**构件图Component Diagram**

**定义：**描绘了各种软件构件之间的依赖关系，构件如何连接在一起以形成更大的组件，例如，可执行文件和源文件之间的依赖关系。

**作用：**用于说明任意复杂系统的结构。

**建模技术**：

* 对系统中的组件建模—分解系统，考虑有关系统的组成管理、软件的重用和物理节点的配置等因素，把关系密切的可执行程序和对象分别归入组件，找出相应的类、接口等模型元素。
* 对相应构件提供的接口建模。
* 对构件之间的依赖关系建模。
* 将逻辑设计映射成物理实现。
* 对建模的结果进行精化和细化。

**构件图的组成要素有**：构件(Component) 、 接口(Interface) 、 关系(Relationship) 、 端口(Port) 、 连接器(Connector)

**构件Component：**

**定义：**

* + 构件是定义了良好接口的物理实现单元，是系统中可替换的物理部件。
  + 构件通过它的提供接口和请求接口展现行为。
  + 在UML2.0中，构件是一种类，因此构件具有属性、操作和可见性。这些概念的含义与在类图中定义的是一样的，只是在这里把这些概念应用在构件上。

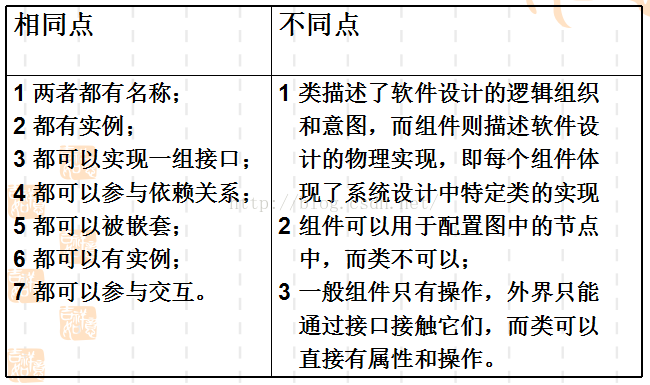
**存在的形式有：**可执行文件，文档，数据库表，文件和库文件

**在UML2.0中：**把构件分为基本构件和包装构件

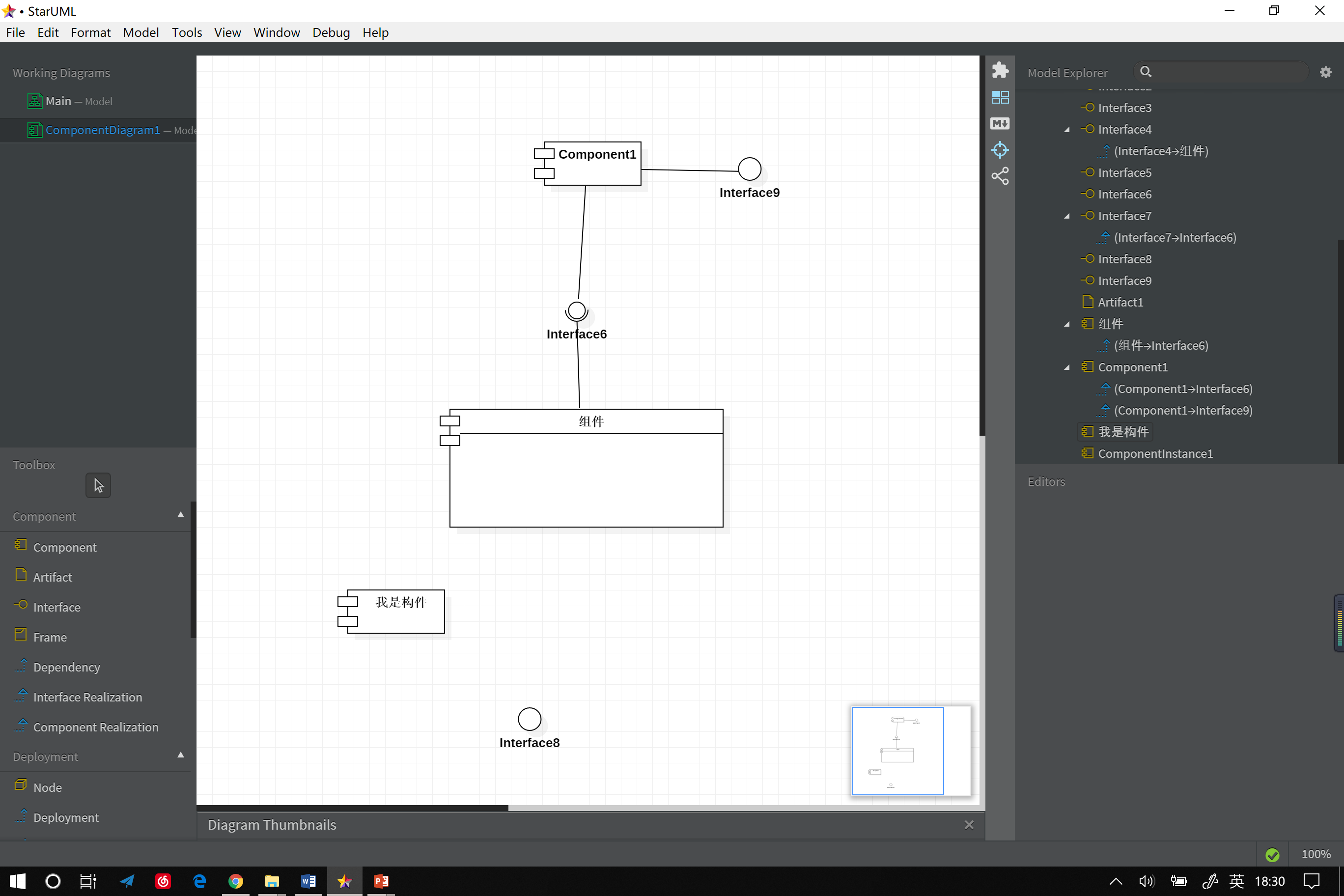
基本构件：注重于把构件定义为在系统中可执行的元素。

包装构件：扩展了基本构件的概念，它注重于把构件定义为一组相关的元素，这组元素为开发过程的一部分。在构件的命名空间中，可以包括类、接口、构件、包、用况、依赖（如映射）和制品。

