

Borland Janeva : 连接 .NET 和 Java

左轻侯

2003.8.19

什么是 Janeva ?

在软件界具有领袖地位的几家大公司，每一家都有自己鲜明的技术特色。Microsoft 背靠操作系统，Oracle 以数据库为中心，SUN 靠小型机称雄，IBM 则以“四海一家的解决之道”为号召。至于 Borland，在一般人的印象中自然是开发工具的专业厂商。事实上，Borland 虽然以开发工具起家，但近年来已经逐渐转变为一家提供企业级整体解决方案的厂商，其产品涉及开发工具、建模工具、数据库、中间件、需求分析、变更管理等多个领域。在 IT 业的大战中，Borland 并不专属于某一具体阵营，它的产品线跨越了 Windows、Linux、Unix、Java、.NET 等主要平台。由于扮演了这种角色，Borland 公司在各个平台的协调工作和互相通讯的技术方面，占据了业界独一无二的地位。

Borland Janeva 是 Borland 公司和 C#Builder 同时发布的一个新产品。从字面上来说，Janeva 这个名字是由 Java 和 .NET 两个单词拼合而成，这预示着它是一种能够连接 Java 和 .NET 的技术。和大多数人一样，作者开始对 Janeva 并没有特别的重视。直到连续接到国内几家大型软件公司技术部门的电话或 email，询问 Janeva 的技术细节和销售情况以后，作者这才对 Janeva 产生了兴趣。

经过一段时间的学习，作者慢慢熟悉了 Janeva 的一些技术细节和市场定位，同时也了解了为什么这些软件公司对 Janeva 如此感兴趣。在和公司同事进行讨论时，大家也一致认为，Janeva 是一个很有技术特色、也很有市场前途的产品，可望在未来的企业级架构中扮演重要的角色。

那么 Janeva 到底是一个什么样的产品呢？

根据 Borland 公司的文档，Janeva 的特征是 Platform Interoperability for the Enterprise，注意两个关键之处，一个是 Platform Interoperability，一个是 Enterprise。Janeva 提供了从 .NET 平台到 J2EE/Corba 平台的无缝互操作性，通过高度可扩展的、安全的 Internet Inter-ORB Protocol (IIOP?)，使得基于客户端或是服务器的 .NET 应用程序能够访问 J2EE 和 CORBA 服务器上的组件。Janeva 以 .NET 组件的形式提供，因此可以方便地嵌入到任何主流的 .NET 开发环境中去，包括 Microsoft Visual Studio .NET 和 Borland C#Builder。Janeva 也可以被任何支持 .NET 平台下的公共语言运行时 (Common Language Runtime) 的编程语言所使用。

在企业级市场，J2EE 正在如日中天，越来越多的大型应用都基于 J2EE 的解决方案。除了以 WebSphere、WebLogic、BES 等 J2EE 服务器为基础的核心应用以外，还衍生出了针对不同环境、不同行业的各种解决方案，可以说，J2EE 在服务端已经是根深蒂固，难以动摇。相对应地，Microsoft 则在客户端占有绝对的优势。特别是 Microsoft 在发布 .NET 架构以后，雄心勃勃地希望将 .NET 架构打入企业级市场。根据分析家的预测，Java 或者 .NET 一统天下的可能性都很小，在可预见的未来一段时间里，将会出现 Java 和 .NET 共同瓜分应用软件基础平台的局面。

不可否认，Java 和 .NET 都是非常优秀的 Framework，而且对分布式组件技术都提供了良好的支持：Java 提供了 RMI，.NET 提供了 .NET Remoting。在各自的领域内，这两种解决方案都工作得很好。但是，如前所述，当面临着 Java 和 .NET 长期共存的局面时，如何在这两大平台之间进行互相通讯的问题就凸现出来了，因为无论是 RMI 还是 .NET Remoting，都不能够连接到对方的平台。特别是大型的企业，不可避免地需要面对在异构平台之间进行集

成的问题，因此它们对这种通讯的解决方案特别关心，也就在情理之中了。

异构平台通讯的解决方案

在 Janeva 出现之前，出现过一些针对异构平台通讯的解决方案。其中比较著名的一种是 Web Service。Web Service 可以用于访问远程机器上的对象，使用标准的 XML 文件作为数据封包的格式。由于它的规格简单而且开放，并且独立于任何平台和编程语言，因此大多数主流的平台和语言都对纷纷提供对 Web Service 的支持，也包括 Java 和 .NET 在内。的确，通过 Web Service，可以实现在 Java 和 .NET 之间的对象访问。从理论上来说，通过 Web Service，可以实现任何平台之间的对象访问，只要它们提供了对 Web Service 的支持。因此，有一段时间，Web Service 成为了一个非常流行的名词。

