Implementing an EtherNet/IP Device DTM

Rick Blair, Schneider Electric

Technical Track
Overview

Introduction to FDT/DTM

Implementation Methods
- From device description files
- From existing application
- From scratch

Style Guide

CIP Annex

Certification

FDT/DTM Evolution
Concept of FDT

Analogy with PC Printer Interface

- **Frame Application**
  - Printer driver is used in same way in different PC software (e.g. office software)

- **Device DTM**
  - Printer driver contains configuration, diagnostics, and other functionalities
Frame Application is like PC software. It provides:

- Common Environment
- Network Configuration
- Navigation
- User Management
- Database Storage
The DTM

...is the driver representing actual device
...is provided by device manufacturer
...is installed on PC
...can be used in any Frame Application
...provides graphical user interface
...allows access to device parameters
How to develop a DTM

Three approaches

- Use of Device Description
  - Dynamic DD interpretation
  - Using DD for DTM creation
- Expanding Existing Standalone Application
- Developing DTM from Scratch

Use of toolkits highly recommended

- Limit DTM spec misinterpretation
- Reduce interoperability issues
Use of Device Description

Best suited for simple devices

- No parameter dependencies
- Adjustable devices with fixed hardware and software

If device description (DD) exists, it is probably easiest way to create DTM
Dynamic DD Interpretation

Device DTM dynamically interprets DD (e.g. EDS) files

- Device DTMs added to a catalog by importing DD files
- No need to recompile source code to create new DTM
- Allows distribution of only DD files to customers

Very generic and universal graphical representation

- Not adjusted for specific device
- Simple table view of parameters
- No parameter dependencies
Using DD to Create DTM

**Toolkit to generate device DTMs from DD file**

- Typically integrated into development tools
- Creates source code for fully functional DTM
- Must be compiled to create device DTM.
- All FDT interfaces, parameter model of device and graphical user interface are created
- Source code can be modified to add custom capabilities, e.g. parameter dependencies
Expand Existing Application

Create Proxy DTM, equip stand-alone tools with FDT-interfaces

- Runs outside FDT frame application in its own process and window
- No need to develop complete DTM with ActiveX user interface.
- A way to migrate towards FDT while preserving user familiarity
- Beneficial if need to maintain both stand-alone tool and DTM

Step-wise approach

- Initial simple user interface integrated into proxy DTM
- Remaining functionality provided by stand-alone tool
- Migrate stand-alone functionality into DTM over time
- All functionality migrated into DTM as final goal
Develop DTM from Scratch

Most time consuming

Most flexible

Use of toolkit highly recommended

Necessary if no current tool exists for device, such as new development or

Current tool interface does not lend itself to DTM model
What is a Protocol Annex?

- Additional specification to integrate a fieldbus into FDT
- Work done in cooperation with respective fieldbus organization
- Selection of services to be used with FDT
- Description of services in XML schema
- Prototyping to verify specification
- Scripts for DTMinspector to test and certify DTMs
CIP Protocol Annex Problems

- Problem 1: Config 1 / Config 2
- Problem 2: Unique Connection ID
- Problem 3: Connections and Internal Modularity
- Problem 4: Process Channel Handling
- Problem 5: Network Scan, No Regular Expression
- Problem 6: Network Scan, No Extended Address
- Problem 7: Module Identification in Modular Devices
- Problem 8: Process Channel with Data Type Array
- Problem 9: Urgent Connection Parameters Priority
Problem 1: Config 1 / Config 2

- During CIP Forward_Open service, optional data segment can be appended to the path attribute
- Config1 and Config2 elements in FDTCIPDTMParameterSchema used to transfer this information
- Definition of element content is unclear

Proposed Solution

- Concatenate configuration buffers config1 and config2 to <cip:ePath> attribute in <CIPConnection> element, if required.
Problem 2: Unique Connection ID

- Devices can export more than one connection
- Master DTM cannot tell if new list of connection is the same as the previous one if Device DTM exports more than one connection and performs IDtmEvents::OnParameterChanged()

Proposed Solution

- `<fdt:nodeId>` attribute used to provide unique identifier for connection
Problem 3: Connections and Internal Modularity

 Unable to directly reference connection in module description

Proposed Solution

 Must uniquely reference connections within a device

 <Module> element contains <ExportedVariables> element used to reference connection described in FDTCIPDTMParameterSchema.
Problem 4: Process Channel Handling

- Method to handle process channels is unclear
- Performance issues if every I/O modeled as a process channel

