

信息科学与工程学院

软件工程 实验报告

**系 别 信息科学与工程学院**

**专 业 计算机科学与技术**

**年 级 2021级**

**姓 名 徐昊博**

**指导教师 阮彤**

**2024 学年 第 二 学期**

**实验三、用Rational Rose绘制UML图**

1. **实验目的**

1. 熟悉 Rational Rose 工具的使用方法。

2. 学习使用 Rational Rose 绘制 UML（Unified Modeling Language）图，包括但不限于 USE-CASE 图、类图、活动图、序列图、状态图、协作图、组件图、包图和部署图。

3. 绘制图6-2（USE-CASE）、7-4（类图）、7-8（状态图）、7-9（活动图）、7-10（带泳道的活动图）、7-7（顺序图）、8-7（部署图）中的示例图。

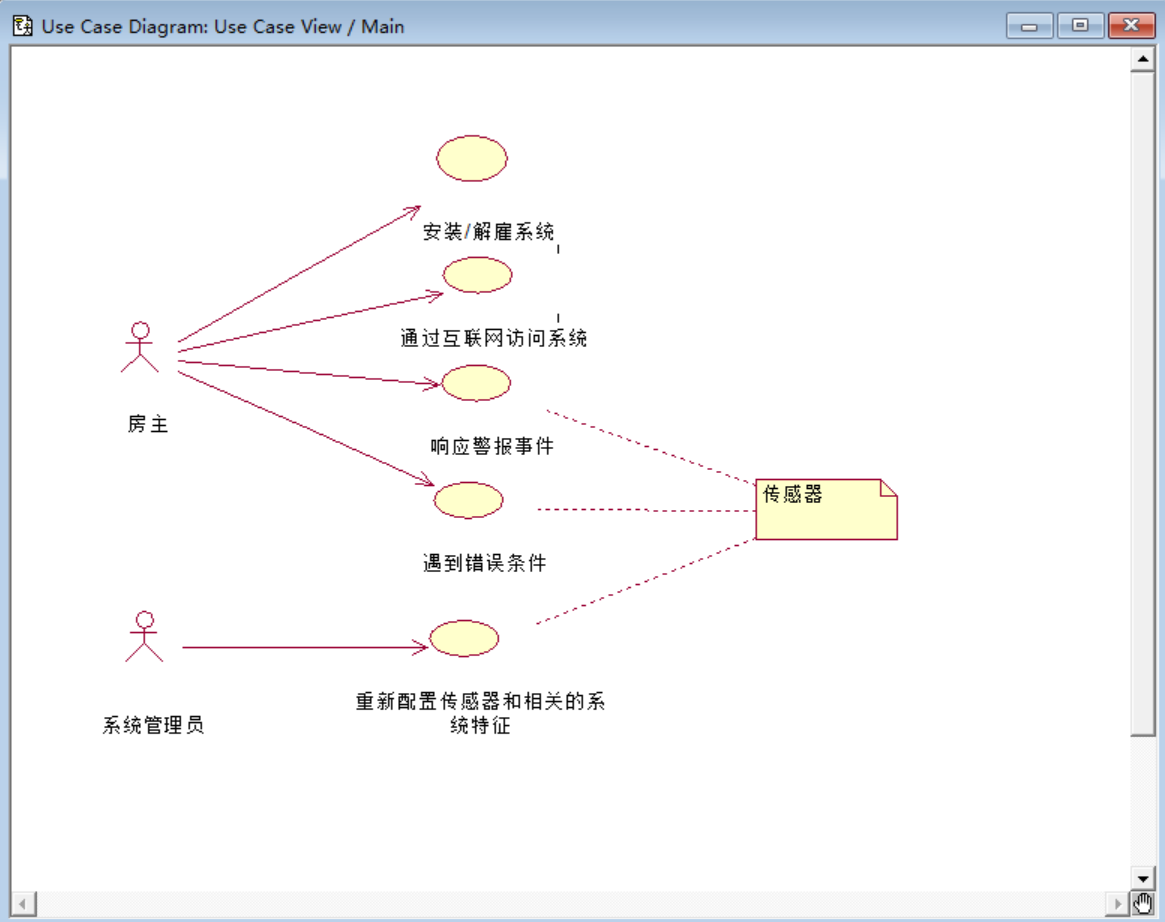
1. **实验主要内容**
2. 下载软件Rational Rose
3. 参照第九版软件工程书本第77,96,100,102,103,104,123页，绘制图6-2（USE-CASE）、7-4（类图）、7-8（状态图）、7-9（活动图）、7-10（带泳道的活动图）、7-7（顺序图）、8-7（部署图）中的示例图。。
4. **实验过程**
5. **用例图的绘制**

