

计算机科学系实验报告

课程名称	<u>UML 与可视化建模</u>	班级	<u>14 网络 1 班</u>		
实验名称	<u>虚拟社交世界</u>	指导教师	<u>曾少宁</u>		
姓名	<u>陈鑫</u>	学号	<u>1414080903122</u>	日期	<u>2017.06.07</u>

一、实验目的

掌握基于 UML 2.0 的建模概念与方法，掌握各种 UML 图的概念与画法，其中包括用例图、活动图、类图、顺序图、组件图和状态图等。

二、实验设备与环境

操作系统：Windows 7；建模工具：StarUML。

四、实验要求

1. 实验及实验报告以增量方式完成，每次作业都在上一次作业的基础上完成，作业提交网站不提供报告下载，所以请同学们自行保管好自己的实验报告；
2. 请将实验报告中“占位符”信息替换为自己的实验相关信息；
3. 请认真撰写实验体会，**实验课结束时**立即上传实验报告：<http://zeng.shaoning.net/uml/>。

四、实验内容、程序清单及运行结果

虚拟社交世界

面对用户：

- 1、私人聊天功能
- 2、修改个人资料

1、实验一：需求建模 - 用例模型

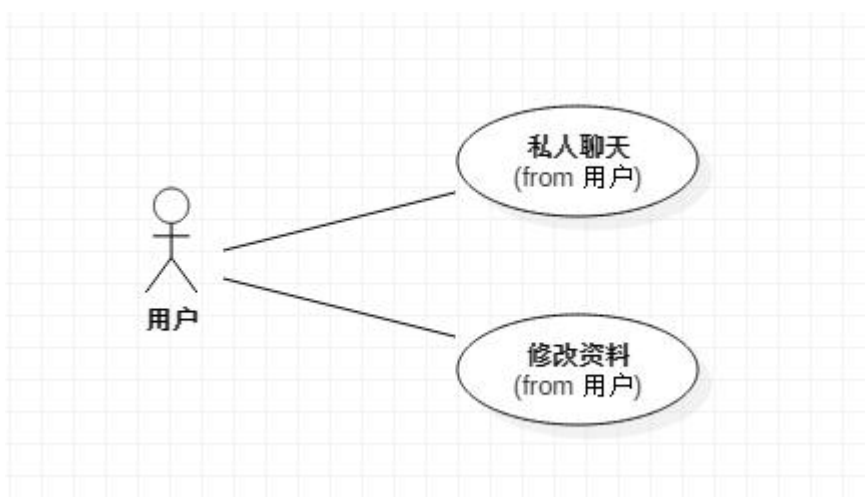


图 1：用户用例图

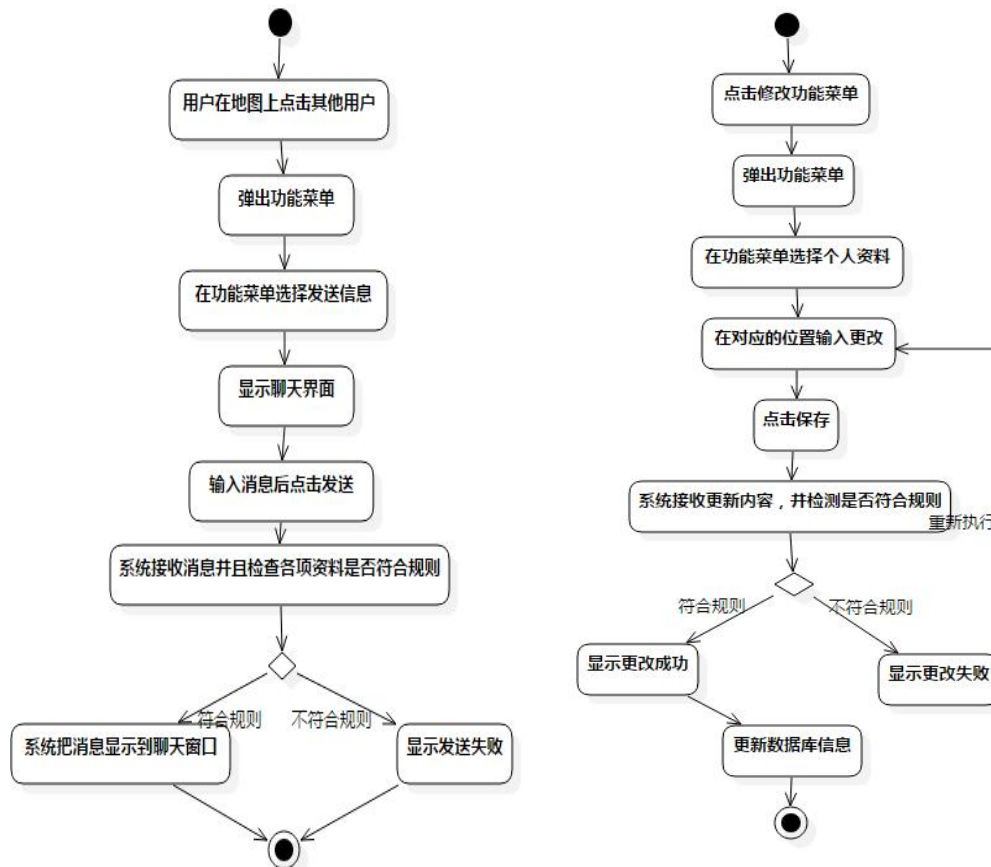
注：用例规约内容及项目可自行增加。

