# *Semantics for PI Application Profiles*

## *White Paper*

## *Cooperation PI and ECLASS*

*Version 1.0 – Date Mar. 09, 2021*

*Order No.: 3.912*

## File name : PI-Semantics-Whitepaper_3912_V10_Mar21

Prepared by Joint ECLASS and PI Working Group in the Committee C3 “Application Profiles”.



In this specification the following key words (in **bold** text) will be used:

**may:** indicates flexibility of choice with no implied preference.

**should:** indicates flexibility of choice with a strongly preferred implementation.

**shall:** indicates a mandatory requirement. Designers **shall** implement such mandatory requirements to ensure interoperability and to claim conformance with this specification.

Publisher:

PROFIBUS Nutzerorganisation e.V.
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone : +49 721 / 96 58 590
Fax: +49 721 / 96 58 589
E-mail: info@profibus.com
Web site: www.profibus.com

ECLASS e.V.
Konrad-Adenauer-Ufer 21
50668 Cologne
Germany
Phone: +49 221 4981 811
E-mail: info@eclass.de
Web-site: www.ECLASS.eu

# Content



 

## List of figures



 

## List of tables



## Revision Log

| Version | Originator | Date | Change Note / History / Reason |
|---|---|---|---|
| 0.1 | JWG PI ECLASS | 25.Feb. 20 | Initial version |
| 0.2 | | 08. Mar. 20 | Improved version after JWG discussion |
| 0.3 | | 30. Jun. 20 | PNO C3 preview distribution |
| 0.4 | | 26. Sept. 20 | JWG internal review, without XML update |
| 0.5 | | 08. Dec. 20 | including all review comments from JWG, without XML update |
| 0.9 | | 22. Feb. 21 | Final review version for JWG |
| 1.0 | | 09. Mar. 21 | released |

 

# Motivation

The intelligent manufacturing networks of digital factories will only become reality with machine-processable, standardized information, because they have to interact over different communication technologies, different companies and even across industry boundaries. PROFIBUS & PROFINET International (PI) sees its role in creating the necessary basis for this fact, especially in view of Industrie 4.0.

This has been accomplished by providing device-related expertise for PI technologies – for instance, in the form of parameters in the open device profiles or other specifications for the application layer, like for an asset management record. However, in order for these to be used as the basis for a machine-processable flow of data across the various systems – from sensor to cloud – the data that is already available today must be transformed into clearly usable information by means of semantic standards.

ECLASS e.V. is playing an increasingly important role in industrial automation and especially for Industrie 4.0. In cooperation with ECLASS e.V. existing PI specifications will be expanded by semantic identifiers. In a first step, PI application profiles (e.g., PA Profile) were selected, having the best preconditions for this endeavor.

The results form a significant basis for the automated interaction of various systems and components from different manufacturers are essential for business processes to run optimally between end users, suppliers, customers, etc. in systems of the Industrie 4.0 generation.

 

# 1 Management Summary - Scope of this Document

This white paper presents the results from the joint working group PI and ECLASS.

It describes how to extend already existing PI Application Profiles (e.g., PA Profile, PROFIdrive) with standardized semantic information.

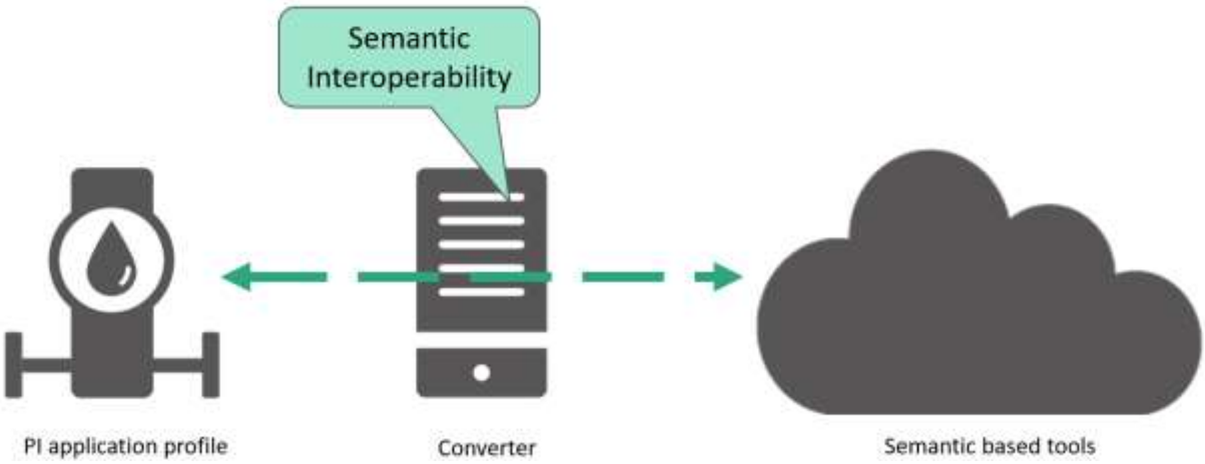


**Figure 1: Overview**

A converter can provide the necessary transformation

- from standardized data defined by PI Application profiles
- to fieldbus neutral technologic oriented semantic information

The PI application profiles itself do not need to be changed. They are extended with a generic approach to support semantic interoperability.

Today not all data provided by PI application profiles already have such a semantic information, it is future work to provide a complete set of information.

The concept supports primarily ECLASS, nevertheless also other semantic standards like IEC CDD (common data dictionary) or semantic web will be supported.

In context with Industrie 4.0, the above-mentioned converter can also provide the content for the asset administration shell.

 

# 2 Related documents and references

## 2.1 Related documents

[1] PI "Identification & Maintenance Functions" Profile Guidelines Part 1, Order No.: 3.502

[2] PI "Data types, Programming Languages, and Platforms" Profile Guidelines, Part 2, Order No: 3.512

[3] PI "Profile for Process Control Devices", Order No: 3.042

[4] PI "Topology and Asset Discovery for PROFINET Guideline", Order No. 7.182

## 2.2 References

[100] ECLASS Wiki: https://wiki.eclass.eu/wiki/Main_Page

[101] ECLASS white paper "Toward smart manufacturing with data and semantics"

[105] IEC Common Data Dictionary: https://cdd.iec.ch/

[110] https://schema.org/

[111] http://iotschema.org/

# 3 Definitions and Abbreviations

## 3.1 Definitions

### 3.1.1 Block

Blocks are a means to structure properties in a convenient and task related way (from IEC 61360-1).

### 3.1.2 COMDO

ECLASS, IEC and ISO - want to provide a Common Data Repository for Smart Manufacturing, Industrie 4.0 and other domains for classifying and describing things (products, services, procedures) along the process life cycle - based on high quality ISO/IEC Standards - to enable electronic data transfer in an unambiguous way and to support digitalization of business by all international market participants.

Since October 1, 2020 COMDO is a project as the 3 partiers (ECLASS, IEC and ISO) support the preparation and signing of an agreement (so-called MoU).

### 3.1.3 Data

representation of information in a formalized manner suitable for human or automatic processing (source: http://www.electropedia.org/iev/iev.nsf/display?openform&ievref=171-01-02)

### 3.1.4 TUF – Transaction Update Files

ECLASS provides the Transaction Update Files (TUF) containing all information that is needed for a semi-automated ECLASS version update (e.g., migration from version 11.1 to version 12.0)

### 3.1.5 IEC CDD

IEC Common Data Dictionary. See [105]. Actually, IEC CDD offers data structures (classification and associated properties) in 3 technical areas based on international standards:

 

- Electric/Electronic Components (IEC 61360-4)
- Process Automation (IEC 61987)
- Low voltage switchgear and controlgear (IEC 62638)

### 3.1.6 Information

Knowledge concerning objects, such as facts, events, things, processes, or ideas (including concepts) that, within a certain context, has a particular meaning (source: http://www.electropedia.org/iev/iev.nsf/display?openform&ievref=171-01-01)

### 3.1.7 Ontology

An ontology is a way of showing the properties of a subject area and how they are related, by defining a set of concepts and categories that represent the subject (source: https://en.wikipedia.org/wiki/Ontology_(information_science))

### 3.1.8 OWL

Web Ontology Language is a family of knowledge representation languages for authoring ontologies. Source: https://en.wikipedia.org/wiki/Web_Ontology_Language

### 3.1.9 Property

A property describes a characteristic of a product (e.g., material, colour, article number) [100]. Source: IEC 61360-1 Chapter 6 "Property".

### 3.1.10 Value

A property has values which determine the characteristics. A value can be a numerical quantity with or without a unit, or a text that describes a subject in more detail (e.g., property: colour, value: red; property: length, value: 25 mm). Source: IEC 61360-1 Chapter 6 "Property".

### 3.1.11 Variable

A PI application profile standardizes variables. They can be

- Parameters (how to parametrize the device)
- Measured values (can be transported cyclic or acyclic)
- Enumerations (e.g., Unit, sensor type)
- Diagnosis information

Variables consist of two main parts

- The value itself
- The access to the value (addressing)

## 3.2 Abbreviations

| | |
|---|---|
| IRDI | International Registration Data Identifier, based on ISO 29002-5 |
| JWG | Joint Working Group |
| LOP | List Of Properties |
| W3C | World Wide Web Consortium |

## 4 Semantic introduction

Goal of this white paper is to describe ways how machines can interact with each other. But first, we look to humans. The science which applies to study of human language is linguistics.

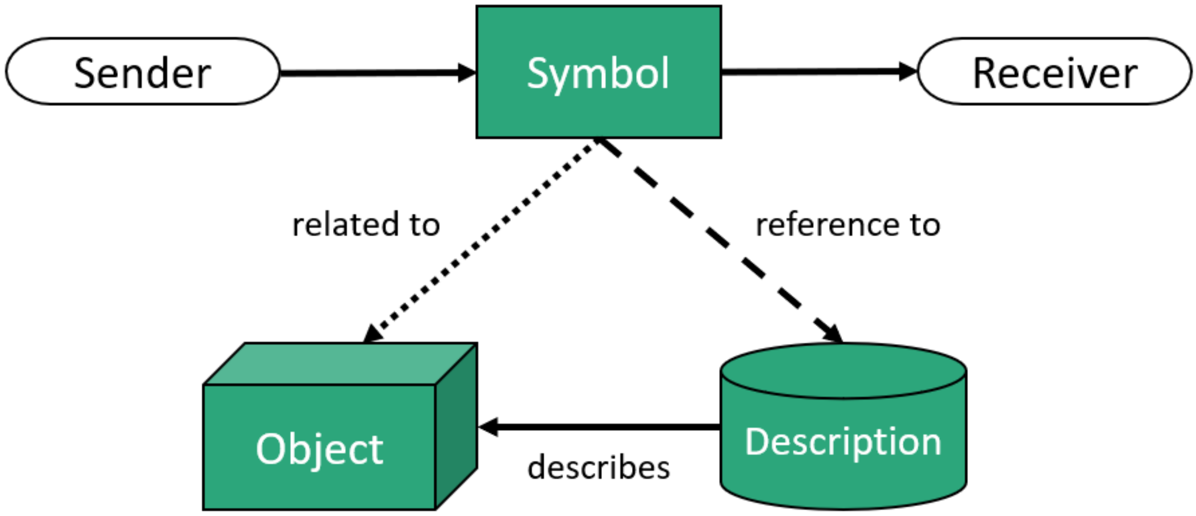


**Figure 2: Semiotic triangle**

The semiotic triangle (also known as the triangle of meaning or the triangle of reference) is a model of how linguistic symbols are related to the objects they represent (from Wikipedia "Triangle of reference"). The sender and receiver can understand each other if both use the same symbol for the related object and have the same description of the object in mind. The description explains the object. To ensure this, the symbol references to a description of the object as well.

