Java网络应用框架的分类

托尼·山瓦乔维亚银行[*tonycshan@gmail.com*](mailto:tonycshan@gmail.com)

温妮·华

*中旅公司。*

[*winniehua@yahoo.com*](mailto:winniehua@yahoo.com)

# 摘要

*本文从技术角度描述了与Java EE模型相关的各种网络应用框架和相关新兴技术。“web应用程序框架”的定义已经明确，因为这个术语已经被广泛使用，并且在不同的上下文中有着截然不同的含义。web应用程序框架的价值主张旨在说明框架如何提高应用程序开发的生产率和质量。阐述了web应用程序框架的设计理念。定义了一个全面的分类方案，将各种软件框架和web应用程序框架分类到适当的类别中。在作为商业和开源解决方案可用的几十个web应用程序框架中，主要的产品被调查，然后是选择指南和建议。构建了一个参考卡来总结web应用程序框架的关键方面。讨论了相关技术和未来发展趋势。*

# 介绍

在信息系统环境中，框架是一种定义好的支持结构，在其中可以组织和开发其他软件应用程序。框架可以包括支持程序、代码库、脚本语言、公共服务、接口或其他软件包/实用程序，以帮助开发和粘合软件应用程序的不同组件。软件框架是软件系统和/或子系统的可重用设计和构建块。

软件框架可以面向构建不同领域的应用，如金融建模应用[1]或决策支持系统[2]。

软件框架由冻结点和热点组成[3]。冻结点定义了整体

软件系统的架构——它的基本组件以及它们之间的关系。这些在应用程序框架的任何实例化中保持不变(冻结)。另一方面，热点代表软件框架中特定于单个软件系统的那些部分。热点被设计成通用的。换句话说，它们可以适应正在开发的应用程序的需求。

软件框架定义了架构中应该对特定功能进行调整的地方——热点。在面向对象的环境中，框架由抽象类和具体类组成。这种框架的实例化包括组成和子类化现有的类[4]。

在用软件框架开发一个具体的软件系统时，根据系统的具体需求和要求，对热点进行专门化。软件框架依赖于好莱坞原则:“不要叫我们，我们会叫你。”[5].这意味着用户定义的类(如新的子类)接收来自预定义框架类的消息。这些通常通过实现超类抽象方法来处理，方式类似于模板设计模式的使用。

# 软件框架

## 软件框架类型

一般来说，在信息系统领域有七种类型的软件框架:

* 概念框架——总体架构模型，如扎克曼框架；
* 应用程序框架——应用程序解决方案(如网络工作)的框架结构；
* 领域框架——为特定业务部门量身定制，如IBM信息框架(IFW)；
* 平台框架–编程模型和运行时环境，例如。Net和Java EE框架；
  + 组件框架——用于对象关系映射的Hibernate、iBatis和Cayenne等应用程序的构建块；
  + 服务框架——面向服务计算的业务和技术服务模型，如语义网服务框架；
  + 开发框架——构建富客户端开发工具的构建基础，通常用于ide，如Eclipse、Netbeans和OSGi。

## 网络应用框架的定义

网络应用程序框架(WAF)是一个可重用的、骨架的、半完整的模块化平台，可以专门用于生成定制的网络应用程序，这些应用程序通常通过Http协议为网络浏览器服务。它包括构建复杂的功能丰富的业务服务和协作系统所必需的服务和组件的构建块。WAF通常实现模型-视图-控制器(MVC)设计模式，通常在Model 2架构中，在Java EE和上开发基于请求-响应的web应用程序。Net模型。它还将搜索、版本控制和权限等服务集成到基本业务对象中，使应用程序能够利用框架服务，而几乎不需要额外的工作。在这种情况下，WAF是一种应用程序框架，如前一节中所定义的，特别是为基于Http的通信服务的HTML/XML。

