

db4o | 开放源代码的面向对象数据库 | Java and .NET

使用db4o实现移动商务

作者 Eric Falsken 译 韩巍

新商务模型和企业组织的不断融合打破了传统的网络计算的边界，在这种情形下更高级的分布式应用程序和支持移动数据的需求变得越来越强烈。

作为填写表单、backoffice数据处理电子表格,或者当从旅途归来时浪费工作时间来输入数据的取代,现在移动办公人员使用手持设备、智能手机、平板或膝上型电脑运行**便携式客户端软件**来输入或取出数据,而且**这些设备**正在开始处理他们的数据并和backoffice系统同步来引发商业过程—例如,自动售货机跟踪记录本机库存清单来实现按需补货。

移动化使企业得到了更多利润和更快的执行时间,企业已经意识到移动化的价值并且已经开始升级自身的IT部门来加速推进这种把竞争优势带入基础部门的整合。

作为移动化趋势的结果,Java和.Net开发人员正在探索应用在分布式、客户端为中心的设备上存储数据的解决方案,并且发现传统的关系型数据库在运行环境需要零管理、占地小和灵活的时候的不足之处,比如,产品化的数据库在更新数据模型方面的能力已经被分散在成千上万的客户端字段内。

而目前大约**50%**的开发者仍然自己编写持久化解决方案来满足这些需求¹,现在一个快速成长的用户社区选择了db4o的面向对象数据库管理系统设计用来满足上面描述的确切需求。

一些公司包括 [Hertz, Eastern Data,](#)

[Mobilanten and Mandala](#) 已经在他们初步的移动解决方案中使用了db4o,而且很多正在开发额外的产品过程中。

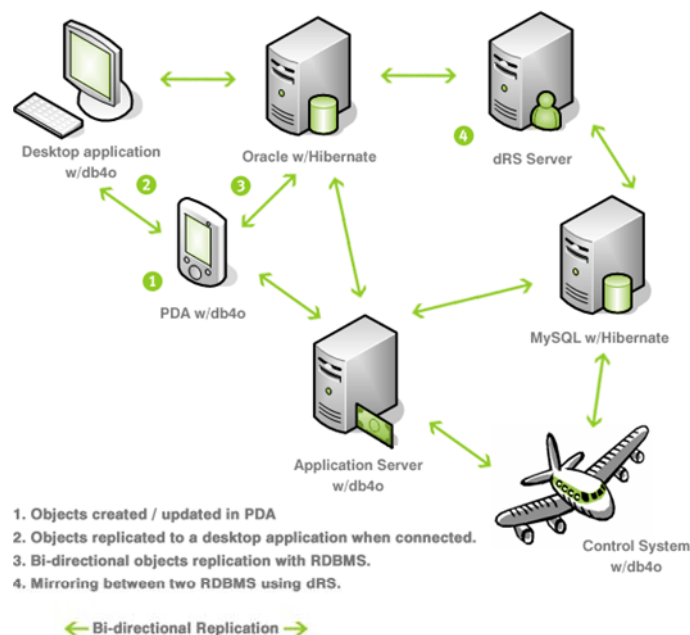
db4o不仅为无缝集成以客户端为中心的持久层提供可移动的,原始语言数据库。从2006年1



Eric Falsken 是db4o在分布式医疗设备领域的长期用户,而且最近作为技术传播者已经加入到db4o的开发团队。

作为一个有经验的企业应用开发者和早期Microsoft .NET技术创始人(他并不太钟情开放源代码运动合作)他喜欢探索开发工具以使开发优美的、有用的、可维护的软件成为可能。

作者博客: www.everylittlething.net



月起，它还提供依靠db4o复制系统(dRS)，基于Hibernate的一种整合分布式、面向对象的backoffice数据，以服务器为中心的，例如Oracle或MySQL这样的关系型数据库的解决方案。

在这篇文章里，我想要通过一个具体用例详细讨论移动企业解决方案以突出挑战性，使用db4o的移动数据库来表明具体的好处。

1. 案例

可移动企业级应用的传统结构是网络化的，在以服务器为中心的数据中心运行着一个或多个中央数据库实例，和许多分布式，周期性连接的客户端。周期性连接意思是：在某些时候，你是离线的，因此数据需要永久的存储在客户端上。但是在另一些时候，你是在线的(无线的或者直接连接到PC或某个网络)，从而能够在数据中心和客户端设备存储中同步数据。

我们假设一个地区销售代表，Larry，他在一家生产装饰品的公司工作。Larry的智能手机上运行着一款以db4o为数据库的软件，并且可以与他公司的CRM和订单处理系统连接。

Larry正打算离开办公室去拜访一些他所负责区域的客户。因为他知道当他在路上的时候不能很好的接收数据，他点击屏幕上的某个按钮与桌面电脑同步他的智能手机，得到他所有客户和他们的相关业务信息(例如订单)的最新数据

当到达第一个客户那里时，Larry发现他要延迟上一笔订单的交货日期。因为Larry已经有了他所有客户的订单信息，Larry在他的智能手机上找出这个订单并且商定了一个新的交货日期。

