



电子质量管理(e-QA)

EQA4398(v1.3) 2011 1 月 10 日



功能

- 监测远程的，以致海外的生产厂产品的质量测量数据。
- 和本公司的以 GPRS/GSM 为基础的数据采集和处理系统结合在一起可监测在使用现场的产品的质量数据
- 瘦客户机： 客户端只完成质量数据的读出和转换成数字信号串，并送往服务器，由服务器完成对质量数据的译码。
- 对于新产品和老产品的改进设计具有极强的适应性和扩展性。
- 可以使用条码扫描器来完成库存控制和管理。
- 可跟踪产品在生产厂，外包厂，推销商，服务中心和顾客之间的流动状况。
- 数据在使用者和应用服务器之间流动时具有很好的安全性和保密性。
- 对于数据访问的范围，不同级别的机构和使用者有不同的限止。
- 支持多种语言（中文，英文）

1. 背景

随着生产的全球化，产品竞争的激烈化，对生产厂的服务提出了越来越高的要求。IKM 公司应顺这种需求，成功地开发出了远程质量管理体系（r-QA）来帮助公司对远程外包生产的产品进行就地和实时的质量检测和管理。

电子质量管理体系（e-QA）是继远程质量管理体系之后，开发的又一个质量管理体系。和 r-QA 相比，e-QA 提供功能更强，更有可扩展性和更安全的性能。e-QA 的应用程序是使用 Java Enterprise 版本 6（Java EE 6）的技术。e-QA 具有服务器的分布性，设计的组合性以及服务器，操作系统的适配性等特点，从而它能在产品生产现场或产品的使用现场对产品的质量进行监测。

服务器的分布性:

当 e-QA 请求一个操作时，它不是分配给一个指定的服务器，而是分配给一个负担最轻的服务器，以达到最快的服务速度。当对某个服务器的服务请求的数量增加时（用户/被监测的产品数量增加），其他负担不太重的服务器可使用到系统中来，以满足服务请求增长的需求。每个新使用的服务器将简单地重复原有服务器的行为（e-QA 应用程序的程序，主要的 Java 服务器管理元件结构等）。这个概

念和云端计算相类似。

设计的组合性:

组合式的设计就是对于一个包含多个任务的操作，只有当该操作中所有任务全部成功地完成后，才去更新数据库里的数据。e-QA 就是使用这种组合式的设计方式来保证数据库里数据的正确性和连贯性。如果一个多任务的操作中，有一个任务由于寄存器操作出错，数据库操作故障或者人为误操作而失败，那么该任务之前的一系列的成功任务会回到原来的状态，不更新数据库里的信息，从而使数据库中的数据仍能保持正确和连贯。通过使用 J2EE 提供的 Java 组合应用接口，e-QA 可保证所有操作在发生软件和硬件故障时将不影响数据的正确和连贯。

服务器，操作系统的适配性:

e-QA 发出的应用服务请求可在任何与 Java EE 相兼容的服务器中得调配和实施，如 glassfish, JBoss 等等。由于应用的客户都是基于 Java 的，所以 e-QA 应用软件可以适配于各种操作系统，如 Java 的虚拟机 (JVM), Windows, Linux, Unix, Mac, 等等。

2. 概述

e-QA 由三部分组成：应用客户，应用服务器和数据库服务器。如图 1 所示

应用客户:

e-QA 的应用客户可有下面几种类别：

- i) 在外包生产厂的一台工作站计算机，它安装有 e-QA 的软件。该工作站通常是直接从产品的本身来测量产品的质量数据的。得到数据后，数据通过以太网送到 e-QA 的应用服务器。
- ii) 在现场运作的产品，它安装有 GDCP 系统（GPRS 为基础的数据采集和处理系统）。该 GDCP 系统在正常工作模式将周期性地测量产品的质量，并把测得的数据通过 GSM 网和以太网传送到 e-QA 的应用服务器。
- iii) 一个网络浏览器，它和应用服务器的 e-QA 的应用网站组件层相联接。当使用者打入登录的名称和密码后，根据该使用者在所在机构中的地位 and 所赋予的权限，能浏览和管理与所赋予的权限相应的数据。

