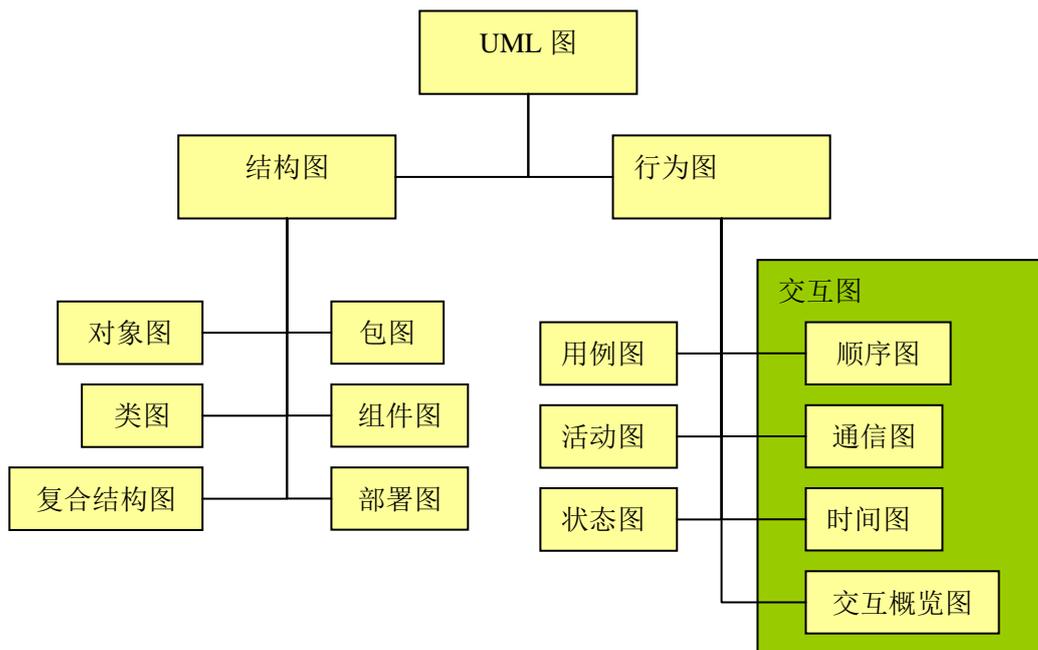


UML 13 种图

UML(统一建模语言)是为软件系统的制品进行详述(specifying)、可视化(visualizing)、构造化(constructing)、文档化(documenting)的一种语言。UML 目前是软件行业标准的建模语言。可以采用 UML 对需求、分析、设计、实现、部署等工作进行标准的形式化描述,达到交流共同的作用。同时基于 UML 的模型驱动开发支持以架构为核心、快速分析设计、迭代增量的开发方法。