用例图是一种用于描述系统功能需求的 UML 图，主要由参与者（Actor）、用例（Use Case）、关系（Relationships）和箭头组成。

1. 参与者（Actor）：参与者是与系统交互并从系统中获得功能的外部实体，可以是人、其他系统或设备。在用例图中，参与者通常用图标表示，如一个小人的图标。参与者与系统之间的关系表示了参与者如何与系统中的用例进行交互。

2. 用例（Use Case）：用例表示系统提供给参与者的一项特定功能或服务。用例通常用椭圆形表示，椭圆内部包含用例的名称。用例描述了系统如何响应参与者的请求，以实现特定的目标。

3. 关系（Relationships）：用例图中的关系描述了参与者和用例之间的交互关系，以及用例之间的关系。常见的关系包括关联关系（Association）、包含关系（Inclusion）、扩展关系（Extension）、泛化关系（Generalization）等。关系用箭头表示，箭头的方向表示了参与者和用例之间的交互方向。如图下所示：



1. **类图绘制**

类图用于描述系统的静态结构，特别是类、类的内部结构以及类之间的关系。一个类图一般包含以下基本组成部分：

1. 类（Class）：表示系统中的类，通常包括类名、属性和方法。类名位于顶部，属性位于中间，方法位于底部。

2. 接口（Interface）：表示一系列操作的集合，没有属性，只有方法。接口在类图中通常用矩形框和类似类的方式表示，但标有«interface»关键字。

3. 协作（Collaboration）：表示一些类、接口和其他元素一起工作以提供某种合作行为。在类图中，协作通常用多个类或接口之间的连接线表示。

4. 关系（Relationships）：描述类之间的关系，常见的关系包括：

（1）继承（Inheritance）：表示一个类是另一个类的子类。

（2）关联（Association）：表示两个类之间的关联关系。

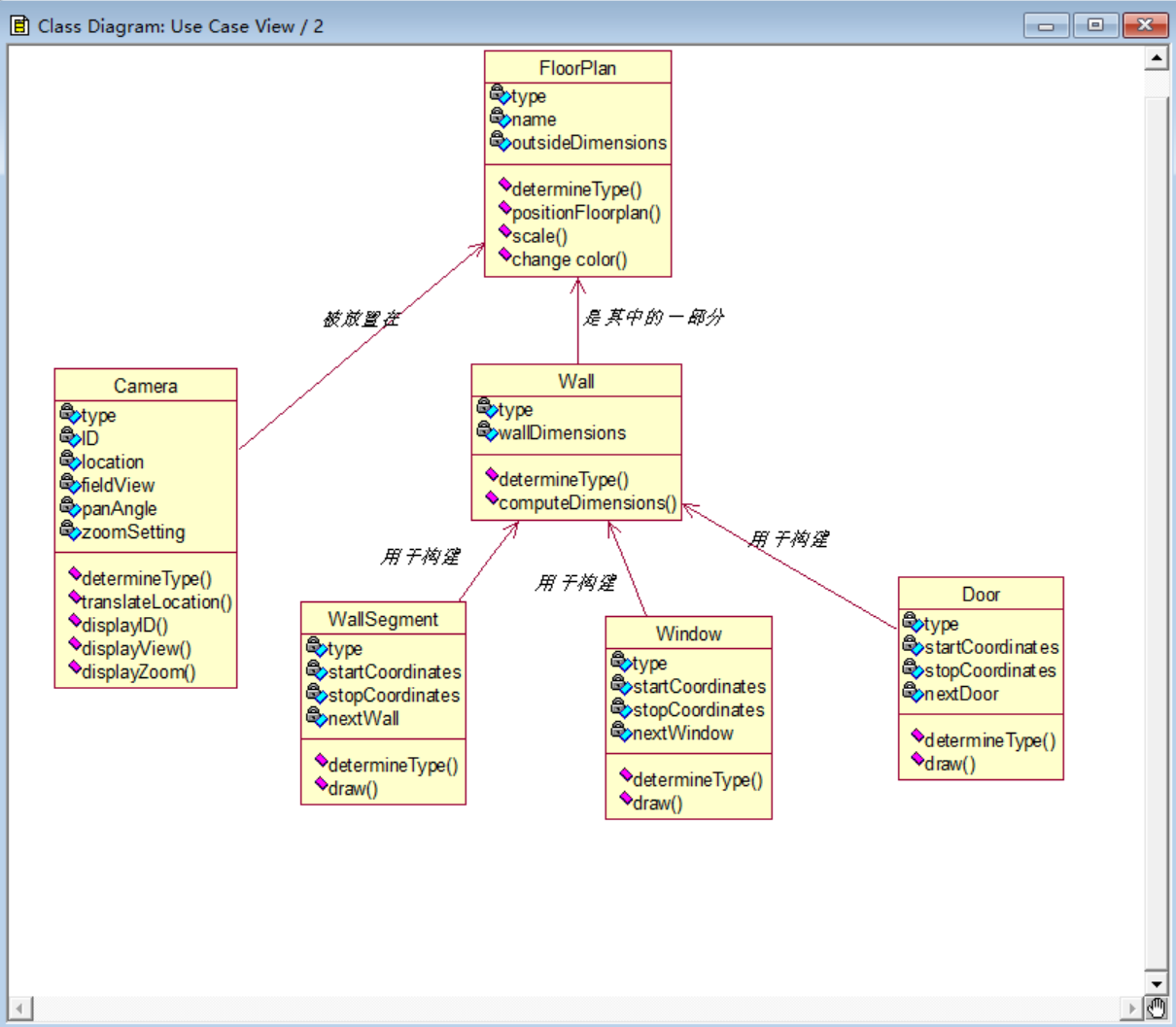
（3）聚合（Aggregation）：表示整体与部分之间的关系。

（4）复合（Composition）：表示整体与部分之间的强关联关系。

（5）依赖（Dependency）：表示一个类依赖于另一个类。

（6）实现（Realization/Implementation）：表示一个类实现了某个接口。

具体表现形式如下所示：



1. **状态图绘制**

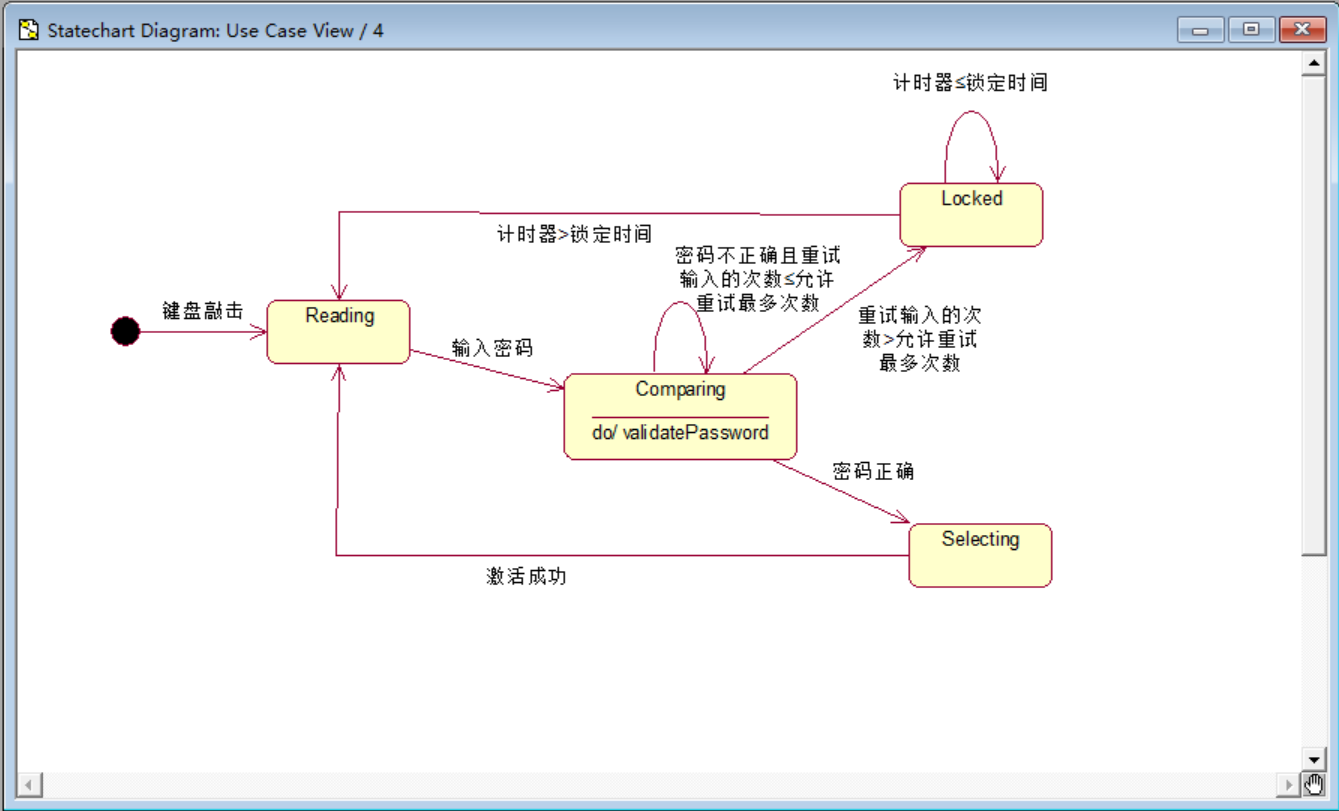
状态图用于描述对象在其生命周期中的各种状态以及状态之间的转换。状态图的各个组成部分包括：

