



企业 Mashup 指南

企业 Mashup 指南

Mashup 是糅合，是当今网络上新出现的一种网络现象，将两种以上使用公共或者私有数据库的 web 应用，加在一起，形成一个整合应用。一般使用源应用的 api 接口，或者是一些 rss 输出（含 atom）作为内容源，合并的 web 应用用什么技术，则没有什么限制。mashup 在 geek 群体和互联网玩家之中获得了极大的欢迎，mashup 未必需要很高的编程技能，只需要熟悉 api 和网络服务工作方式，都能进行开发，所以很快成为一个流行的网络现象。很多公司例如 yahoo/google 都为此提供开放接口，以吸引这个群体。

Mashup 是一个非常 cool 的新的应用程序种类。如果你想真正的了解它们，我们需要回过头来看看你现在的计算机，其实它就是一个非常好的帮助你理解 Mashup 的模型。计算机运行着操作系统，例如 Windows。现在开源的操作系统无疑是一格非常好的 APIs 的集合或者一个应用程序编程接口，帮助开发者去构建他们的应用程序。计算机本身也是一个很好的为用户提供接口的例子，键盘和鼠标可以被理解为你通过计算机的接口而使用的不同的应用程序。本技术手册为读者提供了一些很实用的小技巧，如果需要深入了解 Mashup，读者可以借助其他参考资源。

Mashup 专家解惑

Mashup 在国外应用已经比较广泛，比如 Google 的地图通过开放 API，可以让用户或其他网站制作当地的餐馆分布图等等。在博客、网摘、图片、视频等服务应用越来越广泛的时候，Mashup 把这些分散的内容创造性地组合起来提供了全新的方式。

- ❖ 评价“企业 mashups”的标准是什么？
- ❖ 用企业 mashup 模式作为 API 引擎

企业 Mashup 新策略

Mashup 把 Web 服务和像 Ajax 这样的工具进行融合，提供一种新的应用软件开发模式，开发人员就能够更快地把应用整合在一起。这种简化开发难度的模式，在一定程度上会减少企业和客户的应用难度，加大双方交互性。

- ❖ 企业 Mashup 将成为 SOA 杀手？
- ❖ 用流程 mashups 平衡可重用性和 situationality（一）
- ❖ 用流程 mashups 平衡可重用性和 situationality（二）
- ❖ 企业 mashups 将 IT 和 LOB 协作引进 BPM

企业 Mashup 产品和工具

Mashup 不仅仅是发展的很快，更好的一个地方是你不再需通过某人才可以发布一个新 API 到网上。以前你必须要通过微软才可以发布一个新的 API 加入到 Windows，这种情况同样出现在 Macintosh。另，尽管你也许可以添加一个 API 直接到 Linux，因为它是开源的，但是这对大多开发者来说并不具备这样的权利。

另外，你在发布了一个 API 到网上之后，不需要再通过谁去批准它，它会自动对所有开发者有效。所有的不同的 APIs 都会成为更多 Mashup 的燃料。

越来越多的 Mashup 出现在网上，越来越多的网络用户去使用这些新的 Mashup 应用程序。越来越多的开发者加入到开发 Mashup 这个生态系统当中。在未来两年，你可以听到关于 Mashup 的信息。

- ❖ RIA 和 mashup 工具
- ❖ SOA 数据服务 mashups 出现

-
- ❖ 把握 JavaFX1.2 在富互联网应用方面的关键
 - ❖ 企业混搭：工具构建数据集成

评价“企业 mashups”的标准是什么？



Jason Bloomberg 是面向服务架构行业分析股东和 ZapThink LLC 咨询公司顾问。Bloomberg 在商业技术管理和行业分析有不同的背景。包括作为 IDC 的商业咨询团队的资深分析师，以及以及控股的电子商务管理职位 USWeb/CKS（后来的 marchFIRST）和 wavebend 解决方案（现在的日立咨询）。他是 SearchSOA.com. 新近 RIA 和 mashup 企业专家。

问：评价“企业 mashups”的标准是什么？尤其是在企业 mashups 和“主机包装”项目的关系上？

答：我们对企业 mashups 的定义是：丰富网络环境下，一套建立在 SOA 基础之上的组合，该组合拥有良好的管理和治理。企业 mashups 的这两个特征将其自身和其他服务组合区分开来，是一种可视化组件，可以帮助终端用户创建应用——以便完成自身映射。

如果用 SOA 方法部署这些服务，就可以全面使用基于主机的功能，也可以用 SOA 映射这种功能。问题是你想不想支持这种功能。但是，归根结底还是因为你是否希望某些用户拥有该项组合功能。我们发现，你并经常希望用户得到这样的授权。但是在丰富的可视化环境中，业务需要终端客户使用这样的服务组合，因此企业 mashups 始终是最佳选择。

