第05讲-01　UML顺序图

SUST

2020.04

【主要内容】

- ·定义顺序图
- ·为什么要建立顺序图
- ·了解顺序图的标记符组件
- ·理解如何使用消息进行通信
- ·学习顺序图使用的其他技术
- ·学习如何建模顺序图

UML交互图

在标识出系统的类图之后，仅给出了实现用例的组成结构，这时还需要描述这些类的对象是如何交互来实现用例功能的。即不但需要把用例图模型转化为类图模型，还要将它转化为交互图模型。

交互图表示类（对象）如何交互来实现系统行为。交互图具有如下两种形式。

1） 顺序图

它描述对象按时间顺序的消息交换过程，它体现出系统用例的行为。

2） 协作图

它描述对象间的组织协作关系，它也可体现出系统用例的行为。

顺序图和协作图都可以表示对象间的交互关系，但它们的侧重点不同。顺序图用消息的几何排列关系来表达对象间交互消息的先后时间顺序。而协作图则建模对象（或角色）间的通信关系。

使用顺序图建模

一、定义顺序图

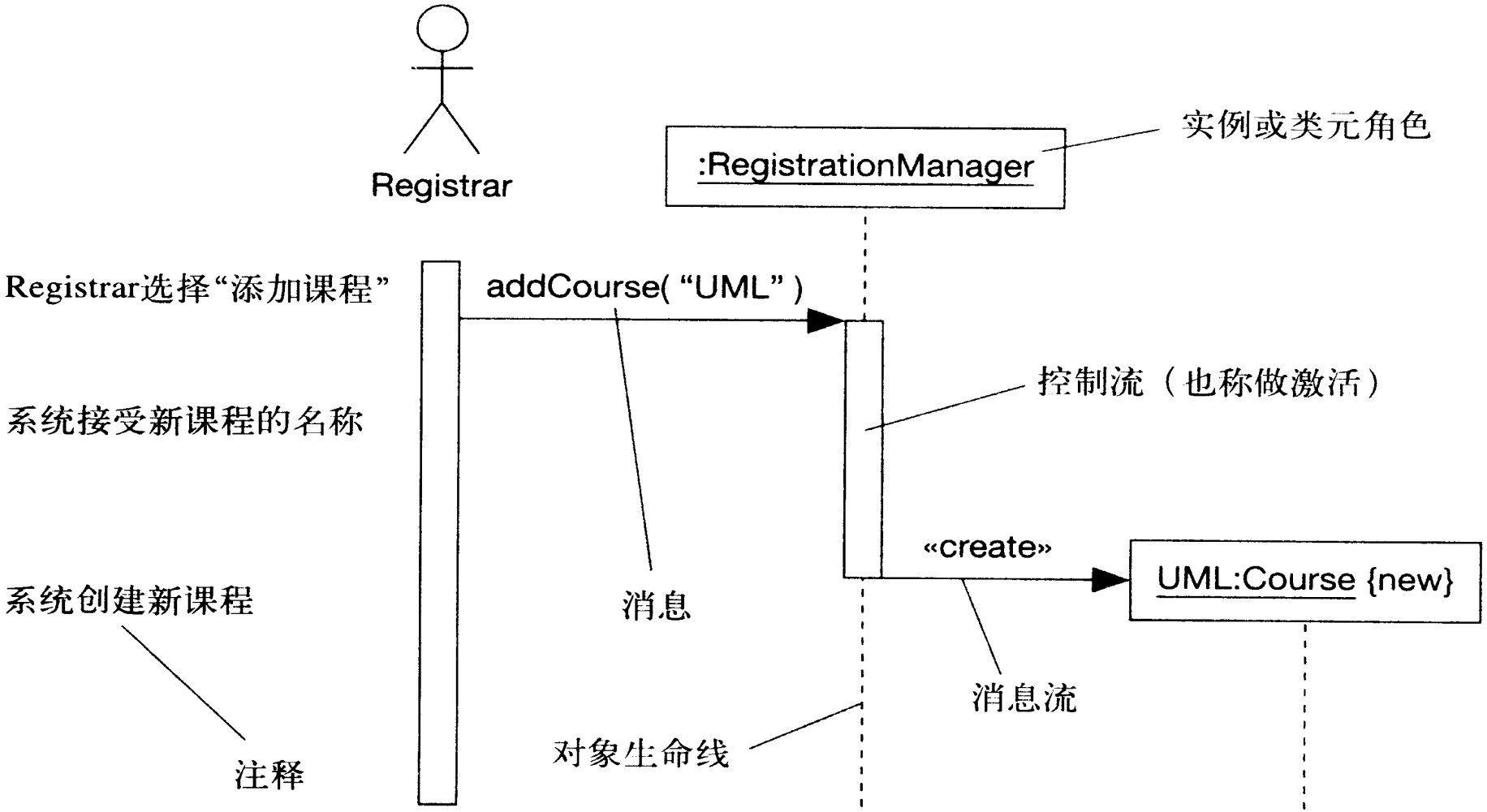
顺序图是两种类型的交互图之一。顺序图用来建模以时间顺序安排的对象交互，并且把用例行为分配给类。它是用来显示参与者如何采用若干顺序步骤与系统对象交互的模型。

二、为什么要建模顺序图

建模顺序图有许多理由，顺序图与活动图具有类似的作用。其中重要的理由就是实现用例。任何用例都可以使用顺序图进一步阐明和实现。

三、顺序图的标记符

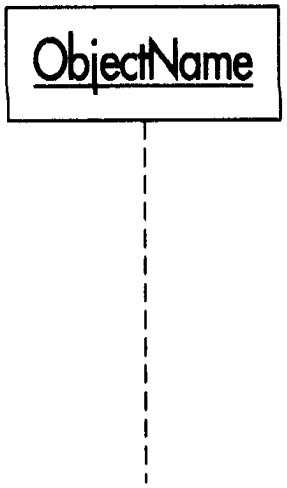
顺序图有两个主要的标记符：活动对象和这些活动对象之间的通信消息。活动对象可以是任何在系统中扮演角色的对象，不管它是对象实例还是参与者，如下图所示。



活动对象之间发送的消息是顺序图的关键。消息说明了对象之间的控制流，对象是如何交互的，以及什么条件会改变控制流。

1．活动对象

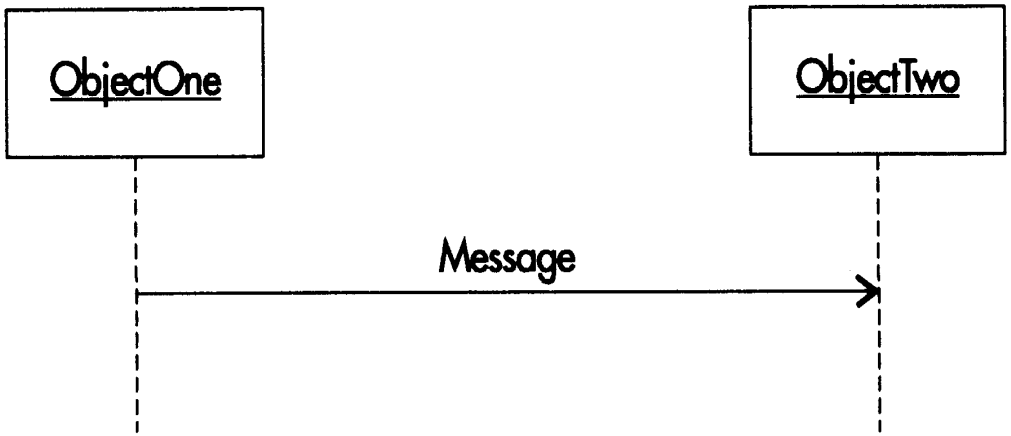
活动对象可以是系统的参与者或者任何有效的系统对象。对象是类的实例，它使用包围名称的矩形框来标记。名称带下划线，顺序图中对象的标记符如下图所示。