用例编号：	UC001
用例名称：	私人聊天
用例描述：	用户可向其他用户建立私人聊天频道
前置条件：	用户已经登录到系统
基本流程：	<ol style="list-style-type: none"> 1、用户在地图上点击其他用户； 2、弹出功能菜单 3、在功能菜单选择发送信息 4、显示聊天界面 5、输入消息后点击发送 6、系统接收消息并且检查各项资料符合规则 7、系统把消息显示到聊天窗口
扩展流程：	<ol style="list-style-type: none"> 2.1 功能菜单：发送消息，拉入黑名单，添加好友 6.1 系统发现目标在用户黑名单里，提示对方已将你拉黑 6.2 系统检测消息含有敏感字，提示消息含有敏感字
后置条件：	建立聊天频道后，系统可保存一段时间的聊天记录

用例编号：	UC002
用例名称：	修改个人资料
用例描述：	用户可在界面修改个人资料
前置条件：	用户已登录到系统
基本流程：	<ol style="list-style-type: none"> 1、点击修改功能菜单 2、弹出功能菜单 3、在功能菜单选择个人资料 4、在对应的位置输入更改

	5、点击保存 6、系统接收更新内容，并检测符合规则 7、系统更新数据库 8、显示更改成功
扩展流程:	6.1 用户名不符合规则，提示用户名不符合规则 6.2 用户上传的头像不符合规则，提示头像未通过审核 6.3 用户更改的角色形象不符合规则，提示角色形象不符合规则
后置条件:	