与客户商议需要制定的日期时，这款软件会根据商业规则来检测期望制定的日期是否可用，通常商业规则是和“订单”对象存储在一起。商业规则会监测多种商业因素，例如公司产品进度优化算法、某些基本库存和关键材料可用性信息。比如系统刚刚更新了一个新的商业规则，作用是用来监测某个反映产品规范中新近变化的关键输入组件的可用性。由于数据同步的面向对象本性和db4o的自动语法演化，智能手机上会已经存储这个信息而不需要再做额外的管理工作，例如在关系数据库中建立或修改某个table。由于这种更新的灵活性，Larry的公司发现更新产品进度表比以前更加简单方便，因此保持了市场竞争力。

db4o案例学习

- 移动篇 -

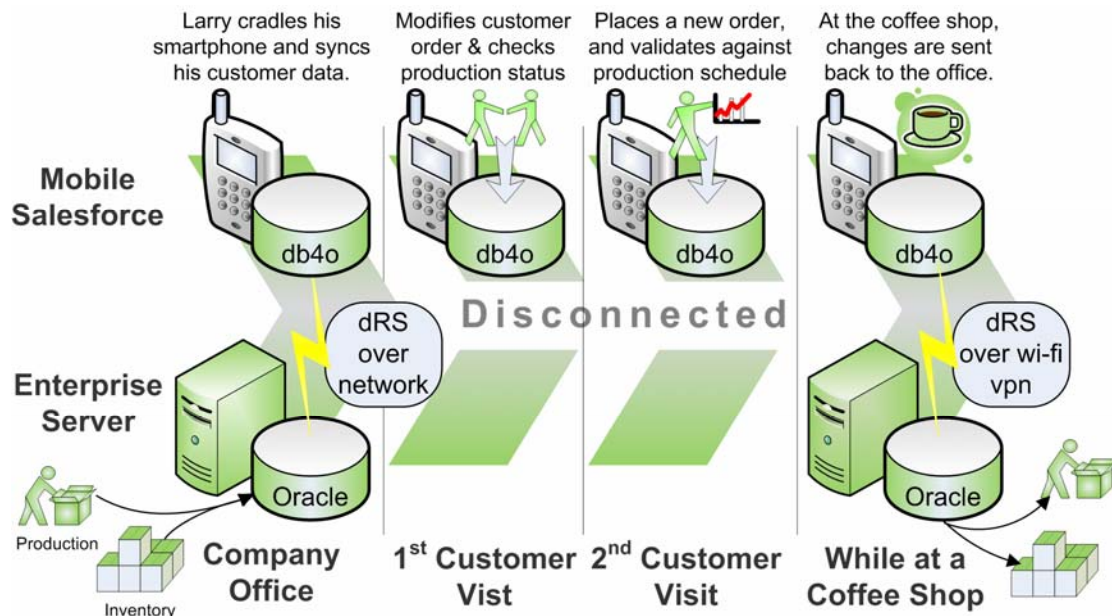
<http://www.db4o.com/about/customers/platforms/mobile.aspx>



Eastern Data 的移动应用程序



Mandala's Smartphone 软件



在第二位客户那里，客户的订单已经交货。客户想要增加订货，所以Larry找出客户的上一笔订单并且增加了一笔新的具有相同产品的订单。

之后Larry在一家咖啡馆短暂停留了一会，他打开他的智能手机，并通过GPRS让手机连接到他的办公室。轻轻点击一下，他的更新的订单就与应用程序服务器完成了同步。两项任务被添加到任务列表里：那笔修改交货日期的订单已经开始生产，所以不能修改交货日期。他电话通知了客户。第二位客户的新订单已经被核准，正在计算交货日期。他同样打电话给第二位客户来商定交货日期。

整个数据同步过程实际发生在Larry的应用程序内部。内嵌的db4o复制系统(dRS)为自动复制任何修改后的数据对象提供了可能。整个复制过程与传统的RDBMS复制能力相比可以包含复杂许多倍的商业逻辑。例如，在这个案例中，产品团队已经开始装配第一位客户的订单，因为这个订单上的产品已经被锁定，所以同步请求被驳回。如果不使用db4o，我们必须手动将这个逻辑编写进网络应用程序内。然而，通过使用db4o和dRS，我们需要编写的代码仅是一个简单的检查订单是否已经被生产的冲突检查对象。dRS的应用程序接口同样胜任自动任务执行，事件产生、或提示用户做出选择。

一旦数据和办公室内的应用程序服务器完成同步，Larry所做的改变对其他用户立即可见。新订单开始被产品部门处理，购买订单所需要的材料会自动产生。

使用传统技术也可能完成所有这些事情，但是那会包含许多关系表格，所有这些表格需要被僵硬的、静态应用程序代码所引用，由许多行非原生SQL字符串组成，结果使重构变得异常笨重。快速进化的产品在基础层面上维护所有这些代码将会是很耗时的。对于每个平台所有这些工作也



