状态机图的问题总结：

1. 转换上标有event等触发转换的transition
2. Decision node 需要标注guard condition

复习重点：

0.基于课件复习

1. 作业布置的图所有元素以及关系
2. 交互图以及其他未布置作业的图
3. 了解各类名词概念以及解释
4. 用例图和类图，活动图，包图，状态机图
5. 一带而过的图了解基本概念以及名词形式

复习总结：

UML：

unified modeling language，a standard language for writing software blueprints. The UML may be used to visualize, specify, construct, and documentthe artifacts of a software-intensive system

Visibility可见性原则：

Public+ protected# private- package~

交互图

1. 顺序图：

Sequence diagram emphasizes the time ordering of messages and shows a set of roles and the messages sent and received by the instances playing the roles

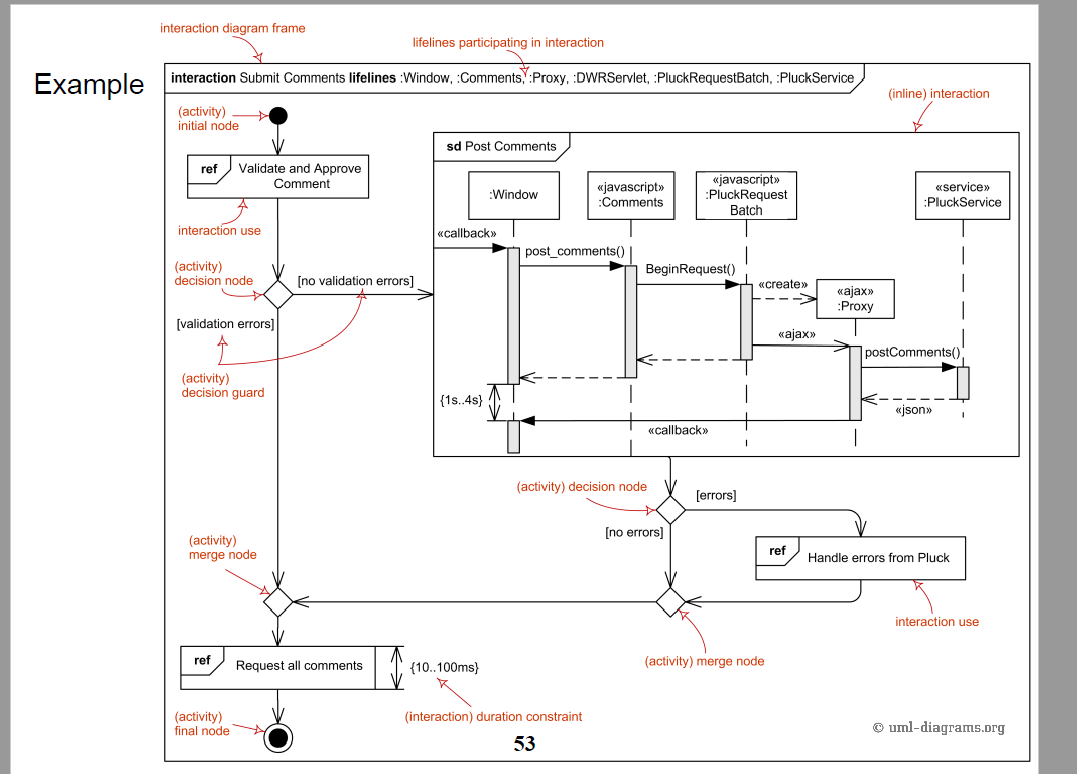
1. 通信图：

Communication diagram emphasizes the structural organization of the objects that send and receive messages

1. 定时图：

Timing diagrams focus on conditions changing within and among Lifelines along a linear time axis

1. 交互概述图：



类图：

**关系**

泛化关系：类似于继承

关联关系：has-a关系 public数据成员 有单向关联和双向关联之分

聚合关系（aggreation）：如果*A*由*B*聚合成，表现为*A*包含有*B*的全局对象，但是*B*对象可以不在*A*创建的时刻创建。

组合关系（*Composition）*是一种强的“拥有”关系，体现了严格的部分和整体的关系，部分和整体的生命周期一致。如果*A*由*B*组成，表现为*A*包含有*B*的全局对象，并且*B*对象在*A*创建的时刻创建。

依赖关系是类与类之间的连接，表示一个类依赖于另一个类的定义，其中一个类的变化将影响另外一个类。例如如果*A*依赖于*B*，则*B*体现为局部变量，方法的参数、或静态方法的调用

实现关系：通常出现在类与接口之间，箭头指向接口类

**类的三个阶段**

Conceptual(概念) 不考虑实现功能的语言，只显示其概念结构 

Specification(规范) 显示每个类的接口，明确每个类在公共阶层上体现的责任以及功能 

Implementation（实现）显示每个类的功能是如何在代码层面具体实现的 

**面向对象类图的设计原则：**

1. 开放封闭原则（OCP：Open closed principle）最重要的原则

开放读取，关闭修改

1. 可替代性原则（LSP : Liskov Substitution Principle）：派生类可以被基类替代
2. 依赖反演原则（DIP：）：依赖于接口或抽象函数和类的策略，而不是依赖于具体的函数和类
3. 接口分离原则（ISP:Interface Segregation principle）：多个客户端接口优于一个通用接口

包图

Package的元素

Classes

Interfaces

Components

Nodes

Collaborations

Use cases

Diagrams

and even other packages