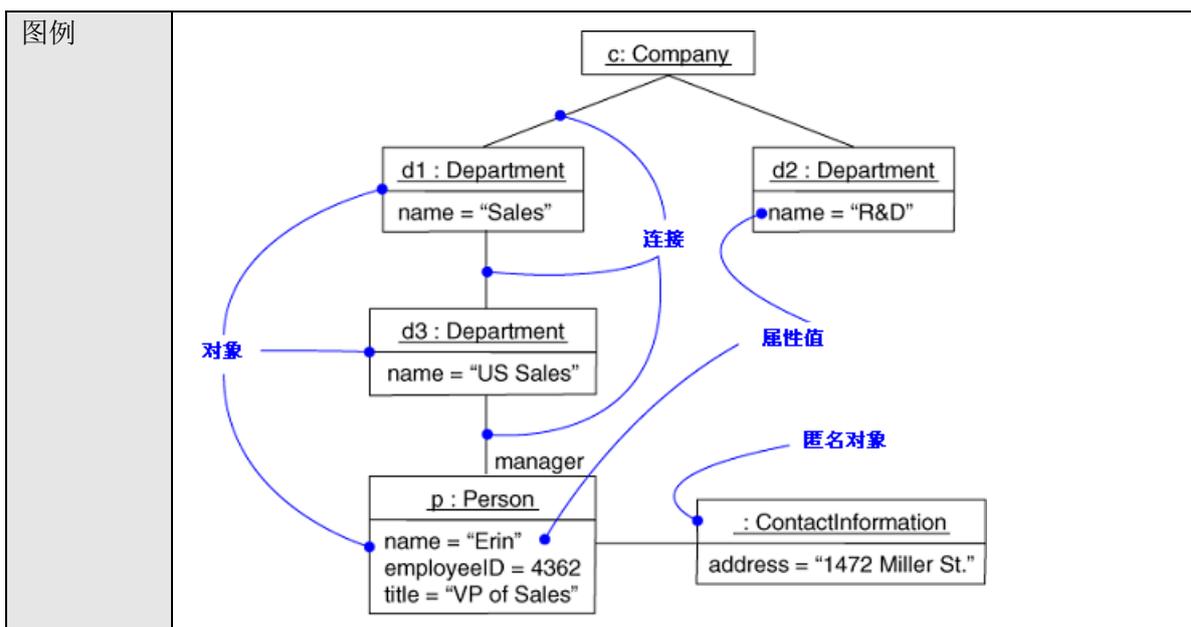
UML 语言由图元、关系、扩展机制和对象约束语言(OCL)。

UML2.1 规范一共定义了 13 种图,分为 2 大类,静态的结构图和动态的行为图:



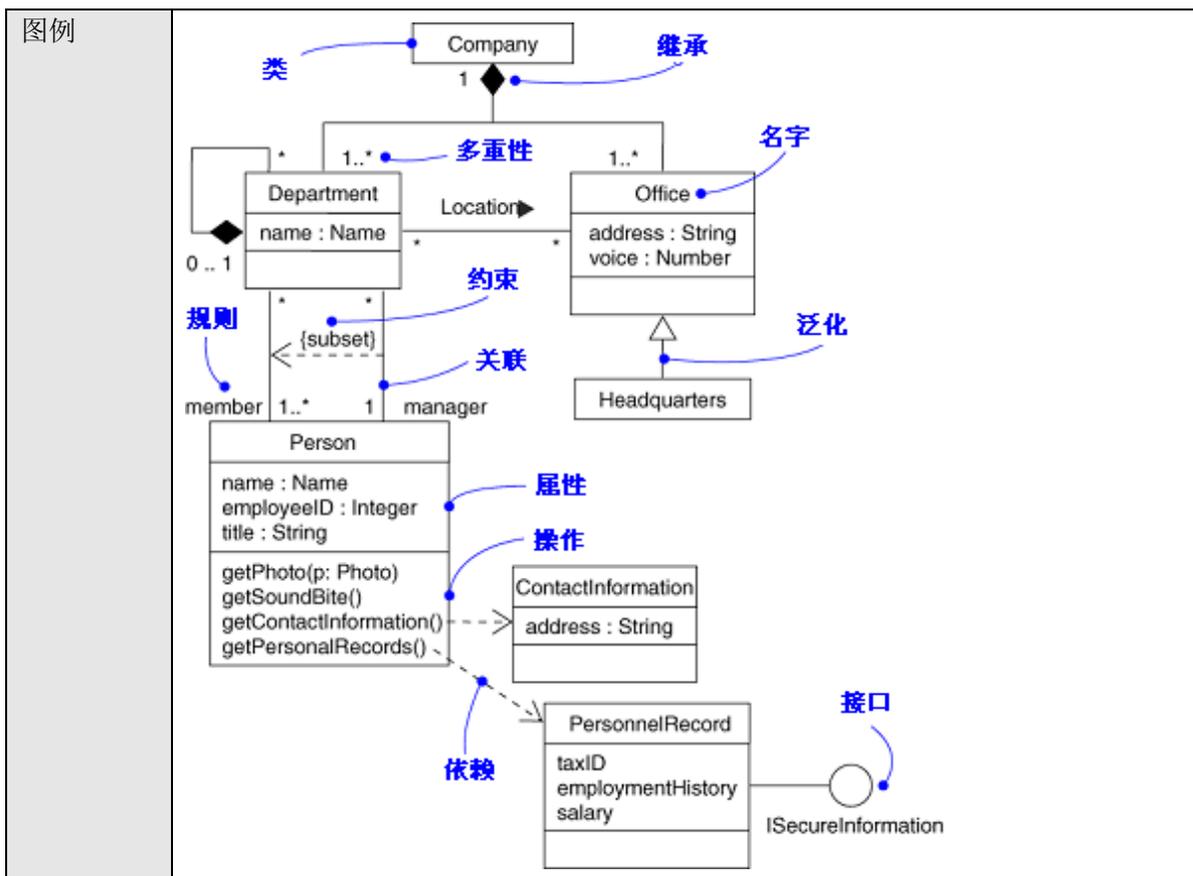
对象图

语义	描述对象在某个时刻的状态和关系
关键元素	对象(object)
关系	对象之间的连接(Link)
用途	可以在分析类的时候描述对象之间的关系 可以用于测试的时候捕获内存的对象快照



类图

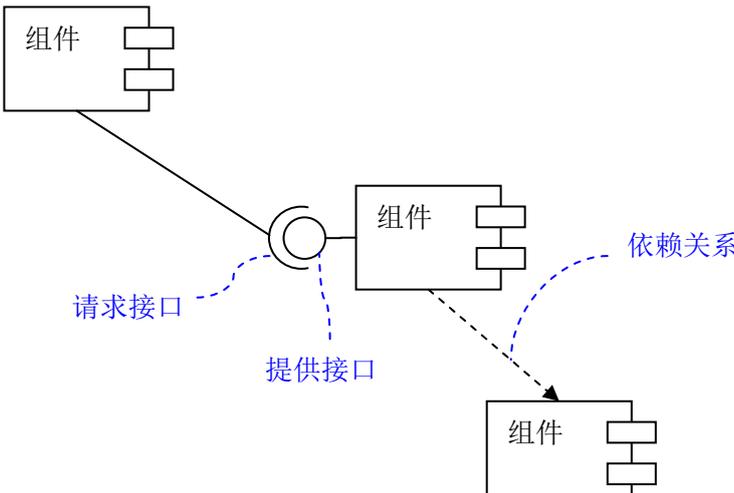
语义	描述类及其之间的关系
关键元素	类 (class), 接口(interface)
关系	类之间的关系: 关联, 聚合, 组成, 依赖, 泛化, 类和接口的关系: 实现
用途	进行面向对象分析设计的时候, 描述类。 应用设计模式的时候, 建立设计和代码实现的映射。



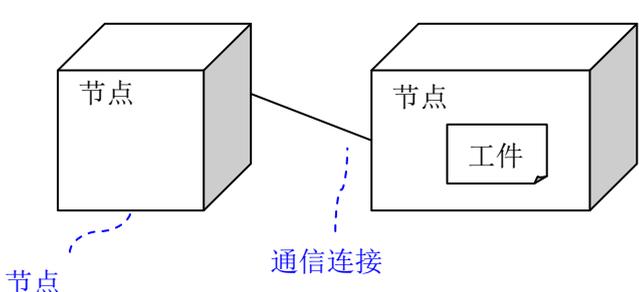
复合结构图

语义	描述复合体内部的组成及其之间的关系
关键元素	复合结构 (composite), 部件 (part), 端口(port)
关系	连接关系(connector)
用途	可以描述具有整体和部分结构的复合结构和元素之间的关系。
图例	<p>复合体</p> <p>部件</p> <p>连接器</p> <p>端口</p> <p>复合体</p> <p>部件</p> <p>部件</p>

组件图

语义	描述组件以及组件之间的依赖关系
关键元素	组件(component), 提供接口, 请求接口
关系	组件之间的依赖关系(dependency)
用途	可以在开发前划分组件, 设计组件的集成关系 可以描述产品的组件配置
图例	 <p>The diagram illustrates component relationships. It shows three component boxes, each labeled '组件' (Component). The top-left component has two provided interfaces (represented by small rectangles). A solid line connects this component to a central component that has one provided interface and one required interface (represented by a circle). A dashed line labeled '请求接口' (Required Interface) points from the central component's required interface to the top-left component's provided interface. Another dashed line labeled '提供接口' (Provided Interface) points from the central component's provided interface to the bottom-right component's provided interface. A dashed arrow labeled '依赖关系' (Dependency) points from the central component to the bottom-right component.</p>