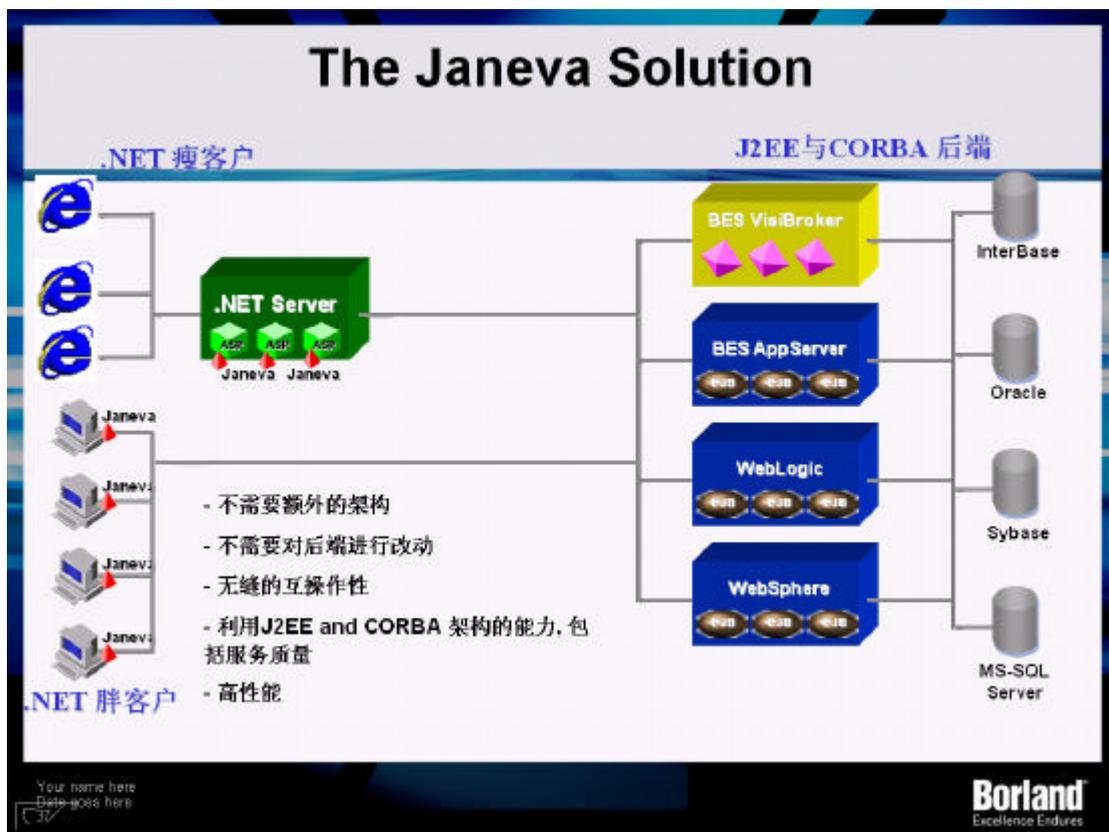


图 1：Janeva 解决方案示意图

但是在实践中，人们很快就发现 Web Service 并不是预想中的终极解决方案。原因是，Web Service 具有如下种种缺陷，而且这些缺陷几乎是无法彻底克服的：

首先，Web Service 技术本身虽然比较简单，但是为了通过 Web Service 进行通讯，服务端和客户端都必须进行额外的架构，将原有的对象进行封装和暴露。这将需要投入额外的资源来进行编程和维护，更重要的是，涉及到对服务端的改动。而在一些关键性的应用中，由于对可靠性的要求极高，因此对于已经经过实际验证能够稳定运行的系统，进行改动带来的风险是很大的。

其次，Web Service 技术的简单性，一方面为开发和维护带来了方便，但是另一方面，也带来了对于分布式组件技术的某些高级特性支持不够的缺陷。Web Service 对于事务处理、安全特性以及服务质量等特性的支持，一向是老大难问题，而且由于 Web Service 本身的性