1. 状态（State）：表示对象可能处于的某种状态。状态通常用圆角矩形表示，矩形中包含状态的名称。例如，一个订单对象可能有“新建”、“已确认”、“已发货”等状态。

2. 初始状态（Initial State）：表示对象在生命周期开始时的初始状态。初始状态通常用一个空心圆角矩形表示，箭头从初始状态指向对象的初始状态。

3. 终止状态（Final State）：表示对象在生命周期结束时的最终状态。终止状态通常用一个带双圆角矩形表示，表示对象已经完成了其生命周期。

4. 状态转换（Transition）：表示对象从一个状态转换到另一个状态。转换通常用箭头表示，箭头从一个状态指向另一个状态，表示状态之间的转换。如图下所示：



1. **活动图和带泳道的活动图的绘制**

活动图是一种 UML 图，用于描述业务流程或系统中的工作流程。一个活动图通常包括以下元素：

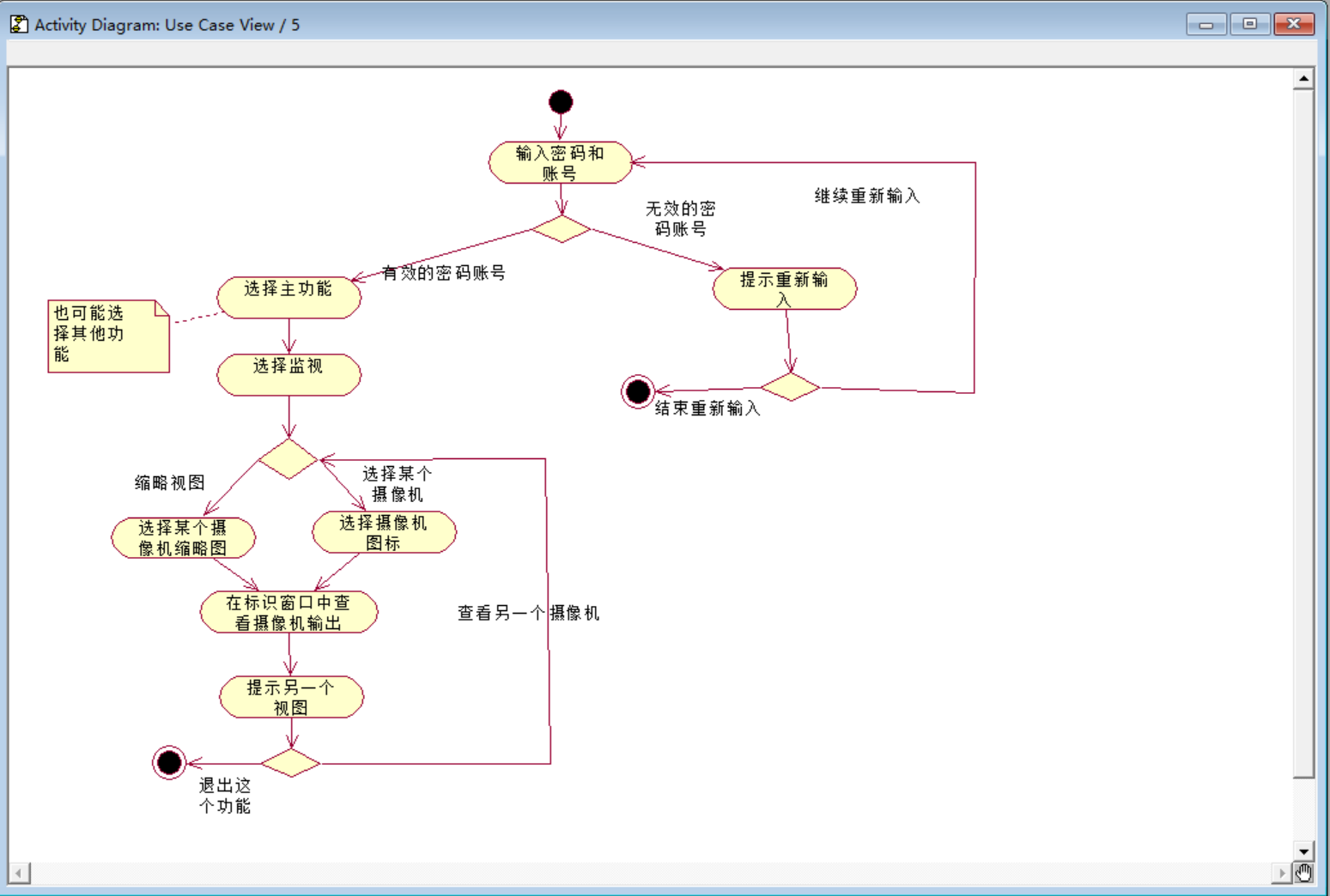
1. 活动状态（Activity State）：表示在工作流程中执行某个活动或步骤的状态。活动状态通常用圆角矩形表示。

