



<AutomationML/>

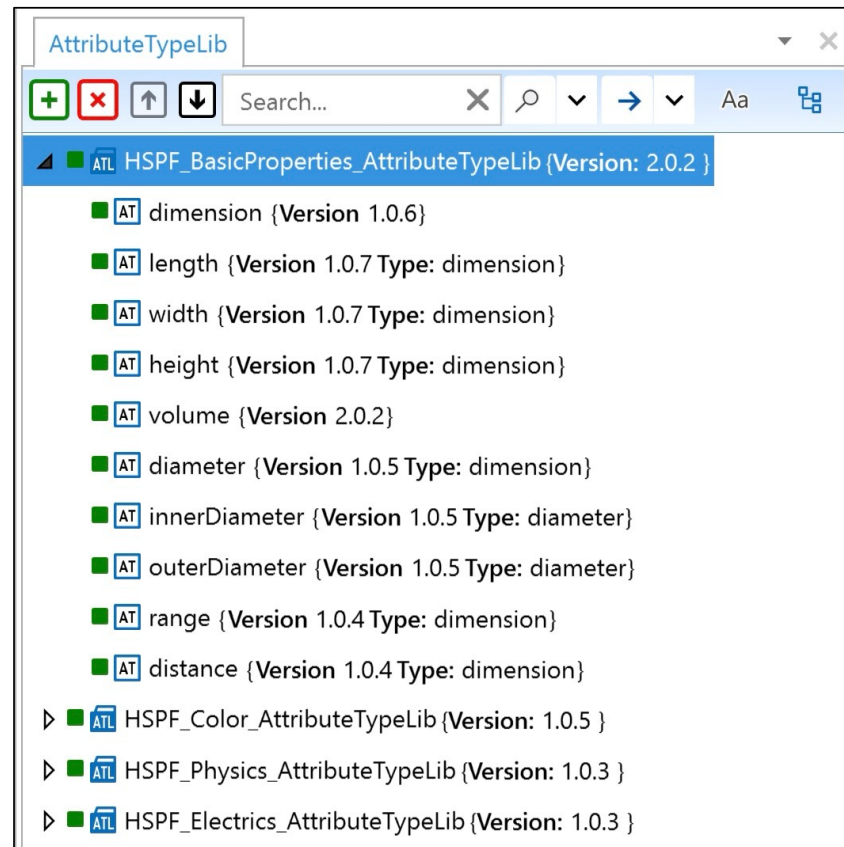
The Glue for Seamless
Automation Engineering

