Vendor-Independent modeling and exchange of Fieldbus Topologies with AutomationML

2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation

Markus Rentschler, Balluff GmbH, Germany
Prof. Dr. Rainer Drath, University of Applied Sciences Pforzheim, Germany

Track 9.3: Complex Automation Systems
07.09.2018
Outline

1. Problem Statement
   - Device Engineering & Lifecycle
   - Lack of instance data handling
   - Lack of modelling capabilities
   - The standardization deadlock

2. Proposed Solution
   - Short Introduction to AutomationML (AML)
   - AML-Device Descriptions (AML-DDs)
   - AML-DD File Packages
   - Topology Descriptions

3. Conclusion
The **Device description (DD)** models a static device type with its properties and parameters:

- e.g. ESD (Ethernet/IP), ESI (EtherCAT), GSD (Profinet), XDD (Powerlink), CSP+ (CC-Link), IODD (IO-Link)

DD’s allow representation of the device properties in a proprietary engineering tool.
Lack of instance data handling over the device lifecycle

Vendors supply devices with some static model data (DD).

The static data gets supplemented with instance related data (configuration, location, etc.)

Does not really work today

Source: ZVEI
Lack of modelling capabilities for mixed technology systems

"Classical-DDs" contain primary (fieldbus) interfaces, but no:

- secondary interfaces
- power interfaces
- interfaces to other technologies
- advanced model data

Source: Balluff
Lack of Modelling Capabilities for Devices

Additional model data is often available in separate files, in distributed locations and different versions.
Goal: DD with full model for the Digital Twin, "Asset Manifest" for I4.0-Component
Semantic world models do not work: The standardization deadlock #1

Standardization requires feedback from the users to get mature.

Tool vendors wait for a mature standard.

Standardization

Practical Use

Standardization Deadlock #1
Semantic world models do not work: The standardization deadlock #2

...it freezes immediately and loses its innovation power

Standardization Deadlock #2

When a small semantic standard reaches maturity ...
**Short Introduction to AML**

**Topology, Fieldbus Communication, Device Description, …**

*Source: AutomationML e.V.*
Initial Idea

Let’s extend the DD standard!

- requires a new version of the legacy DD standard
- requires extensive standardization cycles within the fieldbus communities
- breaks compatibility with the existing tool landscape

How can we extend a DD standard without changing it?
New Idea: Extend the classical DDs with AML

Embedding the existing DD standard into the AutomationML standard

- The legacy DD remains unchanged
- The existing tool landscape can remain unchanged.
- Extended features can be added stepwise into the tool landscape.
- The DD standard elegantly inherits all object oriented capabilities of AutomationML.
AML-DD: How it works

AutomationML Instance Hierarchy

IOLinkDeviceConfiguration
- BSP0086-IOD1.1_instance { Class [BSP0086-IOD1.1] Role IOLinkPhysicalDevice }
- IOLinkDescriptionDocument { Class Role IOLinkDeviceDescriptionFile }
  - IOLinkDescriptionDocument-Interfaces
    - DocumentLink { Class ExternalDataReference }

AutomationML Library

IOLinkDevicesLib
- BSP0086 { Class }
  - BSP0086-IOD1.0.1 { Class BSP0086 }
    - IOLinkDescriptionDocument { Class Role IOLinkDeviceDescriptionFile }
      - IOLinkDescriptionDocument-Interfaces
        - DocumentLink { Class ExternalDataReference }
  - IOLinkRoleClassLib/IOLinkPhysicalDevice
- BSP0086-IOD1.1 { Class BSP0086 }
  - IOLinkDescriptionDocument { Class Role IOLinkDeviceDescriptionFile }
    - IOLinkDescriptionDocument-Interfaces
      - DocumentLink { Class ExternalDataReference }
  - IOLinkRoleClassLib/IOLinkPhysicalDevice

Additional instance information

Additional type information

„Classical DD“

Unchanged legacy DD file
AML-DD File Package

- **AMLX or ZIP File Container**
- **AML Root File with additional model information**
- **Legacy Description File (here: GSD)**
- **Device Picture**
- **i.e. Additional Logo**

AML refers to GSDML Manufacturer_Name.bmp
Summary on AML-DDs

The AML-DD concept is generic

- thus a truly next generation device description
- able to model type and instance information
- starting point for exchangable 360° models of devices with documents, icons, 3D models, function blocks, faceplates etc.
- Provides a migration path!
Description of topology information

1. A given topology of devices exists.
2. Each device comes with an AML-DD that describes the physical (and logical) device interfaces
3. CAEX provides the “InternalLink” element to connect interfaces, even between different AML files.
4. The links between the interfaces in the AML-DDs are described in an AML-Topology file.
5. The AML-DDs are referenced from the AML-Topology file via the “ExternalInterface”-Element.

©2018 M. Rentschler, R.Drath “Vendor-Independent modeling and exchange of Fieldbus Topologies with AutomationML”
Description of mixed technology systems
Conclusion

- Engineering of Automation Systems is traditionally based on proprietary Device Description Languages and associated software tools, usually defined by fieldbus organizations.
- These systems lack of many capabilities and extending them is difficult, because the fieldbus organizations are very restrictive.
- AutomationML based on CAEX is a simple but powerful means to “extend” and finally even replace these legacy Device Descriptions without changing them.
- The way to get acceptance for such a method within the industry is standardization by an accepted body within the automation community and a migration path for the legacy systems because of investment protection!
- In a cooperation between the IO-Link-Consortium and the AutomationML association currently the first standard is defined for the IO-Link infrastructure.
- Because the underlying concept is generic, it can easily be adapted by any fieldbus organization.

DD standards can be extended without changing them! 😊
Thank you!

“Vendor-Independent modeling and exchange of Fieldbus Topologies with AutomationML”

Questions?