



构件图与部署图

内容

- 构件和构件图概述
- 构件图中的元素
- 构件与类的比较
- 构件图的作用
- 构件图建模技术
- 部署图概述
- 部署图中的基本概念
- 节点与构件的比较
- 部署图建模技术
- 小结

- 实现图显示系统实现时的一些特性，包括构件图和部署图。

- 构件图显示代码本身的逻辑结构
- 部署图显示系统运行时刻的结构

构件和构件图概述

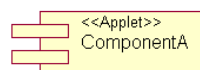
- 构件(Component)是系统中遵从一组接口且提供其实现的物理的、可替换的部分。每个构件体现了系统设计中特定类的实现。
- 构件图(Component diagram)显示一组构件以及它们之间的相互关系，包括编译、链接或执行时构件之间的依赖关系。
- 构件图是对OO系统物理方面建模的2个图之一。

构件的类型

- 实施构件，这类构件是构成一个可执行系统必要和充分的构件，例如动态链接库、可执行文件，另外还包括如COM+、CORBA及企业级Java Beans、动态Web页、数据库表等。
- 工作产品构件，这类构件主要是开发过程的产物，包括创建实施构件的源代码文件及数据文件。这些构件可以产生实施构件。
- 执行构件，作为一个正在执行的系统的结果而被创建的。

构件的类型

- 有一些构件的图标表示形式和标准构件图形表示形式相同，它们包括ActiveX、Applet、Application、DLL、EXE以及自定义构造型的构件，它们的表示形式是在构件上添加相关的构造型，下图是一个构造型为Applet的构件。



构件的类型

- 在Rational Rose 2003中，数据库也被认为是一种构件。
- 虚包是一种只包含对其它包所具有的元素进行的引用的构件。它被用来提供一个包的某些内容的公共视图。虚包不包含任何它自己的模型元素。



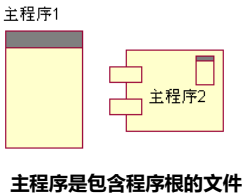
构件图中的元素

(1) 构件



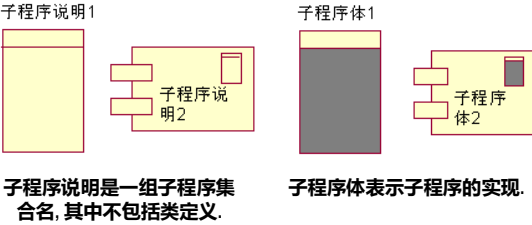
构件图中的元素

(1) 构件 Rose中的扩展构件



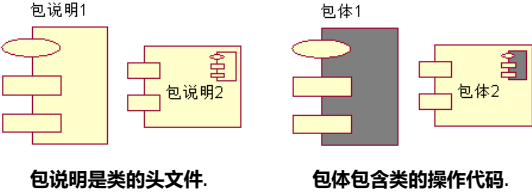
构件图中的元素

(1) 构件 Rose中的扩展构件



构件图中的元素

(1) 构件 Rose中的扩展构件



构件图中的元素

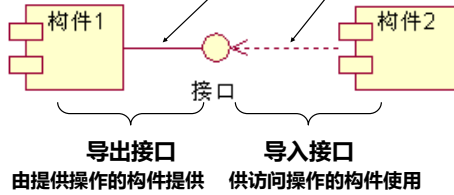
(2) 接口

- 构件具有它们支持的接口和需要从其他构件得到的接口。
- 接口是被软件或硬件所支持的一个操作集。通过使用命名的接口，可以避免在系统中各个构件之间直接发生依赖关系

构件图中的元素

(2) 接口

构件与接口之间的关系: **实现关系和依赖关系**



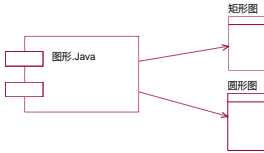
接口实例

- 将整个“在线酒店预订子系统”作为一个构件，考虑其对外接口。显然它首先需要提供用户界面；其次还需要与加盟的酒店系统连接，完成预订工作。



构件与类

- 构件是一组逻辑元素的物理实现。构件包含类，类通过构件来实现，构件与类之间是依赖关系。

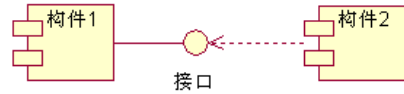


构件图中的元素

(3) 依赖关系



良好定义的构件不直接依赖于其他构件而依赖于构件所支持的接口。在这种情况下，系统中的一个构件可以被支持正确接口的其他构件所替代。



构件与类的显著不同点 (1)

