



开源关键技术选型指南

开源关键技术选型

当然好戏最终以轻量化的 EJB3.0 出世而收场，但是我们还是很有必要思考一下：我们开发者到底需要什么样的技术？这样才能在瞬间变化的发展潮流中坚持自己观点，而不是人云亦云，迷失方向。随着开源技术越来越成熟，一个稍有开发经验的人通过学习就可以用开源的产品和技术构建一套可用的系统。对于从事软件开发的人员，尤其是对 Java 或动态语言相关领域的人来说，“开源”也许是他们最喜爱的单词。每天，当他们坐在电脑前开始编程时，从所使用的 Linux 操作系统，到 Eclipse 集成开发环境，到编程所用的 Java 或 Ruby、Python 语言，“开源”烙印在每件工具身上。但是，很多时候我们需要的不仅仅是一个可用的系统，而是希望这个系统开发更简易、性能更高和扩展性更好等。这确实是一个令人头痛的问题。本指南很多地方都是点到为止，要深入了解相关信息的读者请借助参考资料、网站等自行挖掘。

Web 框架

开发框架的选择，始终是个仁者见仁、智者见智的事情。尤其是 Web 层的开发框架，数量非常多，而且各有特色，如：Struts、WebWork、Spring MVC、Tapestry、JSF、WebPage3.0……等等。框架的选择，就是看哪个框架最合适，从而减少开发的工作量，提高开发的效率和质量，并有效减少维护的工作量，最终达到节约综合开发成本，获取更多的收益。

- ❖ Axis、Struts、ColdFusion 和 WCF 的 REST
- ❖ 开放源工具填补了 SpringSource 应用程序生命周期的空白
- ❖ Skyway：使用 Eclipse，建模和 Spring MVC

❖ **Pierre Fricke: 别轻视 JBoss Seam**

Ajax

AJAX 开发与传统的 CS 开发有很大的不同。这些不同引入了新的编程问题，最大的问题在于易用性。由于 AJAX 依赖浏览器的 JavaScript 和 XML，浏览器的兼容性和支持的标准也变得和 JavaScript 的运行时性能一样重要了。这些问题中的大部分来源于浏览器、服务器和技术的组合，因此必须理解如何才能最好的使用这些技术。

综合各种变化的技术和强耦合的客户服务端环境，AJAX 提出了一种新的开发方式。AJAX 开发人员必须理解传统的 MVC 架构，这限制了应用层次之间的边界。同时，开发人员还需要考虑 CS 环境的外部和使用 AJAX 技术来重定型 MVC 边界。最重要的是，AJAX 开发人员必须禁止以页面集合的方式来考虑 Web 应用而需要将其认为是单个页面。一旦 UI 设计与服务架构之间的范围被严格区分开来后，开发人员就需要更新和变化的技术集合了。

- ❖ **Ajax 为什么可能成为 Web 服务的未来**
- ❖ **Oracle 架构师提倡 agnostic Ajax 方法**
- ❖ **Burton 报告: Ajax, RIA 的不二选择**
- ❖ **Ajax 技术足以成熟应对企业应用**
- ❖ **Ajax 安全指导: 用好的架构和更安全的 API 防御攻击**

动态语言

当关系数据库和 SQL 语言把数据管理的繁琐工作从开发者和操作者身上解放出来的时候，有人抨击关系数据库性能低下；当微软开始用 C 语言开发 PC 应用程序时，汇编语言的拥护者轻蔑的说，只有汇编语言才能发挥 PC 机的全部能力；当 Web 进军企业

计算时，一大批专家跑出来抨击 HTML 界面的呆板和 HTTP 协议的缓慢和幼稚。然而，所有这一切抱怨、质疑和信誓旦旦的否定，都被技术发展的大潮无情的吞噬，消失得无影无踪。原因很简单，所有这些声音，都是着眼于小，着眼于私，一叶障目，而不见 IT 技术发展的大势。IT 技术发展的大趋势永远都是要更快，更简单。

从这个角度来看待动态语言，结论就异常的情系。在今天这个时代、动态语言代表着更快更简单的技术大趋势，因此它将必然成为未来构建软件和互联网技术的主角。

- ❖ Ruby on Rails 为深入企业做好准备
- ❖ 理解 Web 服务客户端流行语言:Ruby
- ❖ SOA 结束了大型程序的年代 为 Ruby on Rails 敞开大门
- ❖ 最新升级版本 Python 语言正在推出

面向服务架构

面向服务的体系结构（service-oriented architecture, SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。

- ❖ 面向服务架构的商业现状
- ❖ 轻量级 Java 框架和面向服务架构
- ❖ 面向服务架构下的企业具体案例实施
- ❖ SOA 敏捷应对 Web2.0 时代变化

Axis、Struts、ColdFusion 和 WCF 的 REST

从十年前相对简单的 XML-RPC(远程过程调用)协议开始，Web 服务的架构和标准变得更为复杂了。有时，SOAP 似乎影响了开发者的想法，然而支持这些想法的开放源和商业结构日趋成熟。作为对这日益增长的复杂性做出的回应。REST 格式越来越为开发者所重视。我认为研究一下这些机构是如何在 REST 颇为大众所接受前响应这个潮流是非常有意义的。

不同程度的 REST 一致性

正如 Fielding 的论文中所叙述的一样，REST 格式强调资源以及获取该资源的 URI 地址，这和强调要在创建资源的运动中设置一个过程的 SOAP 和 XML-RPC 的定向是相悖的。REST 格式提议者坚决认为用简单 HTTP 方法建立的一致结构和开始让你创建 Web 可以解释的服务。

由 Fielding 和其它专家发起的 HTTP1.1 协议标准，RFC 2616 对于如何使用这些方法有着更清楚的建议。该标准强调 GET, HEAD, PUT 和 delete 要求必须是幂等的——这就是说许多相同的要求应该和单个的要求有同样的效果。许多应用忽视了 GET 方法的要求。

我在这里概括了 RESTful 服务应该如何使用 HTTP 方法。我发现大多数旧框架在使用这个格式的道路上才刚刚起步。

- GET –在一个专用表达（网络媒体型）资源中检索一种表现。
- POST –使用请求体中的数据改变资源。URI 验证了一个能够解决密封数据的资源。
- PUT –请求体建立或设置了一个资源的当前状态，该资源由 URI 所验证。
- delete –在服务器删除一个特定资源

·HEAD –检索一个资源的状态信息，比如最后以 HTTP 头部的形式验证的日期。意图为一个程序提供避免额外 GET 的方式，如果资源没有改变。

Apache 软件基金-Axis

最初由 IBM 公司创建的早期的工具箱 SOAP4J，就是人们所知的 SOAP 工程，在 Apache 软件基金安家落户，但是，这是在支持新出炉的例如 WSDL 标准的需求出现不久发生的，Apache Axis 在 2002 被重设计。目前的版本 Axis 1.4 还在使用中。

当第一个 Axis 正在开发中，Axis2 就开始被重设计了。谈到 Axis 工具箱，其还有极大的不可确定的空间。因为 Axis 和 Axis2 二者都在和“1.4 版本”同时使用。

我们研究仔细文档编制时发现，Axis2 支持 RESTful GET 是通过将 REST 格式 URI 重新格式化入 SOAP 而完成的。该 SOAP 要求被转发到标准 SOAP 要求。并和 XML 格式的响应一起进行数据处理，该 XML 格式的响应不包含标准 SOAP 外壳。这足够支持 XML 的 Ajax 格式使用丰富的互联网应用（RIAs），但是它不能处理一个应用程序可能需要的其它资源类型并且也不能提供 PUT 和 delete 方法的使用。

Apache Struts 是用 Struts 1 版本创建 Java 的 Web 应用程序的一个最受欢迎的开放源码框架之一。但是要想完全重新设计 Struts 2 还需要 REST 格式要求的支持。有一个 Struts 2.1.1 的“REST 插件”显然仍处于测试阶段，这使得在 Rails 格式下使用 Ruby 支持 REST 格式变得很容易。该方法把 URLs 映射为编码。

ColdFusion 框架

ColdFusion 是第一个（1995 年）Web 开发框架。现在归 Adobe Systems 所有，它的主要目标是建立由 ColdFusion 标记语言编写的网站所驱动的数据库。最新的第 8 版，可以处理查询格式 GET、POST 请求并且可以以 JSON 或者 XML 格式将数据返回到丰富的客户程序。但是，解释 URI 格式要求不受支持。ColdFusion 引擎可以在任何 Java J2EE 服务器或 NET 服务器环境下运行。

微软的 Web 服务工具包

在微软最早开发 SOAP Web 服务和 SOAP 协议的研究人员在不同的微软开发环境下都得到很好的支持。不像现有框架那样缓慢演进，Microsoft 建立一个新的框架。微软最

新框架 Windows Communication Foundation 或者 WCF 是用 Microsoft .NET 框架建立连接系统的一个广义的 Web 服务堆栈。其仍然强调支持 SOAP 和 WS-*标准但是为建立服务所使用的最新的(.NET 3.5)版本可以被配置为可以用 SOAP 或者 REST 格式所处理的版本。我们已经为支持 GET, POST, PUT 和 delete HTTP 做了准备。我发现这些例子强调 JSON (JavaScript Object Notation)的返回内容类型和 XML, 并为识别要求的内容类型和建立正确的回应通过机制。

结论

一般来说, 为了跟上新发展的步伐, 旧的框架在过去几年中已经历重大重新设计。这样您的旧框架, 可能会有创新, 让您一开始就拥有休息 Web 服务。但是你不能只关注这一个框架, 你也需要看看我在最近的文章中讨论的新方法。像 Jersey 和 IBM's Project Zero 以前方法中多余的东西。

(作者: William Brogden 来源: TechTarget 中国)

开放源工具填补了 SpringSource 应用程序生命周期的空白

在 Java 领域，对于管理应用程序的需求不断升温。正因为如此，SpringSource 收购了 Hyperic 这家性能管理工具供应商公司。通过这次收购，SpringSource 有望提供一套建立在开放源技术基础之上的完整应用程序生命周期工具箱，以便建立、运行并管理基于 Spring 之上的轻量级 Java 应用。

