

**SAE技术**  
文件系列

**2006-01-1568**

---

# 使用ODX诊断仪的经验 数据库标准

**Kevin Mullery, Gareth Leppla, David Boyd and Brendan Jackman**

沃特福德技术学院



ISBN 0-7680-1633-9



**SAE** *International*™

2006年SAE世界大会  
密歇根州底特律，  
2006年4月3-6

**400 Commonwealth Drive, Warrendale, PA 15096-0001 U.S.A. 电话：(724) 776-4841 传真：(724) 776-5760 网址：  
www.sae.org**

工程会议委员会已批准本论文发表。在会议组织者的监督下，它已成功完成了SAE的同行评审程序。这个过程需要至少三（3）个行业专家的审查。

保留所有权利。未经SAE事先书面许可，不得以任何形式或通过任何手段，包括电子、机械、影印、录音或其他方式，对本出版物的任何部分进行复制、存储在检索系统中或传播。

关于许可和授权请求，请联系。

SAE权限

400 Commonwealth Drive  
Warrendale, PA 15096-0001-  
USA 电邮：

permissions@sae.org 电话：

724-772-4028

传真：724-776-3036



如需多份打印件，请联系。SAE

客户服务部

电话：877-606-7323（美国和加拿大境

内） 电话：724-776-4970（美国境外）。

传真：724-776-0790

电子邮件。CustomerService@sae.org

**ISSN 0148-7191**

**Copyright 2006 国际汽车工程师学会**

本文中提出的立场和观点是作者的观点，不一定是SAE的观点。作者对论文的内容负全责。如果论文在SAE Transactions上发表，可以通过一个程序将讨论内容与论文一起印刷。

希望提交论文以考虑由SAE发表或出版的人，应将手稿或300字的摘要寄给SAE工程会议委员会秘书。

在美国印刷

# 使用ODX诊断数据库标准的经验

Kevin Mullery, Gareth Leppla, David Boyd and Brendan Jackman

沃特福德技术学院

Copyright © 2006 国际汽车工程师学会

## ABSTRACT

ODX（开放诊断数据交换）是由欧洲ASAM集团指定的标准诊断数据格式，以简化制造商、供应商和服务经销商之间的车辆诊断数据交换。ODX数据库包含整个车辆的诊断数据，以及所有车辆ECU的详细信息。

本文概述了ODX诊断数据库中包含的主要数据类别。开发了一个基于Windows的诊断应用程序，在ECU上执行ISO 15765诊断服务，使用ODX数据库来定义每个ECU的允许服务和参数。本文描述了诊断应用程序的结构以及处理ODX和为生产ECU量身定做的必要步骤。

## 简介

ODX是一种用于存储和检索单一车辆的电子控制单元（ECU）诊断数据的格式。设计一辆车的过程通常涉及许多公司开发的系统、子系统等，因此这意味着大量的ECU诊断数据在这些公司之间传递。这种方法的固有问题是，它缺乏适当的结构，公司之间传递的数据通常是纸上谈兵，而且通常没有商定的格式。

ODX试图通过允许将数据输入到一个标准格式中来解决这个问题。然后，一个ODX格式的数据库（ODX XML文件）可以为一辆车建立，包含该车的所有ECU诊断数据。一旦这个ODX数据库建立起来，它就可以传递给测试工程师和服务经销商，对车辆进行诊断测试和检查。

## odx信息

如前所述，ODX文件包含单一车辆的ECU诊断数据。它的设计是为了减少数据的冗余，也是为了保存对车辆进行诊断检查所需的所有信息。由于这个原因，它存储了许多不同的信息，其中一些可以在图1中看到。

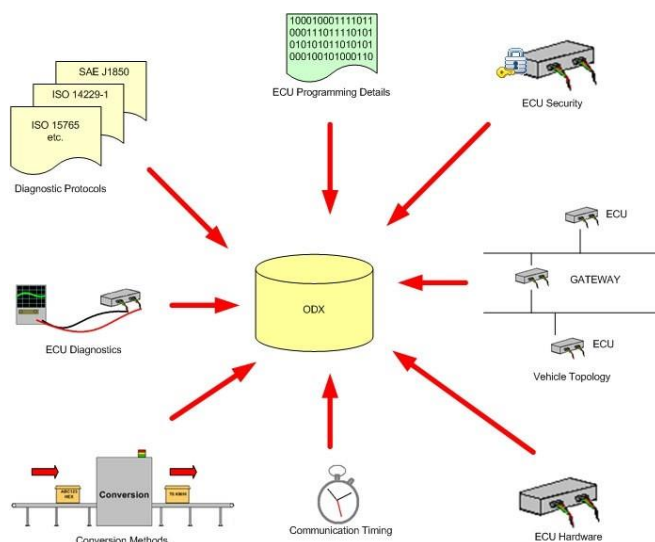


图1 ODX信息类型的子集

可以看出，ODX数据库所包含的信息是多种多样的，但它都有助于对车辆进行诊断监测。

图1只显示了ODX数据库中可以包含的不同信息的一个子集，还有许多其他类型。它所包含的另一类信息是行政和公司数据。这两类信息使ODX文件的不同区域可以完全追踪到填写者。这引入了关于ODX数据库的编制和使用的更多组织和问责制。