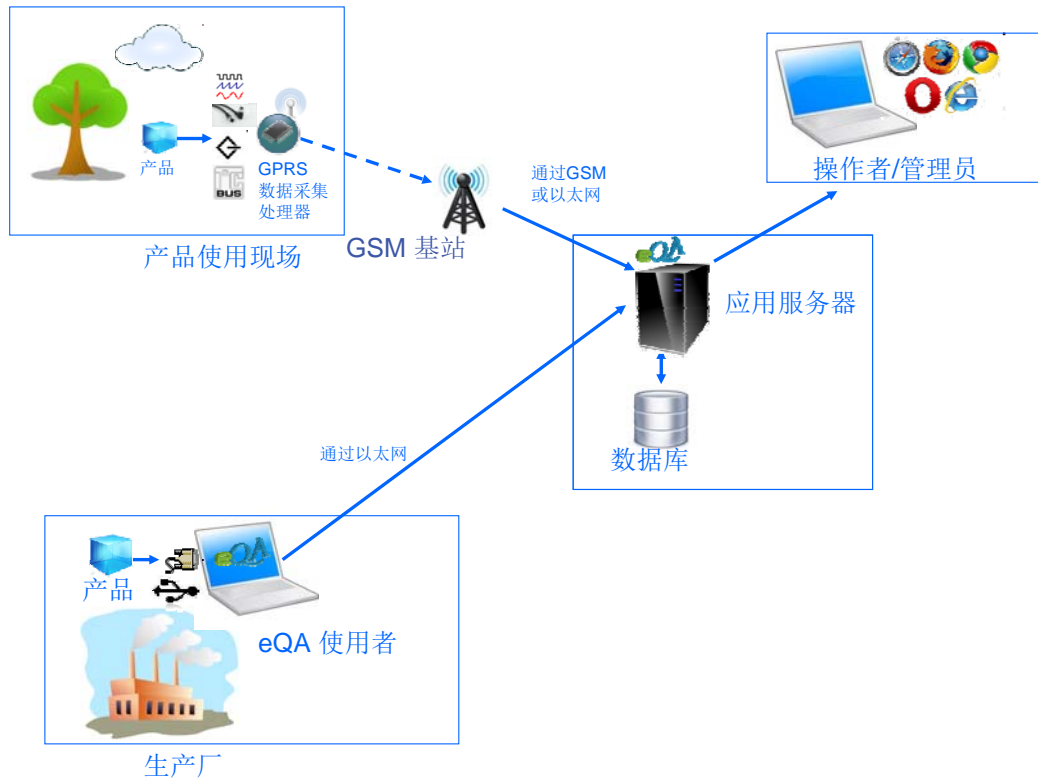


图 1

应用服务器:

应用服务器掌控大量 e-QA 的应用软件程序, 即: Java 服务器管理元件结构(EJB), 它包含有主要的 e-QA 的商务逻辑和提供与 HTTP 请求以及网络服务接口的组件。

在生产厂方面的使用者可以通过两种方式和 e-QA 的在服务器上的应用商务软件相连来提交测量到的质量数据: 通过 EJB 远程接口 (JNDI 的查找) 或者网络服务。

对于已经在现场运行的产品, 它们要通过由 HTTP GET/POST 方法所提供的 GPRS/GSM 把质量数据上载到 e-QA 的应用程序中。

对于那些希望浏览和管理已经采集到的数据的使用者和管理者来讲, 他们只需要简单地进入 e-QA 应用程序的统一资源定位器 (URL), URL 由在应用服务器的 e-QA 的部署所指定。

数据库服务器:

e-QA 的商务应用是与各种各样的数据库服务器相兼容的,如 MySQL, Apache Derby, Oracle, 等。这是由于使用了 J2EE 6 技术的 Java Persistence 语言, 其他 Java 的资源 and Java 的应用接口才得以有实施的可能。

2. 瘦客户机

现今大部分高档的产品都安装有微控制器。这些微控制器 可用来报告产品的现况等与质量有关的数据。对于车辆而言, 如火车, 轻便的电动助动车, 传统的汽车都具有关键的元件, 比如电动助动车的电池组是个关键元件, 电池组内可有个小型的微控制器, 它可读出电池的温度, 剩余的电能, 剩余的时间等参数。一台计算机可把这些信息再从那个微控制器中以数字串的型式从各种接口读出: 如串行口, USB, CAN 总线, SPI 总线等。这些数字串被译码后, 以人们可阅读的型式 (如 XML/HTML) 显示给生产厂或服务中心的操作者。

这些有价值的信息可被送到服务器里存储起来, 并加以管理。换句话说, 这些从设备来到原始数字串的译码可以在客户端完成, 而译码后的数据可存在服务器。但这种方式, 微控制器要做很多计算和译码工作。

e-QA 使用了不同的处理方式: 客户端只从设备中读到以数字串格式的质量数据, 然后把整个数字串简单地送到应用服务器中, 这个过程被称为“请求”。该数字串在服务器那儿被译码成人们可读的数据, 再送回到应用客户端, 这个过程被称为“应答”。图 2 显示了这两个过程