质,在未来也难以出现理想的解决方案。很容易理解,在企业级应用的领域,对上述特性不支持或支持得不够完善的技术,是很难有什么前途的。

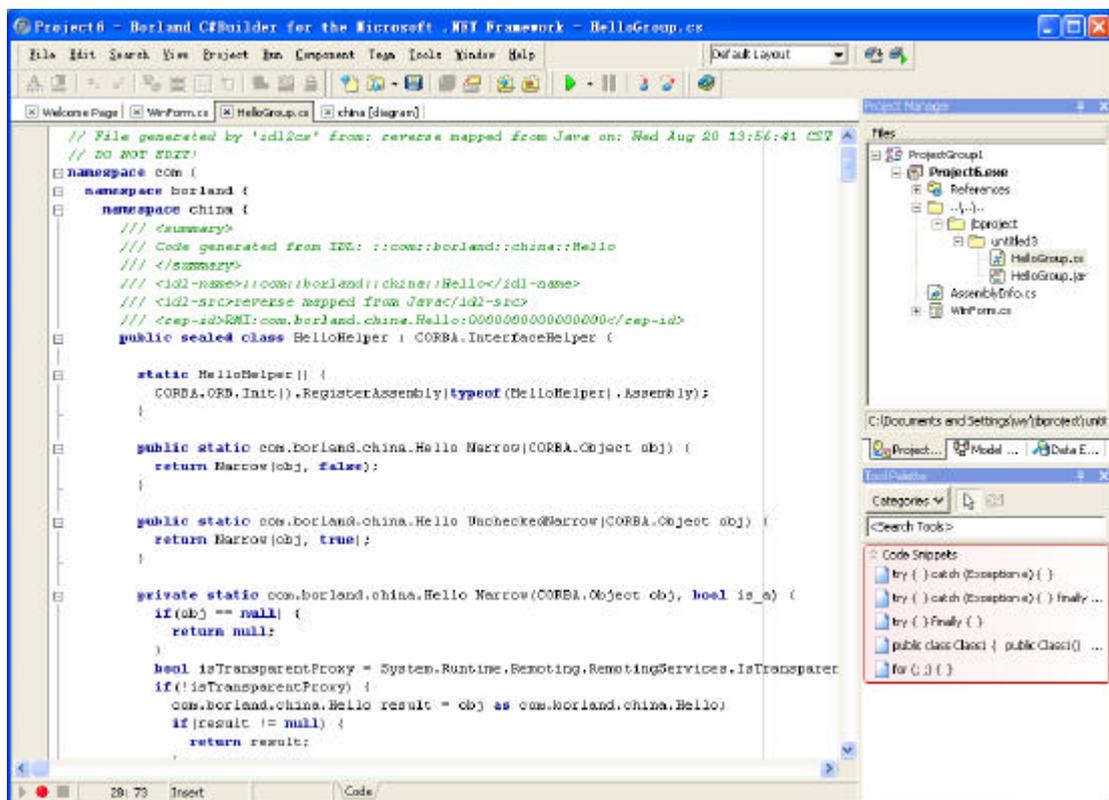
第三,Web Service 使用了标准的 XML 作为数据封装的格式,虽然获得了最好的兼容性,和简化了开发,但是也导致了在性能上的损失。XML 的字符流和传统的二进制流,处理的信息量可以相差一个数量级以上,再加上编码/解码的过程,差距就更大了。因此,在一些对性能要求苛刻的实时系统中,Web Service 基本上不被考虑。

另外一种解决方案是所谓的“Bridge”,即桥接器技术。这种技术使用专门的桥接器,作为服务端和客户端的中转站。当客户端向服务端发送请求时,桥接器将请求转换为服务端能够理解的格式,再转发给服务端。当服务端发送回应时,同样也通过桥接器中转。很显然,这种桥接器解决方案同样也存在着性能上和兼容性上的问题。

Janeva 的优势

相比之下,Janeva 采用了一个独特的技术角度切入,这使我们不得不佩服 Janeva 的设计者的天才构思。Janeva 以.NET 组件的形式提供,本身是一组符合.NET 规范的 assembly,因此可以非常方便地在.NET 开发中使用,只需要导入相应的包名,就可以访问 Janeva 中的对象和方法。在 Janeva 的底层,Borland 的工程师们实现了完整的 IIOP 协议,通过 IIOP 协议和 J2EE/Corba 服务器上的远程对象进行交互。Janeva 的这种架构具有如下优势:

采用 Janeva 的解决方案,无需对服务端进行任何改动,对客户端的影响也降低到了最小程度。由于直接采用 IIOP 通讯,服务端会认为它正在与一个 J2EE/Corba 客户端进行通讯,因此不需要额外的架构。而在客户端,J2EE/Corba 对象会被自动映射成.NET 的本地对象的形式,只需要针对该对象编程即可。这样,有效地保护了已有的投资和技术,避免了因为架构变动带来的风险。



图二：在导入 EJB 包以后，C#Builder 通过映射机制自动生成的 cs 文件

由于 Janeva 的解决方案建立在对 IIOP 的完全兼容性的基础之上，因此它对分布式组件技术的高级特性有着良好的支持。Janeva 能够支持双向提交的事务处理，加密、验证、授权等的安全特性，以及各种复杂数据类型（自动进行 Corba 和 J2EE 的映射）。因此，Janeva 完全可以适应复杂的企业级应用环境。

在采用 Janeva 的解决方案中，数据流是二进制流而不是 XML 那样的字符流，同时也省去了编码/解码的开销，并且不需要任何桥接器进行中转。因此，Janeva 在性能上有足够的保证，有相关的测试数据表明，Janeva 和 Web Service 的解决方案，在效率上有 2~40 倍的性能差距。在对性能要求苛刻的系统中，Janeva 是当之无愧的选择。

Janeva 的未来

目前已发布的 Janeva 版本，暂时还只支持从 .NET 客户端到 J2EE 服务器端的连接。在后续的开发计划列表中，Borland 公司公布了一系列新的目标，包括从 Java 客户端和 .NET 服务端的双向连接，对 WebSphere、WebLogic 等业界流行的 J2EE 中间件的完善支持，以及其它的高端特性。可以预料，在未来的分布式组件市场中，Janeva 将会占有非常重要的地位。