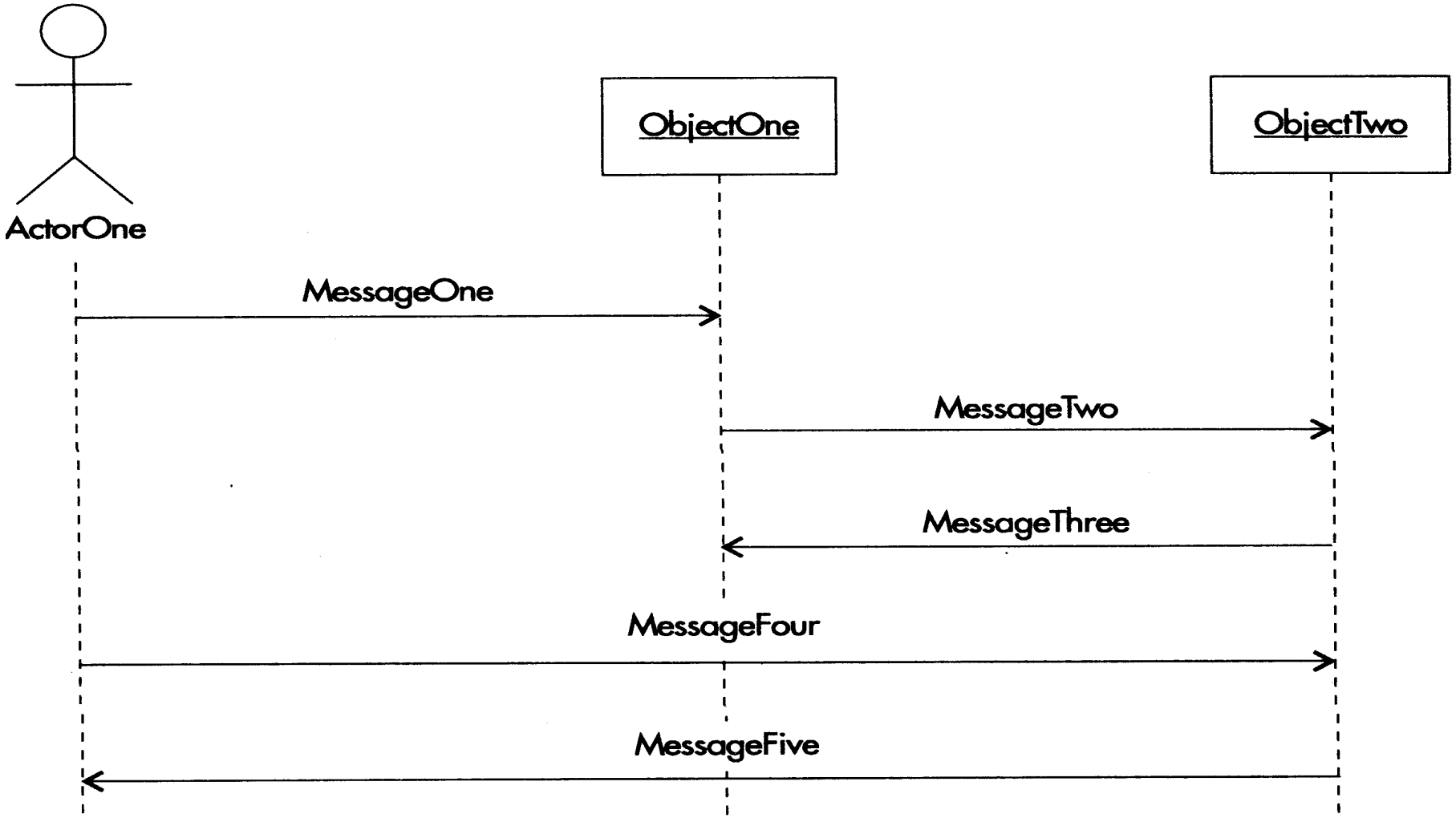
2．消息

消息用来说明顺序图中不同活动对象之间的通信。它可在一个对象需要取消不同对象的进程时或者需要向另一个对象提供服务时，使用消息。

消息从活动对象生命线到接收对象生命线的箭头表示。箭头上面标记要发送的消息，如下图所示。

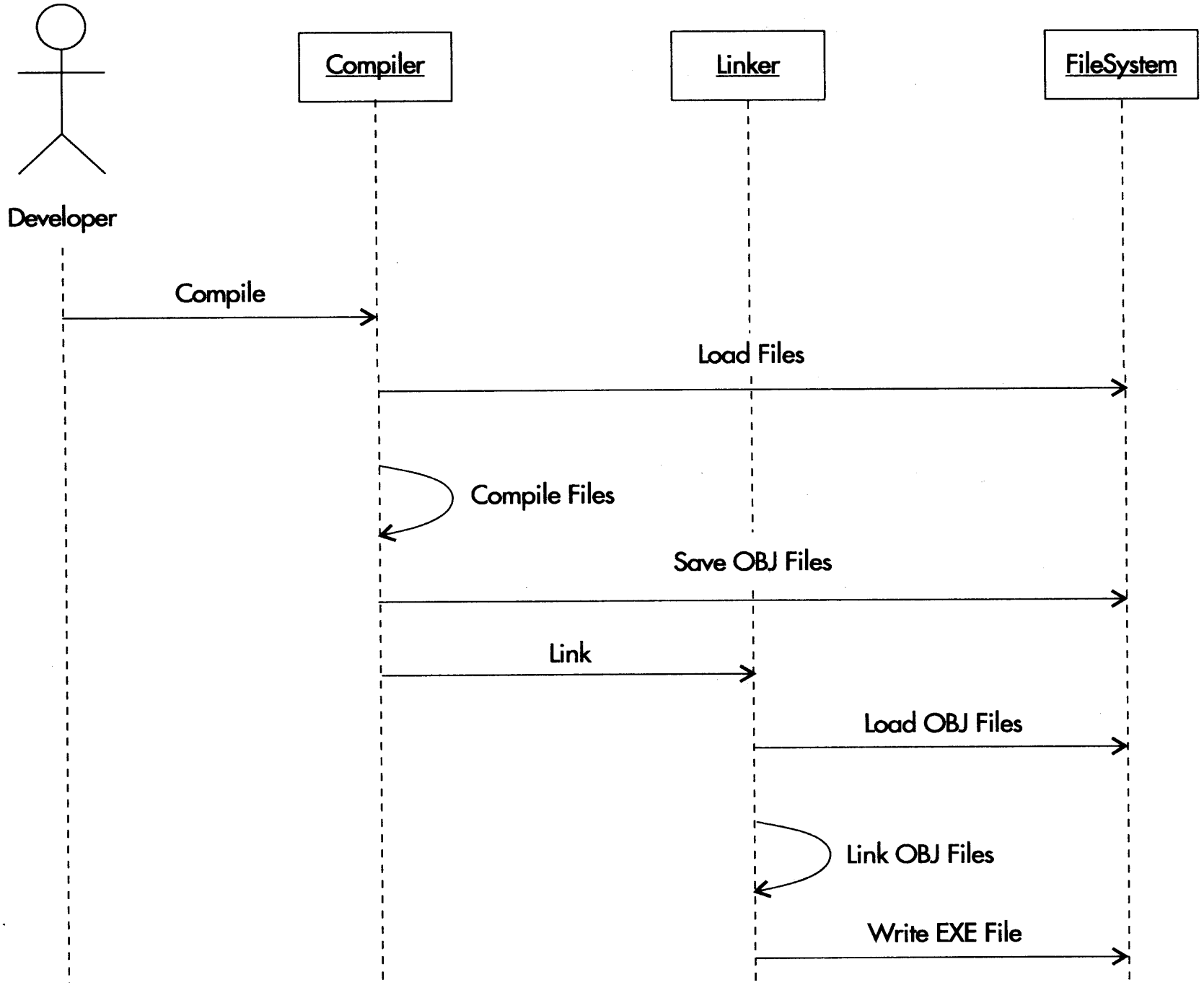


把参与者表示为活动对象的建模可以说明参与者如何与系统交互，以及系统如何与用户交互。参与者可以调用对象，对象也可以通知参与者，如下图所示。



上面的图例说明了参与者和对象可以把消息发送给顺序图中的任何参与者或者对象。它们可以把消息发送给不是其直接相邻的参与者或者对象。

下面看一个意义更加丰富的示例。对于Compile Application用例，我们可以创建一个成功编译工作流的顺序图，如下图所示。


Developer
Compiler
Linker
FileSystem
Compile
Load Files
Compile Files
Save OBJ Files
Link
Load OBJ Files
Link OBJ Files
Write EXE File

这个顺序图中有4个活动对象：Developer、Compiler、Linker和FileSystem。Developer是系统的参与者。Compiler是Developer交互的应用程序。Linker是一个用来链接对象文件的独立进程。FileSystem是系统层功能的包装器，用来执行文件的输入和输出例程。

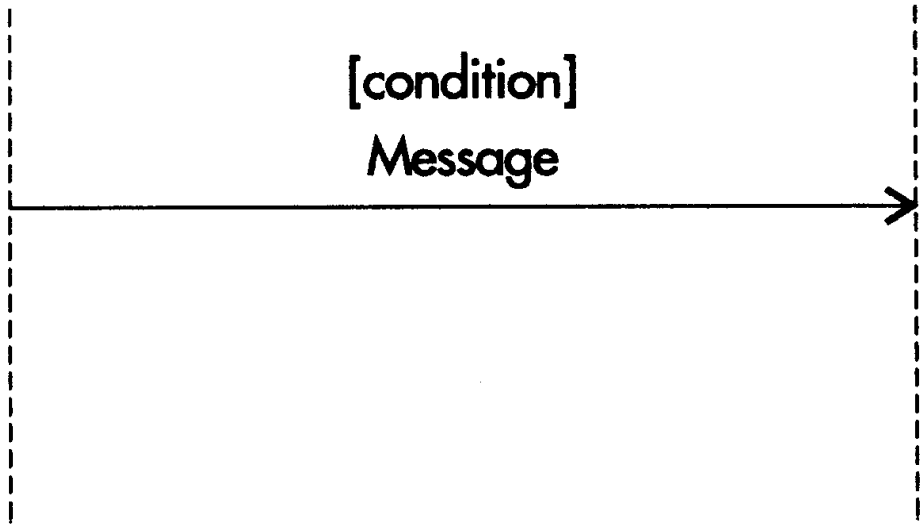
Compile Application用例的顺序图操作：

Developer请求Compiler执行编译
Compiler请求FileSystem 加载文件
Compiler通知自己执行编译
Compiler请求FileSystem 保存对象代码
Compiler请求Linker链接对象代码
Linker请求 FileSystem加载对象代码
Liker通知自己执行链接
Linker请求FileSystem保存编译的结果