2、实验二：过程建模 – 活动模型

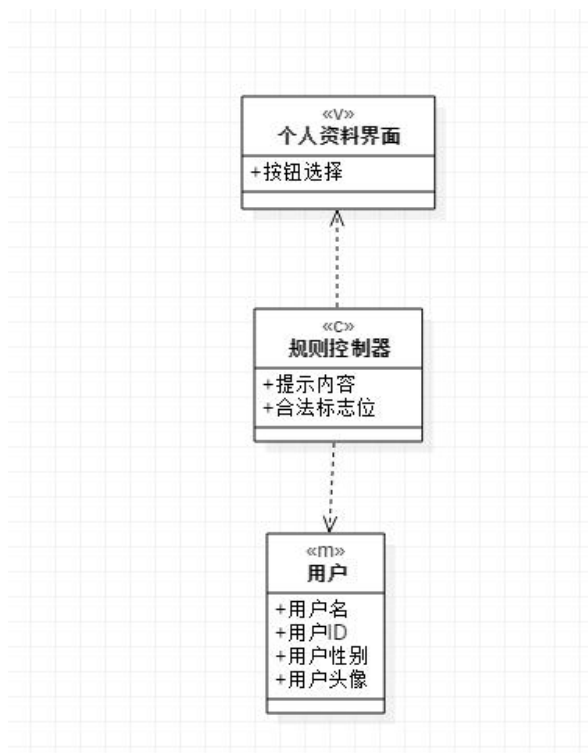
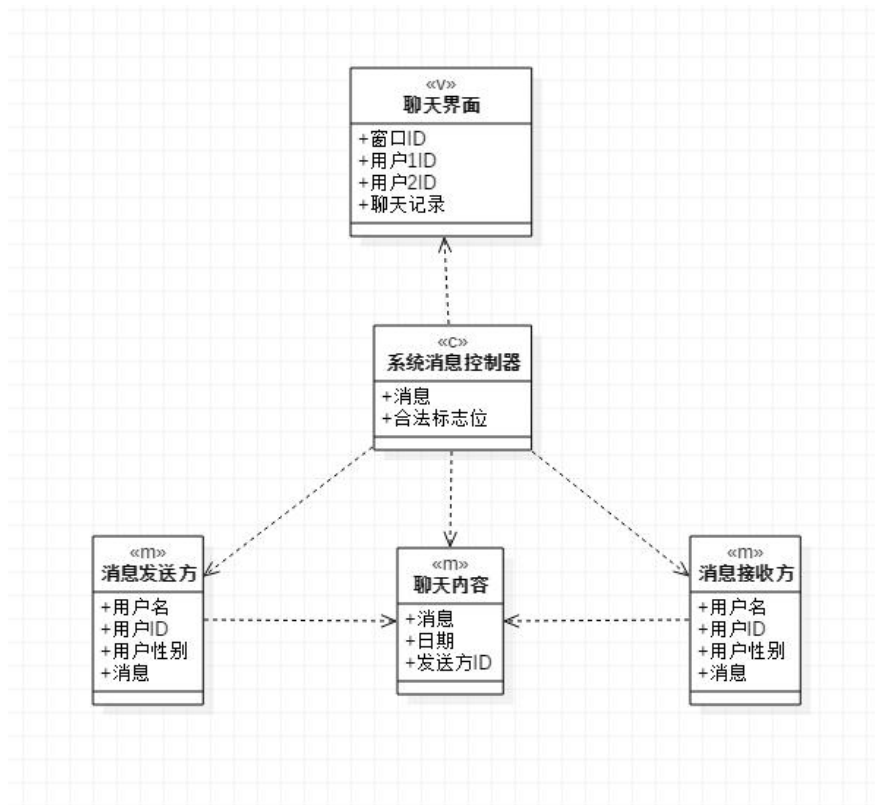