原 Hyperic 公司 CEO，现 SpringSource 公司产品管理部 CTO，Javier Soltero 说“对于大多数开发商和 IT 设计师来说，SpringSource 是创建 Java 应用的默认选择，对于众多 IT 运营专家来说，Hyperic 则是管理这些程序的默认选择。”“管理企业 Java 需要拥有通观整个栈、公司网络以及数据中心(其中包含了虚拟化和云计算环境)的高度洞察力，将开发和 IT 营运分开似乎不足以涵盖整个领域。

去年，SpringSource 收购了 Covalent，Covalent 是一家致力于 Tomcat 服务器管理的专业公司，目前 Tomcat 服务器已经并入到 SpringSource TC 服务器当中。Tomcat 的应用十分广泛，但是对于大多数数据中心管理者来说，它仍然需要管理软件的支持。

Hyperic 公司的收购为 SpringSource 监督 Web 服务器、应用服务器、数据库、消息队列以及物理平台、虚拟平台和云平台中部署的其他应用基础设施提供了一个管理套件。

SpringSource 公司工程设计和产品管理部的常务理事 Peter Cooper-Ellis 说“有了 Hyperic 之后，我们就可以管理并监督数据库和 Web 服务器了”。

分析师 Jay Lyman 和 451 Group 撰写了一篇文章，大意是：在目前的经济状况下，机构更倾向于采用开放源软件，特别是针对 Java 应用。Lyman 说“促使

SpringSource-Hyperic 合并的主要原因并不是开放源软件，而是因为开放源软件为这些供应商提供了一个新的出路：企业 Java 应用.....”

(作者: George Lawton 来源: TechTarget 中国)

Skyway：使用 Eclipse，建模和 Spring MVC

令人惊讶的是建模在追求 JEE 应用的应用开发团队中不被予以重视——这可能是更真实的草图设计式发展。但建模可以作为一种帮助实现良好的、可重复使用的服务手段。有些人认为这就是“SOA”。

在该领域，Skyway Builder 是其中一个产品。我们有幸与该公司探讨，基于 Eclipse 的 Skyway 软件为 JEE 应用的发展提供了一个模型驱动的方法。此外，它支持了讨论颇多的轻量级面向方面的 Spring 框架。

今年年初，该公司伪造与 IBM 达成了协议，将其 Skyway Builder Enterprise Edition 与 IBM 的 Rational Software Architect, Rational Software Modeler 和 Rational Application Developer (RAD) 相集成。而 IBM 的努力是自然的 WebSphere 为本，值得注意的是，这一协议使得 IBM 开发者有机会进入 Spring。

协同 IBM Rational Software Architect 7.5.1 的 Skyway Builder 可以帮助团队动员者 UML 进入 Spring 工作。这包括 Spring MVC 的脚手架功能，该功能使用户能够生成一个 Spring 的应用程序。

kyway 总裁兼首席执行官 Sean Walsh 表示“与其手工编写代码，我们创建了一个基于模型的方。该软件还允许开发人员做作为发展的出发点的‘脚手架’。这意味着基于开发人员的输入创建模式，然后生成可用于自定义的代码。脚手架已被 Ruby-on-Rails 方面的倡导者证明是非常受欢迎的。

(作者: Jack Vaughan 来源: TechTarget 中国)

Pierre Fricke: 别轻视 JBoss Seam

在 JavaOne 大会上, JBoss 公司产品主管 Pierre Fricke 并没有谈与 Red Hat 公司合并一事。他谈到了支持新的 Java EE5 的第一版 JBoss Seam, 并且他还谈到了 Seam 与 Ajax 的前景以及 JBoss 将加入 NetBeans 社区。

在 JavaOne 大会上, JBoss 公司产品主管 Pierre Fricke 并没有谈与 Red Hat 公司合并一事。他谈到了支持新的 Java EE5 的第一版 JBoss Seam, 并且他还谈到了 Seam 与 Ajax 的前景以及 JBoss 将加入 NetBeans 社区。

你能对 Seam 做一个明确的定位吗?它是否是轻量级的 Web 开发框架?

Pierre Fricke:不, 它不止是这些。Seam 不止是一个 Web 应用框架, 它是对 EE5 的进一步简化。在 EE5 中, 你依然必须写很多代码才能在 Java Server Faces 模型上搭建 EJB3 模型。并且, 你也没有工作流的元模型。因此, 你还要搭建很多不同的模型才能完成所有的应用。我们所做的基本上就是建立更加集成化的 EJB3 和 Java Server Faces, 然后与 jBPEL[Java Business Process Execution Language] (Java 环境中的工作流工具, 与 XML Web 服务环境中的 BPEL 不同)一起协作。因此, 你可以完成页面流并通过该程序框架自动进行过程类型处理。

Seam 和 Spring 比较起来如何?

Fricke:Spring 是这个领域人们讨论的比较高的东西。Spring 是一个简化的、轻量的模型, 而我们的当然也是轻量的, 也相当简单。但 Spring 是一种类似无状态类型的应用, 而 Seam 能做无状态应用, 也能做有状态应用, 能做交易型应用和工作流应用。事实上, 我们构建了更丰富类型的主流应用, 我们比 Spring 走的更远, 处理这些类型场景的架构也完全不同。因此, 尽管我们具有轻量 Web 应用框架的所有特征, 而且编程简单, 但是我们的模型要更加强大。

Seam 如何操作 Ajax?

Fricke:现在, 你可以用 Seam 做很多交互型的应用, 但我们还有一些工作要做来确定 Ajax 能够为现有的工具增加多少能力。我们还没有完成这项工作。因此, 这是一件与标准兼容或模型兼容的事情。但我认为我们还不适合投入到在 Seam 中 Ajax 的讨论中。这件事还需要斟酌, 但它的确为交互型应用增了不少功能。而现在, 你需要研究一下 Ajax 会带来哪些价值, 不会带来那些价值, 然后再来确定下一步事情。如果有意义的话, 我们当然不会不做。但我们认为 Seam 编程模型在理解 Web2.0 内涵的上还有很长的路要走, 既如何制作更加健壮、交易型、有状态、互交互的应用。

你觉得 Web2.0 是什么?

Fricke:我认为它只是对这种新的, 更加可交互的的应用的一个术语名称, 它是 Web 环境的进步。

JavaOne 大会上, Jboss 宣布你们会加入 Netbeans 开源社区。什么让你们选择 NetBeans 而不是 Eclipse 呢?

Fricke:我们并没有不选 Eclipse。我们只是把 NetBeans 加入我们的社区, 我们也加入 NetBeans 的社区。因此, 这不是一件二选一的事情。我想强调的是, Eclipse 也是我们的战略。JBoss IDE 以 Eclipse 为基础, 此为 IDE 的贡献都源自 Eclipse 插件。所以, Eclipse 依然很重要。而 NetBeans 也同样重要。它采用的技术很精致, 尤其是在处理 Web 服务方面。我们是一个开放的社区, 我们欢迎所有人, 越多开放工具越好。因此, 我相信我们能够带来一个 JBoss 生态系统并且开放它, 让 Eclipse 和 NetBeans 开发人员都来使用。这是一个双赢的大好事。

(作者: Rich Seeley 来源: TechTarget 中国)

Ajax 为什么可能成为 Web 服务的未来

一项崭新的基于 Web 的重要技术多年以来一直默默无闻，但是突然间变得极受欢迎，这主要是因为 Google 以及其它网站上建立的程序。被称作 Ajax，是一项用于创立丰富的以网络为基础的程序的程序设计技术。尽管它们以浏览器为基础并且使用的资源极少，它们的工作原理和桌面软件极为相似。

分析家说，同样的技术将最终为 Web 服务和 SOA 前端提供动力，事实上，他们已经开始这样做了。虽然，未来的一切无法预见，但是，一些人认为 Ajax 最终会在大多数 SOA 和人类所面临的 Web 服务中形成主要的用户界面。

本周的两部分专栏对 Ajax 对于网络服务和 SOA 的意义进行了考察。第一部分专栏探讨了什么是 Ajax，它是怎样工作的，以及它为网络服务和 SOA 所提供的效益。第二部分专栏研究了 Ajax 是否有可能得到广泛的应用，以及其在广泛应用的途中将要遭遇的挫折。

Ajax 如何运转

我们首先要认识到，Ajax 不仅是一项单纯的技术。它的使用并没有受到标准和标准体的支配。尽管最近它很热门，但并不是一个十分新颖的话题。

Ajax 代表异步的 JavaScript 和 XML，它使用几种技术建造丰富的基于 Web 的程序（这些程序的工作原理很像桌面软件），这些技术包括 CSS, JavaScript, XHTML, XML, XMLHttpRequest, Document Object Model 和 XSLT。该程序包括了我们希望在桌面软件中获得的互操作性和快速响应时间，不像 Web 服务那样明显地反应迟缓，工作停滞。

那么 Ajax 是如何施展其魔力的呢？在一个典型的非 Ajax Web 服务中，浏览器中的操作要求将一个 HTTP 请求传送到一个 Web 服务器。服务器进行诸如检索数据或者其它任务

的数据处理，然后把结果和一个 HTML 网页一并返回给客户（Web 浏览器），当请求双向进行传送时，服务器进行数据处理或撮取，用户在一旁等待。

但是，对于 Ajax 来说，一切都在改变。当用户浏览网页时，JavaScript 编写的 Ajax 引擎就会加载数据，从此以后，用户便和 Ajax 引擎相互作用，甚至在请求和回应引擎和服务器之间往返之时，二者一直都在相互作用。

这些非常重要，正如引擎本身可以执行无需向服务器调用请求的操作，它同时也可以编辑内存中的数据，完成确认数据等一系列的操作。因为并没有涉及服务器间的通信，一切都在瞬间完成。——其工作原理和桌面程序相似。

结果：产生了一个丰富的，和桌面程序类似的程序，但是该程序是在 Web 的浏览器上工作。

这里有许多 Ajax 网络程序。Google 杠杆作用像 Gmail 和 Google Local（以前的 Google Maps）这样的程序时，不断地开拓进取。A9 也在使用 Ajax 网络程序，将来会有更多的人使用它。

