|  |  |  |
| --- | --- | --- |
| Graphical user interface, website  Description automatically generated | | |
|  | | |
| Document title: | | |
| **NOAKA DEXPI Transfer Requirements** | | |
| Document no.: | Rev.: | Page: |
|  | **3.8** | **1 of 83** |

NOAKA DEXPI Transfer Requirements

Prepared forNOAKA DEXPI Pilot Group

*Prepared by*

Pedersen, Tonia

**Revision and Signoff Sheet**

**Change Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Reason for issue, key changes and or decisions** | Prepared |
| 0.1.1 |  | Initial | TLP |
| 0.1.2 | 28.10.21 | Updated after comments from Manfred and added ItemTag attribute for all types other than Equipment. | TLP |
| 1.0 | 02.11.21 | Update to include equipment box definitions | TLP |
| 2.0 | 29.11.21 | Equipment box not required to transfer. Updates to OPC and slope sections. | TLP |
| 2.1 | 01.12.21 | Pipe graphical model, Flow Arrow & Virtual Nozzle. | TLP |
| 2.2 | 10.12.21 | Actuated valve example | TLP |
| 2.3 | 14.12.21 | Update metadata example | TLP |
| 2.4 | 17.12.21 | Shutdown and alarm models added | TLP |
| 2.5 | 19.01.22 | Added in-line instrument and updated instrumentation section headers, heat tracing & connected equipment details. | TLP |
| 3.0 | 26.01.22 | A number of updates to be in alignment with symbol legend mapping to DEXPI | TLP |
| 3.1 | 31.01.22 | Added Loop number attribute to SafetyValve type.  Added C01\_NOA1 example. | TLP |
| 3.2 | 22.02.2022 | Valve attribute transfer update to support label definition. | TLP |
| 3.3.2 | 29.03.2022 | Move attributes from PipingNetworkSystem to PipingNetworkSegment. Proteus definition for rotation/mirroring. Add extra shut-down to example. Fix attribute names. Added Proteus example for label symbolreference. | TLP |
| 3.4 | 07.04.2022 | Added InstrumentLoopNumber to InLinePrimaryElement class type and updated Signal Off Page Connection sections. | TLP |
| 3.5 | 28.04.2022 | Added Label section, actuator label details and added Proteus example for piping name label. | TLP |
| 3.6 | 24.05.2022 | Update Note section based on decision to use NOAKA DEXPI note class, Add/update Valve label section, radioactive symbol, ProcessPlant. Persistent ID section is still outstanding. | TLP |
| 3.62 | 01.06.2022 | Update attribute details for PipeReducer, radioactive symbol model | TLP |
| 3.7 | 07.07.2022 | Update file metadata, Persistent ID, Instrumentation (off-line section) - additional example for off-line instrument connected to pipe. New attribute HeatTraceRequirement for PipingLineSystem. New Annex H. Fix symbol rotation example. | TLP |
| 3.8 | 25.09.2022 | Update approved Notes, model for off-line instrument, propertybreak and SetPoint. | TLP |
|  |  |  |  |
|  |  |  |  |

**Technical Advisor**

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **Email** |
| Manfred Theißen | DEXPI technical resource | theissen@plants-and-bytes.de |

**Table of Contents**

[NOAKA DEXPI Transfer Requirements 2](#_Toc115013603)

[Change Record 3](#_Toc115013604)

[Table of Contents 5](#_Toc115013605)

[Overview 7](#_Toc115013606)

[Purpose 7](#_Toc115013607)

[Abbreviations 7](#_Toc115013608)

[References 7](#_Toc115013609)

[Constraints/Assumptions 8](#_Toc115013610)

[Process and Instrumentation Diagram Transfer 9](#_Toc115013611)

[Overview 9](#_Toc115013612)

[P&ID Transfer Breakdown Grouping 9](#_Toc115013613)

[P&ID Example C01\_NOA1 June MileStone Example 10](#_Toc115013614)

[DEXPI Standard 11](#_Toc115013615)

[Analysis: 11](#_Toc115013616)

[Proteus Schema 11](#_Toc115013617)

[Analysis 11](#_Toc115013618)

[Transfer Requirements 12](#_Toc115013619)

[Symbols 13](#_Toc115013620)

[Persistent ID 17](#_Toc115013621)

[File & Drawing Metadata 19](#_Toc115013622)

[Process Plant, Plant System & Plant Area 22](#_Toc115013623)

[Equipment 24](#_Toc115013624)

[Piping 26](#_Toc115013625)

[Instrumentation 33](#_Toc115013626)

[Equipment Box 50](#_Toc115013627)

[Equipment Internals 52](#_Toc115013628)

[Labels 53](#_Toc115013629)

[Note 54](#_Toc115013630)

[Miscellaneous Graphics with Symbol Reference 57](#_Toc115013631)