**构件和类的区别**：



**表示方式：**带有两个标签的矩形



**接口Interface**：

**定义：**接口由一组操作组成，它指定了一个规则，这个规则必须由实现和使用这个接口的构件的所遵循

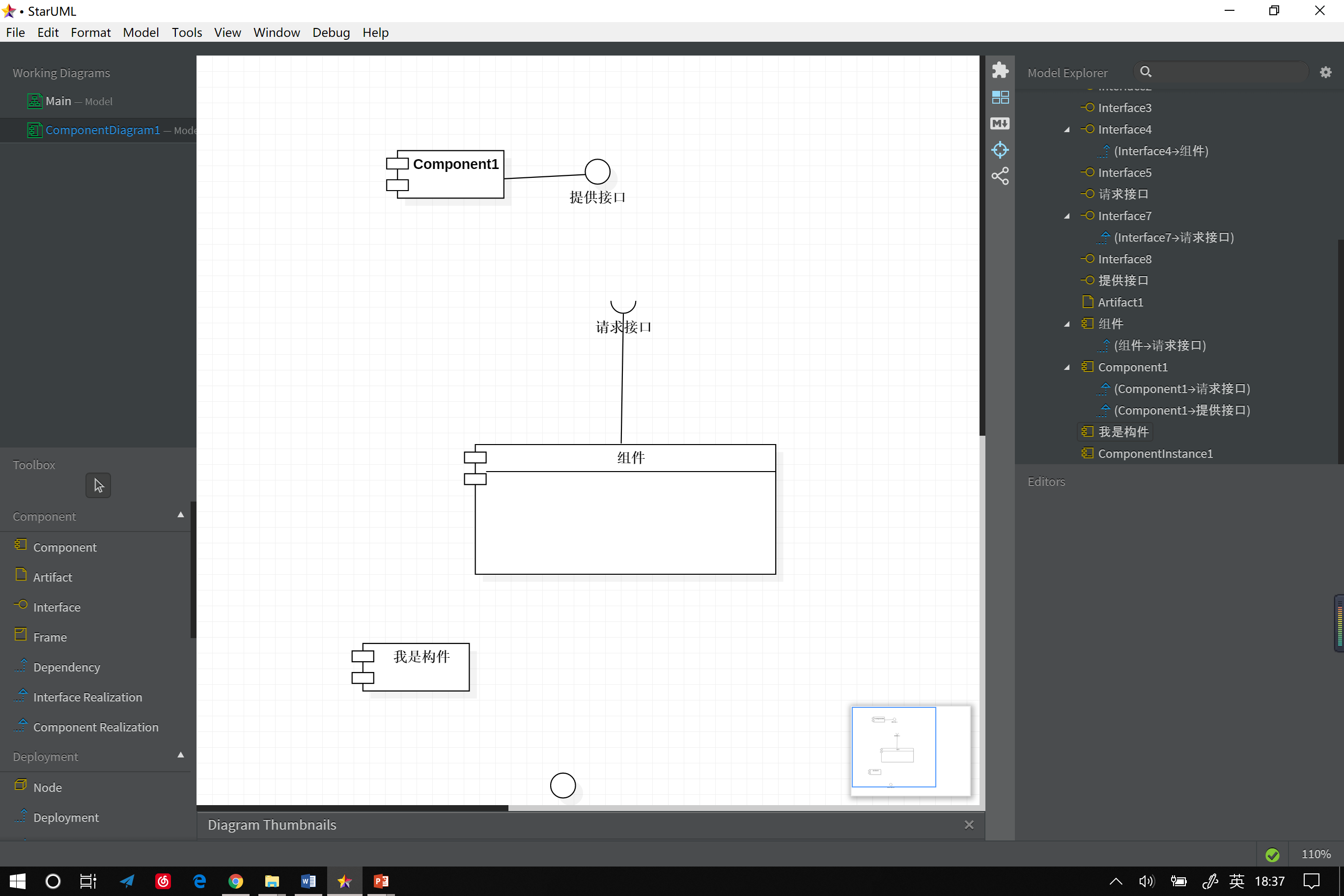
**接口的分类：**提供接口和请求接口

把构件实现的接口称为提供接口，这意味着构件的提供接口是给其它构件提供服务的。实现接口的构件支持由该接口所拥有的特征，包括接口拥有的约束。