3、实验三：逻辑建模 – 类模型

基于 MVC 设计模式找出实现用例的类。

方法：分别找出实现用例的模型（Model）、视图（View）和控制器（Controller）类，确定类之间的关系及其关键属性，画出类图。

参考：讲义 P26 页。

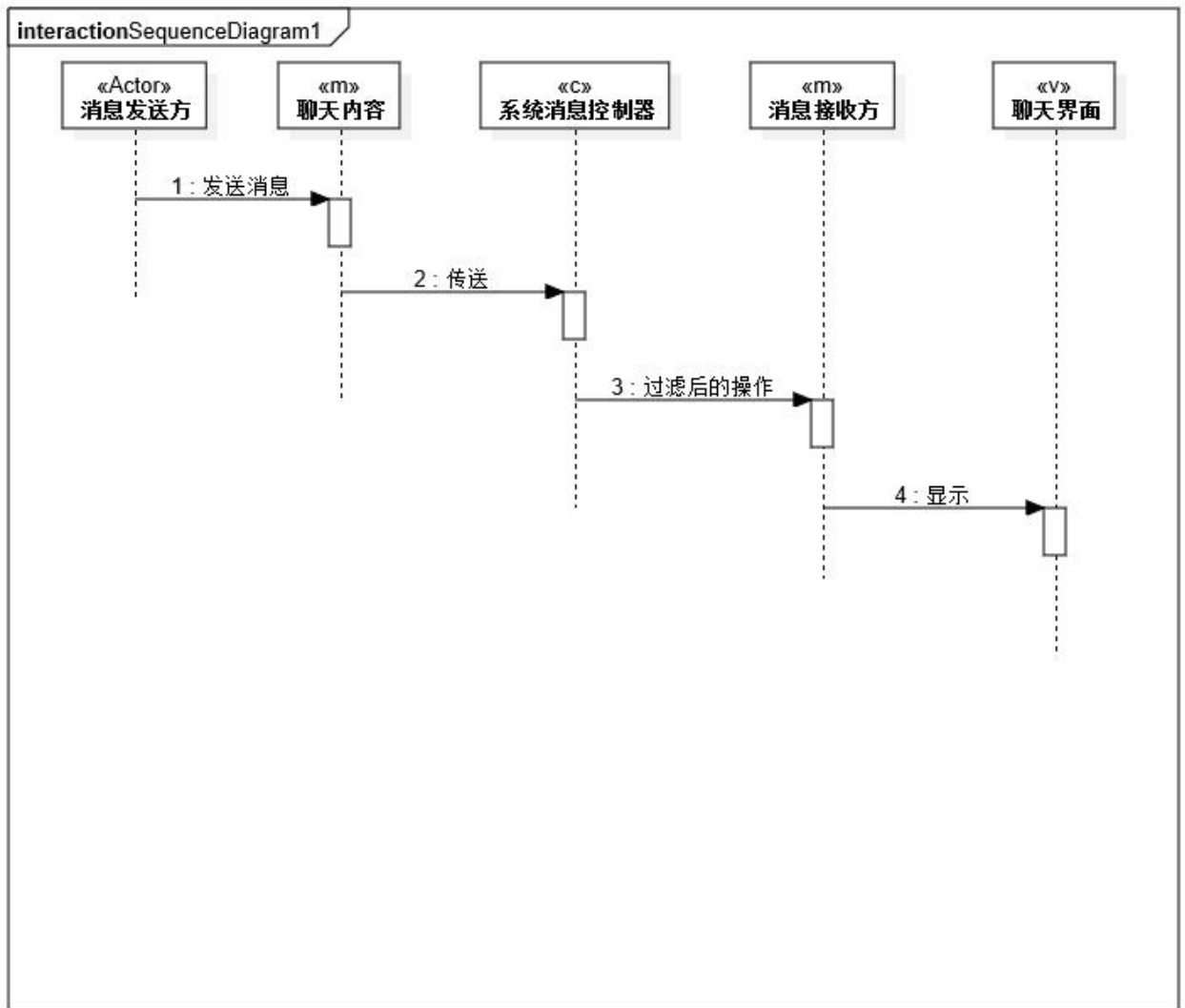