Web应用程序框架域层通常对用户、组和权限等基本概念进行建模。WAF还可能包括其他相关部分，例如用户界面(UI)框架、为web用户界面的快速开发和重用而设计的UI组件库，以及强大的对象关系持久性引擎/实用程序。

## 为什么要使用网络应用框架？

### 开放标准架构

软件框架大大减少了开发和维护web应用程序所需的时间、精力和资源。此外，框架是基于普遍接受的标准(例如，Java)的开放体系结构。Net、XML、XSLT、JAAS、Servlet、JSP、JDBC、ADO.Net)和技术(如JUnit、XUnit、Ant、Log4j、JDom、Xalan、Xerces、Lucene)，使任何有经验的开发人员都能够快速开发和支持系统，而无需陡峭的学习曲线。

这种采用和集成技术的最佳方法使应用程序设计人员能够专注于解决他们的业务问题。因此，采用网络应用框架作为标准

web应用程序的开发基础架构是确保开发不被锁定在任何专有的、死胡同架构中的最佳方式。这种方法极大地减少了技术变动和风险，因为行业标准开源框架由世界上高技能的专业人员积极维护和增强。这些开发人员负责识别合适的技术，集成软件，测试这些技术，并为现有用户提供到最新技术的迁移路径。

网络应用程序框架通常部署在n层架构中，并使用成熟的标准技术。通过使用标准技术，可以利用现有的硬件、软件、流程和人员，在现有的企业基础架构中轻松部署框架。

### 相关域服务

几乎所有的web应用程序都有一组通用的基本要求，例如用户管理(例如，安全用户登录、密码恢复)、组管理和访问授权。一个网络应用程序框架通常包括所有这些功能，通过数百个生产部署进行优化，使开发人员能够专注于他们特定应用程序的需求。

除了一组基本服务之外，web应用程序通常还有另外两个重要的相似之处:它们将重要数据存储在关系数据库中，并通过基于web的用户界面与用户交互。复杂的对象关系持久层自动管理模型对象在数据库中的存储方式。持久层通过从元数据生成优化的SQL，极大地减少了建模和重构数据库模式或支持其他数据库体系结构所需的工作量。

网络应用程序框架还可以包括基于组件的呈现框架，它使开发人员能够扩展现有的用户界面组件或构建可在整个应用程序中重用的新组件，例如XSL和标签库。可以提供广泛的领域服务，例如数据验证、版本控制、分类、打印、页面导航和全文搜索。任何基于网络应用程序框架编写的应用程序都可以透明地立即利用这些基本服务。

# 网络应用框架解决方案

## 设计原理

用于开发web应用程序框架的关键设计原则如下。

* + 1. 简单性——应该编写越来越少的代码来使用框架。避免过度使用XML配置文件。利用以POJO为中心的设计。
    2. 一致性-组件、容器和约定应该一致
    3. 效率-应用程序应该运行良好，并通过粘性会话支持扩展。
    4. 集成——框架不应与现有的好解决方案竞争，而应促进无缝集成。
    5. 可重用性——框架中的构造应该是完全可重用的，并且易于分发/部署。
    6. 非侵入性——HTML或其他标记不应被编程语义所污染，与普通HMTL编辑器兼容，图形设计者易于识别和避免框架标记。
    7. 诊断——当事情出错时，框架不应该碍事；事实上，它应该提供有用的诊断和调试信息。
    8. 开发工具–最大限度地支持工具，最大限度地减少对特殊工具的依赖。

## 网络应用框架类型

几乎所有的Java网络应用框架都是基于MVC模式的。一般来说，目前有五大流派的网络应用框架:基于请求的、基于组件的、混合的、元的和基于RIA的框架。

基于请求的框架非常接近最初的CGI规范。它使用直接处理传入请求的控制器和动作。每个请求本质上都是无状态的。随着服务器端会话的引入，已经实现了一定程度的状态。各种框架基本上通过它们将逻辑映射到URL的方式以及数据是如何被结构化并提供给业务处理程序来区分自己。

