

## 目录

1、用例图.....	2
● 扩展 (extend) .....	4
● 包含 (include), 使用 (uses) .....	5
● 泛化 (generalization) .....	5
2、包图.....	7
3、状态图.....	8
4、时序图.....	8
5、协作图.....	10
6、活动图.....	10
7、类图.....	12

## 参考材料

UML1. X 中包括 8 种视图：静态视图、用例视图、实现视图、部署视图、状态视图、活动视图、交互视图和模型管理视图，以及 10 种图：类图、对象图、用例图、构件图、部署图、状态图、活动图、时序图、协作图和包图。在 UML2. 0 中作了修改和扩充，如下所示。

主题域	视 图 view	图 diagram
结构化 structural	静态视图 (static view)	类图 (class diagram)
	设计视图 (design view)	内部结构 (internal structure)
		协作图 (collaboration diagram)
		构件图 (component diagram)
	用例视图 (use case view)	用例图 (use case diagram)
动态的 dynamic	状态机视图 (state machine view)	状态图 (state machine diagram)
	活动视图 (activity view)	活动图 (activity diagram)
	交互视图 (interaction view)	时序图 (sequence diagram)
		通信图 (communication diagram)
物理的 physical	部署视图 (deployment view)	部署图 (deployment diagram)
模型管理 Model management	模型管理视图 (model management view)	包图 (package diagram)
	外围 (profile)	包图 (package diagram)

表 1 视 图

常用的是用例图、包图、时序图、状态图、协作图、活动图和类图（及对象图）等，正是论文需要涉及的。

### 1、用例图

用例模型的主要成分有用例、参与者和系统。系统被看作一个提供用例的黑盒，系统如何做、用例如何实现、内部它们如何工作，这些对用例建模都是不重要的。系统的边界定义了系统所具有的功能。用例表示功能，每个用例说明了一个完整的功能，其粒度要适当，如信息维护是一个功能，就不需要再细化到“增删改查”了（其对应的图称为四脚马车）。为清晰，需要以方框界定系统、子系统，框外是参与者（一般，左边为客户，右边提供服务），框内是用例，如图 1、图 3 等等。

用例图中的关系

关系	说明	记号
关联	执行者与他所参与的一个用例之间的通信路径	——
扩展	扩展的用例到基用例的一种关系，它指出扩展的用例所定义的行为如何插入到基用例所定义的行为中。扩展的用例通过模块化方式 (modular way) 增量地修改基础用例	《extend》 ----->
包含	从基用例到另一个用例（称为包含用例，inclusion use case）的一种关系，它指出包含用例定义的行为被包含在基用例所定义的行为中。基用例能看到包含用例，并依赖于执行包含用例后的结果，但两者相互间不能访问其它属性	《include》 ----->
用例流化	一个一般用例与一个更特殊的用例之间的关系，特殊用例可继承一般用例的特征	——>

表 2 用例图中的关系

用例需要涉及 INCLUDE (USES)，EXTEND 和 GENERALIZATION。

用例需要有文档，说明用例的前置条件、事件流（基本路径、可选路径）和后置条件，也可能有扩展事件流。请参见下述文档例。

论文需要提供 **3 个用例文档**，用例文档需要紧随在相关用例下，以利阅读理解。

以下介绍康健物品进销存信息管理系统、销售系统（含电话销售），金融贸易系统的部分用例图。图中的方框代表系统，椭圆代表用例，线条人代表参与者，他们之间的连接线表示关联。

