

# 计算机科学系实验报告

课程名称 UML 与可视化建模 班级 14 计科 2 班  
实验名称 视频监控管理系统 教导教师 曾少宁  
姓名 易文豪 学号 1414080901240 日期 2017.05.19

## 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

题目：视频监控管理系统

功能 1：配置摄像头。对未配置的摄像头进行配置，或修改已配置摄像头的配置。

功能 2：实时监控。实时绘制摄像头采集到的视频数据。

功能 3：导出录像。把存储在内置存储器的录像导出到外部存储器，如 U 盘。

### 1、实验一：需求建模 - 用例模型

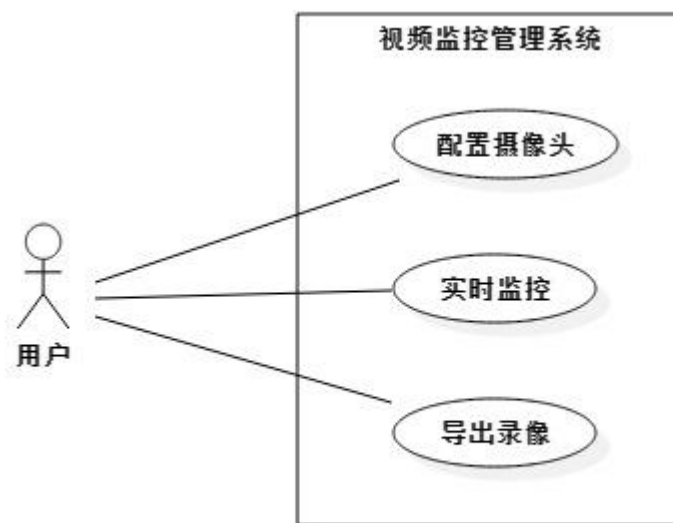


图 1：用例图

注：用例规约内容及项目可自行增加。

用例编号：	UC001
用例名称：	配置摄像头
用例描述：	
前置条件：	
基本流程：	<ol style="list-style-type: none"> <li>1. 用户打开“配置”窗口，系统枚举所有可发现的摄像头，用列表显示出来。</li> <li>2. 用户双击列表中的一个摄像头，系统弹出“配置参数”对话框，用户填写 FPS、压缩格式、输出大小。</li> <li>3. 用户点击“确定”按钮，系统把参数保存到数据库。</li> <li>4. 系统关闭对话框，返回“配置”窗口。</li> </ol>
扩展流程：	<ol style="list-style-type: none"> <li>3.1 如果存在没有填写的参数项目，系统弹出错误信息，如“您没有填写 XXX”。</li> <li>3.2 如果数据库中已经存在参数记录，系统更新参数记录。</li> </ol>
后置条件：	

用例编号：	UC002
用例名称：	实时监控
用例描述：	
前置条件：	
基本流程：	<ol style="list-style-type: none"> <li>1. 系统在主窗口过程，从数据库中查询参数记录。</li> <li>2. 系统用参数初始化摄像头。</li> <li>3. 系统从摄像头中采集视频数据，在子窗口中绘制图像，并存储视频数据到内置存储器的文件中。</li> </ol>
扩展流程：	<ol style="list-style-type: none"> <li>2.1 如果无法初始化摄像头，如连接不上（如摄像头已移除、已损坏）、配置不了（摄像头已不支持某参数），就参数记录从数据库中删除。</li> <li>3.1 如果没有为某个摄像头创建子窗口，就创建一个。</li> <li>3.2 录像文件按某种固定的格式来命名，如“摄像头 ID-时间段.后缀”。</li> <li>3.3 如果无法存储视频数据到内置存储器的文件中（如存储空间不</li> </ol>

	足、内置存储器损坏等), 系统就不保存视频数据, 但不会停止绘制图像, 而是在子窗口中绘制出错信息, 如在子窗口的左上角绘制“存储空间不足”的字符串等。
后置条件:	