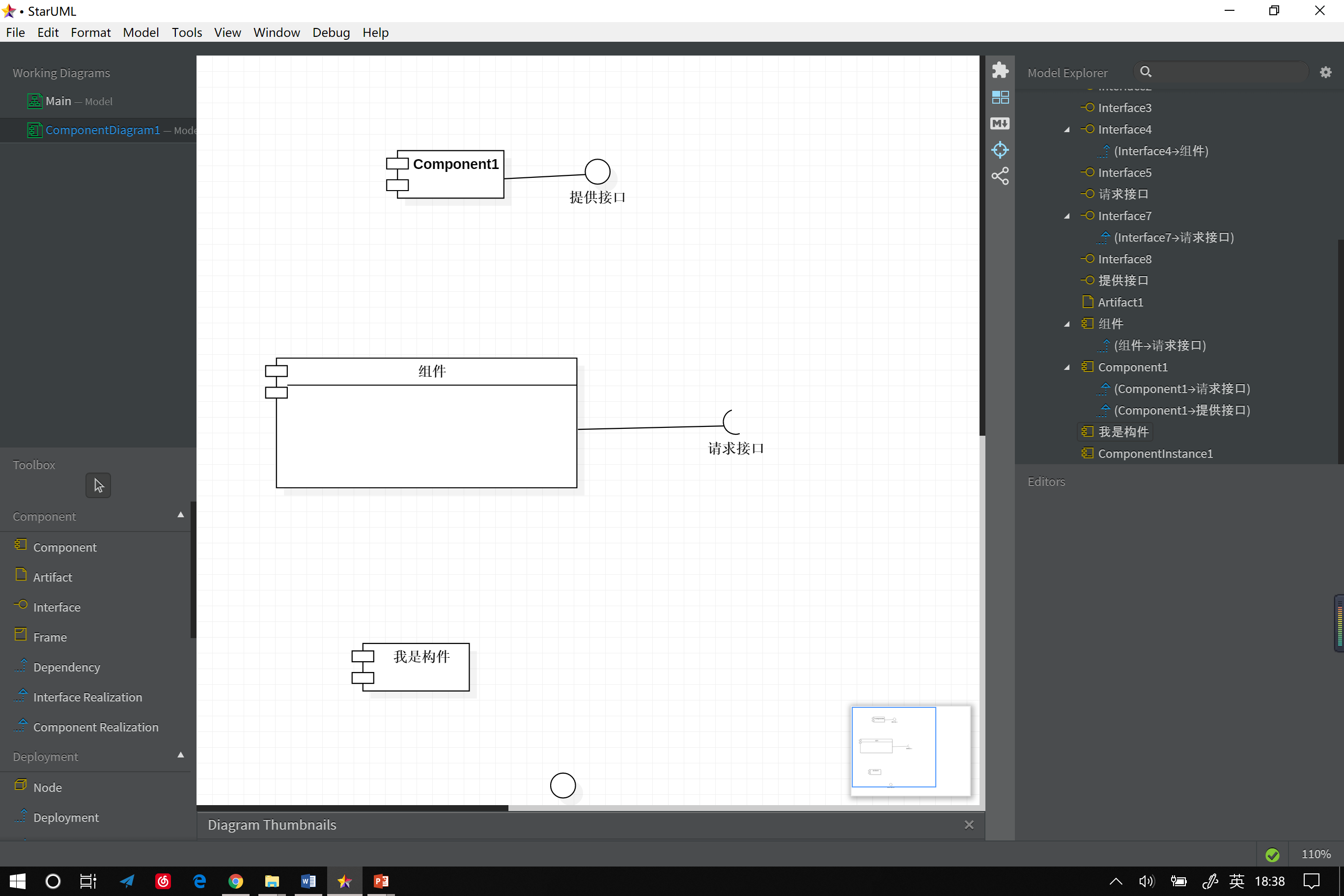
构件使用的接口被称为请求接口，即构件向其它构件请求服务时要遵循的接口

**表现方式：**

提供接口：一个封闭的圆形与一条直线组成



请求接口：一个半圆与一条直线组成

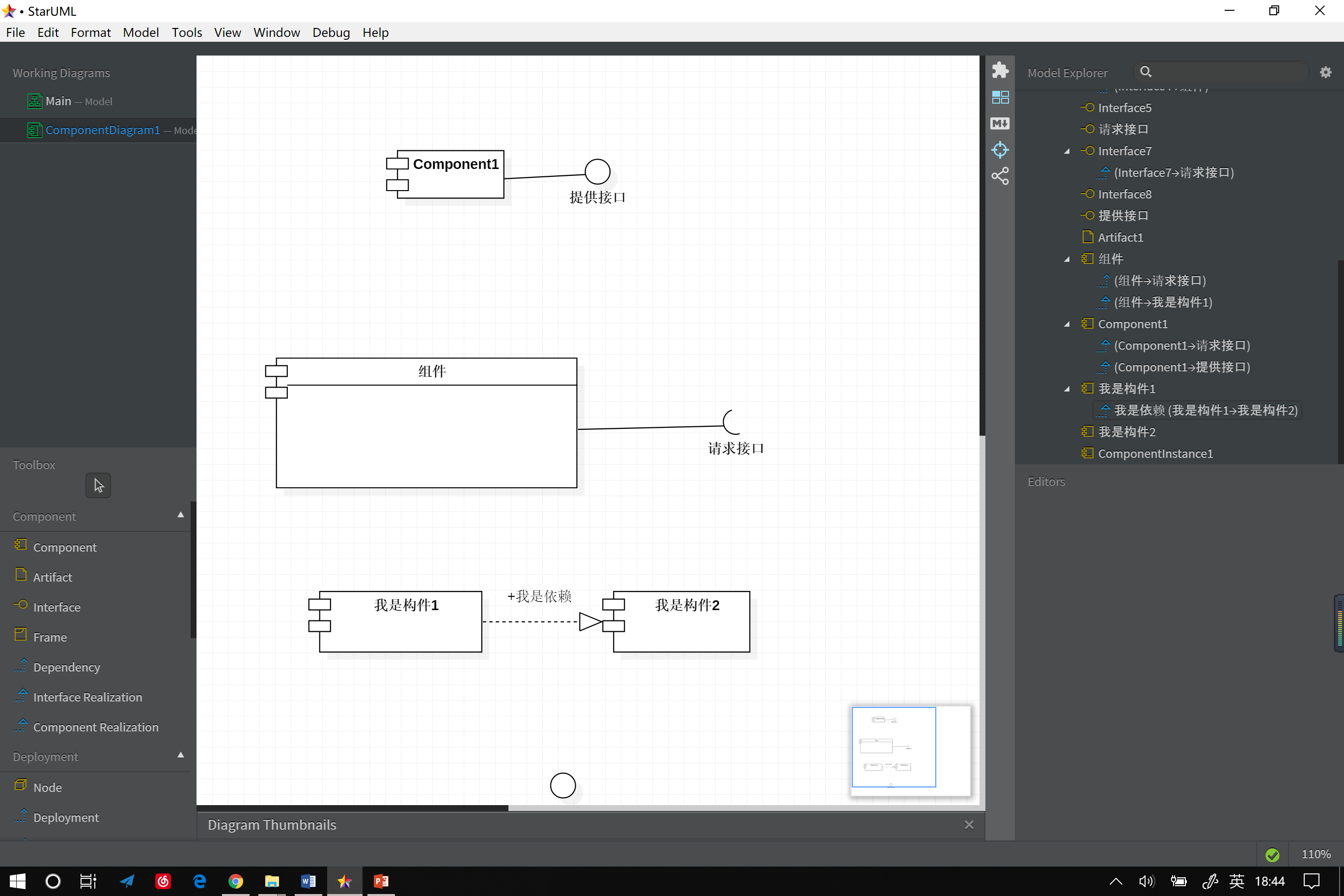