部署图

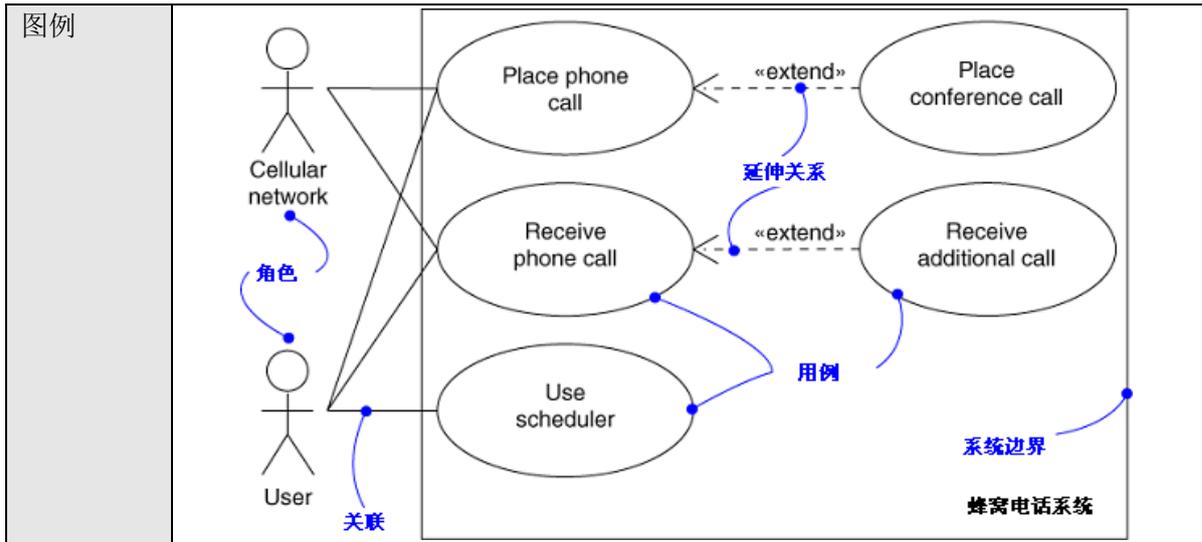
语义	描述系统的分布
关键元素	节点组件(Node), 工件 (artifact)
关系	节点之间的通信连接依赖关系(connection) 工件到节点的部署关系 (deploy)
用途	可以描述系统的节点分布, 以及节点上的工件部署
图例	 <p>The diagram shows two nodes, represented as 3D boxes and labeled '节点' (Node). A solid line connects the two nodes, labeled '通信连接' (Communication Connection). The right node contains an artifact, represented as a rectangle with a folded corner, labeled '工件' (Artifact). A dashed line labeled '节点' (Node) points to the left node, and another dashed line labeled '通信连接' (Communication Connection) points to the line between the nodes.</p>