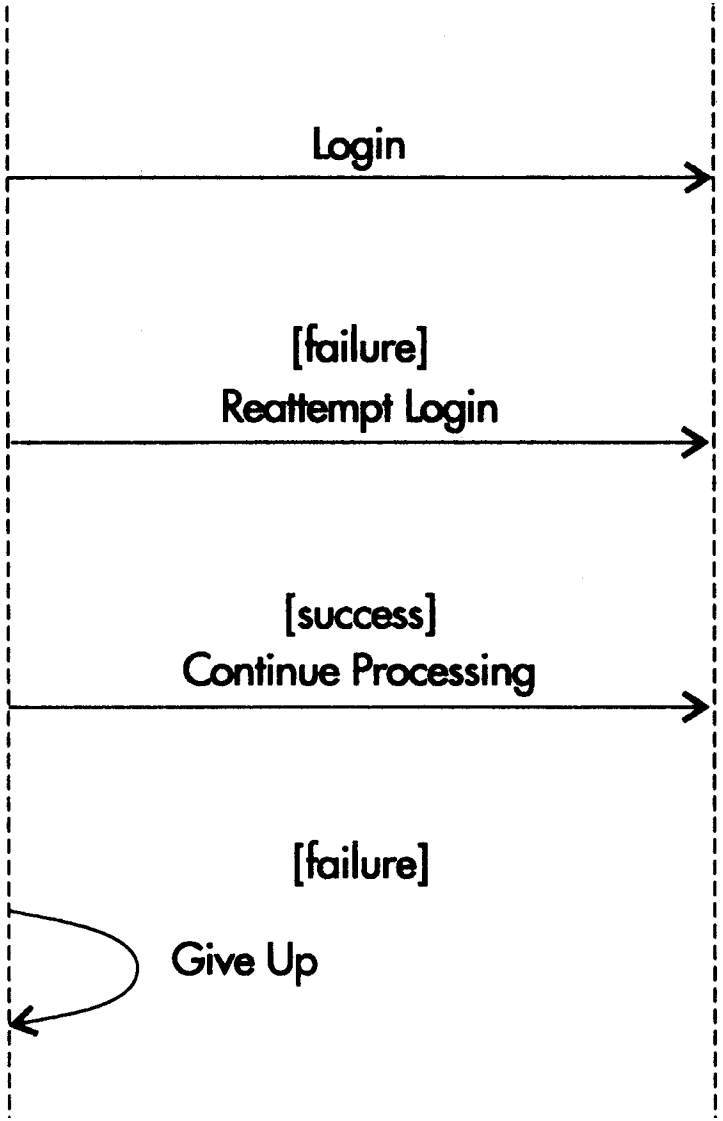
四、如何使用消息进行通信

消息是顺序图活动对象之间通信的惟一方式。UML中的消息使用了一些简洁的标记符。

消息可以包含条件以便限制它们只在满足条件时才能发送。条件显示在消息名称上面的方括号中，如下图所示。



下面示例演示了如何建模一个顺序图来显示登录尝试。如果登录失败，会在放弃登录之前重试一次，如下图所示。

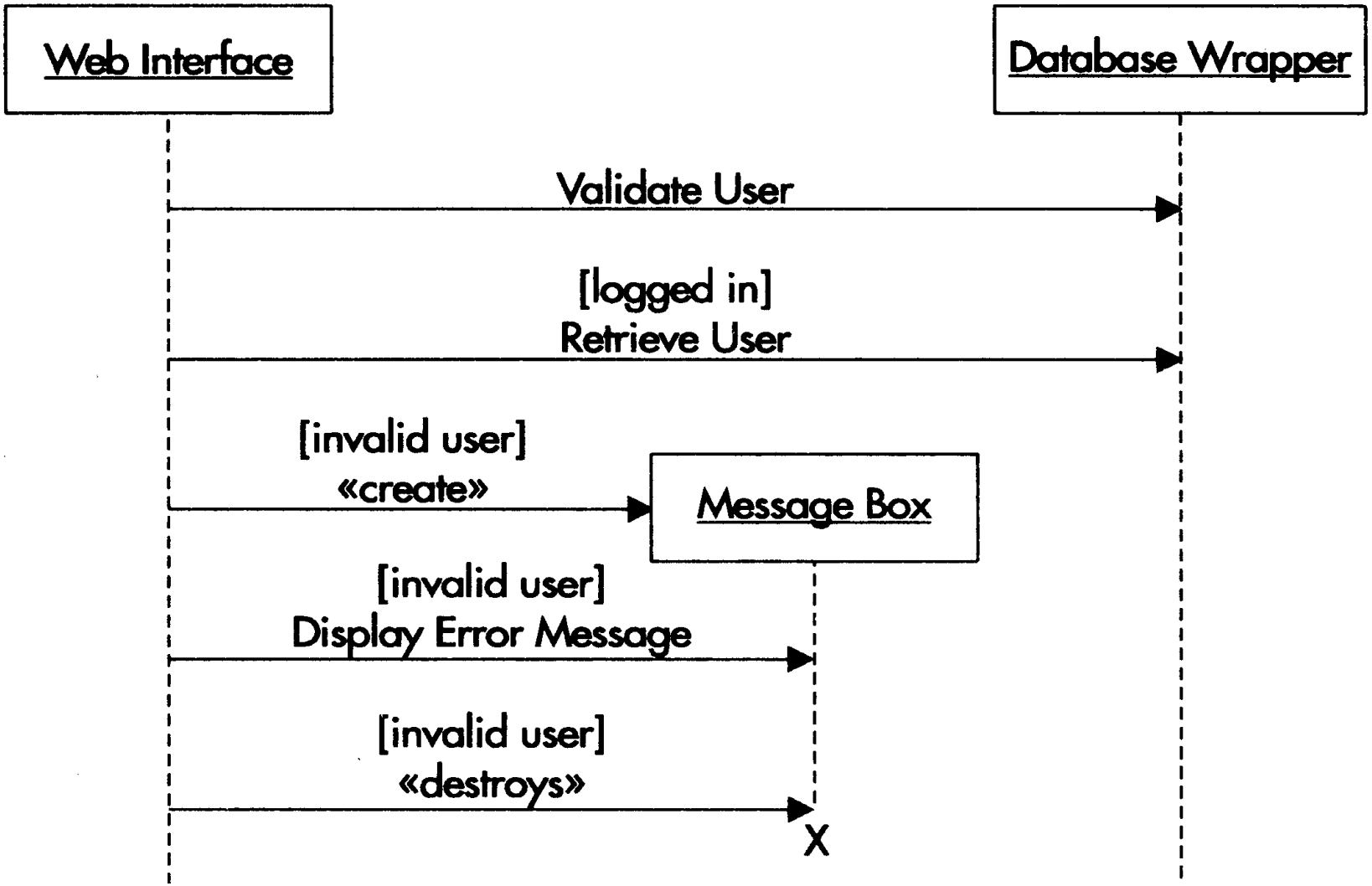


五、 顺序图的其他技术

学习如何在创建顺序图的过程中创建对象。与活动图一样，可以在顺序图中设置拥有控制权的对象状态。另外一点和活动图相似的是，可以通过使用分支和从属控制流来以多种方式修改顺序图的控制流。