对于 SOA 和 Web 服务这些意味着什么？

对于 SOA 和 Web 服务这些意味着什么，考虑到主要是用户层的服务使用 Ajax，Ajax 会真正在企业找到归宿么？

ZapThink 高级分析师 Ron Schmelze 说，是的，尽管这个过程相当缓慢，但是 Ajax 最终会在企业找到归宿，Ajax 将最终取代那些和 Web 服务以及 SOA 绑定的企业门户网站。

他说，“人们将用 Ajax 建立的丰富用户代替基于 Web 的门户网站”，“因为它们能提供更多的功能”你会在企业内部以及企业和其它客户相互作用时发现它们。

他强调，银行网站已经开始使用 Ajax 程序，以方便客户执行金融服务。他说“他们希望尽量像使用桌面软件一样使用 Ajax 程序”。

Schmelzer 也看到 Ajax 的使用不断增长，因为他注意到了这一趋势。他在 SOAs 说，自己看到“一个新兴的混合程序——它既不是一个 Web 浏览器，也不是一个桌面程序。它们将二者相结合，在享受用户相互作用所带来的好处的同时，使用分布式程序。但是，这些程序遍及整个网络，并且容易部署。

他相信未来会使用 Ajax 来建立这些程序

Kevin Hakman 长期以来一直参与和 Ajax 相关的工作。他是 General Interface（通用接口）的创立者，General Interface（通用接口）是一年前被 Tibco Software 公司购买的 Rich Internet Application 开发工具供应商。他相信 Web 服务和 SOAs 的前途命运与 Ajax 紧紧的联系在一起。

他认为，“Ajax 非常重要，因为它准许人们使用自己的浏览器直接与 SOA 和 Web 服务相互作用，这是人们使用 Web 服务、获取 SOA 资产的最佳方式。”他说，目前 Web 服务和 SOA 在机器对机器层面表现良好，但是当人们需要和服务相互作用时，在建立可用接口方面就会出现問題。他认为 Ajax 就是解决问题的方法。

但是，现在 Ajax 在 Web 服务和 SOA 中并未得到广泛应用。

Schmelzer 说，“Ajax 目前还处于开发阶段，需要提升到业务层面”。

如何将其提升到业务层面呢，这一切有可能实施吗？这将是我们的下一部分专栏所要探讨的话题。

(作者: Preston Gralla 来源: TechTarget 中国)

Oracle 架构师提倡 agnostic Ajax 方法



Enterprise Web2.0 不只是给老化的、传统应用添加一个漂亮的 Ajax 外观，实际上要比这复杂的多，Oracle 公司的工具和中间件的首席架构师和副总裁 Ted Farrell 说。

企业研发人员所面临如何将传统应用和多种新的 Web2.0 技术混合这一复杂问题，需要一个“实现了 agnostic 技术的 UI 框架”来解决，Farrell 在加利福尼亚州的 Santa Clara 举办的 AJAXWorld 今日开放中的一个会见中，在他的 keynote (指具有电影效果的演示文稿) 之前先发表了如上言论。

他的 keynote 的题目是，“为何用于企业的 Web2.0 远不止于翻修门面，”，Farrell 将解释他所关注到的、与如 Amazon 和 Google 这样的 Web 用户公司中所做的事情相反的，在企业中构建 Ajax 类型的界面的差异。

“如果你关注用户领域，问题简单的多，环境也简单的多，你可以创建你在企业领域中不能创建的功能，” Farrell 说。

对于 Google 和 Amazon 这样的面向个人的公司而言，用户界面是头等大事，所以他们引入了在 Web2.0 领域身怀绝技的程序员，他说到。

然而，在事务可能是构造自动化部分的企业领域，Farrell 说企业研发人员没有典型性的“bleeding edge” (描述有一定风险的新生事物，如可能带来经济灾难或只是昙花一现的新事物) 编程技巧。

在用户 Web2.0 应用和企业应用之间的另一个不同之处在于，对于那些为 Ajax 之前的领域所设计的传统应用，两者所能做的范围不同，Farrell 说。

“很多人讨论 mashup (一种 Web 应用程序的名称) 和 Ajax，在它们之上创建用户界面来将所有的信息聚合到一起，以这种方式将所有的一切构造成网络服务，” Farrell

说。”这在企业领域不一定可能，因为很多传统系统并不是用这种方法来分解数据，或者由于 UIs 太复杂以至于实际上不能将两个分开。在这些情况下你通常需要在界面中使用如 portlet 这样的工具。在 Ajax 领域，访问 portlets 的 JavaScript 和数据共享是很难的，而且在很多情况下这实际上是不可能的。 “

他说企业研发人员告诉他虽然他们能够创建从传统后端系统中读写数据的网络服务，他们仍然被陈旧的 UI 所困住而无法进行下去，因为这些旧的 UI 使得到 RIAs 的转换变得困难。在他们开发 RIAs 的过程中，他们发现一旦需要将一个不同的 Web2.0 技术，例如 Flash，添加到已有的基于 Ajax 的界面中，他们就得重写 UI 以包含此技术，他表示。

“我们所发现的成功之道以及我们所推荐的方法是一种实现了 agnostic 技术的 UI 框架，” Farrell 说。“我的意思是通过这个框架，能够在不管具体内容是如何实现的基础上，混合和搭配这些内容以创建 Web 界面。因此我能创建一个 Web 界面，访问 JavaScript Ajax 表来与后台数据库交互的。但是在同样的界面中我还能访问一个 Flash 圆形分格统计图来与后台交易智能系统交互并显示我的交易统计片段。然后还在这个界面中访问运行 PeopleSoft、包括私有语言工具的 portlet。我可以在一个简单的用户界面中串连起这一切。作为一个开发人员，目标就是不被这三种独立的技术——用于 Ajax 的 JavaScript、用于 Flash 的 ActionScript 或者 PeopleSoft 工具——所牵制就能完成这一工作。”

Farrell 说 Oracle 定型在基于 JavaServer Faces (JSF) 的 RenderKit (用于表示层的工具包)，这允许已经学会了 JSF 的研发人员将完全不同的组件装配到一个 Web2.0 类型的 mashup 中。

“他们处理组件的方法，而不是他们实际上实现的方法，”他解释道。“由 RenderKit 来承担对不同技术的调配。因此我能在一个页面中丢弃一个可能是用 Ajax 实现的组件，我也能在一个页面中丢弃另一个可能是用 Flash 实现的组件。研发人员从这些技术中抽象了出来。你不需要教你的开发人员学会 JavaScript 或者 ActionScript 来做这些事情。而且我能用同样的技巧混合和搭配所有的一切。”

用这种方式，如果出现一个需要整合到 RIA 的新技术，可以扩展 agnostic 框架，而不需要重写 UI，他说。

(作者: Rich Seeley 来源: TechTarget 中国)

Burton 报告：Ajax, RIA 的不二选择

犹他州 Burton Group Inc 公司的首席分析师 Richard Monson-Haefel 在最近发表的一篇报告中这样写到，Ajax (Asynchronous JavaScript+XML) 逐渐成为网络开发人员的主流工具，它被定位为用来代替富因特网应用程序 (rich Internet application, RIA) 技术的技术集合。

犹他州 Burton Group Inc 公司的首席分析师 Richard Monson-Haefel 在最近发表的一篇报告中这样写到，Ajax (Asynchronous JavaScript+XML) 逐渐成为网络开发人员的主流工具，它被定位为用来代替富因特网应用程序 (rich Internet application, RIA) 技术的技术集合。

报告中建议，正在考虑选择 RIA 的公司应该开始使用 Ajax 技术，因为 RIA 不支持 Adobe Flash、Java applets、微软公司发表的面向多平台网络浏览器的画面绘画引擎 (Windows Presentation Foundation/Everywhere, WPF/E)、Mozilla 公司的基于 XML 的用户界面标记语言 (XML User Interface Language, XUL) and 可缩放矢量图形 (Scalable Vector Graphics, SVG) 技术。不过，Monson-Haefel 预计需要复杂动画的应用程序也会选择使用 Flash，因为他认为 Ajax 不太可能发展到提供动画功能。

“与其他 RIA 技术相比，使用 Ajax 的最大优势在于它能够与超文本标记语言 (Hypertext Markup Language, HTML) 进行无缝集成，” Burton 公司的分析师说。“不再被分为一包或一箱组件，Ajax 可以与超文本标记语言 (HTML) 连接融洽，不需要重新实现内容，允许在现有网站上增加图形用户界面 (GUI) 功能。”

据 Monson-Haefel 的报告，Ajax 技术的另一个优势在于它可以用于任何支持超文本传输协议 (HyperText Transfer Protocol, HTTP) 的应用程序平台。他注意到 Ajax 与 PHP、Perl、Active Server Pages for .NET (ASP.NET) 及 Java 2 Platform, Enterprise Edition (J2EE) 兼容。

Burton 报告“Ajax:一种 RIA 技术”，介绍说很多组织通过各种方法应用 Ajax，尽管 Ajax 中涉及的大部分技术产生于上世纪九十年代，但是现在来看用于 Ajax 开发的商务工具仍不成熟。

尽管报告的主要目的是为企业介绍 Ajax 技术，使他们初步了解 Ajax，不过报告中已经提到许多复杂网络开发人员已经开始使用 Ajax。随着网络复杂度的增加，Monson-Haefel 估计 Ajax 会与 Adobe Flash 产品在动态内容和动画方面进行更多的整合。

他认为 Ajax 并没有 Adobe Flash 技术成熟，而且一致被看作是一种复杂的 RIA 开发方法。Ajax 依赖于不是所有 Java 开发人员都熟悉的 Java 脚本，也就是说只有高级网络开发人员才是 Ajax 技术的主要使用者。但是，对于启用 Ajax 的框架就不同了，例如 Java Server Faces (JSF)、Rails 上运行的 Ruby、Struts、ASP.NET 和 PHP，现在即使不是 Java 脚本程序员也可以简单地使用 Ajax 技术。