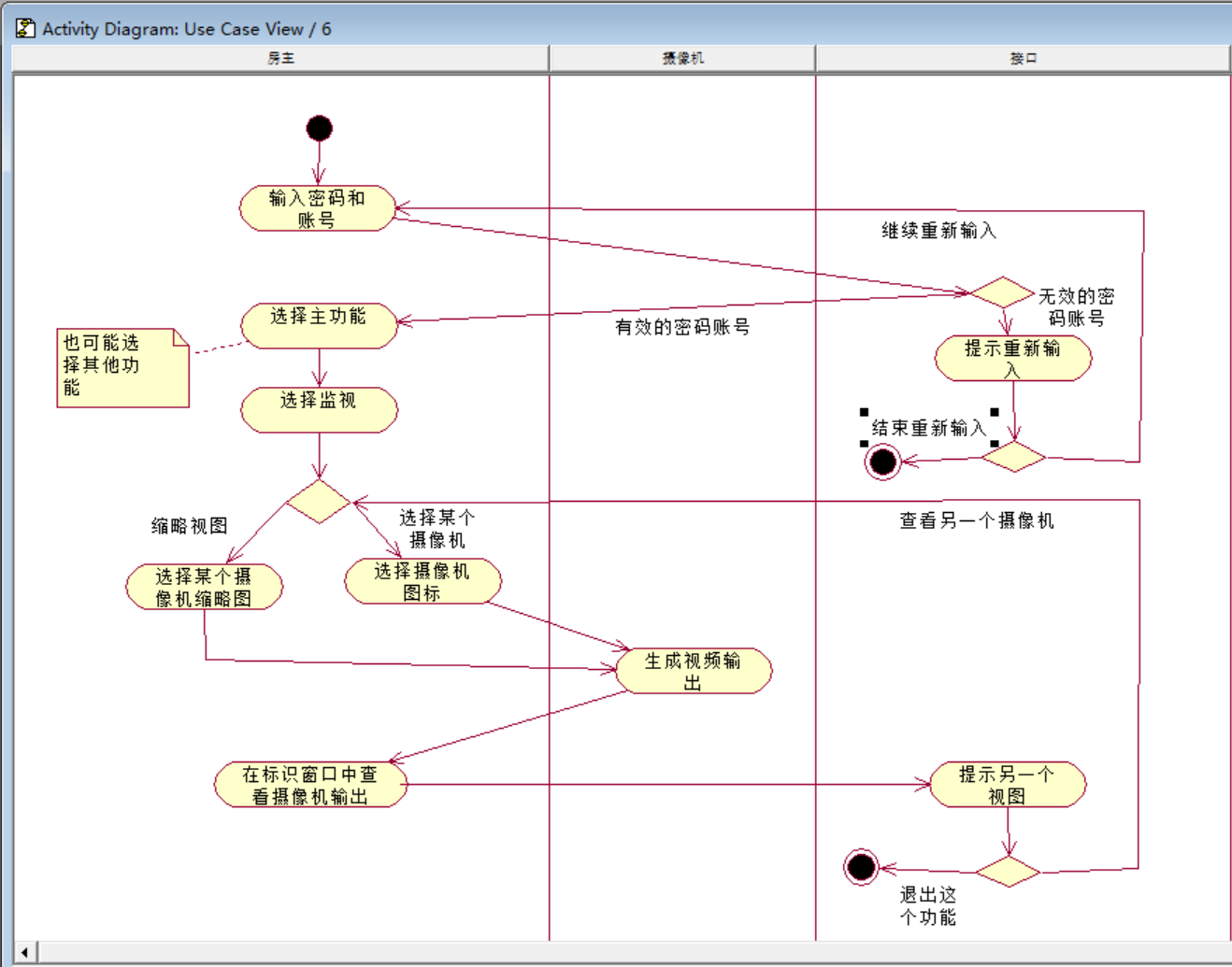
2. 转移（Transition）：表示活动状态之间的先后顺序。转移用箭头表示，箭头指向后继状态。完成转移是一种特殊的转移，它不需要明显的触发器事件，而是通过前一个活动的完成来触发。