康健物品进销存信息管理系统的系统用例图如下所示：

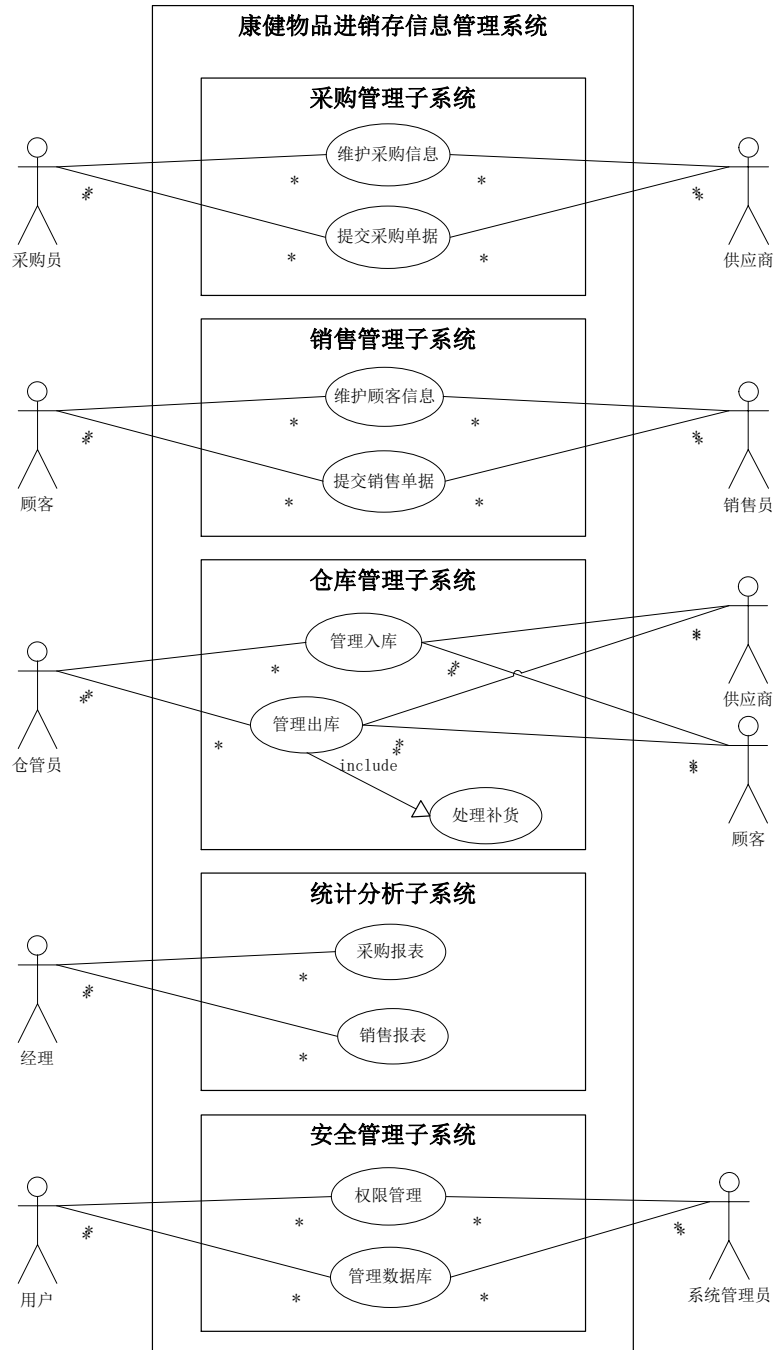


图 1 示例进销存信息管理系统

康健物品进销存信息管理系统由采购、销售、仓管、统计分析和安全等五个子系统组成。（图的说明，论文中的图都需要说明）

其采购管理子系统用例图如下所示：

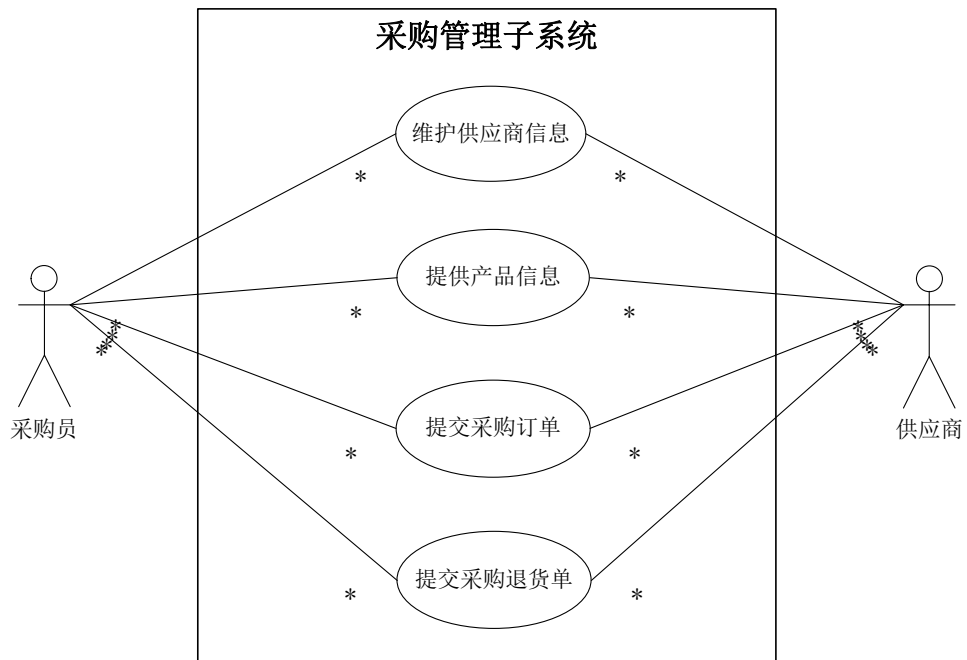


图 2 示例采购管理子系统

采购管理子系统用例由 2 个参与者和 4 个子用例组成。子用例是维护供应商信息、提供产品信息、提交采购订单和提交采购退货单。

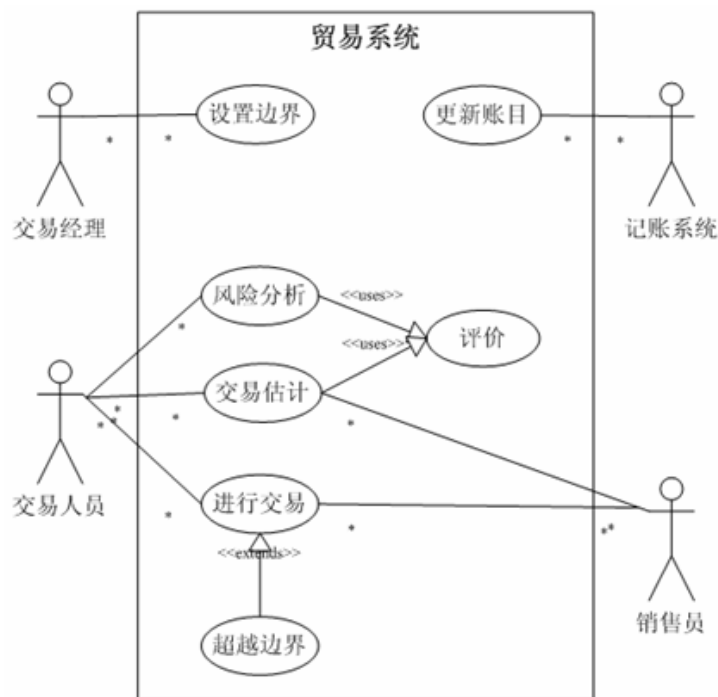


图 3 某个贸易子系统

该贸易子系统由 4 个参与者和 7 个子用例组成，其中含有包含和扩充关系。

#### ● 扩展 (extend)

扩展用例与基用例的一种关系：它指出扩展用例所定义的行为如何插入到基用例所定义的行为中。基用例中定义有一至多个已命名的扩展点，扩展关系是指将扩展用例 (Extension) 的事件流在一定的条件下按照相应的扩展点插入到基用例中。扩展用例通过模块化方式 (modular way) 增量地修改基用例的行为。

扩展（extend）是一种泛化关系，在这种关系中，通过向被扩展的用例添加动作来扩展用例。被扩展的用例（基用例）是一般用例，扩展的用例是特殊用例。图中“进行交易”用例是描述正常的交易，但如出现了某个特定客户超过了规定的最大交易量（超出了交易边界）时，就不能进行正常的交易，此时通过扩展，在“超越边界”用例中处理这种情况。