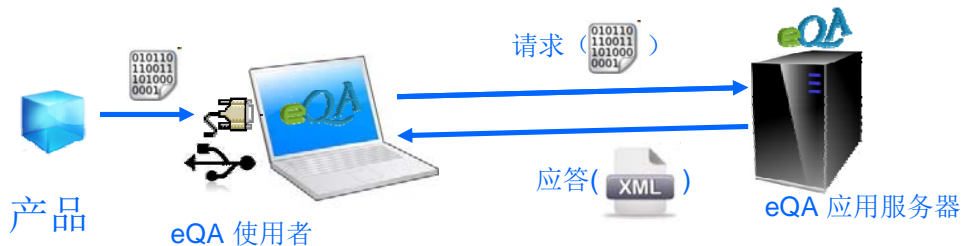


图 2

这种处理方法有如下的优点:

i) 在应用客户端有很少的计算。客户也不用对原始数据进行解码。由于不需要计算, 这样的客户称为“读送”客户。这种情况对于已经在现场运行, 而又要通过微控制器和 e-QA 应用服务器通信的产品特别重要, 因为一般微控制器的计算功能是十分有限的, 所以瘦客户机是个理想的处理方式。

ii) 扩展性: 如果一家公司开始生产新产品, 使用这种方式, 客户应用软件不需要修改就可以容纳新产品的质量参数。只要产品输出二进制的数字串作为它的质量数据, 安装在不同计算机上的客户应用软件也不用改变。这是因为数据串的解码是在服务器端完成的, 服务器端的程序只需要有一次更改即可。

iii) 适应性: 当现有的产品进行改进设计时, 它的有关质量的参数有可能改变。如果对原始质量数据的译码在应用客户端实施, 那会导致已经安装在所有计算机中的所有有关软件都要更新。但 e-QA 中该译码工作在应用服务器的程序中实施, 所以只要对该应用服务器的程序进行一次修改即可。

应用客户可以安装插件式的软件, 如果与网络连接中断的话, 它可被启用对原始的数字串在本地就进行译码。但是当译码过程有变化时, 这种本地插入式的译码软件必须更新。

对网络中断的另一个备用方法是把数字串临时地存放在应用客户的工作站计算机中, 当网络恢复后, 再把数据串送到应用服务器进行译码。

3. 可扩展性和云端计算

当有成百上千的应用客户送出数据, 等待译码时 一种可能的办法是“分流法”也就是把重载的译码计算移到下侧的服务器中。只也就是云端机算的基本原理。

非常幸运, e-QA 的管理程序是用 J2EE 技术的, 而 J2EE 技术是一种云端计算完美的工作平台。

什么是云端计呢?

云端机算是一个普及的, 方便的工作模型或模式, 它可使所需的网络访问到有各种可配置计算资源的共享源。

资源

- 服务器,
- 存储器,
- 应用程序
- 等等

云端计算的概念满足了资讯科技 (IT) 极为重要的要求

这是一种非常好的方法: 不断地增加性能而又不必投资于基础设施, 新员工的训练和要许可证的新软件。另一方面, 当一台应用服务器由于应用客户的众多请求而超载时, 在云端计算中, 只需简单地加入另一台应用服务器就可在过程的进行中解决问题了, 系统只需要做极少的工作即可。新加入的服务器只需要重复原来服务器的行为如 e-QA 的应用软件。相类似的, 如果需要增加一台数据库服务器来得到更多的存储空间或为了冗余的需要, 也是同样的处理方式。这些都不会影响 e-QA 的运作因为 e-QA 是使用 J2EE 的持久性存储(persistence), 资源和应用接口 (API) 的连接来和数据源进行通信的。关于 J2EE 它的服务器下面还有详述。



图 3

图 3 显示了 e-QA 是如何解决扩展性问题的。EJB 代表 Java 服务器管理元件结构。它们在本质上是一种自容式的自我执行的软件。EJBs 能够通过产生它的不同软件的实体而被复制。每个 EJB 通常是设计来执行某个特定的任务的，在 e-QA 的情况下，它是接受数据串，进行译码，把译码后的结果以 XML/HTML 的格式送回。对于新到达的请求，一个 EJB 的译码软件实体被产生，由于每一个软件实体都是简单地互相复制，所以译码将以一种划一的方式进行的。

当太多的请求以排山倒海之势到达应用服务器时，第二个服务器会启动了。第二个服务器将有和原来服务器同样的译码 EJB。当原来的服务器收到新的请求时，它会该请求转送到第二个服务器，第二个服务器将产生一个译码 EJB 的软件实体来执行译码操作。

非常明显，这种容纳新的资源（新的应用服务器）所需的额外的工作是非常小的，而系统的能力是翻倍了。

这种重新分配请求的构架对于 e-QA 的应用来讲是透明的。该应用只是收到请求，处理它和做出反应。

4. 安全性

如图 4 所示，e-QA 的应用客户要用用户 ID 和密码来登录到应用服务器的。登录不光是认证使用者的身份，而且决定了该使用者有如何的权力来执行任务，包括该使用者能容许看到的质量数据的范围。



The image shows a login interface for e-QA. At the top, there is a logo with 'eQA' in green and blue, and 'IKM Technology' below it. Below the logo, there are two input fields: 'username:' followed by a text box, and 'password:' followed by a text box. Below the password field is a 'submit' button. At the bottom, there is a copyright notice: '© 2011 IKM Technology'.