AML core property libraries

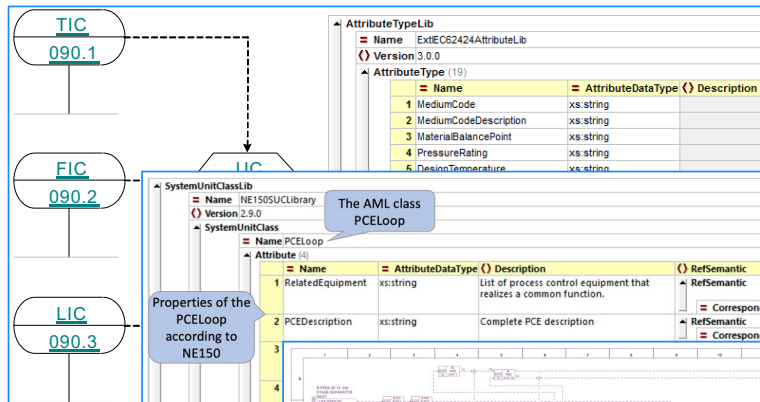
Rainer Drath

Pforzheim University of Applied Sciences

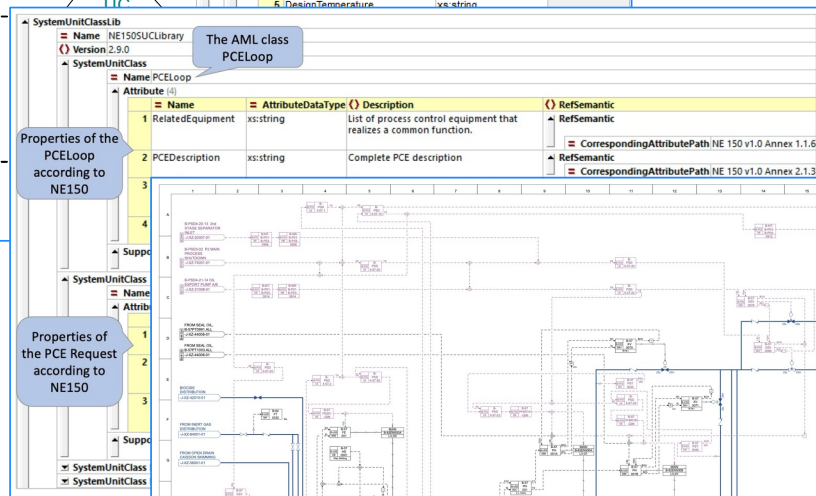
Property modelling with AutomationML



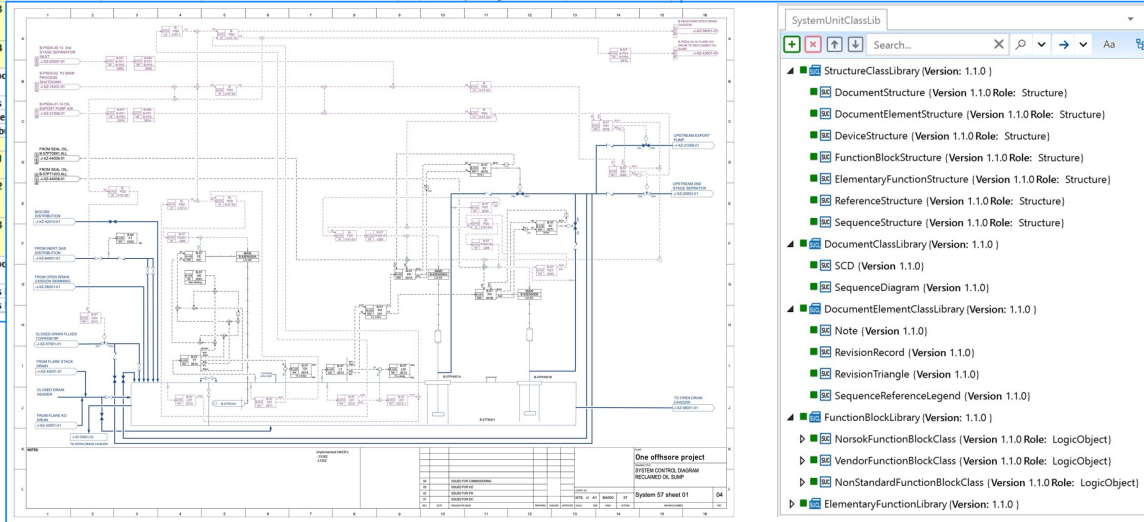
Examples of good AML Models



IEC 62424, a simple digital object model for PCE requests (Mess-Stelle)

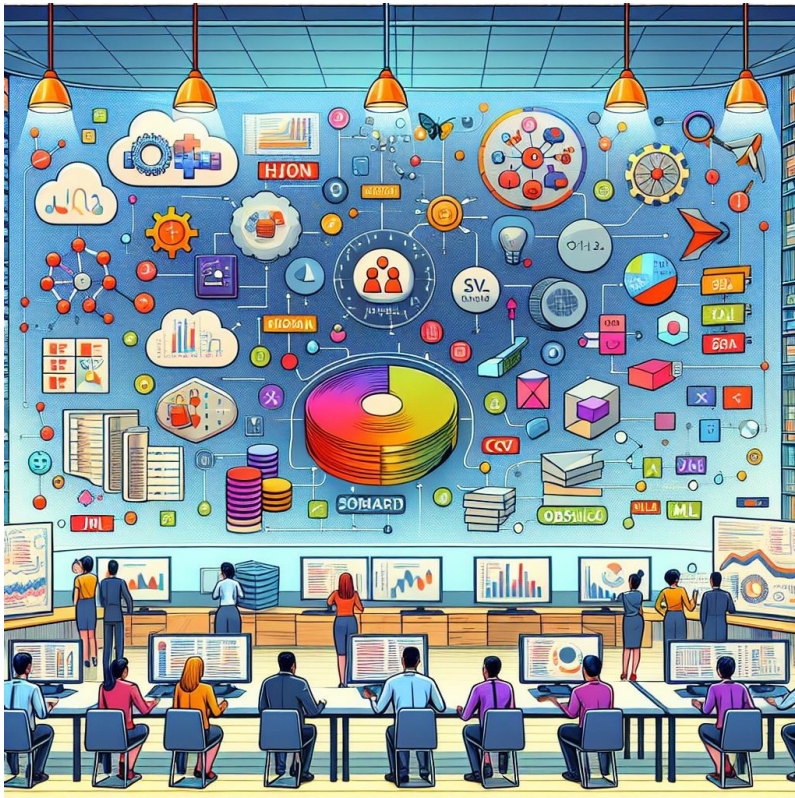


Namur Container, a comprehensive object model for PCE requests



SCD Diagrams for Oil Platforms

Creating data models has become easy



Result

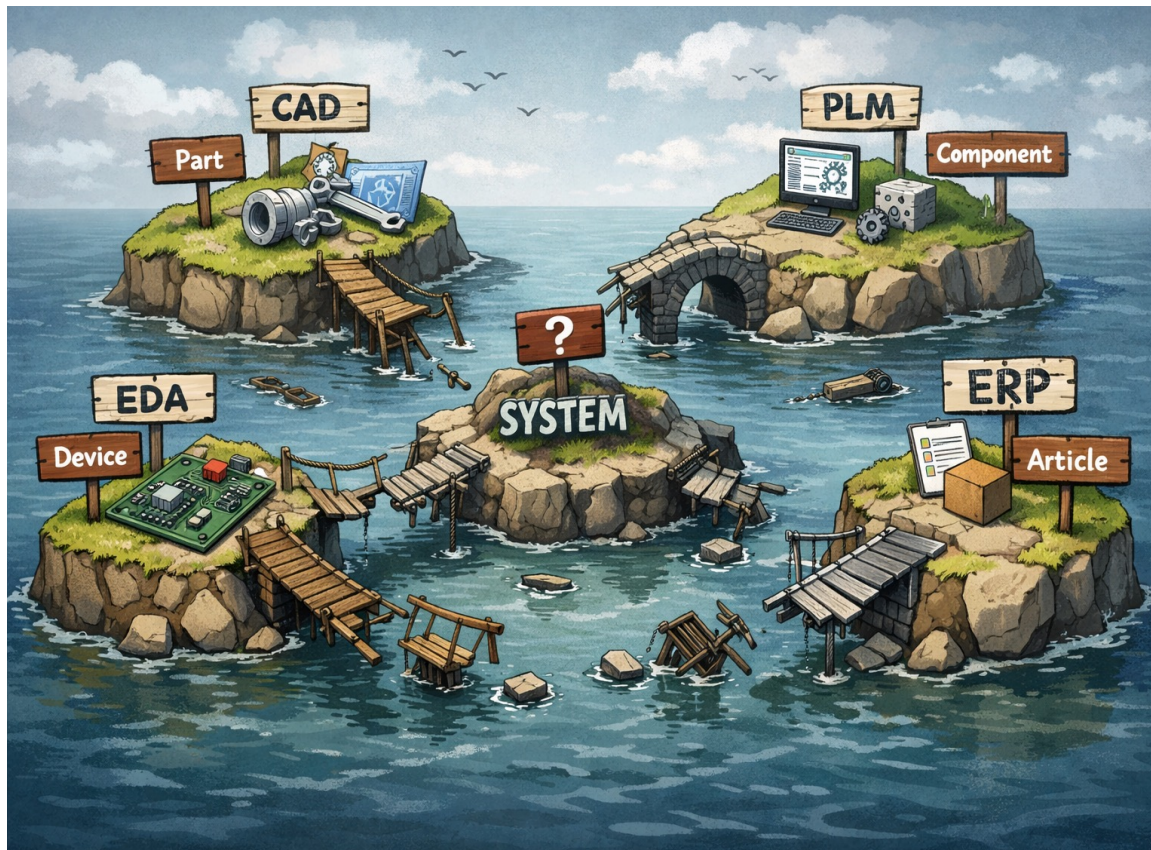
- » **Mindset:** Better to start from scratch than to reuse
- » Many overlapping and isolated data models
- » A Babel-like **diversity** of data models, semantics and ontologies
- » No standardized methodology for managing this diversity
- » No interconnectivity among these diverse elements
- » Vendor-dependent, non-standardized, and expensive siloed solutions

