

为硬件实时把脉的火星星人

由于市场上存在的自动化设备的类型甚多，且不同领域的设备之间存在极大的差异，过去若想远程监控这些自动化设备，只能针对某个固定的领域个别做项目式的系统开发。这样的差异性意味着较高的开发成本、维护成本并导致功能扩充与系统整合的困难。且看火星星人(MARS)如何克服这样的差异，用较少的成本轻松建置不同领域专属的，甚至跨领域的远程监控系统。

MARS - 可以一次实时监控多种跨领域设备的火星星人

MARS (Monitor Adapter Real time System)，顾名思义，这个火星星人是一种实时(Real time)的监控(Monitor)系统，它具有高度的适应性(Adapter)，可以应用到多种跨领域的自动化设备里。事实上，除了这些特性，它还是个远程服务，而且是跨平台的。

举例来说，台湾的工具机产业，产值是全世界第六大。体积庞大且设计精密的工具机，就是一种自动化设备。在过去必须由操作员亲自在机台旁透过控制器来进行机台的监控，如果机台出现了问题，只能「被动」的被操作员发现，而不能「主动」的通知相关人员，而且原则上一个操作员在同一个时间里只能控制一座工具机。

如果有一套系统，不论操作员人在哪里，只要有网络，输入网址就可以监控所有的工具机，那该有多好。再如果，这一套系统还可以在工具机发生异常状况的时候，发送电子邮件甚至传送紧急简讯到操作员的手机里，那就更棒了。这样的远程监控系统不但可以大量节省机台的维护成本，还能降低人为疏失，减少耗损。MARS 就是为了这样的需求而被设计出来的。

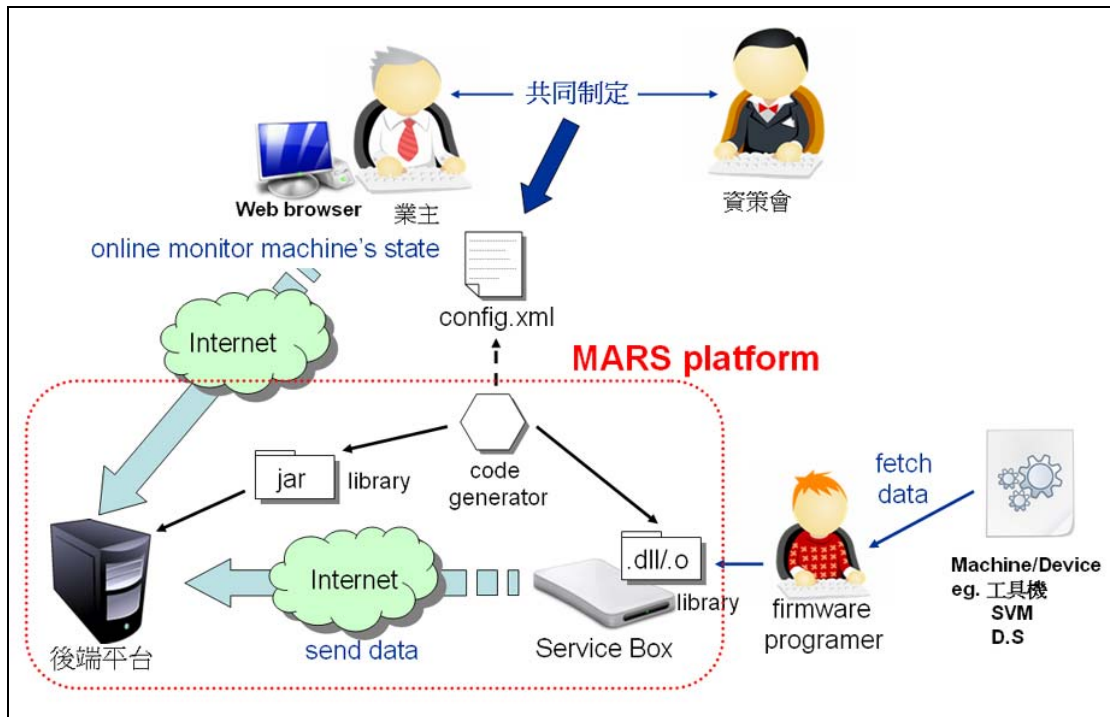


MARS 是实时的。为了达到「远程」监控的目的，MARS 是透过 internet 来传送信息的。透过 internet 传送信息，却必须同时具备「实时」反应的能力，MARS 运用了传输速率较快的 Message Queue 协议以及 RIA 的因特网技术，大量减少网络传输的信息量，让监控数据的显示图面，可以达到每数个毫秒就更新一次。应用 Ajax、Adobe Flex 这些 RIA 技术让 MARS 除了可以大量降低网络流量之外，也可将服务器的运算负载大量转移到客户端。网络流量以及伺服端的负载若能有效降低，可以同时监控的设备数量就能增加，使得 MARS 具备集中控管较多设备的能力。

MARS 是具高度适应性的。透过 XML 标准化的对象格式技术，即使要监控不同类型的设备，只要将监控数据的格式予以标准化，MARS 就可以直接监控，不需要另外做任何的客户定制。因为这样的特性，MARS 同时也是跨领域的。同一套系统，不但可以用来监控工具机；也可以让自来水公司同时监测园区内的净水场、取水站、加压站及监测站；或是让儿童游乐园管理处同时监控园区内的数字广告牌和自动贩卖机。

MARS 是开发快速的。MARS 自行研发的原始码产生器，会根据标准化后的监控数据格式，自行产生相对应的原始码，不需要工程师为每个不同的装置另外开发。简单来说，想要安装一套 MARS，只需要标准化所有的监控数据，然后直接交给原始码产生器，开发就完成了，未来甚至可能只需要文件作业，完全不需要工程师的帮忙。

客户的反馈会让 MARS 质量越来越好。不管客户有多少，MARS 只有一套。和一个客户一套的项目式系统开发方式相比，因为 MARS 只有一套，所以每个客户的建议和回馈，都能实实在在的提升了 MARS 的质量。一个客户的回馈，可以让所有的客户都获益，MARS 的质量只会随着时间越来越稳定而强大。



让「服务」也可以被自动化生产

在过去的软件工程应用之中，「原始码产生器」常被设计用来加速程序的开发速度。**MARS** 不只是开发了自用的「原始码产生器」，也延伸了「自动产生」的概念，希望「服务」也可以被自动化生产。

自动化产生的「服务」可以视作是一连串的自动化过程。以 **MARS** 的应用步骤来说，第一个步骤是利用自行开发的「原始码产生器」读取一份描述所有的监控数据以及被监控设备细节的 **XML** 设定文件，自动产生相对应的原始码同时编译，接着根据底层的操作系统将编译后的程序包装成与操作系统兼容的函式库。例如前端 **Service Box** 的操作系统若为 **Windows**，包装为 **dll** 函式库；而 **Linux**，则包装为 **so** 函式库。后端平台因为使用跨平台的 **J2EE** 开发，则一律包装为 **jar** 函式库。函式库安装之后，同时自动建立对应的关系型数据库，最后整个服务就建置完成了。

这个「服务产生器」将冗长的服务开发流程改为自动化完成。如果要建置一个 **MARS** 监控系统，唯一需要耗费人力成本的工作，只有制定一份描述监控数据和监控设备细节的 **XML** 设定文件。

因为 **MARS** 可以自动产生服务，这一个完整的自动化过程不但加速程序开发速度，同时也能降低开发成本；而自动产生的原始码，因为不需工程师为不同装置重新开发，也兼顾了系统质量，大大降低臭虫的发生机率。