IBM Rational Software Architect是IBM在2003年二月并购Rational以来，首次发布的Rational产品。IBM Rational Software Architect是 在Eclipse 3.0 的基础之上创建的，它不仅对你的团的中的所有技术工种（架构师，设计师以及开发人员）提供一个统一的接口，而且还允许这些工具支持 Eclipse 提供的使用特性。因为 Rational Software Architect 是在 Eclipse 基础上建造的，你可以将 Rational Software Architect 用于Java 2 平台，企业版 (J2EE 平台) 技术。 除此之外,使用代码生成功能，这是 Rational Software Architect 的一部分,你可以把设计和画在建模视图中的 UML 图转换为代码。最后，底层的 Eclips 平台也提供强健和功能丰富的集成开发环境给开发者。

然而，Rational Software Architect 并不局限于 Java 技术或 J2EE 平台。 你也可以在需要时把你创建的 UML 模型转换为 C++ 代码。这个工具的代码生成功能可以让你很轻易产生基本的代码，然后你可以进行定制修改。这个特性允许开发人员在非J2EE平台开发时，也可以使用Rational Software Architect的建模和设计模式功能。

用 Rational Software Architect 进行UML建模

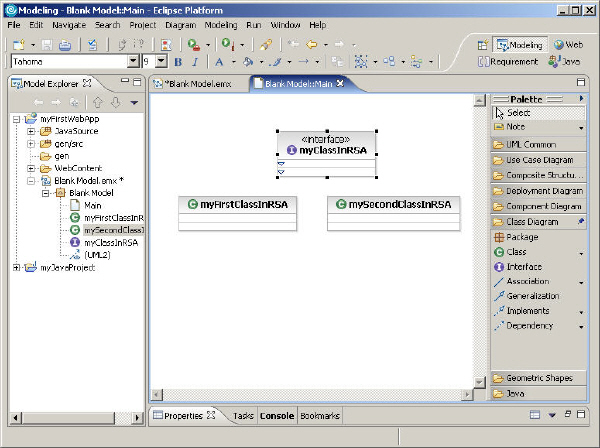
Rational Software Architect 符合UML 2.0 规范；使用 Rational Software Architect，你可以创建几种不同类型的 UML 图。让我们看一些你可以创建的图类型的例子。

创建一个 UML 类图

使用类图来描述一个系统中对象类型以及它们之间的关系。对于一个单独的类，你可以定义类的名字，属性和操作。 除此之外，你可以定义类之间的关系。 了解这个过程的最容易的方法就是创建一个简单的类图。我创建了一个有一个接口和两个实现的类图。执行下面的步骤来重新创建这个类：

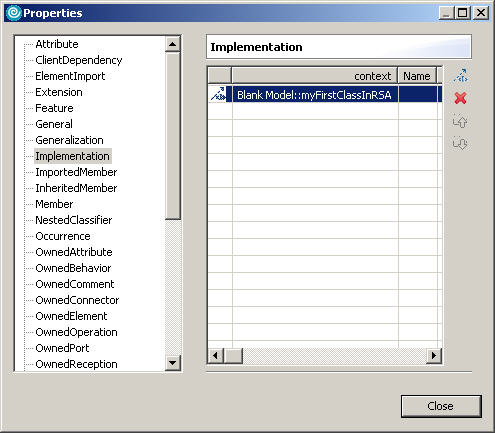
1. 在 Model Explorer 中, 鼠标右键点击 **Blank Model**，然后增加一个新的类图。
2. 为新的类图键入一个名字（对于这个实例，使用名字 *Main*。）。一个叫做 **BlankModel::Main** 新页将会出现(如图 3 所示)。
3. 在 **BlankModel::Main** 窗囗中的任何地方点击鼠标右键, 然后增加一个新的接口。
4. 在这个窗囗中的任何地方再次点击鼠标右键, 然后创建一个类。
5. 重复第 4 步创建第二个类。

图 3. Blank Model::Main页



定义一个关系，指明那两个新建的类是这个接口的实现。点一下这个类的名字，你将发现在这个对话框的右边会出现上下箭头。点击一个箭头，然后点击 **Create Implementation**。你现在就已经在 Rational Software Architect 里创建了你的第一个类。为了看到关于你创建的类的所有有效选项，鼠标右键点击任何类，然后选择 **Properties**（如图 4 所示）。

图 4. 类属性



UML 到编码

现在，让我们从你刚刚创建的 UML 类图中生成代码。为了这么做，请执行下列步骤：

1. 选择在图 3 中所显示图的所有类。
2. 鼠标右键点击你的选择, 然后点击 **Transform**。
3. 选择 **UML to Java** 选项。
4. 在转换向导中,创建一个新的目标容器。

因为你原来创建的项目是一个 UML 项目，所以你现在需要创建一个 JAVA 项目来生成代码。让其它页保持默认值，然后在转换向导中点击“ **Run** ”。你将会在浏览器窗口的左边看到所有正在生成的代码显示出来。点一下这些代码看看怎么样。列表1显示了从 UML 中已经定义好的类生成的代码。

列表 1. 从 UML 中一个已经定义好的类生成代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | /\*   \* Created on Nov 20, 2004   \*   \* TODO To change the template for this generated file go to   \* Window - Preferences - Java - Code Style - Code Templates   \*/    /\*   \* @author Kunal Mittal   \*   \* TODO To change the template for this generated type comment go to   \* Window - Preferences - Java - Code Style - Code Templates   \* @uml.annotations   \* derived\_abstraction="platform:/resource/DevWorksProject/Blank Model.emx   \* #\_PnydkDsQEdmyqsBhJhf6RA%2cuml2.Interface%2cBlank+Model%3a%3amyClassInRSA"   \* @generated "UML to Java   \* (com.ibm.xtools.transform.uml2.java.internal.UML2JavaTransform)"   \*/  public interface myClassInRSA {  } |

