

## 1.8 UML 对象模型图阅读指南

ArcInfo 的使用方式由两种：其一便是直接使用 ArcGIS 桌面应用程序，ArcMap 和 ArcCatalog，进行 GIS 操作；其二，可以使用其中提供的框架构建开发自己的 GIS 应用程序。

数据建模人员需要站在以上比这两种方式更高的位置来完成自己的工作。用户大多数情况下都会直接使用预先提供的 ArcGIS 桌面应用程序创建 Geodatabase。当然有时候也会需要你进行一些代码定制，特别是在专业化程度很高的复杂数据模型的模拟上，代码的工作更是不可或缺。

本书的目标之一便是介绍数据建模的概念，这样的数据建模不仅在 ArcInfo 的应用程序中适用，还在 ArcInfo 的软件组件中使用，这个软件组件便是我们熟悉的所谓的 ArcObjects。

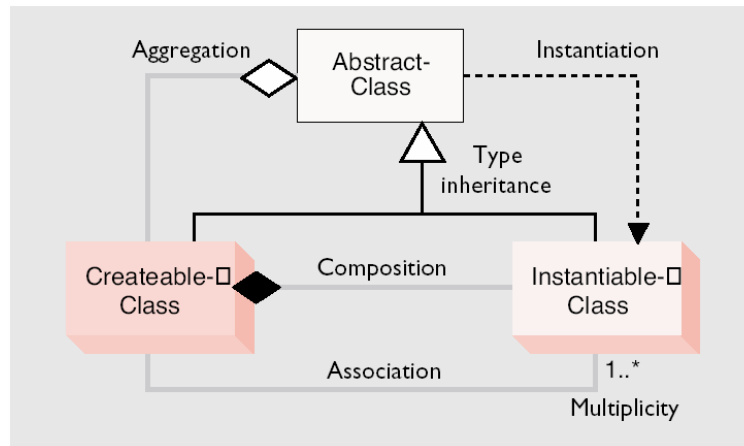
本书中，从使用 ArcInfo 软件时遇到的概念说起，然后再使用对象模型图来表现 ArcInfo 中的这些概念，以利于读者理解。

比如，首先从一个 ArcCatalog 用户的角度来讨论 geodatabase、要素数据集和要素类的结构。接着从程序员的角度用模型图的相关部分介绍 geodatabase 数据的访问对象。

这两种视图模式有相似之处，但是也有很大差别。应用程序的用户界面有时隐藏了对于程序员来说非常重要的一些软件组件细节。本书的目标是让你能够站到比用户和开发者这两个角度更高的位置处理数据建模的工作。

### 1.8.1 解读类模型图

下面图示中，便是贯穿本书的对象模型图的钥匙。



这些符号是基于 UML 画图工具创建的，UML 符号是面向对象分析和设计的工业图样标准。

对象模型图中提供的信息非常多，是对象浏览器中信息的重要补充。Visual Basic，或者其它的开发环境，都会列出所有的类和成员，但不会指明这些类之间的关系。所以，对象模型图是非常有利于读者对 ArcInfo 组件的理解的！

本书使用 UML 来描述 ArcInfo 组件，即 ArcObjects，并描述你能够创建的数据模型。以下详细说明。

### 1.8.2 类和对象

在 UML 图中有三种类型的类：抽象类（abstract class）、可创建类（createable class）与可实例化类（instantiable class）。

抽象类不能用以创建新对象，但可以用来指定子类。举个例子，“line”（线）是“primary line”（干线）和“secondary line”（副干线）的抽象类。

可创建类指的是那些你能够直接使用开发环境中的对象定义语法来创建对象。比如在 Visual Basic 中是这样书写：Dim As New <object> 或者 CreateObject <object>。

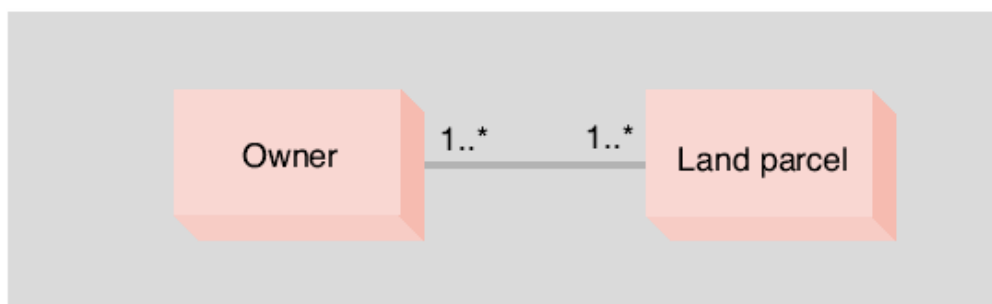
可实例化类不能够直接创建新对象，但是这种类的对象能够作为其它对象的属性被创建或是从其它类的方法中创建。