基于组件的框架抽象了请求处理的内部，并将逻辑封装成可重用的组件，通常独立于网络媒体。该状态由框架根据每个组件实例中存在的数据自动处理。加上某种形式的事件处理，这个开发模型非常类似于桌面GUI工具包提供的特性。各种框架基本上通过提供的组件应用编程接口和组件如何集成在一起来区分自己。

混合框架通过在基于请求的模型中控制整个数据和逻辑流，将基于请求的框架和基于组件的框架结合在一起。

开发人员仍然接近CGI应用程序的体系结构，并完全控制URL、表单、参数、cookies和路径信息。然而，混合框架不是将动作和控制器直接映射到请求，而是提供了一个组件对象模型，该模型在许多不同的情况下表现相同，例如单个页面、截获的请求、类似门户的页面片段和可集成的小部件。组件可以连接在一起，并被打包为单独的组件组。它们可以单独分发，也可以无缝集成到其他项目中。这将基于组件的框架中的可重用性形式与基于请求的方法的原始控制相结合。

元框架有一组通用服务的核心接口和一个高度可扩展的集成组件和服务的主干。该结构通常实现控制模式的反转来分离关注点，以灵活地合并其他框架和组件。元框架有时被认为是框架的框架。

富互联网应用程序(RIA)是指在浏览器中运行的基于网页的应用程序，具有“胖”客户端中常见的丰富用户界面功能，如拖放、树控件和选项卡式面板。基于RIA的框架使用客户端容器模型，这将服务器通信量降至最低——框架不是每次用户点击时都加载整个HTML页面，而是在本地处理点击(不涉及服务器)，或者以XML格式从服务器请求数据。这意味着真正存在一个具有状态和用户交互模型的客户端应用程序——客户端远不止是服务器上生成的网页。

## 主要产品

### 基于请求的框架网络工作

WebWork [6]为构建可重用的UI模板提供了强大的支持，例如表单控件、UI主题、国际化、动态表单参数到JavaBeans的映射，以及强大的客户端和服务器端验证。

WebWork最初是由Rickard Oberg在2001年开发的，并于2002年3月作为开源项目在SourceForge上发布。WebWork在2002年夏天加入了OpenSymphony项目。截至2005年11月，WebWork被合并到Struts中，成为Struts Action 2.0框架的一部分。

### 支杆

Struts [7]使用并扩展了Java Servlet API，以采用“模型2”方法，这是

经典的模型-视图-控制器(MVC)设计模式。在模型2下，一个Servlet(或等效的)管理业务逻辑的执行，而表示逻辑主要驻留在服务器页面中。

Struts最初由Craig McClanahan开发，并于2000年5月捐赠给Apache基金会。Struts已经成为一个事实上的框架，拥有强大而充满活力的用户社区。Apache Struts项目现在由两个不同的框架组成。这两个框架是Struts行动框架和Struts Shale框架。此外，简单网络框架(SWF)是一个基于事件的框架，面向希望构建丰富的网络应用程序但不想迁移到JSF的Struts开发人员。

### 蜂箱

Beehive [8] (Apache 2006)是一个可扩展的Java应用程序框架，具有集成的元数据驱动的编程模型，用于web服务、web应用程序和资源访问。该框架利用了Java 5中的最新创新，尤其是JSR 175元数据注释。关键组件是网络界面页面流、控件和网络服务元数据。

Beehive从BEA Weblogic Workshop产品的一部分发展到2004年5月的Apache项目。

### 条纹

Stripes [9]是一个健壮而轻量级的表示框架，每页/每个动作没有外部配置，绑定引擎根据请求参数构建复杂的对象网，内置支持每个表单多个事件，透明的文件上传功能和向导表单。