Approaches

- » **Committees** for the standardization of data models
- » **Committees** for the standardization of terminology, semantics, ontologies, and metadata models
- » Local **standards**, corporate **standards**, de facto **standards**, DIN/IEC/ISO standards
- » ECLASS, IEC CDD, many more

Reference System
QUDT
IEC-CDD
ECLASS
ETIM / bSDD
UNSPSC
GS1 Digital Link
IEC 61406 IDLink
VEC Ontology
Eclipse ESMF (SMM)
SAREF & SAREF4INMA
OPC UA Information Model
Wikidata
Semantic Hub (proposed)

Weaknesses of Semantic Data Models



There is a lack of **mechanisms for mapping taxonomies** as the basis for semantic linking

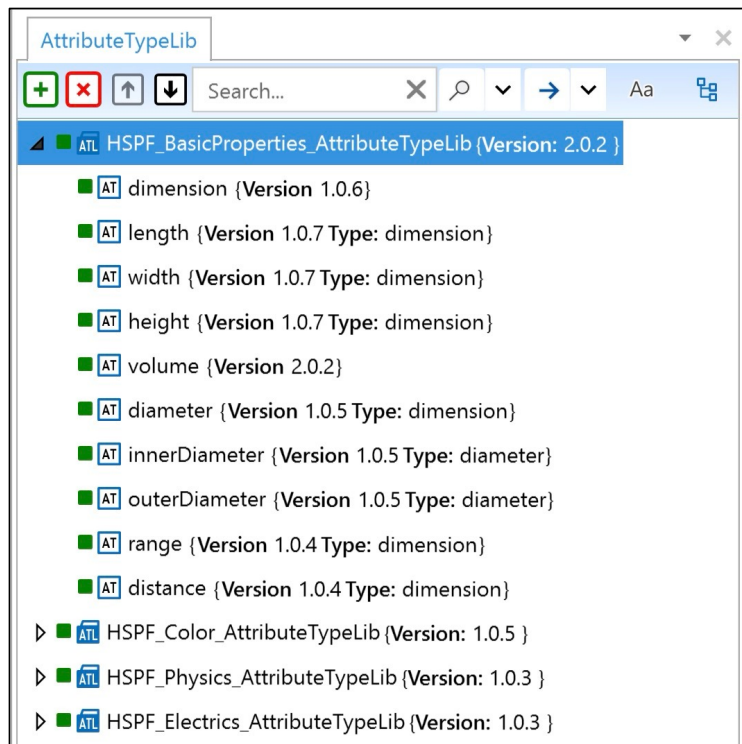
Examples of taxonomies:

ECLASS, IEC CDD, many proprietary taxonomies

Problems:

- » Paywall
- » Interoperability between different taxonomies
- » Lack of cross-referencing

Challenges for AutomationML modellers



Question 1: How can we provide a semantic system to AML users??

Question 2: How can such AML attributes be semantically referenceable?

Question 3: How can we relate AML properties to existing reference systems?

To solve these questions:

- » How to identify relevant properties?
- » How to publish property libraries?
- » How to make our property attributes referencable?
- » How to reference each property?
- » How to reference other taxonomies?
- » How to make this available to the world?

How to identify relevant properties?

SI Units	Base units	high	distributionFunction	ISO 3534-1
ISO 80000-3	space and time	high	cumulativeDistributionFunction	ISO 3534-1
ISO 80000-4	mechanics	high	probabilityDensityFunction	ISO 3534-1
ISO 80000-5	thermodynamics	medium	probabilityMassFunction	ISO 3534-1
ISO 80000-6	electromagnetism	high	normalDistribution	ISO 3534-1
	light and optical radiation in the wavelength range of		lognormalDistribution	ISO 3534-1
ISO 80000-7	approximately 1 nm to 1 mm	low	exponentialDistribution	IEC 61703
ISO 80000-8	quantities of acoustics	low	weibullDistribution	IEC 61649
ISO 80000-9	physical chemistry and molecular physics	low	gammaDistribution	IEC 61703
ISO 80000-10	atomic and nuclear physics	low	erlangDistribution	IEC 61703
ISO 80000-11	transport and transfer	low	poissonDistribution	IEC 61703
ISO 80000-12	condensed matter physics	low	chiSquareDistribution	ISO 3534-1
IEC 61508	Functional Safety	high	studentTDistribution	ISO 3534-1
			fDistribution	ISO 3534-1
			ISO 3534-1: Statistics — Vocabulary and symbols — Part 1: General statistical terms and terms used in probability	
			IEC 61703: Mathematical expressions for reliability, availability, maintainability and maintenance support terms	
			IEC 61649: Weibull analysis (für Weibull-spezifische Eigenschaften)	