**关系Relationship**：

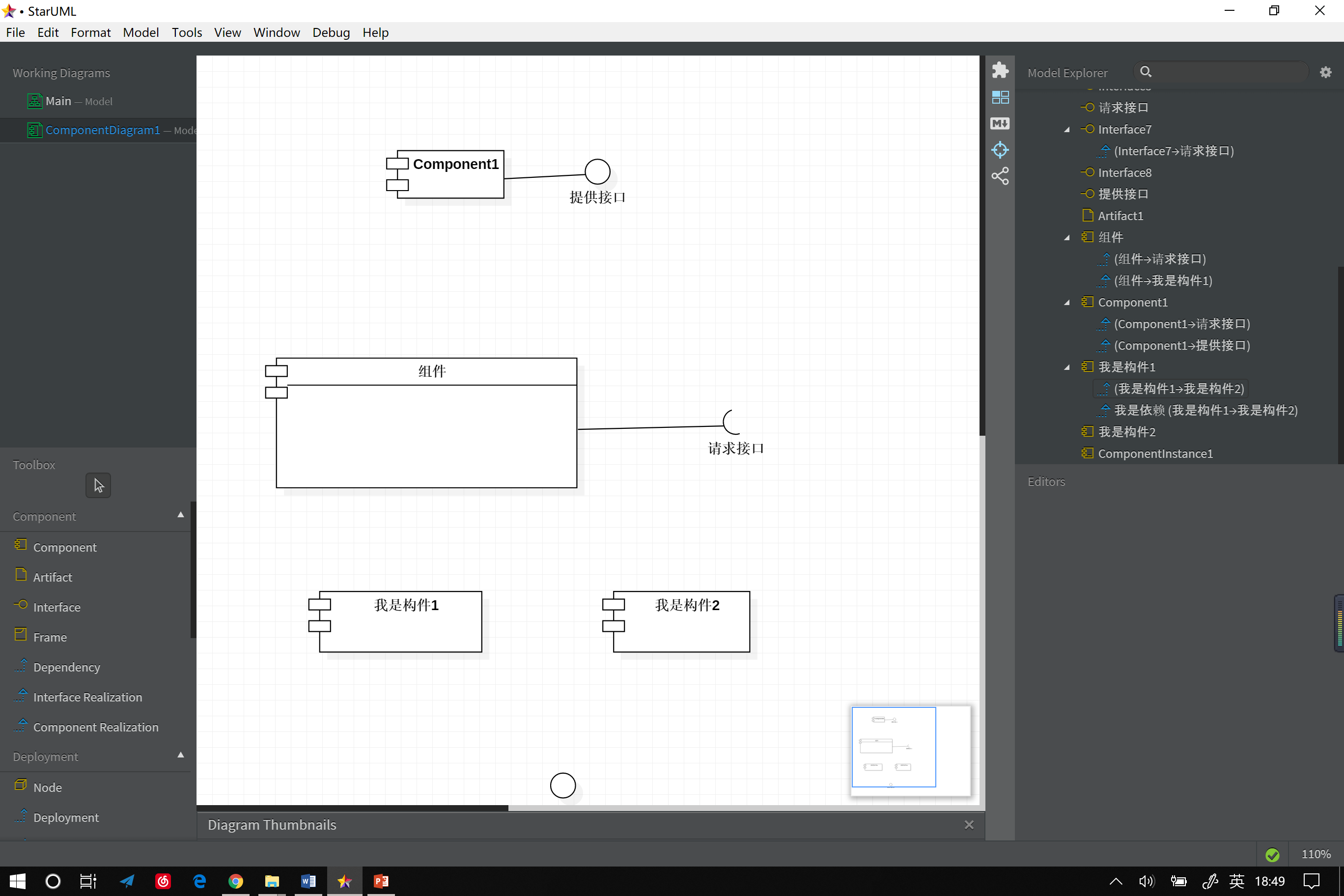
定义：事物之间的联系

表现方式：

依赖：构件依赖外部提供的服务，虚线表示，多用于构件之间



实现：构件向外提供的服务，实线表示，多用于构件和接口



**端口Port**：

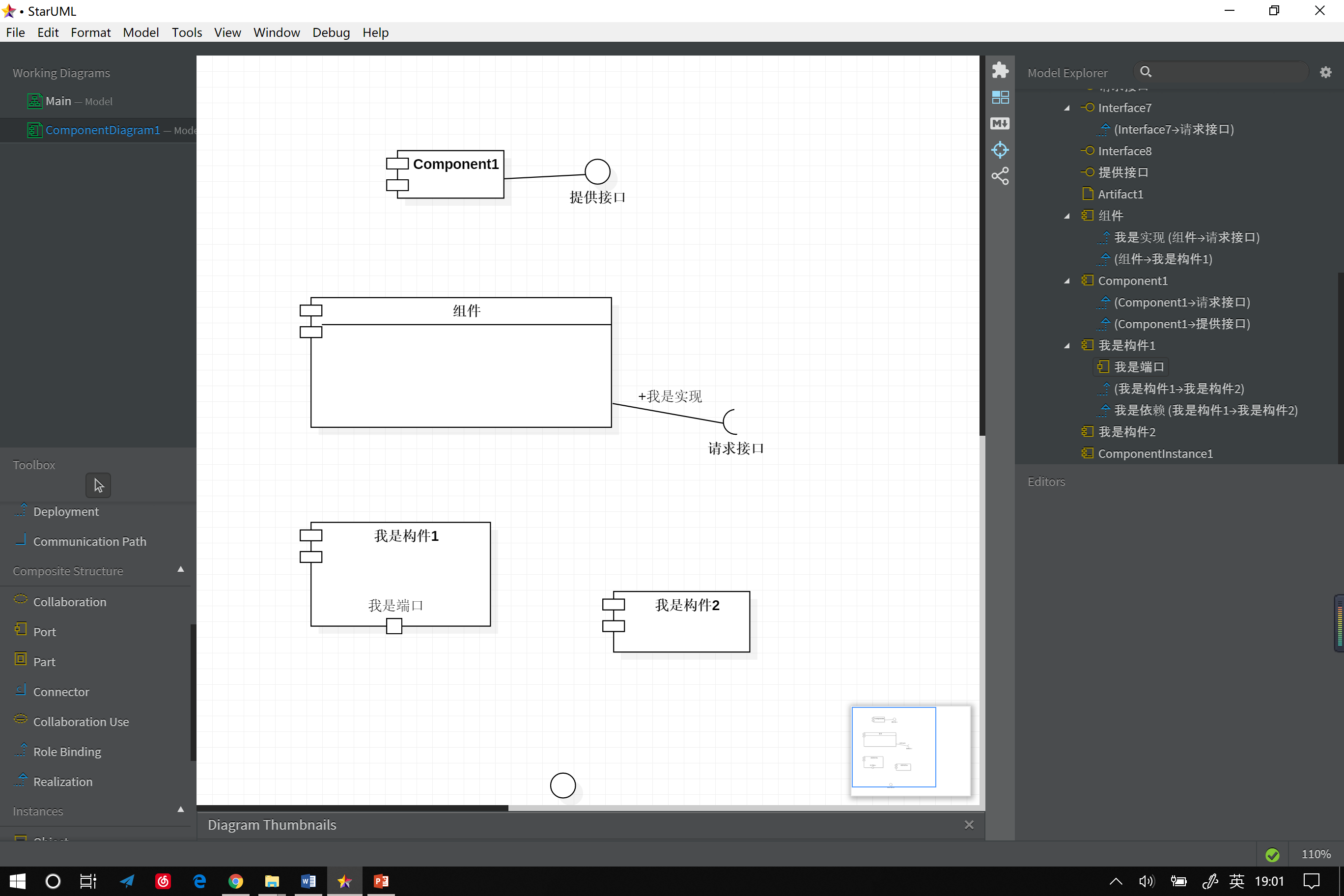
**定义：**

* 端口是UML2.0引入的概念
* 端口描述了在构件与它的环境之间以及在构件与它的内部构件之间的一个显示地交互点
* 端口是一个封装构件的显示的对外窗口，所有进出构件的交互都要通过端口
* 使用端口能在更大的程度上增加构件的封装性和可替代性。
* 端口是构件的一部分

**接口与端口的关系**：

* 提供接口说明了通过端口来提供服务，请求接口说明了通过端口需要从其它构件获得服务。
* 一个构件可以通过一个特定端口同另一个构件通讯，而且通讯完全是通过由端口支持的接口来描述的。