1．创建对象

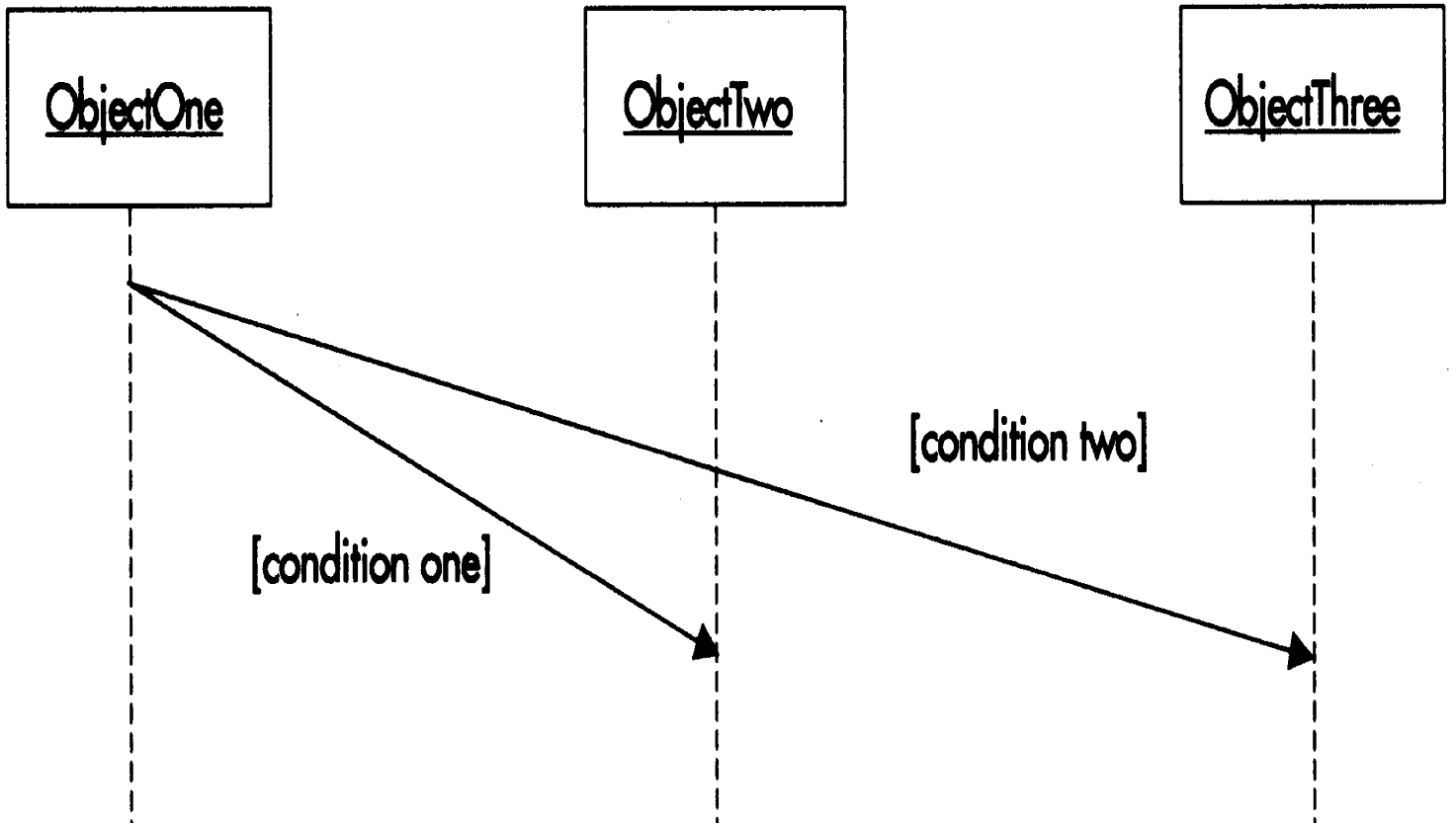
创建对象的标记符如下图中的示例所示。有一个主要步骤用来把“create”消息发送给对象实例。对象创建之后就会具有生命线，可以使用该对象发送和接收消息。在处理新创建的对象，或者处理顺序图中的任何其他对象时，都可以发送“destroys”消息来删除对象。若要想说明某个对象被销毁，需要在被销毁对象的生命线上放一个X字符。


Web Interface
Database Wrapper
Validate User
[logged in]
Retrieve User
[invalid user]
«create»
Message Box
[invalid user]
Display Error Message
[invalid user]
«destroys»
X