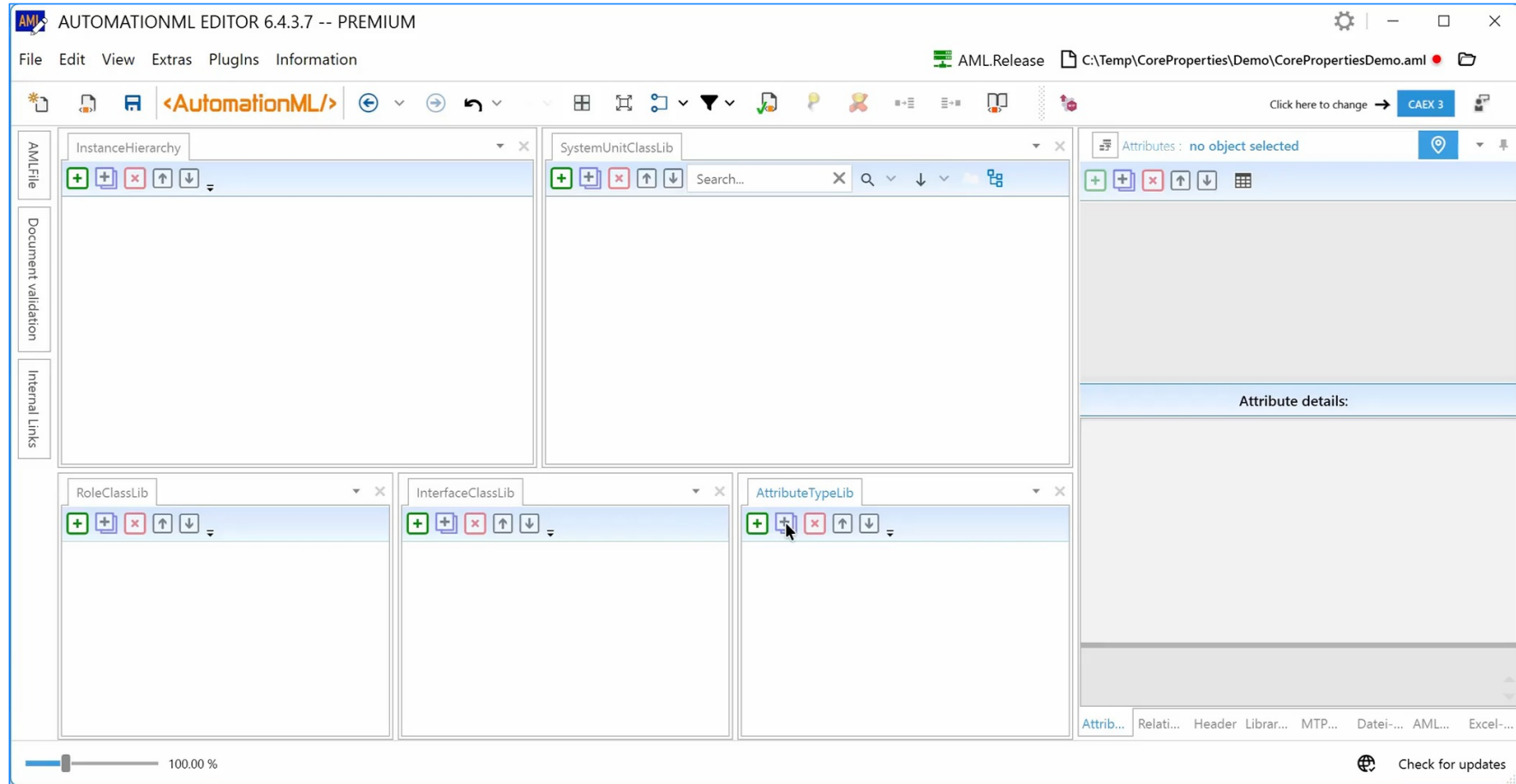
New in 2026:

Series of AML Core Property Libraries

- ▷  AutomationML_CoreProperties.Foundation_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.IdentificationAndClassification_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Time_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Kinematics_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Physical_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Performance_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Lifetime_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Electrical_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Mechanical_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Physical.Constants_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.EnvironmentalConditions_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.StatisticsAndDistribution_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.MeasurementQuality_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.SafetyAndReliability_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.CommercialProperties_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Contact_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Nameplate_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Appearance_AttributeTypeLib {Version: 1.7.9 }
- ▷  AutomationML_CoreProperties.Geometry_AttributeTypeLib {Version: 1.7.9 }

Life Demo

Demo



<AutomationML/>

Main Questions of this talk

- » How to identify relevant properties? [Standards](#) 
- » How to publish property libraries? [AML Library Space](#) 
- » How to make our property attributes referencable? [AML -ID](#) 
- » How to reference each property?
- » How to build bridges with other taxonomies?
- » How to make this available to the world?

How to reference each property

Several options



Plain IRDI Code

0173-1#01-AAB906#022

→ No standard way to resolve it online



Web-Resolvable IRI

<https://prod.etim-international.com/class/EC000003>

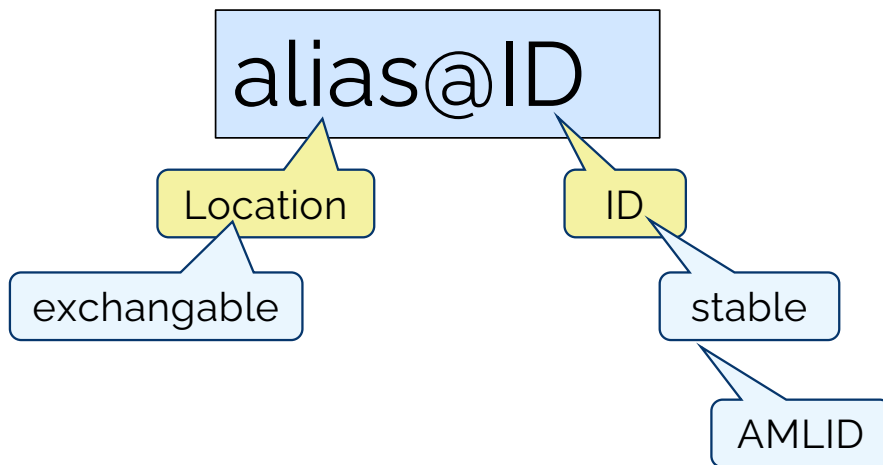
→ Plain HTTP GET returns the definition in some format

- » IRDI: cannot be looked up by a machine.
- » IRI: once the location changes, the IRI is broken
- » Publishing an identifier and making its definition actually reachable over the web are two different engineering problems -today, only the first one is reliably solved.