**表示方法**：一个在正方形



**连接器Connector**：

**定义：**连接线是用来连接两个或者多个实例使他们直接能够进行交流协作。主要用来连接两个端口之间的交流

**分类：**

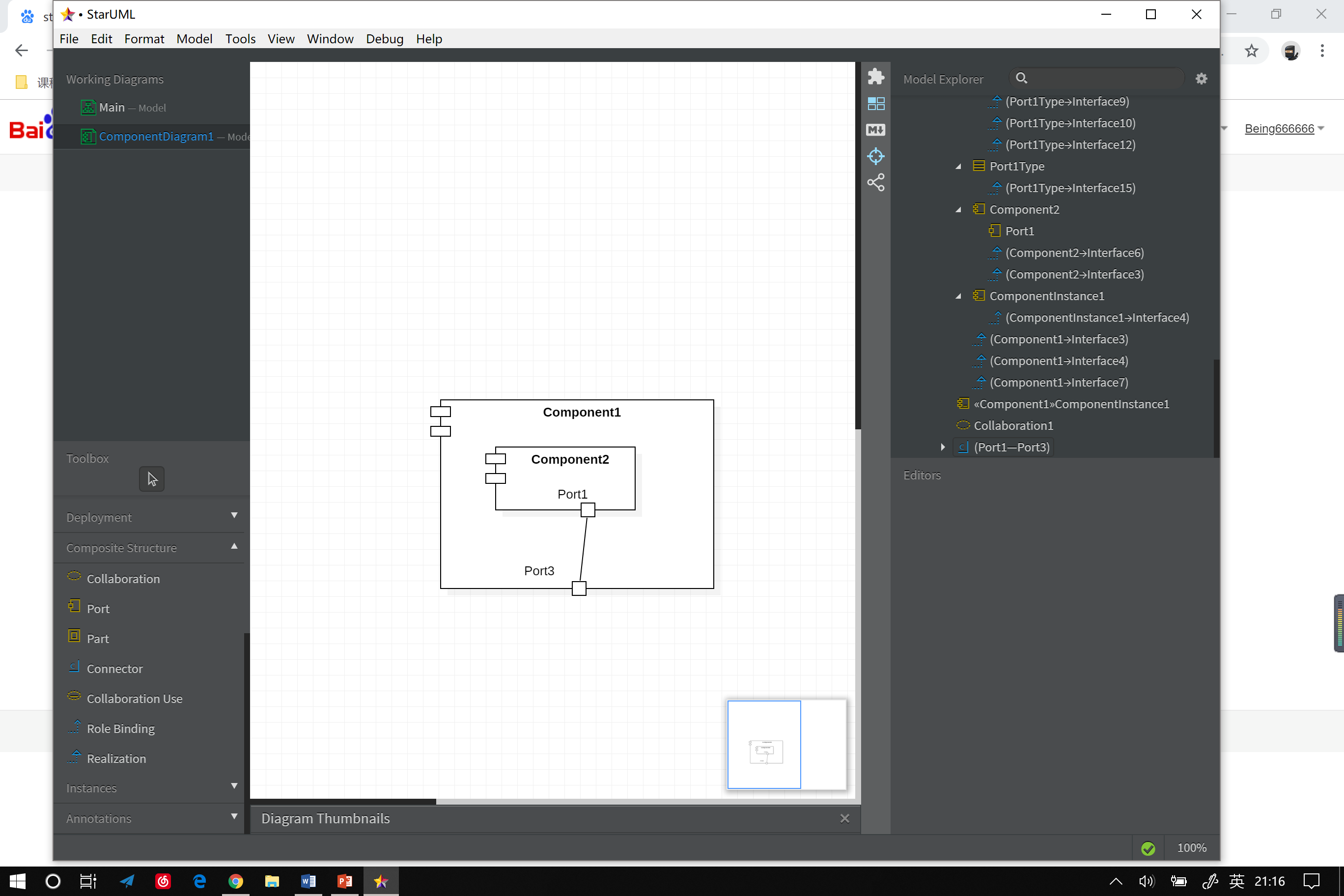
UML2.0提供两种类型的连接器：