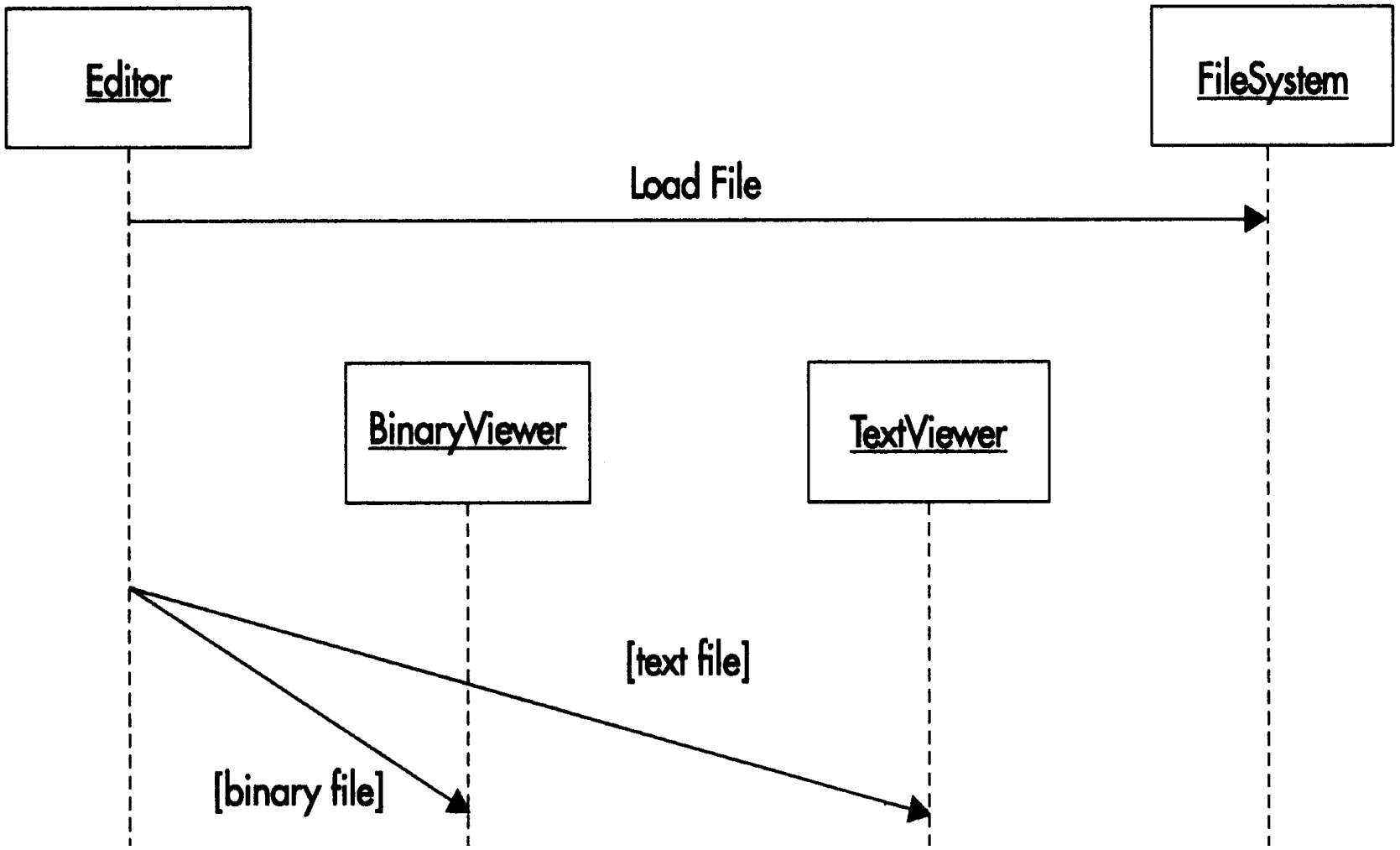
2．分支和从属流

有两种方式来修改顺序图的控制流：使用分支和使用从属流。这两种方式很相似，各自的标记符略微不同。控制流的改变是由于不同的条件导致控制流走向不同的道路。

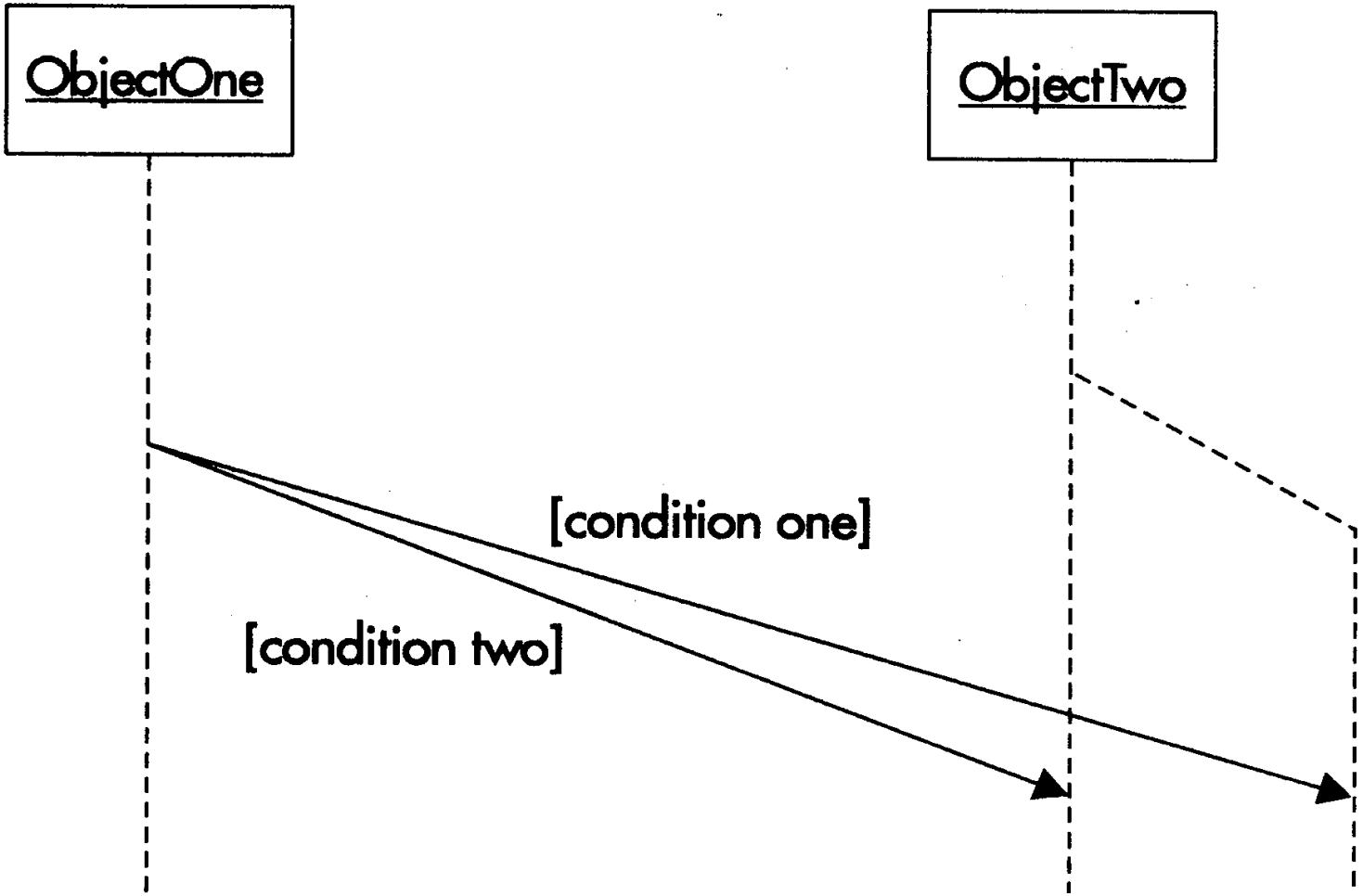
分支允许控制流走向不同的对象，如下图所示。



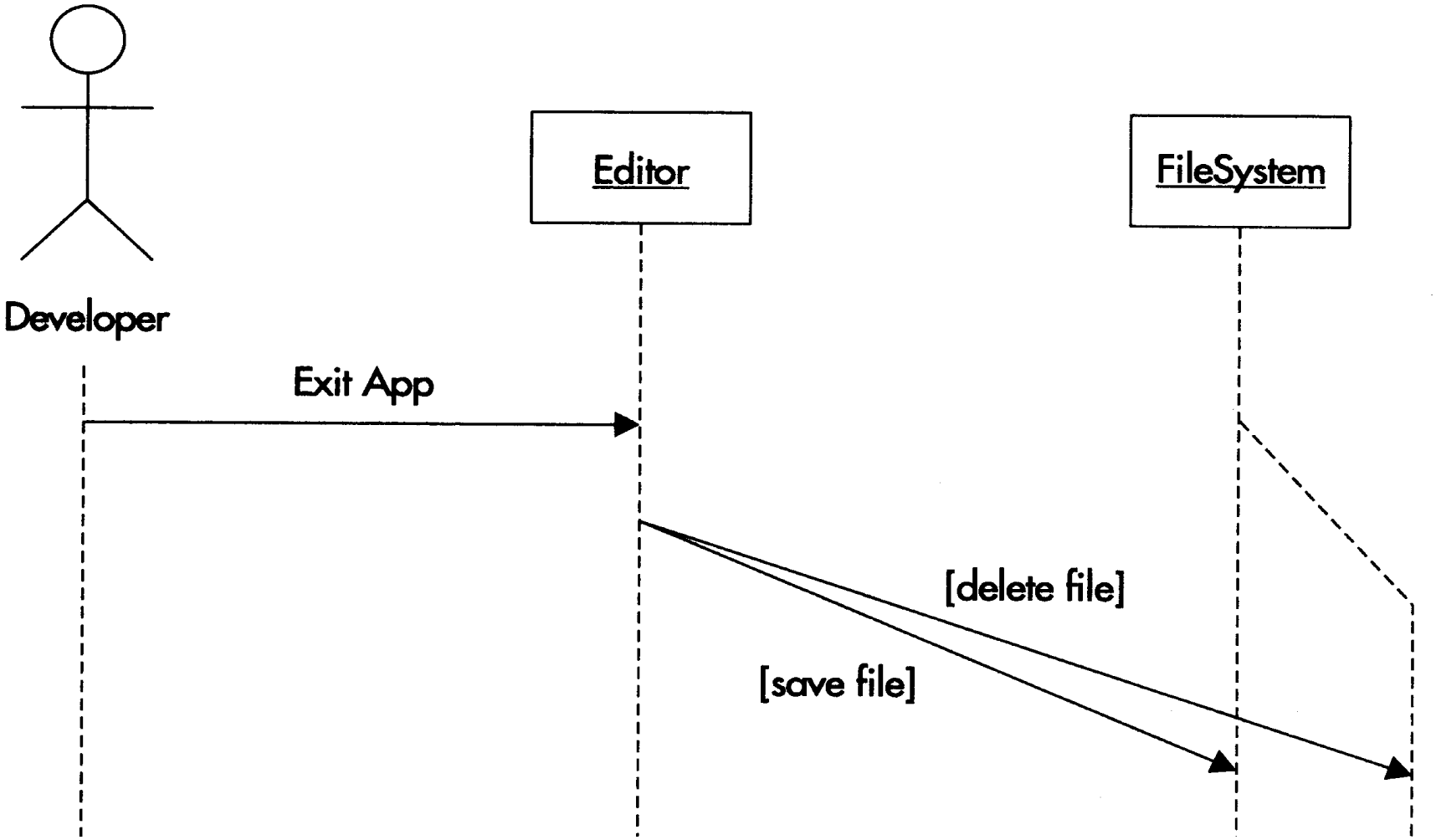
注意消息的开始位置是相同的，分支消息的结束“高度”也是相等的。这说明在下一步中，其中之一将会执行，如下图所示。



从属流还允许控制流根据条件改变，但是只允许控制流改变为相同对象的另一条生命线分支，如下图所示。



在下面的示例中，Editor在用户删除文件或者保存文件时向Filesystem发送一条消息。显然，Filesystem将会执行两种完全不同的活动，并且每一个工作流都需要独立的生命线，如下图 所示。



练习：阅读一个顺序图

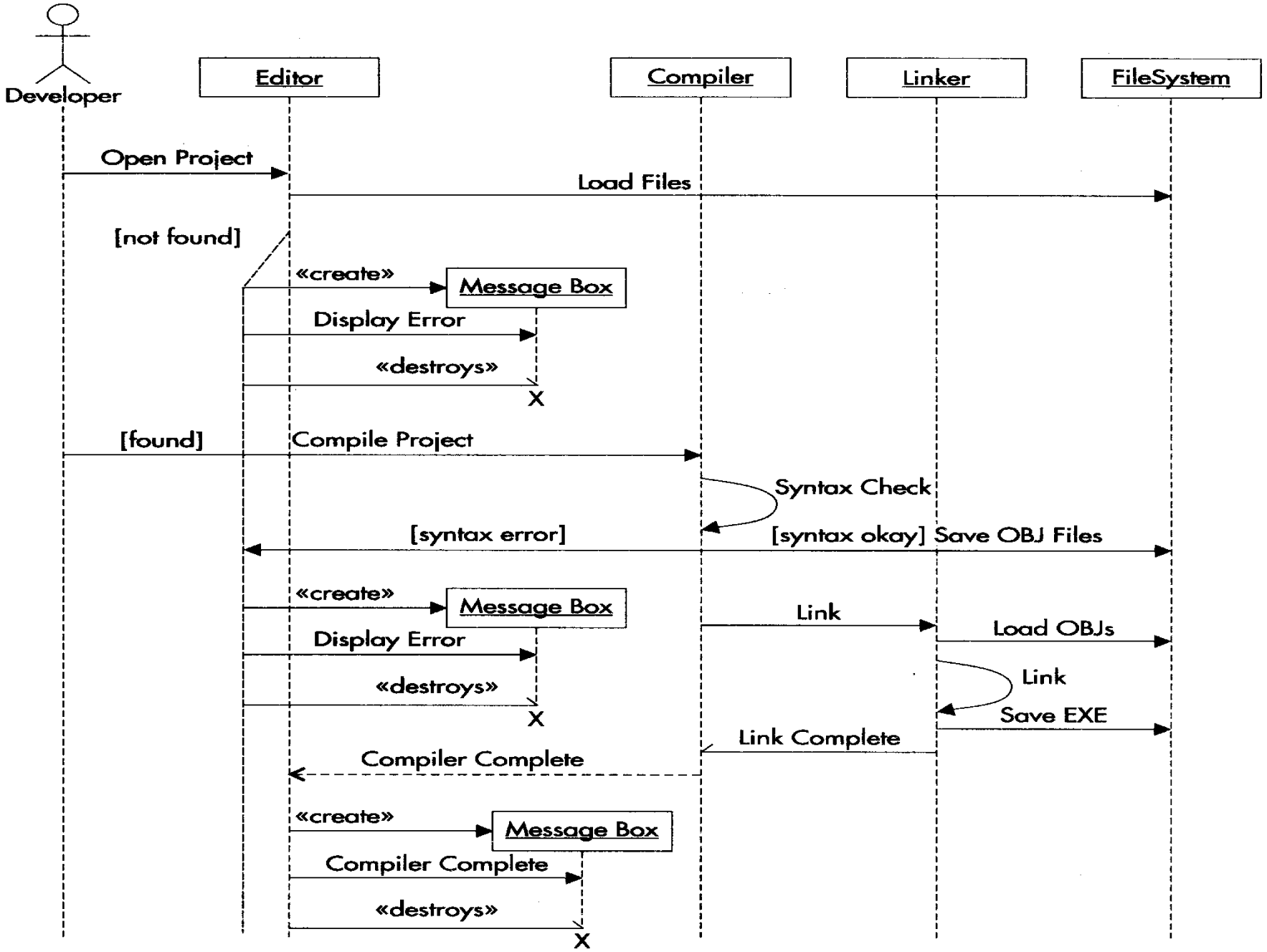
阅读下图所示的顺序图，该图说明了开发者编译应用程序的步骤。在阅读顺序图时，请指出我们到目前为止已经学习过的标记符组件。