(作者: Jason Bloomberg 来源: TechTarget 中国)

用企业 mashup 模式作为 API 引擎

在最近出版的“mashup 模式：现代企业设计和实例”（Addison-Wesley, 2009）一书中，作者 Michael Ogrinz 表示，PC 终端年代的“屏幕抓取”给所有从用户界面抽取原始数据的方法一个诠释，然而这仅仅是企业 mashups 所面临障碍中的其中一个。

Ogrinz 说，Mashup 发起人应该还记得，无论单个的 HTML 要素如何分散，基本文档对象模式（DOM）都可以强化数据结构和‘屏幕抓取’相比，“收获”这个词似乎更为实用。

Ogrinz 在该书中探讨了 Alerter、超前经济指标。API 引擎、Infinite Monkeys 以及时间序列模式。以上所提到的这些可以看做是一套收获模式。如果用这种模式来看待 mashup 项目，就可以在前端和后端剪掉 mashup 开发时间。这一点对所谓的 REST 设计模式的帮助极大。

Ogrinz 注意到，大多数 mashup 设计都要求将 API “链接”到一起。他向我们展示，在使用网络接口的远程旅游交易中，是如何利用变量链来建立当前环境的。

例如，当你只寻找两个数据点时——该数据点用于比较当前的数据读取同时也可以观察偏差数，知道自己有一个 Alerter 模式。Ogrinz 注意到在实施 Alerter 的过程中要格外小心，以便确保其不会影响到受监测系统的稳定性。

Alerter 的著作也介绍了其他种类的企业 mashup 开发设计模式。此外，SearchSOA.com 仅仅节选了“mashup 模式：现代企业设计和实例”一书中的部分章节，这些章节中涉及的问题为设计企业 mashup 创造了条件。

(作者: Jack Vaughan 来源: TechTarget 中国)

企业 Mashup 将成为 SOA 杀手?

为什么企业 Mashup 比面向服务架构 (SOA) 更具吸引力?

因为一个引起了业务人士的共鸣, 另一个没有。John Crupi 这样回答。Crupi 曾任职 Sun Microsystems 的 CTO, 负责 SOA。他现在是 JackBe 公司 CTO, 负责企业 Mashup。

“我从事了多年 SOA 方面的工作, 我可能是第一个告诉你业务方面不在乎 SOA 的人。”他说, “对他们而言, SOA 没有什么实质性的内容。他们听到的只是一大堆承诺。”

Mashup 远比 SOA 容易销售, 因为厂商能够很快地演示其商业价值, 例如分散的企业数据能够集合为一个 Mashup 提供具有商业价值的信息。” Crupi 指出。

ZapThink 的高级分析师 Jason Bloomberg 说, 他对 Crupi 分析为什么 Mashup 是一个如此具有吸引力的概念表示赞同, 不过, Mashup 不会取代 SOA, 而是在 SOA 之上。

“我们已经预言了很久, 随着 SOA 的最佳实践方案被广泛接纳为 IT 和业务的最佳实践方案, SOA 将从人们视野中逐渐消失。这也是我们目前正看到的发展趋势,” Bloomberg 说, “进一步来说, 企业 Mashup 正日渐成为 SOA 应用案例的杀手。从业务的角度来讲, 应用案例是采用 SOA 的表面理由。所以, SOA 比以前都要强大, 成为产品的一部分。企业 Mashup 是展示产品的那一部分。”

Crupi 说, 拿数据服务提供商举例, 相对而言, Mashup 工具厂商容易演示如何将他们已有的信息重新打包到一个 Web 浏览器, 提供新的收入流。

“所有的数据服务提供商用几种方式打包数据提供给用户，” Crupi 说，“无论是一个文件夹，还是每几周上传的 FTP，或者你做一份定制报告，他们如何将数据提供给客户的方式受到限制。”

使用 Mashup，房地产业务的数据服务提供商就可以创建一个应用，让房地产代理商只访问他们出售房子所在地区的数据。这类似 Apple iTunes 卖单曲而不是专辑的商业模式，房地产代理只需要为所显示的房子所在地区的数据付费。

“它是一个小包装，” Crupi 说，“就像我喝酒是按杯付费的。”

房地产数据服务就是按杯销售的。酒吧赚钱并不是通过向客人买一整瓶杜松子酒，而是一杯马丁尼就有钱可赚。所以，采用 Mashup，一种新的商业模式应运而生。

“业务部门能够获得以前无法获得的数据，利用该数据创造收入，” Crupi 说，“我们对企业 Mashup 所做的事情就是演示某种业务并进行提供。这能够对数据加以利用，进行访问，加以表现，甚至提供新的收入渠道，因为他们可以访问以前没有的数据。这是为业务部门而设计的。这是他们可以销售的东西。”