包图

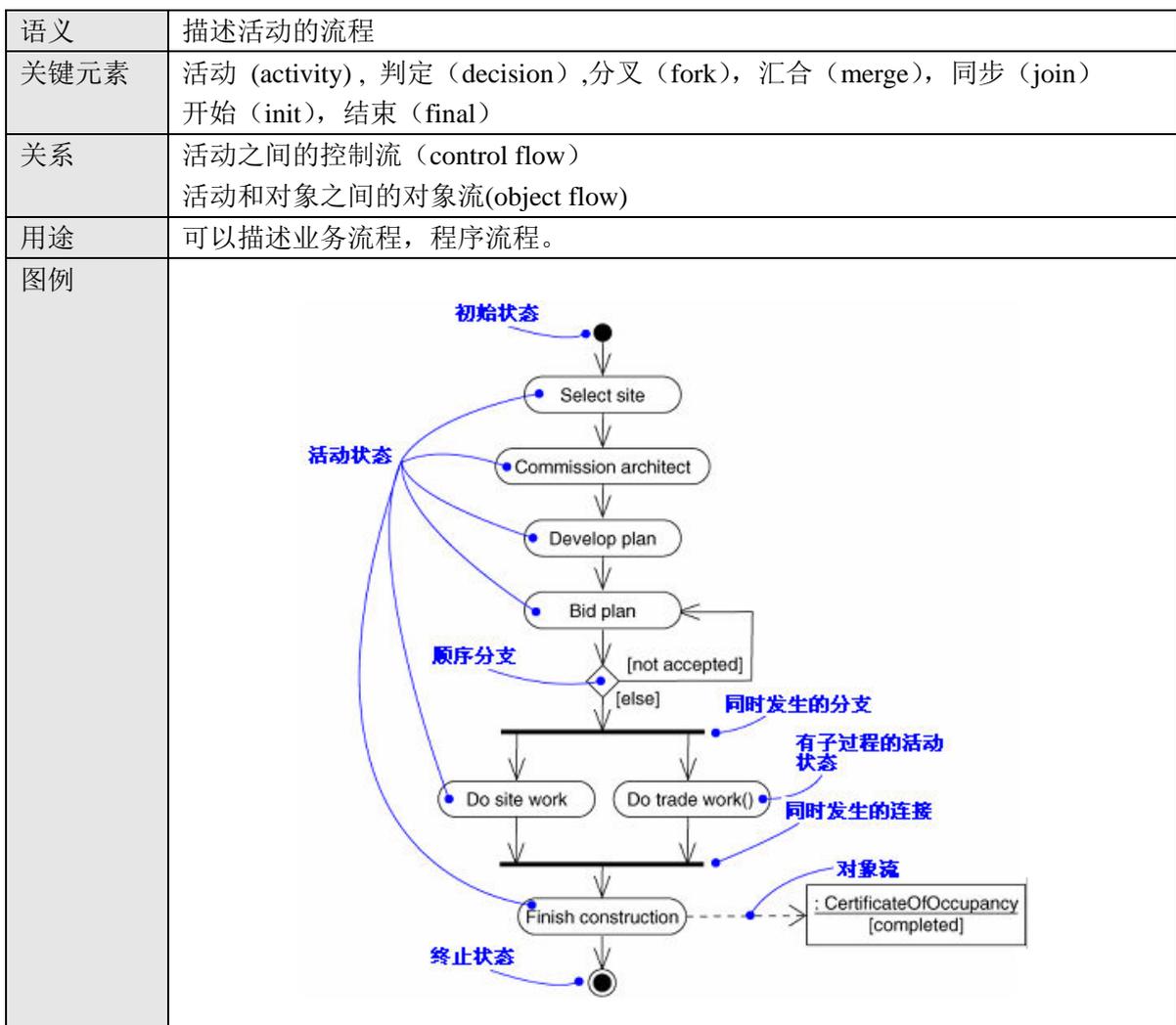
语义	描述包之间的依赖关系
关键元素	包 (package)
关系	包之间的依赖关系(dependency)
用途	可以描述包的内部元素，以及包之间的依赖关系。
图例	<p>The diagram illustrates a package dependency. At the top, a package named 'AWT' is shown. Inside this package, three classes are defined: 'Frame', 'Color', and 'Window'. Each class is represented by a rectangle with two small rectangles on its left side, indicating they are classes. Below the 'AWT' package, another package named 'mypackage' is shown. A dashed arrow with an open arrowhead points from the 'mypackage' package to the 'AWT' package, representing a dependency relationship where 'mypackage' depends on 'AWT'.</p>

用例图

语义	描述用户使用系统的场景
关键元素	用例 (use case), 角色 (actor)
关系	用例之间的包含关系(dependency), 扩展关系 (extend), 泛化关系(generalization), 角色之间的泛化关系(generalization)
用途	可以从用户的角度描述用户和系统的交互需求。