练习步骤

1）指出顺序图中的参与者和对象。

2）指出出错处理发生的位置。

3）按照控制流的顺序指出各个消息。


Developer
Editor
Compiler
Linker
FileSystem
Open Project
Load Files
[not found]
«create»
Message Box
Display Error
«destroys»
X
[found]
Compile Project
Syntax Check
[syntax error]
[syntax okay]
Save OBJ Files
«create»
Message Box
Link
Load OBJs
Display Error
«destroys»
X
Link
Save EXE
Link Complete
Compiler Complete
«create»
Message Box
Compiler Complete
«destroys»
X

六、学习如何建模顺序图

创建顺序图包含4项任务：

1）确定需要建模的工作流。

2）从左到右布置对象。

3）添加消息和条件以便创建每一个工作流。

4）绘制总图以便连接各个分图。

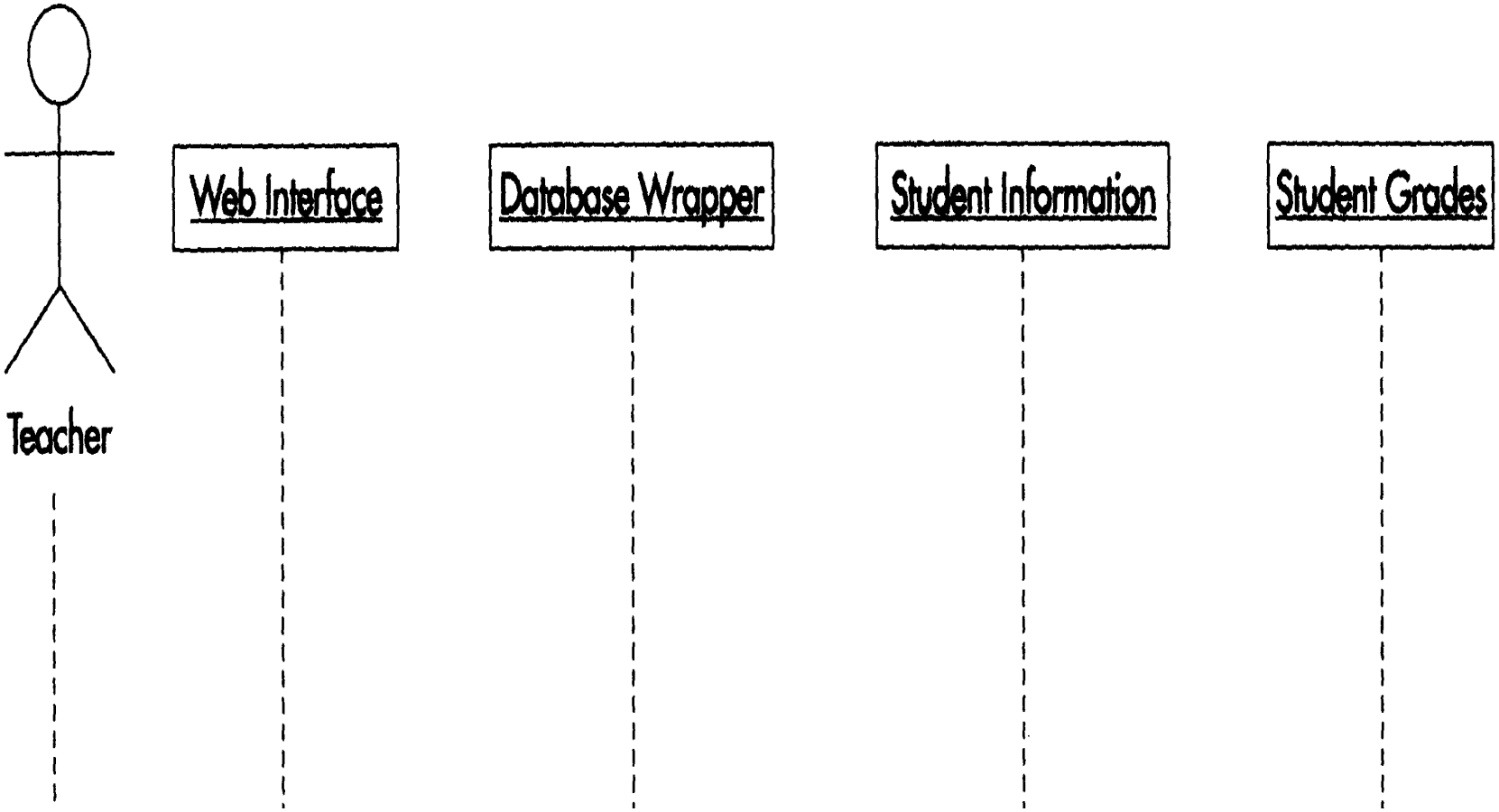
1．确定工作流

建模顺序图的第一步是确定将要建模的工作流。对于这个练习，我们将要建模Grading system的View Grades用例。为此，需要至少标识出3个要建模的工作流：