拿消费类电子产品打个比方，HDTV 显示器内的电路可能提供了高清晰度图片，但是，对顾客而言，他们感兴趣的只是球赛的清晰度。

“SOA 是一个标准的获取数据的方式，可以实现某种价值，” Crupi 说，“但我认为企业 Mashup 会使 SOA 黯然失色。”

多数 HDTV 购买者不会走进商店，询问电路。他们只在乎图像。

“有件事情很有意思，” Crupi 说，“我们会接到客户的电话询问：‘给我们讲讲 Mashup。’这是来自业务部门。但是我参加过的客户会议中从来没有业务人员说：‘给我们讲讲 SOA。’”

但是，他并不认为 SOA 没有前途或已经过时，而是 SOA 可能最终成为幕后技术，而不是主流技术。

“我认为 SOA 将在今年长足发展，” Crupi 预测道，“只是不会被讨论得太多。”

(作者: Rich Seeley 来源: TechTarget 中国)

用流程 mashups 平衡可重用性和 situationality (一)

面向服务架构（SOA）的核心是面向网络架构（WOA）以及以业务为中心的 Web 2.0，我们称其为 Enterprise 2.0，这是对企业 mashups 的称呼。它被定义为以 Web 为基础的用户接口环境中被治理以及被管理服务部分。尽管这样的应用在企业中范围相对有限，但企业 mashups 仍然可以成为 SOA 的主要驱动力量。

实际上，是 mashups 限制了这些程序的应用范围：使用一个 mashups 需要建立一个 mashups，换句话说，就是让业务用户管理 mashups 的功能即准许用户在使用 mashups 的过程中创建应用。这个想法听起来不错，但是 IT 真正希望程序使用者建立并修改这些程序的时候有多少呢？

像这种“使用即创建”的应用，我们称其为情景应用。情景应用是为了满足一部分人群短期需求而专门设计的应用。短期内在某个特定的情景下可能会很有用，但是不会产生长期的效益。许多情景应用都是数据 mashups，通过把不同的数据源信息混合在一起，用户在 mashups 中创建应用以便满足特定的业务需要。

情景应用和数据 mashups 的结合会影响企业 mashups 的全部力量，在 SOA 环境内编排服务的关键就是要支持灵活的业务流程。数据 mashups 主要是一个将数据混合在一起的流程：要保证其是一个有用的流程，不过它只是企业想要自动化的众多业务流程中一个有限的实例。我们可能把这样的流程看做是一个“流程 mashups”，而不是一个数据 mashups。那么什么是流程 mashups，你为什么需要流程 mashups，在企业环境下你如何实施流程 mashups？

一个业务 mashups 实例

ZapThink 在 Licensed ZapThink Architect 课程上一直在谈论流程 mashups。我们在课上使用的是呼叫中心应用这个例子。由于“swivel-chair 集成”问题，呼叫中心也是面向服务业务流程（SOBAs）诞生的地方，它是用于实施业务流程的一整套服务组合。传统呼叫中心的代表需要访问多层系统才能完成工作，这些系统主要包括主机绿光屏，用于客户关系应用的客户/服务器接口，用于公共门户或者网站的网络接口，以及为手机系统量身订做的接口。为了满足呼叫客户的需求，CSR 必须在两个屏幕之间不停旋转，这种方法很费时间，而且容易出错。

当然，使用 SOA 效果更佳，通过将应用抽取为服务，呼叫中心可以为 CSR 建造一个流线型的灵活接口，这样 CSR 就可以解决用 swivel-chair 集成问题了。实施客户服务 SOBA 支持这样的接口非常简单，现在许多呼叫中心都实施了 SOA。

实施 SOBA 只是其中的一部分，因为 CSR 只是这个阶段的一个部件。如果我们看一下其它组件，就会发现一个更为复杂的应用布局图，正如如下的图表所示。

运转中的流程 mashups

上面的图表向我们展示了单独的几个人是如何以不同的方式和客户服务 SOBA 发生交互作用的，这主要取决于它们的作用：

- 执行程序（1）会对呼叫中心的关键效绩指标感兴趣，即呼叫时间（越短越好）客户满意度（越高越好）。这样关键效绩指标就可以通过管理仪表板和 SOBA 发生交互作用，通过管理仪表板，人们可以看到这些关键效绩指标。

- 呼叫中心管理员（2）负责处理呼叫中心执行的业务流程。换句话说，呼叫中心管理员负责 CSRs 所执行的脚本，并且要注意 SOBA 是如何支持脚本描述流程的。管理员同时也要负责 KPI 仪表板，以及 SOBA 实施的流程流只读窗口。基于老板的意愿，管理员还要适时地调整呼叫中心流程，并将这些变动上报给业务分析师。

- 在这个实例中，业务分析师可能是唯一有能力改变 SOBA 性能的人。换句话说，业务分析师是呼叫中心拥有 mashups 接口的人。按照管理员的指令，它会直接在 SOBA mashups 的接口中改变 SOBA 的性能。