通常将一些常规的动作放在一个基用例中，而将非常规的动作放在它的扩展用例中。

对于包含关系而言，子用例中的事件流是一定插入到基用例中去的，并且插入点只有一个（一定运行）。而扩展关系可以根据一定的条件来决定是否将扩展用例的事件流插入基用例事件流（不一定运行），并且插入点可以有多个。

如对于电话业务，可以在基本通话（Call）业务上扩展出一些增值业务如：呼叫等待（Call Waiting）和呼叫转移（Call Transfer）。可以用扩展关系将这些业务的用例模型描述如下。

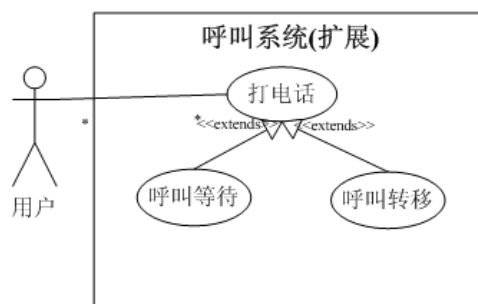


图4 某个呼叫子系统扩展图

- **包含（include），使用（uses）**

包含关系是在多个用例中重复出现，且单独形成价值。在使用（包含）关系中，一个用例（基用例）使用另一个用例（作为自身行为的一部分）。如贸易系统图中，“风险分析”用例和“交易估价”用例都需要对交易进行评价。此时可把“评价”作为一个单独的用例，被“风险分析”和“交易估价”两个用例使用。

通常在描述一般行为的变化时采用扩展关系，在两个或更多的用例中出现重复描述，又想避免这种重复时可采用使用关系。

- **泛化（generalization）**

一个一般用例与一个更特殊的用例之间的关系，特殊用例可继承一般用例的特征。

当多个用例共同拥有一种类似的结构和行为的时候，我们可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。

在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。在实际应用中较少使用泛化关系，子用例中的特殊行为都可以作为父用例中的备选流存在。

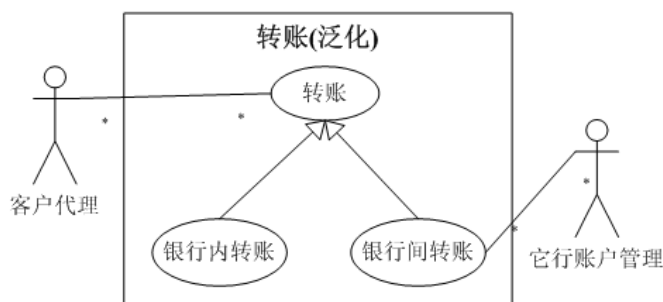


图5 某个转账子系统泛化图

注意，用例描述的是一个功能，其**粒度**要适当，如“信息维护”用例，就不需要再细化到CRUD——“增删改查”了（其对应的图俗称为**四脚马车**）。

请选择三个用例（不是子系统）书写其文档，说明前置条件、事件流（基本路径和可选路径）和后置条件，也可能有扩展事件流。（**注意**，论文中用例文档必须紧随在相关用例下方）

**用例文档【用例文档可以用于外包给其它公司实现编程的，故必须详细、清晰】**

“新增供应商”用例文档

用例名称	新增供应商
参与者	采购员、供应商
前置条件	一个合法的采购员已进入该界面
基本路径	<ol style="list-style-type: none"> <li>1. 系统显示增加供应商界面用例开始</li> <li>2. 采购员输入供应商提供的信息，名称、所在城市、详细地址、联系电话（要列出系统所需要的<b>全部</b>数据项）</li> <li>3. 系统校验供应商信息是否有效 <ol style="list-style-type: none"> <li>a) 系统校验供应商信息已经存在（这里需要具体化，怎么判断？）</li> <li>b) 若已存在，提示采购员重新输入供应商信息，转步骤 2.</li> </ol> </li> <li>4. 供应商表增加一条记录。</li> <li>5. 系统显示新增成功信息，用例结束</li> </ol> <p>可选路径：</p> <p>步骤 2, 3 都可以取消本次新增，用例结束。</p>
后置条件	供应商表出现新增加的供应商记录

表 3 新增供应商用例文档

“出库”用例文档：（注意：步骤 4-7 有瑕疵，需要整理完善，以外包给第三方编程。）

用例名称	出库
参与者	仓管员、顾客
前置条件	一个合法的仓管员已进入该界面
基本路径	<ol style="list-style-type: none"> <li>1. 界面显示待出库的订购单信息列表</li> <li>2. 双击选择一订购单</li> <li>3. 系统显示该订购单的明细列表</li> <li>4. 选择订购单明细（即商品）并输入数量，确认出库</li> <li>5. 生成出库单及出库单明细</li> <li>6. 更新库存</li> <li>7. 检查各出库商品库存量是否有小于最小库存量的，若有，则生成各相应商品的补货清单</li> <li>8. 如果订购单中的商品全部出库则更新订购单状态为已出库，否则订购单状态更新为部分出库，等待下一次处理或发给其他仓库处理，用例结束</li> </ol> <p>可选路径：（需要增加内容）</p>
后置条件	<ol style="list-style-type: none"> <li>1. 若出库成功，则库存表中的相应商品数量被更新，否则商品数量不变</li> <li>2. 若有小于最小库存量的商品则补货清单表中有记录生成</li> <li>3. 若出库成功，则顾客可以看到订单状态变为已出库</li> </ol>

表 4 出库用例文档

“取款”用例文档：

参与者：银行职员（用户代理）

◆ 前置条件：一个合法的银行职员（用户代理）已进入该界面。

◆ 事件流：

基本路径

1. 当选择取款功能时用例开始
2. 当输入客户信息：姓名、帐号等（有瑕疵，需要列出全部数据项）后
  - a) 如果客户信息与帐户不一致，显示错误信息，可以重新输入或结束用例
  - b) 如果该帐户被冻结（如因挂失而冻结），显示冻结信息并结束用例
3. 输入密码，系统校验密码，若正确，转 4；  
若错误则计数，计数小于 4，返回 3，否则终止用例。
4. 输入取款金额，如果该帐户的余款小于取款金额，显示错误信息，转 4.
5. 打印取款单，交客户签字
6. 建立取款事件记录，更新帐户信息
7. 存折打印，用例结束

