

为了能够在CORBA系统使用那些不是为CORBA编写的对象，可以为每个对象装备一个对象适配器（object adapter）。对象适配器是一种包装器，它处理诸如登记对象、生成对象引用以及激发一个在被调用时处于未活动状态的对象等琐碎事务。所有这些与CORBA有关部分的布局如图8-39所示。

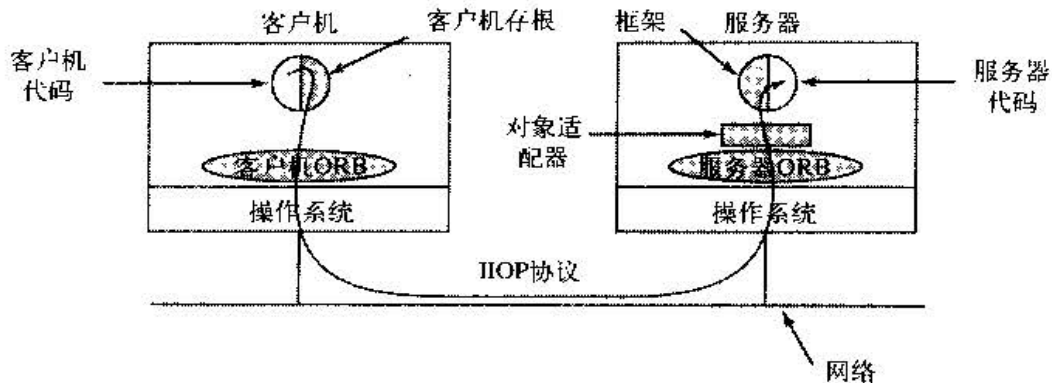


图8-39 基于CORBA的分布式系统中的主要元素（CORBA部件由灰色表示）

对于CORBA而言，一个严重问题是每个CORBA对象只存在一个服务器上，这意味着那些在世界各地客户机上被大量使用的对象，会有很差的性能。在实践中，CORBA只在小规模系统中才能有效工作，比如，在一台计算机、一个局域网或者一个公司中用来连接进程。

8.4.6 基于协作的中间件

分布式系统的最后一个范型是所谓基于协作的中间件（coordination-based middleware）。我们将从Linda系统开始，这是一个开启了该领域的学术性研究项目。然后考察主要由该项目所激发的两个商业案例：pubilsh/subscribe以及Jini。

1. Linda

Linda 是一个由耶鲁大学的David Gelernter和他的学生Nick Carriero（Carriero与Gelernter，1986；Carriero与Gelernter，1985）研发的用于通信和同步的新系统。在Linda系统中，相互独立的进程之间通过一个抽象的元组空间（tuple space）进行通信。对整个系统而言，元组空间是全局性的，在任何机器上的进程都可以把元组插入或移出元组空间，而不用考虑它们是如何存放的以及存放在何处。对于用户而言，元组空间像一个巨大的全局共享存储器，如同我们前面已经看到的（见图8-21c）各种类似的形式。

一个元组类似于C语言或者Java中的结构。它包括一个或多个域，每个域是一个由基语言（base language）（通过在已有的语言，如C语言中添加一个库，可以实现Linda）所支持的某种类型的值。对于C-Linda，域的类型包括整数、长整数、浮点数以及诸如数组（包括字符串）和结构（但是不含有其他的元组）之类的组合类型。与对象不同，元组是纯粹的数据；它们没有任何相关联的方法。在图8-40中给出了三个元组的示例。

```
("abc", 2, 5)
("matrix-1", 1, 6, 3.14)
("family", "is-sister", "Stephany", "Roberta")
```

图8-40 三个Linda的元组

在元组上存在四种操作。第一种out，将一个元组放入元组空间中。例如，

```
out("abc", 2, 5);
```

该操作将元组("abc", 2, 5)放入到元组空间中。out的域通常是常数、变量或者是表达式，例如

```
out("matrix-1", i, j, 3.14);
```

输出一个带有四个域的元组，其中的第二个域和第三个域由变量*i*和*j*的当前值所决定。

通过使用in原语可以从元组空间中获取元组。该原语通过内容而不是名称或者地址寻找元组。in的域可以是表达式或者形式参数。例如，考虑

```
in("abc", 2, ?i);
```

这个操作在元组空间中“查询”包含字符串“abc”、整数2以及在第三个域中含有任意整数（假设*i*是整数）的元组。如果发现了，则将该元组从元组空间中移出，并且把第三个域的值赋予变量*i*。这种匹配