不得不重复一次：从移动设备到应用服务器，再从应用服务器到销售团队的分析服务器。与此相反，db4o只需花费开发人员的很少量时间在应用程序中添加几行代码来完成这些。

2. 用db4o实现分布式应用程序

大多数流行的应用程序都是用面向对象语言开发的，大多数真实世界的数据也很容易用面向对象语言表示，从这方面讲开发人员也应该考虑使用db4o。

在商业程序里，使用“客户对象”表示比用“客户表格里的一行数据”表示更加有价值。例如：客户对象可以包含这样的逻辑，决定当客户可以订购某个产品时是否允许其订购，然后算出相应的价格。传统的做法，开发人员不得不编写大量的数据库查询并用很多方法计算出结果，然后这样的逻辑拙劣的被打包重用并存储到移动设备上。

db4o允许开发人员把精力放在如何让应用程序更好的工作而不是花时间在设计数据库schema并编写持久化代码上。实际应用中，已经证实db4o可以减少开发数据持久层90%左右的费用。

分布式db4o应用程序也同样简单。db4o设计用来内嵌入软件组件，并且对于最终用户完全不可见。部署db4o就如同你在安装程序中包含一个单独文件那样简单。没有安装、配置、维护数据库方面的花费。

此外，db4o会自动适应应用程序的升级。它假定没有数据库管理员，并且允许应用程序从旧数据库更改到新数据库的无缝转换，结果是：简单、平滑、快速的部署和低廉的支持花费。db4o几乎完全消除了当你发布新版本时用来维护数据库方面的费用

db4o允许你用Java 或者Microsoft .NET语言开发的应用程序部署在PDA、移动电话、Windows桌面、Linux或Java服务器上，而且他们之间可以无缝的互相通信。

3. 收集你的分布式数据

db4o独特的面向对象复制功能很简单地允许你在db4o数据库之间，类似Oracle,SQL Server,MySQL这样的关系型数据库之间，和他们任意组合之间实现数据同步。通过使用流行的Hibernate对象—关系映射机制实现了对关系型数据库的包装。

dRS是100%的面向对象软件模型。正是这样，它对于采用快速开发软件环境的敏捷式企业来说是种完美解决方案。因为dRS设计用来嵌入您的应用程序，它在您的程序代码内保留您所有的领域逻辑和商业过程，而不用分散存储在企业服务器不同系统内。因为每个独立系统都需要额外的独立开发工作。取而代之，dRS允许您专注于独立的应用程序开发和整合。dRS能够减少开



发时间和分布式功能的测试时间，让你可以专注于你的商业逻辑。

dRS系统一个比较独特的特点是解决应用程序完整性冲突的功能。传统的复制只关注数据库的结构，这意味着复制处理的是行级别或列级别的改变。然后通常这种“行”和“列”的表达式对大多数应用程序没有太多意义。同步真实的“客户”和“订单”对象不是更好吗？实际上如果使用dRS的话，大多数情况下在两个数据库间解决一个单独的冲突所需的代码量几乎可以忽略不计：因为这就象写一个If语句那样简单。

当冲突出现时，传统的数据复制系统是向处理器发出一个冲突日志指令。处理器向数据库添加标识，应用程序开发人员在处理他们之前不得不做定时性的检查。而如果复制系统是与应用程序整合在一起的，应用程序就可以立即接收任何冲突的变化情况。冲突就可以立即被应用程序或者用户处理。

任何企业级开发成果中可维护性都是很关键的。当开发人员听到“复制”这个词时，他们中大部分都“敬而远之”。这是因为对数据格式的任何改变都需要与所涉及到的数据存储同步，其中包括复制脚本，和所有应用程序涉及到的可能使用的那部分数据访问代码。使用db4o和dRS则意味着无论何时您的应用程序的对象模型改变，您的应用程序都可以无缝的更新到您的数据库。

4. 结论

对那些想要使用科技驱动商业过程来与自己的竞争对手或者包含发展进程中价值链占用的费用相区分的企业来说，移动商务是必然的趋势。

新的商业模型和过程需要正确的技术做支持。移动化软件不同于以服务器为中心的软件—领先的以服务器为中心的数据库(Oracle,IBM,Microsoft)在嵌入式数据库领域却是落后的，在市场占有率方面大概只占到2%-5%左右。

一批具有前瞻性思维的开发人员已经尝到了面向对象计算在构建更加分布式的移动商业程序中的益处。他们把面向对象带到了数据层来允许快速应用开发(这带来低成本和快速投入市场)并且增加更多灵活性，因此敏捷的商业过程正以一个前所未有的速度快速改变着商业环境。

db4o是仅有的、明确的、以客户端为中心而设计的，Java和.NET环境可移动持久化原生面向对象数据库。新发布的dRS通过提供一个开拓性的复制系统确立了领先地位，它允许应用程序即使是把最复杂的对象结构与遗留下来的关系型数据库进行同步也仅仅是一行代码。