符合odx标准的工具

为了诊断测试和维修车辆，可以使用符合ODX的工具。这通常是一个计算机程序，它将连接到车辆或单个ECU，然后连接到该车辆或ECU的ODX文件。通过参考ODX文件中的不同信息，计算机程序将能够对车辆内的ECU进行诊断检查和编程。图2显示了连接到车辆的台式电脑上的ODX兼容工具的典型设置。

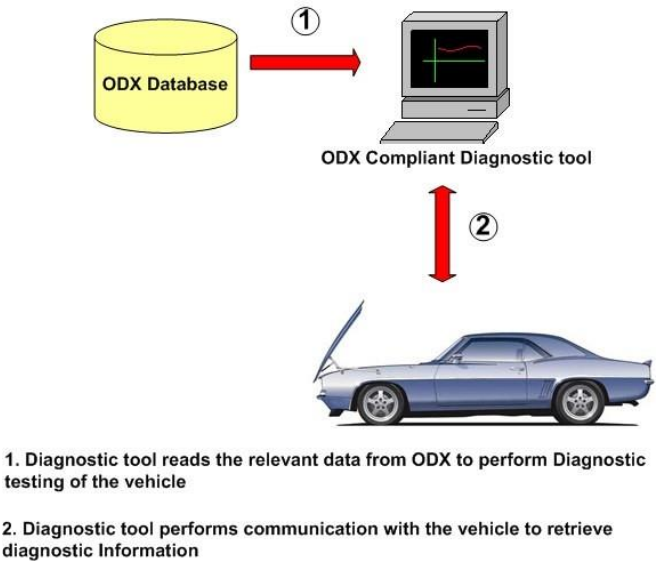


图2 兼容ODX的工具与ODX文件和车辆进行通信

ODX数据库使诊断工具更加通用。诊断一辆车的所有数据都可以从ODX文件中获取。因此，要对许多不同的汽车进行诊断，只需要一个符合ODX标准的工具连接到许多不同的ODX文件。工具所连接的ODX文件会改变，而不是工具本身。相对于在工具中存储静态数据，如转换算法，这是一个更好的方法。

XML概述

本文并不打算详细解释XML。然而，由于ODX规范是以XML格式编写的，快速的介绍将有助于读者了解本文的其余部分。

XML（可扩展标记语言）是一种存储信息的格式，与HTML（超文本标记语言）非常相似。它包含在被称为元素的命名标签内的信息，如图3所示，这些标签表明了开始标签和结束标签内所包含的数据类型。



图3 XML元素示例

以三本书的合集为例，可以说明这一点。

```
<Books>
  <Book>
    <Title>Java Gently</Title>
    <Author>Judith Bishop</Author>
    <ISBN>0-201-71050-1</ISBN>
  </Book>
  <Book>
    <Title>Modern Control Systems</Title>
    <Author>Richard C.Dorf</Author>
    <ISBN>0-201-03147-7</ISBN>
  </Book>
  <Book>
    <Title>Formal Specifications Using Z</Title>
    <Author>David Lightfoot</Author>
    <ISBN>0-333-76327-0</ISBN>
  </Book>
</Books>
```

图4 快速XML示例

在这个例子中，信息被存储在三本书上。可以看出，最外层的元素是'书'，它存储了关于一本书或多本书的信息。每个'书'元素依次存储了每本书的'标题'、'作者'和'ISBN'号码的信息。这些元素又存储了有关书籍的实际信息。从图4可以看出，一个元素可以包含另一个元素和文本。这些信息的结构非常好，很容易解释。

在ODX和本文件的其余部分中，诸如 "书籍 "这样的XML元素被称为对象。这个快速教程对于本文的其余部分已经足够了。关于 XML 的进一步教程，请访问 <http://www.w3schools.com>。

顶级ODX类别概述

ODX结构有五个主要类别。它们是：

- 图-层-容器
- COMPARAM-SPEC
- 多样化的ECU-JOB-Spec
- 车辆信息-规格
- 闪存

这五个类别在ODX文件中持有不同类型的信息，可以在表1中看到。

ODX-CATEGORY	信息类型
容器二层	诊断性
COMPARAM-SPEC	沟通
多样性--ECU--工作--规格	用于与多个ECU通信

车辆信息-规格	车辆信息
闪存	ECU编程

表1 ODX类别

一个ODX XML文件（ODX实例）一次只能包含一个类别。这意味着诊断信息和通信参数信息不能被保存在同一个ODX XML文件中。图5显示了两个独立ODX文件的开始，第一个将包含诊断信息，如DAG- LAYER-CONTAINER对象所示，第二个将持有通信信息，如COMPARAM-SPEC对象所示。一个文件不能同时包含两种类型的信息。

```
<ODX xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <DIAG-LAYER-CONTAINER ID="DLC.ODX_EXAMPLE.ID">
    <SHORT-NAME>VehicleADiagnostics</SHORT-NAME>
    <LONG-NAME>Vehicle A Diagnostics</LONG-NAME>
OR
<ODX xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespace
  <COMPARAM-SPEC ID="COMPARAM.ISO14229.ID">
    <SHORT-NAME>STDISO14229COMPARAM</SHORT-NAME>
    <LONG-NAME>Standard ISO 14229 Communication Parameters</LONG-NAME>
```