- CSR（4）有可能是这个程序最高级的用户，因为在运行不同呼叫中心脚本的同时，它要求 SOBA 的全部功能都要满足客户的需要。应该不会改变应用的功能。

- 但是，用户会和 SOBA 发生交互作用，这是因为它支持共有网站（5）或者是因为他们直接通过手机呼入了呼叫中心（6）。用户在网站上的阅读和写入权利是有限的，但是这仍然不能改变应用性能。

(作者: Jason Bloomberg 来源: TechTarget 中国)

用流程 mashups 平衡可重用性和 situationality (二)

运行中的业务授权

这个实例告诉我们企业 mashups 的意义是什么？以及流程 mashups 的特性。首先治理在这里发挥了重大的作用。实际上业务授权需要治理。因为 IT 不会为流程提供强大的工具，除非你可以灵活的管理这些工具的使用情况。

其次，业务分析师的企业接口是针对某个特殊用途而设计的，作用有限，只有一小部分人拥有 mashups 接口，他们可以通过这些接口做出改变。在这个实例中，业务分析师被纳入了呼叫中心小组，而在实际过程中，这个认为可能是 IT 的，尤其是 mashups 工具技术性极强时。理论上讲，mashups 是为业务为中心的知识工作者而服务的，但是其可用性取决于这些工具。

第三，需要注意的是，这个程序是一个流程 mashups 而不是数据 mashups，因为该程序的目标是支持呼叫中心流程。从技术角度来说，业务分析师修改程序所采取的步骤本身就是一个流程，这个流程不是程序的关键，流程和数据 mashups 的关系才是关键。换句话说，这个 SOBA 的 situationality 要比典型的数据 mashups 小，每次用户和工具发生交互作用时，这个 mashups 流程都会发生变化。

因此，mashups 最重要的不是 situationality，而是可重用性。毕竟我们没必要多次运行一个流程，所以我们不用最先将流程自动化，否则自动化就不划算了。

situationality 和可重用性是一个领域的两个极端，从我们的观点来看，这个流程则处于中间位置，这个流程具有一定的可视性需要测量实施这些流程的程序 situationality。就是这些流程计算这些流程 mashups（总的来说就是 SOA）。

ZapThink 采取的措施

同流程 mashups 相比，支持数据 mashups 所需的工具和基础设施是不尽相同的。因此，二者明显在技术角度划清了界限。也许，业务人员对二者的区别不是很清楚。毕竟，从业务角度来说，流程通常会涉及信息。如果不能使用 mashups，很难从信息中获取价值，当业务运转时，这个活动就可以构成一个业务流程。对于业务用户来说，这个授权的意义远比数据和流程之间的差别要大得多。

不管是数据 mashups，还是流程 mashups，企业 mashups 在全局中的意义重大，正处于这场完美风暴的风眼，而现在的这场风暴就是业务授权，业务灵活性以及业务流程。对于机构来说，有这么多技术细节，实在很难保证不亏欠。但是，为了了解 IT 是如何在 SOA 启用的企业 2.0 环境下提交业务值，我们必须要对企业 mashups 的多样性和功效有所了解。

(作者: Jason Bloomberg 来源: TechTarget 中国)

企业 mashups 将 IT 和 LOB 协作引进 BPM

越来越多的应用程序通过使用与精简业务流程管理（BPM）的目标相结合的企业 Mashup 工具组合在一起。Serena, IBM, Cordys 和其他人已经开始提供企业 Mashup 工具, 进一步推动这一趋势。有时, 这些工具和程序曾经模糊了的应用程序开发和业务线（LOB）之间定义完善的分界线。

企业 mashups, 有时被称为 Web 2.0 或 Web 应用程序的混合物, 是组合两个或多个数据源创建一个新的服务的应用程序。这项新服务可以节省成本和时间, 而且似乎越来越受欢迎。

Forrester 研究中心业务流程管理的高级分析师 Clay Richardson 表示 “Mashup 是一种能快速的构建流程的方式, 人们希望在数周内推出从而取代几个月的事情。”

企业需要一种快速、低成本的解决方案, 所以他们转向 mashup。尽管 BPM 在强劲增长, 一些公司被集成项目的成本所挑战。Richardson 表示, 超过一半的 BPM 项目已收到预算削减, 尽管对 BPM 的需求正不断加快。

Serena 软件产品营销总监 Tim Zonca 表示其公司最近由 US Forest Service 公司进行了 mashup 软件的实施, 他认为对 mashup 的需求已经不仅仅是经济。这是正确的时间。人是为了省钱, 但是他们还是有这些问题需要解决。这也是企业试图这样做 IT 以及多喝另一边工作。