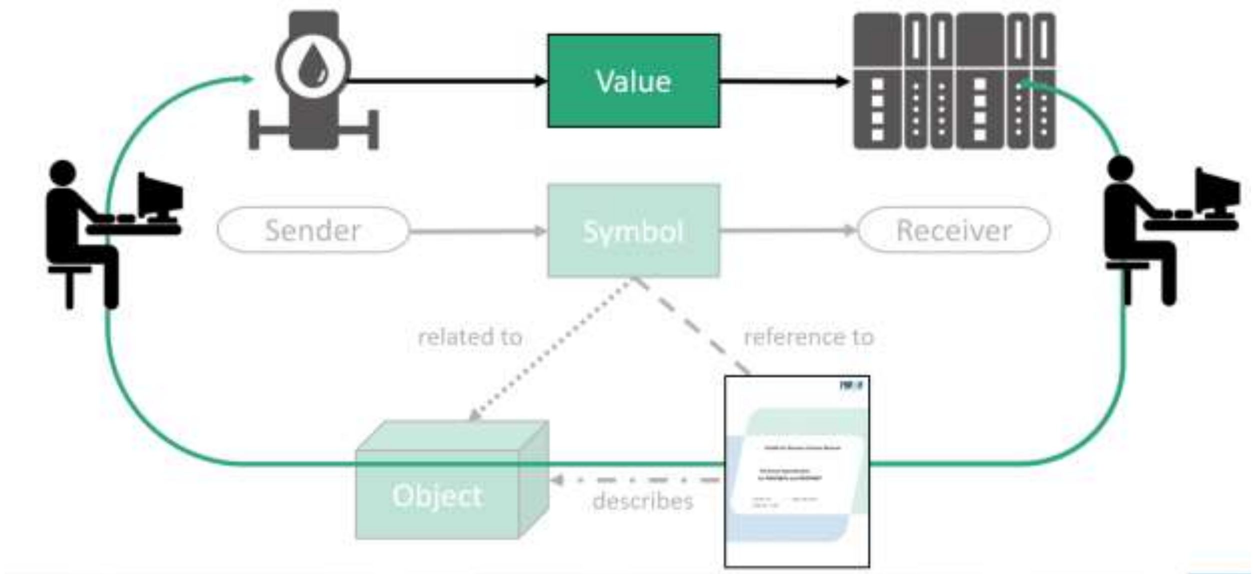


**Figure 3: Semiotic triangle for PI application profiles**

Today PI application profiles provide in the context of field devices such a "description" for different types of devices. But this description is made for humans: The meaning of the "symbols" is known by the programmers and coded in the software of the devices and controllers.

In future scenarios, e.g., smart manufacturing and Industrie 4.0 machines are directly talking with each other. To understand each other, the exchange of the value itself is not enough, an additional identification (ID) to specify the value transferred is necessary. The “symbol” in this case is a combination of ID and value and the ID now references to a machine interpretable description.

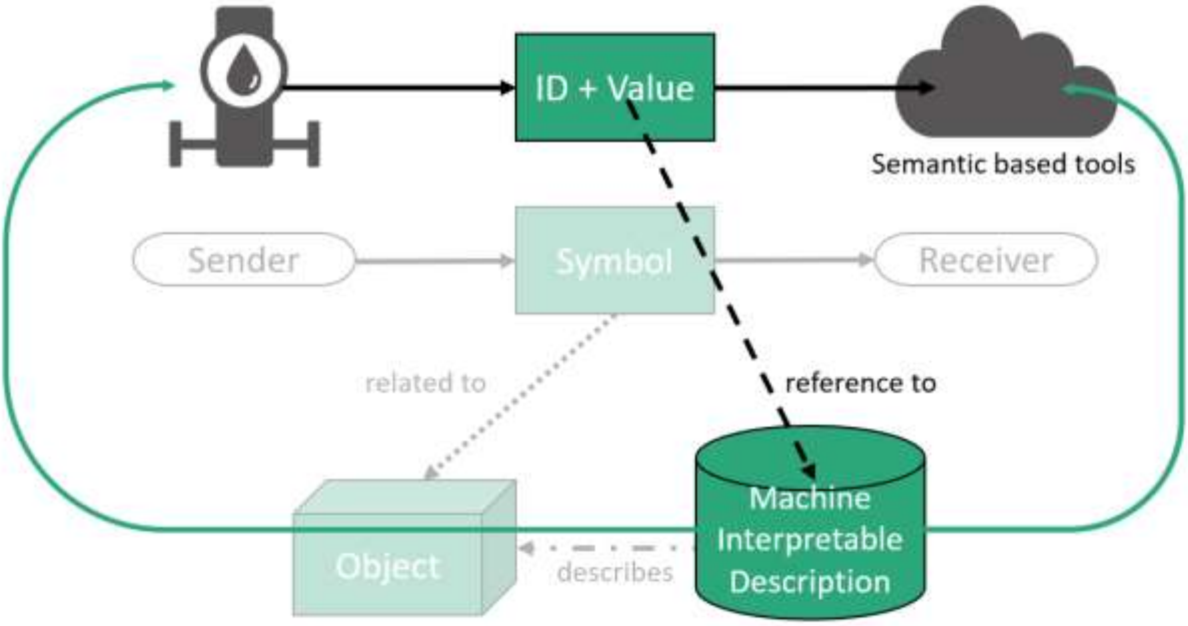


**Figure 4: Semantics for PI application profiles**

This concept can be used for machine-to-machine communication and / or any semantic based tools analysing information. The additional ID needed is called the semantic identifier.

We all know, that there is not only one single device used in a plant but many devices even of the same type measuring different technological values. Semantic also provides a concept to handle this problem. This is called the “context”. The context describes the application area and is again represented by a semantic identifier.

The concatenation of the semantic identifiers for the context(s) and for the symbol(s) result in a complete technological description. The next figure illustrates this concatenation, building up a path of semantic identifiers.

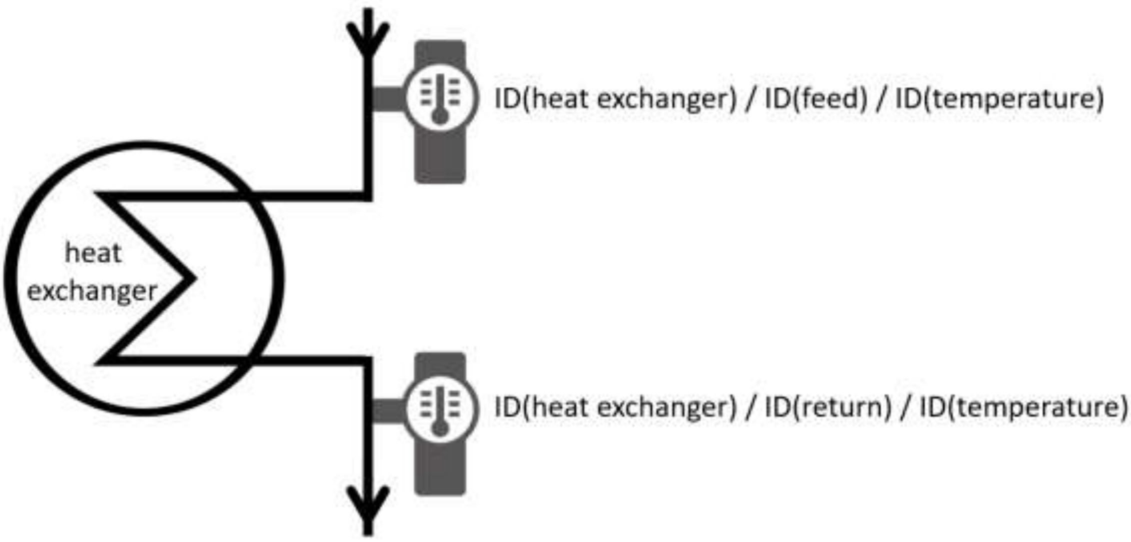


**Figure 5: Semantic context (example heat exchanger)**

 

# 5 Use cases for semantics

The following use cases were discussed during the various meetings of JWG PNO ECLASS. They do not claim to be complete but provide an overview on possibilities of using semantics.

All these use cases rely on an international, standardized description of semantic identifier.

## 5.1 Engineering Workflow

The engineering workflow is characterized by offline scenarios, in which the instance of the device itself is not present or accessible.

### 5.1.1 UC1: Selection of devices

During the engineering phase the plant engineer decided, where a pump, a valve, a flowmeter or a temperature measurement device is needed. Later the engineer has more detailed data for the plant like the flow and the pressure of the pump, the materials, which are resistant against the process demands, the minimal and maximal range of flow and the temperature range, the process will have.

With this information it is possible to start a request to the device supplier to ask for a quotation of a full specified device. Due to the huge number of variants, a pump, a valve, a flowmeter can be manufactured, the supplier provides a web configuration tool to consider all technical data to be able to provide a quotation for a device. After the selection, the supplier can calculate the price, the delivery time, the exact dimensions and weight etc. At the same time, when the engineer gets these data, he can use all the available (technical) product information to work with these (real) data in the engineering and construction software at the earliest possible time.

### 5.1.2 UC2: Ordering devices

The procurement of an engineering company has the task to compare and to buy the specified devices (or services) for the best price. For this, the buyer has to compare the quotations of several manufacturers to find the most suitable product (depending on his/her internal guidelines like price, durability, accuracy, service or spare part expenses, etc.). With a detailed specification of the demand, it is possible to get a high-quality quotation with all relevant or mandatory information to select the most suitable product by using a (semiautomatic) selection tool for the ranking.

### 5.1.3 UC3: Offline parametrization of devices

During the engineering phase and after the selection of a device (see UC1) the (technical) product information is available. The engineer can now start to define the parametrization of the devices' technological variables like min. / max. flow or any other.

The device itself doesn't need to be present at this time, only a list of the supported variables and their standardized meaning is needed as input. If the device supports a PI application profile, PI can provide so called mapping tables, which supply these informations.

### 5.1.4 UC4: Offline parameter migration

If a device must be replaced, then the new device may have a newer version or may be from a different vendor. In this case the parameter set for the device needs also to be replaced. For the transfer of the parameter content from the old device onto the parameter set of the new device, the semantic information can be used to match corresponding parameters and to adapt parameter contents to the semantic of the parameters of the new device.

## 5.2 Runtime Workflows

Runtime workflows are characterized by having an established online communication to the instance of the device. This is typically true during commissioning and operation phase.

 

### 5.2.1 UC5: Inventory list

PI devices support an electronic faceplate, called “Identification and Maintenance Information”. [1] Access to this information is different due to e.g., different communication systems like PROFIBUS or PROFINET.

The PI mapping table “hides” the access and provides the electronic faceplate in a semantically standardized manner. An online scan over all active devices [4] connected to the network provides an inventory list of all devices with their name, manufacturer, order number, serial number, hardware and software revision available in a plant.

### 5.2.2 UC6: Download parametrization

The offline parametrization (see UC 3) can be downloaded to the device instance to ensure the proper functionality.

PI mapping table supports this download by defining the refence between technological value and write access within the device. Of cause the parameter download can also be done by a device specific tool.

**Note:** Typically, more devices instances of the same type are used in a plant. Make sure to identify the desired device instance first.

### 5.2.3 UC7: Check parametrization

Over time the parametrization of device instances may be changed (temporarily or not). Goal of this use case is to detect differences between the engineering phase (planning) and the real configuration which is active in the field.

The PI mapping table supports this comparison by defining the reference between technological value and read access within the device

### 5.2.4 UC8: Provide information for analytics

Typically, PI devices use internally more information from additional sensors built in to provide the right value for the control system.

Goal of this use case is to provide this already available not time critical information for analytics. PI mapping tables offer a technology oriented and standardized way to get this information out of the devices to be used directly as input for any analytics.

 

# 6 Introduction Semantic Organizations

This chapter gives an overview on semantic organizations already specifying semantic identifiers. They provide a neutral technological definition of e.g., product data. The scope of product data may cover catalogue information, parameter and actual values of devices including their engineering, their diagnostic information or other needed information.

This technological definition is independent from any underlying communication system.

There are several independent organisations worldwide supporting such semantic identifiers. This chapter gives a short overview of selected organisations (we do not claim a full worldwide view). The content and examples shown, will focus on PI technology wherever possible, not every other aspect is covered.

## 6.1 ECLASS

ECLASS is most developed worldwide ISO/IEC-compliant data standard for goods and services[1]. ECLASS contains tens of thousands of product classes and unique properties. This standardizes procurement, storage, production, and distribution activities in and between companies - across sectors, countries and languages.

ECLASS is a hierarchical classification system for grouping materials, products and services according to an agreed, systematic structure. All products and services can be described by product-specific and standard-compliant properties, which are identified by IRDI's. They are assigned to the four-level numerical ECLASS class structure. The 4th level of the hierarchical classification is always the level that carries all structures and properties for the exact description of the product type or product instance (depending on the use case).

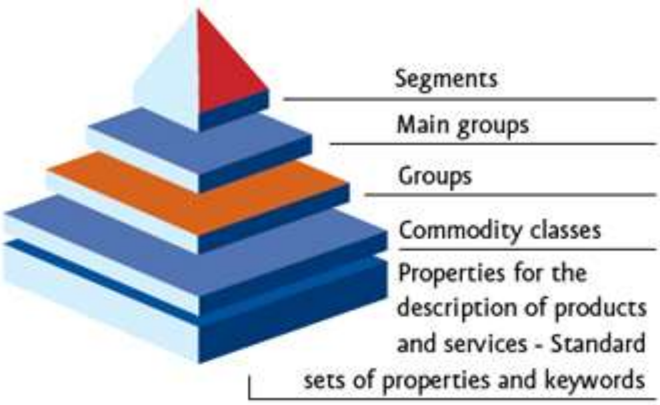


**Figure 6: ECLASS – classification scheme**

Since ECLASS release 7.0, a BASIC Application Class (BASIC AC) and an ADVANCED Application Class (ADVANCED AC) coexist in the same classification system.

The BASIC Application Class is mainly used for e-business catalogues, where people with a limited number of properties can compare and select the device or service they are looking for in a flat list.