图5 ODX XML中的不同类别。

对每个ODX类别进行了简要的概述。diag-layer-

#### container

这是ODX类别中最大和最重要的一个。这个类别包含ODX文件所建模的车辆的诊断信息。本文后面将更详细地讨论这一领域。

#### 多样化的ECU-JOB-Spec

该类别用于允许一个应用程序同时与一个以上的ECU进行通信。这类似于功能寻址，但不使用功能寻址。

#### COMPARAM-SPEC

该ODX类别持有允许与ECU有效通信所需的信息。这种通信参数的例子包括。

- 一个ECU的目标地址
- 与ECU通信的超时值。
- 一个信息请求可以被发送到ECU的最大次数。

本文后面将更详细地讨论这一领域。车辆信息-规格

这个类别涉及到有关物理车辆的建模信息。这分为两个主要领域。第一个区域涉及实际识别车辆的信息，如OEM、MODEL-Year、VEHICLE-MODEL等。第二个区域是关于车辆网络拓扑结构的信息。这包括 VEHICLE-CONNECTOR 等信息，它通过使用 VEHICLE-CONNECTOR-PIN对象保存每个针脚的信息来模拟车辆的各种连接。其他对象对ECU之间的通信链路进行建模，并持有诸如物理通信链路的类型（例如CAN、LIN）等信息。网关也被建模，并且与诊断模型也有各种联系。这种拓扑结构的建模允许应用程序快速和方便地访问各种ECU，而不考虑网关或总线类型，这对用户是透明的。

#### 闪存

该类别允许应用程序和ECU之间的数据传输（ECU内存编程）。这里的数据可以被下载到ECU或从ECU上传。ODX中的一些对象（XML元素）被用来允许这些过程的发生。主要的对象是。

- 会期
- DATABLOCK
- FLASHDATA
- 物理学-数学
- 项目-信息

这些对象的组合允许对ECU进行有效编程。它们描述了物理和逻辑存储器的结构、要下载的数据、硬件和软件细节等。

#### odx诊断数据

ODX规范是一个非常有效的工具，用于对车辆的诊断数据进行建模。执行服务、接收和解释结果所需的所有信息都被建模了。这使得诊断应用程序很容易从ODX文件中抓取它所需要的诊断服务，将它们发送到ECU，并能够使用ODX文件中的转换方法来解释返回的结果。使用这种类型的模型，诊断应用程序可以非常通用，因为除了ODX文件，它不需要太多的数据来对ECU进行诊断。这允许通过简单地改变ODX文件对不同的ECU进行诊断。

#### 诊断服务

**ODX规范的核心部分是服务。服务是一个被发送到ECU的命令**



它要执行某种类型的动作。这个动作可能是让ECU打开一个执行器，送回一个单一的结果，送回定期的结果，等等。

一项服务包括一个请求和一个响应。请求是发送到ECU的消息，而响应是ECU在收到服务请求后发出的消息，如图所示。

6. 在ODX中，一个服务由DIAG-SERVICE对象建模。还有REQUEST和RESPONSE对象。

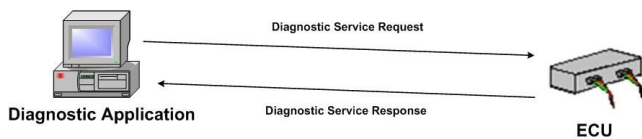


图6 执行诊断服务，使用请求和响应。

ODX数据模型（DIAG- LAYER-CONTAINER）的诊断类别中包含的大多数其他信息以某种方式与诊断服务相关。例如转换方法、数据类型等。

诊断层

ODX诊断模型中的所有诊断信息都包含在五个不同的诊断层中。这些层是：

- 协议
- 功能组
- 编码
- 基准变量
- ECU-VARIANT

所有这些层都很常见；它们都包含类似的数据，如诊断服务，但以这样的方式分组，以增加ODX数据模型的结构。

这些层的核心是BASE-VARIANT和ECU-VARIANT两个层。这是因为这些代表实际的ECU。当一个服务需要被执行时，（即向ECU发送请求），相关的ODX文件必须有这两个层中的一个为物理ECU定义。

图7显示了诊断ODX文件中这些诊断层的树状视图。这里的SHORT-NAME对象只是给诊断文件一个名称。每个诊断层都包含在一个包装器对象中。这只是在层的名称后面加上一个'S'，例如PROTOCOL层有一个叫做PROTOCOLS的包装对象。

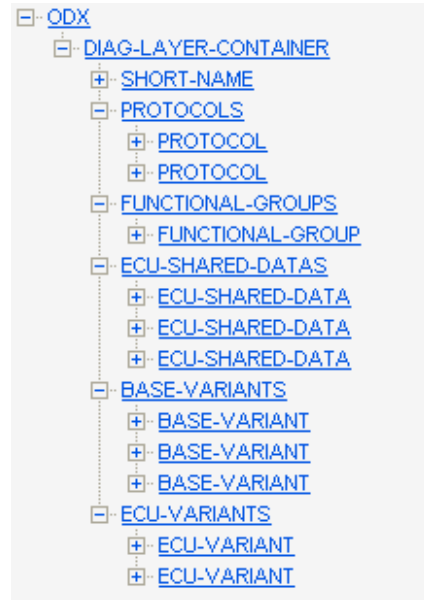


图7 一个标准ODX文件的树状视图

可以看出，在一个诊断ODX文件中可以有許多相同类型的诊断层。给出了每一种的简要描述。

### 协定书

这一层包含与ECU一起使用的诊断协议的信息，如ISO 15765（CAN上的诊断）。所包含的信息主要是该协议所包含的不同服务，以及相关的信息，如请求结构、响应结构、转换方法等。