- 教师成功地检查学生分数
- 教师试图检查某个学生分数，但是该学生在系统中不存在。
- 教师试图检查某个学生分数，但是该学生分数在系统中不存在。

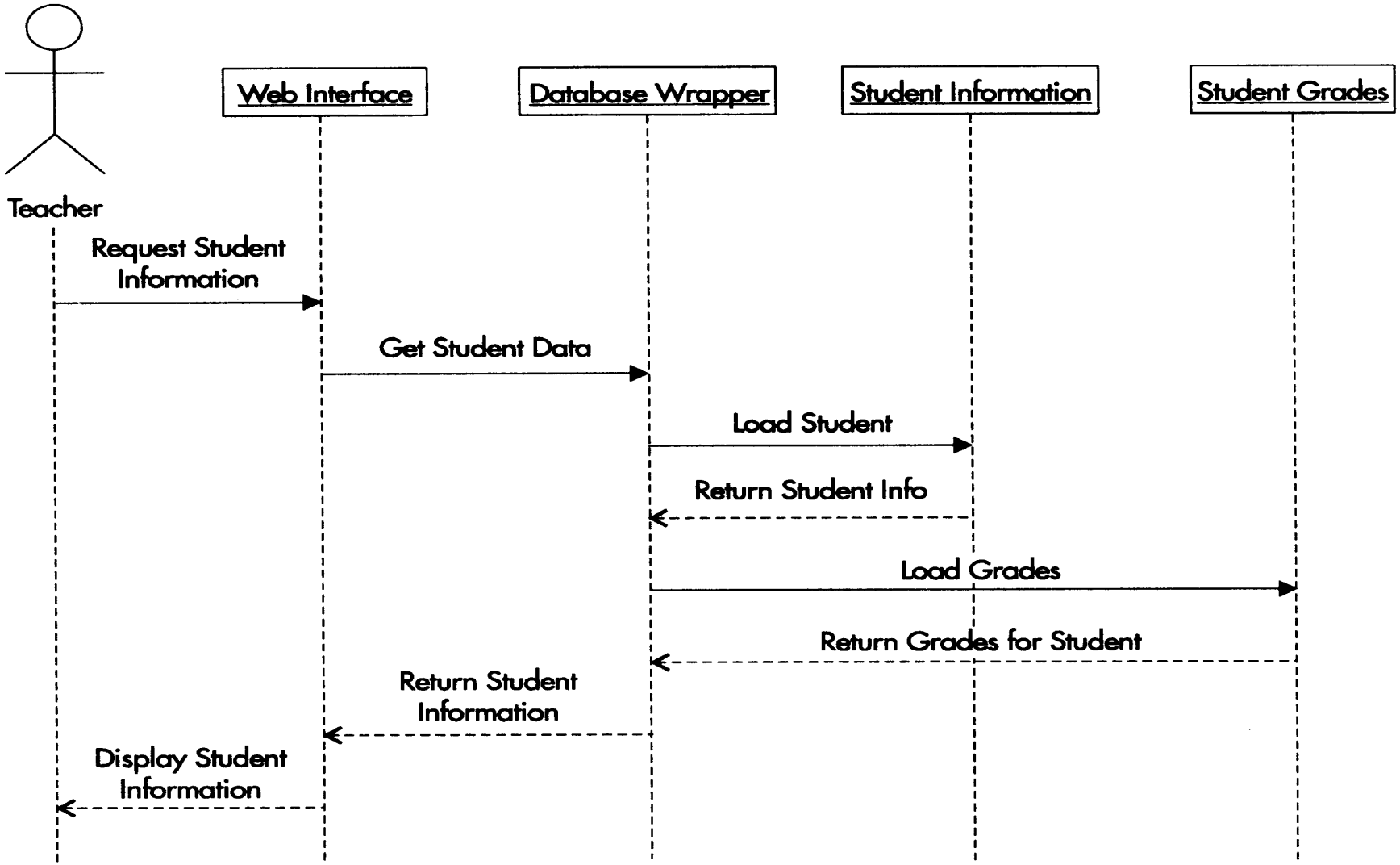
2．布置对象

建模顺序图的下一步是从左到右布置所有的参与者和对象，包含要添加消息的对象生命线，如下图所示。

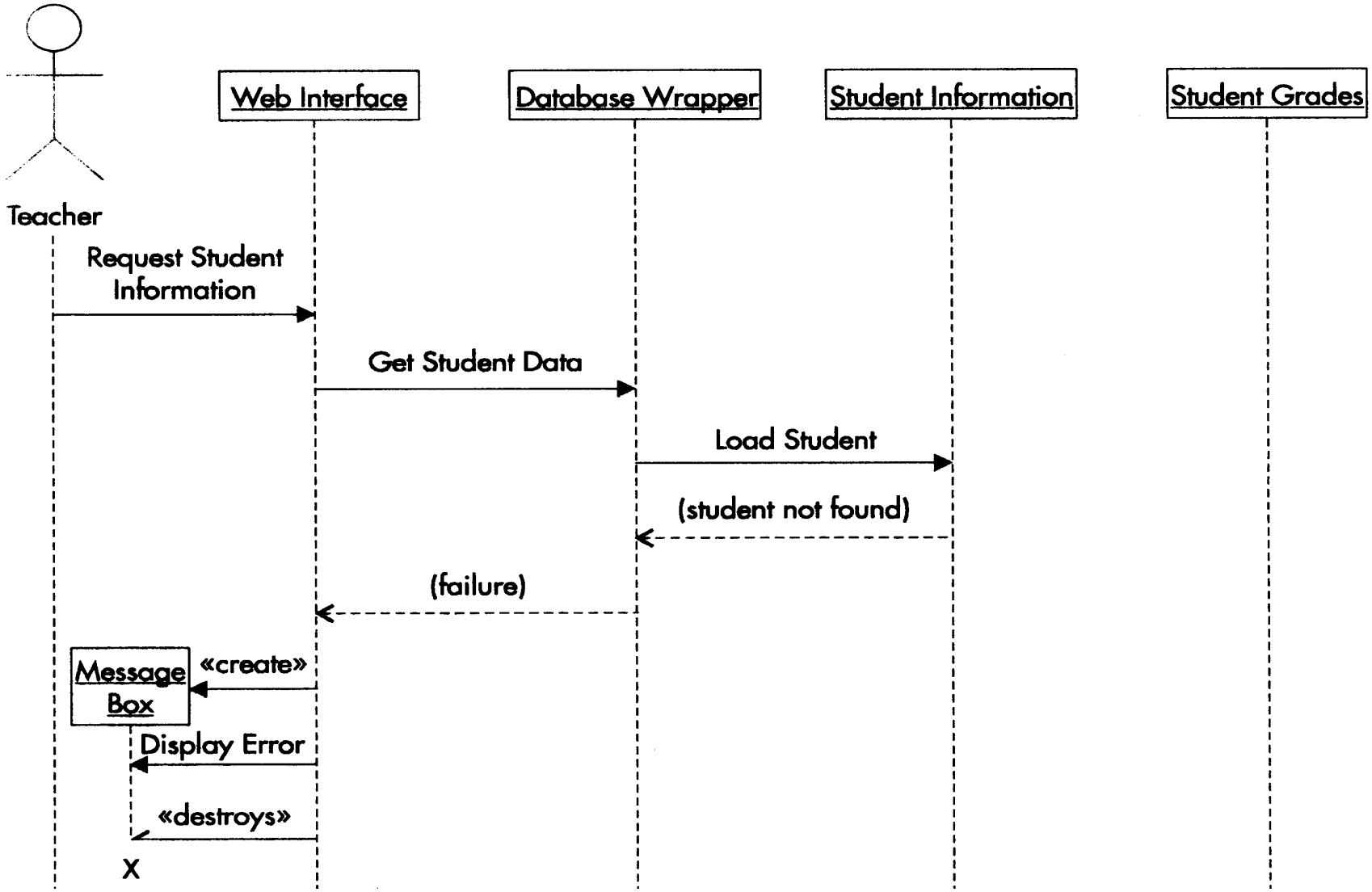


3．添加消息和条件

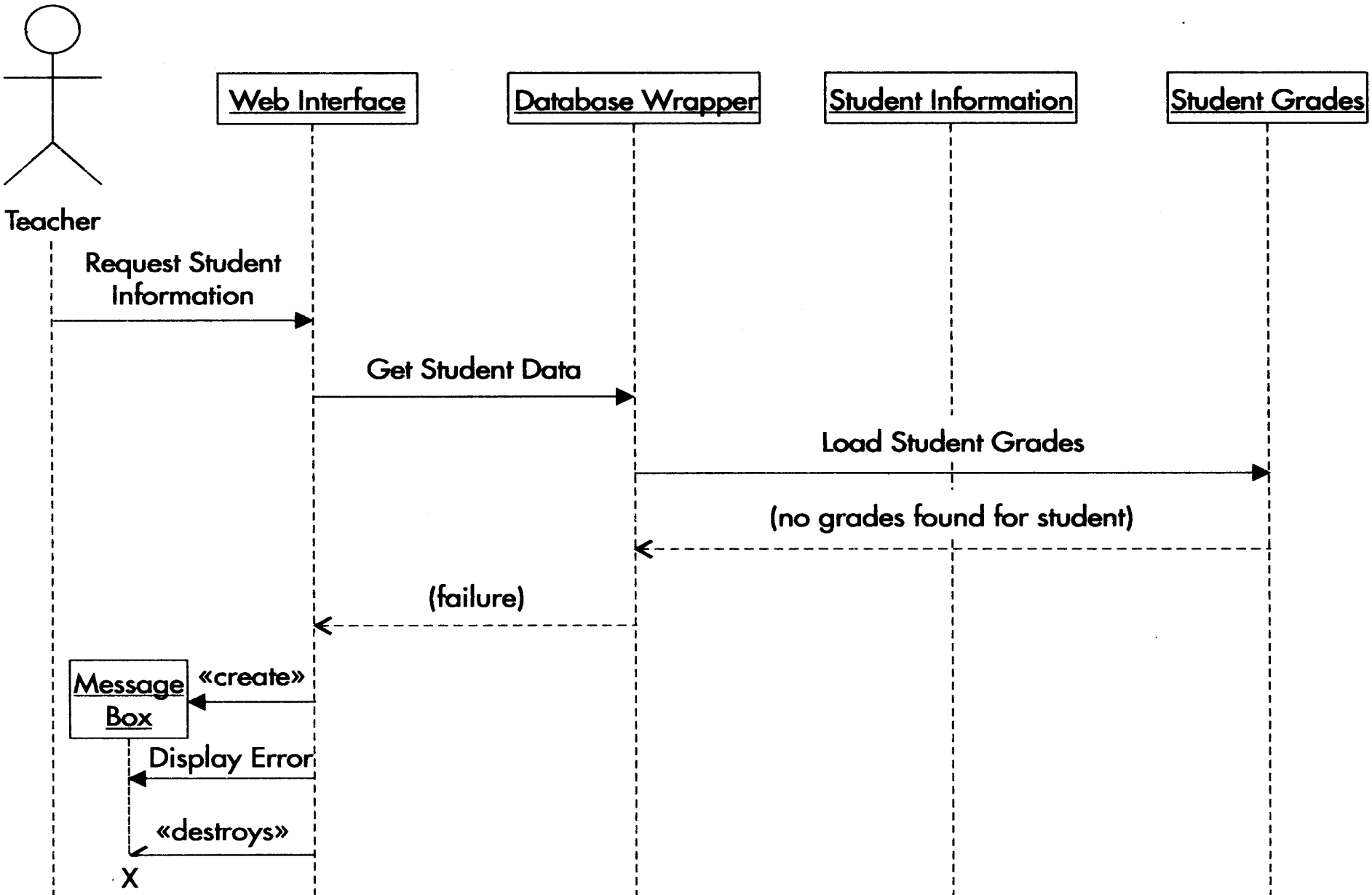
接下来，对每一个工作流作为独立的顺序图建模。从基本的工作流开始，它是没有出错条件，并且需要最少决策的工作流。在本例中，基本工作流是教师成功地检查某个学生的分数，如下图所示。