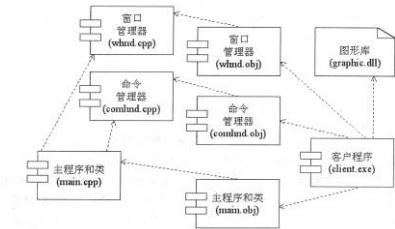
- 抽象的方式不同：
 - 构件：是程序代码的物理抽象；构件可以驻留在节点上。
 - 类：是逻辑抽象；不能单独存在于节点上。
- 抽象的级别不同
 - 构件：表示一个物理模块，构件可以包含多个类；构件依赖它所包含的类
 - 类：表示一个逻辑模块，只能从属于某个构件，类通过构件来实现。

构件与类的显著不同点 (2)

- 访问方式不同
 - 构件：不直接拥有属性和操作，只能通过接口访问其操作
 - 类：直接拥有自己的属性和操作，可以直接访问其操作
- 与包的关系不同
 - 构件：包可以包含成组的逻辑模型元素，也可以包含物理的构件
 - 类：一个类可以出现在多个构件中，却只能在一个包内定义。

构件图实例1

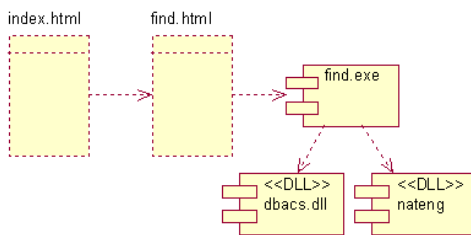
- 一个大型复杂的软件系统，一般由多个可执行程序和相关持久对象库构成。
- 下图描述了一个客户的软件系统从 C++ 源代码构件编译成相应的二进制代码（目标码）构件，最后通过连接成为可执行程序构件的全过程。



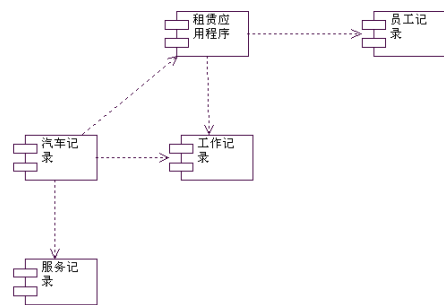
构件图实例1

- 在图中，3 个源代码构件 “main.cpp” 构件、“comhnd.cpp” 构件和 “whnd.cpp” 构件构成一个完整的客户软件源代码程序。在这 3 个构件中，源代码构件 “main.cpp” 构件依赖（调用）“comhnd.cpp” 构件和 “whnd.cpp” 构件才能达到系统要求的完整功能。
- 3 个源代码构件分别经过编译，产生相应的 3 个二进制代码构件：“main.obj” 构件、“comhnd.obj” 构件和 “whnd.obj” 构件，这 3 个二进制代码构件依赖于对应的 3 个源代码构件才能产生。
- 这 3 个二进制代码构件经过连接，形成一个可执行构件 “client.exe”。可执行构件 “client.exe” 的产生依赖于对应的 3 个二进制代码构件；
- 同时，它在执行过程中还要依赖和调用动态链接库 “graphic.dll” 才能完成系统要求的功能。动态链接库用注释节点表示。

构件图实例2



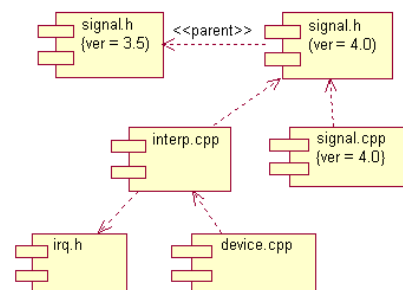
构件图实例3



构件图的作用:源代码建模

- 识别出感兴趣的相关源代码文件的集合，并把它们建模为构件；
- 对于较大的系统，利用包来进行分组；
- 通过约束来表示源代码的版本号、作者和最后修改日期等信息；
- 用依赖关系来表示这些文件间编译的依赖关系。