3. 决策（Decision）：用于定义一组警戒条件，决定在活动完成后执行哪个备选转移。决策通常用菱形表示。

4. 同步示意条（Synchronization Bar）：用于显示平行分支流。同步示意条表示并行线程在某个点重新合并。同步示意条通常用双竖线表示。

5. 泳道（Swimlane）：用垂直实线将活动图划分为泳道，每条泳道代表整个工作流程中某个部分的职责，该职责由特定组织或部门执行。泳道可以帮助清晰地表示不同部门或角色在工作流程中的责任。下两张图分别表示活动图和带泳道的活动图：





1. **顺序图的绘制**

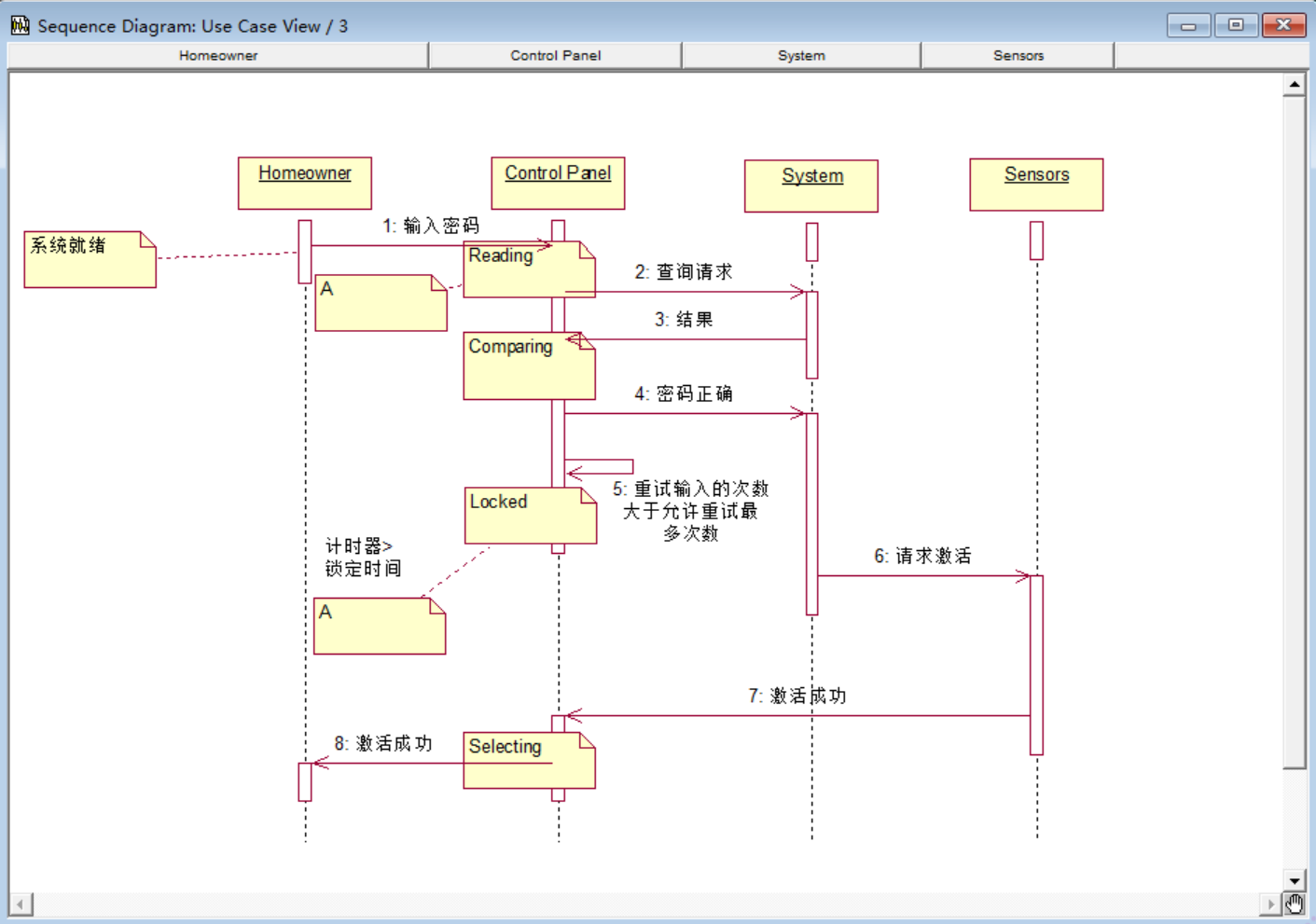
顺序图（Sequence Diagram）用于描述对象之间交互的顺序。顺序图的各个基本部分组成包括：