可选路径

1. 在第 5 步客户签字之前的任何时刻，客户可以取消本次取款，用例结束
2. 第 3 步校验密码时，如发现密码不一致，则可重新输入密码 3 次，或用例结束

◆ 后置条件：如果取款成功，产生取款记录，客户帐户中的余额被更新（减少），否则余额不变。

用例文档（与时序图、活动图等配合），可以供第三方编程实现，故需要详细描述。

## 2、包图

以包图描述的示例销售系统，如下所示。

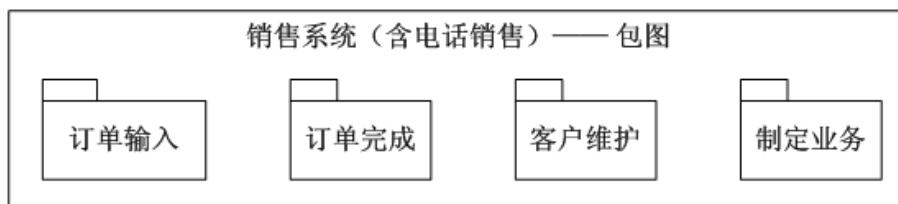


图 6 某个销售信息管理系统包图

该销售系统由订单输入、订单完成、客户维护和制定业务等四个子系统组成。

### 3、状态图

状态图通常是对**类描述的补充**（类的状态），它说明该类对象所有可能的状态，以及哪些事件将导致状态的改变。状态图描述了对象的动态行为，是一种对象生存周期的模型。

状态图的基本符号如下所示。

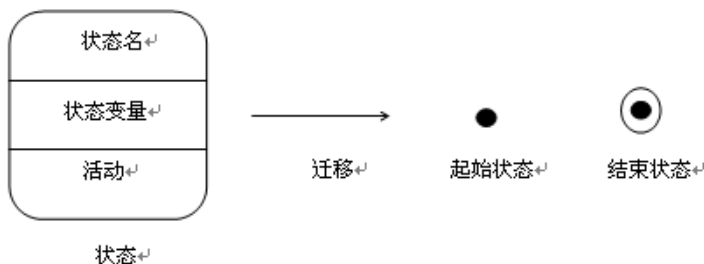


图 7 状态图图符

如简化的“货物”有 3 个状态，状态转换有：无货状态（from/to）有货状态（from/to）需补货状态等。

客户、供应商、员工等具体实体会多个状态（如 VIP、等级、晋级、降级等等）。

“采购单”的状态可以有：制作、废弃、执行（多次）、完成等状态。

描述**电梯升降的状态图**如下所示。

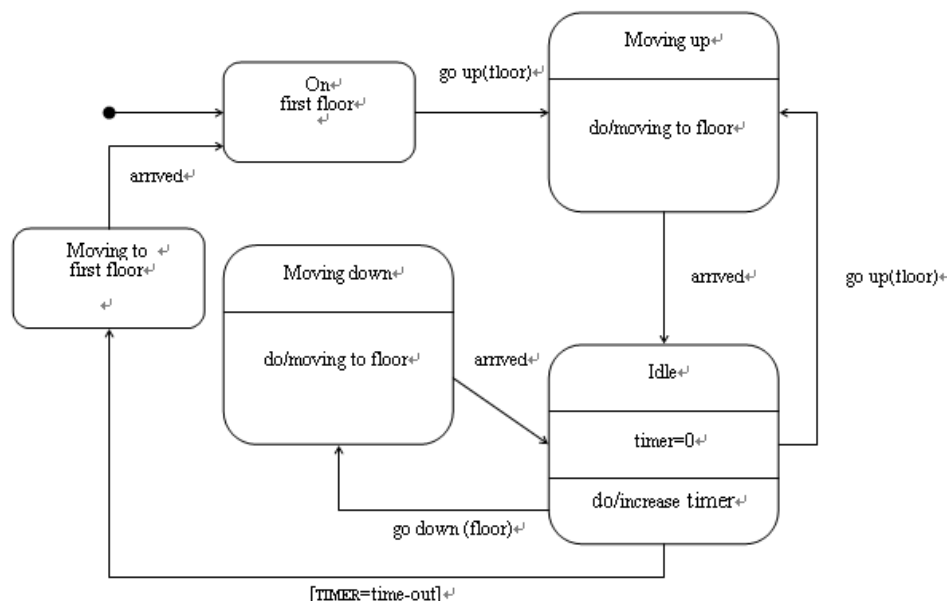


图 8 电梯升降状态图

### 4、时序图

时序图用来描述**对象间的动态协作关系**，它关注于消息的顺序，即对象间消息的发送和接收的顺序。时序图还揭示了一个特定场景的交互，即系统执行期间发生在某时间点的对象之间的特定交互。它适合于描述实时系统中的时间特性和时间约束。

时序图有两个坐标，垂直坐标表示时间（从上到下），水平坐标表示一组对象。

从对象框垂下的虚线称为**对象的生命线**（故最左的应连续不断），表示用例建模时对象的生命期，生命线上的细长框是**对象激活条**。对象间的通信用对象生命线间的水平消息线表示。