用例编号:	UC003
用例名称:	导出录像
用例描述:	
前置条件:	
基本流程:	<ol style="list-style-type: none"> <li>1. 用户打开“导出”窗口, 系统从内置存储器中枚举所有录像文件, 用列表显示出来。</li> <li>2. 用户选中列表中的一个或者多个录像文件, 点击“导出”按钮。</li> <li>3. 系统弹出“选择目标外置存储器”对话框, 枚举所有已连接的外置存储器, 用列表显示出来。</li> <li>4. 用户选中一个或者多个目标外置存储器, 点击“确定”按钮。</li> <li>5. 系统把录像文件复制到目标外置存储器。</li> <li>6. 系统关闭“选择目标存储器”对话框, 返回“导出”窗口。</li> </ol>
扩展流程:	<ol style="list-style-type: none"> <li>3.1 如果用户没有选中录像文件, 系统无对应的响应。</li> <li>5.1 如果用户没有选中外置存储器, 系统无对应的响应。</li> <li>5.2 如果系统不能够完成复制操作, 如外置存储器是只读设备, 或者存储空间不足等, 就弹出出错信息。</li> </ol>
后置条件:	

## 2、实验二：过程建模 – 活动模型

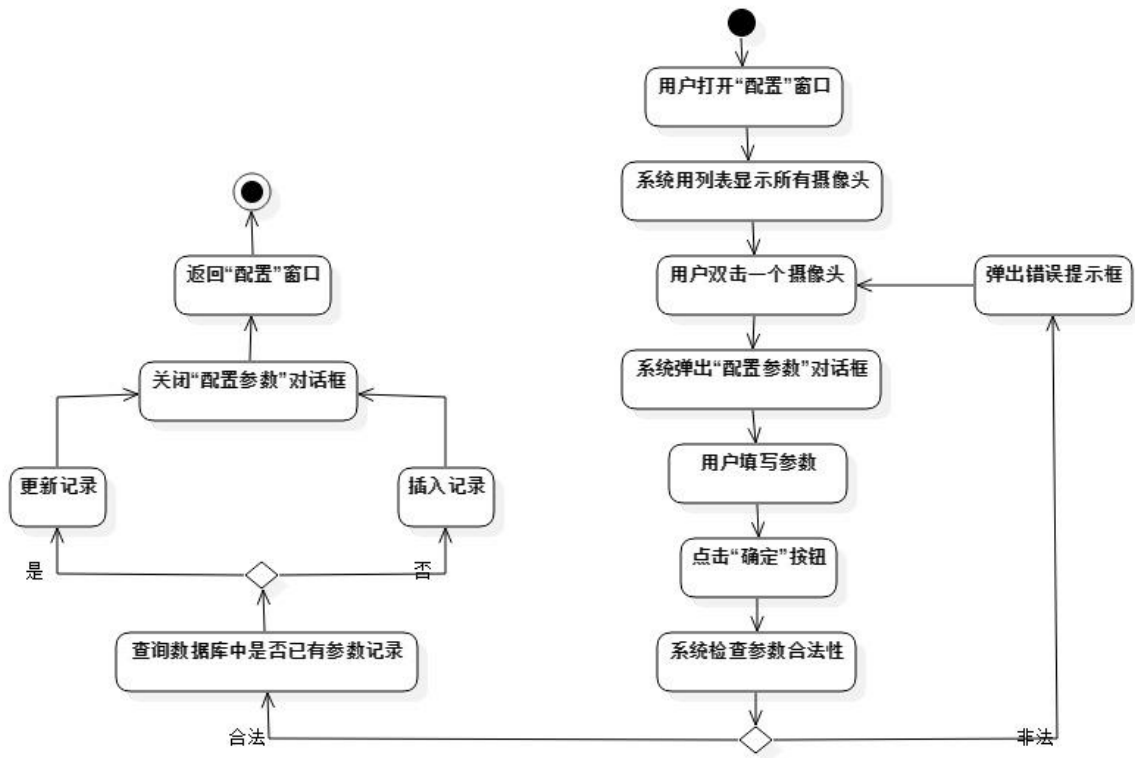


图 1：“配置摄像头”活动图

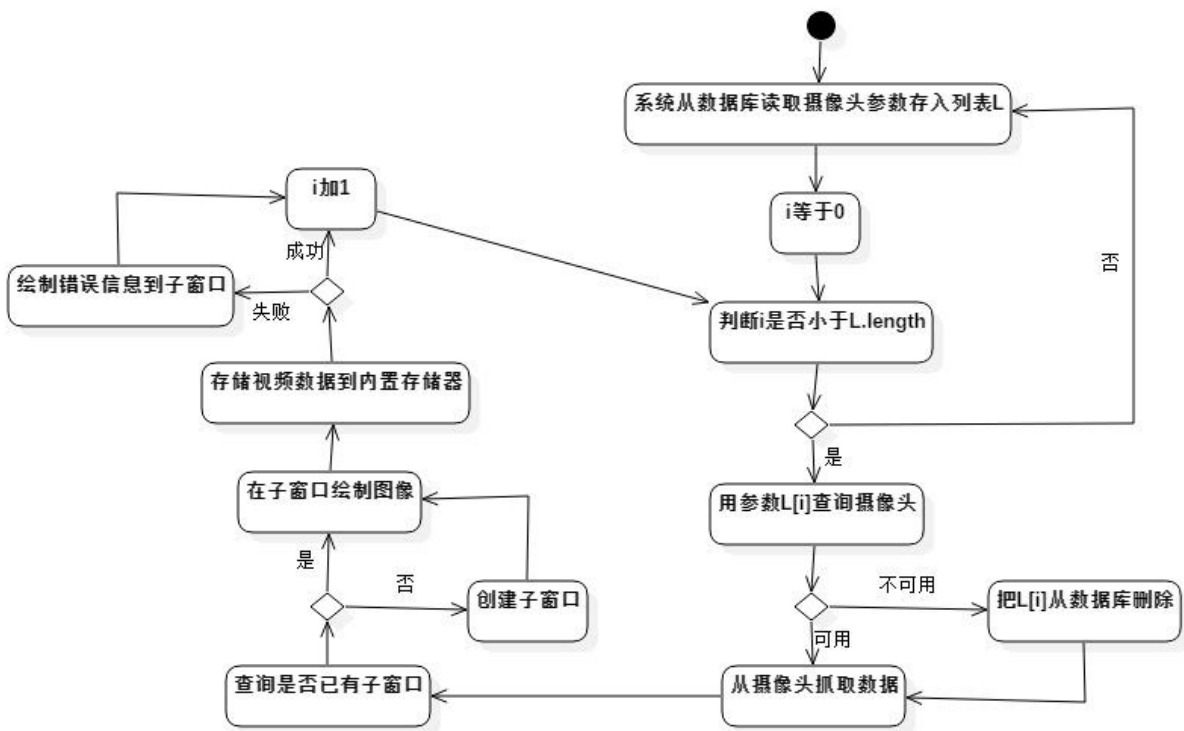