4、实验四：交互建模 – 顺序模型

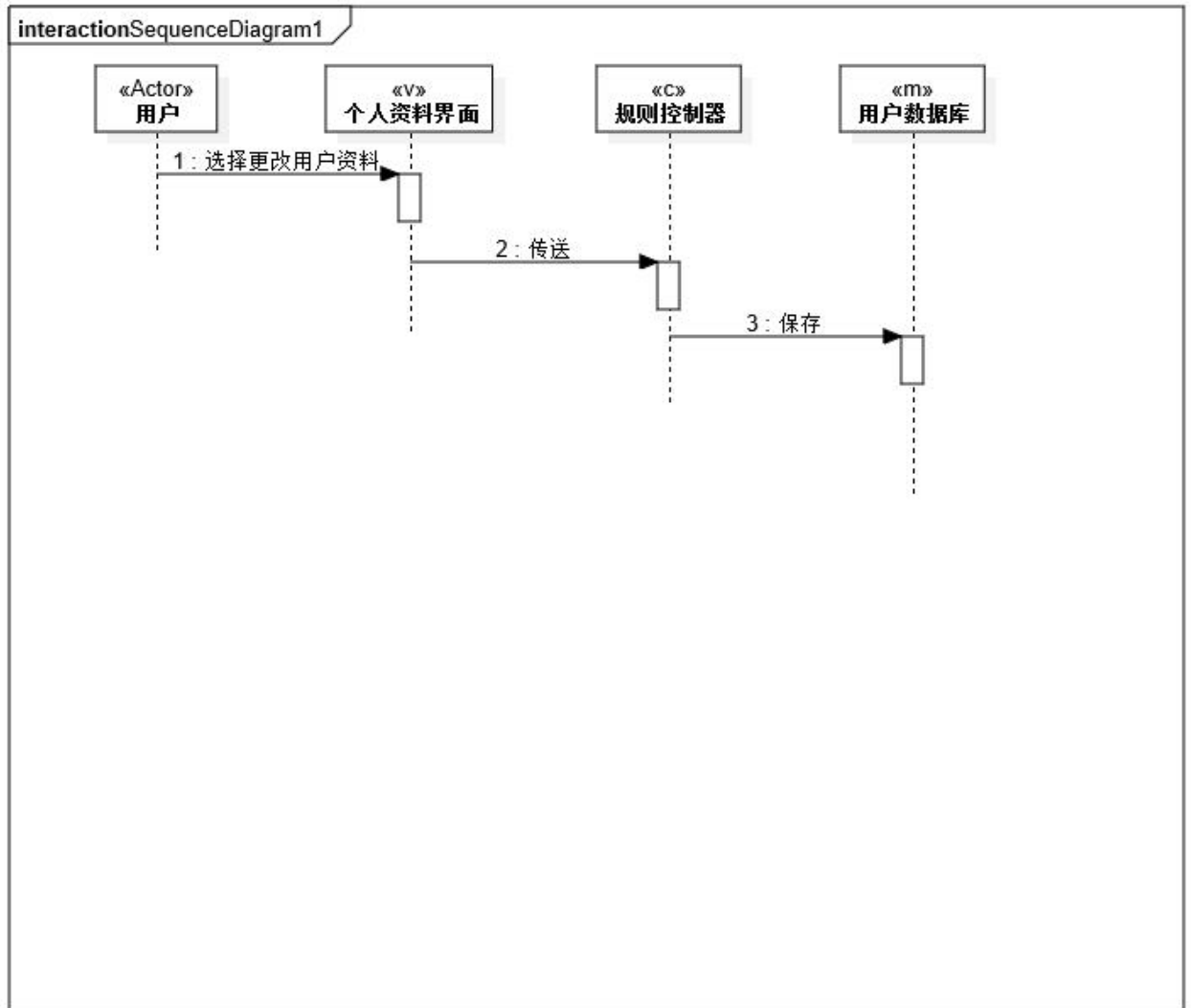
创建各个类（MVC 及 Actor）的对象，并描述对象之间的交互。

方法：分别创建参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）的对象，描述各个对象之间的消息及其顺序，画出顺序图。