Feature comparison of semantic reference systems

Reference System	Web-resolvable ID	Open/Public API	License	Versioning	Linked Data / SPARQL
QUDT	Yes (persistent URIs)	Yes (SPARQL endpoint)	Open	Yes (versioned releases)	Yes (native RDF)
IEC-CDD	No (no standard URI scheme)	No public API	Proprietary	Partial (no default-version resolution)	No
ECLASS	No (IRDI not directly resolvable)	Read-only, coarse-grained	Licensed	Yes (releases)	No
ETIM / bSDD	Yes (persistent URIs)	Yes (GraphQL + HTTP)	Open (ETIM data licensed per domain)	Yes	Partial
UNSPSC	No (plain code)	Limited (UNGM helper endpoints)	Open	Yes (annual)	No
GSI Digital Link	Yes	Yes (resolver network)	Open (GSI governance)	Yes	Yes (JSON-LD)
IEC 61406 IDLink	Yes	Yes (domain-owner operated)	Open standard	Yes	Partial
VEC Ontology	No (single-file RDF download)	No	Open	Yes (releases)	Partial (offline)
Eclipse ESMF (SAMM)	No (URN-based, needs resolver)	Yes (tooling, not a public registry)	Open source	Yes	Yes (RDF/Turtle)
SAREF & SAREF4INMA	Yes (versioned hash IRIs)	No (portal search, no SPARQL/REST)	Open (ETSI Software License)	Yes (per-version namespaces)	Partial (RDF/OWL, no endpoint)
OPC UA Information Model	No (NodeId/namespace, server-bound)	Yes, but server-specific	Open (companion specs)	Yes	No
Wikidata	Yes (persistent URIs)	Yes (SPARQL + dumps)	CC0	Continuous (no discrete releases)	Yes (native RDF)
Semantic Hub (proposed)	Yes	Yes (REST + SPARQL)	CC0 for native content; provenance-tagged for imports	Yes	Yes

AML Referencing approach



AML elements get a unique UUID

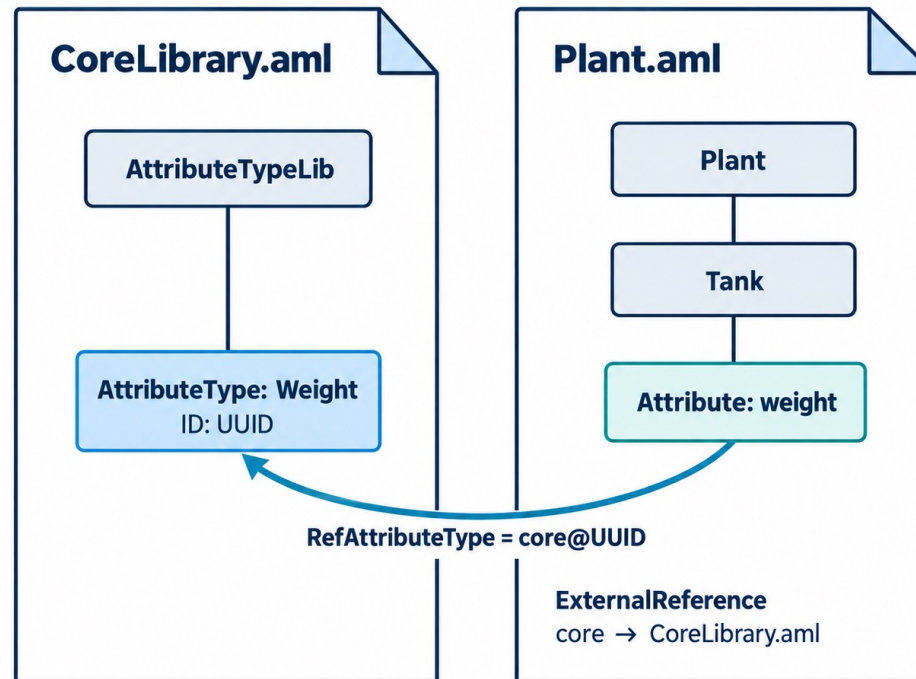
For resolvability: references to AML elements combine an element's UUID with an alias for external documents and an attribute path where required.

- » reference to AML element: alias@UUID
- » reference to Attribute in an AML Element UUID/attribute/path
- » Example: demoAlias@60c49768-ef30-496e-gdfc-6agbdcdbfa893/address/street

Result: Stable alias, flexible file location

- » An `ExternalReference` maps an alias to a document's `Path`. Keep the alias unchanged when updating the path, so existing references remain valid.
- » The alias is local to the referencing document—not a persistent identity of the external document.
- » The alias identifies the referenced AML document only within the scope of the referencing AML document and does not constitute the persistent identity of the referenced AML document.

