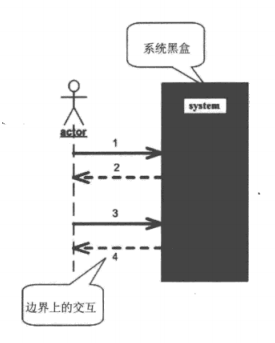
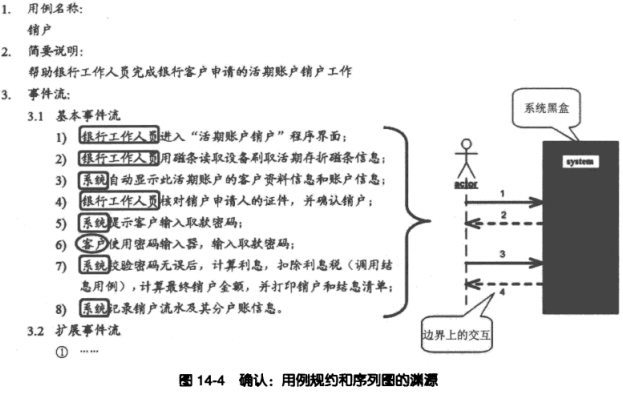
**描述需求的序列图与描述设计的序列图**

**描述需求的序列图**

描述需求的序列图把系统看作黑盒，描述系统与外部的交互

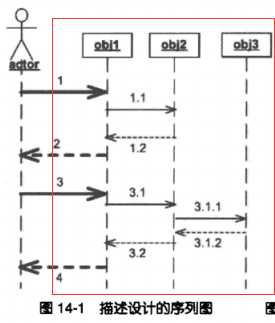


如何从用例规约构建需求序列图？



**描述设计的序列图3**

描述设计的序列图由架构师推导出来，描述完成某个用例的系统类间的交互



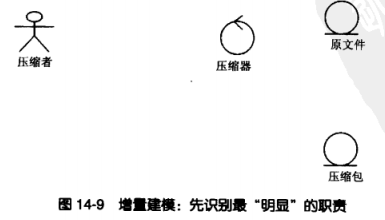
**用例驱动的模块划分过程**

用例驱动的模块划分是自底向上的方式，先推导出类，在划分模块

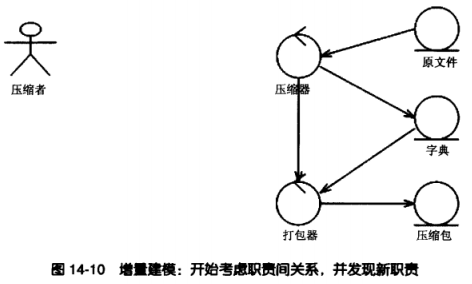
以下使用压缩用例进行示例

**应用描述需求的序列图，运用鲁棒图，发现用例需要的类**

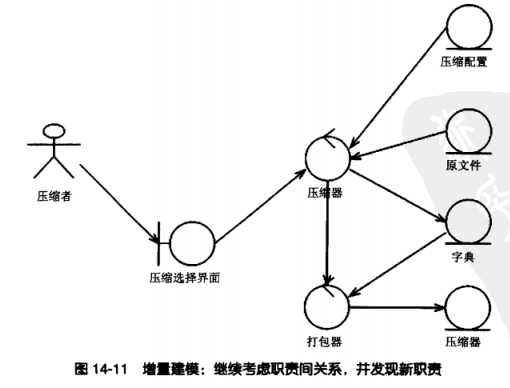
你认为压缩就是把原文件变成压缩吧，于是你想出了几个对象



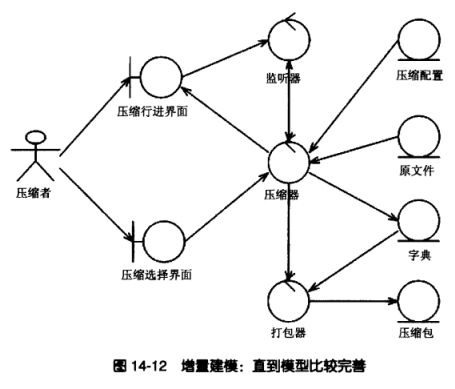
你心想，不应该由压缩器直接生成压缩包，于是你加入了打包器，那打包器通过什么来生成压缩包？于是你添加了一个压缩器产物字典



你看了一下需求序列图，哪里描述“打开压缩界面，选择压缩选项”，于是你加了一个压缩界面，又加了一个压缩配置保存压缩选项

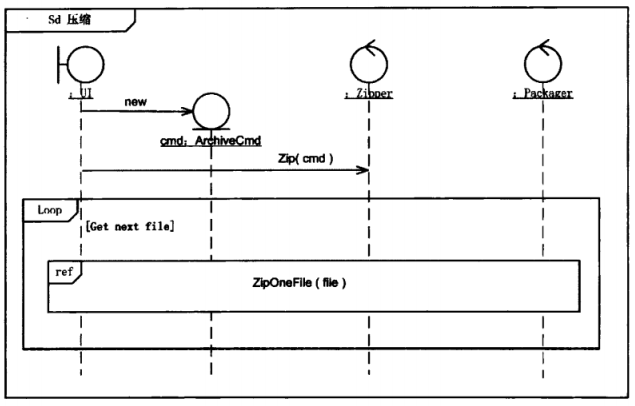


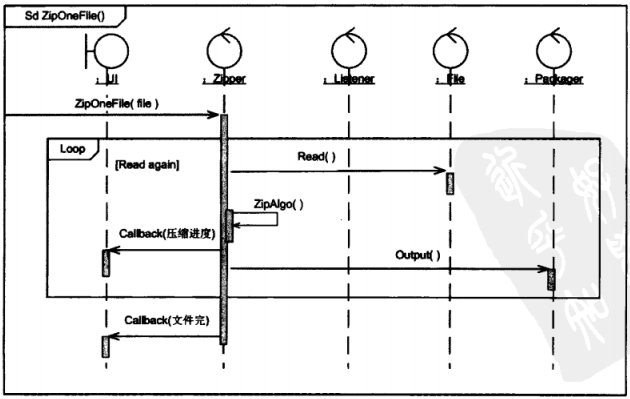
你又看了一下需求，那里描述“显示压缩进度”，于是你加了压缩行进界面，需求有描述“当点击取消时取消压缩”，于是你决定添加一个监听器



**通过鲁棒图识别类，构建序列图，明确类之间的关系**

我们通过鲁棒图识别了界面(UI)、 压缩器(Zipper)、 打包器(Packager)、监听器(Listener)、 压缩配置(ArchiveCmd) 等类





**划分模块**

确定了类以及类间的交互，我们依据这些类进行模块的划分，这里没有明确的规定应该要怎么划分，我们常用依据职责划分

如下我们划分了4个模块