甲骨文公司的首席架构师、工具和中间件部副总裁 Ted Farrell 说，在 JSF 上，提供商可以取得进步。他说甲骨文公司、太阳公司和 IBM 现在都采用 JSF，这样开发人员可以在 Ajax 应用程序中几乎不使用 Java 脚本编码。

“作为提供商，JSF 为我们节约了大量设计时间，” Farrell 说。“我想在过去的两年中，我们在制作组件方面更加智能化，我们使组件对用户来说更简单，而不是复杂化。所以，组件的越来越好和工具的越来越好减少了人们开始成功使用 Java Server Faces 应用程序的障碍。”

Farrell 认为 Ajax 技术是 RIA 技术的一种选择，而 JSF 则是大多数开发人员使用 Ajax 技术的方法和途径。

“开发人员不需要学习任何新东西，”他说。“他们还和以前一样，在 JSP 和 JavaServer Faces 中编程。不过他们已经可以创建基于 Ajax 的应用程序。”

然而，ZapThink LLC 咨询公司的高级分析师 Jason Bloomberg，仍然对 JSF 工具持怀疑态度。

“我们已经看到其他提供商大肆吹捧 JSF 的优点——ICEsoft 公司除外，” Bloomberg 说。“这些提供商也为开发人员提供创建 Ajax 应用程序的 JSF 方法，使开发人员不需要挣扎于 Java 脚本代码之间。JSF 的另一个优点就是它可以解决跨浏览器问题，但是美中不足的是它是一种仅被 Java 开发人员所用的 Java 工具。在我看来，这些以 JSF 为基础的方法都是在小范围使用的工具，而那些与语言无关的(Language-Neutral)，像 Nexaweb 和 JackBe，以及 Adobe 和 Microsoft 才是 IT 行业的主流工具。”

(作者: Rich Seeley 来源: TechTarget 中国)

Ajax 技术足以成熟应对企业应用



Edwin Aoki, AOL LLC 的技术人员. 上周三他在旧金山举行的 Ajax 经验讨论会上做了名为“真实世界中的 Ajax”的演讲……

Edwin Aoki, AOL LLC 的技术人员。他在去 AOL 之前曾在苹果公司和 Netscape 工作，对于用户界面开发经验丰富。他谈论了 Ajax 和丰富的因特网技术是怎样发展成熟到适用于现实世界。基于他的论点，SearchSOA 与 Aoki 讨论了他在 AOL 的最近的工作，这包括执行会议的一个热点话题，“渐进增强”，它是由用于有限的浏览器的最简单的 HTML 开始的，慢慢成长为高端 Ajax。这是一种服务于不同浏览器包括屏幕浏览器和移动电话的方式。他也谈论了关于 Ajax 和 JavaScript 要达到完全成熟所要做的，包括身份管理和支持数据保存。他说这些技术发展的下一阶段是他和 Ajax 领域的其他人员在 Ajax 经验方面将要面对的问题。

在 AOL 公司你在 Ajax 方面正在做些什么？



Edwin Aoki: 我们拥有一个完整的团队，曾有段时间致力于基于 web 的动态的应用。一旦你有了自己的很好的想法，选择了你想用的任何工具，并落实了你的观点，那么能够将这些想法在真实世界中实施和部署的快乐时刻就开始了。

在真实的世界中你有遇到什么特殊的问题吗？

Aoki: 当我们谈论 Ajax 的时候，很多次的话题都是围绕性能和可访问性，在这里可访问性不只是定义为围绕屏幕浏览器和对在视觉上削弱的情况的支持等方面，除此以外还被定义为内容和服务可能会达到最广泛的用户，最多的浏览器的数目和最多的设备数目等方面。

这的确是真实的，特别是当对象可能不是人的时候。对于许多那样的应用来说搜索引擎是一个巨大的对象。Ajax 为此造成了一些特殊的挑战。

你正在做很多 Ajax 的工作吗？

Aoki:我们在 Ajax 方面拥有广大的支持。我们已经用 Dojo [开源的 JavaScript 工具包]完成了很多工作。我们有很多使用 Ajax 的产品。我们致力于 Dojo 工作已经有段时间了。我们以前的一些产品，包括我们已经发布的 MyAOL，使用了 Ajax 来传入内容流或提高负载体验。我们还利用 Ajax 做渐进增强。

什么是渐进增强？

Aoki:它是这样一个概念;采用以 XHTML[可扩展超文本标记语言]和微格式形式的原始语义内容通过一个提供给简单的 CSS[串联式样式表]或性能较差的浏览器的层来显示，然后在那基础上用 JavaScript 和 DOM [文本目标模型]创建更多的交互，达到可选式的增加和删除元素。

你通过渐进增强得到了什么？

Aoki:这个模型相较于传统的功能衰减的模型而言，优势在于你可以利用那个可访问性的宽广的概念创建更容易访问的站点。它们甚至可以在 JavaScript 关闭的情况下工作。他们也可以在移动浏览器上工作。即使你有丰富的媒体用户也并不会感觉到有什么不同。他们获得了对他们的输出模型来说最好的体验。而搜索引擎知道怎样能搜索到这个原型。这个帮套 Ajax 原型没有将它们分离出来。

所以它帮助搜索引擎找到你的内容？

Aoki:是的，事实上在我今天所做的演示中我展示了我们的视频集线器的一个老版本，它很大程度上基于 Ajax。我们有好几万的完整版的电视节目，但是如果你去 Google 搜索，会发现根本就找不到，因为 Ajax 接口阻止搜索引擎查找到所有内容。

你是怎么做的呢？

Aoki: 我们使用一些渐进增强的原则当然也有一些设计技术来浏览和重新设计它。我们回退回去做了很多 Ajax 工作，在没有妨碍主要内容的情况下使它变得不那么以使用 PE 为核心来驱动交互作用。

渐进增强模型做了什么使得搜索引擎可以找到内容？

Aoki: 以前的情况是我们有非常多的内容是通过滚动条 DIV 罗列的。我们用 XHR[XMLHttpRequest] 从后台将内容放入 DIV。因为我们有点匆忙的写作这个，当搜索引擎通过时它不解析 JavaScript，所以它所能看到的就只有这个空的 DIV 而从来不会进入。

所以搜索引擎对内容是视而不见的？

Aoki: 没错，因为我们将内容通过 Ajax 管道动态地放入。

现在它工作的怎么样？

Aoki: 现在我们所做的是所有的内容以原始 HTML 的形式存在，如果你是一个搜索引擎或是个文本浏览器或者你关闭了浏览器中的 CSS 和 JavaScript，那么这就是你所获得的内容。但是它是一个凭借自身条件的非常易读，非常可视化的页面。然后我们在它之上应用一系列样式单，这样你就得到了一个更好的显示。如果你是一个具有电力不足的 web 浏览器的移动电话，那这就可能是你所看到的。如果你关闭了 JavaScript，那你看到的也是这个显示。如果你在低于 IE7 的浏览器上关闭了 ActiveX 控件，那么你不能得到 XML HTTP 请求，你看到的会是这个显示。如果你将这些都设置为可用的，我们利用 JavaScript 来操纵 DOM 以便你得到 flash，图片和所有的那些东西，但是他们是体验的增加而没有减少过去的体验。

那么渐进增强是否是一个具有多种多样的能力的更好的获得浏览器的方式呢？

Aoki: 当我们谈论它的时候它总是引起一些小的争论, 因为它和人们思考功能衰退的方式是相反的, 你首先积累全面的体验, 然后指出如果你没有完全的浏览器能力那么什么会失败。但是结论是渐进增强是非常有利于搜索引擎的。很显然, 它确实有利于对解析 JavaScript 有困难的屏幕浏览器。而且它也表现出对性能很有帮助, 因为它产生了更为清晰的标记和隐蔽性更好的 JavaScript, 这是因为你的 JavaScript 与你的内容之间是更为独立的。它给予了你一个很好的编程模型。

AOL 的开发者主要使用 JavaScript 吗, 还是你们也用 PHP 和 Ruby?

Aoki: 我们每种都用一点。我们的一些产品是构建在 Ruby on Rails 上的。我们还有一些 PHP 的内容。我们的团队倾向于选择他们觉得最适合他们特定的工作的工具。

在过去的一年, 有些谈论说 JavaScript 是 Ajax 的脆弱的一环, 你的体验是什么?

Aoki: 它确实有改进。在 IE7 和 Firefox2 还有 Safari3 中完成的工作使得为 JavaScript 提供的完整支持好了很多, 但是在我今天的演示中我们讨论了浏览器厂商所作的工作, 去年同行已经基于像

Dojo 和 Prototype [JavaScript 框架] 还有 Script.aculo.us [JavaScript 库] 这些东西做了很多工作, 但是它还没有完成。还有一些基础部分是我们感觉与一个功能完善的 web 开发环境相比所欠缺的。我们想和同行还有其他与会的人们共同合作来充实这些部分。

你希望看到 JavaScript 的哪部分得到改进?

Aoki: 像身份管理之类的。我们有个做身份管理的 API 叫做开放验证。我们认为它功能很完善。它集成了 OpenID [单点登录技术], 但是它没有构建到任何框架中去。它并不和 Ajax 所接受的档案模型一起工作。我们没有做这个的框架。通信也一样。我们有一个非常好的基于 web 的即时信息平台, 我们正在将其扩展为一个基于 web 的数据缓存平台, 但同时你还是要选中它, 把它放到一起。正在构建的块也是相当粗糙和原始的。我们想和做

Dojo, Prototype 和 Script.aculo.us 方面的人合作创建一个环境, 你可以依赖这个环境中的服务, 并且你可以只关注你的应用要做的事情而不用关心存储, 身份或是怎样向前向后传递数据。这就是 Dojo 和其他框架在客户端所做的工作, 但是我们认为下一个阶段就是使它们和我们自己和其他人提供的一系列后台服务一起工作。

那就是你认为的在 Ajax 成长方面的下一个阶段?