条纹最初于2005年9月发布。

### 基于组件的框架JSF

JavaServer Faces (JSF) [10，11]是一个基于Java的Web应用程序的服务器端用户界面组件框架。JSF包含一个用于表示用户界面组件和管理其状态的应用编程接口；处理事件、服务器端验证和数据转换；定义页面导航；支持国际化和可访问性；并为所有这些特性提供可扩展性。它还包含两个JSP (JavaServer Pages)自定义标记库，用于在JSP页面中表达用户界面组件，以及将组件连接到服务器端对象。

JSF 1.0 (JSR-127)的规格最初于2004年3月发布。JSF 1.2规范(JSR- 252)是下一代JSF，其最终草案于2005年8月发布。

### 挂毯

Tapestry [12]补充和构建了标准的Java Servlet API，并将一个web应用程序分成一组页面，每个页面都是由组件构建的。这提供了一个一致的结构，允许Tapestry框架承担关键关注点的责任，例如URL构建和调度、客户端或服务器上的持久状态存储、用户输入验证、本地化/国际化以及异常报告。开发Tapestry应用程序包括使用纯HTML创建HTML模板，并使用(可选的)XML描述符文件将模板与少量Java代码相结合。在Tapestry中，应用程序是根据对象以及这些对象的方法和属性创建的，而不是根据URL和查询参数。Tapestry为Java web应用程序带来了真正的面向对象开发。

Tapestry最初是由霍华德·刘易斯Ship创建的，该项目于2004年初被转移到阿帕奇基金会。

### 小门

Wicket [13]是一个将简单性、关注点分离和易于开发提升到一个新水平的框架。Wicket页面可以使用标准的所见即所得HTML设计工具进行模拟、预览和后期修改。动态内容处理和表单处理都是使用组件模型在Java代码中处理的，该组件模型由可以使用不同技术轻松持久化的POJO数据豆支持。Wicket有一个透明的状态管理，没有XML配置文件。

Wicket 1.0的第一次发布于2005年6月。

### 混合框架RIFE

RIFE [14]是一个全栈web应用程序框架，具有实现大多数常见web功能的工具和API。它的每一个工具包都可以单独使用，它们一起提供了强大的集成功能，可以提高您的工作效率。RIFE确保每个声明和定义都在代码中的一个地方处理。这通过减少代码复制、增强一致性和简化维护来简化开发人员的任务。

RIFE 1.0于2005年9月发布。

### 元框架龙骨

龙骨[15]是现成的服务器端基础设施。龙骨结合了多个开源项目，提供了一个最佳的框架，可以很好地工作

盒子的。龙骨与众不同之处在于它的组件设计，允许添加新的实现和替换旧的实现，而不需要大量重写应用程序代码。

龙骨的第一次预发布是在2003年1月，尽管最初的小组是在2002年2月成立的，龙骨是在同年3月命名的。

### 春天

Spring [16]是一个分层的Java EE应用程序框架，它包括一个轻量级容器，用于通过控制反转(又称依赖注入)自动配置和连接应用程序对象，一个用于事务管理的抽象层，一个JDBC抽象层，AOP功能，以及与O-R映射器的集成。

Spring的起源可以追溯到Rod Johnson的一本书[17]，他提出了他的interface 21框架，后来发布到开源世界，这个框架构成了Spring框架的基础。第一个正式的1.0版本于2004年3月发布。

### 基于RIA的框架DWR

直接网络远程处理[18]是一个在浏览器中直接从JavaScript调用服务器上的Java代码的框架。DWR由两个主要部分组成:运行在用户浏览器中的JavaScript，用于与服务器通信并动态更新网页；以及运行在服务器上的Java Servlet，用于处理请求并将响应发送回浏览器。DWR通过动态生成基于JavaScript代码的Java类，对Ajax采取了一种新颖的方法。因此，web开发人员可以使用JavaScript中的Java代码，就像它是web浏览器的本地代码一样；而实际上，Java代码在web服务器中运行，并且可以完全访问web服务器资源