构件图的作用：对源代码文件之间的关系建模

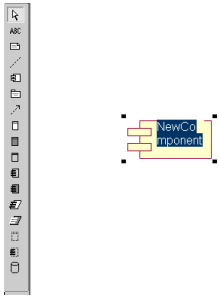


使用Rose创建构件图

- 创建一个新的构件图，可以通过以下方式进行。
 - (1) 右键单击浏览器中的Component View（构件视图）或者位于构件视图下的包。
 - (2) 在弹出的菜单中，选中“New”（新建）下的“Component Diagram”（构件图）选项。
 - (3) 输入新的构件图名称。
 - (4) 双击打开浏览器中的构件图。

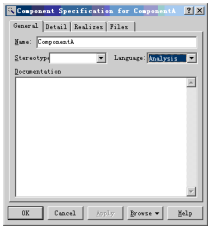
使用Rose创建构件图

- 通过构件图的图形编辑工具栏添加构件的步骤如下：
 - (1) 在构件图的图形编辑工具栏中，选择构件按钮，此时光标变为“+”号。
 - (2) 在构件图图形编辑区内选择任意一个位置，然后使用鼠标左键单击，系统在该位置创建一个新的构件。
 - (3) 在构件的名称栏中，输入构件的名称。



使用Rose创建构件图

- 对于构件图中的构件，和其它Rational Rose 2003 中的模型元素一样，我们可以通过构件的标准规范窗口设置增加其细节信息，包括名称、构造型、语言、文本、声明、实现类和关联文件等。



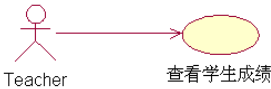
构件图建模技术

- 我们使用下列的步骤创建构件图：
 - (1) 根据用例或场景的确定需求，确定系统的构件。
 - (2) 将系统中的类、接口等逻辑元素映射到构件中。
 - (3) 确定构件之间的依赖关系，并对构件进行细化。

用Rose部署一个实际的项目

1. 确定需求用例

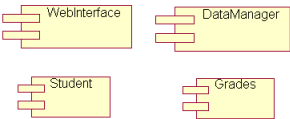
- 我们将在在序列图中介绍的一个学生信息管理系统的简单用例为例，介绍如何去创建系统的构件图。



用Rose部署一个实际的项目

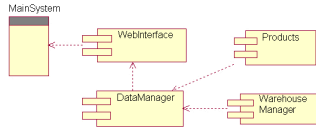
2. 创建构件图

- 我们可以和确定用例中的类和对象一样，根据用例的流程确定系统的构件。根据上面的用例，我们可以确定最明显的二个实体类是学生类（Student）和班级类（Grades）。系统的操作界面（WebInterface）是一个边界类。还有，数据访问操作过程中离不开与数据库交互的数据库管理类（DataManager）。除此以外，我们还需要一个系统的主程序（MainSystem），用来表示整个系统的启动入口。



用Rose部署一个实际的项目

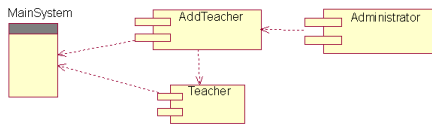
- 第二步，将系统中的类和接口等映射到构件中。一个构件不仅仅包含一个类或接口，可以包含几个类或接口。
- 第三步，确定构件之间的依赖关系，并对构件进行细化。细化的内容包括指定构件的实现语言、构件的构造型、编程语言的设置以及针对某种编程语言的特殊设置，如Java语言中的导入文件、标准、版权和文档等。



练习题

在“远程网络教学系统”中，以系统管理员添加教师信息用例为例，我们可以确定“Administrator”、“Teacher”、“AddTeacher”等类，根据这些类创建关于系统管理员添加教师信息的相关构件图。

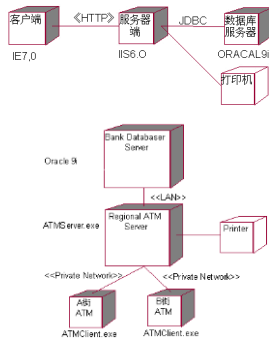
练习题



部署图概述

- 部署图(Deployment diagram)也称配置图、实施图，是对OO系统进行物理建模的图。
- 部署图用来显示系统中计算节点的拓扑结构和通信路径与节点上运行的软件件等。
- 一个系统只有一个部署图，部署图通常用于理解分布式系统。
- 在一个部署图中，包含了两种基本的模型元素：节点和节点之间的连接。
- 部署图由体系工程师、网络工程师、系统工程师等描述。

部署图概述

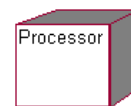


部署图中的基本概念

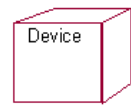
(1) 节点(node)