图 4.

e-QA 能够工作在各种各样的安全环境中，而不需要改变它的原代码。这是因为 e-QA 使用了 J2EE 所提供的各种安全管理的功能。Java EE 的安全模式容许 e-QA 有权限限制系统的资源，只容许具有合法身份的使用者才能访问它们

一旦一个使用者登录后，该使用者就和有关的安全角色和安全机构相关联了。安全角色可以是一个测量者或一个管理者。安全机构可以是总部的生产厂，外包的生产厂，经销商，服务中心或最终用户。例如是一个测量者，他（她）只能看到自己提交的数据。如果是总部的生产管理员可以看到所有已出售的和尚未出售的产品数据，这样提供了产品被客户购买后支持的连续性。

所有在数据库的有关产品质量的数据，任何客户无权修改，（除了“顾客名称栏”可由出售前的生产厂名改成顾客的名字），以保证质量数据的权威性和可信性。



图 5.

e-QA 也提供传输层的安全以保证把数据在应用客户和应用服务器之间的传输时的隐秘性。该传输层的安全是基于 HTTP 传输 (HTTPS), 使用软件的安全插座层 (Secure Sockets Layer 或 SSL)

5. 用条形码的库存管理

e-QA 把质量管理和库存管理结合在一起。当产品首次生产出来后, e-QA 为该产品生成一个独一无二的条形码。当该产品的质量数据测量出来后, 有关的数据就和这条形码联系在一起了。根据这个条形码可以查找和检索有关该产品的所有质量数据。

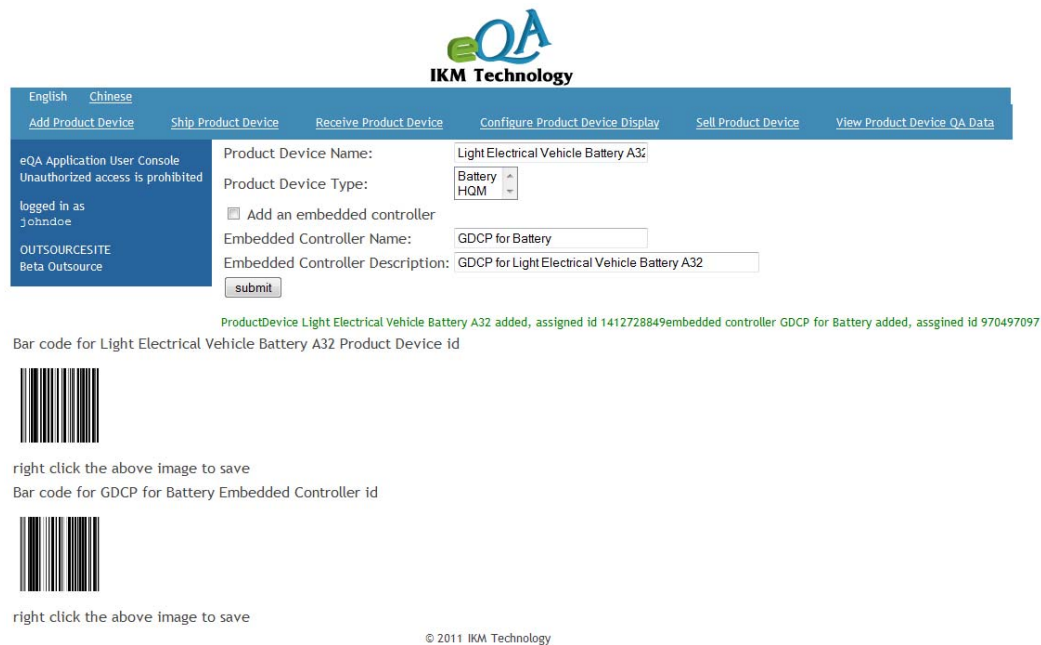


图 6.

图 6 是一个用于电动助动车的电池的例子。它有一个以 GPRS 为基础的数据采集和处理系统（GDCP）。该产品和它的 GDCP 由这个唯一的条形码所识别。该条形码的图像被下载和打印出来贴在该电池上。

当产品卖给经销商或顾客后，条形码可用来跟踪产品的流通和位置。当一个产品送到服务中心时，或送回到原来的生产厂时，它的条形码可用来检索原来测量的质量数据。

6. 支持多种语言

由于许多外包生产厂在不同的国家，e-QA 支持多种语言以便使。使用者可以很方便地按一下联接按钮把一种语言转成另一种语言。目前 e-QA 支持两种语言：英语和中文（见图 7）。支持新的语言可很方便地加入相应的属性文件。



图 7