[ANNEX A: Custom class definitions - Non Graphical Elements 70](#_Toc115013632)

[Virtual Piping Connector 70](#_Toc115013633)

[Requirement Details: 70](#_Toc115013634)

[Virtual Instrumentation Connector 71](#_Toc115013635)

[Requirement Details: 71](#_Toc115013636)

[ANNEX B: Custom attribute defintions 72](#_Toc115013637)

[ANNEX C: Draft update ‘P&ID Profile file specification 3.3.3’ 73](#_Toc115013638)

[ANNEX D: Symbol Rotation and mirroring example 75](#_Toc115013639)

[Rotation and Mirroring in DEXPI and Proteus 76](#_Toc115013640)

[DEXPI 76](#_Toc115013641)

[Proteus 76](#_Toc115013642)

[ANNEX E: Signal conveying Line Types 77](#_Toc115013643)

[ANNEX F: Valve Label Details 78](#_Toc115013644)

[ANNEX G: NOAKA DEXPI Notes Implementation 79](#_Toc115013645)

[ANNEX H: DEXPI Connection Points & Associations 80](#_Toc115013646)

[Piping Connector 80](#_Toc115013647)

[Instrument Connector 82](#_Toc115013648)

# Overview

## Purpose

This document provides the requirements for the NOAKA DEXPI project transfer of the P&ID between intelligent engineering systems using the DEXPI 1.3 standard.

## Abbreviations

|  |  |  |
| --- | --- | --- |
|  | **Description** | **Comment** |
| DEXPI | Data Exchange in the Process Industry |  |
| P&ID | Process & Instrument Diagram |  |
| OPC | Off Page Connector |  |
| PIF | ProcessInstrumentFunctions |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## References

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref** | **Document / Standard** | **Title / Description** | **Rev** |
| [[1](https://15926.org/home/)] | ISO 15926 | Interoperability standard for process plants |  |
| [[2](https://www.standard.no/fagomrader/energi-og-klima/petroleum/norsok-standard-categories/z-technical-info/z-004/)] | Z-004 | NORSOK CAD Symbol Libraries |  |
| [3] | TR3111 | Equinor LCI Requirements - Data Content and Transfer |  |
| [[4](https://github.com/ProteusXML/proteusxml)] | Proteus schema | [Transfer schema](https://github.com/ProteusXML/proteusxml) | 4.1 |
| [[5](https://dexpi.org/wp-content/uploads/2020/09/DEXPI-PID-Specification-1.3.pdf)] | DEXPI | [DEXPI P&ID Specification Standard](https://dexpi.org/wp-content/uploads/2020/09/DEXPI-PID-Specification-1.3.pdf) | 1.3 |
| [[6](https://gitlab.com/dexpi/Specification/raw/master/specification/DEXPI%20Specification%201.2.pdf)] | DEXPI | [DEXPI P&ID Specification Standard](https://gitlab.com/dexpi/Specification/raw/master/specification/DEXPI%20Specification%201.2.pdf) | 1.2 |
| [[7](https://equinor.github.io/NOAKADEXPI/Symbols.xlsm)] | Symbols.xlsm | [NOAKA DEXPI Symbol Library](https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2FEquinor.github.io%2FNOAKADEXPI%2FSymbols.xlsm&wdOrigin=BROWSELINK) | 1.0 |
| [[8]](https://github.com/ProteusXML/proteusxml/blob/master/additional_documents/P%26ID Profile file specification 3.3.3.doc) | P&ID File Specification | [ADI / IDS ISO-15926 P&ID file specification](https://github.com/ProteusXML/proteusxml/blob/master/additional_documents/P%26ID%20Profile%20file%20specification%203.3.3.doc) | 1.5 |
| [9] | POSC Caesar Reference Data Library | [Reference Data Service (posccaesar.org)](http://data.posccaesar.org/rdl/) |  |

## Constraints/Assumptions

The following points indicate general constraints/assumptions made when writing this document that were used to determine the scope and should be referenced when determining the design solution.

1. DEXPI 1.3 standard shall be used within the project.
2. It is required that all parties shall use the same graphical representation and ID of each symbol as defined in the NOAKA DEXPI Symbol Library (Ref: [7] ). It is vital that dimension, origo, rotation and mirroring point information is correctly defined in the symbol as per the definition.
3. NOAKA DEXPI Symbol Library symbols shall be used as the basis for mapping towards DEXPI class definitions.
4. Any Symbol that does not have a match to a DEXPI class definition will be defined with a DEXPI custom class and contain a TypeName and TypeURI reference to provide the type details. POSCCAESAR Ref: [9] will be used as the reference library where possible.
5. Each DEXPI class element shall include a maximum of one symbol reference.
6. No graphical primitive information shall be transferred when the Symbol Reference ID is provided for the object.
7. Current drawing revision number shall be transferred – no historical drawing revision information shall be transferred.
8. It is assumed that all main objects; pipelines, equipment, instruments and main piping components shall exist in the target system.
9. Assume DEXPI diagram item will always have MinX = 0, MinY = 0, BackgroundColor = “white”
10. Project shall assume ‘en’ as the default language for MultiLanguageString types it is therefor not required to transfer ‘Language’ value for these types.
11. Equipment box/table shall not be transferred.