### 功能组

该层持有与一组特定ECU有关的服务，如安全气囊ECU。每一组ECU都有自己的功能组。该层主要包含属于所选组的服务的信息。

### ECU-shared-data

这一层包含不同ECU之间可能共享的信息，如服务、转换方法等。因此，这一层的原因是通过将所有这些共享数据保存在一个层中，而不是在若干层中复制，以减少冗余。

### 基础变量

该层包含了关于ECU的信息。本层的每个实例代表一个不同的ECU。这一层持有所有只属于这个ECU的服务。这些通常会是标准化的供应商特定服务。

## ECU-variant

这一层代表了一个基本变体的后期版本。例如，如果一个基础变量（ECU）是1.0版本，那么ECU-VARIANT可能是1.1版本。从本质上讲，这只是一个已经存储在ODX文件中的ECU的后期版本。这一层可以被看作是BASE-VARIANT的一个孩子，继承了它的所有信息，比如它的服务，但也有只属于这个后来版本的信息，比如后来的服务是原始ECU版本（BASE-VARIANT）所没有的。ECU-VARIANT层所拥有的信息只是仅属于这个版本的新的诊断信息，而所有其他的诊断信息都来自于原始ECU版本。这样做也是为了减少冗余，从而减少在许多地方为每个ECU版本存储相同数据的开销。当然，ECU-VARIANT必须与现有的BASE-VARIANT相连。

### 诊断数据的打包/拆包和转换

在数据作为服务请求的参数被发送出去之前，必须对其进行改变，以便它能成功地被ECU发送和解释，反之亦然，当服务响应被送回应用程序时。这需要了解正在发送的服务和数据的特性，如返回值的转换。ODX已经允许对这种类型的信息进行建模。ODX中模拟这种情况的主要对象被称为DATA-OBJECT-PROP（DOP）。

这个对象描述了如何。

- 为一个请求或从一个响应中转换数据。
- 将数据打包到请求中/从响应中解包数据。
- 将返回的数据表示为一个单位，如mph、km/h等。

下面简要介绍一下这些情况。

### 数据的转换

作为服务请求的一部分发送的数据，或作为服务响应的一部分接收的数据，有时需要转换为编码值或物理值。物理值是一个数字，通常有一个单位，如128mph、5000rpm等。编码值是一个被ECU解释为代表该物理值的数字，例如，某个ECU可能将123ABC六位数解释为50mph。图8显示了这方面的一个例子。

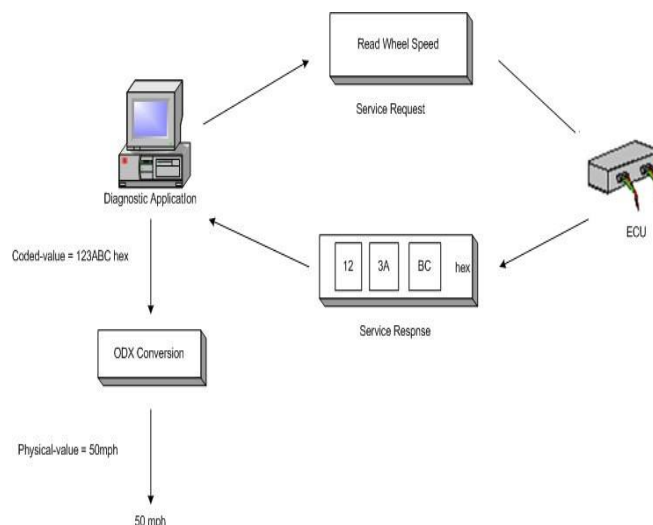


图8 检索数据的转换

在这里，一个读取轮速的服务请求从一个诊断应用程序发送到一个ECU。ECU返回一个服务响应，其编码值为123ABC十六进制。诊断应用程序使用适当的ODX文件将该编码值转换为物理值。这里123ABC十六进制被转换为50mph的轮速值。

ODX中用于进行这种转换的对象是COMPU-METHOD对象。这可以让返回的值以多种方式进行转换，通过使用线性函数、有理函数，甚至将返回的值转换为一个单词或短语。

例如，从ECU返回的值23hex可能被转换为“所有车门锁定”。

### 包装/拆包数据

一个服务请求/响应在低级别的ECU中被发送/接收，可以看作是必须被打包的数据，并沿着通信链路发送，如CAN。使用DIAG-CODED-TYPE ODX对象，诊断服务的所有数据都可以被打包，而应用程序不需要知道它是如何完成的。这也将以同样的方式反向工作，数据从连续的数据流中解包。

### 单位

为了帮助解释返回的数值，可以使用单位，如mph、rpm、amps等。单位允许在数值上附加一些更多的意义。单位也可以转换为同一组中的单位。例如，一个值在转换后可以返回为50，一个与之相关的单位可能告诉我们它是50mph，这个值现在可以转换为不同的单位，也许可以转换为80km/h。

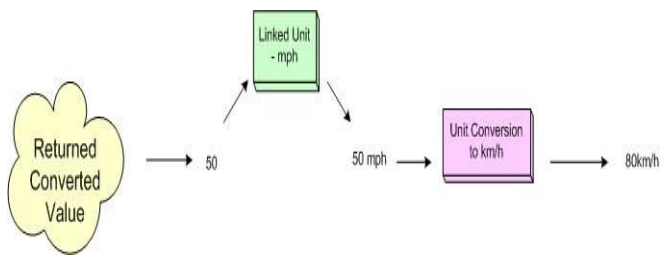


图9 单位转换

odx中的通信参数

通信参数是可以设置的参数，以控制应用程序和相关ECU或ECU之间的信息通信。这些通常是可改变的值，可由技术人员设置。通信参数是多种多样的，通常取决于正在使用的诊断协议，如KWP2000、CAN上的诊断等。下面的表2显示了两个典型的通信参数和它们的用途，指定用于CAN协议的诊断。