Aoki: 它如此重要的一个原因就是它已不只是关于建立新的 web 应用。同样的模型将用于桌面部件和像 Google 那样的集合环境。那是 Nintendo Wii 的开放环境。它是适用于移动的唯一开放环境, 无论是掌上 IE 或是 Nokia SE 还是 Apple iPod。一旦你们了解了 Ajax 是第一种适用于这些平台的开发环境, 那么你们立刻就达到了一个共识: 它将开启人们想将这些后台工具和安全, 节奏和以便人们可以获得关于他们的应用的分析而构建在内的测量系统结合起来的想法。那就是我的主题。事实上它已经无处不在。我们已经走了很长的路。我们在 AOL 遇到并努力解决很多这个领域中的每个开发者都遇到并努力解决的相同的问题, 但是目前我们还有很多的工作要做, 也有很多的机遇在等待。

(作者: Rich Seeley 来源: TechTarget 中国)

Ajax 安全指导:用好的架构和更安全的 API 防御攻击

Andrew van der Stock, 他写了开放 Web 应用安全项目 (OWASP) 指南中关于 Ajax 的新的章节, 这一指南是为了构建安全 Web 服务, 因此, 本文主要讨论如何预防 Ajax 应用受到攻击。

Andrew van der Stock, 他写了 OWASP 指南中关于 Ajax 的新的章节, 这一指南是为了构建安全 Web 服务, 因此, 本文主要讨论如何预防 Ajax 应用受到攻击。

你写到在使用 Ajax 时需要仔细考虑架构、服务器端访问控制、状态管理以及严格验证。我们已经讨论了架构。那么你可以谈论一下服务器端的访问控制么?以及它是如何帮助 Ajax 安全的呢?

Andrew van der Stock 说:访问控制是由以下几个组件所组成的:

身份验证(authC)——你是谁?

授权(authZ)——你有足够的权限做某件事情么?

审核——对所有访问的登录进行安全保护, 以便及时确定稍候会进行什么操作

身份验证的强度是与初始身份和凭证类型(口令、令牌等)这些因素直接相关的。显而易见, 如果事务处理的级别越高或者说受攻击的风险越高, 那么需要验证的程度也就越强。强验证需要服务器端的基础架构, 比如访问控制服务器或者智能卡的 PKI。

授权是关键。许多应用程序的授权模型性能很差, 缺少任何授权验证, 以至于允许调用程序去执行那些有特权限制的操作。如果一个 Ajax 应用程序使用客户端授权, 那么就可以在在一定程度上预防攻击, 尤其是在没有服务器端控制的时候。攻击者只能够改变或者删除在磁盘操作监督程序(DOM)中的授权代码以及与安全有关的令牌, 比如一个 admin 标记。

对 Ajax 应用程序而言,唯一的安全方式就是使用服务器端的 authC/authZ 验证和审核。如果存在 Ajax 路径和普通 Web 应用路径这两种路径的话,那么采用服务器端验证审核的方式,就是对任何一种路径都不进行优先处理,并且是在服务器端对安全进行维护而不是这两个地方分别进行。

关于状态管理和客户端的安全状态存储,开发人员需要知道的哪些知识呢?

van der Stock 说:客户端没有安全状态这样的概念。每次收到服务器端的数据时,你都必须进行重新验证。开发人员不应该向客户端传送敏感数据,比如授权令牌、数据库口令或者访问级别,除非目的就是为了显示这些信息。返回这样的敏感数据更是个糟糕的方法,因为如果不先通过验证的话,返回永远无法被执行。

最后,发送所有数据就变得越来越通用。

强验证是由哪些内容组成的呢?

van der Stock 说:大多数用户不会故意和你过不去,但是有些会。强验证包括:

对于字符串:最大长度以及对可允许的字符的筛选。

对于整型数字:验证范围和符号。通常,缺省情况是使用“unsigned int”而不是“int”。

对于数组,比如下拉列表和 radio 组,使用简单的数值(“1”..”2”..”3”)作为选择项而不使用数据本身。这种情况下的完善的例子包括:含有信用卡号码的下拉列表菜单。不要使用信用卡号码作为返回值。因为这样会导致对号码的直接篡改。取而代之的是,选择时采用简单索引值(1, 2, 3, 等)或者使用账户号码的一部分作为显示选项。

对于复选框和布尔型值:强制只有 true 或 false 选项。

创建可以被验证的新复合类型,比如 Zip 代码等,这是由于 regex 类自带的语法规则。

否定验证与病毒定义的原理非常相似。要保持与新的攻击方法同步是很困难的。在过

去的大约 20 年内，采用黑名单这种方法来保护信息资源确实不断的遭遇很严重失败。开发人员对采用黑名单方法说“不”，这一点是至关重要的。

筛选是唯一的可能安全的验证方法。采用黑名单(否定验证)进行常规检查可能会导致接受某些恶意输入，因此，采用测试并且阻止数据的方式要好得多。

你上面着重讲述的步骤可以使 Ajax 开发人员领先于黑客么？

van der Stock 说:通过做这些基础工作——数据验证、好的架构、更安全的 API (没有理由对 SQL 入侵) 等等，你的程序就正在被很好地保护。那些恶意的开发者都是机会主义者，假如在你的应用程序上开发一个简单的攻击程序要比在别人的程序上多花 20 倍的时间的话，那么攻击者肯定会把别人的应用程序当攻击目标。我可以毫不虚伪地说:没有一个应用程序是“安全的”。这就是为什么防攻击者如此艰难的原因——我们必须防止任何攻击，而且我们到现在为止还没有预算和时间来针对每一个漏洞写测试代码，包括现在已知的潜在漏洞或者是还没有被发现的漏洞。

许多开发人员完全没有意识到他们需要学习这些知识。参与 OWASP 本地章节写作的人员已经是我们这个事业的皈依者了，而且他们知道的知识可能比我还多。相反，我倒很想与业务人员——创造新软件的架构师和工程师们——进行探讨。他们是重要的可启发的生力军，他们也是我们这个事业的皈依者。

安全研究人员，比如我，做的基础性研究比那些黑客多，但是黑客在广泛应用的软件上找到漏洞就可以获得很高的报酬。他们和我们使用相同的技术来识别并利用这些漏洞，某些情况下还会创建自己的程序区——SAMY 蠕虫病毒就是一个例子。

我希望更多的 Ajax 的脆弱性和使用上的问题能够显现出来，而且我也希望研究人员能够找出更多“新的”需要被保护的漏洞。不是这些漏洞不存在，而是我们还没有发现并描述它们。希望我们能够在那些零时攻击者之前先找到这些漏洞。我们的研究以及开发人员能够——或者更有可能，无法——对一个安全的应用程序实现必要的控制，这两者之间是有时间延迟的。

Andrew van der Stock 是澳大利亚 Web 应用程序的主要研究人员。他是 webappsec 的仲裁者，曾经帮助组织墨尔本地区 OWASP 章节。Van der Stock 正在带领编写 OWASP 指南 3.0 版本——其中包括关于 Ajax 的新章节，该指南是为了构架安全的 Web 应用程序。

(作者: Colleen Frye 来源: TechTarget 中国)

Ruby on Rails 为深入企业做好准备

5 月 17 号到 20 号在波特兰俄勒冈会议中心举行的 RailsConf，让 O'Reilly Media Inc. 和 Ruby Central Inc. 的联合出席者提前了接近三周的时间以结束注册。虽然这样的会议可能还会有很多，但该会议确实做了一些足以让大家兴奋不已的事。

该会议谈论了 ‘buzz’ 和与基于 Ajax、Ruby 和 Rails 开发环境以及与开放源相联系的令人兴奋的东西。这可能会是近期即将发布的还在生成环境中的产品，将加强已经拥有很强吸引力的 Ruby on Rails。我所在意的是这个技术在开发者社区中所包含的兴趣和样式。

生产应用上的一个突然上升是引起我们关注的一个重点，现在的使用和 Ajax 样式的开发路径一起的 Ruby on Rails 编写。Rails 是一个完全集合，使用 Ruby 编程语言建立的开放源网络应用框架。Rails 的要素指标的列表如下：

1. 迄今为止 Rails 已经下载过 933972 次，而在最近一年中，它就下载了 700000 次。
2. Rails 上的 Ruby 将和下一版的 Apple's OS X operating system (Leopard) 一起发布。
3. 显著应用数目的增加和大流量的网络站点都是将 Rails 作为其一项基础；这些站点包括 Backpack, Twitter, Jobster 和 Revolution Health。

关于这个兴趣点最常引用的原因就是结合改进了开发的速度(报告说相当于提升了 10 x 加速度)和开发者的经验，开法律和更加丰富的应用界面及低成本的结合归因于它的开放源属性使其对个人开发者更具有吸引力。它还吸引了越来越多的来自于新兴解决方案提供商和希望推进人员效率的商业 IT 执行者的注意力，加速了新方案进入市场的时间并减少了开发成本。

另一个使用 Rails 更快、更容易创建 Web 应用的好处是其允许对于交互式网络站点的更快的更新和修改。在开发者中的声望意味着它们在交互式 Web 站点上拥有更快的更新和改变信息的能力——在今天的环境下是一个重要的有竞争优势的优点，因此更有活力和效率。现在已经存在足够多的活动在显著创造性的活动的指示下向这个新的环境的挺进，旨在填满商业生态系统以利用和驱动市场扩张。

上述这些都是它的优点，它存在缺点吗？存在一个缺点，那就是难以实现和维持开发环境，还有缺少对于开发和生产应用的管理工具。在 RailsConf 2007 上所发布的公告将明显改变所有这些。

(作者: Richard Ptak 来源: TechTarget 中国)

理解 Web 服务客户端流行语言:Ruby

许多被人描述为“脚本语言”的语言令人感兴趣，具有极大的可能来构建 Web 服务客户端应用程序。术语“脚本语言”被广泛的用于一些语言，这些语言是从现有构件建立的“quick and dirty”解决方案，而不是一个完整的系统编程语言。这些语言通常具有很多新的语句，并且具有细节的自动管理功能，例如内存管理。