The ADVANCED Application Class can be used to provide a detailed description of any device with an almost unlimited number of properties in a hierarchical list with blocks and other structure elements. It is used for engineering and (in the future) for Industrie 4.0 [101] applications, where machines will exchange data without any human interaction.

For the use with PI only the Advanced application classes can be used.

[1] The term "services" in context of ECLASS has a much broader scope than the PI technical term

 

ECLASS covers various segments of product types. Segment 27 contains "Electric engineering, automation, process control engineering" definitions.

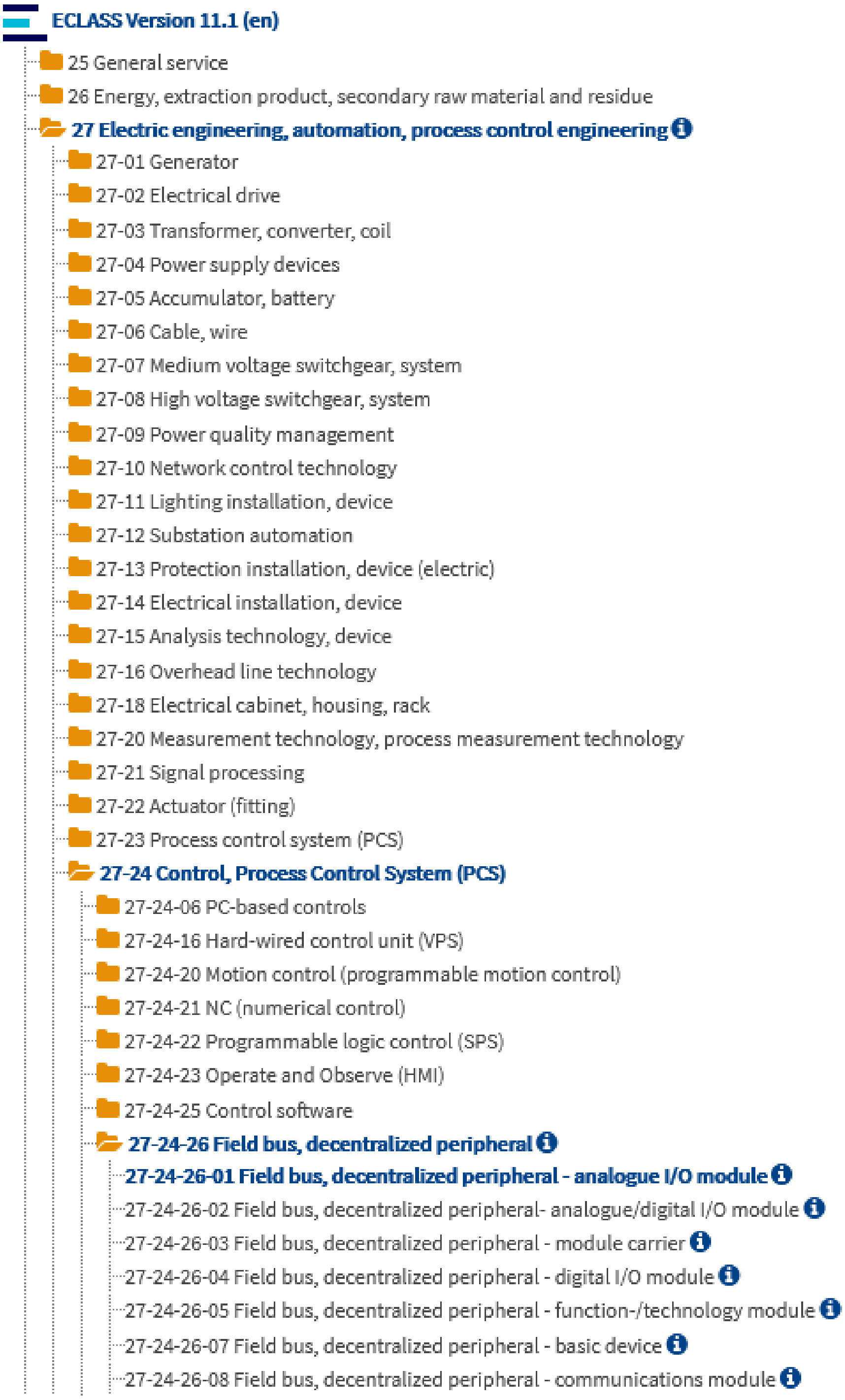


**Figure 7: ECLASS – Segment 27 excerpt (https://www.eclasscontent.com/)**

The list below covers the most relevant main groups and groups relevant for PI technology:

- 19-17 Network technology (computer communication)
- 27-02 Electrical drive
- 27-20 Measurement technology, process measurement technology
- 27-21 Signal processing
- 27-23 Process control system (PCS)
- 27-24-22 Programmable logic control (SPS)
- 27-24-26 Field bus, decentralized peripheral

 

- 27-32 Industrial weighing technology
- 27-37 Low-voltage switch technology
- 37-01-10 Valve (with actuator)

Let's make an example: We choose application class 27-24-26-01: Field bus, decentralized peripheral - analogue I/O module.

This type of module has the Identifier "0173-1---ADVANCED_1_1#01-ADN577#010".

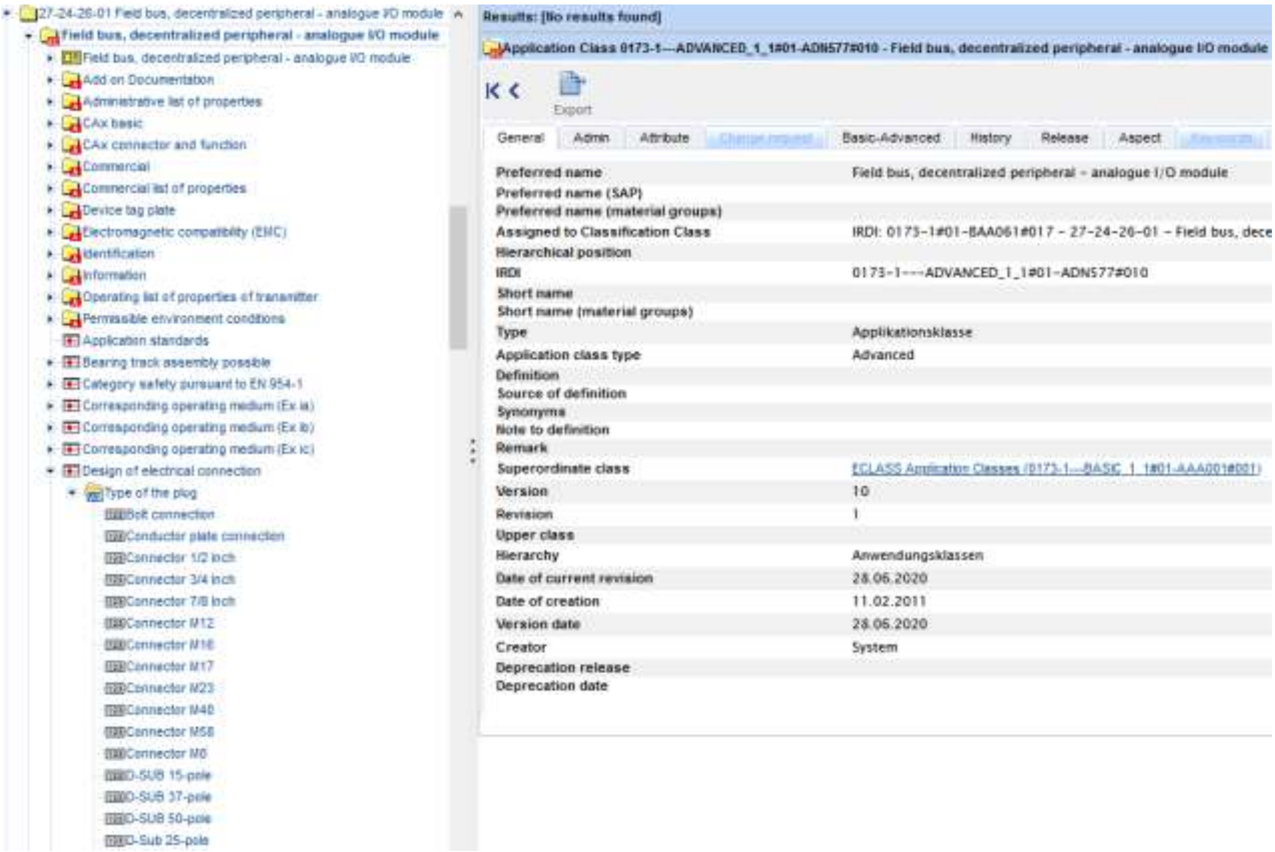


**Figure 8: ECLASS – example: fieldbus, analogue I/O module**

The figure above shows on the right side the definition and identification of the I/O module type. On the left side, the properties listed there are characterizing the I/O module.

We select the property "design of electrical connection", with its identifier 0173-1#02-BAD831#012.

 

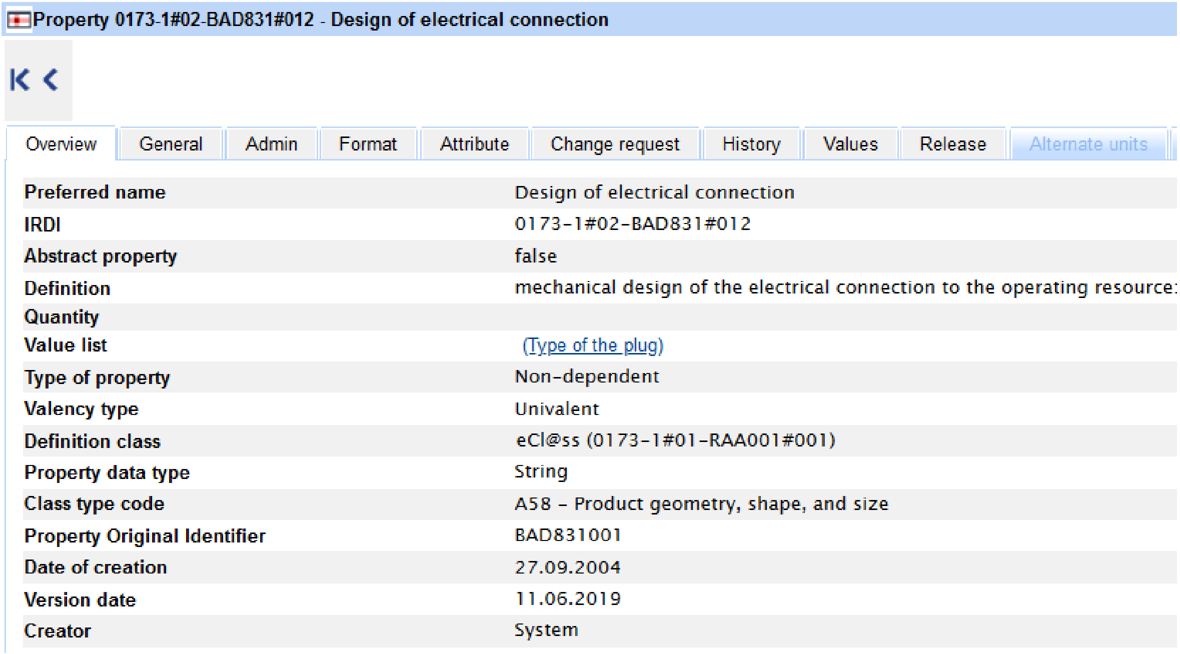


**Figure 9: ECLASS – example: Design of electrical connection**

This property has a value list “type of the plug”, containing in this case 39 different types of plugs (ECLASS11.1).

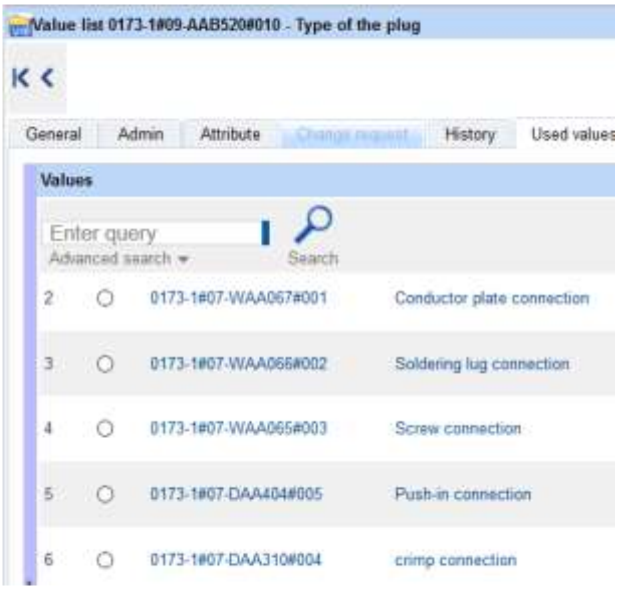


**Figure 10: ECLASS – example: excerpt of value list “type of the plug”**

This example shows how to characterize product features of an I/O module.