注意选择适当的消息类型（异步、同步、简单和返回）。接下来以独立的顺序图建模从属工作流。此处只建模否定的条件，如下图所示。



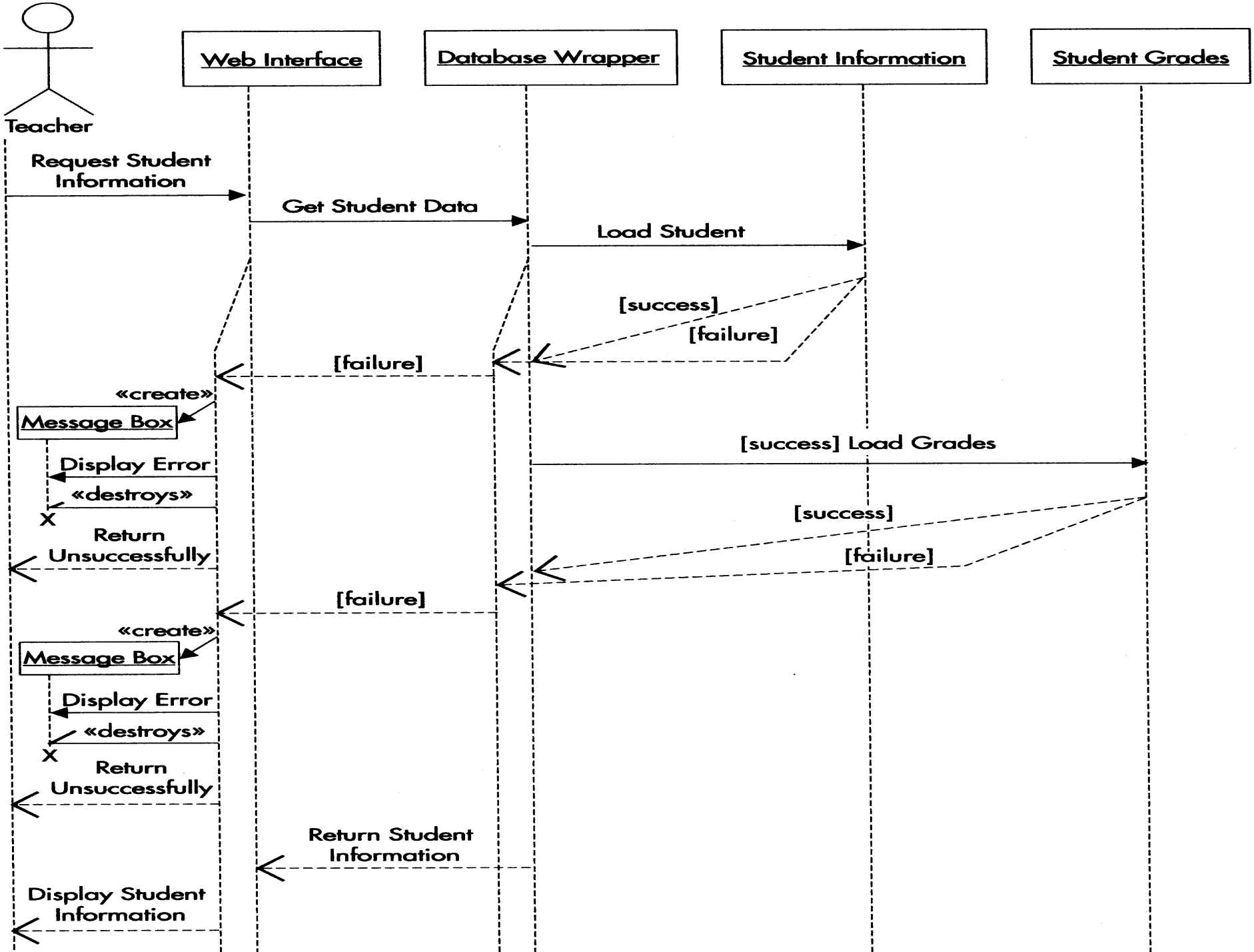
注意使用条件来指示在什么时候发送什么消息，如下图所示。现在已经完成了每一个工作流的顺序图。



4．绘制总图

建模顺序图的最后一步是把所有独立的工作流连接为一个总图，如下图所示。

在此阶段，如果觉得前面的消息和交互对于当前的顺序图过于详细，可以让它们更加泛化一些，但是在软件建模的下一个阶段，就会觉得初始的各个顺序图越详细越好。

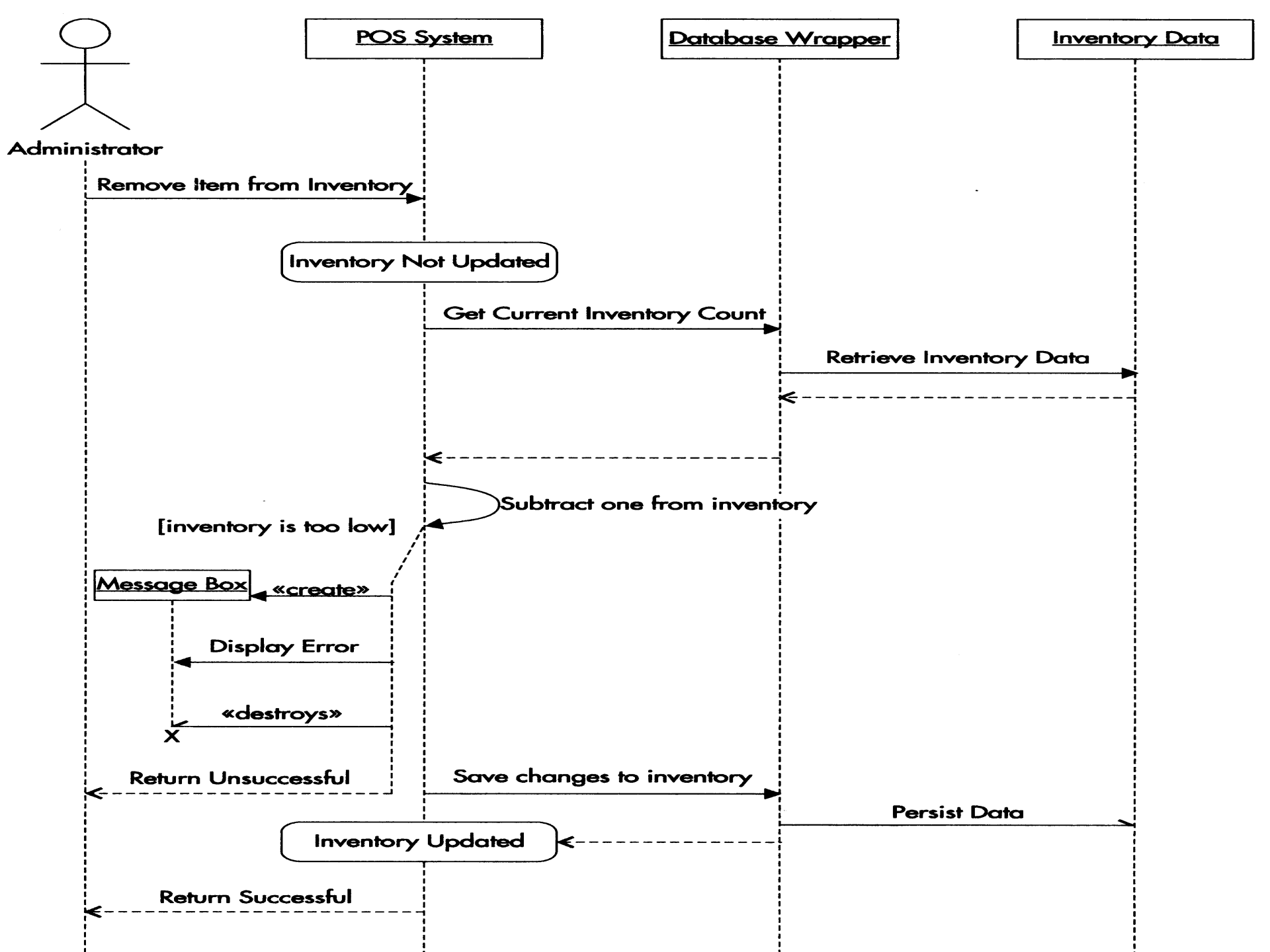

Teacher
Web Interface
Database Wrapper
Student Information
Student Grades
Request Student Information
Get Student Data
Load Student
[success]
[failure]
[failure]
«create»
Message Box
[success] Load Grades
Display Error
«destroys»
[success]
Return Unsuccessfully
[failure]
[failure]
«create»
Message Box
Display Error
«destroys»
Return Unsuccessfully
Return Student Information
Display Student Information

练习：建模一个顺序图

在这个练习中，将要为在销售商品后从库存清单中删除该商品条目的用例创建顺序图。综合运用自己掌握的顺序图的各种UML标记符，包括消息传递和消息类型、条件、分支，以及从属流。

练习步骤：

1）确定将要作为独立的顺序图建模的工作流。
2）布置各个独立的顺序图的对象。
3）为各个独立的顺序图添加消息和条件。
4）从各个独立的顺序图建模一个总顺序图。


POS System
Database Wrapper
Inventory Data
Administrator
Remove Item from Inventory
Inventory Not Updated
Get Current Inventory Count
Retrieve Inventory Data
Subtract one from inventory
[inventory is too low]
Message Box
«create»
Display Error
«destroys»
X
Return Unsuccessful
Save changes to inventory
Persist Data
Inventory Updated
Return Successful