FDT/DTM, and Enhanced EDDL, what's best for the user.

Introduction

A very wise man once said that the way to succeed in business is to solve customer problems. There is a great deal of conversation on the topics of EDDL and FDT/DTM right now, and they tend to center on technology. But the true conversation should center on the best way to solve the customer problem. The first step is to define the customer problem. The second step is to determine what technology best solves what problem.

The basic customer problem is how to run a plant that is safe, clean, and profitable. Transmitters, valves, control hosts, and asset management systems are important tools for running the safe, clean, profitable plant. FDT/DTM and EDDL are both technology enablers for the field devices and hosts.

The role of a technology enabler is to be open, interoperable, provide access to all functionality in all environments, and be as painless as possible to implement, use, and maintain. The expectation should be that enabling technologies are there, and they just work.

Device functionality must be properly configured, openly communicated to control and asset hosts, openly available to business systems, routed to the appropriate users in the appropriate ways, and displayed and retained as needed for current and future use.

We've established some background, now it's time to look at the appropriate role(s) of EDDL and FDT/DTM.

Configuration / Maintenance

Device functionality is invoked using Electronic Device Description Language, EDDL or DTM's. The DD or DTM tells the host what functionality the device has, and how the functionality is invoked. It also tells the host how to do common maintenance functions such as calibration, trims, tests, and other device activities.

Commissioning Foundation fieldbus devices

Commissioning Foundation fieldbus devices on a control host requires DD's. Most control hosts have a set list of applications that are considered safe to install on the host engineering or operator station. Each DTM is an application, and the testing required to ensure hundreds, or potentially thousands of DTM's are compatible with a control host user interface is not practical. EDD's are files, not application programs. Therefore there is no program installation risk loading EDD's on a control host.

EDD's are also frequently required to connect a device to an asset management application, and conduct basic parameterization before the device can be accessed using

FDT/DTM. If FDT/DTM is used, a complete and up to date set of EDD's is also usually required for the control host, and for many asset management hosts.

Communication

Communication with devices is accomplished via the communication stack. The communication stack for the vast majority of field devices may be HART, Foundation fieldbus, or Profibus PA. The communications stack is functionally a path for data. As indicated above, the device functionality is defined by EDDL or DTM's.

Data availability

Data availability is a big issue. Data should be available to any host or appropriate other device or application. Since EDDL is used to invoke device functionality and define that functionality for the host, EDDL is the path for data availability that originates from a device, or is going to a device. The OPC Foundation support for the enhanced EDDL will broaden the use of EDDL for applications such as ERP, maintenance management, and other applications.

Data display

Data display is an area where either EDDL or a DTM can provide a solution. EDDL is supported in the host by DD services. DTM is supported in the host by a frame or FDT. For many applications and hosts either EDDL or DTM can be used for data display. For hosts that are not based on a windows operating system, EDDL will be used as DTM requires a windows operating system. EDDL has defined display objects such as charts, graphs, etc. DTM is more of a free form environment using a variety of programming languages.

Note: EDDL has been available for many years. Over time it became apparent that extensions would be beneficial for data display and persistent data. Extensions have been defined. This is frequently referred to as *enhanced EDDL*, or *EEDDL*. Data display and data retention and archiving assume the enhancements are used.

Persistent Data

Data retention, archiving, and recall are supported by both FDT/DTM and EDDL. Again, EDDL has defined capabilities while FDT/DTM is more freeform.

What are the user choices?

DTM and EDDL are both used for display and archiving. Therefore the user choices are EDDL or a combination of EDDL and DTM. The question is where a single solution (EDDL) is appropriate, and where a combined (EDDL and DTM) solution is appropriate.

There are some easy decisions. EDDL is the solution wherever a windows operating system is not available. This may include currently available handheld devices, and anywhere a UNIX or other operating system is used.

Where a windows operating system is available, there is an option of either a single or combined solution. The best solution involves a tradeoff between functionality and complexity / support, as indicated in the introduction.

Functionality needed to support a device is dependent on the complexity and functionality of the device itself. For example, a full valve signature diagnostic with archived information, thousands of data points and the need to manipulate data for analysis and display may not be possible or practical in EDDL. A program such as a DTM or "snap-on" application may be needed. Common device configuration, parameter display, charts, graphs, and images, are all supported in EDDL. In this case the entire application can be implemented using either EDDL or DTM.

The clear domains of one technology or the other are:

EDDL is required for any host that is not based on a windows operating system supported by DTM. This may represent a significant minority of applications.

EDDL is required if the control host uses Foundation fieldbus. It may also be required for the asset management host to perform initial setup of the Foundation fieldbus device.

DTM is required for any application sufficiently complex as to not be supportable by EDDL, and where a "snap-on" type of application is unavailable or not desired. This may be represented by full valve signature diagnostics where no snap-on for the host environment is available.

Is there a best choice?

Most devices fall in between these two domains. The host uses a windows operating system, and the device application is supportable using EDDL. Is there a better technology or is it a coin toss?

In my opinion, there is a better technical implementation based primarily on ease of implementation and support. That solution is to use EDDL for all devices where EDDL is technically capable of delivering complete device functionality, and to use a DTM or a snap-on application to handle only the exceptions. I make this recommendation because it is simpler to implement a single solution than a combined solution. EDDL is a single solution that will work for the vast majority (95%) of HART, Foundation fieldbus, and Profibus PA devices.

There is also lower risk in certain situations with EDDL vs. DTM. The combined solution requires both an EDD and a DTM. Both are required for proper display and persistent data. If either is not present, or if either is not up to date or supported on the current platform, the user cannot invoke the functionality. A mental exercise may be beneficial. It's 2:00 AM on a Sunday (this is the chosen time for all major plant problems). A critical device has failed. What is the fastest way to get a new device in service? I can rely on EDDL, or a combination of EDDL and DTM's. By definition, simpler is more reliable.

Using DTM, a vendor needs to provide a DD, and one or more DTM's depending on the operating system support required. Given that there may be thousands of devices in a plant, and dozens of different device types, this becomes more complex, requiring modification to typical plant procedures. Add to this the potential need to have different DTM's for different versions of windows, and the process becomes much more complex and prone to mistakes.

Microsoft operating system migration is another potential issue. Windows XP has been given a new breath of life with the release of service pack 3. Microsoft seems intent, however, to migrate the world to Vista. Vista will almost certainly create some problems with DTM's. Users may want to delay implementing FDT/DTM until all DTM's you may use have been tested and are proven compatible with Vista. If a user decides to implement FDT/DTM, and will be moving to Vista over time, duplicate sets of DTM's may be needed. Plant procedures should be modified to accommodate this.

Overcoming the difficulty

There are ways to overcome the difficulty with various versions of DMT's. One is to exercise vigorous control over the number of different versions of windows within a facility. This approach is technically good business practices, but may involve significant work at upgrade time, and the cost of widespread hardware upgrades and compressed upgrade schedules to keep consistent. Another solution is to keep a master library with each DTM version maintained in a specific place, and with training on what versions to use with what hosts.

The best way to overcome this difficulty is to as much as possible avoid it entirely. Using EDDL wherever it provides the needed functionality, and using DMT's only to handle exceptions reduces both the number of DMT's, and the number of target hosts for the DMT's.

The final recommendation

The final recommendation is to use EDDL as the required standard since each device must have a DD. Allow the use of DTM's on an exception basis where the functionality is required, and EDDL cannot provide it. Make sure that all the functionality to replace a failed device, or place a new device in service is available in EDDL. This will simplify implementation and maintenance, mitigate operating system migration issues, and provide a lower risk more error free working environment.