《DWR》的第一版于2005年上市。

### 回声2

Echo2 [19]是Echo Web Framework的下一代，这是一个开发基于Web的应用程序的平台，这些应用程序接近富客户端的功能。2.0版本忠实于Echo的核心概念，同时通过其新的基于Ajax的渲染引擎提供了引人注目的性能、功能和用户体验增强。

Echo2使开发人员不必考虑“基于页面”的应用程序，并使他或她能够使用传统的面向对象和事件驱动的用户界面开发范式来开发应用程序。不需要了解HTML、HTTP和JavaScript。应用程序可以使用任何Java Servlet容器托管。回声2，像它的

前身是根据Mozilla公共许可证或GNU LGPL许可证的条款分发的开源软件，如果愿意的话。

Echo2最初的alpha版本于2005年3月推出。

### JSON-RPC-Java

JSON-RPC-Java [20]是Java中动态的JSON-RPC实现。它允许您使用包含的轻量级JSON-RPC JavaScript客户端从JavaScript透明地调用服务器端的Java代码。它被设计成在一个像Tomcat这样的Servlet容器中运行，并且可以与JBoss和其他J2EE应用服务器一起使用，以允许从一个JavaScript DHTML web应用程序中调用普通的Java或EJB方法。

需要对现有服务器端Java代码进行最小或零更改，以允许从JavaScript调用(例如特殊类型的编组和解组)，因为JSON-RPC-Java使用Java反射动态地将JavaScript对象映射到Java对象和从Java对象映射。JSON-RPC-Java允许通过对Java对象的方法签名进行反射来简单地导出Java对象(需要一行代码来提供对Java对象的所有公共方法的访问)。

第一个主要版本1.0于2006年3月发布。

## 选择指南

这些框架之一的选择取决于整体架构决策以及应用程序的特定功能。框架并不总是必需的，但是可以显著减少从头开始开发应用程序所需的时间和精力。在设计过程中，其他架构考虑可能会推动特定框架的使用，或者从旧框架向主流框架的迁移。

IBM WebSphere应用服务器和WebSphere Studio应用开发人员(WSAD)集成开发环境以及取代WSAD的Rational应用开发人员支持Struts。

同样，JSF在IBM WebSphere应用程序服务器和WebSphere Studio应用程序开发人员集成开发环境以及新的Rational应用程序开发人员中得到支持。太阳微系统公司提供了一个JSF参考实现，它可以自由地以二进制形式重新发布。此外，Apache MyFaces项目提供了JSR-127的开源JSF实现。

MyEclipse企业工作台(4.0版)支持Sun JSF参考实现1.1.01和MyFaces 1.0.9。它还提供了Tapestry集成。

使用Beehive需要JDK 1.5。Beehive产品支持JSR-175元数据注释。Beehive的NetUI页面流是基于Struts的。授粉是一个Eclipse技术项目，计划构建一个基于Eclipse的集成开发环境和工具集，利用开源Apache Beehive应用程序框架。

Struts框架的主体是控制器组件，它可以与任何Java表示技术一起使用。JSF和JavaServer标准标签库(JSTL)是对Struts的补充，扩展是可用的。

Spring提供了一个AOP框架、一个独特的事务管理抽象和一个独特的数据访问抽象。Spring Modules是一个相关的项目，它将Spring平台的范围扩展到了不一定是Spring核心组成部分的区域。Spring IDE是Spring框架使用的配置文件的图形用户界面。它是作为Eclipse平台的一组插件构建的。

作为web应用程序设计中的一般实现准则，Spring可能首先被认为是首选基线，随后适当的前端控制器实现是合理的——Struts、JSF、Wicket和RIFE。应该评估灵活的呈现机制。Beehive可以和Struts结合使用。Beehive中的页面流比Spring中的网络流更成熟。如果应用程序需要向导表单，条带是一个很好的选择。挂毯可以酌情用在小众领域。DWR和Echo2是开发RIA系统的可行选择。