AML Referencing approach



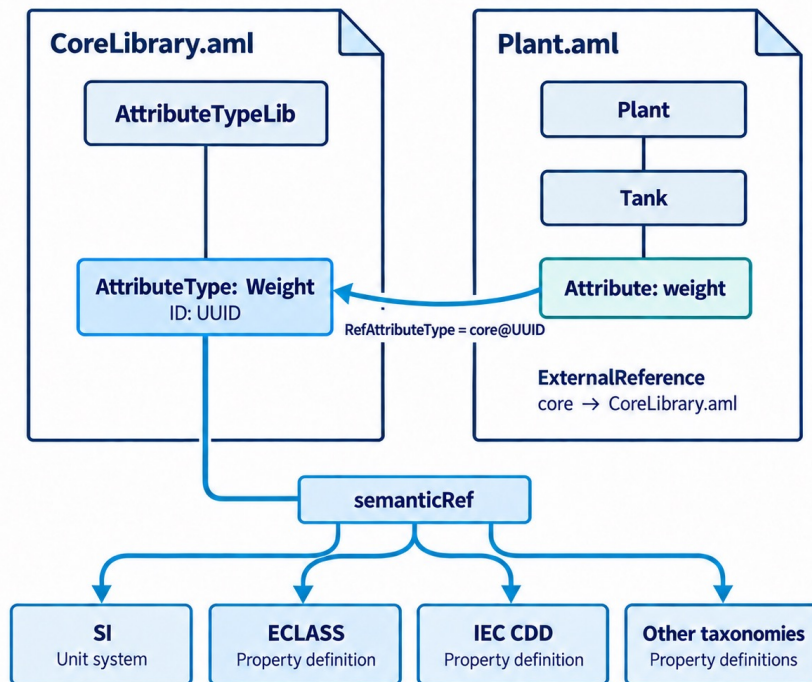
UUID = unique ID of the attribute type

Main Questions of this talk

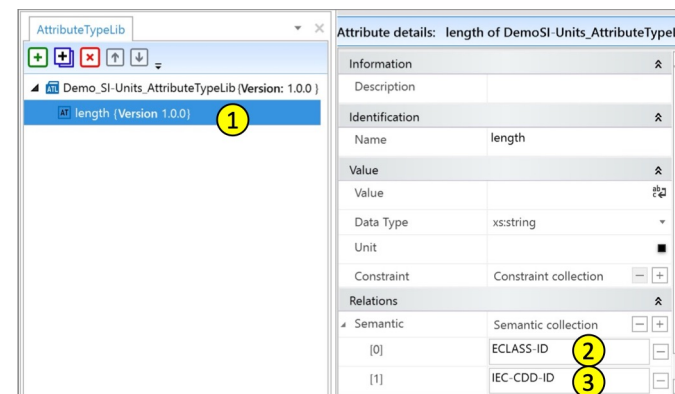
- » How to identify relevant properties? 
- » How to publish property libraries? 
- » How to make our property attributes referencable? 
- » How to reference each property? 
- » How to build bridges with other taxonomies?
- » How to make this available to the world?

How to build bridges between taxonomies

AML References & External Semantics



- » Idea: we add semanticReferences to out Attribute Types as much as we can
- » via versioning, we can add more and more, even multiple per type.



ID	f4338767-4a68-4cef-99e5-e9b490af060f
Name	length

Main Questions of this talk

- » How to identify relevant properties? 
- » How to publish property libraries? 
- » How to make our property attributes referencable? 
- » How to reference each property? 
- » How to build bridges with other taxonomies? 
- » How to make this available to the world?

How to make this available to the world?

1. Via AML Engine (available)
2. Via API (REST, MCP) on our server (next step)
3. Via API on customers server, just file access to our server (next step)

Main Questions of this talk

- » How to identify relevant properties? 
- » How to publish property libraries? 
- » How to make our property attributes referencable? 
- » How to reference each property? 
- » How to build bridges with other taxonomies? 
- » How to make this online available to the world?
 - Solved within the AML Editor
 - Still open: as web service

Markus Rentschler, ETFA 2026

The AAS approach for Semantic Identifiers



- **semanticId**: allows every Submodel and Property to carry a global, unambiguous reference to its intended meaning.
- **supplementalSemanticIds**: additional references that can be assigned.

Table 2. Attributes of the Submodel instance

IdShort:	HandoverDocumentation		
Class:	Submodel		
semanticId:	0173-1#01-AHF578#003		
Parent:	-		
Explanation:	The Submodel defines a set meta data for the handover of documentation from the manufacturer to the operator for industrial equipment		
Element details:	-		
[SME type]	semanticId	[valueType]	card.
idShort	Description@en	example	
[SML]	0173-1#02-ABIS00#003	[]	1
Documents	supplementalSemanticId: https://api.eclass-cdp.com/0173-1-02-ABIS00-003	1 elements	
	Information for a document identity		
[SML]	https://admin-shell.io/vdi/2770/1/0/EntitiesForDocumentation	[]	0..1
Entities	List of Entities	1 elements	

```
https://smt-repo.admin-shell.io/api/v3.0/submodels/aHR0cHMGLy9hZG1pb1t2agVisC5pb3pZHRRL
{
  "modelType": "Submodel",
  "kind": "Template",
  "semanticId": {
    "keys": [
      {
        "type": "Submodel",
        "value": "0173-1#01-AHF578#003"
      }
    ],
    "type": "ModelReference"
  },
  "supplementalSemanticIds": [
    {
      "keys": [
        {
          "type": "GlobalReference",
          "value": "https://api.eclass-cdp.com/0173-1-01-AHF578-003"
        }
      ],
      "type": "ExternalReference"
    }
  ],
  "administration": {
    "revision": "0",
    "templateId": "https://admin-shell.io/Idta-02004-2-0",
    "version": "2"
  },
  "id": "https://admin-shell.io/Idta/SubmodelTemplate/HandoverDocumentation/2/0",
  "description": [
    {
      "language": "en",
      "text": "The Submodel defines a set meta data for the handover of documentation from the manufacturer to the operator for industrial equipment"
    }
  ],
  "idShort": "HandoverDocumentation",
  "submodelElements": [
    {

```

But: Assigning a semantic reference is only half the job. Making that reference actually resolvable to a machine-readable definition is the other half — and that's where things get hard!