图 2：“实时监控”活动图

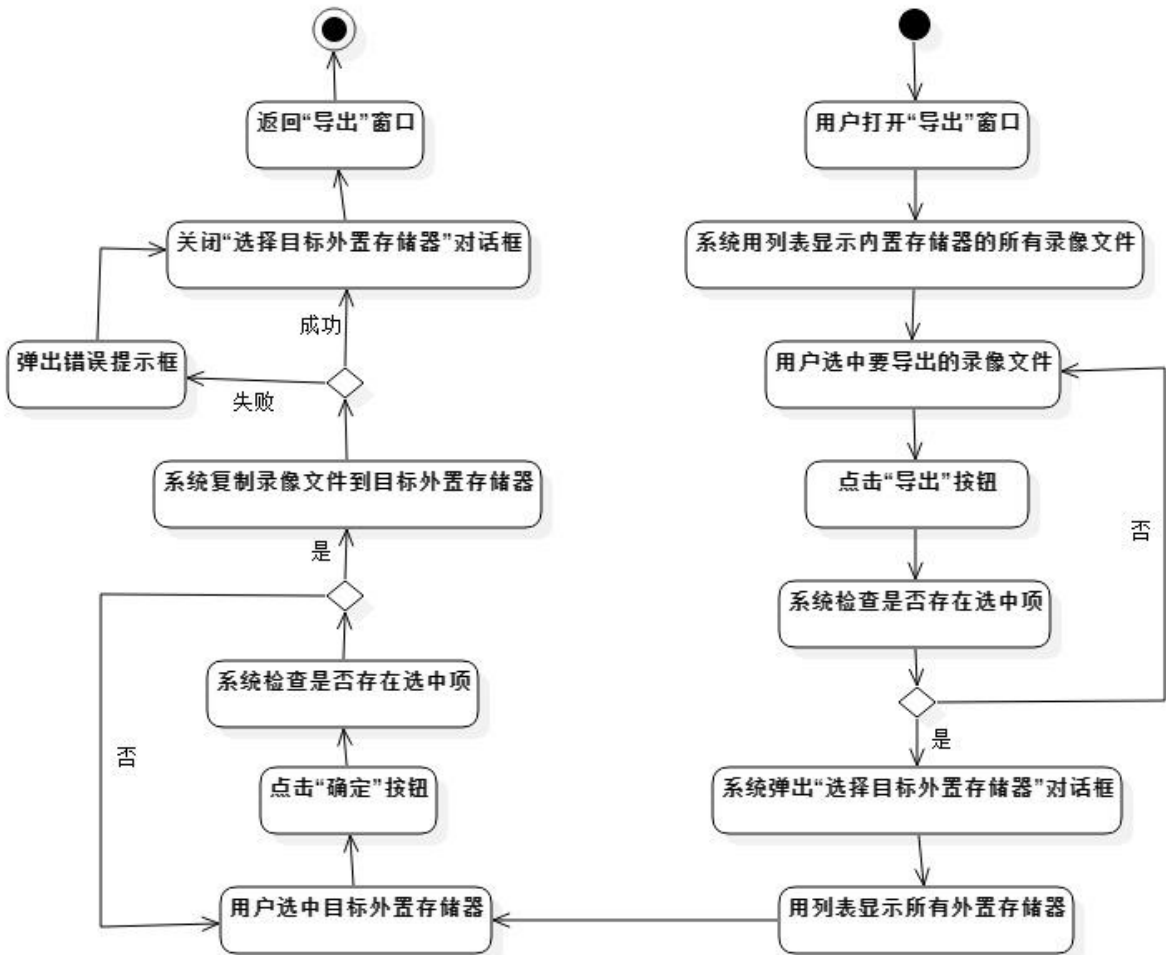


图 3：“导出录像”活动图

### 3、实验三：逻辑建模 - 类模型

备注：本项目使用 C/C++ 开发，所以类和变量的命名不像 Java 那样习惯用驼峰法。本项目的开发团队使用匈牙利命名法。

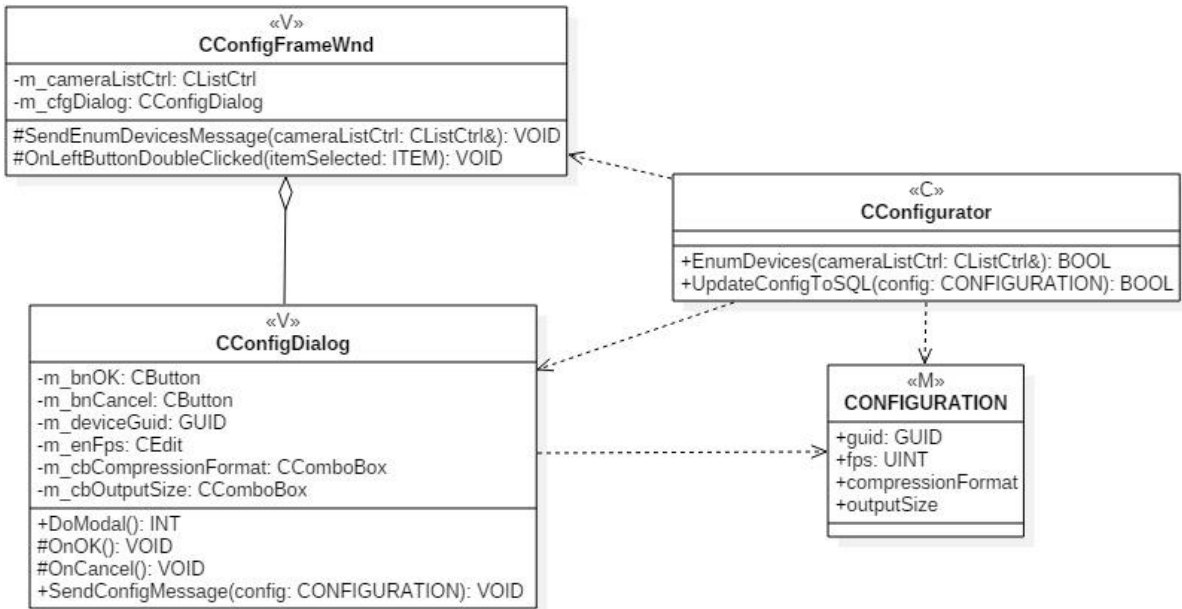