构建了一个参考卡来总结web应用程序框架中的关键方面，如表1所示。

# 其他框架

## Java 语言(一种计算机语言，尤用于创建网站)

下表列出了前面章节中没有提到的主要Java web应用程序框架。

## 相关技术

在。NET世界，当前版本的微软企业库包含七个应用程序块。这些应用程序块是组件和面向方面的API。NET应用程序。在这种情况下，企业库不被认为是一个网络应用程序框架。

微软的另一个应用程序块名为用户界面进程(UIP)，不包含在当前版本的企业程序库中。2.0版实现了MVC模式，并为web会话状态管理、导航和布局管理以及向导支持提供了可重用的代码组件。UIP可以归类为一个网络应用框架。

Spring框架已经移植到。net Spring富客户端项目(RCP)是一个子项目，通过利用Spring框架和丰富的用户界面工厂和支持类库，提供了一种优雅的方式来更快地构建高度可配置的、遵循图形用户界面标准的富客户端应用程序。

此外，DotNetNuke是一个开源的网络应用程序框架，它是微软IBuySpy门户解决方案工具包的演变。DotNetNuke有一个基本核心，可以使用可插拔模块和具有附加功能的提供程序进行扩展。单个网站的外观和感觉可以使用皮肤来定制，这提供了设计和内容之间的明确分离，使网页设计师能够开发皮肤，而不需要任何ASP开发的专业知识。NET:只需要HTML的知识和了解如何准备和包装皮肤本身。

OpenLaszlo是一个开源平台，用于创建具有桌面客户端软件用户界面功能的零安装web应用程序。OpenLaszlo程序是用XML和JavaScript编写的，对Flash透明编译。最近增加了DHTML支持。

# 未来趋势

对web应用程序框架的需求非常迫切，因为多年来在Java领域已经构建了几十个框架。在这些竞争产品中，一些产品已经脱颖而出，成为主要竞争者，原因有很多，例如强大的用户社区、标准化、良好的文档、工具集成、供应商认可和易用性。例如Struts/Shale、JSF/MyFaces、Spring、Tapestry、Beehive和DWR。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 茧 | 重担 | 磺唑氨酯 | 索非亚 | Canyamo |
| 标新立异的 | jppublish | 喷射机起飞助推 | 薄层 | Jucas |
| 主教 | 使隐隐不安 | 梭鱼 | 震惊 | Jeenius |
| 边缘 | 翅膀 | Expresso | Melati | Xoplon |
| jZonic | 脑筋急转弯 | JBanana | 千叶 | 提花织物 |
| 开放式主持人 | 便当 | 妖怪 | JWarp | 金刚鹦鹉 |
| 巴里图斯 | Nacho | 涡轮 | 吻合 | 范围 |
| JWAA | 微笑 | 雅法 | 冲突 | JFormular |
| 吻合 | 日本人 | 卡梅洛恩 | 卡桑德拉 | 赫尔马 |