在 Visual Basic 对象浏览器中，你能够查看 ArcInfo 所有的可创建类和可实例化类，但看不到抽象类。

### 1.8.3 关联

在抽象类、可创建类和可实例化类之间，有几种存在的关联（或称关系）。

联系（association）便描述了类之间的关联。在两端的类中可以定义多重性（Multiplicity）关联。

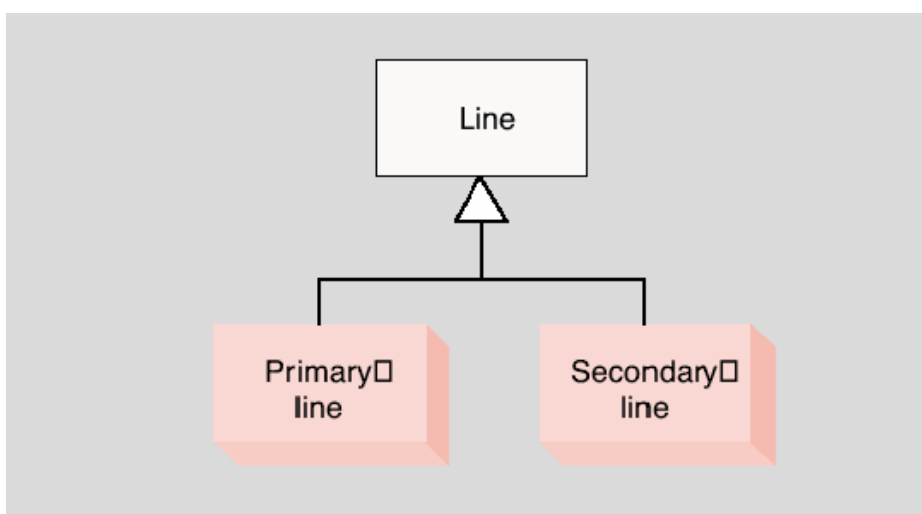


在这张图上，一个业主能有拥有一块或多块宗地；同样地，一块宗地可以被一个或多个业主所共有。

多重性关联就是限制对象类与其它对象关联的数目关系。以下是用于多重性关联的符号：

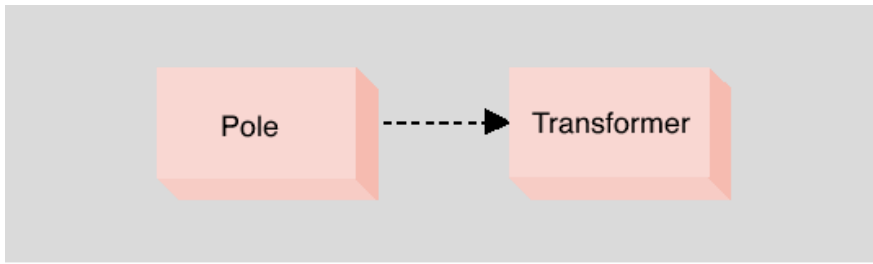
1	——	一个并且只有一个，这种多样性是可选的；如果不标明，则默认为“1”
0..1	——	零个或一个
M..N	——	从 M 到 N（正整数）
*或者 0..*	——	从零到任意正整数
1..*	——	从一到任意正整数

类继承（type inheritance）定义了专门的类，它们拥有超类的属性和方法，并且同时也有自身的属性和方法。



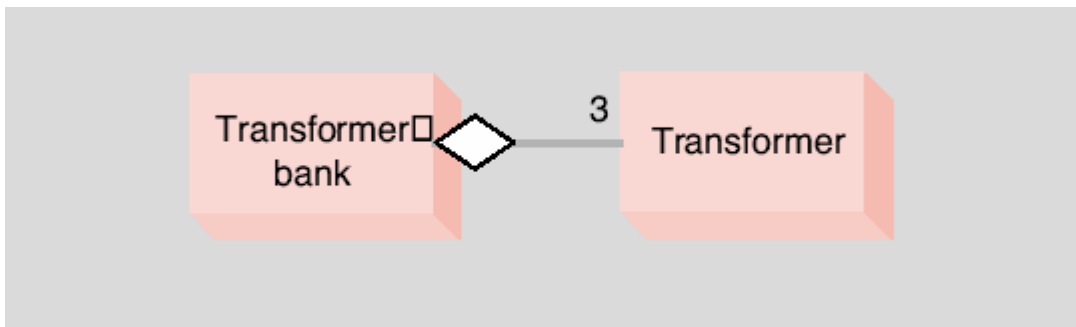
上图说明 primary line 和 secondary line 是 line 的一种类型。