活动图



状态图

语义	描述某个对象在生命周期中的状态变化
关键元素	状态(state), 判定 (decision) 开始状态 (init), 结束状态 (final)
关系	状态之间的转移 (transation)
用途	可以描述数据对象, 控制对象或者界面的状态变化。
图例	<p>The diagram illustrates a state machine with the following components and labels:</p> <ul style="list-style-type: none"> 终止点 (Final State): A solid black circle at the top. 起始点 (Start State): A solid black circle on the left. 状态 (State): <ul style="list-style-type: none"> Idle: Contains the activity <code>keepAlive / check()</code>. Working: Contains nested states Connecting and Connected. 转换 (Transition): <ul style="list-style-type: none"> <code>onHook</code> from Idle to Working. <code>offHook / reclaimConnection()</code> from Working back to Idle. <code>ready(3) [signalOK]</code> from Connecting to Connected. 其他标注: <ul style="list-style-type: none"> 内部转换 (Internal Transition): A dot on the Idle state. 事件 (Event): A dot on the <code>ready(3)</code> transition. 活动 (Activity): The <code>keepAlive / check()</code> and <code>ready(3)</code> blocks. 嵌套状态 (Nested State): The Connecting and Connected states within Working. 防护 (Guard): The <code>[signalOK]</code> guard on the ready(3) transition.

顺序图

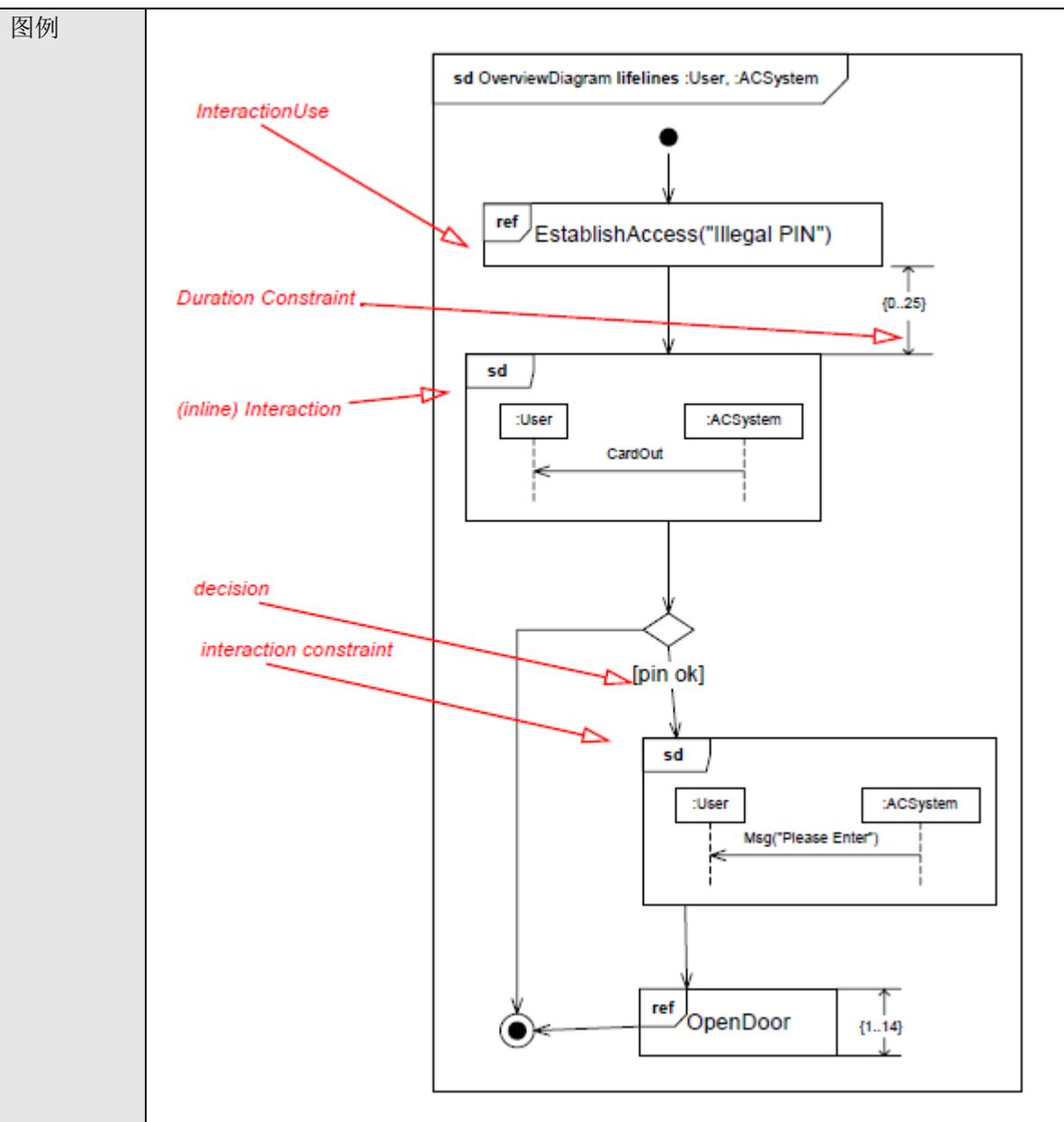
语义	描述几个对象随时间顺序而进行的交互
关键元素	对象 (object), 生命线 (lifeline) 激活, 也称之为执行定义 (execute specificaiton)
关系	对象之间的消息(message)
用途	可以描述用户和系统的交互, 也可以描述程序对象之间的交互。
图例	<p>The diagram shows the following interaction:</p> <ul style="list-style-type: none"> 对象 (Objects): <code>t: Thread</code> and <code>: Toolkit</code>. 生命线 (Lifelines): Vertical dashed lines for each object. 消息 (Messages): <ul style="list-style-type: none"> <code>a1: run(3)</code> (Message with activation bar) from <code>t: Thread</code> to <code>: Toolkit</code>. <code>run()</code> (Call) from <code>t: Thread</code> to <code>: Toolkit</code>. <code>callbackLoop()</code> (Return) from <code>: Toolkit</code> to <code>t: Thread</code>. <code>«create» p: Peer</code> (Create) from <code>: Toolkit</code> to <code>p: Peer</code>. <code>handleExpose()</code> (Call) from <code>: Toolkit</code> to <code>p: Peer</code>. <code>«destroy»</code> (Destroy) from <code>: Toolkit</code> to <code>p: Peer</code>. 其他标注: <ul style="list-style-type: none"> 时序标签 (Time Label): <code>a1</code> on the first message. 控制焦点 (Control Focus): A dot on the <code>run()</code> message. 调用 (Call): The <code>run()</code> message. 递归 (Recursion): A self-call on <code>: Toolkit</code> during <code>callbackLoop()</code>. 返回 (Return): The <code>callbackLoop()</code> message. 销毁 (Destroy): The <code>«destroy»</code> message. 创建 (Create): The <code>«create»</code> message.