打印的时序图如下所示（带分支）

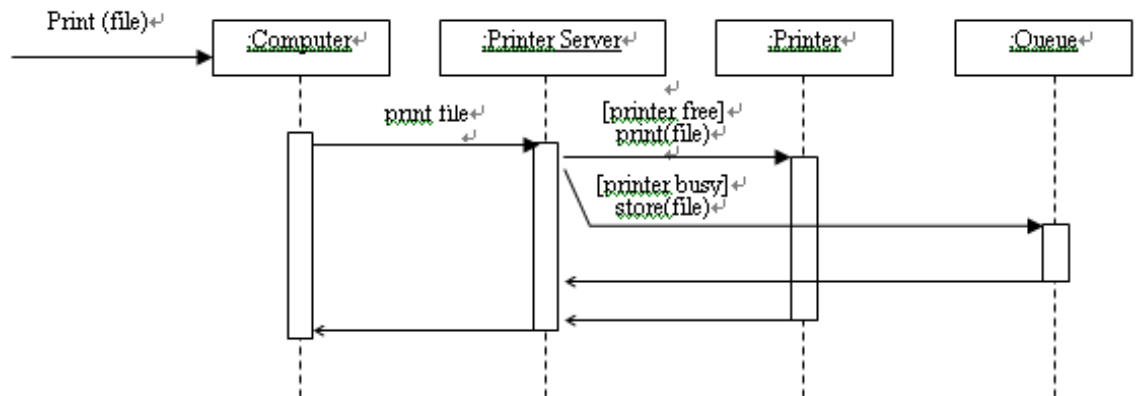


图 9 示例打印时序图

领料出库时序图：（简化了 ADO，需要补全）

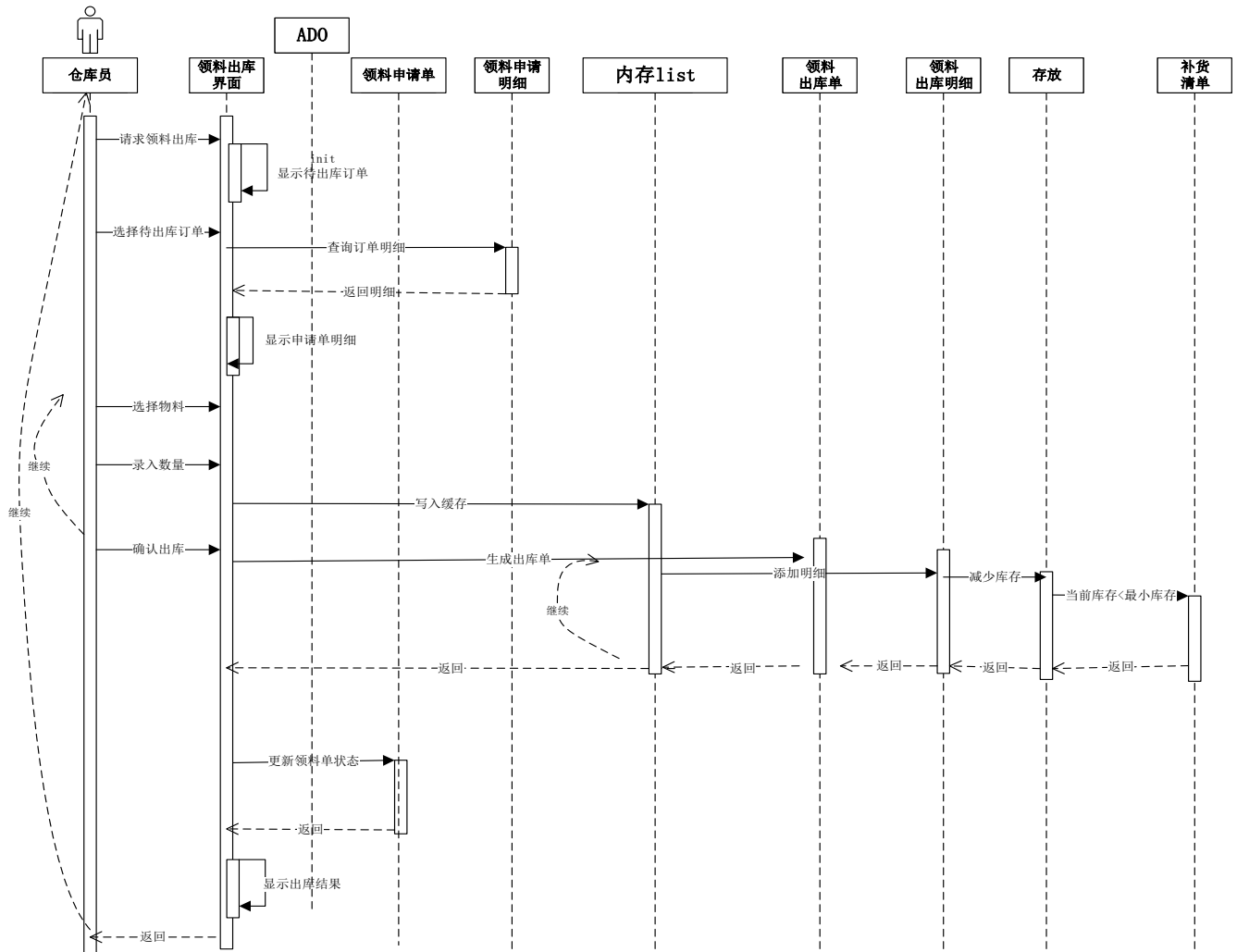


图 10 示例领料出库时序图

出库是按领料单(或销售订购单及其明细产生出库单和明细，减少库存时会产生需要补货信息。

## 5、协作图

协作图着重于**协作对象之间的交互和链接**（link，指对象实例之间的物理或概念上的连接，一个链接是某关联的一个实例）。它可用于图示系统中的操作执行、用例执行或一个简单的交互场景。协作图展示了对象及其间的链接，还展示了链接的对象之间如何发送消息。

协作图和时序图都可以用来展示系统的交互，时序图强调时间和顺序（对象间通信模式），协作图则强调对象之间的组织，即协作（周境 context）。

剧院售票协作图如下所示。

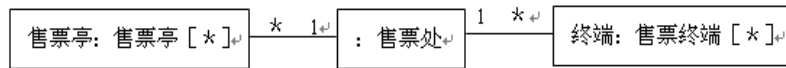


图 11 售票协作图

控制电梯运行的协作图如下所示。

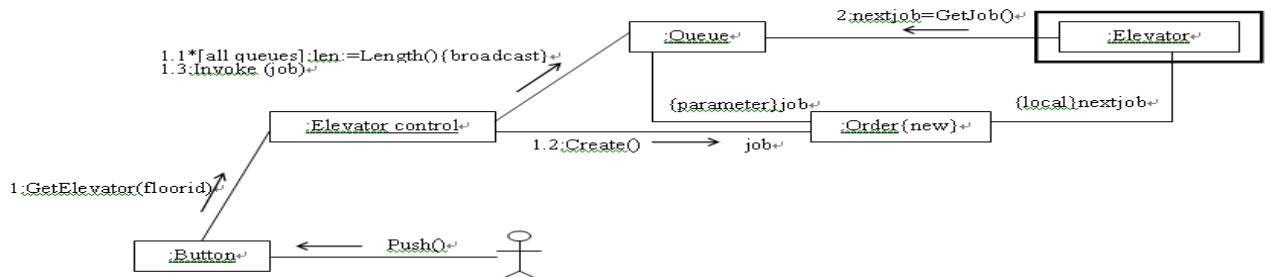


图 12 示例电梯运行协作图

在协作图的对象框中，可用 {new} 或 {destroyed} 表示该对象在协作期间被创建或消亡。{transient} 则表示对象在同一个协作期间被创建并消亡，如下所示。