据 US Forest Service, Serena 的业务混合做到这一点。一些重建的服务通过 Serena mashup 软件帮助桌面流程, 该公司估计, 在结算重复服务桌面票务时, 它节约了近 40 小时。其结果是业务和 IT 部门之间更好的连接。

Cordys 的首席战略官 Jon Pyke 也认为，mashup 的需求有所增加。Cordys 最近宣布，它通过添加 Cordys Process Factory 到 Google Apps，从而计入了谷歌企业合作伙伴计划（Google Enterprise Partner program）。这便是平台即服务，旨在帮助企业迅速建立起 mashup。

Pyke 说“我认为这是不可避免的，真的。工具演变，出现了 mashup 转变”。

WD Advisors 研究士、分析师 Neil Ward-Dutton 也表示认为 IT 和业务融合比以往任何时候都要好。越来越难告诉一个公司的起点和其他的终点。您的 IT 和业务之间有了更紧密的合作。随着业务方接管越来越多的 IT 职责，有人建议，在工作责任中的重大变革正在进行中。

对于他的部分，Ward-Dutton 表示，他认为这是因为人们认为 BPM 是专门 IT 部门的工作的转移，但它确实依赖如何集中组织。Richardson 表示同意。他建议新的角色会应运而生，进而结合业务和技术能力。

IBM WebSphere 业务流程管理和连接主任 Angel Diaz 认为仍有绝对的应用程序开发者的作用，而且在 IT 业务方面，将允许开发者贡献自己的时间来面对更多的具有挑战性的项目。

Diaz 解释道“在过去，如果业务‘家伙’想改变什么的颜色，他们不得不去找 IT。现在，他们可以自己做那样的事情。”

Diaz 表示 IBM 通过其用于快速和简单的企业 Mashup 建筑设计的程序设计的 WebSphere sMash 使商业方面有更多的控制。在上月公布中讲到 sMash 是 BPM 的 BlueWorks 心脏。这是一个基于云的战略，而且业务流程设计工具还允许不涉及 IT 的快速部署。

Diaz 还表示，发展商没有必要担心自己的工作消失，它依然存在。

Zonca 讲到这种变化将会使开发者腾出来做更多的关键工作。他们将能够将其时间和精力专注于应用程序需求的编码。

IBM 公司，Cordys 和 Serena 已经在内部开始实行混合工具，以简化自己的业务流程。

随着企业 mashups 赢得了知名度，必然将 IT 和业务交织成网，新的角色将会出现。Diaz 认为这种关系已成为共生。他说：“你不能缺其一。力量就蕴藏在在 IT 和业务的合作之中。”

(作者: Lauren Kelly 来源: TechTarget 中国)

RIA 和 mashup 工具

问：当建立一个丰富的 Internet 或 mashup 模式的应用程序时应该坚持什么样的建筑原则？

答：在建筑师的工具箱，建筑的原则是工具，以及使用哪些工具时，大概权利工具的工作是真正重要的-这取决于遇到的问题设法解决。例如，松耦合是一项重要的建筑原则，但不是普遍的提倡。相反，只有当企业需要灵活性，松耦合可以提供建立必要的基础设施支持它。因此，问题的答案是：“这取决于”。

(作者: Jason Bloomberg 来源: TechTarget 中国)

SOA 数据服务 mashups 出现



Kirstan Vandersluis 是 Xaware 公司的创建者和首席科学家，是开放源数据集成供应商，他为建立数据服务层提供工具，其工作主要是关于为面向服务架构(SOA)应用，创建数据服务混合，本月在 JavaOne 会议上，在同 SearchSOA.com 网站的访谈中，他解释到，数据服务 mashups 和 XML 一样也是一种标准。

你能解释一下什么是数据服务 mashups 吗？

把许多资源中的数据放到一个逻辑单位中，就是一个数据服务 mashups。所以数据的方式同样也是一种 mashups。这是完全不同的球类游戏，我们将虚拟世界的任何地方的数据放入逻辑单位。并且正试图获取一束末端数据系统，这些系统非常复杂，按照 XML 模式，该系统更加合理化，有些人将一些理念注入到设计中。

当你获得了所有的数据资源，你最终会将这些数据放入何种业务功能中呢？

最后，你将拥有一个和模式相匹配的 XML 数据集，你可以调用服务从根本上取回数据。

你能举一个类似旅游公司的例子么？

对于一个旅游公司来说，你可能对客户针对旅游史提出的不同观点很感兴趣。所以，网站需要信息。这就是我们没有涉及的 GUI 块。前端应用会提出要求：“获取用户意图。

这里有用户 ID”于是数据 mashups 就成了来自许多资源的数据组合。在这里有一个 CRM 系统，因此我们要在 CRM 系统中查看基本的客户文件。我们要进入旅游管理系统获取旅游历史。如果数据返回，可能会有一些粗格式，这也许是来自 CRM 或者 API 的直接文件，旅游管理系统可能是一个主框架，这些数据会依次返回。然后这些粗数据被自动的转化为 XML 数据集。这些数据集被连到了更大的用户意图结构上。并将其返回应用程序。