基于 TIOBE 语言流行性的调查，现在最为流行的“脚本”语言是 PHP，Perl，Python，JavaScript 和 Ruby。Perl 是早期的 Internet 的重负荷载体，但其流行性正在减退。PHP 作为一个创建网页的工具还在不断的变得更加流行，但是我决定要看一看 Ruby。整个工业界轰动于用 Ruby 开发的基于面向数据库的开源项目“Rails” Web 架构，我对此留有深刻的印象。

Ruby 编程语言大概是在 1995 形成的，它差不多和 Java 是在同一个时代。尽管如此，由于其创建是基于单独的创建者 Yukihiro Matsumoto 的预想，而不是一个联合的开发团队的合作的结果，所以它的发展始终是更加缓慢的，并且也没有取得那么广泛的使用。在过去的几年中，一个关于在开源项目的兴趣的调查发现，整个社区已推动了许多 Ruby 开发。现在 Ruby 的版本是 1.8.4，即将到来的最新版本是 1.9。

Ruby 是一个具有动态类型的解释型面向对象语言。Ruby 的一个典型的运行环境包括一个快速测试的交互接口。现在的标准库包括处理 XML 文档的扩展类和访问用 XML-RPC 和 SOAP 类型的 Web services 的具体类。

一个 XML-RPC 的 Ruby 例子

下面是所有的访问一个 XML-RPC 类型的 Web 服务的 Ruby 代码，加入行号以便于讨论。

```
1. require 'xmlrpc/client'
```

```
2. server = XMLRPC::Client.new2("http://time.xmlrpc.com/RPC2")

3. result = server.call("currentTime.getCurrentTime")

4. printf("xmlrpc time %i:%i:%i ", result.hour, result.min, result.sec)
```

第 1 行使“xmlrpc/client”代码从标准库中装载出来。第 2 行创建了一个使用一个 Url 和默认标准的客户端类的实例。在第 3 行，你也许能够从以前的文章中回忆起来，XML-RPC 协议使用一个简单的数据类型集合，并且不要求命名空间。其中的一个数据类型是 timestamp，它在 Ruby XML-RPC 中成为一个 DateTime 类的实例，而 DateTime 类将会由第 3 行的服务调用返回。第 4 行使用大家熟悉的“printf”语法来显示 DateTime 对象的独立成员的值。

一个 Ruby 的 SOAP 例子

许多版本的 Ruby 构建的 SOAP 工具已经由动态开源 Ruby 社区开发出来了。在我的 XML-RPC 的成功的鼓励下，我决定试一试可以直接依靠 WSDL 工作的标准库工具。然而它却比我预想的要难，因为它的文档和示例并没有和开发相关的，例如，Sun 的 Web Services Developer 的 Java 包。下面是取得一个股票引用的所有程序，加入行号以方便讨论。

```
1. require 'soap/wsdlDriver'

2. wsdl_url = "http://ws.invesbot.com/stockquotes.asmx?WSDL"

3. soap = SOAP::WSDLDriverFactory.new( wsdl_url ).create_rpc_driver

4. soap.wiredump_file_base = "soapresult"

5. param = {"symbol" => "WAG" }

6. result = soap.getQuote( param )
```

```
7. puts result.getQuoteResult.stockQuote.prevClose
```

第 1 行从标准库载入“soap/wsdlDriver”代码。第 2 行代码创建一个包含访问描述股票引用服务的 WSDL 文件的 Url 变量。注意在 Ruby 中，被命名为“wsdl_url”的变量并不包含一个相关的类的类型，它可以引用任何的对象类型。

第 3 行进行了许多幕后的工作，最后创建了一个定制化的类的实例，该定制化类是由 WSDL 描述的 Web service。最后的对象被创建出来，并具有了一系列的方法，这些方法的名字都是从 WSDL 包含的变量得到的，例如在第 4 行取得设置的“wiredump_file_base”。如果变量是一个集合，SOAP 请求和响应的全文都将会在本地文件中记录下来，这在处理复杂的 Web 服务中是一个很大的帮助。

第 5 行创建了一个“hash”——一个和具有股票标识的命名符号相关的数组——在例子中，“WAG”代码表示 Walgreens。这个 hash 是把股票代码映射为 SOAP 请求元素。下面的就是 WSDL 中描述 GetQuote 方法的那部分。

```
<s:element name="GetQuote">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="symbol" type="s:string"
/>
    </s:sequence>
  </s:complexType>
</s:element>
```

第 6 行调用了 getQuote 方法，getQuote 方法是在一个由 Ruby 从 WSDL 中动态创建的定制类中的。这个方法完成了所有的格式化 SOAP 请求，联系 Web service 和解析 SOAP 响应的工作。它所返回的“结果”是一个一般的“映射”对象，包含了 SOAP 响应的数据。断定确定的语法来从这个响应中得到有用的数据并不是那么重要的。熟悉 SOAP 响应所透

露的全部信息将会揭示出这个服务返回了许多的数据，并且这些数据大都在 WSDL 中简单的以以下的形式描述的：

```
<s:element minOccurs="0" maxOccurs="1" name="GetQuoteResult">
  <s:complexType mixed="true">
    <s:sequence>
      <s:any />
    </s:sequence>
  </s:complexType>
</s:element>
```

我们实际需要的内容是在 `GetQuoteResult` 元素中的 `StockQuote` 元素所包含的 `PrevClose` 元素内的。第 7 行输出了这个内容，并把它标记为 `"getQuoteResult.stockQuote.prevClose"`，而不是使用元素的名字的以大写字母开始的形式，如 `"StockQuote"`。我并不确定为什么 Ruby 库要这样设计，但是发现这个敲门还真浪费了不少时间。

正如你在例子中看到的，一个 Ruby 中的 Web service 客户端可以通过使用标准库以很少的几行代码来创建。和一个具有更长的历史和更多支持者的语言来对比，你可能需要做更多的试验，所以具有脚本语言的快速测试能力将会是一个很大的帮助。

(作者: William Brogden 来源: TechTarget 中国)

SOA 结束了大型程序的年代 为 Ruby on Rails 敞开大门

Engine Yard 公司的首席技术官 Tom Mornini 表示，单机百万线应用的时代已经结束，面向服务架构(SOA)是这一时代的终结者。该公司提供 Ruby and Rails 主机服务器。

他在最近的采访中说“我认为使用大型程序的年代已经结束了”“有些程序看起来很大，但是随着时间的推移，它们将最终成为许多小程序的结合体。”

通过为全球市场的业务提供灵活性，SOA 的可组合性改变了应用开发比赛。在全球市场中，商业机会不是一成不变的。

Mornini 说“我实在看不出任何其他方式可以满足存取数据，改变流体的需求，以便在企业内外跟上时代的步伐。”“这就是为什么未来能解决所有问题的单机百万线应用在这一点上仅仅是个遗迹。”

Mornini 认为，这不再是 SOA 是传统应用开发选择的问题，而是除了 SOA 以外，我们没有其它的选择。

他说“这些大型程序很难管理和维护，很难想像单机应用会成为未来发展的方向”。

Engine Yard 公司的首席技术官认为带有 REST 的 Ruby on Rails 是为 SOA 建立新一代的服务和应用的一种方法。与 Java 不同，Java 是在 SOA 应用开发时代前开发的项目，他注意到，Ruby on Rails 和 REST 怀抱 SOA 为理念向世人提供了一个前所未有的方法。

Mornini 说“拥有一个服从该框架的牢固而又深厚的面向服务架构就是 Rails 的秘诀”该架构的开发商认为（它的 SOA 功能）是该平台的一大优势。

他认为 Ruby on Rails 非常适合 SOA 开发。新发布的 Rail 2.0 令该框架更容易为 SOA 应用以及旧数据存取所接受。他承认，原有的 Rails 框架与旧数据存取关系并不是十分融洽。今年推出的新模型已经超过了前者。

他说，例如，Rails 组提供的代码增加了许多新的功能，通过以服务的形式将旧数据曝光，使得在 SOA 应用中访问旧数据变得更为简便。

Engine Yard 公司的首席技术官说 “由于遵循了售后服务书籍和网络视频记录的规程，Rails 令开发商使用 RESTful 数据变得更为简单”。

他说，“如果你遵循 RESTful Rails 的标准过程，在系统外用 Rails 编写了一个程序，就会自动得到该程序展示的一个建立在 XML-over-HTTP 基础之上的 API。

但是如果要使其运转，“继续使用 Rails”很重要。Mornini 说这就是 Rails 遵循既定规程的妙招。

(作者: Rich Seeley 来源: TechTarget 中国)

最新升级版本 Python 语言正在推出

我仔细看了几年前 Web 服务客户的 Python2.4 版本。作为脚本语言，Python 很受效用快速项目的欢迎，许多系统都预先安装了 Python 语言。通过 .py 文件类型的验证的 Python 脚本在很多地方都看得到。大部分平台都可以使用 Python。基本语言完全时开放源，但是许可证准许开发客户商业版本。Sun 公司致力于开发 Jython，以便访问 Java 数据库是在 Java 虚拟机器上运行 Python 程序。许多开发商说，他们的 Python 着手使用工作代码的速度惊人的快。

Python 开发商群创建了 Python 3.0，并在 2008 年下半年公布了这个版本。处理字符串的主要变化以及清理过时方法意味着 Python 2.0 版本失掉了向后兼容性和 Python 2.0 同时期开发的 2.6 版本，也在同时间发布，意在为 Python 开发商打开使用 3.0 版本的通道。通过 Google 公司的 App Engine，可以在 Google 硬件上就可以运行 Web 应用，现在支持 Python2.5，但是现在已经升级到 3.0 版本了。许多包括 Guido van Rossum（Python 的创始人）在内的 Python 开发商团体，都为 GOOLE 工作，升级是早晚的事。

支持联网

从套接字级别到 HTTP 电子邮件协议，Python 标准数据库对联网的扩展支持，插座工具支持 IPv4 和 IPv6 访问。低级别的协议 UDP 和 TCP 也可以得到支持，标准数据库中包含创建个人服务器的支持。

