

JBoss5迎来中间件彻底的可配置时代 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/167/2021_2022_JBoss5_E8_BF_8E_E6_c104_167300.htm 面向对象奠基人之一Grady Booch说：The great thing about objects is they can be replaced.对象最伟大的之处是其可被替代（这也是使用OO的主要原因之一）。每个对象都是可替代意味着高度的灵活性，我们曾经梦想的“按需装配”时代已经来临，由Ioc模式/依赖注入组成微容器可以帮助我们实现对象的可替代性。Spring/HiveMind 包括Jdon Framework都是Ioc组成的一种微容器，在Java企业系统架构选择考量一文中，我已经在灵活性方面对几个组件架构进行了比较。其中一个重要的疑问：EJB3是POJO吗？这里面有两个概念：EJB3是否支持POJO？EJB3本身是否是POJO？前者答案是肯定的，但是后者则曾经是否定的。在回答之前，我们必须对POJO有一个详细了解，最初POJO是相对EJB提出的，Martin Fowler 对POJO定义是：我发现：人们已经忘记了原来正常的Java Object，因为这些对象还没有一个很特殊的名字，....这样我它们取名为POJO(Plain Old Java Object)，一个POJO domain model容易放在一起，快速build，在EJB容器以外运行和测试，并且不依赖EJB

。 <http://mindprod.com/jgloss/pojo.html> 但是，随着EJB3支持POJO，POJO的概念从原来相对EJB的定义已经引申开来，代指一种相当灵活的对象，也就是可被随时替换的对象，不因为依附任何框架而不能被替换。那么，EJB3本身是否是POJO？实际意义是EJB container是否是POJO，也就是说：EJB本身组件是否可被替换？正如我在Java企业系统架构选择

考量一文中所写，当我们只需要EJB3的集群，而事务等基础功能都不需要时，EJB服务器是否支持我们这种任意配置和切割？或者我想替代其中一个基础功能，是否可任意供我们切换，也就是Grady Booch那句话：对象是否可替换？当然，在这场“EJB3是否是POJO”讨论中，有人引用一些老外名言：EJB3本身是否是POJO没有讨论意义，可惜说这话的老外自己的概念没有达到最新理念上。那么，作为一种组件结构，是否可以既支持应用系统的任何一个组件对象可替换，而且也支持框架本身的组件也是可替换，这个境界是否可以达到呢？完全没有问题，目前,开源软件HiveMind和Jdon框架都是支持彻底的可替换，所谓彻底的可替换就是框架本身一些功能也是可配置，可嵌入的，而不只是应用程序是可替换的。这就实现了组件架构的完全的、彻底的可配置性，是一种Embeddable或Plug-in架构，这样的架构可允许开发者介入任何一个层次进行拓展和维护，从而形成强大的可定制性和可拓展性，可以使用建筑的一个比喻，这种Embeddable架构类似钢筋结构建筑，它只有固定几个框架和板筋，你喜欢划分什么样的房间完全由你来决定。唯一的限制是你的想像了。现在，作为EJB3 container设计领先的开源软件JBoss即将推出JBoss 5版本，在其JBoss 5版本中，其微核心本身将是可配置的，最终将实现EJB3的彻底的可配置性。我们看看JBoss Blog(<http://www.jboss.org/jbossBlog/blog/>)上这段文字：JBoss微容器将是彻底的反转控制，依赖注射的轻量容器，它允许你通过XML配置POJO，这些POJO有自己的生命周期，能够作为服务Service，它并不需要JBoss的应用服务器，..大多数JBoss提供的功能将都会转为POJO，并且可配置...这些都将在2006

年的JBoss 5版本中完全实现。在这篇现场录像

(<http://www.javalobby.org/av/javazone/69/aardal-jboss>) 中 Thomas Roka-Aardal介绍了JBoss 5 lightweight middleware with EJB3，他介绍了企业Java将被简化和增强，通过结合新的JBoss微内核，显示什么是真正的轻量应用服务器，它又是怎样影响未来企业开发市场的，中间件将会到处被看到。对于现在大部分初学者来说，首先需要从Jsp中嵌入Java代码的坏习惯中改变过来，将你的Java代码使用组件JavaBeans来实现，然后逐步走上面向组件（面向构件）的开发方式，进而上升到可彻底配置的组件化编程层次。在JBoss 5推出之前，J2EE曾经被指责为In-House，也就是说，很多功能被绑定在J2EE服务器上，诞生很多基于JBoss、基于Weblogic的、基于Oracle的、甚至基于JMX的J2EE应用系统，当这样的J2EE应用系统需要移植时，那些依附特性因为和容器/服务器粘性太牢，而无法跟着应用系统跑。因此多年来，应用者业界一直呼吁Out-of-Box，Spring/HiveMind/Jdon等框架应呼而生，并且Rod Johnson在2005 JavaOne大会上演讲上预言：J2EE/Java EE将走向一个以框架为中心的新的开发时代。这样，应用系统依赖的很多功能在框架中实现，而框架是可以和应用系统到处移植的。显然，这种Out-of-Box的倡导对JBoss路线提出了挑战，JBoss奋起反应，你们既然指责我的容器是一个Box，那么我们就一不做二不休息，打破这个Box，JBoss 5即将诞生，因为JBoss 5本身是可肢解可配置的，因此，使用JBoss 5编制的Java EE应用系统需要移植时，可以将应用系统依赖的那些功能从JBoss 5容器中分离出来，带着跑，这样你的应用系统又符合Java EE标准，在特殊之处，也可以将容器作为一种框

架帶了跑。当然，因为目前的J2EE标准包括EJB离实际开发还有一段路，它没有提出一种编程模型，因此作为符合标准的应用服务器JBoss在实际应用中还是需要开发框架辅助的

，Spring提供强大全面的API库；HiveMind提供灵活的配置功能；而Jdon框架在彻底可配置基础上，瞄准应用开发中的增删改查这些功能进行缺省实现，提高开发效率，避免大规模开发中这些纯劳动量的低层次工作。围绕Out-of-box 和break the box，未来组件（构件）架构领域将有一番争夺，有人说，你怎么忘记谈Weblogic和Websphere了？有一点我忘记说了，这场完全可配置运动是由开源领域挑起的，也就是说，在组件架构设计上，开源运动已经走在了工业界前面，工业界巨头们都跑去搞他们赚钱的强项：集成和SOA了。100Test 下载频道开通，各类考试题目直接下载。详细请访问

www.100test.com