图 1：“配置摄像头”的相关类图

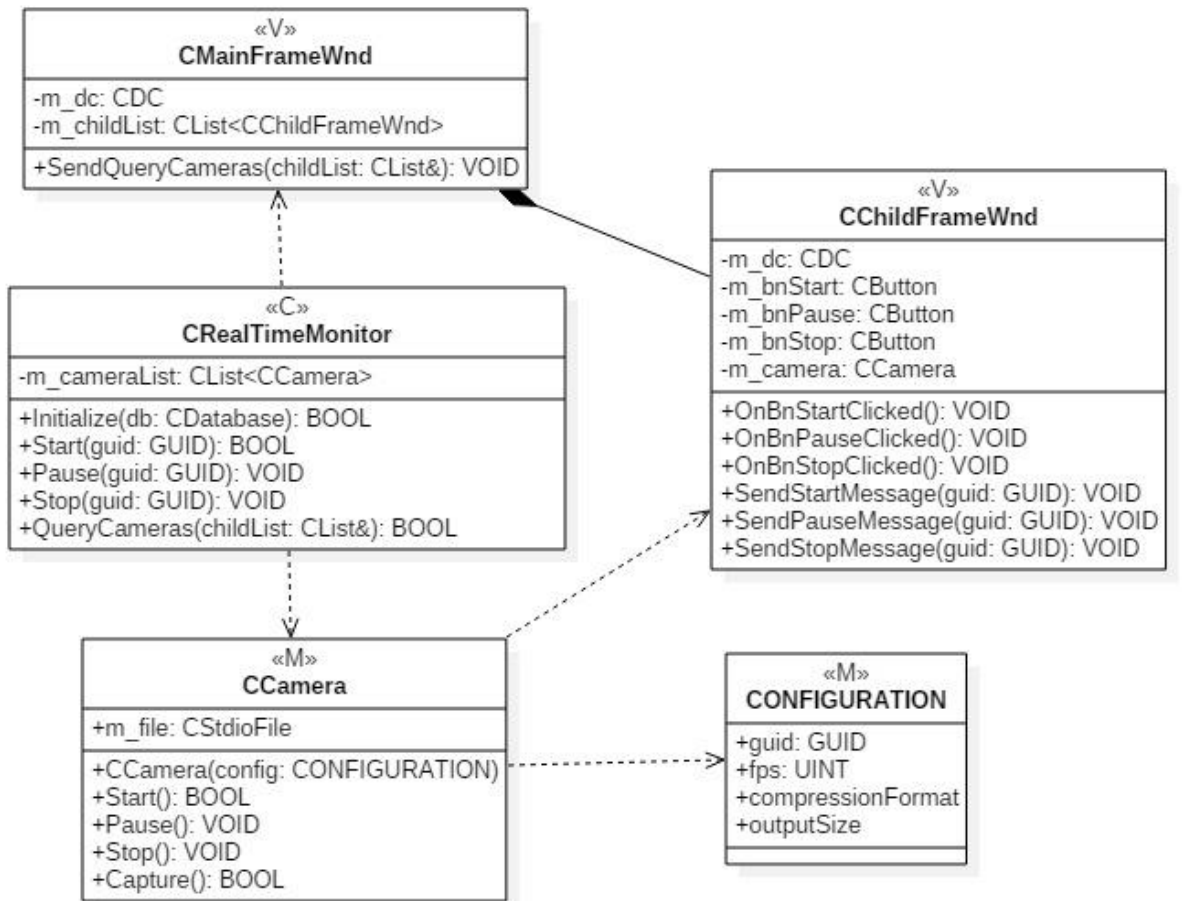


图 2：“实时监控”的相关类图

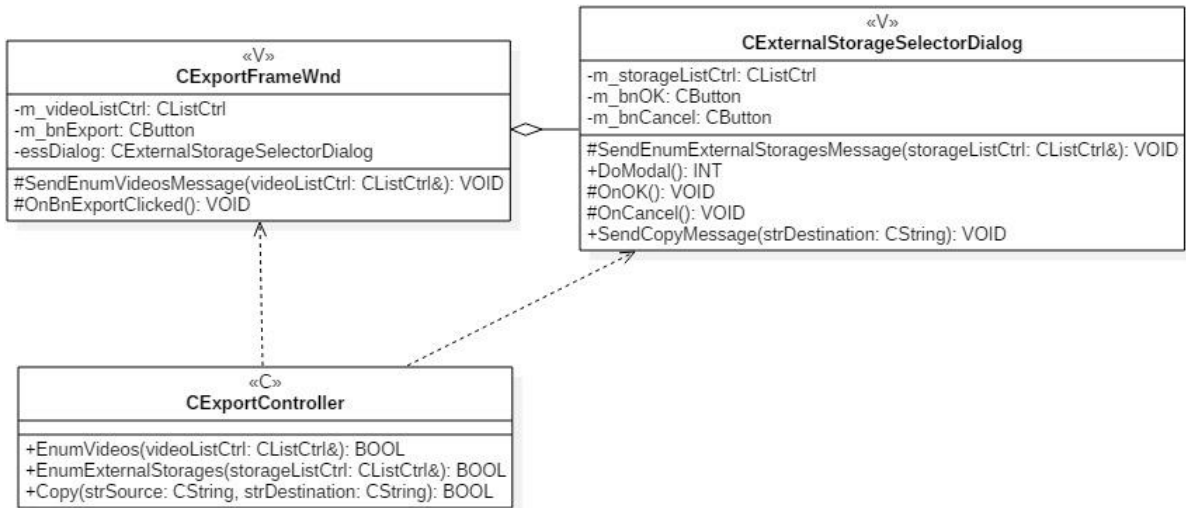


图 3：“导出录像”的相关类图

#### 4、实验四：交互建模 – 顺序模型

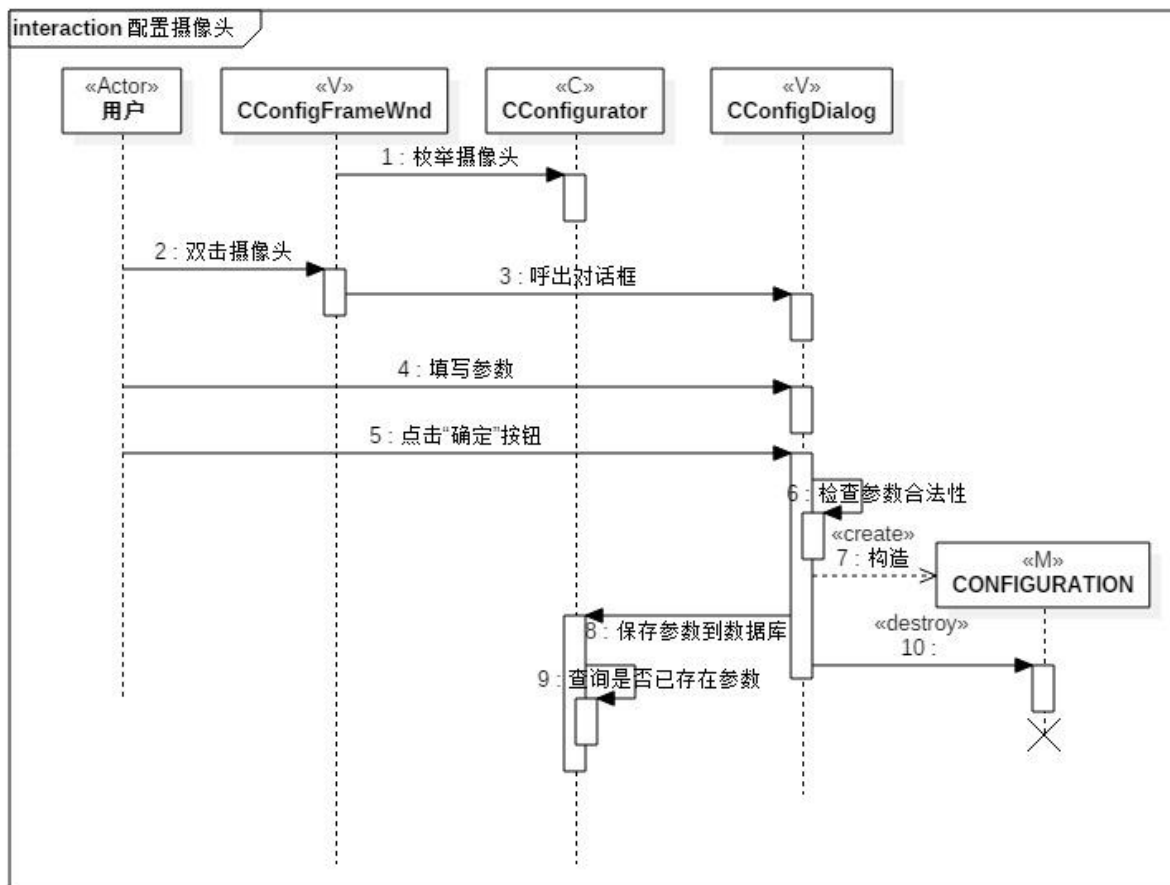


图 1：“配置摄像头”的时序图