The same ideas is used to characterize variables (e.g., actual values, parameters or diagnosis information) of PI application profile compliant devices. This allows a technologic oriented standardized access to information.

Access by humans to ECLASS elements (via ECLASS internet page) is free of charge but a license is required to download ECLASS data structures, to assign data to ECLASS elements and to exchange data.

 

ECLASS and its service providers supply professional IT support, transaction- and version management and a fast-track procedure (ECLASS ACCELERATED) to get ECLASS elements (classes, properties, …) defined quickly.

## 6.2 IEC CDD

IEC Common Data Dictionary (IEC CDD) [105] is an International Standard (IEC 61360-4 DB) and serves as a common repository of concepts for industrial/technical domains. It is based on the methodology and the information model of IEC 61360 series, and provides [105]:

- unambiguous identification of classes and properties, and their relations
- commonly accepted terminology and definitions based on accepted sources such as IEC International Standards, other International Standards, Industry Standards, or Public Authorities
- hierarchies of concepts enabling users to appropriately characterize their products and services
- all kind of data element types and concepts like classes, properties, value lists, values, cardinality, polymorphism, blocks, conditions, qualifiers, conditions, constraints, relations, …
- technical representation of concepts including units and data types and their identification

IEC CDD actually supports the following domains:

- IEC 62683 (low voltage switch gears)
- IEC 61360 (electric / electronic components)
- IEC 61987 (process automation)

IEC 61987 covers process automation topics in detail. The following picture shows an overview of supported transmitters.

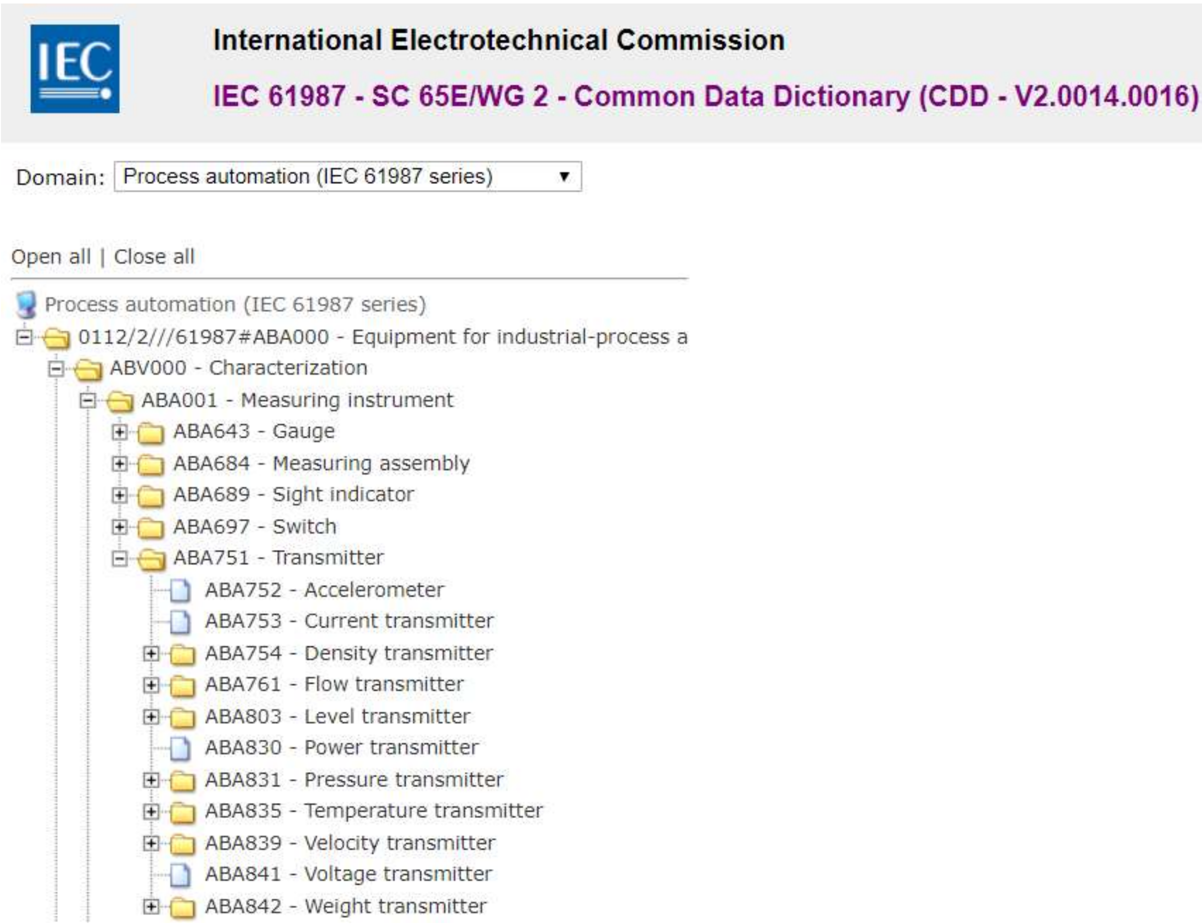


**Figure 11: IEC CDD – transmitter overview**

 

Unlike in ECLASS, IEC CDD already provides semantic definitions for device parameters, which in this case can be mapped to PA Profile [3]. This example shows the parameter low flow cut-off.

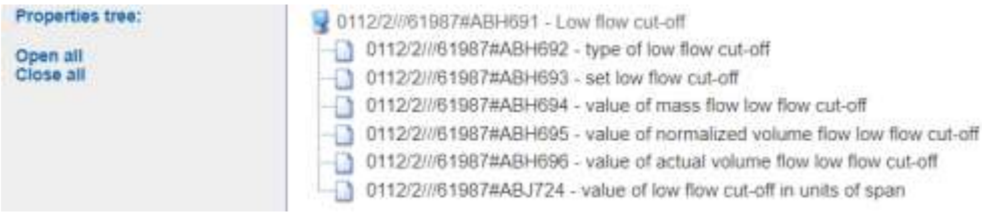


**Figure 12: IEC CDD – example property list**

The exact parameter definition is given in the following picture. The identifier (IRDI), the description and the unit in which the parameter value is shown.

PROPERTY

| | |
|---|---|
| Code: | 0112/2///61987#ABJ724 |
| Version: | 002 |
| Revision: | 01 |
| IRDI: | 0112/2///61987#ABJ724#002 |
| Preferred name: | value of low flow cut-off in units of span |
| Synonymous name: | |
| Symbol: | |
| Synonymous symbol: | |
| Short name: | Low flow cut off |
| Definition: | value of flow cut-off in units of span |
| Note: | If the span is not defined the range-limit is to be used instead. |
| Remark: | |
| Primary unit: | % |
| Alternative units: | |
| Level: | |
| Data type: | REAL_MEASURE_TYPE |
| Format: | |
| Definition source: | |
| Value source: | |
| Property data element type: | NON_DEPENDENT_P_DET |
| Drawing: | |
| Formula: | |
| Value list code: | |
| Value list: | |
| DET class: | |
| Applicable classes: | 0112/2///61987#ABH691 - Low flow cut-off |
| Definition class: | 0112/2///61987#ABA000 |
| Code for unit: | 0112/2///62720#UAA000 - percent |
| Codes for alternative units: | |
| Code for unit list: | 0112/2///61987#ABT533 - Percentage |
| Status level: | Standard |
| Published in: | IEC 61987 |

**Figure 13: IEC CDD – example property content**

 

Access by humans to IEC CDD (via internet page) is free of charge. IEC also offers so-called “free attributes” to enable electronic data exchange free of charge. Download and distribution of elements and concepts with the full set of attributes requires a license from IEC.

IEC SC3D offers support for IEC TCs/SCs to model and implement concepts in IEC CDD. IEC CDD offers version management for each and every data element (not for the total database). It covers only a few numbers of product types, but being an international standard, the definition depth is in parts deeper compared to ECLASS.

### 6.3 Harmonization IEC CDD – ECLASS

An international cooperation between the IEC and ECLASS e.V. was signed in 2016. Under this Cooperation Agreement the content for mapping belongs to

- IEC 61987, Industrial-process measurement and control - Data structures and elements in process equipment catalogues;
- IEC 61360, Standard data element types with associated classification scheme for electric components; and
- ECLASS Segment 27: Electric engineering, automation, process control engineering.

For the benefit of ECLASS and IEC CDD users, common content shall be identified and easily accessible in a harmonized form. Actual status see COMDO in section definitions.

It can be expected that for several standards (IEC, ISO, ECLASS) several identifiers (IRDI’s) will be defined for the same semantic topic.

### 6.4 International Registration Data Identifiers (IRDI)

IEC CDD and ECLASS both rely on the same basic definition of semantic identifiers, the IRDI.

The structure and syntax of IRDI is defined in ISO/TS 29002-5.

 

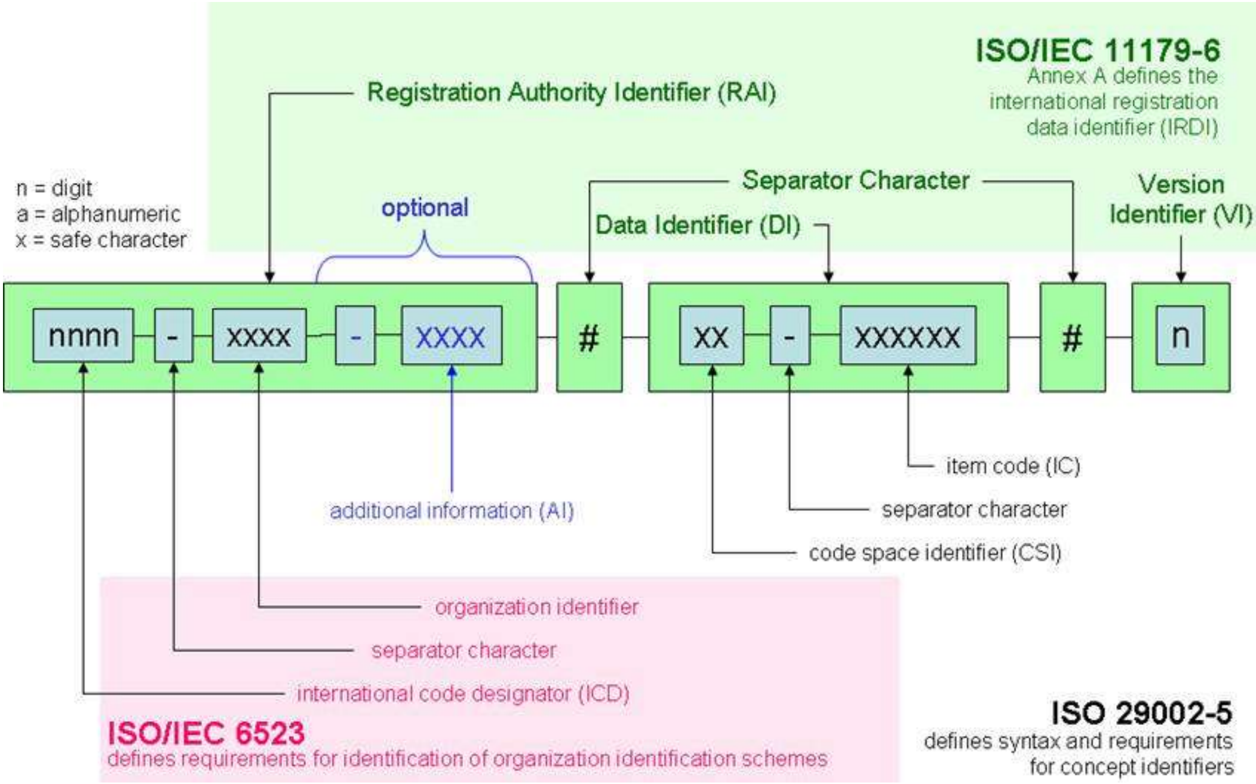


**Figure 14: IRDI structure [100]**

To reach the goal of globally unique identifiers each publishing organization defining IRDI's requests an ICD (International Code Designator) at the relevant registration authority to globally identify the organization and the OI (Organisation Identifier) if specified.

Examples: ECLASS: 0173-1; ISO: 0112/1; IEC: 0112/2

**Note**: As the separator is not specified in ISO/TS 29002-5 the organisations are using different separators. Separator examples: ECLASS: "-"; ISO and IEC: "/"

| Data element | IRDI |
|---|---|
| Dictionary: IEC CDD<br>Domain: Electric/electronic components (IEC 61360-4)<br>Data element: designation of IP protection | *0112/2///61360_4#**AAH011**#003* |
| Dictionary: IEC CDD<br>Domain: International Classification for Standards (ISO ICS)<br>Data element: Printed circuits and boards | *0112/1///ICS#**SAB156**#002* |
| Dictionary: ECLASS<br>Data element: degree of protection | *0173-1#02-**BAG975**#013* |

**Table 1: Examples for IRDI's**

The item code (IC) - marked bold in the table above - unambiguously identifies a specific data element in the dictionary.