委托连接器（Delegation Connector）：连接外部接口的端口和内部接口。

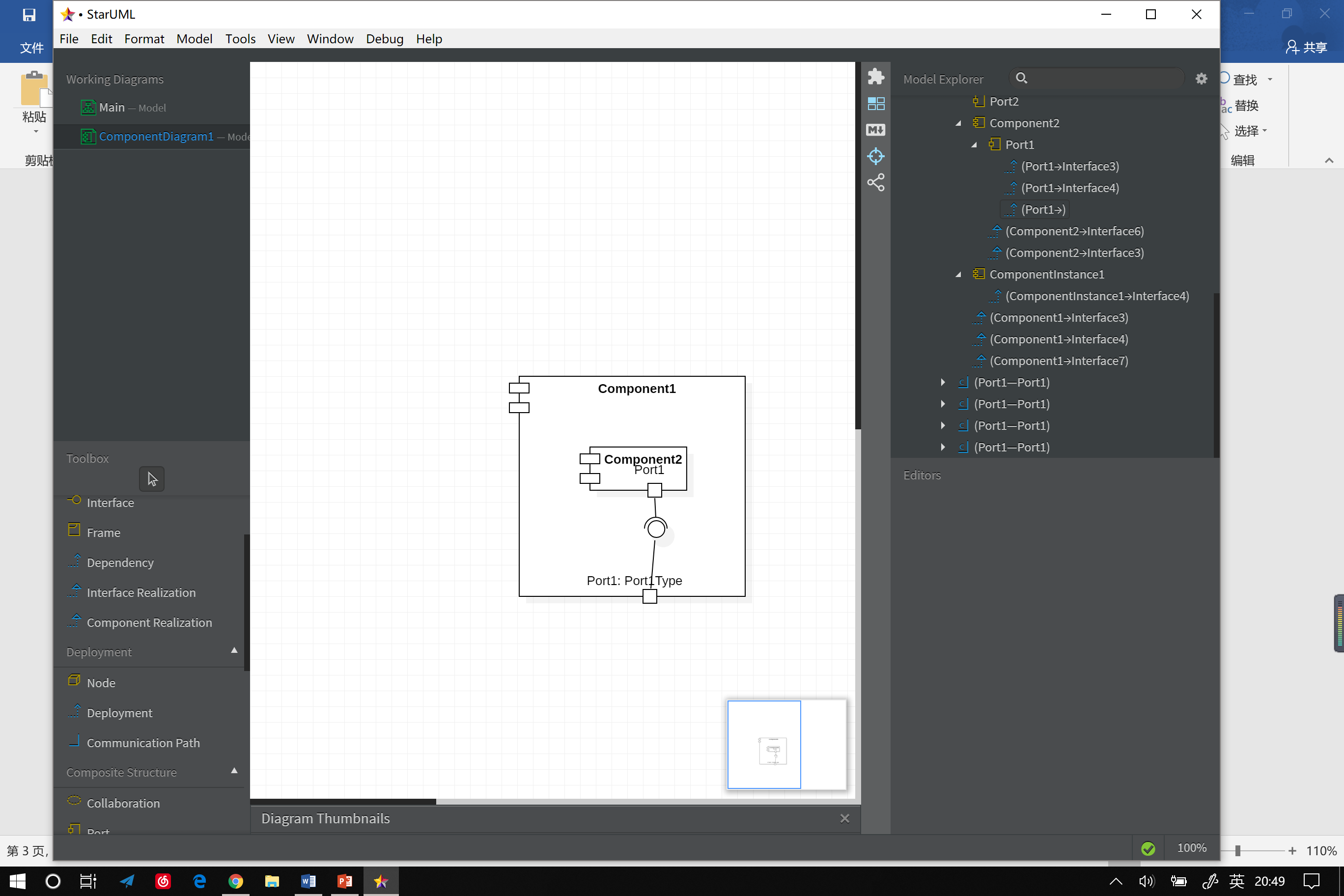
组装连接器（Assembly Connector）：表示构件之间的关系，它连接构件内部的类，将一个构件的提供接口和一个构件的请求接口捆绑在一起