在数据服务 mashups 有那些比较么，我们可以对这些数据库和数据市场做些什么？

这是我们做事的另一种方式，一种物物交换。我们肯定不是在说我们正在取代数据库。你可以在一个小的应用中使用它。这个方法带来的一个好处就是所有一切都是实时的。实时的将一个子要求传递到任意一个系统。可以一定程度上将数据库技术融入到你的数据服务 mashups 中。你可以直接查询操作系统。这在一定程度上是由应用程序的要求所驱使的。如果你需要实时数据，我们就有。成为一个抽取层也很好，因为你可以依照模式建立自己的应用。这样你就可以随着时间的推移做出变化，而不会影响到程序。应用程序依赖 XML 模式规定的合同。所以如果你一开始你将所有的数据服务都指向你的数据库，过后你就会发现那并不是实时的，你可以改变正在混合起来的数据集，因此其中的一个可能干扰了一个操作数据源。

你会如何描述那些使用工具的人的工作？它是一个数据架构么？谁真正在做这项工作？

通常，它要么是一个数据设计师，要么是一个 Java 开发商，或者是一个熟悉数据的人。当然数据源是有比例关系的，任何数据设计师或者数据库管理员在这种情况下都是很舒服的，，也许那是一个 Java 开发商，一个数据设计师经常是典型的情况。

你是在典型的数据设计师的工具箱中找到这些技能组的么？

是的，我在那里找到了。一般来说，我在人们那里得到了良好的反馈。我们从数据设计师那里得到很好的反馈，因为它是视觉化的。当我们可以使用元数据时，我们就将其

曝光。这只不过是个拉住鼠标的拖放问题。所以我认为，这完全在他们的技能可以解决的范围内，这比我们以前做的要好的多。

你认为业务方的人们理解数据服务么？当你和他们交谈时，他们意识到数据服务的价值了么？

我认为二者是不可调和的。有些人做到了，有些人做不到，我必须要说大多数人没做到，我认为这更主要的是他们理解面向服务架构的问题，这是他们使用数据服务层的典型例子。在一定程度上，他们引进了面向服务架构，他们就理解了数据服务层。我要说他们大多数并不理解。有些经常和商务人士打交道的客户把大量的精力投入到将业务方参与到 SOA 战略中，但是他们的速度太慢，以致成果并不明显。

我们在 SOA 中的数据服务的采用程度如何？你有没有从中获利或者发现一些真实的应用程序？

数据服务的采用肯定在增长，我们已经有了 50,000 下载量。这仅仅是在我们有了 GA 产品之后过去两个月的下载量。当然人们不会在在两个月的时间里将理论付诸实践。从某种程度来说一切还为时太早，但是我相信数据设计师和开发商在 SOA 方面是理解的。数据服务层应用的另一更为广阔的领域就是丰富的网络应用方面。在这个领域，使用 Ajax 的屏蔽部分被服务所迁移。从数据的颗粒性角度来说，这确实是一个不同的领域。在该领域中你需要高度的粒状数据和许多服务，而不是最终会被业务和服务所驱动并且被业务所规定的 SOA 方面。这就导致你只有一些粗研磨的服务，但是我看到它在两个领域都有增长。

这就是我们要令其更加成熟的数据服务标准，有你需要的标准么？

我们的世界主要是以 XML 模式为中心。这是关键的标准，建立和 XML 模式相适应的数据类型。这还有一些关于服务调用的标准：WSDL 和 SOAP 标准以及 REST。

那些类似保险业 ACORD 为特定行业量身订做的标准如何，这些标准足够成熟么？

我认为 ACORD 发展状况良好。保险业比其它行业采用 XML 和 XML 模式都要多并且时间更长。在 XML 公布之后不久他们就开始研究 ACORD，由于行业的不同，成熟的程度也不相同，但是保险业发展尤为成熟。

考虑到你对 XML 的关注，XML 的数据十分整齐，在你做数据服务时会因为机构没有一直遵循标准而碰到问题么？

从标准的角度来说，它是十分整齐的。你会在用户想要应用标准的用户化过程中遇到问题。ACORD 是我们尤其需要考虑的一个标准。每一个主要信息组都有一个为用户设计的扩展区域。当然，你越将其用户化，你就增加了无法和其它公司配合操作的风险。从标准的角度来说，XML 非常整齐，但是在有配合操作性的地方肯定有用户化。这些都是人们在设计应用时需要考虑的因素。