代码到 UML

现在,让我们反过来做一下：用你的代码生成一个 UML 模型。这个过程实际上相当容易。只是创建一个空白的类图。从 Java 项目中把代码拖到这个空白的图就行了。UML 类图会被自动生成。当你执行这个任务时，请注意在这个过程和以前版本的 Rational Rose 或者 XDE 有些细微的差别：

* 如果你的类图是在 UML 模型里，你事实上可以引用在 JAVA 项目里的源代码。
* 你也可以从你的 JAVA 项目里来创建和查看类图。这种情况下，这些图仅仅是这些代码的可视化。

关于这个主题的更多信息，参见文章，*How to migrate from code-centric to model-centric development using Rational Software Architect*，列在了本文结尾的 [资源](https://www.ibm.com/developerworks/cn/rational/kunal/#artrelatedtopics) 小节。

创建一个 UML 用例图

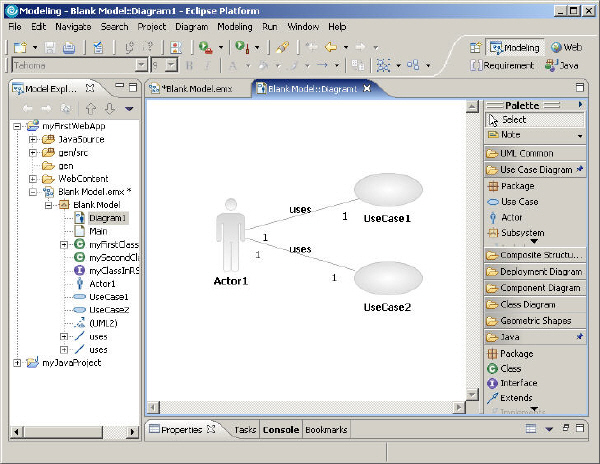
*用例* 是指参与者（Actor）在系统上执行的一系列操作和产生的响应。*用例图* 是一个反映系统中所有用例以及与它们交互的参与者的上层视图。(从本文结尾的 [资源](https://www.ibm.com/developerworks/cn/rational/kunal/#artrelatedtopics) 小节中的引用，你可以阅读到更多有关用例的内容)

用例图很容易创建。 为了创建一个用例图,请执行下列步骤:

1. 创建一个叫 *Diagram1*的图。
2. 使用面板,创建一个被称为 *Actor 1*的参与者。
3. 创建两个用例，叫 *Use Case 1* 和 *Use Case 2*。
4. 点击每个用例，并将它拖到这个主角上，使它们连接在一起。

Figure 5图 5. 已经完成的用例图

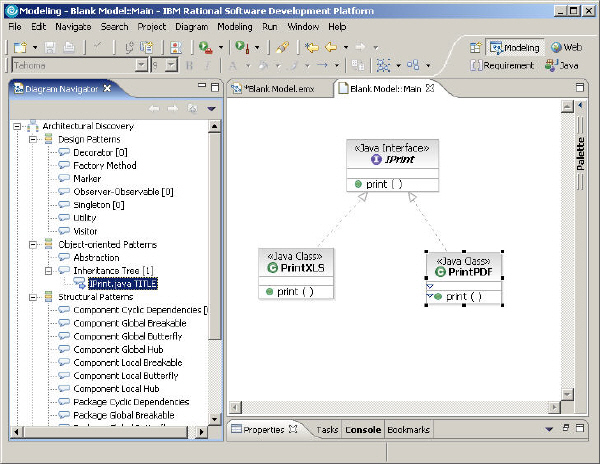
图 5. 已经完成的用例图



架构挖掘

Rational Software Architect 包括了很多可以帮助你在模型或者代码上完成架构挖掘的功能，这能帮助你通过一个架构视图来理解你的模型。 图 6 显示了我创建的一个简单 UML 模型。调出图导航视图，然后展开一些属性。注意一下如何执行分析的，而且你开始看到更多关于你创建好的简单类图的信息。 左边的面板显示了那些架构挖掘。看看 Object-oriented Patterns 下的部分: 你会看到在UML类图中表述的抽象类和继承树。

图 6. 在一个 UML 模型实例的架构挖掘



架构挖掘特性非常重要，因为它对于洞察代码的总体质量非常有价值。它能帮助你识别出潜在的问题区域和结构上的问题。 尽早识别和解决这些问题能最后获得一个更可预期的，强健的，和可维护的产品

总结

Rational Software Architect 是一个基于 Eclipse 的工具，它支持开发者和架构师获得 Eclipse 平台可用性功能。 然而，Rational Software Architect 超越了一个典型集成开发环境的功能 -- 它提供了丰富的建模、架构设计和挖掘的能力。 本文中，我已经向你介绍了关于如何使用 Rational Software Architect 的基本概念, 而且简要地谈到了建模场景及其不同的视图。这些信息是深入 Rational Software Architect 提供的所有建模能力的窗口。

不过,我才只浏览了 Rational Software Architect 表面上的东西。 我已经向你展示的那个建好的简单 UM L用例图，会成为在以后的文章中讨论内容的基础。在其中，你可以根据这个信息 在Rational Software Architect 中去实践一下，创建其它的 UML 图,包括序列图和协作图。