节点是运行时代表计算资源的物理元素，节点通常有内存及处理能力。它可以是物理设备及运行在该设备上的软件系统。

节点有2种类型：



处理机



设备

处理器与设备

- 处理器是能够执行软件、具有计算能力的节点。
- 设备是没有计算能力的节点，通常情况下都是通过其接口为外部提供某种服务。

节点中的部署

- 部署图可以将节点和构件结合起来，以建模处理资源和软件实现之间的关系。
- 当构件驻留在某个节点时，可以将它建模在图上该节点的内部。
- 为显示构件之间的逻辑通信，需要添加一条表示依赖关系的虚线箭头。

部署图中的基本概念

(2) 连接(connection)

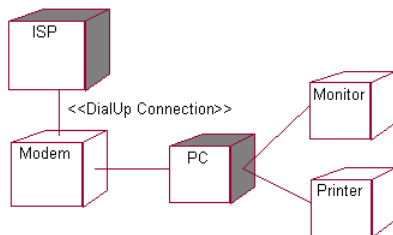
连接表示系统之间进行交互的通信路径,它表示2个硬件之间的关联关系.可以有角色、多重性、约束、版型等.



节点与构件的比较

相同点	不同点
二者都有名称和关系；都可以有实例；都可以被嵌套；都可以参与交互。	构件是参与系统执行的事物，而节点是执行构件的事物；构件表示逻辑元素的物理包装，而节点表示构件的物理配置。

部署图的例子



PC和外设与ISP连接的部署图

使用Rose创建部署图

- 在每一个系统模型中，只存在一个部署图。在使用 Rational Rose 2003创建系统模型时，就已经创建完毕，即为Deployment View（部署视图）。如果要访问部署图，在浏览器中双击该部署视图即可。
- 其中包括：
 - (1)创建和删除节点。
 - (2)设置节点。
 - (3)添加和删除节点之间的连接。
 - (4)设置连接规范。

用Rose部署一个实际的项目

创建部署图

- 我们可以使用下列的步骤创建部署图：
 - 根据系统的物理需求，确定系统的节点。
 - 根据节点之间的物理连接，将节点连接起来。
 - 通过添加处理器的进程、描述连接的类型等细化对部署图的表示。
- 建模一个简单的学生信息管理系统，该系统的需求如下所示：
 - 学生或教师可以在客户的PC机上通过浏览器，如IE6.0等，查看系统页面，与Web服务器通信。
 - Web服务器安装Web服务器软件，如Tomcat等，通过JDBC与数据库服务器连接。
 - 数据库服务器中安装SQL Server 2000，提供数据服务功能。

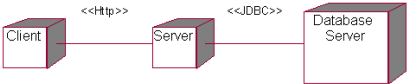
用Rose部署一个实际的项目

1. 确定系统节点。



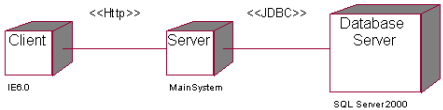
2. 添加节点连接，我们可以从上面的需求列表获取下列的连接信息：

- 客户的PC机上通过Http协议与Web服务器通信。
- Web服务器通过JDBC与数据库服务器连接。



用Rose部署一个实际的项目

3. 细化部署图，接下来需要确定各个处理器中的主程序以及其它的内容，如构造型、说明型文档和特征描述等。

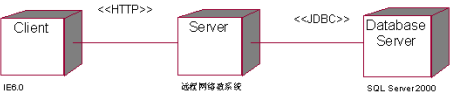


练习题

在“远程网络教学系统”中，该系统的需求如下所示：

- 学生或教师可以在客户的PC机上通过浏览器，如IE6.0等，登录到远程网络教学系统中。
- 在Web服务器端，我们安装Web服务器软件，如Tomcat等，部署远程网络教学系统，并通过JDBC与数据库服务器连接。
- 数据库服务器中使用SQL Server 2000提供数据服务。
- 根据以上的系统需求，创建系统的部署图。

练习题



练习题

- 根据以下描述，试完成一个“销售管理系统”的部署图。该系统采用的硬件设备有：
 - 3台联想PC6000作为终端机，分别用作销售管理、仓库管理和合同管理。
 - 1台联想PC6000作为网络数据库服务器。
 - 2台Ls1000打印机。一台为网络共享、一台为仓库管理专用。
 - 客户机之间采用《TCP/IP》通信协议，3个客户机与服务器之间采用《Ethernet》网连接。