One advantage of the IRDI is language neutrality. The name and the semantic definition of a data element can be translated in every language worldwide without changing the semantic meaning.

**Note**: IEC CDD uses the number of the standard (IEC or ISO) as an additional information (AI) to distinguish between different domains and their different semantic definitions.

 

## 6.5 Thing description

Schema.org [110] is a collaborative, community activity with a mission to create, maintain, and promote schemas for things (structured data) on the Internet, on web pages, in email messages, and beyond. Schema.org provides a collection of shared vocabularies webmasters can use to mark up their web pages in ways that can be understood by the major search engines: Google, Microsoft, Yandex and Yahoo!

Actually schema.org does not provide semantic definitions relevant for PI technologies.

## 6.6 IoT thing description

iotschema.org [111] is an extension of schema.org for Internet of Things. The goal is to enable web applications to interact with the physical world based on machine interpretable information. iotschema.org enables semantic interoperability for connected things across diverse IoT ecosystems.

The further development in this direction should be monitored carefully. PI shall not only rely on IRDI's but also have the possibility to support definitions like iotschema.org in future as well.

## 6.7 Summary

Actually, ECLASS is the most promising semantic organization to work with. Reasons are as follows:

- ECLASS today mainly supports properties for product catalogues. That means, ECLASS today already supports use cases UC1 and UC2
- Scope of ECLASS is extended towards Industrie 4.0 [101]. Industrie 4.0 uses ECLASS.
- Professional IT support
- Transaction update management of versions
- Well established version- and change management
- Has fast track procedure getting new properties defined
- Curated long-term specification
- Supports the full life cycle of property definition

For international applications other semantic database standards like IEC CDD needs to be considered.

# 7 Working model JWG PI – ECLASS

In the near future, PI and ECLASS e.V. expect a great value mapping semantically well-defined product classes / properties to PI technologies. Therefore, a joint working group was established to work out the technical details to support this mapping automatically executed by machines.

PI brings in the technological know-how of the devices used in automation, ECLASS adds it's semantic know how. This unique combination offers a significant step towards smart manufacturing and Industrie 4.0.

The working model is shown in the next figure.

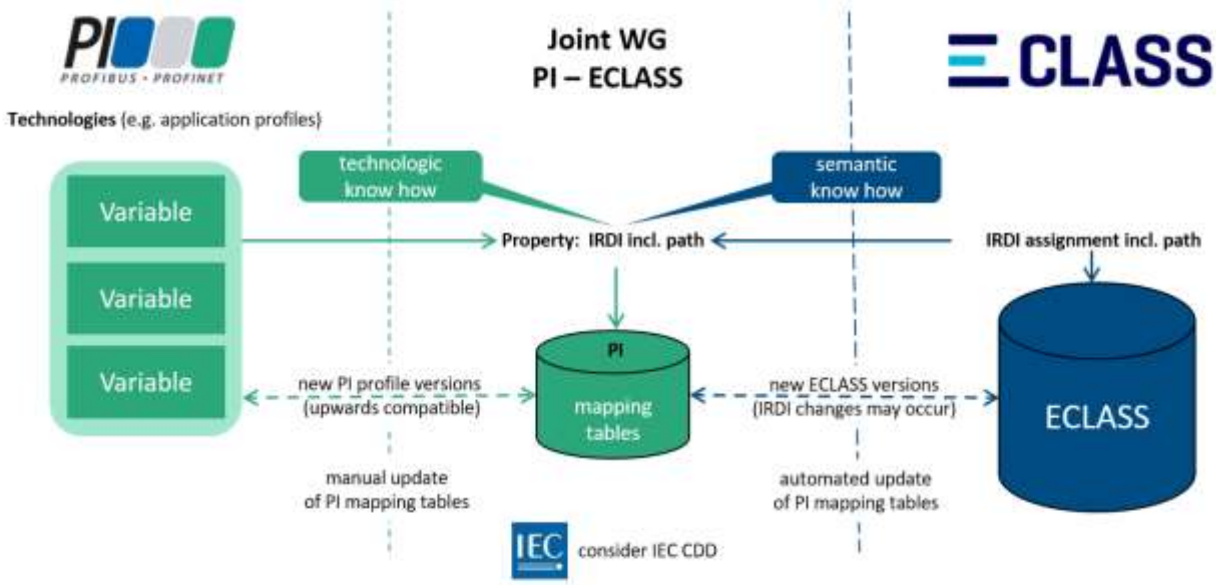


**Figure 15: Working model PI – ECLASS**

Goal of the JWG is to define fieldbus neutral technologic oriented semantic information.

## 7.1 Current status

At first, the discussions were focused on the semantic organizations ECLASS e.V. and IEC CDD. The chapter "Introduction Semantic Organizations" provides technical background, similarities and differences.

Second the JWG focused on the technical details, e.g., how to map data types and so on. The results of this discussion and technical solution proposals can be found in chapter "mapping model".

Finally, a meta model was built up how to add semantics to PI Application profiles resulting in a first version of an XML scheme for the mapping tables. This includes a conceptional verification of the XML scheme by considering several aspects of existing PI application profiles.

## 7.2 Proposed next steps

Define concrete mapping for selected PI application profiles and refine XML scheme if necessary.

 

# 8 Mapping model

The mapping model provides the necessary reference between

- the variable (content and access) defined by PI application profiles
- the technologic meaning by using semantic identifiers (e.g., IRDI paths) without knowledge of the underlying communication protocol

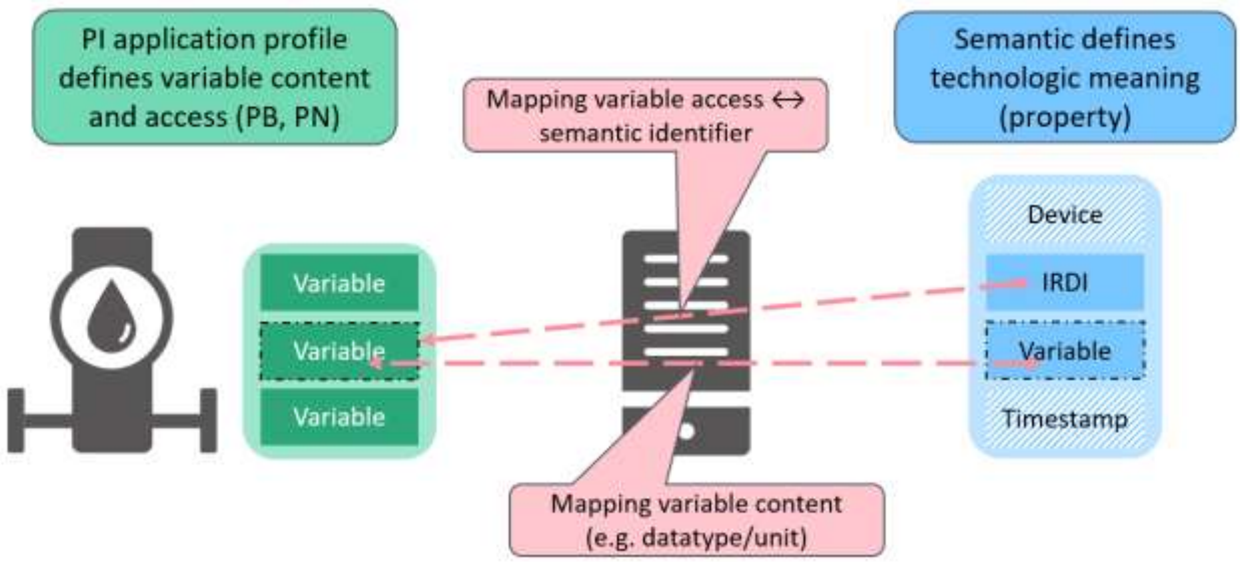


**Figure 16: Mapping model overview**

The mapping model covers the device types defined by PI application profiles only.

Concrete mapping implementations use instances of these device types. The figure above shows additional instance related information indicated by the dashed fields. This example contains e.g., the instance name of the device and a timestamp when the variable was read.

## 8.1 Mapping requirements

The mapping is an add on to a PI application profile. It is recommended not to reference semantic identifier directly in the application profile because semantic identifier may be changed asynchronously by semantic organizations.

There is no single PI mapping table. PI will provide mapping tables for each application profile or even for each device type defined within an application profile. This allows an incremental approach and later a flexible handling of updates.

To support different semantic organizations and their identifiers the mapping provides the possibility to use different types of identifier (e.g., IRDI) for the same variable.

PI application profiles support different underlying communication protocols like PROFINET or PROFIBUS and use specific methods to access variables. The mapping shall support different types of access methods for the same variable.

A variable consists of a value and for several data types of a unit (e.g., ˚C) and/or a status (e.g., good / bad). This fact shall be supported by the mapping.

 

PI application profiles often use tables in which a numeric value relates to a selection. Example is the unit code table provided by the PA Profile, the decimal value 1001 relates to ˚C (degree Celsius) and has the ECLASS IRDI 0173-1#05-AAA567#004. The mapping shall support the generic use of tables.

A property may be related (belong) to another property resulting in a series of variables. This is modeled by IRDI paths representing the series and shall be supported by the mapping. Example: One property damping belongs to mass flow signal, or the property damping belongs to density signal.

## 8.2 Preconditions for using the mapping

The use of the semantic mapping requires for the runtime use cases only some preconditions in the communication environment.

- A discovery mechanism used to identify the devices connected to the network. See [4].
- A communication connection to the devices is established
- To detect devices / slots following any PI application profile, the PROFILE_ID (and optional the PROFILE_SPECIFIC_TYPE) supported by I&M functions can be used. See [1] for further details.

Using this basic communication information, the semantic access to the devices' information is possible. The mapping table describes the link between semantic identifier (e.g., IRDI) and their associated access and data already standardized by PI application profiles.

## 8.3 Mapping of data types

PI application profiles support constrained devices. This is the reason why much more data types [2] than of other concepts are used to save device resources. Example: there are data types for integer with different lengths: int8, int16, int32 and int64.

Semantic definitions do not need this restriction. They typically support native data types e.g., the data type integer. Combining data type and data format, restrictions can be defined as well.

However, the different data types must be converted to ensure the right values to be used. There is almost no conversion problem reading information from a device. Writing to a device requires an online check whether the given value from outside can be correctly transferred to the device or not.

The mapping table provides the data type used by the device to perform this check.

**Hint 1:** There is no unsigned data type in ECLASS yet. Reading may cause problems when data type unsigned64 is used and the values exceed $2^{32}$. In this case a wrong negative value is assumed.

**Hint 2:** PI data types use big endian coding. Typically, PC data types use little endian coding. In case of need, the byte order has to be exchanged. See [2] for further information.

## 8.4 Mapping units

Units are also coded as IRDI's. IEC/TS 62720 "Identification of units of measurement for computer-based processing" provides the definition for IEC CDD. ECLASS provides own IRDI's based on UnitsML. There is a 1:1 relationship between units for IEC and units for ECLASS.

 


**Figure 17: Example unit nanofarad from IEC TS 62720**

For practical reasons, ECLASS assumes units to be fixed. This means the version information will not be used, especially for ECLASS updates.

Depending on the definition of the PI application profile, a unit of a variable can be fixed or read / written from / to the device. Due to constrained resources, the PI devices deliver a coded number instead of a string or IRDI.

| Value | Symbol | Description | Equivalence |
|---|---|---|---|
| 1244 | µV | microvolt | = $10^{-6}$ V |
| 1245 | F | farad | = 1 C/V |
| 1246 | mF | millifarad | = $10^{-3}$ F |
| 1247 | µF | microfarad | = $10^{-6}$ F |
| 1248 | nF | nanofarad | = $10^{-9}$ F |
| 1249 | pF | picofarad | = $10^{-12}$ F |
| 1250 | F/m | farad per meter | |
| 1251 | µF/m | microfarad per meter | |

**Figure 18: Example unit code nanofarad from PA Profile**

In this example, the coded number for unit “nanofarad” is 1248. The IRDI for “nanofarad” is 0112/2///62720#UAA903#001. A special transformation table for units is necessary to automatically map coded number for unit to IRDI.

**Hint:** PI needs different unit transformation tables depending on the different application profiles supported.

**Hint**: The transformation tables should support ECLASS and IEC CDD identifiers for units.

## 8.5 Mapping using transformations

Constrained devices often do not support strings directly. Therefore, often lists with value – string pairs are defined in PI Application profiles. Examples:

- Manufacturer ID – company name https://www.profibus.com/IM/Man_ID_Table.xml

 

These lists shall be converted in machine readable format to perform the automated transformation from value to string.

The second type of lists defines triples: value – string – IRDI’s. Example:

- Type of linearization<br>PA Profile 4.0 value 119 string RTD Ni100 IRDI (CDD) ABB088/ABK987

- ECLASS (AAT180 defines type of RTD sensor)<br>PA Profile 4.0 value 119 string Ni100 IRDI (ECLASS) 07-AAX026

**Hint**: in contradiction to units, transformations in general are version depended.

## 8.6 Proof of concept: electronic faceplate

PROFINET and PROFIBUS DP V1 devices support the PI wide standardized identification and maintenance data (I&M).