通信参数	描述
P2 CAN客户端	请求信息成功传输后，应用程序等待传入响应信息开始的时间。
请求数	一个应用程序在停止尝试通信之前向ECU提出的请求的数量

表2 通信参数的例子  
所有这些通信参数都被赋予一个值，以使通信成功。

通信参数可以存储在ODX文件中，通过将它们存储为一些COMPARAM对象来方便检索。COMPARAM是一个持有信息的对象，如名称、描述和特定通信参数的值。所有这些COMPARAMs本身被保存在一个COMPARAM-SPEC对象。一个COMPARAM-SPEC是一个顶层的ODX类别，用于将一些COMPARAMS组合在一起。常见的分组可能是 "CAN COMPARAMS 的诊断"，"KWP2000 COMPARAMS"，"最佳产量 COMPARAMS"，"我的个人 COMPARAMS"。

获得诊断层的通信参数

一个单独的ODX XML文件中的诊断层可以继承通信参数。这种继承允许诊断层能够检索适当的通信参数的值。

一个单独的ODX XML文件中的协议层继承了通信参数。该协议层中的所有服务（DIAG- SERVICES）现在都有通信参数，当它们被发送到ECU或从ECU发出时，它们可以使用。对于任何其他诊断层（即功能组、ECU-共享数据、基础变量或ECU-变量）来说，都要继承这些通信参数，它们必须继承协议层。图10显示了通信参数如何向下传播（继承）到不同的诊断层。

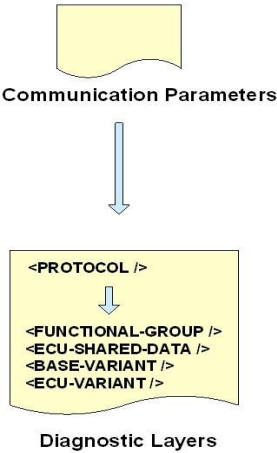


图10 通信参数的传播

一旦一个诊断层继承了通信参数，那么这些通信参数对属于该诊断层的所有服务都可用。

凌驾于通信参数之上

拥有一个包含一组一般通信参数的外部ODX XML文件是一个非常好的主意，因为它允许这个文件被许多诊断ODX XML文件中的许多诊断层所引用。这减少了不必要的冗余，如果每一个诊断ODX XML文件都必须复制相同的通信参数集（COMPARAMS），就会出现冗余。然而诊断层或服务可能需要改变一个或多个通信参数，与它所继承的上一层的参数不同。这并不意味着需要创建一个新的通信参数文件，而只是需要改变的值被覆盖（overwritten）。这是通过创建一个需要改变的通信参数的引用来完成的，通过其ID属性，为这个引用单独设置一个新的值。值得注意的是，该值只在它被引用的地方发生变化，但原来的通信参数值仍然是不变的。这是通过使用一个叫做COMPARAM-REF的对象实现的。

通信参数可以为诊断层或服务而被覆盖。如果它被诊断层重写，那么该诊断层中的所有服务现在都使用重写的通信参数值，而不是原始值。如果它是为一个服务重写的，那么只有这个服务将使用这个新的通信值。

下面的例子更好地说明了这一点。

图11显示了一个名为P2CANClient的通信参数，它将被存储在一个单独的ODX XML文件中，用于诊断服务。图12显示了一个BASE-VARIANT诊断层。为了这个例子的目的，我们认为图12中的基础变量诊断层继承了图11中的通信参数。在图11中可以看到P2CANCLIENT的值是23，但在图12中这个值被COMPARAM-REF覆盖（overwritten），并给出一个新值45。

```
<COMPARAM-SPEC ID="CS1">
  <SHORT-NAME>General_Communication_Parameters</SHORT-NAME>
  <COMPARAMS>
    <COMPARAM ID="A.P2CANCLIENT.ID">
      <SHORT-NAME>P2CANCLIENT</SHORT-NAME>
      <PHYSICAL-DEFAULT-VALUE>23</PHYSICAL-DEFAULT-VALUE>
    </COMPARAM>
  </COMPARAMS>
</COMPARAM-SPEC>
```

图11 原始通信参数

```
<BASE-VARIANT ID="BASE.VAR.1">
  <SHORT-NAME>BASE_VARIANT_1</SHORT-NAME>
  :
  :
  <COMPARAM-REFS>
    <COMPARAM-REF ID="A.P2CANCLIENT.ID">
      <VALUE>45</VALUE>
    </COMPARAM-REF>
  </COMPARAM-REFS>
  :
  :
</BASE-VARIANT>
```

图12 重写一个通信参数

## 使用odx的诊断应用程序的实施

开发了一个诊断应用程序，用于测试ODX标准。该应用程序用于对ECU进行诊断测试，被称为ACTS（汽车校准和测试系统）。它被设计成一个通用的诊断测试工具，对任何有相应ODX数据库的ECU进行诊断检查。该应用程序主要是针对开发人员和测试技术人员。该应用程序使用CAN协议（ISO 15765）上的诊断，然而主要的概念可以应用于所有的应用程序，无论使用什么协议。下面的图13显示了该应用如何连接到ODX数据库和ECU。