### 表1。网络应用框架参考卡

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **晶圆类型** | **主要属性** | **产品** | **主要特征** | **指导方针/趋势** |
| **基于请求** | * 处理传入请求的控制器和操作 * 无国籍的 * 映射到URL的逻辑 | 支杆 | * 扩展Java Servlet API * 采用“模式2”方法 * 行动和行动形式 | * 分解成两个框架:Struts   行动框架(与网络工作合并)和Struts Shale框架(合并JSF)   * 主要的前端控制器实现 |
| 蜂箱 | * 元数据驱动的编程模型 * 网络界面页面流 * 控制 | * 必需的JDK 1.5 * 网络界面页面流基于Struts * 比春天的网络流量更成熟 * 授粉是一个基于Eclipse的集成开发环境 |
| 条纹 | * 轻量级表示框架 * 内置支持每个表单多个事件 * 透明文件上传 | * 非常适合在应用程序中构建向导窗体 * 索引属性支持 * 重用ActionBeans作为视图助手 |
| **基于组件的** | * 封装在组件中的逻辑 * 组件实例中处理的状态 * 类似于图形用户界面开发 | JSF | * 用户界面组件 * 用于处理状态、事件和服务器端评估的应用编程接口 * 2个JSP自定义标签库 | * MyFaces是一个开源实现 * 太阳微系统公司提供的参考实施 * MyEclipse企业工作台支持 |
| 挂毯 | * 构建页面的基于组件的结构 * 模板 * 表示组件的特殊HTML属性 | * 适用于利基领域 * 使用普通的HTML编辑器轻松编辑模板 * MyEclipse企业工作台支持 |
| 小门 | * 通过所见即所得的HTML工具模拟页面 * POJO数据豆 * 透明的状态管理，没有XML配置 | * Echo中的组件模型和特殊HTML属性的组合，用来表示Tapestry中的组件 * 使用POJO自动化服务器端状态管理 |
| **混合物** | * 基于请求的模型中的数据和逻辑流 * 处理动作和控制器的组件对象模型 * 中原始控件和可重用性的结合   两种框架 | 流行的 | * 无逻辑的HTML模板 * 统一组件模型 * 元编程 * 具有范围的多维对话状态管理 * 独立于语言的模板引擎 * 具有内容管理集成和版本控制的持久层 | * 使用特定于领域的应用编程接口和语言 * CRUD的自动生成 * JavaBeans实例及其属性的丰富动态元数据API * 与DWR的一体化 |
| **自指的** | * 公共服务的核心接口 * 集成组件和服务的可扩展主干网 * 控制模式的反转 | 龙骨 | * 多层结构:安全性、数据库抽象、消息传递、业务逻辑和用户界面层 * 角色和接口的严格分离 * 自动配置 | * 基于服务的体系结构 * 可扩展结构 * 开放平台 * 收集的反模式 * 利用Avalon框架 |
| 春天 | * 用于自动配置和连接应用程序对象的轻型容器 * 事务管理的抽象层 * AOP功能 | * 首选基线 * 在设计中加入适当的前控制器实现 * Spring IDE是Spring框架所使用的配置文件的图形用户界面 |
| **基于RIA** | * 丰富的互联网应用 * 客户端容器模型 * 有状态用户交互 | 除沾染洗消器 | * 在浏览器中从JavaScript调用服务器端的Java对象 * 动态生成基于JavaScript代码的Java类 | * 支持DWR基于请求和组件的框架 * 反向Ajax * 可分页和可排序的列表 |
| 回声2 | * 基于Ajax的渲染引擎 * 面向对象和事件驱动的用户界面开发范例 * 服务器推送技术 | * 面向事件的设计 * 支持模态对话框 * EchoStudio2是一个Eclipse插件，用于可视化   开发:表单编辑器和样式表编辑器 |
| JSON-  RPC-Java | * Java中的JSON-RPC实现 * 轻量级RPC JavaScript客户端 * 对象请求代理的基本功能 | * 轻量级JSON格式，而不是XML，以提高速度 * 远程过程调用协议 * 基本类型的透明编组/解编组 |

整合已经发生，并且趋向于变得更强，正如最近WebWork和Struts(行动框架)的合并，以及JSF合并到Struts (Shale)和Spring (JSF-Spring)中所证明的那样。

源源不断的新框架继续激增，特别是在不断发展的开放的RIA领域。几乎所有这些都是开源的，即使是供应商支持的

像Echo2这样的产品。显然开源是未来框架开发和增强的方向。

元框架的使用正在快速增长，正如Spring的采用所显示的那样。预计基于元框架会出现更多的集成和融合，这为即插即用各种组件和服务提供了良好的基础，例如

作为Freemaker。该结构趋向于变得更加灵活和松散耦合，允许竞争技术的轻松交换，以构建适合特定领域的最佳解决方案。

简单性将是一个框架在这个拥挤的空间中得以维持的关键因素。像Ruby on Rails和Laszlo这样的轻量级技术对企业计算环境中的“传统”编程框架提出了强有力的挑战。其他相关产品和新兴技术将影响web应用框架的未来前景，如XUL、通用界面、XAML、Ideaburst和Kanemea。