Proposed Solution

- DTM exposes only one process channel per assembly
  - Contain assembly member references
  - assembly member references point to assembly member definition
Problem 5: Network Scan, No Regular Expression

- Regular expressions not supported in current schema

Proposed Solution

- Define new schema definition file for DTMCIPDataTypeSchema.xml containing `<cipident:RegExpr>` element
Problem 6: Network Scan, No Extended Address Identifier in XSL

- Network scan result XSL transformation does not handle extended identifier for `<IdAddress>` element
- Needed for EtherNet/IP address in format 192.168.1.160

Problem Solution

- Current FDT Container applications don’t check validity of scan result documents
- Use short identifier for CIPNodeID in the scan result
- Allows use of current XSL transformation defined in released CIP Annex
Problem 7: Module Identification in Modular Devices

- Module identification within modular device not possible.
- `<CIPDeviceIdentity>` element related to one CIP node, i.e. one complete CIP device.

Proposed Solution

- `<InternalTopology>` element describes modularity of device
- Each module described with a `<Module>` element, containing a `<ExportedVariables>` element
  - Vendor Code
  - Product Type
  - Product Code
  - Major Revision
  - Minor Revision
Problem 8: Process Channel with Data Type Array

- Process channel of data type “array” cannot be defined
- Array length cannot be defined

Proposed Solution

- fdt:BitLength attribute indicates bit size of data type
- If bit length greater than bit length of data type then default item is interpreted as array
Problem 9: Urgent ConnectionParameters Priority

- “urgent” priority for connection parameters added after annex

Proposed Solution

- Add parameter to annex
Rules and guidelines describing DTM graphical user interface behavior

- layout instructions
- representation of parameters
- menu structures
Style Guide GUI

**Identification area**
- Device photograph and vendor logo
- Device type and name

**Menu bar and toolbar**
- Directly under identification area

**Navigation area (optional)**
- Organizes elaborate parameter models in functional groups

**Application area**
- Parameter name, value, unit and status

**Action area**
- Buttons like OK, Cancel and Apply

**Status bar**
- Indicates whether data is changed online or offline
- Displays current user role
Style Guide Benefits

Guideline and rules are defined for:

- Uniform user guidance regardless of device or DTM manufacturer or communication protocol.
- Uniform behavior of DTM
- Clear identification of DTM and the assigned device
- Inform users if GUI input affects device or offline configuration.
- Consistent user input validation
- Uniform installation/un-installation procedure
Certification

Benefits
Test Objectives
Test Sites
Certification Process
Arbitration
Certification Benefits

Certified DTM verified to specification
Listed on FDT website
Certification tool dtmINSPECTOR

• Check conformance during development
  • data set access in multi-user environments
  • bad cases, such as loss of communication.
Certification Test Objectives

Checks DTM conformance to FDT specification

- Begin with correct installation
- End in a un-installation test.
- Mandatory interfaces are always tested
- Optional interfaces tested only if implemented
- DTM state machine and return values of defined interfaces
- Check for correct behavior
- Stimulate with errors or go against specification.

Does not cover device specific functional tests
Certification Test Sites

FDT Group audits and accredits test sites

- ... to have technical competence
- ... be able to handle the test tool
- ... to have adequate facilities at their premises
- ... be impartial and independent
- ... to cooperate with clients and the FDT Group

Test site list on FDT Group website
Certification Process

- Setup and device sent to test site
- Perform official tests on a clean system with other certified DTMs
- Report provided to DTM vendor
- Vendor applies for formal certificate
- FDT Certification Office checks for positive report
- DTM added to list of certified DTMs
- Formal certificate issued
- Certificate valid for three years
Certification Arbitration

In case of a negative report

- DTM vendor can challenge correctness of tests
- Vendor contacts FDT Group
- FDT Group inspects tests
- FDT Group members provide arbitration, which either invalidates vendor’s objection or specific test case
New annex

- Field bus independent integration of process channels
- Defines a process image description which is independent of field bus

PLC programming tool able to integrate signals independent of field bus using FDT technology
FDT Evolution

FDT2 project main objectives:

▸ Technical transition from COM/ActiveX® to .NET
▸ Replacement of XML documents for data exchange between FDT components by .NET data types
▸ Performance increase and simplification of implementation in particular on DTM side
▸ Backward compatibility (support for V1.2.x DTMs in V2 Frame Applications)
▸ Reduced number of interfaces

New specification draft currently in review within the FDT Group

Final draft target April 2011