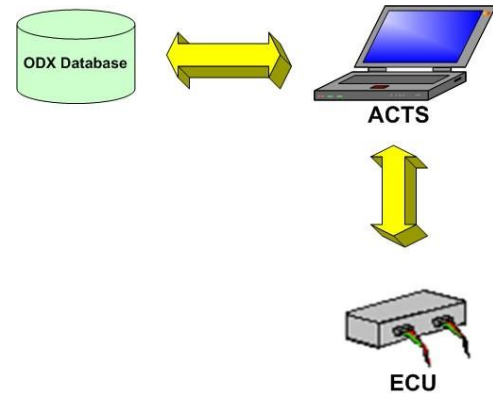


图13 ACTS应用程序与ODX文件和ECU进行通信

可以看出，这与符合ODX的工具的图基本相同（图2）。

## 臭氧数据和行为

该应用程序被构建为非常通用，因此它依赖于ODX文件的很多信息。所有的服务、计时参数、寻址信息、转换方法、语义数据、包装数据等，都直接取自ODX文件。这意味着不需要编程的数据，也不需要从多个来源引用数据。所有需要的数据都包含在ODX文件中。下面的图14显示了一些类型的数据在ODX文件和ACTS应用之间的流动。

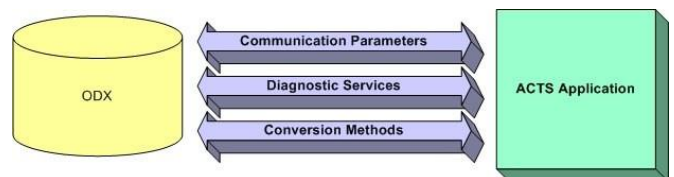


图14 从/向ODX文件检索和发送数据

双面箭头用于从ODX文件中获取执行服务、通信参数等的的数据，并发送至ODX文件，因为该应用程序也允许创建和编辑ODX文件。

## 创建一个odx文件

这个应用程序允许用户创建和编辑ODX文件。这是通过使用一些窗口来实现的，这些窗口向用户隐藏了ODX XML。用户只需要知道关于ODX数据模型的非常基本的知识，就能够创建一个完整的ODX数据库。

用户可以创建的两个核心ODX对象是。

- 诊断服务。
- 通信参数。

诊断服务

如前所述，诊断服务是ODX数据模型的核心组成部分。用户能够创建一个由协议指定的标准诊断服务或创建一个独特的用户定义服务。创建服务包括创建一个服务请求和一个服务响应。请求和响应都可以有任何数量的参数，这些参数可以由用户设置。用户也可以从CAN协议的诊断中挑选他们可能需要的服务，而不需要创建它们。

下面的图15显示了ACTS应用程序中服务创建窗口的截图。

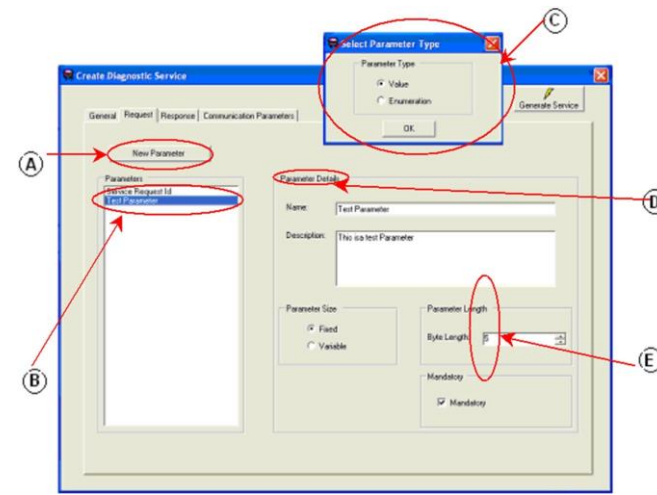


图15 创建一个服务请求。

对这个窗口的标示部分进行了简要说明。

- A. 新参数。这将在请求中添加一个参数。
- B. 参数列表。这显示了所有已经添加到请求中的参数。
- C. 这允许用户选择要添加到请求中的参数类型，这在用户点击 "新参数" 时显示。
- D. 参数详情。这显示了从参数列表中选择的参数的所有信息。
- E. 值型参数的一个例子。

通信参数

用户可以设置的通信参数包括寻址信息和定时信息。计时信息可以为诊断层或服务而设置。允许用户设置通信参数意味着他们可以测试许多不同的定时值，并挑选最适用于该ECU的定时值。