参考：讲义 P33 页 8.7.2。



顺序图 1



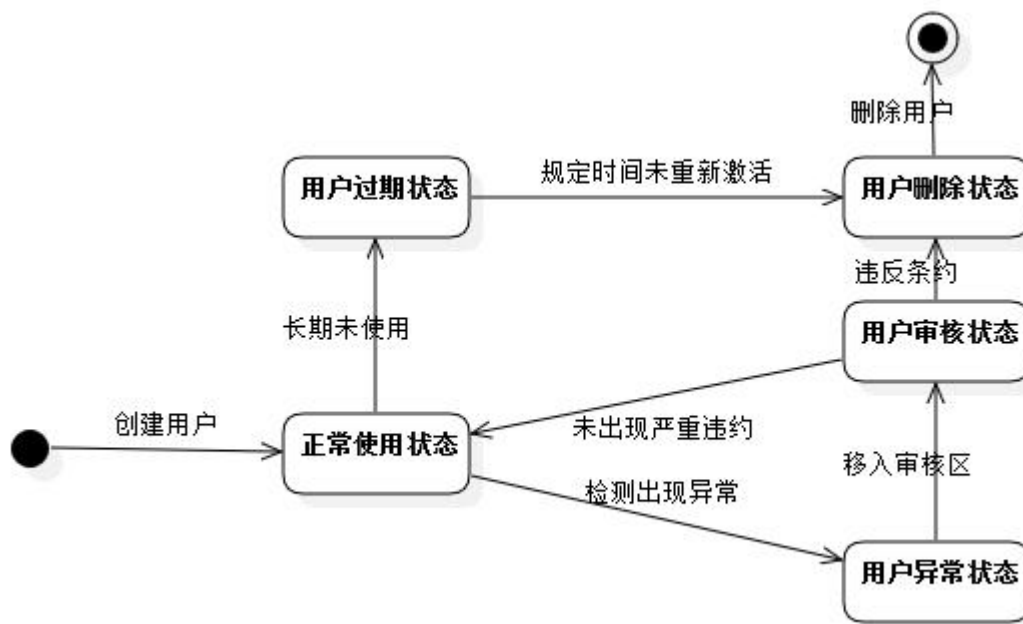
顺序图 2

5、实验五：状态建模 – 状态模型

对系统中最重要的对象进行状态建模。

方法：选择一种对象，定义该对象的状态，描述状态之间的切换及条件，画出状态图。

参考：讲义 P9 和 P10 页。



五、实验体会

实验一：

本次实验完成对所选题目的需求建模，通过 StarUML 工具绘制用例图，并给出用例规约。在绘制用例图的过程中，顺利地绘制出简单的用例图。

➤ 概念解释：用例

用例图主要用来描述“用户、需求、系统功能单元”之间的关系。它展示一个外部用户能够观察到的系统功能模型图。用例图多用于静态建模阶段（主要是业务建模和需求建模），帮助开发团队以一种可视化的方式理解系统的功能需求。

实验二：

本次实验主要完成活动图的绘制，以描述系统业务流程。实验通过活动图的绘制，将用例规约中的基本流程与扩展流程抽象为过程步骤，以活动图的形式呈现。

➤ 概念解释：活动图

活动图是 UML 用于对系统的动态行为建模的另一种常用工具，它描述活动的顺序，展现从一个活动到另一个活动的控制流。活动图在本质上是一种流程图。活动图着重表现从一个活动到另一个活动的控制流，是内部处理驱动的流程。

实验三：

本次实验基于 MVC 设计模式绘制了类图。通过找出用例的模型（Model）、视图（View）和控制器（Controller）类，确定类之间的关系及其关键属性，画出类图。本次实验难点在于各个类之间关系的理解，大体分为依赖关系、关联关系、聚合关系、组合关系、继承关系五种。类图用来定义系统中的类，包括描述类的结构和类之间的关系，主要作用是描述系统的静态结构。本次实验让我对类的概念有了进一步的理解，掌握了在 UML 建模语言中如何表示类之间的关系。

➤ 概念解释：类图

类图展示了系统的逻辑结构，类和接口的关系。类图表示不同的实体(人、事物和数据)如何彼此相关，显示了系统的静态结构。类图可用于表示逻辑类，逻辑类通常就是业务人员所谈及的事物种类，类图还可用于表示实现类，实现类就是程序员要编写的类。

实验四：

在本次实验中，创建了参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）

对象，并描述各个对象之间的消息及其顺序，画出对象之间的顺序图。有了之前实验的参考，所以相对比较容易。

➤ 概念解释：顺序图

顺序图用于显示对象是如何在一个给定的情境下进行交互的。顺序图的一个重要特征是时间从顶部到底端：交互从图例顶端开始，终止于底部，也就是说，越低越晚。顺序图的一个典型用法是用来演示面向对象的系统中的动态演化。对于每个关键的协作而言，就会为之建立一些顺序图来演示该协作各个场景中对象是如何进行交互的。

实验五：

本次实验完成了对系统中最重要的课程对象进行状态建模，描述了课程对象状态之间的切换及条件。实验较为简单，操作难度不大。

➤ 概念解释：状态图

状态图（*Statechart Diagram*）主要用于描述一个对象在其生存期间的动态行为，表现为一个对象所经历的状态序列，引起状态转移的事件（*Event*），以及因状态转移而伴随的动作（*Action*）。一般可以用状态机对一个对象的生命周期建模，状态图用于显示状态机（*State Machine Diagram*），重点在与描述状态图的控制流。