当我们跨行业使用这些程序时，例如我们需要从保险业、金融业和医疗行业获取数据时，会遇到问题吗？

不会。你可以创建和任何模式相匹配的服务。该服务可以是翻译成数据服务模式中任何一个一个模式。

在数据服务开发中能够产生问题的最坏的实践是？

不合适的 XML 设计就是个问题。有时人们甚至没有设计出一个模式。当他们开始设计 XML 实例时，并没有将其按照逻辑组织。这就使得他们不得不在整个设计周期不断的做出改变。如果你的机构设计不是很合理，就会遇到再处理后端系统的麻烦。在你的逻辑视图和真正的物理结构之间，总会有阻抗失配。所以，你可以在保证自己的逻辑结构是合理的，你就能够避免这些麻烦。

(作者: Michael Meehan 来源: TechTarget 中国)

把握 JavaFX1.2 在富互联网应用方面的关键

在 JavaFX1.0 发布后不久，我通过这篇文章对它研究了一下。经过两年的时间，javaFX 已经发展到 1.2 版，新发布的版本在语言和 API 方面有显著的变化。JavaFX1.1 创建的应用程序必须被重新编译，甚至还需要稍微改动。

JavaFX 的基本特性还是用声明式语言格式来创建的 GUI（图形用户接口）时，这种语言格式在 java 成熟的“Swing”接口工具套件和各种动画、矢量图形、音频视频及所有运行在 java JVM（虚拟机）上的显示效果的基础上来定义接口组件。1.2 中许多新特性都和 flash 显示效果有关。在访问 Web 服务方面也有很好的补充。

JavaFX 语言类似 java 编程风格，创建计算功能的时候也需要扩展基类和实现接口。1.2 版本有一个令人惊喜的新特性：“mix-in”类声明修饰符。mix-in 类不能被直接实例化，和 java “抽象”类非常相似，但一个定制类可以扩展任意多个 mix-in 类。Java 在类层次结构中坚持单继承原则。如果你想用多继承，可以用这个试试。我担心 mix-in 是否会使标准 java 也随着退化。

下载 JavaFX 的 release 版本

JavaFX1.2 的 SDK 可以从这个网站下载，是一个 46M 大的安装包。和 JavaFX 之前发布的版本一样，你也可以下载一个附带 JavaFX release 版本的 NetBean6.5.1.，如果你已经安装了最新的 NetBean，只下载一个 JavaFX 插件也可以。虽然 Sun 网站说 java JDK5 够用，JDK6 的 13 升级版对开发者来说仍是最小化的，推荐 14 升级版。除了 Windows（XP 或 Vista），还支持 Mac OS，Solaris 和 Linux 仅在 beta 版本中支持。

这个 SDK 扩展了 API 文档，使熟悉的 JavaDoc 介绍绘声绘色。提示：Sun 网站不是所有的 JavaFX 在线参考材料都更新到 1.2 最新版本，所以要多留意。SDK 文档有很大的空

白，希望 Sun 能尽快补全。也许由于赶进度才使他们如此仓促。幸运的是 Sun 开发者和热情的程序员是博客的活跃作者，因此通过在 Web 上搜索也能找到很多有用的例子。

支持从 Web 服务获得数据

和使用任何语言作 GUI 编程一样，从来你不能使用线程来处理用户的键盘或鼠标输入来执行长期的任务。JavaFX API 有很多启动和控制任务的工具。对于 Web 服务客户端编程就要用 `javafx.io.http` 中的 `HttpRequest` 类。`HttpRequest` 让你指定一个 URL 和 HTTP 方法，启动一个持续的异步请求，还定义了 `Callback`，这样你的代码在请求被处理和请求返回时将会收到通知。HTTP 方法支持 GET、POST、PUT 和 DELETE，这些方法使访问 RESTful Web 服务变得很容易。

根据 JavaFX 开发者所撰写的各种博客，我期望 1.2 版本有一个类似 `javafx.xml` 的包和一个 DOM 模型类，这个以前是没有的。只有一般的能解析 XML 或 JSON 流，并在找到数据元素时通知你代码的“拉”模型解析器。当然，如果你想使用 DOM 中 XPath 这样的工具，可以直接使用标准 java 中的 XML 类库。除此之外，对 XML 和 JSON 还有很多 JavaFX 类库的开发者。

RSS 和 ATOM 是简单“喂”格式的 Web 服务，用来发布格式定义良好、更新频繁的小信息块。JavaFX 为依据 RSS 或 ATOM 标准信息组件从 XML “拉”解析器中解释事件提供特别的类。这些类使创建集成整个 Web 信息的“混搭”风格的应用变得容易。

部署 javaFX 应用程序

作为一个 applet 在网页中部署，这个网页使用 JavaScript 解释 JavaFX 应用程序描述并从 jar 文件中加载。使用 Java Web Start 可以很容易创建独立的应用。这项高度精简的技术也被当做 java 网络加载协议 (JNLP)，是标准 java 环境的一部分。使用 JNLP 就能构造一个 Web 网页链接，它将下载 javaFX 应用定义文件，并且在用户桌面通过下载 jar 文件来安装应用。这个 NetBean IDE 支持将应用打包，并将其作为 applet 或 JNLP 应用，用来创建所需要的支持文件。