1. 参与者（Actor）：参与者是与系统或对象交互的外部实体，通常表示为矩形框。在顺序图中，参与者通常指的是对象或系统的角色，它们在交互中扮演特定的角色。

2. 对象（Object）：对象表示系统中的具体实例，它们可以是类的实例或者其他对象。在顺序图中，对象通常表示为矩形框，包含对象的名称和类别。

3. 生命线（Lifeline）：生命线表示对象存在的时间跨度，通常表示为一条垂直的虚线。生命线上方的小矩形表示对象在该时间段内活动的区域。

4. 消息（Message）：消息表示对象之间的交互，可以是一种请求、响应或通知。消息通常用带箭头的直线表示，箭头指向消息的接收者。消息可以是同步的（Synchronous）或异步的（Asynchronous），也可以带有返回值。如图下所示：



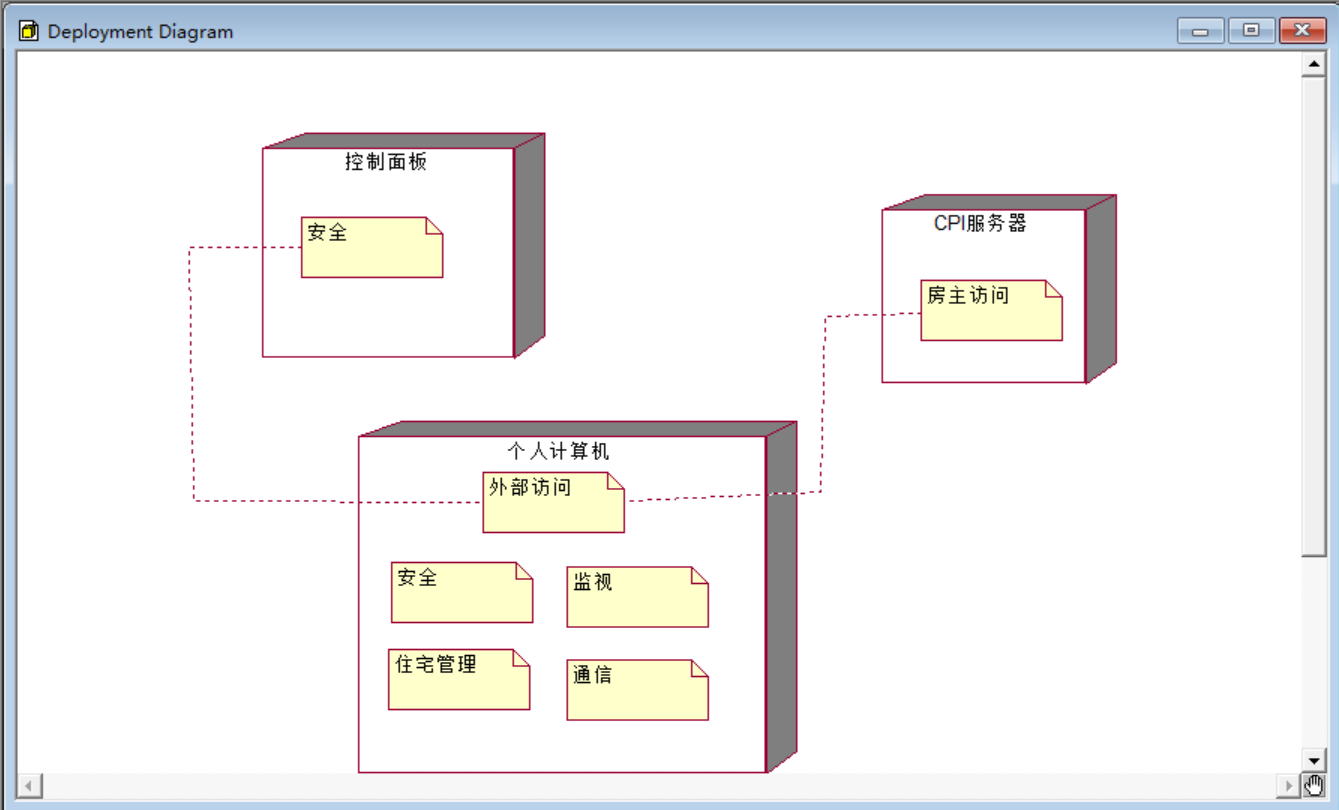
1. **部署图绘制**

部署图（Deployment Diagram）是用于描述系统中物理节点（例如服务器、计算机等）之间的部署关系以及软件组件与这些节点之间的关系。部署图的各个部分组成包括：

1. 节点（Node）：节点表示系统中的物理设备或软件执行环境，例如服务器、计算机、硬件设备等。节点通常用方框表示，并在方框内写明节点的名称。

2. 组件（Component）：组件表示系统中的软件模块或组件，例如应用程序、库、服务等。组件通常用方框表示，并在方框内写明组件的名称。

3. 连接线（Connector）：连接线表示节点之间的物理连接或软件之间的通信路径。连接线通常用实线或虚线表示，箭头指向通信的方向。如图下所示：



1. **实验相关问题**

**（1）理解UML的每个图是描述啥的？**

每个 UML 图都有其特定的用途和描述内容**：**

1. 类图（Class Diagram）：描述系统中的静态结构，包括类、类的属性和方法，以及类之间的关系。

2. 用例图（Use Case Diagram）：描述系统的功能需求，包括系统与外部参与者之间的交互以及参与者与系统之间的功能需求关系。

3. 活动图（Activity Diagram）：描述业务流程或系统中的工作流程，展示了各种活动状态之间的转移和条件。