许多基于 Python 的 Web 应用框架纷纷涌现出来，但是 Python 团体看起来已经选定了 WSGI，Web 服务器公共网关接口，作为统一的规范。支持 WSGI 的商业和开放源框架包括 Google App Engine。总体上来看这些框架并没有完全升级到 Python 3 版本。

Xml 支持

对于处于第二阶段 DOM 实施的 XML 来说，Python 3 支持 W3C DOM 模型。这也是和第一阶段（大约是 1998 年）W3C 相对应的轻量级 DOM，该 DOM 不用添加命名空间支持。另一个就是精明的“pulldom”API，在 SAX 解析时期为创建部分 DOM 提供结构，尤其是在处理大型 XML 文档时，这种结构尤为方便。在这里并没有用 Python 编写原始解析功能，但是却使用了 C 语言编写的“expat”解析器。Expat 是非确认性解析器，有限的 XPath 存在，但是并没有很好的记录下来。

尽管有些项目存在，但是 Python 团体并不热衷于为 SOAP、WSDL 和 WS-* 标准创建支持。REST 格式的 Web 服务可能更符合 Python 对于简单性的要求。

Python 和 JSON

JSON 是 JavaScript 对象表示法的缩写，它支持 Python 标准数据库，有 2.6 和 3.0 两个版本。JSON 已经成为众多开发商在因特网应用程序（RIA）传达数据的技术首选。JSON 十分紧密，而且容易访问，只要用类似评估 XMLHttpRequest 结果的编码就可以进行访问。但是不能像 XML 那样传递复杂数据层。

JSON 可以让你序列化 Python 对象的数据内容，例如回应浏览器请求。此外，你可以用 JSON 文档组装 Python 数据。Google App Engine 为创建和使用 JSON 格式化的数据提供支持。

Python 和 XML-RPC

XML-RPC 是执行远程过程调用的简单协议，该远程过程调用使用的是 XML 编码的信息。这种观点积累了额外的复杂性，并卷入了 SOAP。除了 Python，XML-RPC 对很多语言都适用。

3 版本分布的范例代码展示了创建服务器的多种方法，该服务器用于处理 XML-RPC 格式的请求，以便执行 Python 方法，返回合适的格式化 XML。你的程序不必对 XML 的一切都了如指掌，Python 工具包会用变量和方法解释所有请求，并格式化。

为了用 Python XML-RPC 方法创建 HTML 文件，以便回应简单定向服务 URL 的 HTML GET，这里还有一个独特的特性。为了让你更清楚和 SOA 相比 XML-RPC 有多简单，我在 Python 3.0 标准数据库中运行了一个范例标 XML-RPC 服务器和客户程序，捕获了请求和响应。这个代码是由两行编码编写的：

```
POST /RPC2 HTTP/1.1
Host: localhost:8080
Accept-Encoding: identity
Content-Length: 187
Content-Type: text/xml
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
```

```
<? xml version='1.0' ? >
<methodCall>
<methodName>pow</methodName>
<params>
<param>
<value><int>2</int></value>
</param>
<param>
<value><int>5</int></value>
</param>
</params>
</methodCall>
```

从服务器方面来看，很多行创建了服务器实例注册了“pow”方法，服务器应答是：

```
HTTP/1.0 200 OK
Server: BaseHTTP/0.6 Python/3.0
```

Date: Thu, 29 Jan 2009 20:29:17 GMT

Content-type: text/xml

Content-length: 122

```
<? xml version='1.0'? >
<methodResponse>
<params>
<param>
<value><int>32</int></value>
</param>
</params>
</methodResponse>
```

结论

如果你已经有了 Python 支持，如果从原来的版本升级到 3 版本，过程很复杂，但是有 Unicode 的支持，最终还是可以取得成功。如果你需要一种语言开发应用以便支持 Unicode，并且希望这种语言在不同的操作系统上都能够运行。Python 是一个不错的选择。如果你在寻找一个脚本语言以便快速完成 SOA 格式的典型，你需要 WS-*s 支持 SOAP/WSDL，你可能要看看别处了。

(作者: William Brogden 来源: TechTarget 中国)

面向服务架构的商业现状

技术可用性的真实标准是它对于用户的透明度。对于美国的国防承包巨头 Raytheon 公司来说，采用面向服务的方法进行应用程序开发使其轻松的达到了标准。

面向服务的架构（SOA）是 Raytheon 公司 Web 服务的核心，Waltham, Mass.-based Raytheon 使用其 Web 服务来连接一个部件搜索应用程序和大型机中存储的数据。Raytheon 公司企业应用组的高级商业技术专家 Rob Vettor 说，用户并非真正了解，也并不关心这是否是一个面向服务的架构，他们只要知道：不管使用什么操作系统，他们都可以访问每一部件的数据就行了。

SOA 的商业状况取决于它所提供的让企业为达到其它目的而复用其组件的能力。这些成为了可以被网络上的其它应用程序共用的“服务”——这就是所谓的“Web 服务”。

SOA 的前身，面向对象的程序设计（OOP），已经不那么流行了，因为更多的商业交易成为了从电脑到电脑的通讯。

重协议 vs 袖珍协议

Zap Think LLC 的高级分析师 Ronald Schmelzer 说：“面向对象的架构主要是用于解决单机的操作。当人们打算把面向对象程序设计的概念推广到多机系统时，效果就不那么好了。这是因为面向对象概念是建立在“重”协议（如公用对象请求代管器体系结构 Common Object Request Broker Architecture、Internet 对象间协议和分散式组件对象模式 Distributed Component Object Model）基础上的。这些协议对于互联网上操作并不是很有效。而另一方面，SOA 使用的是袖珍的灵活的协议（如 HTTP 协议和 SOAP 简单对象捕获协议 Simple Object Access Protocol）。”

复用性所需的动机

开发商没有动机去为其它 IT 项目开发可复用的服务。Thomas Manes 认为有两种途径来打破这种理念体系。第一个途径是想那些为创建可复用代码做出努力的人提供经济上的奖励。第二个途径是为现有的体系设置一种障碍。例如，她的一个客户，一个已经厌倦了重复代码浪费的金融服务公司 CIO，某一次突然告诉他的开发人员：如果他看见两个人在编写相同的“开户”应用程序，那么这两个人都会被解雇。

她说，虽然有些严厉，但有时这样的措施是必要的。通过使用可复用组件编写唯一的“开户”应用程序，公司可以更加容易的创建相关的财务管理、商业账户、仓库账户和 401k 账户的应用程序。相比于为某种特殊目的开发应用程序，创建可复用的服务所面临的挑战要大得多。

Raytheon 公司从它的部件搜索 Web 服务中领略到了改变传统观念的重要性。这项服务可以将基于 Web 的应用程序和存储关于部件的关键数据的大型机联系起来。

适应新的 IT 模式

除了建立复用性所面临的技术挑战，Rob Vettor 的组还不得不让其他 IT 专家相信这是一个好主意。他说：“这是一个模式的改变，必须要让人们明白，现在是时候让人们搭建一个联系大型机和数据的 Web 应用程序服务了。我认为 IT 界的每个人斗争在调整适应这种新的模式。”

这也是金融服务公司的 IT 项目经理所面临的问题。这个公司也许要马上要开通两项内部保险服务，它不得不在很多职位上雇佣新人手，包括程序开发人员、基础设备管理员和数据库管理员。

令人惊奇的是，为 SOA 而进行的商业购入对这些公司来说是很容易的。对于在六个商业部门有总共 78000 雇员的 Raytheon 公司来说，其投资回报率已经被新的搜索服务所创造的效率所证明。而金融公司 IT 部门的投资回报率体现在其创建了一个新服务，这项服务所需的成本和时间只是原来的企业应用集成（EAI）项目的一部分。

Fortune 100 公司的 IT 执行官说：“从企业的观点，一个单独的服务项目现在可以在根本上优化你的各项应用的成本。”

对开发人员进行适当培训

引入 SOA 所面临的一个重要问题是寻找有相应技能的开发人员来创造实际的面向服务的应用程序。

Knoxville, Tenn.-based Windows consultancy Wintellect 的合伙人 John Robbins 认为，在过去二十年间，IT 开发人员已经具备了很好的编写客户应用程序的技能，但是，当编写 SOA 所要求的服务器应用程序时，他们还面临着很严峻的学习问题。

Robbins 说：“我们现在遇到的很多许多性能问题都是由于开发人员使用客户服务程序的开发理念来解决服务架构问题而造成的。”

他认为，有远见的公司将会对它们的开发人员进行服务架构方面的适当培训，然后再让他们结构主要的基于 SOA 的项目。

(作者: John Hogan 来源: TechTarget 中国)

轻量级 Java 框架和面向服务架构

有些事情碰巧以架构的名义不完全按计划成功进行。企业 Java Beans 是其中之一。尽管许多个人和公司在较大规模的 J2EE 标准内创建这个组件的标准有一定任务，EJB 把一些循环如 IBM 对 Java 的重大贡献带进来。有时，EJB 被描述为臃肿的，有时更糟糕。

EJB 在思想方面要早于 Java，因为它紧随在 IBM 用旧金山组件为代码命名的努力之后。EJB，特别是因为它的演变，被广泛认为过于复杂。这可能是因为它试图完全涵盖所有。

更窄的焦点框架已经出现，用来作为完全成熟的具有 EJB 的 J2EE 的替代。其中最主要的是 Spring 框架，它使用面向外观的编程技术，以确定总体的设计模式，而集中在当天最常见的工作，网络应用程序成为重点。

来自 SearchSOA.com 姊妹网站 TSS.com 下 2008 年的 Java 发展趋势调查的初步数据显示，76.8% 的调查参与者使用了 Spring。作为商业代理管家的 Spring 是 SpringSource，致力于促进 Spring-style 应用建设。我们同公司的创始人总裁 Rod Johnson 进行了交谈。他告诉我们，Spring 和 SOA 有很强的互补性。

便携性是驱动 Java 的关键，如果服务捆绑到服务器，就还没有最终决定。

Johnson 认为“SOA 是区别于整合的不同事情，这也正是我们认为 Spring 是有益的。”

