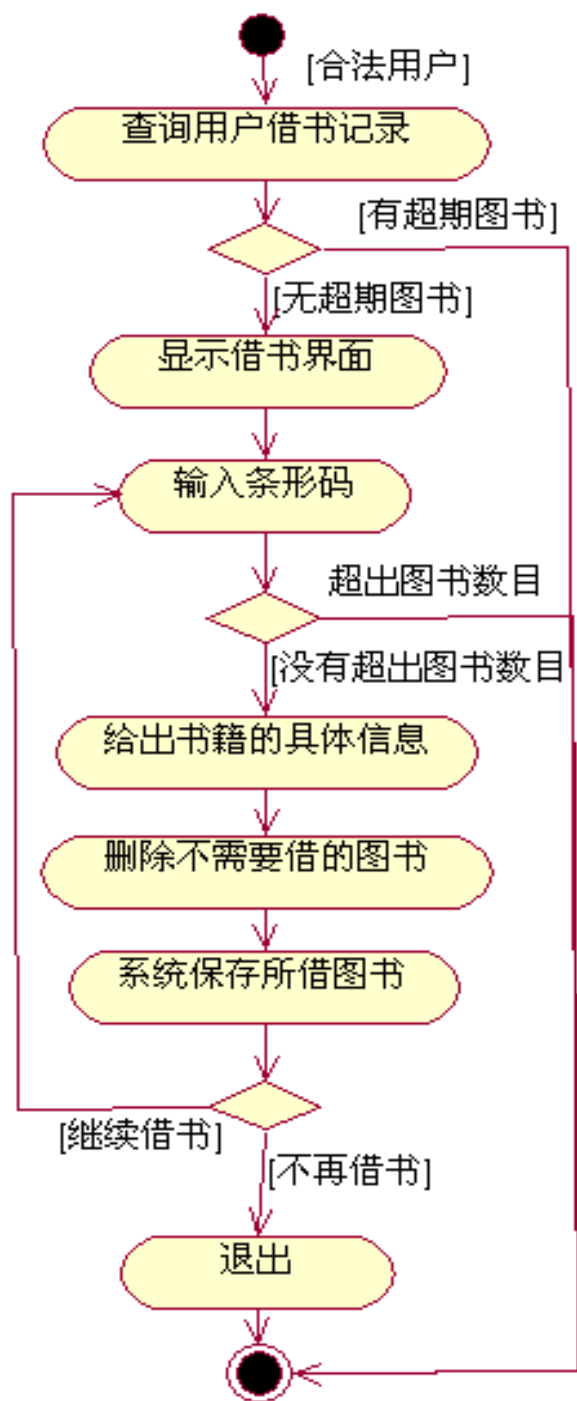
前置条件：一个合法的用户已经登录系统

事件流：

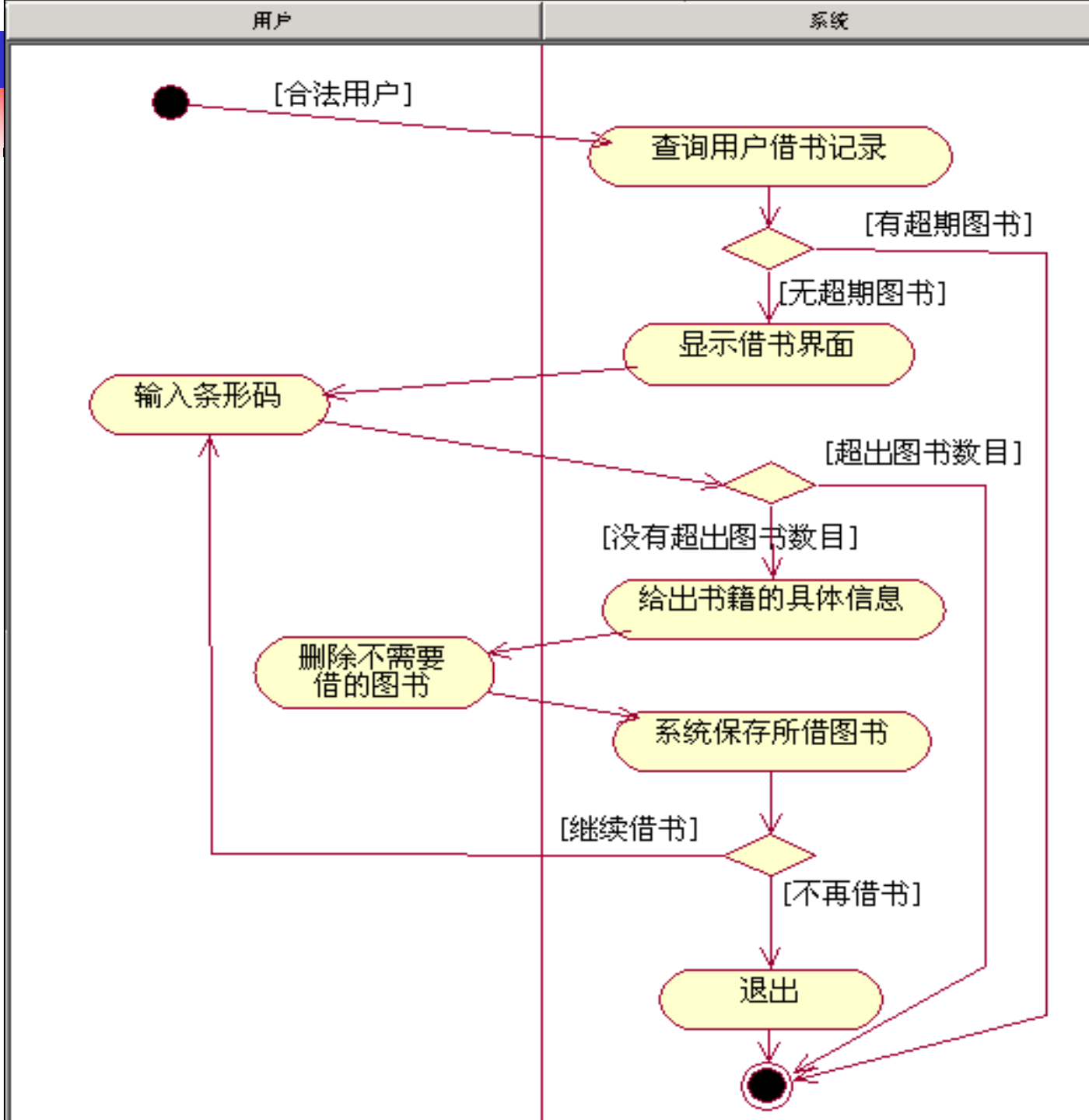
1. 当用户选择借书时,用例开始。
 2. 系统查询用户借书记录。
 - a) 如有超期图书，用例结束。
 3. 系统显示借书界面。
 4. 用户输入条形码
 - a) 系统查询用户的借书记录，如超出所借图书数目(8本)，用例结束。
 - b) 系统给出书籍的具体信息：书名、作者、出版社、日期。
 - c) 用户可以删除不需要借的图书。
 - d) 系统保存所借图书
 - e) 用户选择退出，用例结束。
- 循环结束。

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。





对借书用例的可选路径
建模后的活动图



增加泳道
(swimlane)后
借书用例的活
动图