One part of the I&M data is the electronic faceplate. The following table shows the mapping by adding the ECLASS semantic identifiers and their data types.

| PROFINET / PROFIBUS | Offset, length | Data type | ECLASS device identification | ECLASS IRDI path | ECLASS data type |
|---|---|---|---|---|---|
| MANUFACTURER_ID | 0, 2 | unsigned 16 | Manufacturer name | /:ADN228/AAQ373/AAO677 | String_ translatable |
| ORDER_ID | 2, 22 | visible string 20 | Product article number of manufacturer | /:ADN228/AAQ373/AAO676 | String_ translatable |
| SERIAL_NUMBER | 22, 38 | visible string 16 | Serial number | /:ADN228/AAQ373/AAM556 | String_ translatable |
| HW Revision | 38, 54 | visible string 16 | Hardware version | /:ADN331/AAN270 | String_ translatable |
| SW Revision | 54, 70 | visible string 16 | Firmware version | /:ADN331/AAM985 | String_ translatable |

**Table 2: Mapping electronic faceplate**

This proof of concept shows the general idea works. Using an ECLASS IRDI can be mapped to a standardized data access in a device.

This example also shows the use of transformations. The MANUFACTURER_ID represents the name of the manufacturer. PI provides a list transforming ID’s to name, see https://www.profibus.com/IM/Man_ID_Table.xml.

**Note:** Keep in mind that the transformation Manufacturer → name to ID is not 1:1. There might be several entries for one manufacturer due to mergers and acquisitions over time.

 

## 8.7 Verification of mapping using PA Profile V 4.0

To verify this general concept, we used the PI application profile “PA Profile” V4.0 [3]. One reason is being a real complex example, the other reason is that there are already IRDI definitions by IEC CDD available.

### 8.7.1 Support different device types within a profile

PA Profile [3] defines different types of devices (so called transducer blocks) for e.g., flow or temperature transmitter. Interesting problem here is that a flow transducer block also supports a temperature but the address is different (slot 3 vs. slot 1) compared with temperature transducer block.

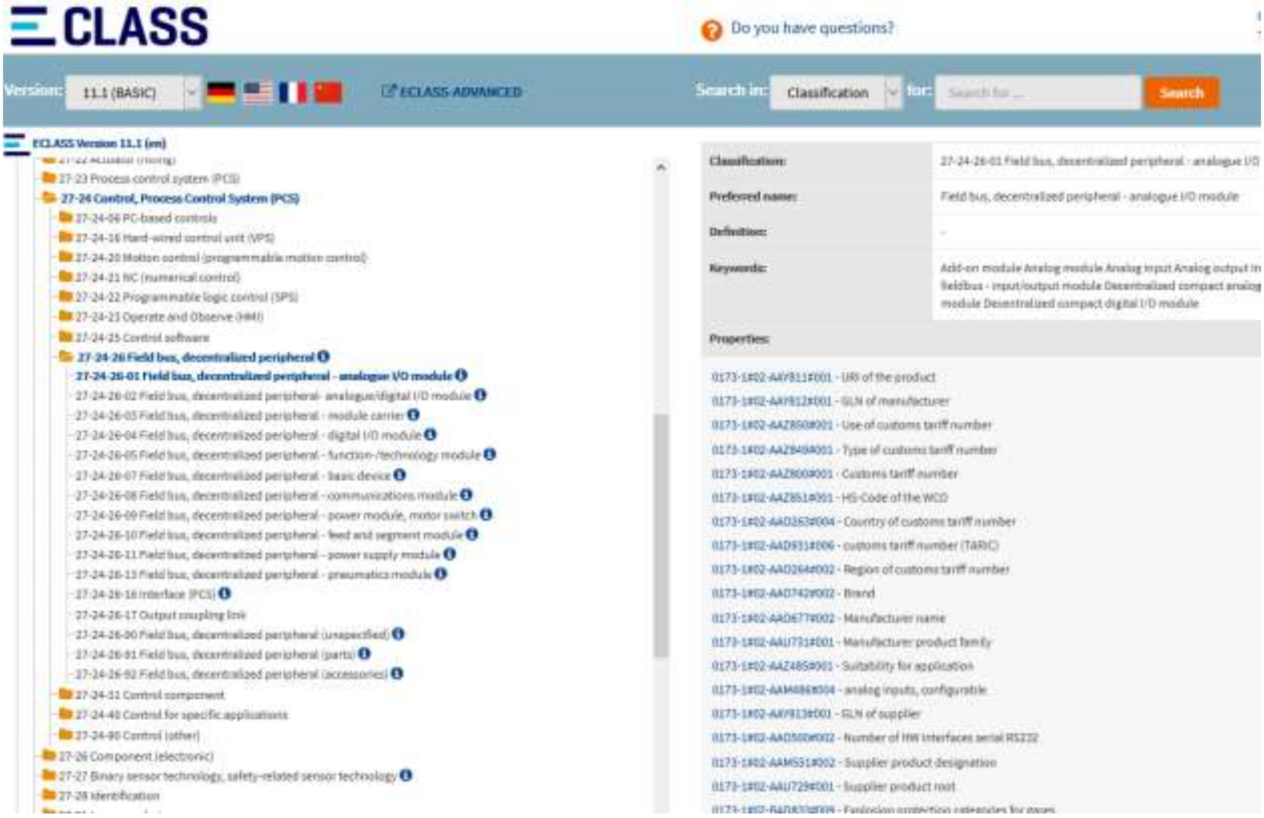


**Figure 19: Example different ECLASS device types**

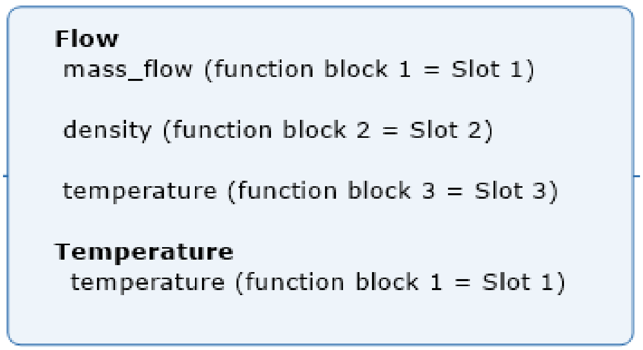


**Figure 20: Example different PA Profile device types**

**Solution:**

The mapping file must be specific for the device type used. That means, there is a mapping table for the device type flow transmitter and another mapping table for temperature. PA profile e.g., specifies device type with the help of online readable information. To select the right file, the file name must include such device type information.

### 8.7.2 Variable names are used multiple times

In the example shown, the variable damping is used several times. In fact, the device supports for each primary variable (e.g., mass flow) an own value for damping. This requires that not only the variable but also the context is necessary.

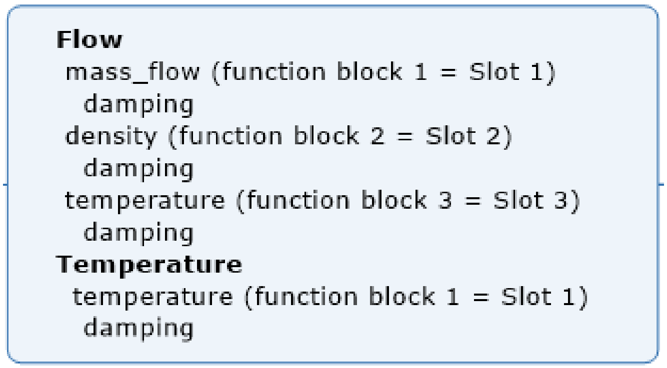


**Figure 21: Example context**

**Solution:**

Instead using the IRDI for damping ABH526 (from CDD) the IRDI path which indicates this is the damping for e.g., the variable mass flow shall be used. This link shall be explicitly modelled within the mapping file to avoid time and processing power consuming searches.

### 8.7.3 Unit Codes need transformation

The chapter “Mapping Units” already describes the principle. A transformation is used to convert PI application profile unit codes to IRDI’s and vice versa.

### 8.7.4 Shown display language

PA devices support different languages on their local display. ECLASS supports a list of all supported display languages (AAT401). There is up to now no ECLASS identifier which language is actually shown on the display.

**Solution:**
Add a new IRDI in ECLASS.

### 8.7.5 Different types of linearization

PA Profile [3] Table 240 provides a list “Type of linearization”, supporting e.g., different temperature sensor types (e.g., PT 100, Ni 100).

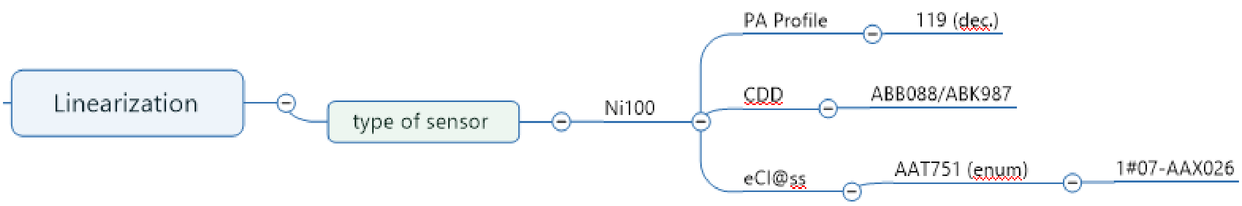


**Figure 22: Example linearization**

 


**Figure 23: Example linearization IEC CDD**

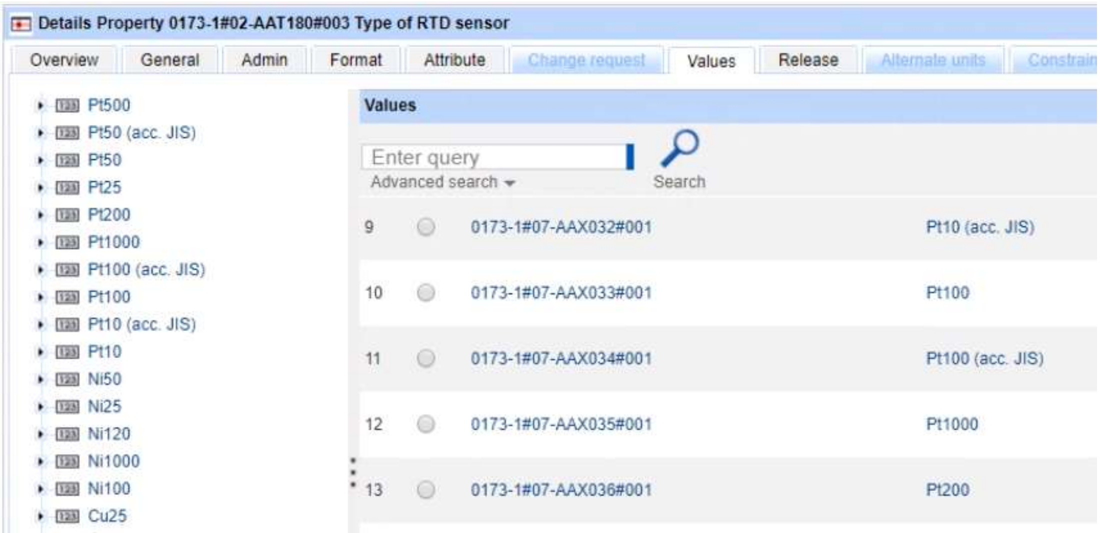


**Figure 24: Example linearization ECLASS**

**Solution:**

Use triple transformation: **value** (from PI application profile), **string** (human readable), **IRDI** (from ECLASS, IEC CDD)

**Hint**: Linearization is version dependent

### 8.7.6 Conclusion

From theoretical point of view, the semantic mapping of PA Profile with all its specifics is possible.

 

# 9 Recommended Actions

The recommended actions base on the discussions in the JWG PI – ECLASS. They will be extended or refined if needed.

## 9.1 ECLASS

Delete PROLIST information within ECLASS. In ECLASS10.1 the PROLIST information is already signed as “phased out”.

Build up a definition which focusses on technology properties for PA devices only and align to IEC CDD 61987-Series). Keep this definition as slim as possible. Do not introduce any communication or fieldbus specific properties / information. Incorporate results from project COMDO.

Continue already started activities on ECLASS – OWL mapping to enable use of W3C semantic tools (e.g., search engines) together with ECLASS semantic definitions.

Additional tool support is necessary: Simple export of complete path for use in PI mapping tables. This path shall be the same path as used in TUF file for ECLASS version updates. Support also value lists with its values.

## 9.2 PI

Today not all data provided by PI application profiles already have such a semantic property defined, it is future work to provide a complete set of information.