RIA 技术的竞争格局如何？

富互联网应用领域的技术竞争仍然很激烈。我认为领先的 RIA 技术公司 Adobe 得益于长期以来大规模地安装 Flash 浏览器插件和 PDF 阅读器。使用 Flash 技术的桌面应用可以被 Adobe 集成运行时创建，现在到了 1.5.1 版本。Adobe 仍在进行一项野心勃勃开发，已经把主要的技术块转移到开源的许可体系之下。一个活跃的开发社区已经创建了大量的 AIR。

微软在 RIA 领域的入手点为 Silverlight，作为浏览器插件分发，和 JavaFX 相似，用户接口使用声明式 XML 格式，并利用 .NET 框架的编程逻辑来创建。支持从 Web 服务下载除了各种图形化媒体格式外的 XML 和 JSON 的内容，这是个巨大的进步。Silverlight 的第三个版本，现在发布了 beta 版，在其他各项改进中提供了一种在桌面上安装 Silverlight 应用的方式。

令我惊讶的是 Google 在 RIA 角逐中的表现。作为一个浏览器插件和桌面应用，和上面提到的串行化工具相反，Google 地球让我们看到了一个能被实现的，致力于某种特别数据的工具。巨大的研究预算和创新思维已经创建出大量的桌面应用、组件和浏览器插件。

(作者: William Brogden 来源: TechTarget 中国)

企业混搭：工具构建数据集成

组织苦于长期经营下的数据整合的突出积压、商业智能和自定制报告工作。多年来，各种各样的技术的兴起，解决了这些或类似的问题——最近，称之为企业混搭。

近年来，企业混搭作为工具出现，可让最终用户——通过开发人员的协助——建立自己的数据集成。服务对这些产品最为重要——往往也是 Ajax 的前端。尽管如此，企业混搭空间值得注意的是多样化的。数据分级，发布与订阅消息，XML 数据挖掘和其他特性，可包含在不同的时间内。

在一个统一的仪表板混搭一些来自内部和/或外部不同的数据源的现有数据。在许多混搭的企业中和相关的软件专家如 ActiveActive Endpoints, Actuate, Composite Software, Cordys, Denodo Technologies, IBM, JackBe, Kapow Technologies 等都这样做。

一如既往，在评估选择，文化依赖时，也就是说：不同的机构青睐不同的做法。如何使用户接近数据，分析减轻多少混搭带来处理数据的工作——这一点非常重要。

厂商关注不同的面向 BPM 的混搭，面向数据的混搭和生动的前端——终端数据可视化。此刻混搭人群的广泛兴趣是业务集成应用，即不同的数据源被敲响来提供一种有用的该公司的分析意见。

Forrester Research 的分析家 James Kobielus 列出企业 BI 的混搭类型的范围：

1、轻量级表达混搭拉动应用程序数据输出，使用户可以生成报告。没有数据仓库不是用在这里。

- 2、深层表达混搭由数据仓库和分析大量数据池的支持。
- 3、联合环境下全智能混搭。用户会看到一个统一的语义逻辑的虚拟化，可以用透明的方式从各种系统/数据库拖出数据。
- 4、通过在可以混搭什么的周围用 IT 治理策略全协作的混搭。此处的数据存储于中央存储库，并通过设置访问权限进行访问。

在这个分组中，第一层的混搭成熟化暗示着在今天的企业事先没有 BI 经验，而实际上是不存在第 4 层的。

Forrester 的 Kobiellus 表示有企业的地方就有多种方式实现混搭。集成风格取决于数据基础架构和企业文化。主要的厂商表示那些对企业 BI 混搭感兴趣的人应该看看 Microsoft Project Gemini 和 IBM Cognos。这两个产品足够帮助达到第 4 层混搭成熟。

但是在任何企业开始涉足混搭之前，需要方法落实。Kobiellus 表示“混搭不会在 BI 中成功除非它是有趣的，除非它是高度交互并且用户说‘哇，使用真方便，我可以用它创建我的报告’。如果 IT 迫使用户不愿自助服务，我认为这将会失败。用户的积极性将会推动这一点。”

Composite Software 数据虚拟化厂商市场营销执行副总裁 Robert Eve 认为“关键是要打破所做工作，我们工作在数据垂直面。面向数据的人是你的数据架构师。那么他们更多关注的业务消费者和企业应用程序使用的可视化。”

Composite Software 通过提供服务器虚拟化和从多个数据源的联合帮助客户开发商业智能 (BI) 混搭。实际上，它是一层映射数据和直接发送到系统提供的混搭。