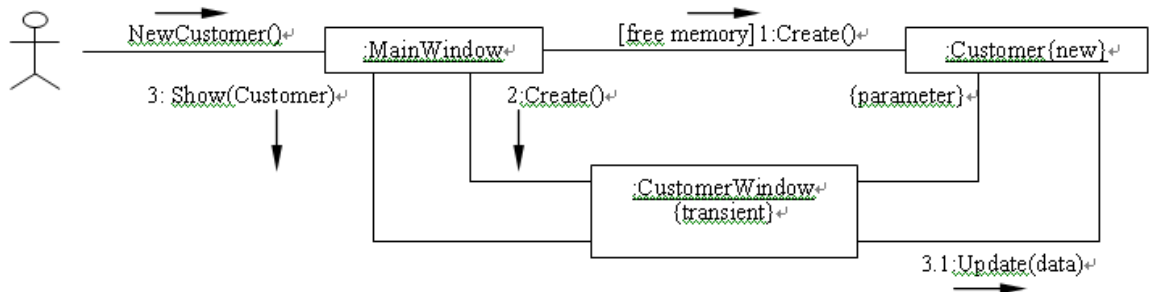


图 13 对象创建、消亡协作图

## 6、活动图

活动图是用活动描述**用例的活动顺序**。描述 workflows 中需要做的活动和执行这些活动的顺序。故包含活动、系统状态和执行活动的条件。

活动图使用状态图的符号表示，它有与状态图相同的开始点和结束点，活动图中的状态称为动作状态，用圆角矩形表示，动作状态之间的迁移用箭头表示，迁移上可以附加警戒条件、发送子句和动作表达式。**活动图**是状态图的变形，它根据对象状态的变化捕获动作（所完成的工作和活动）和它们的结果，它表示了各个动作及其间的关系。