他继续说道“在使用 EJB 的 J2EE 的传统方法往往在非常特定的环境下约束您的代码：单片机应用服务器环境。虽然 Spring 可以令人满意[在单片服务器环境]，Spring 容器本身也是便携式”。

能够使用 Spring 可以更好地保证，‘对分离关注’。这是 Spring 的目标，并以它的模型视图控制器（MVC）来概括。

Johnson 表示使用 Spring “您可能有更大的自由，重用您的业务逻辑，并在非传统的部署方案上进行部署。”

Johnson 解释道“EJB 先于 SOA。这是组件模型和分布式计算技术为一体的技术。它还假设你会使用 RMI 或 CORBA 的远程风格。EJB 模型仍然是耦合到应用服务器。问题是在大多数情况下，结合组件模型和分布并没有工作。并提出了对最终服务的很多假设，这是一种均匀分布的粒度的世界，你在导线两侧都有 Java。它并不能反映今天的现实。”

他表示 Spring 框架的哲学是你应该有一个独立于任何远程战略或分布模型的高品质的组件模型，这打破了通过应用服务器耦合的 EJB，为企业架构师提供更多的选择。

现在，Spring 社区正在制订一项建立在‘概念合同第一’的模式上的 Web 服务项目。Spring 3.0，将透过大量的 Spring MVC 的支持 REST。Spring 组件模型将自动支持 REST 风格的通信。

虽然许多成功的应用已使用 EJB 建成，它从来没有完全被欢迎。在表面看来，对于轻量级框架的东西，有一个‘让我们做吧’的态度。这并不意味着，这种办法不能在大公司的 SOA 项目存在合作存在，但并不意味着他们支持 SOA。

(作者: Jack Vaughan 来源: TechTarget 中国)

面向服务架构下的企业具体案例实施

面向服务架构 (SOA) 的核心是 Web 服务，并且国防承包巨头 Raytheon 用其来连接用于跟踪的在其主机上存储了数据的应用程序。Raytheon 的企业应用组的一名高级商务师 Rob Vettor 说到，用户并不完全明白或者关系这是面向服务的架构。所有他们知道的是他们能够获取数据而不论他们使用的是什么系统。

SOA 上的商业案例是基于其允许企业因为一些目的重用应用程序的组件的潜力。这些发展成为“服务”所以他们能够被网络上的其他应用程序所共享——因此有了“Web 服务”这个词。然而当 SOA 追溯到 1980 年代，直到在互联网泡沫时期基于网络的应用程序突然出现，编程的概念才得到了广泛的关注。

与此同时，SOA 的前身，面向对象的编程 (OOP)，开始失宠，因为越来越多的商业交易成为计算机与计算机之间的通信。

重型与轻型协议

“谈到面向对象的架构，它实际上意味着在一个单独的机器上运行。” ZapThink 的一个高级分析师 Ronald Schmelzer 说道，“当人们开始将面向对象的编程思想扩展到多台机器的时候，情况就开始变得糟糕了。”

这是因为面向对象是建立在重型的协议上，例如通用对象请求代理架构 (CORBA)，在互联网上跨对象请求代理协议 (IIOP) 的和分布式组件对象模型 (DCOM 的)。“这些东西不太好横跨互联网进行操作，” schmelzer 说到。从另外一方面来说，使用轻巧并且灵活的协议，例如超文本传输协议 (HTTP) 和简单对象访问协议 (SOAP)。

Midvale 的研究中心的主任 Anne Thomas Manes 说道，她已经提倡 SOA 有 25 年了。
“面向服务的方法是你想建立一个系统并且希望它们具有灵活性和适应性以及重用性时的一个办法。”

重用性需要鼓励

问题是开发商没有一个诱因来促使他们为其他的 IT 项目建立可以重用的服务。
Thomas Manes 说到有两种办法可以打破这种心态。第一个是对那些额外编写了可以重用的代码提供一定的财政上的奖励。第二个是阻挠。例如，她的一个客户是一家财政服务公司的 CIO，越来越厌倦看到资源重叠浪费。有一次，他告诉开发者们说如果他再看到两个人都编写同样的“开户申请”的应用程序，他们俩都将被开除掉。

尽管听起来有些严厉，但是在某些时候是必要的，她说到。通过使用可以重用的组件编写“开户申请”的应用程序，那么公司可以为投资组合管理, 商业账户, 仓库账户和医疗保险账户更方便快捷地编写相关的应用程序。Thomas Manes 说到：“创建一个可以重用的服务比为一个特定目标更有趣更有挑战性。”

必须改变心态，其重要性是 Raytheon 从它的部分跟踪 Web 服务中获得的一个教训。
这个 Web 服务是一个内部服务，它在基于 Web 的应用程序与存储关键数据的主机之间架起了一座桥梁。应用来自西雅图 WRQ 公司的 Verastream 集成软件的服务第一次是与设在北德州的务单元一起测试的，并且很快在佛罗里达州和加州也推出了。

适应一个新的 IT 范例

除了技术上创造可重用性的挑战外，Rob Vettor 的工作组不得不说服其它的 IT 领域的专业人士，这是一个好的想法。“它是一个模型转变，使得人们谈论说‘嘿，那好，让人们写一个逆主机的服务并且把数据移居到一个 Web 应用程序上去。’” Vettor 说到。
“我认为 IT 领域的每个人都在试图适应那个范例。”

一个负责财富 100 强的金融服务公司的 IT 项目经理也有过类似的例子，不过他要求不要写出他和他所在公司的名字。那家公司将把两项内部的基于保险的服务在四月份推出，不得不在许多“接触点”保持收购，包括开发者，基础设施管理员以及数据库管理员。

出乎意料的是，对两家公司来说，倾向商务的 SOA 的收购容易得多。对于在其六个业务单元拥有 78,000 个员工的 Raytheon 公司来说，通过追踪服务提高效率已经证明了投资回报。财政服务公司的 IT 部门的投资回报是为其为一个传统的企业应用程序集成项目的开支和时间的一部分创建一个服务的能力。

“一个单一的服务项目现在基本上能够证明你能够利用的东西的成本，从一个企业的角度来说。”这家财富 100 强的公司的 IT 技术主管说道。

给予开发着适当的训练

采取 SOA 方式的另外一个重大的挑战是通过适当的技能集来寻找开发着，从而开发真正面向服务的应用程序。

Knoxville 的联合创始人 John Robbins 说到 IT 开发者在过去的 10 年里在编写客户端应用程序方面已经很优秀了，但是当编写 SOA 需要的高容量的服务器端应用程序的时候，他们现在面对的是一个陡峭的学习曲线。

“许多我们称之为臭虫的性能问题直接关系到使用客户端应用程序的开发者对于服务架构问题的心态” Robbin 说到。

他说到明智的公司采取主要基于 SOA 的项目之前他们会确保他们的开发者已经接受了适当的训练和在服务架构方面的经验。

(作者: John Hogan 来源: TechTarget 中国)

SOA 敏捷应对 Web2.0 时代变化

没有什么比把两个热门话题放在一起更有趣的了。比如，在过去几年里不断被提起的 Web2.0 与面向服务架构(SOA)……

通常他们都与类似“丰富互联网应用”(RIA)，软件即服务(SaaS)等词一起出现。处于市场部门的人员能使人们倾向于购买一些与多个流行词汇有联系的产品，这并不稀奇，这样一来，你就知道“Web2.0 软件能协助你在 SOA 架构中使用 RIA 和 SaaS。”

第一，我们应该强调的是：SOA 必须是真实的。你才能做到这点。SOA 有着一套核心原则，技术团体花了很长时间来定义 SOA。OASIS 整合了一个 SOA 的参考模型。

SaaS 和 RIA 也是真实的，但什么是 Web2.0 呢？W3C 正致力于建立语义上的 Web。而当前 Web 得到实现时，Web2.0 就将成为现实。近来，标有 Web2.0 标志的产品已经有所不同了。这只是单纯的一个流行词汇，一个为企业全面利用开放式多媒体网络(即互联网)的力量提供的占位术语。这个动作是在静止网页之外的，也标志着企业使用互联网的时代的来临。

本质上来说，Web2.0 是对于更加全面利用互联网的一种呼吁。从互联网获取更多的资源是一个不错的主意，而 Web2.0 也许将成为此观点比较简单的代名词。

表面看来，似乎 SOA 是企业的服务解决方案，而 Web2.0 属于纯 Web 商务。但是，这两种模型都可以帮助对方成为未来的服务模型，因为它们在很多方面相互补充。真正值得感兴趣的是 SOA 的发展前景，这其中包含了部分 Web2.0 中的特性。

SOA 主要关注于企业内的最优化服务，而 Web2.0 是包含了内部与外部企业的共享服务网络。当 SOA 的内部服务可以利用外部服务时，将会有有一个真实的共享、互操作、创

建集合组件的机会。今天，很少有企业了解如何连接到企业防火墙外的远端服务，同时保护数据的安全性和完整性。

SOA 在配置控制与 workflow 管理中非常健壮；相反，今天的 Web2.0 以缺乏控制结构为荣。开发者在 Web2.0 环境中可以很容易的获得来自全球开发者资源的产品就绪代码，他们通过社区博客、wikis、podcasts 等等培养了内部创造性的增长。

增加社会元素以利用 Web2.0、并以 SOA 为中心的结构为商业提供了新的机会，它将最佳的社会交互方式与合成应用结合在了一起。

未来 SOA 与 Web2.0 结合起来的想法并不是不可想象的。Dion Hinchcliffe, AJAX Magazine 主编、SOA-Web 2.0 space 的领袖，认为这种联合已经不远了。在 SYS-con.com 网他的 Web2.0 博客上(开始的六个月他的博客就被阅读了一百万次)他一直致力于 Web2.0 与 SOA 联合的话题。IT 业界的分析家也表达了他们的看法。

IT 商家们也正努力地使得 Web2.0 理念能给他们的业务带来正面的影响。互联网对于商业领域的影响才刚刚开始，改革仅触及皮毛而已，SOA 在很多方面能使你敏捷应对和吸收即将到来的猛烈的变化。

(作者: William Murray 来源: TechTarget 中国)