实例化(Instantiation)指定一个类的对象有这样的方法,它能够创建另外一个类的对象。



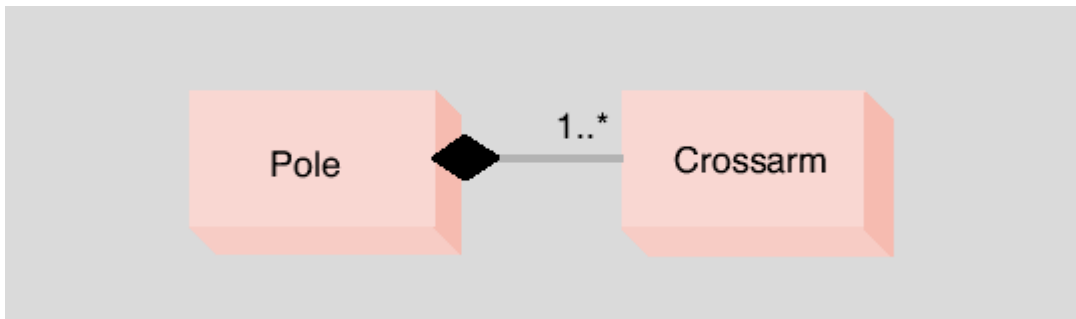
*pole* 对象有一个方法能够创建 *transformer* 对象。

聚合(Aggregation)是一种不对称的关联方式,在这种方式下一个类的对象被认为是一个“整体”,而另一个类的对象被认为是“部件”。



一个 transformer bank 正好有 3 个 transformer。在这个图中 transformer 能和一个 transformer bank 相关联,但当 transformer bank 移除以后,transformer 依然能够存在。

组成(Composition)是一种更为强壮的聚合方式,此种方式下,“整体”对象控制着“部分”对象的生存时间。



一个 pole 包含一个或多个 crossarm。在这个图中当 pole 被移除后, crossarm 就不能再使用了。因为 pole 控制着 crossarm 的生存时间。

#### 1.8.4 用图形符号描述模型

起初你也许不大习惯使用这种图形符号,可以先试着读解上面的例子,并结合自己的实际情况考虑。尝试使用这样的工作方法,而后你将会发现这样的图的读解释很容易的,而且你也觉得其实花点力气去理解这些符号真的帮助很大——它用简单而富有表现力的方法将对象模型很好的展示给我们,这样理解 ArcInfo 组件的概念便轻松简单多了。

如果你需要定制要素,扩展 geodatabase 数据访问对象,那么这些 UML 符号也是很重要并可以使用的。

## 1.9 技术动向

GIS 核心在于一个强大的存储、检索和显示空间数据的数据库管理系统。

ArcInfo 8 是一个采用全新 GIS 技术的新一代软件，它的构建采用了非常重要的主流技术，也为这个软件走向市场做好了铺垫。这些主流技术从 GIS 的角度来看体现为“地理的可用数据库”。

ArcInfo 8 的出现是非常幸运的，那段时间正是软件开发技术和数据库技术的日益成熟的时候。下面介绍构成 ArcInfo 8 框架的主要技术。

### 1.9.1 空间数据和数据库

Coverage 数据模型实现的时候，当时的技术能力决定了只能将要素空间信息存储在二进制文件中，同时将要素的属性信息存储在关系型数据库表的带有唯一标识符的行中。

随着数据库技术的不断发展，如今在数据库中能够将空间数据和属性信息共存于同一个数据库中。

将空间数据直接存入商用数据库有利于提高空间数据库的管理、数据访问效率和管理服务能力也得到提高，并且更方便地与其它数据库进行数据交换。

还有一点需要指出的是，ArcInfo 用户能够选择有很多符合工业标准的数据库来构建他们的地理数据库。现在支持的关系数据库有 IBM DB2、Oracle、Informix、MS SqlServer。

### 1.9.2 用户界面

ArcInfo 桌面应用程序在 MS Windows 操作系统上开发了一套全新的、简单易用且风格一致的用户界面。用户可以使用习惯的工具进行操作，比如鼠标操作、菜单、对话框等等。标准的用户界面对用户的需求不高，使得很多大型软件的上手和使用变得更加容易。

ArcInfo 8 完全实现了标准的 Windows 用户界面，矗立了 GIS 软件的一个里程碑。

### 1.9.3 软件组件技术体系

现行的软件都遵循组件技术，这些技术有如 Component Object Model (COM)、Common Object Request Broker Architecture (CORBA) 和 Java Remote Method Invocation (RMI) 等。

组建技术的理念就是将软件划分为离散的、能够独立开发、测试并组成程序的功能模块。这样，组件能用来构建很多应用软件而不需要修改很多东西。这也是软件重用的高级层次。

组件技术的优势在于更好的软件质量、更好的使用性能，并且在不影响其它安装软件的前提下可进行软件版本升级。

ArcInfo 8 是基于微软的 COM 技术而构建的。

### 1.9.4 开发环境

可视化的开发环境已经成为了应用程序开发的标准，比如 Visual Basic。

使用这样的语言进行开发的好处在于有大量的经验丰富的程序员以及实用的开发环境。现在已经不再满足于使用专用的宏语言来完成工作了。

ArcInfo 8 将 Visual Basic for Applications (VBA) 作为内置的定制应用程序 ArcMap 或者 ArcCatalog 的宏语言。支持 COM 的其它开发语言，如 Visual C++，也可以用来扩展 geodatabase 的数据模型和桌面应用。

### 1.9.5 技术动向总结

这些主流技术的共同点是便是“开放标准”和“互用性”。这些主流技术的实现吸收了其它工业技术的优势。ESRI 在这个基础上，便能够集中力量进行核心 GIS 功能的研究和开发了。

张俊 编译  
邢超 审校、策划  
曾杉 主编

( ArcGIS 中国培训中心 [www.arcgisworld.com](http://www.arcgisworld.com) 转载请声明 )