与状态图不同的是，活动图中动作状态之间的迁移不是依靠事件触发的，当动作状态中的活动完成时迁移就被触发。在活动图中，事件只能附加到开始点到第一个动作之间的迁移。

例如：描述示例**采购入库**活动图和**取款**用例的活动图如下所示。（参考出库时序图，下图缺少缓存——内存 list，需要补全）

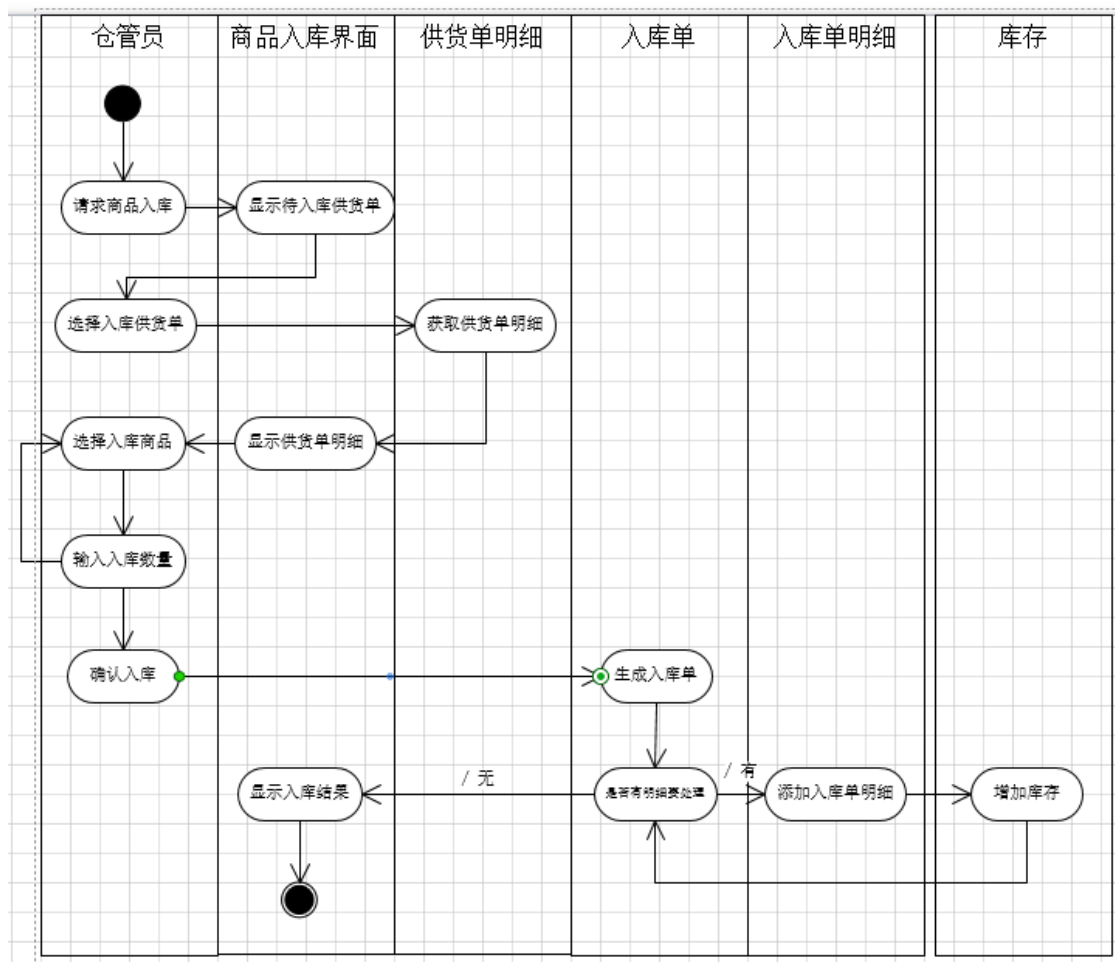


图 14 示例采购入库活动图

采购入库涉及采购订单和明细，按明细产生入库单、明细，并修改库存。

例如：描述取款用例的活动图如下所示。

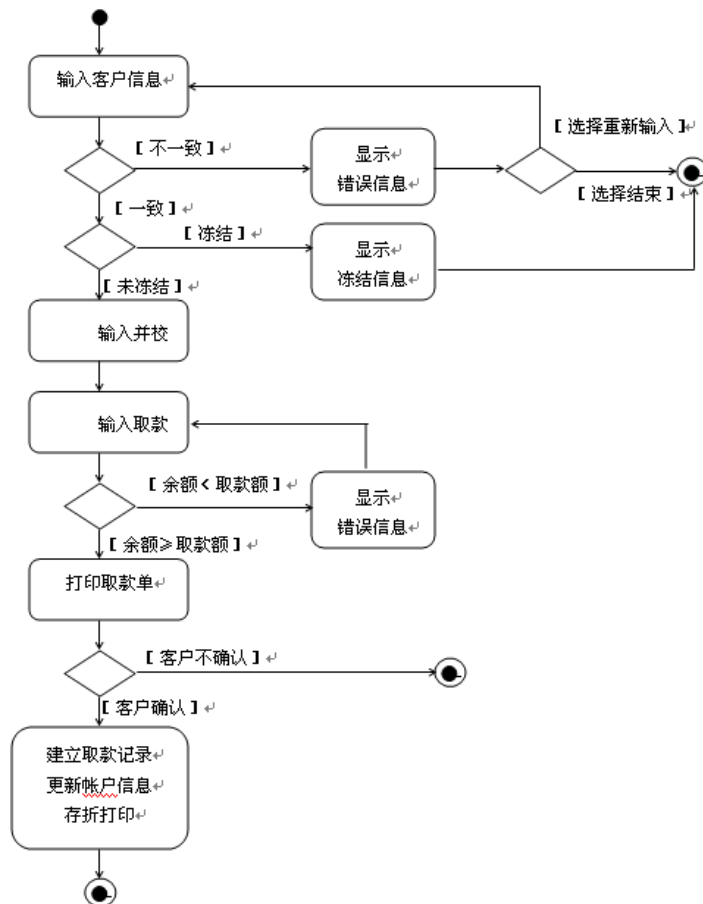


图 15 取款活动图

时序图和协作图用于对共同完成某些动作的对象群体进行建模。

状态图和活动图用于对单个对象（类、用例等）的生命期建模。

活动图一般为一个过程中的活动序列建模，状态图为对象生命期中的各离散状态建模。

注意，协作图、活动图是与时序图一致的。

## 7、类图

类图中的关系有关联（association）、依赖（depending）、泛化（generalization）、实现（realization）等。

关系	功能	符号
关联	类实例间连接的描述	——
依赖	二个模型元素之间的一种关系	---▶
泛化	更特殊描述与更一般描述之间的一种关系，用于继承和多态性类型声明	——▷
实现	规约（specification）与它的实现之间的关系	---▷