Up to the PG’s to

- Select parameter / values defined by profile to be extended with semantic identifiers
- Support ECLASS IRDI’s and other semantic identifiers (e.g., IEC CDD IRDI’s) as well
- Provide XML file(s) for mapping
- Support ECLASS TUF update mechanisms

## 9.3 JWG PI – ECLASS

This white paper describes the results of the JWG PI – ECLASS so far. Future work:

- Refine the meta model (and XML mapping file) when necessary
- Give support and advice to PI PG’s working on semantic definitions for application profiles and beyond
- Evaluate PI goal to shorten IRDI paths vs. TUF update file

Support ECLASS version updates by providing updated mapping tables. Use ECLASS beta versions to prepare XML files in advance for testing purposes.

 

# 10 Annex: XML Schema for Mapping table

The mapping table models how to access to information provided by PI application profiles using a given semantic identifier (e.g., IRDI path). The complete property definition of semantic organisations like ECLASS will not be part of the mapping table in order to avoid inconsistencies or double definitions.

This XML schema can be found in the attachments to this white paper, file name “PI_ECLASS_SemanticMapping_1.0.xsd”.

The XML schema contains type information only. The number of instances needed depends on the number of connected devices.

## 10.1 Overview

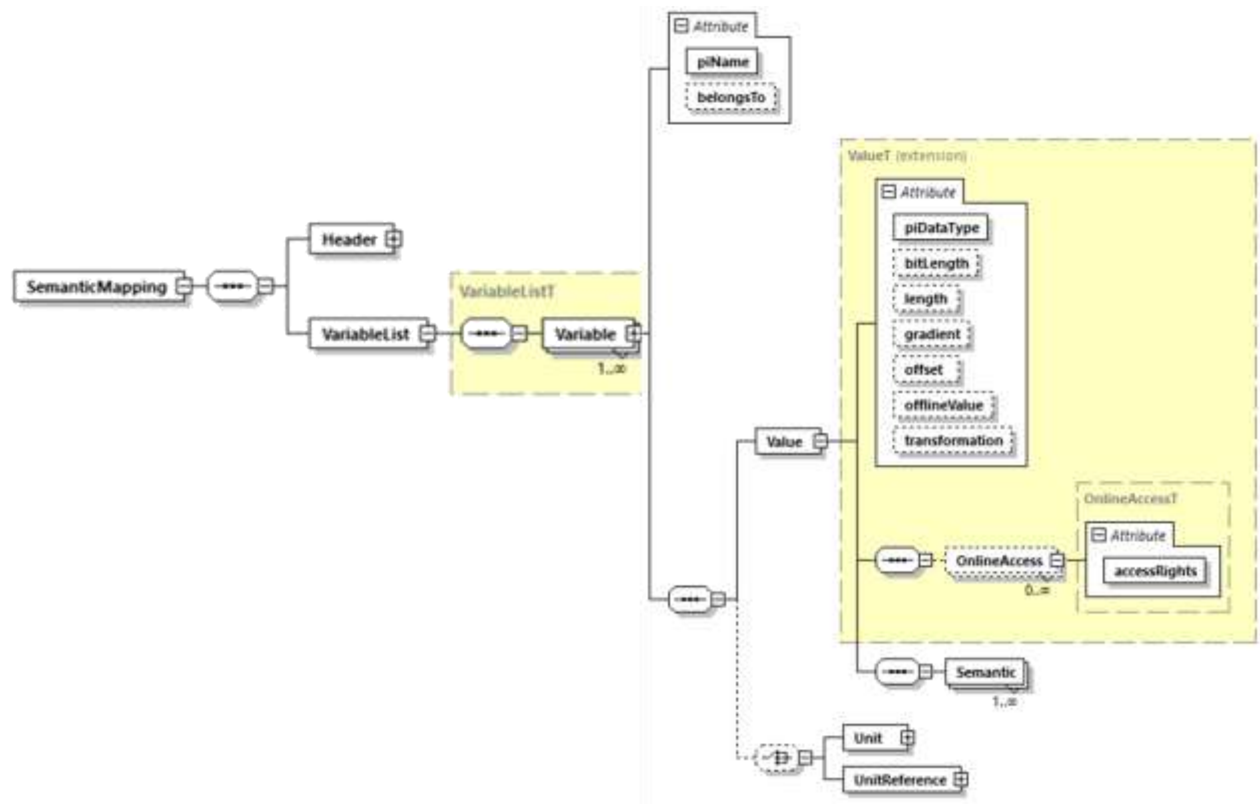


**Figure 25: Overview XML mapping table**

The *SemanticMapping* XML file consists of one *Header* and one *VariableList* for each device type.

The *VariableList* contains all *Variables* (e.g. PI data record elements) of this device type for which a semantic definition is needed.

Each *Variable* has a *Value* and optional a *Unit* or *UnitReference*.

The *Value* is defined in the PI application profile with its piName, meaning, data type, online access, … . The *Semantic* identifiers (e.g. IRDI path) are added. There may be different *Semantic* identifiers from different semantic organizations or versions.

Beside the *Value* itself, the *Unit* is necessary (e.g. length 10 mm) to describe the complete information. The *Unit* may also be read from the device (like the *Value*) and has also a *Semantic* identifier. Alternatively, *UnitReference* refers to the unit of another *Variable*.

 

## 10.2 XML filename

The XML filename for a specific device type enables any mapping application to determine the right mapping table to be used for the connected device. The filename shall fulfill the following conventions

PI_SEM1.0-<profileID>-<profileSubID>-<releaseDate>

**PI_SEM1.0**
Indicates a PI semantic mapping table with the used schema version

**<profileID>**
Indicates the PI application profile e.g., 0x9700 for PA Profile.

**<profileSubID>**
Indicates different device types supported by PI application profile

**<releaseDate>**
Indicates the date of the mapping table released. Format is yyyymmdd

Background: The online detection of devices delivers exactly the information, which profile and profile sub type is supported. This can be easily used to open the corresponding XML file.

This definition supports the following Requirements

- Machine readable
- no additional information (see XML Header inside the file)
- global unique file names (one directory for all mapping files))
- contains version of XML scheme to optimize file open with the right scheme

**Hints**:

- For some PI profiles a specific logic is needed to determine the exact type of the device.
- IO-Link is currently not supported. Reason: One IO-Link device may support more than one profile ID.

## 10.3 Header

The *Header* within the XML file supports the same information that is given in the filename. This is intended to doublecheck eventual misuse of windows rename function.

 

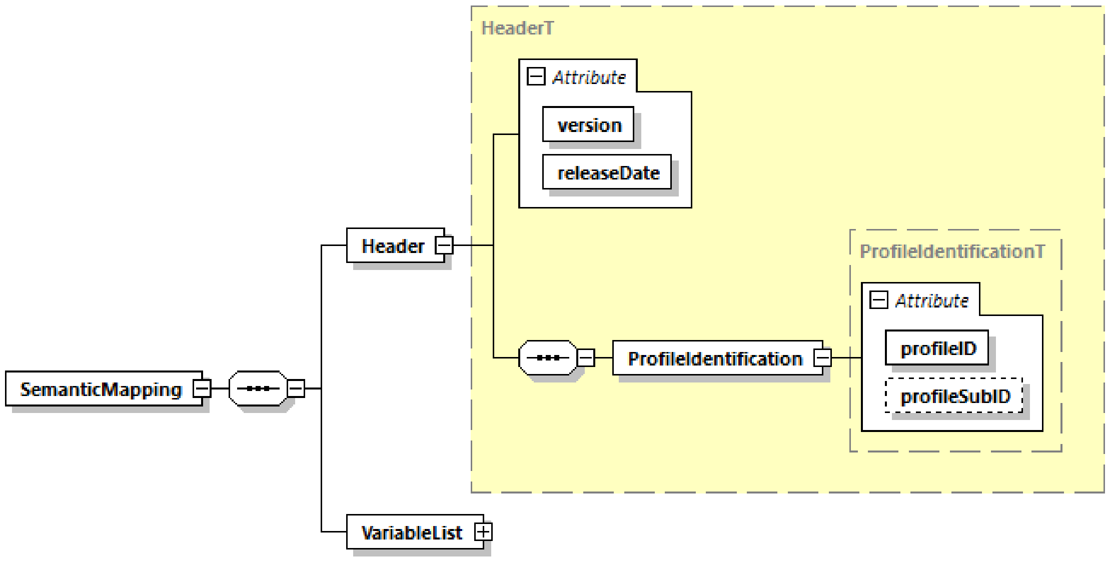


**Figure 26: XML Header**

**Version**
Version of this XML schema for *SemanticMapping*

**ReleaseDate**
release date of the XML mapping file for the specific device type supported

**ProfileID**
Profile ID defined by PI for each application profile, see I&M0 data [https://www.profibus.com/IM/Profile_ID_Table.xml](https://www.profibus.com/IM/Profile_ID_Table.xml)

**ProfileSubID (optional)**
Specifies sub types of devices, see I&M0 data Profile Specific Type or IdentNumber

## 10.4 VariableList

The *VariableList* contains all *Variables* (e.g. PI data record elements) of this device type needing a semantic definition.

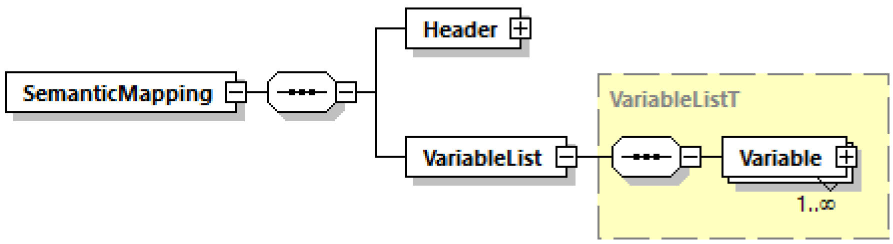


**Figure 27: XML Variable List**

**VariableList**
Container for all *Variables* defined

### 10.4.1 Variable

Each *Variable* has a *Value* and optional a *Unit* or *UnitReference*.

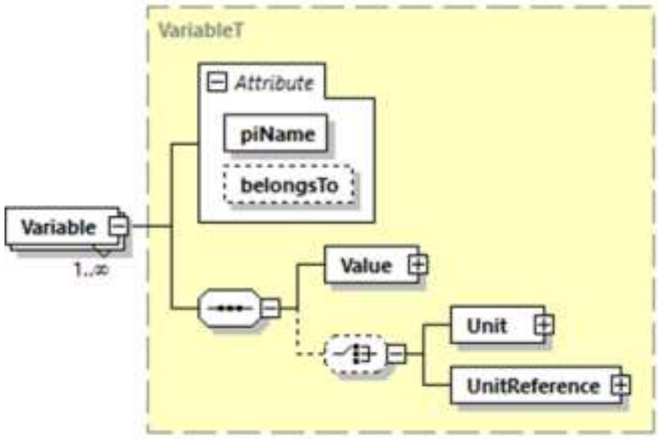


**Figure 28: XML Variable**

**piName**
name of the *Variable* from PI application profile

**belongsTo (optional)**
Link to another *Variable*. Can be used e.g., for damping as reference to primary variable e.g., mass flow.

### 10.4.1.1 Value

The *Value* is defined in the PI application profile. The textual references to the data type, the online access (for e.g., PROFINET and / or PROFIBUS), the referring *Unit* etc. and the *Semantic* identifier are modelled for computerized use.

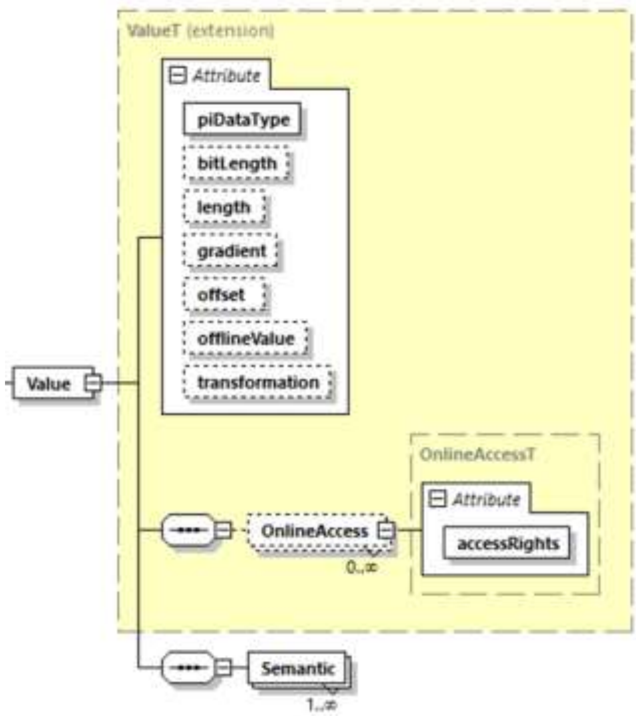


**Figure 29: XML Value**

**piDataType**
data type of the *Value* defined by PI application profile

**bitLength (optional)**
length of bits of PI data type. Only needed if not aligned to byte boundaries

**length (optional)**
length of octets of PI data type

**gradient / offset (optional)**
enables linearization of the V*alue* if needed

**offlineValue (optional)**
predefined *Value* by PI application profile

**transformation (optional)**
Reference to a transformation with a keyword having input and output parameters. Each transformation has to be coded within the application. This is used for

- coding /decoding a list with “value ↔ string” tupels. E.g., useful for decoding manufacturer ID’s to manufacturer names. This transformation could be indicated with the keyword “Manufacturer List”

- containing business logic. E.g. for IO-Link, the MinCycleTime ranges from 0.4 to 132.8 milliseconds with varying resolution, and is encoded in a single byte. It uses 6 bits multiplicator and 2 bits time base encoding, as specified in the IO-Link Interface and System Specification. This transformation could be indicated with the keyword “IO-Link CycleTime”

#### 10.4.1.2 Online access (optional)

Depending on the fieldbus and the application profile, different types to access the data from the device are needed.