4. 顺序图（Sequence Diagram）：描述对象之间的交互顺序，展示了对象之间消息的传递和响应顺序。

5. 状态图（State Diagram）：描述对象在其生命周期中的各种状态以及状态之间的转换。

6. 部署图（Deployment Diagram）：描述系统的物理部署架构，包括节点（服务器、计算机等）和软件组件之间的部署关系。

每种 UML 图都可以单独使用，也可以与其他图结合使用，以更全面地描述系统的不同方面。不同的图适用于不同的场景和目的，通过综合运用可以更好地理解和设计复杂系统。

**（2）说明Rational Rose建模软件和一般绘图软件的差别**

与一般的绘图软件相比，Rational Rose 具有以下特点和差别：

1. 专注于建模：Rational Rose 主要用于软件系统的建模和设计，提供了丰富的 UML 图元素和工具，支持软件工程师进行系统的需求分析、设计和文档化。

2. 支持 UML：Rational Rose 针对 UML 标准进行了设计和开发，可以方便地创建符合 UML 标准的各种类型的图，而一般的绘图软件可能没有针对 UML 的特定支持。

3. 模型驱动开发：Rational Rose 支持模型驱动开发（Model-Driven Development），可以根据建模结果自动生成代码或者其他文档，从而加快软件开发过程。

4. 专业性：Rational Rose 是一款专业的建模工具，具有较高的建模精度和专业性，适用于大型软件系统的建模和设计。

5. 学习曲线：相对于一般的绘图软件，Rational Rose 学习曲线可能更陡峭一些，因为需要掌握 UML 的概念和建模方法。

综上所述，Rational Rose 与一般的绘图软件相比，更适合于软件工程领域的建模和设计需求，提供了更专业、更全面的建模工具和功能。

**五、实验总结和收获**

通过本次UML图绘制的实验，学习了如何使用 Rational Rose 创建和编辑不同类型的 UML 图形，加深了对 UML 建模的理解和掌握。同时，也了解到了 Rational Rose 在建模工具中的应用和优势。掌握了使用 Rational Rose 进行 UML 建模的基本方法和技巧，为今后在软件工程领域的建模和设计工作打下了基础。