表 5 类间的关系

关联关系例 1：关联名 name

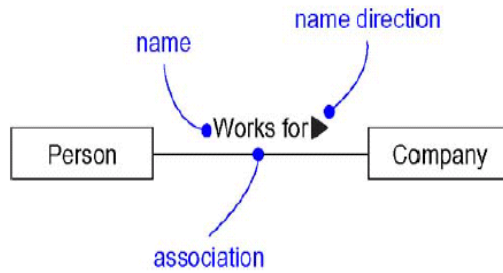


图 16 关联名 name

关联关系例 2：角色（role）与多重性（Multiplicity）

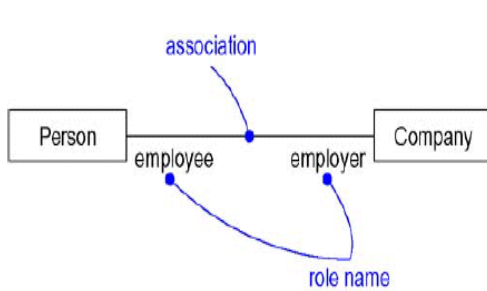


图 17 角色（role）与多重性

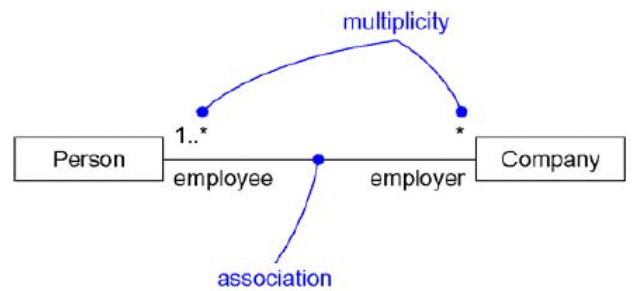


图 18 角色与多重性（Multiplicity）

关联关系例 3：聚集和组合（Aggregation and composition）——能否独立存在，与场景有关

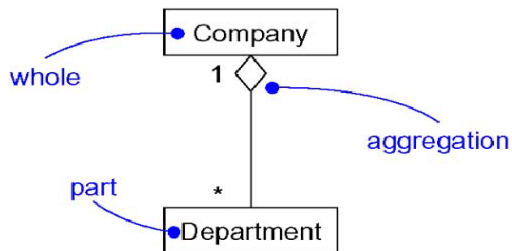


图 19 聚集（Aggregation）

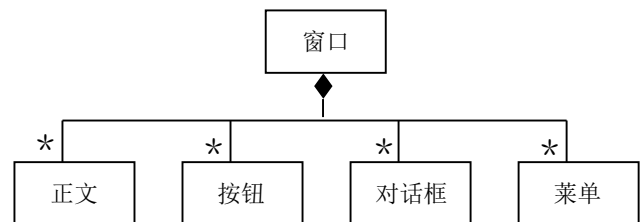


图 20 组合（composition）

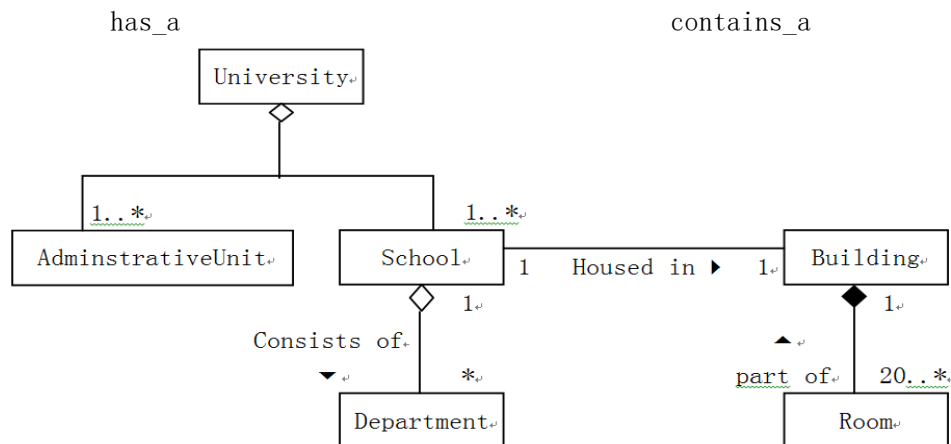


图 21 关于聚合和复合的类图

泛化关系例（Generalization）

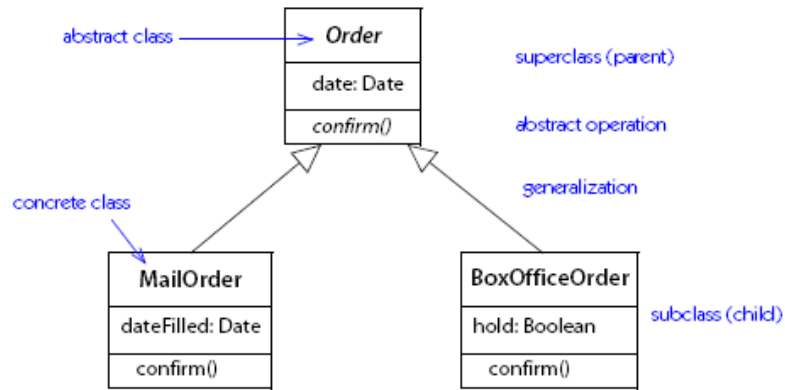


图 22 泛化

受限关系例 (Constraints)

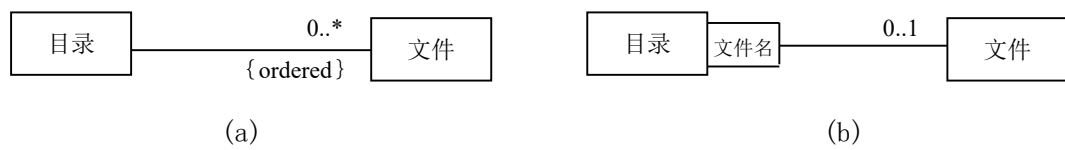


图 23 受限关系

类之间的关系 例 1:

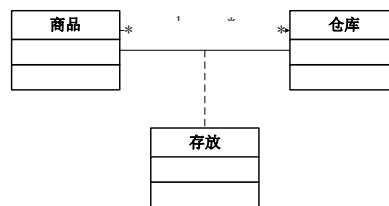


图 24 二元关系图

例 2: 货物、仓库关系图

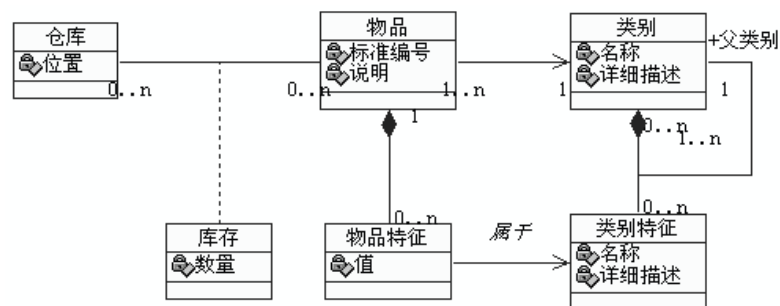


图 25 货物、仓库关系图 (不需要箭头)

例 3: 公司、部门和员工 (派生或约束):

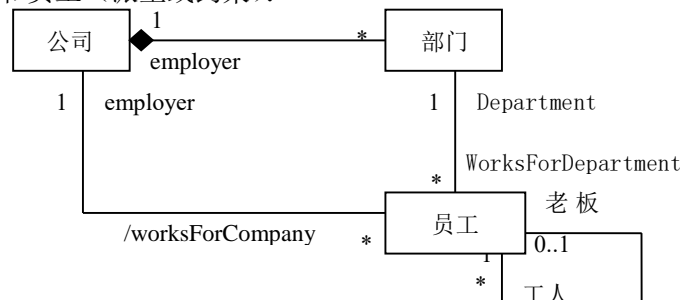


图 26 公司、部门和员工关系图



说明:

采购员、销售员等泛化为员工；员工聚集于部门；部门组合为公司。

采购订单涉及采购员和供应商，采购订单由多条采购明细组成，明细涉及商品。

因采购而入库，即一个采购单可以关联多个入库单。

供应商因采购而供应货物，即系统保留了供应商可以供应的商品（只是本系统需要的，有兴趣的商品）——回顾，综合实践中的“供应”是因采购而供应(即收货)。

商品与仓库的关联类是存放。(为效率，需要这个类——对应于瘦表)

销售类似于采购。

销售订单可以导出出库单，出库单可以导出补货单。

注意：若论文与此类似，需要增加采购退货或销售退货。