"The Semantic Hub - Unifying Fragmented Semantic Dictionaries for Industry 4.0 and AAS",
ETFA26-000364, Session WS06a, Tuesday, 8th September, Markus Rentschler

6



<AutomationML/>

The Glue for Seamless
Automation Engineering

One more thing...

A new universal reference concept for AutomationML

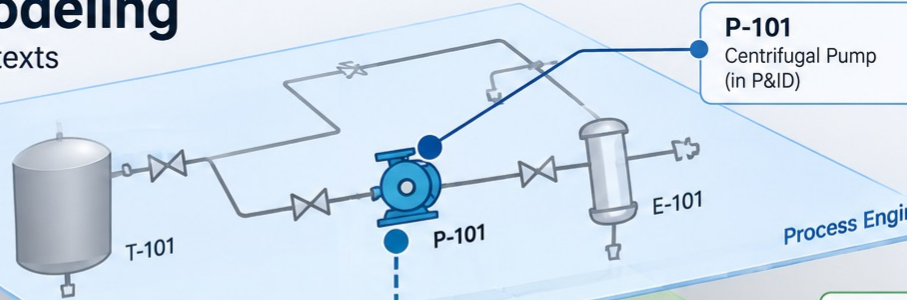
New: multi context modelling,
multi user engineering,
modelling of semantic relations

Multi-Layer Modeling

One object – multiple contexts

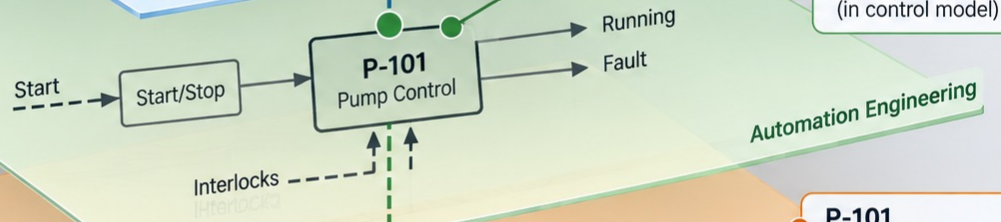
Process Layer

P&ID
Certified model



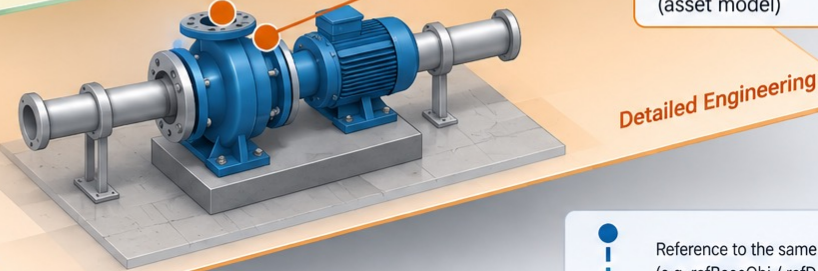
Control Layer

Function Block Diagram
Control logic



Asset Layer

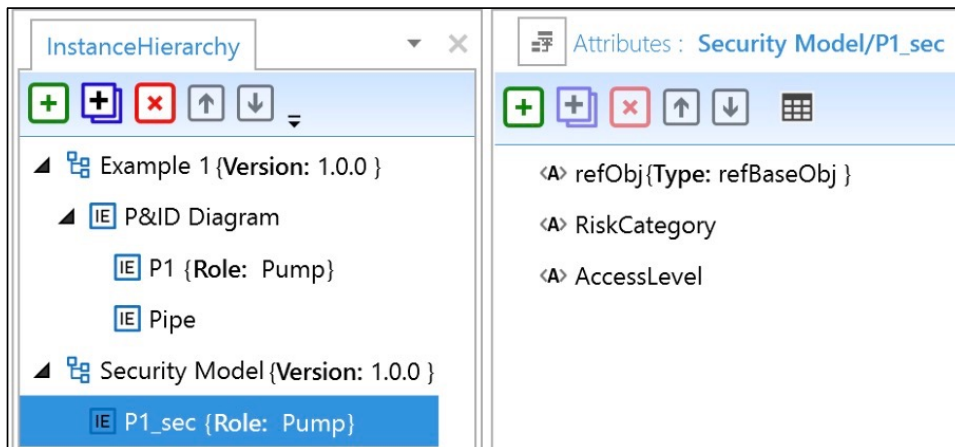
Detailed model
Mechanical, electrical,
documentation



Same object.
Different levels of detail.
Different context.
One source of truth.

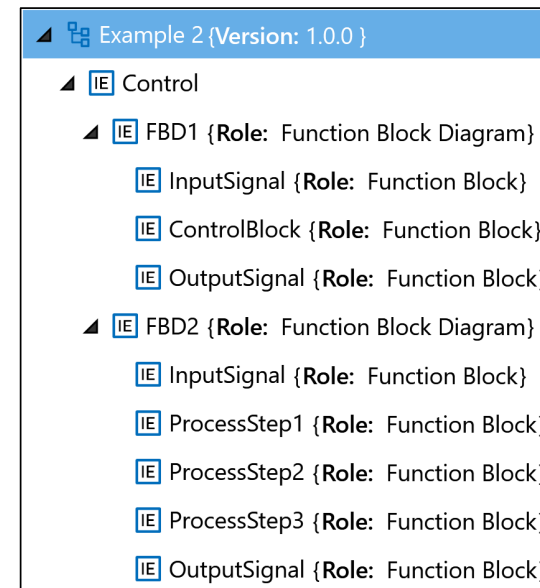
Reference to the same object
(e.g. refBaseObj / refDetailObj)

Two engineering use cases



Same object — different context

- P1 exists in the certified P&ID and in a security model.
- Security data belongs to another team and must not modify the P&ID.



Same object — different detail level

- A control block is part of a Function Block Diagram.
- Its internals are modeled separately, but both representations describe the same object.