通信图

语义	描述几个对象的交互通信关系，更关注对象之间的连接关系而非交互顺序。以前也称之为“协作图”
关键元素	对象 (object),
关系	对象之间的连接(Link),对象之间发送的消息(message)
用途	可以描述用户和系统的交互，也可以描述程序对象之间的交互。
图例	<p>The diagram illustrates a collaboration between three objects: c : Client, : Transaction, and p : ODBDProxy. The : Transaction object is labeled as «local» and {transient}. The p : ODBDProxy object is labeled as «global». The interactions are as follows:</p> <ul style="list-style-type: none"> Client to Transaction: A vertical line connects them. Three messages are sent: 1: «create», 2: setActions(a, d, o), and 3: «destroy». Transaction to ODBDProxy: A horizontal line connects them. Two messages are sent: 2.1: setValues(d, 3.4) and 2.2: setValues(a, "CO"). <p>Labels in the diagram: 连接 (Link) points to the vertical line; 对象 (Object) points to the Transaction object; 消息 (Message) points to the horizontal line.</p>

交互概览图

语义	描述几个对象交互场景之间的流程关系
关键元素	交互引用(interaction use),
关系	交互引用之间的转移
用途	可以描述用户和系统的交互，也可以描述程序对象之间的交互。



时间图

语义	描述对象随时间而发生的状态变化，以及几个对象之间的交互
关键元素	对象生命线 (lifeline) 状态条件 (state condition)
关系	对象生命线之间的消息
用途	可以描述用户和系统的按照时间顺序的交互，也可以描述程序对象之间按照时间顺序的交互。