# 结论

web应用程序框架是分布式系统开发中的一个关键设计方面。本文从技术角度描述了与Java EE模型相关的各种网络应用框架和相关新兴技术。“web应用程序框架”的定义已经明确，因为这个术语已经被广泛使用，并且在不同的上下文中有着截然不同的含义。web应用程序框架的价值主张旨在说明框架如何提高应用程序开发的生产率和质量。阐述了web应用程序框架的设计理念。定义了一个全面的分类法来将各种软件框架和web应用程序框架分类。在作为商业和开源解决方案可用的几十个web应用程序框架中，主要的产品被调查，然后是选择指南和建议。构建了一个参考卡来总结web应用程序框架的关键方面。讨论了相关技术和未来发展趋势。

# 参考

1. 金融工程领域的框架:经验报告。in (eds)，《面向对象编程欧洲会议论文集》，Springer-Verlag，德国凯泽斯劳滕，1993年，第21-35页。
2. “开发决策支持系统的软件框架——决策支持系统开发工具分类中的新组成部分”，《决策系统杂志》，拉瓦锡，法国，2003，12(3/4): 271-281。
3. 元模式——一种捕捉可重用面向对象设计本质的方法。M. Tokoro和R. Pareschi(编辑)，《经济合作组织会议记录》，斯普林格-弗拉格，意大利博洛尼亚，1994年，第150- 162页。
4. 面向模式的软件体系结构:一个模式系统，威利，奇切斯特；纽约，1996年。
5. 《应用统一建模语言和模式:面向对象的分析和设计以及统一过程导论》，普伦蒂斯霍尔PTR，上马鞍河，新泽西州，2002年。
6. 开放交响乐，网络工作框架，[http://www.opensymphony.com/webwork,](http://www.opensymphony.com/webwork)2006.
7. Struts，Struts框架，[http://struts.apache.org，](http://struts.apache.org/)2006.
8. Apache，Beehive框架，[http://beehive.apache.org，](http://beehive.apache.org/)2006.
9. 条纹，条纹框架，[http://www.mc4j.org/confluence/display/stripes/Home,](http://www.mc4j.org/confluence/display/stripes/Home)2006.
10. JSR-127，沙威瑟面孔，[http://www.jcp.org/en/jsr/detail?id=127,](http://www.jcp.org/en/jsr/detail?id=127)2004.
11. JSR-252，JaveServer Faces 1.2，[http://www.jcp.org/en/jsr/detail?id=252,](http://www.jcp.org/en/jsr/detail?id=252)2006.
12. 雅加达，Tapestry框架，[http://jakarta.apache.org/tapestry,](http://jakarta.apache.org/tapestry)2006.
13. Wicket，Wicket框架，[http://wicket.sourceforge.net，](http://wicket.sourceforge.net/)2006.
14. RIFE，RIFE框架，[http://rifers.org，](http://rifers.org/)2006.
15. 龙骨，龙骨框架，[http://www.keelframework.org，](http://www.keelframework.org/)2006.
16. 弹簧，弹簧框架，[http://www.springframework.org，](http://www.springframework.org/)2006.
17. J2EE一对一设计和开发专家。英国Wrox，2002年。
18. DWR，DWR框架，[http://getahead.ltd.uk/dwr,](http://getahead.ltd.uk/dwr)2006.
19. 回声，回声2框架，[http://www.nextapp.com/platform/echo2/echo,](http://www.nextapp.com/platform/echo2/echo)2005.
20. JSON，JSON-RPC-Java框架，[http://oss.metaparadigm.com/jsonrpc,](http://oss.metaparadigm.com/jsonrpc)2006.