Idea: a small library of base object reference types

AutomationML_ObjectReferences_AttributeTypeLib {Version: 1.1.1-beta }

AT refObj {Version 1.0.0}

AT refBaseObj {Version 1.0.0 Type: refObj}

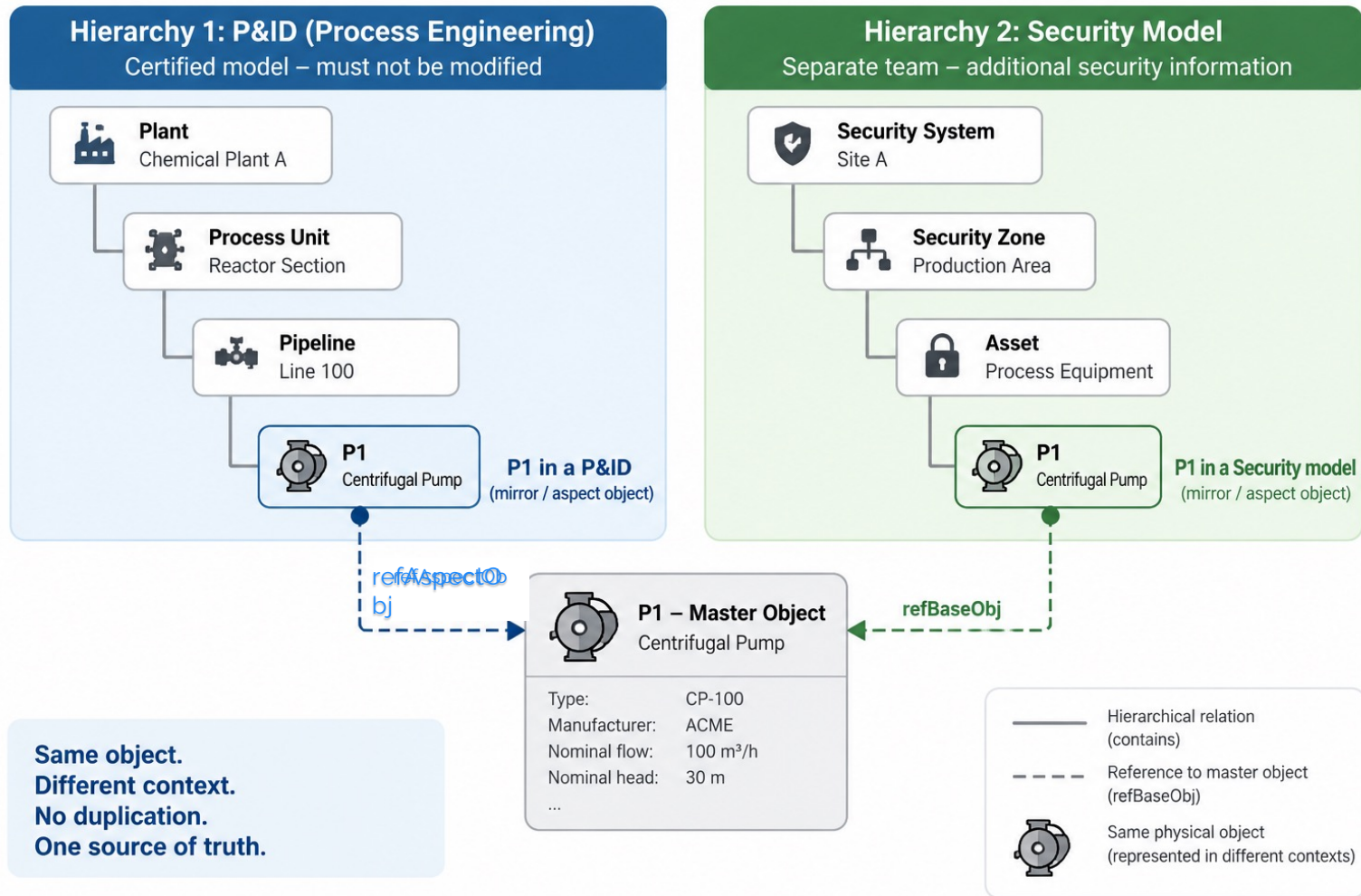
AT refAspectObj {Version 1.0.0 Type: refObj}

AT refDetailObj {Version 1.0.0 Type: refObj}

AT refAbstractObj {Version 1.0.0 Type: refObj}

- **refObj** — generic abstract reference
- **refBaseObj** / **refAspectObj** — base object ↔ aspect object
- **refDetailObj** / **refAbstractObj** — abstract ↔ detailed representation
- Back references are optional.

Generic rule:
Add a reference attribute and point its value to the referenced object.



Result: a new reference concept for AutomationML

New

- Multi-context modeling without changing the CAEX / AutomationML core.
- Distributed multi user ownership of a model
- Supports context-specific information, different detail levels, model protection
- Reference library is already available in the AML Editor.

**In comparison to the CAEX Mirror Concept:
Keep one source of truth —
without forcing every representation to be identical.**

Next Steps

1. Sanity Checks of the libraries
2. Add amounts of semantic IDs (as much as possible) directly into each attribute type
3. Development of domain specific property libraries
4. AML Editor support of the reference concept
5. API (REST, MCP, ...)

Summary



- » For the first time, we are using modern, Git-inspired workflows and adapting them for engineering information models.
- » For the first time, we have a comprehensive approach to provide a domain agnostic library of core properties
 - Semantically referenceable
 - Directly derivable
 - Derivable later or semantically referenceable
 - Publicly available, MIT License
 - Versionable → expands over time
 - Stable yet extensible.
- » AML Library Spaces and AML ID makes AML features semantically referenceable
- » Enabled for the mapping of external taxonomies
- » Available today as beta



AutomationML e.V. c/o IAF
Universitätsplatz 2
39106 Magdeburg
office@automationml.org