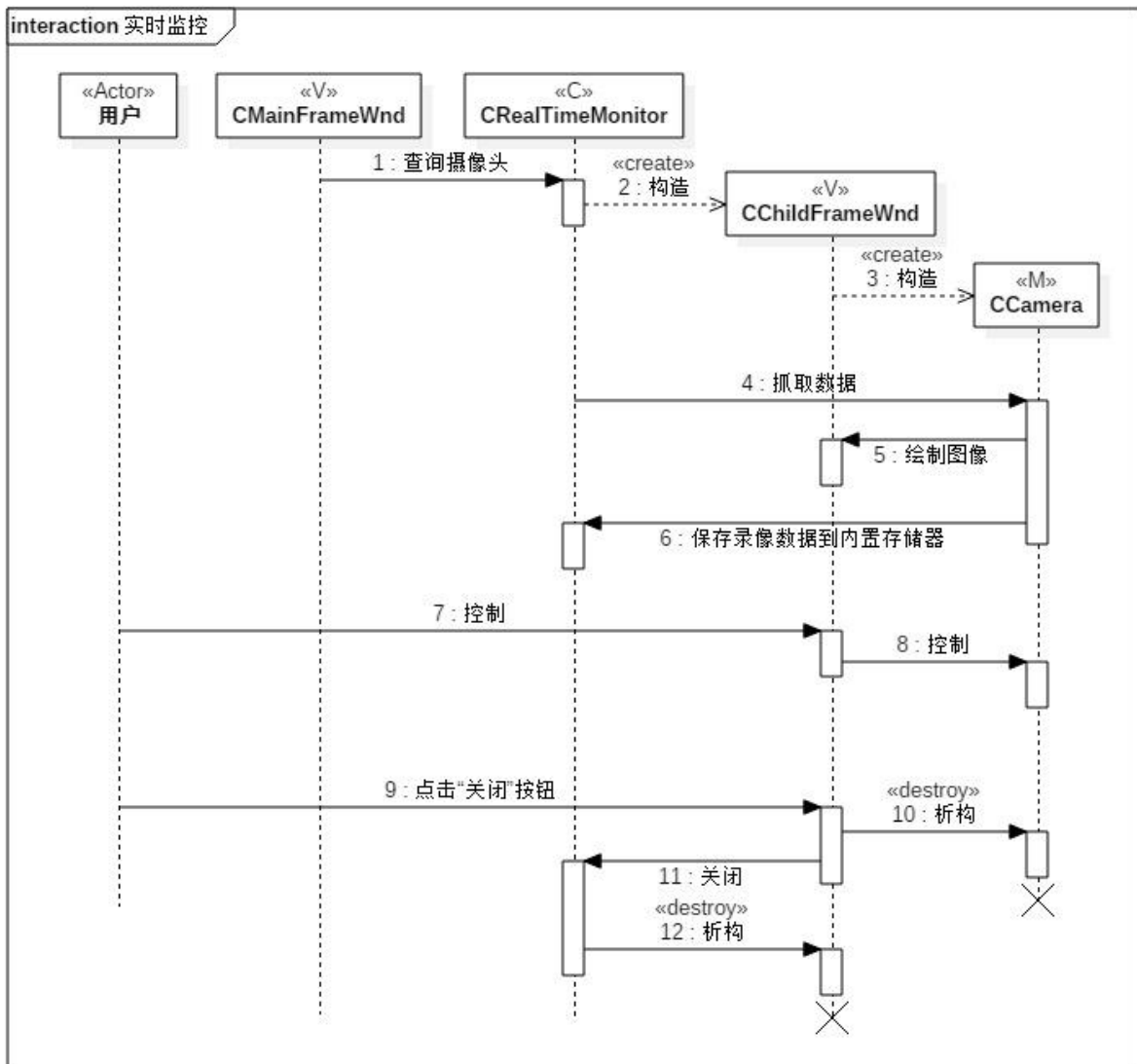


图 2：“实时监控”的时序图



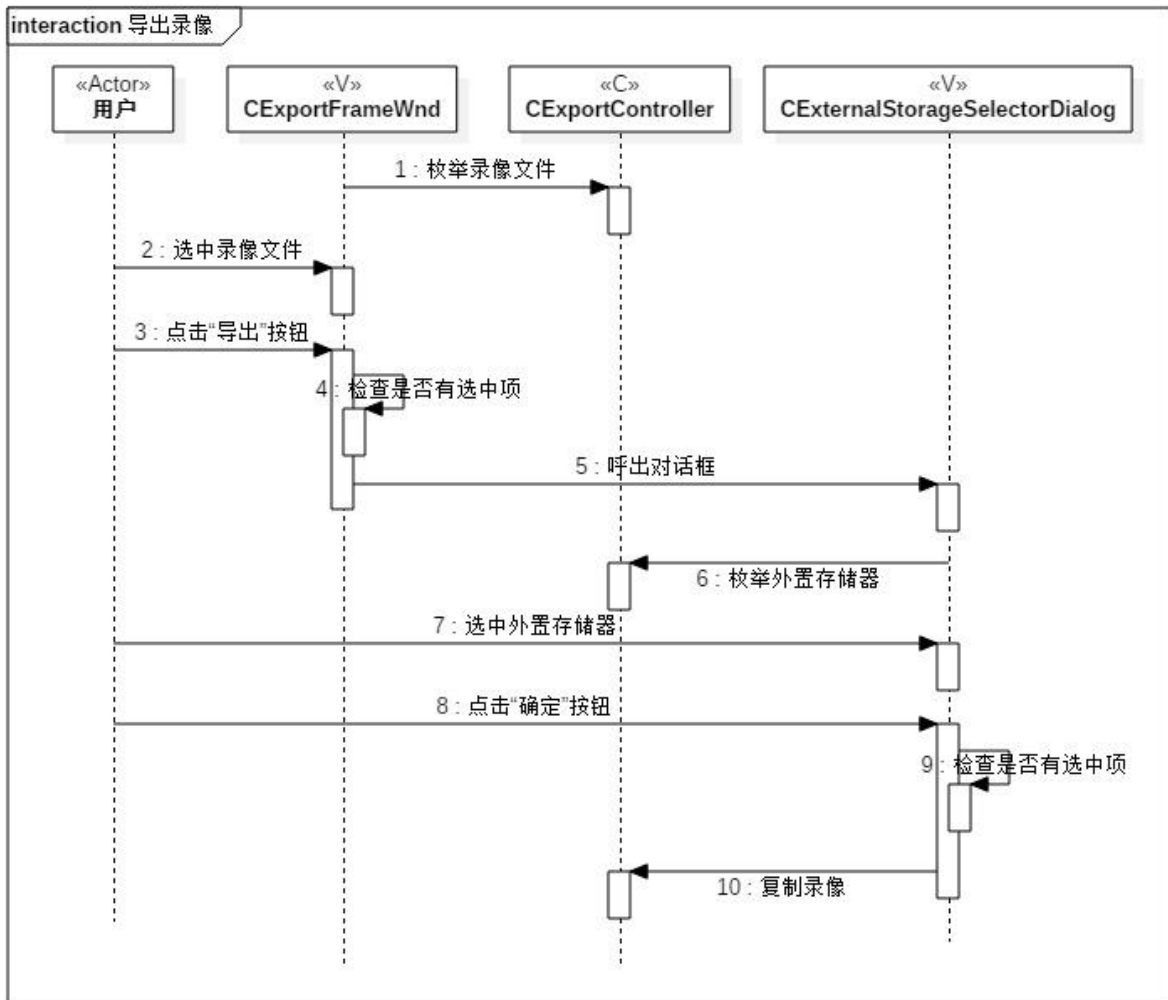


图 3：“导出录像”的时序图

## 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：

用例不能嵌套；用例图需要指明参与者；用例规约的基本流程主要写无错流程，错误写在扩展流程。

### 实验二：

用例规约的扩展流程必须在活动图有所体现，而不能只有基本流程；活动图必须有 Initial。

### 实验三：

可以使用项目所用的开发语言的命名习惯来绘制类图；类图有很多种箭头，分别代表不同的含义，

需要我们牢记和区分。

#### **实验四：**

时序图的 Lifeline 可以是类，也可以是对象；要区分同步消息和异步消息、注意消息的嵌套。

#### **实验五：**