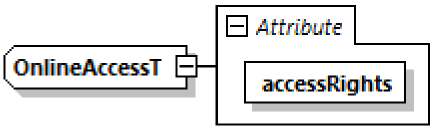


**Figure 30: XML OnlineAccessT**

**accessRights**
*Value* may be read only, read/write or write only. Defined by PI specifications.

The standard access types for PI fieldbus communication technologies have to be coded within the application using the following definitions:

- PROFINET: *PNOnlineAccessT* – native access type to data

- PROFIBUS: *PBOnlineAccessT* – native access type to data

- PROFIBUS DS 255 call mechanism (see [1]): *PBCall255OnlineAccessT* – access type for e.g. I&M data

The XML model allows to add more than one *OnlineAccessT* for a *Value*, supporting the different PI fieldbus systems.

These types can be extended for e.g.:

- PROFIdrive: PDOnlineAccessT and PDOnlineAccessUnit

- PA Profile V3: PA3OnlineAccessT

#### 10.4.1.3 Online access PROFINET (optional)

Native data access type for PROFINET: *PNOnlineAccessT*

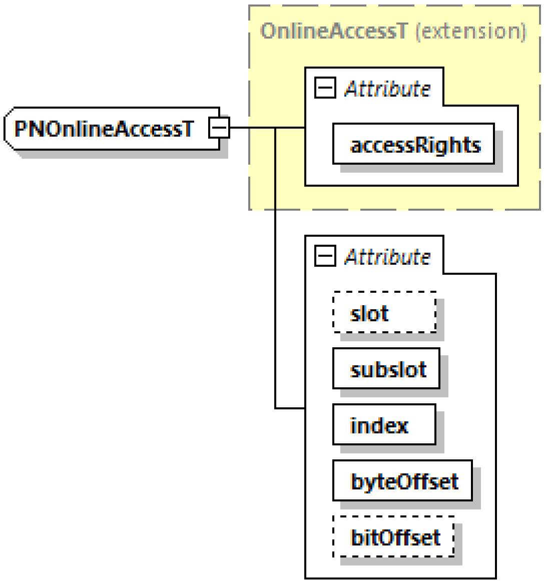


**Figure 31: XML PNOnlineAccessT**

**slot (optional)**
the *slot* within the device. Defined by PI specifications. Needed e.g., for PA devices

**subslot**
the *subslot* to address within the device. Defined by PI specifications

**index**
the record number within the *subslot* to address. Defined by PI specifications

**byteOffset**
the byte offset within the record to address the value. Defined by PI specifications

**bitOffset (optional)**
The bit offset within the addressed byte. Defined by PI specifications

#### 10.4.1.4 Online access PROFIBUS (optional)

Native data access type for PROFIBUS: *PBOnlineAccessT*

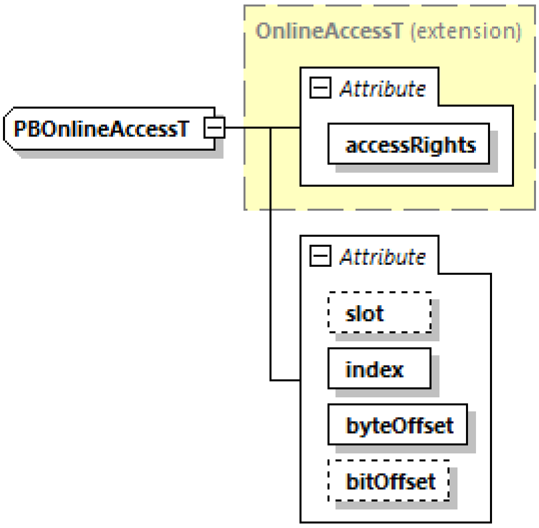


**Figure 32: XML PBOnlineAccessT**

**slot (optional)**
the *slot* within the device to address. Defined by PI specifications

**index**
the record number within the *slot* to address. Defined by PI specifications

**byteOffset**
the byte offset within the record to address the *Value*. Defined by PI specifications

**bitOffset (optional)**
The bit offset within the addressed byte. Defined by PI specifications

#### 10.4.1.5 Online access PROFIBUS Call 255 mechanism (optional)

Access type to data using PROFIBUS DS 255 call mechanism (see [1]). Needed e.g., for I&M data.

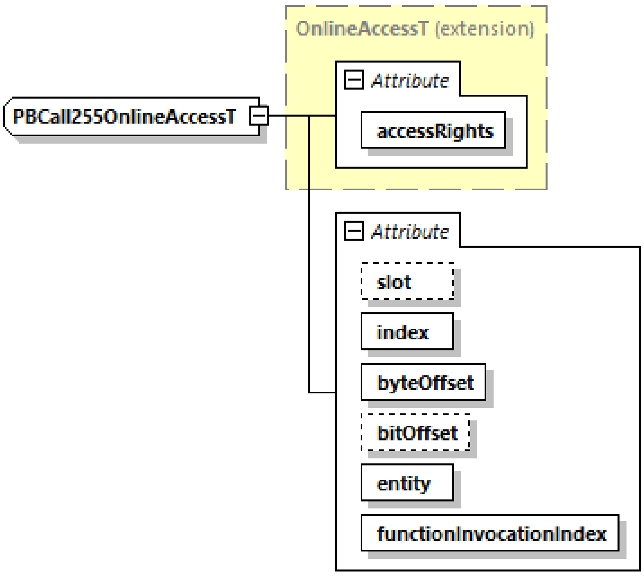


**Figure 33: XML PBCall255OnlineAccessT**

 

**slot (optional)**
the *slot* within the device to address

**index**
the record number within the *slot* to address

**byteOffset**
the byte offset within the record to address the *Value*

**bitOffset (optional)**
The bit offset within the addressed byte

**entity**
The sub-component within a *slot*

**functionInvocationIndex**
the communication service to use. See [1], table “Services of the Load Region state machine”

#### 10.4.1.6 Semantic

For each *Value*, at least one semantic description is defined.

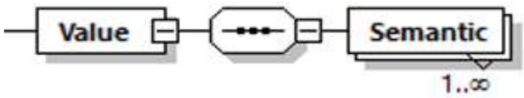


**Figure 34: XML Semantic container**

**Semantic**
container for all types of semantic identifier

There can be different types of semantic identifiers, supporting e.g., ECLASS or IEC CDD at the same time. In this version of XML semantic mapping, ECLASS and IEC CDD are different types to distinguish between both, but the layout of the content is the same.

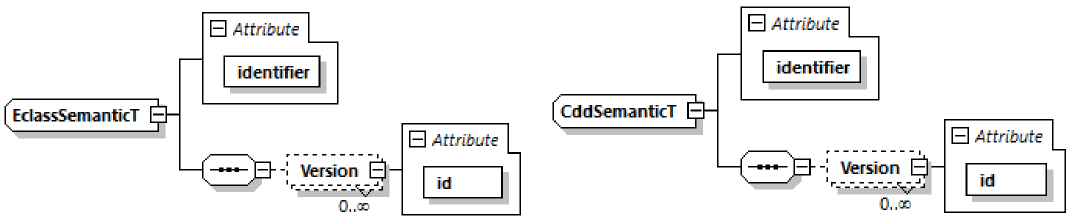


**Figure 35: XML SemanticT**

**identifier**
the semantic identifier defined by the appropriate semantic organization. The IRDI path is included.

**Version (optional)**
container for all versions of the semantic organization the *identifier* is valid.

**id**
defines the version of the originator of the semantic *identifier*. E. g. ECLASS11.1

 

### 10.4.2 Unit and UnitReference

The *Unit* of a V*ariable* can be fixed or read / written from the device, depending on the definition of the PI specifications. Therefore, *Unit* is modeled in the same way as *Value*, with data types and *onlineAccessT*.

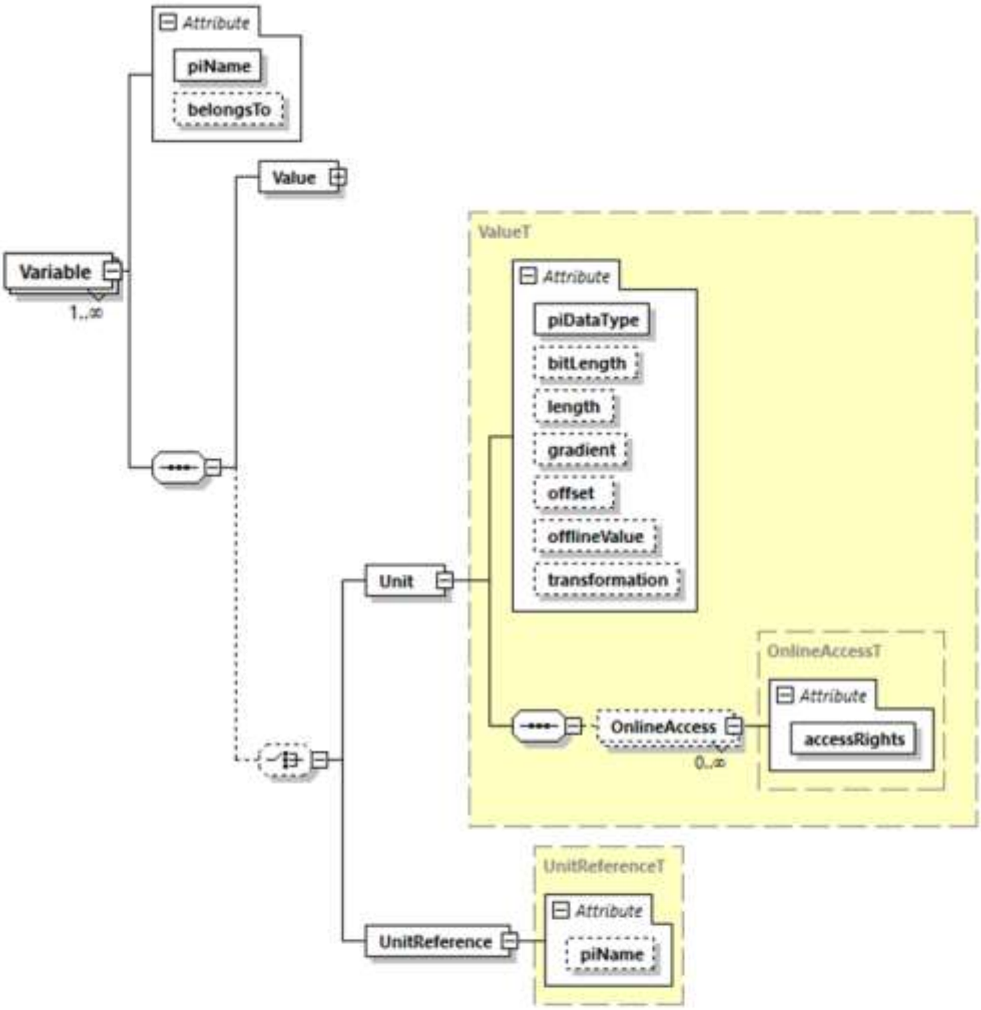


**Figure 36: XML Unit and UnitReference**

Due to constrained resources, the PI devices deliver a coded number instead of a string or IRDI for units. A special *transformation* is needed to get the IRDI of the unit.

**transformation**
coding / decoding a list with "value ↔ string" tupels. In this case, a table with the coded number of the unit, the ECLASS IRDI and the CDD IRDI could be provided. This transformation could be indicated with the keyword "Unit List"

**UnitReference**
A *Value* may have a *Unit* or alternatively a *UnitReference*. The *UnitReference* supports the fact that not every *Value* has an own *Unit* defined. For example, a process value for pressure defines the unit itself, but upper and lower pressure limits (being also independent values with own semantics) always use the same unit definition as the process value.

**piName**
reference to the *Variable* from which the *Unit* has to be used

 

PROFIBUS Nutzerorganisation e. V. (PNO)
PROFIBUS & PROFINET International (PI)
Haid-und-Neu-Str. 7 ● 76131 Karlsruhe ● Germany
Phone +49 721 96 58 590 ● Fax +49 721 96 58 589
E-mail info@profibus.com
www.profibus.com ● www.profinet.com