图16显示了ACTS应用程序中通信参数选项卡的屏幕截图。

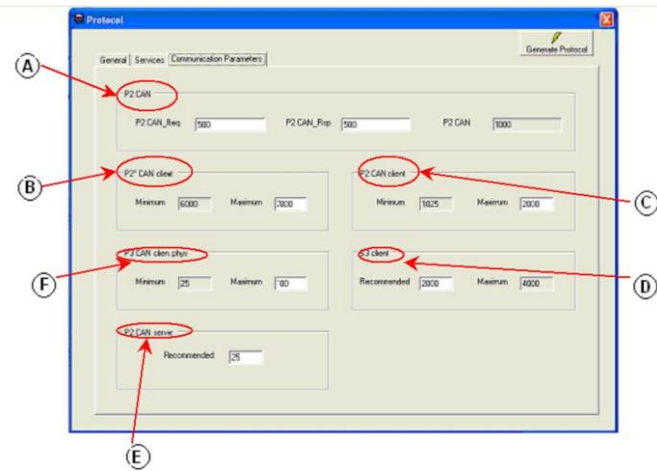


图16 改变通信参数。

这个窗口被设计成允许用户改变CAN通信参数的诊断方法。

表3中显示了突出的领域。

突出显示的领域	通信参数	描述
A	P2CAN	保存通信过程中请求和响应所需的时间。
B	P2*CAN客户端	P2CAN客户端的增强版。如果ECU要求更多的时间来响应，则使用。
C	P2CAN客户端	客户端在成功传输请求信息后开始接收响应信息的超时。
D	S3客户端	这个定时器在每次定时器超时后，都会发送一个测试者在场的请求，从而保持非默认会话的活动。
E	P2CAN服务器	对服务器（ECU）的性能要求。
F	P3CAN客户端物理	后客户端等待的最小时间。



		成功传送物理寻址的请求信息，并且在传送下一个物理寻址的请求信息之前不需要回应。
--	--	---

表3 为ACTS提供的通信参数。设置和执行诊断服务

该应用程序让用户执行一个诊断服务。用户首先选择相关的ODX文件。然后用户会看到一些不同的ECU，这些ECU在ODX文件中都有记录。然后可以选择一个ECU。然后会显示所选ECU的服务列表。ECU。然后可以选择一项服务。然后会显示所选服务的参数列表。然后可以填写每个参数。一旦所有的参数被填满，服务就可以被执行。

刚才描述的过程都是通过从ODX文件中提取信息完成的。

下图17显示了ACTS应用程序中的服务执行窗口。

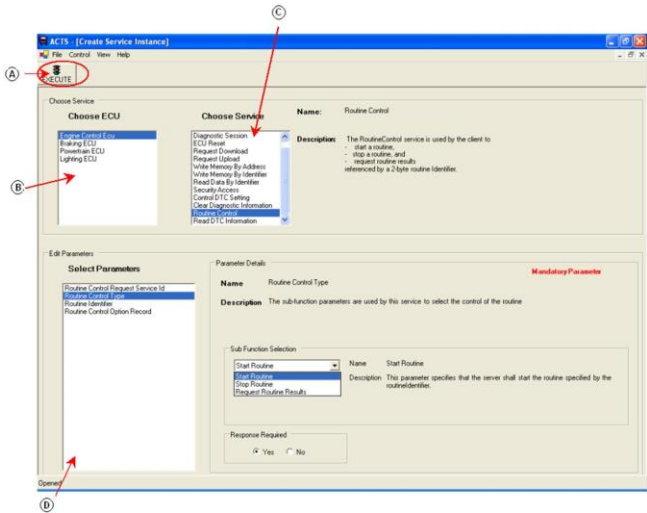


图17 ACTS执行窗口

图17显示了执行窗口，包含了一个给定ODX文件的所有ECU、服务和参数。验证确保了一个糟糕的服务请求不能被发送到ECU。

突出的领域是。

- A. 执行按钮。
- B. ECU选择列表。
- C. 服务选择列表。

接收一个诊断服务响应

一旦发送了服务请求，那么只需要几秒钟就可以返回响应。服务响应最初只是一个数据流。然而，相关的服务响应信息是从ODX文件中提取的，因此它被用来解释返回的原始数据。然后，诊断信息会呈现给用户。

工具验证

ACTS应用被用来测试一个生产型ECU的诊断服务。它能够与ECU通信，执行服务并接收准确的结果。这表明ODX数据模型可用于生产ECU的诊断监测。

D. 参数选择列表。

## 关于接近odx规范的准则

ODX规范是一个很大的文件，有440多页，当你开始阅读它时，会有一点混乱。造成这种混乱的部分原因是，它在早期引入了一些术语和概念，而这些术语和概念在规范的更远处才被解释。正是由于这个原因，本节被纳入本文。

下面列出的要点应该能使阅读该规范更容易。这些要点是为一个需要深入了解ODX的人编制的，因此可能并不适用于所有人。

- 尽早学习如何解释UML（统一建模语言）图。它们在规范中被广泛使用，清楚地了解这些将对你有很大帮助。该规范在一开始就给出了一个快速有效的教程，不应该被忽视。没有必要深入学习UML。
- 学习XML。这就像UML图一样在整个规范中使用，因此应该学习。它也是ODX模型所使用的格式，因此要深入了解ODX，了解它是必不可少的。它也使UML图更容易理解，因为图可以与规范后面的ODX例子交叉引用，以便更清楚地理解。下面的教程简单快捷，应该足够了，[www.w3schools.com/XML/default.asp](http://www.w3schools.com/XML/default.asp)。
- 如前所述，这个ODX规范是围绕诊断服务的。因此，需要对服务有一个清晰的理解。强烈建议阅读诊断协议，如ISO 15765规范。这样做的额外好处是，在规范的后面有这些协议的例子