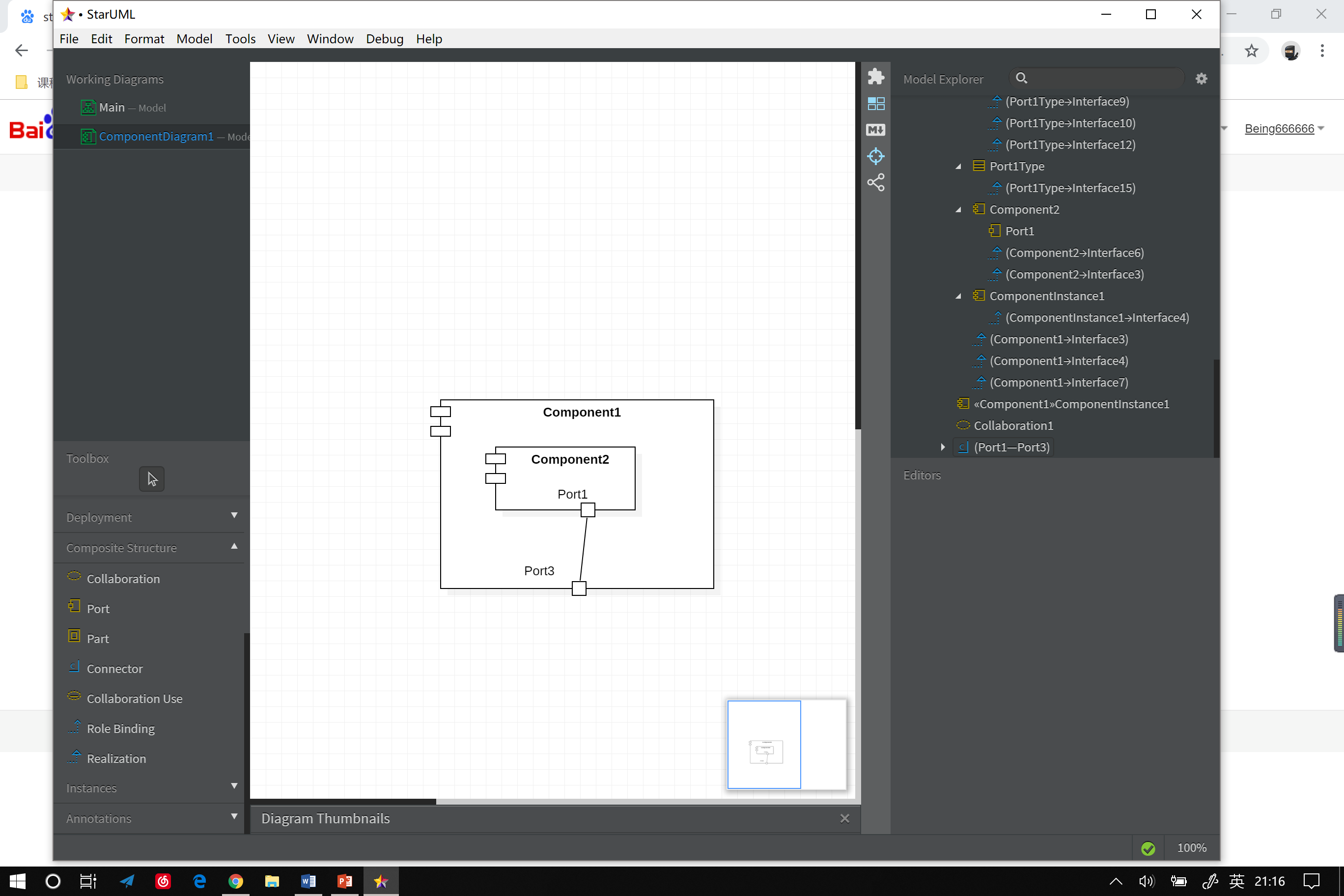
**表现方式**：

委托连接器

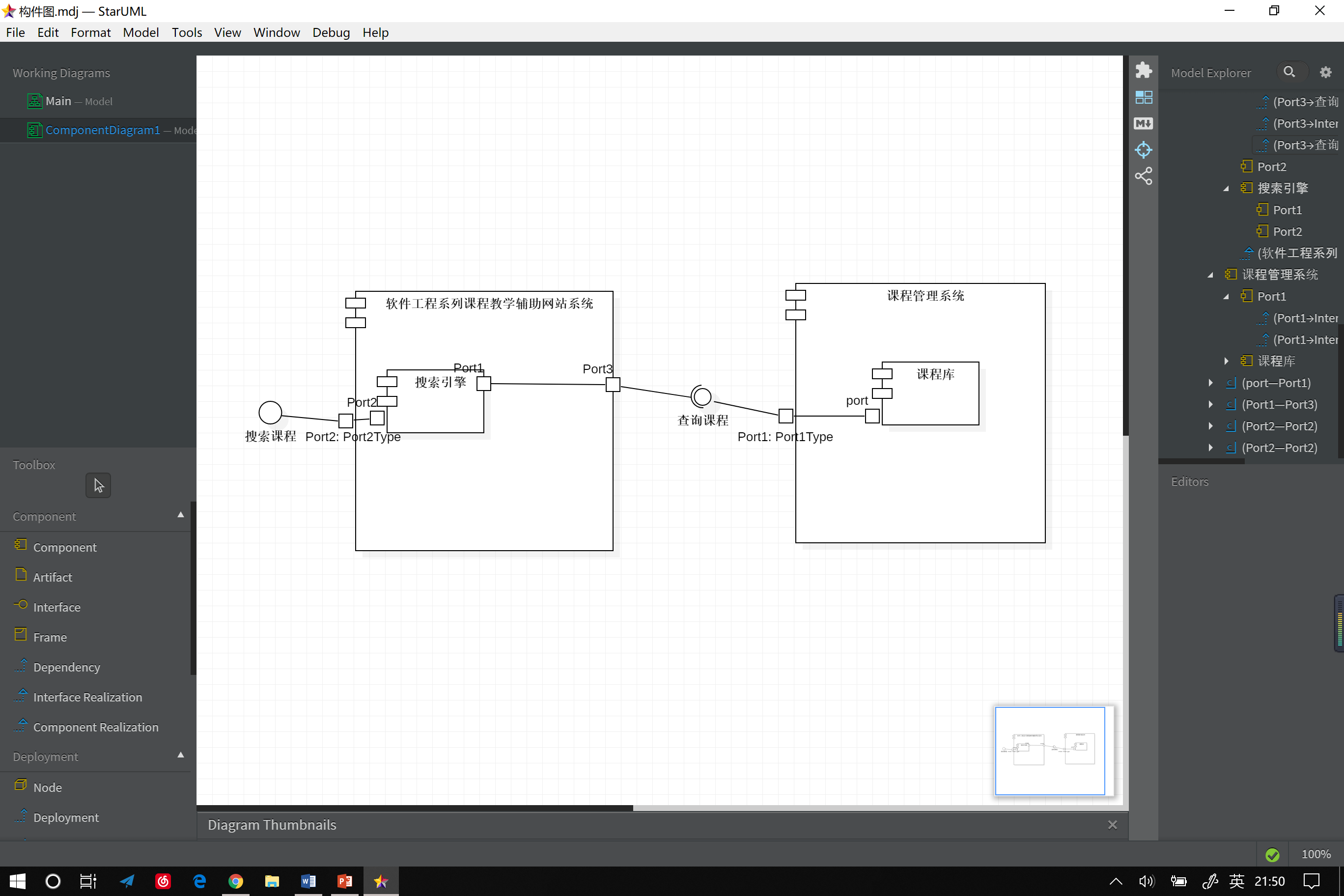


组装连接器





这个不知道对不对



参考资料：

<https://en.wikipedia.org/wiki/Component_diagram>

<https://blog.csdn.net/fanxiaobin577328725/article/details/51647248>

https://www.jianshu.com/p/a0704aa2b092