在ODX格式中，这将澄清一些可能出现的问题。

- 在ODX规范中大量使用了链接和引用。然而，在规范的更远处，才对这些进行了适当的解释。这个领域绝对应该在早期就被关注。跳过前面的内容，阅读规范中的这些地方，会使理解规范更容易。
- 在规范的后面有一个非常有用的部分，叫做 "XML 中的数据模型实现"。这是一个非常好的部分，可以澄清ODX规范中的许多领域。应该经常检查这一节，看看它是否能提供帮助。
- 如果你有一定的计算机编程背景，可能值得写一个程序，将规范后面的ODX例子表现为一棵分支节点的树。一个ODX文件的树状视图将使你更清楚地看到ODX文件的结构和嵌套对象。这并不像它看起来那么困难。互联网上有许多免费的树状视图的例子，可以非常容易地复制和粘贴。图7显示了不同诊断层的树状视图。
- 多读一遍规范。第二次阅读规范会使其更加清晰，并澄清第一次阅读时可能出现的一些不确定性。

如果你遵循这些准则中的一个或全部，规范应该更容易阅读和理解。

## 使用odx遇到的问题

ODX是一个代表ECU诊断数据的强大模型。然而，尽管ODX在整体上是一个非常好的模型，它也有一些问题。下面列出了其中的一些问题。应该指出的是，这些问题是在ASAM MCD-2D (ODX) 2.0版中遇到的。

- ODX规范（而不是数据模型）的一个大问题是，它写得很糟糕。该规范在早期就提出了一些术语和概念，而这些术语和概念直到后面才被解释。这使得其他领域看起来比实际情况更难。
- ODX规范的另一个问题是它后面给出的例子。这些例子有很多错误，一般在两个方面不正确。首先，它们对ODX数据模型（ODX XML模式）是无效的。其次，它们对形成XML文档的正常规则无效。这意味着，在理解这些例子之前，必须对其进行修正。

- XML Schema可以被看作是ODX文件的模板，它代表了ODX的数据模型。然而，XML Schema与规范中给出的ODX数据模型并不匹配。这意味着ODX的XML SCHEMA也必须被改变。
- 如前所述，ODX规范使诊断工具更加通用，因为不需要将静态数据硬编码到应用程序中。然而，这只是在一定程度上的通用。ODX数据模型的表现力不足以使诊断应用程序非常通用。例如，某些服务有强制性和非强制性的参数。为了对这些参数进行错误检查，以确保它们在执行服务前被正确填写，诊断应用程序需要知道哪些是强制性的，哪些不是。ODX并没有为此提供便利。然而，这个问题可以通过为每个参数设置一个IS-MANDATORY属性而轻松解决。
- 在ODX规范中没有提到如何存储ECU地址的问题。我觉得这应该被显示出来，因为这种模糊性可能导致产生不同的ODX文件，以不同方式存储地址。

ODX规范的表达能力不足以让诊断应用完全通用。这是ODX数据模型的一个短板。但除了这一点，ODX数据模型证明了它非常有能力完成它的使命。还必须强调的是，ODX规范是相对较新的，因此在以后的版本中还有改进的余地。

## 结论

ODX是一个非常有表现力和强大的数据模型。它有很多优点，使它成为一个非常好的规范。它允许诊断应用更加通用。它试图为诊断数据的记录方式引入更多秩序。它是以一种简单而富有表现力的格式编写的，即XML。然而，它也有不足之处，比如它需要比现在更加通用。

ODX作为一个整体是一个良好的数据模型。它被成功地用于ACTS诊断应用程序，对一个生产型ECU进行诊断检查。这是很容易做到的，因此显示了ODX的表达能力。

## 参考文献

1. (2004) ASAM MCD-2D (ODX) , 2.0版。
2. XML Tutorial. [在线]。  
Available: <http://www.w3schools.com/xml/default.asp>  
[2006, January 9]



## 联系我们

### 凯文-穆勒里理学学士

凯文以班级第一名的成绩毕业于沃特福德理工学院的应用计算机专业。他目前在Brendan Jackman的指导下，正在攻读汽车安全关键系统的研究生。他在ODX标准方面的经验来自于与David Boyd和Gareth Leppla在一个诊断项目组的工作。

电子邮件：kmullery@wit.ie

### Gareth Leppla B.Sc

加雷思在沃特福德技术学院学习了五年。在这段时间里，他获得了工业计算机的证书，应用计算机的学位，并被授予年度科学人物奖。他在与David Boyd和Kevin Mullery组成的诊断项目组中获得了ODX标准的第一手经验。

### David Boyd B.Sc

大卫-博伊德最近从沃特福德技术学院毕业，获得了应用计算机的学位。在那里学习期间，他还获得了工业计算机的证书。戴维通过与各种诊断协议的合作获得了在汽车诊断方面的工作经验。他与Gareth Leppla和Kevin Mullery在ODX标准的工作中应用了这些知识。

### 布伦丹-杰克曼 理学学士，技术硕士。

布兰登是沃特福德理工学院汽车控制小组的创始人和领导者，他指导研究生从事汽车软件开发、诊断和车辆网络研究。布伦丹还为沃特福德技术学院应用计算机学士学位的本科生讲授汽车软件开发课程。布伦丹在实施实时控制系统方面有丰富的经验，他曾在爱尔兰的数字设备公司和荷兰的Logica BV工作。

电子邮件：bjackman@wit.ie

## 定义、首字母缩写词、缩略语

**ACTS:**汽车校准和测试系统。

**ASAM:** 自动化和测量系统标准化协会。

**CAN :** 控制器区域网络。

**ECU :** 电子控制单元。

**ISO :** 国际标准化组织。

**LIN :** 本地互连网。

**ODX :** 开放式诊断数据交换。

**UML :** 统一建模语言。

**XML :** 可扩展标记语言（eXtensible Markup Language