# Process and Instrumentation Diagram Transfer

## ****Overview****

The P&ID is a detailed graphical representation of the process flow and interconnection of the involved components, including piping, equipment, valves, instrumentation, and other process components.

Within the NOAKA DEXPI project the content of the P&ID has been broken down into the groups shown in the figure below; requirements regarding the transfer of each group are detailed in the sections below with a focus on the C01 example as redrafted by Aibel.

### ****P&ID Transfer Breakdown Grouping****

Diagram, engineering drawing

Description automatically generated

Figure 1: P&ID NOAKA DEXPI breakdown

## ****P&ID Example C01\_NOA1 June MileStone Example****

As part of the DEXPI initiative the following example P&ID drawing has been provided. This drawing shall be reproduced in each of the drawing tools as shown in the example below and shall be used as the DEXPI export/import example to demonstrate compliance with the June 2022 milestone requirements.

All symbols shall be shown in the rotation, placement, naming and label placement as below.

\*\*Note: colour has been used to indicate instrument bubbles used for labels (black) vs field instruments (green). Colour information is NOT required in the DEXPI transfer or example file details.

Diagram, schematic

Description automatically generated

Figure 2: Example C01\_NOA1 (June milestone)

## ****DEXPI Standard****

The DEXPI P&ID Specification defines an information model for P&IDs as well as a mapping to the exchange format Proteus Schema.

For the NOAKA DEXPI project all transfers and configuration shall be based on the DEXPI 1.3 standard.

Changes to the DEXPI standard from 1.2 to 1.3 include:

* Equipment types added for solid processes
* Units of measurement cleaned up
* More comprehensive specification of graphics
* Introduction of CustomClass and CustomAttribute elements

### Analysis:

The following sub-section details weaknesses with the DEXPI 1.3 standard that should be addressed with the DEXPI group.

#### Engineering Notes

Engineering notes may be applied to one or many objects on a P&ID to provide additional information to the end user. Notes are an import method of communicating additional information about the process or requirements and are commonly used on the P&ID.

**Issue:** The current solution to provide for the transfer of notes relies on the CustomAttribute element and specialized rules to define the association of the note and the graphical object.

**Suggestion**: A suggestion would be to introduce a new object type within DEXPI

## Proteus Schema

Proteus 4.1 is the current exchange format used to implement the DEXPI 1.3 standard for the P&ID. This format may be replaced in future versions of DEXPI. The analysis below shall be provided as feedback to the DEXPI group to support discussions related to enhancing or replacing Proteus going forward.

### Analysis

The following sub-section details weaknesses with the Proteus 4.1 standard that should be addressed with the DEXPI group.

#### Object Symbol Reference

P&ID design can require the use of more than one symbol to graphically define a single tagged object.

**Issue:** The Proteus format does not support more than one symbol reference per class definition, requiring some the definition of new combined symbols within the project.

**Suggestion**: Create a new symbol as a combination of symbols where two or more symbols are used to represent a single DEXPI class.

## Transfer Requirements

Within the NOAKA DEXPI project it has been decided that the symbols shall form the basis of the mapping toward DEXPI to support the initial decision that we shall only transfer the graphical information from the P&ID in the DEXPI transfer file.

The NOAKA DEXPI project requires that each equipment object displayed on the P&ID shall be transferred within the DEXPI format with the required class mapping, symbol reference ID mapping, attribute mapping and association references to the piping / instrumentation systems as shown on the P&ID. The following sections provide details for the class and attribute mapping and implementation of the standard to reflect NOAKA process engineering drafting standards and symbols. Any additional rules or assumptions to be applied will also be noted in these sections.

**General Requirement Details:**

* PersistentID referencing shall be used to uniquely identify each class object. The context given within the PersistentID shall include a reference to the exporting application.
* DexpiCustomAttributes shall be used as the GenericAttributes ‘Set’ value within the Proteus export to group custom attributes that are defined within ANNEX B: Custom attribute defintions
* All Dexpi custom class definitions (DEXPI CustomObject subtypes) must use the DEXPI TypeName and TypeURI attributes as part of the class element transfer to provide the type details for the object. TypeName & TypeURI are provided in the symbol legend mapping.

### Symbols

The NOAKA DEXPI symbol legend is based on Z004 NORSOK standard with some additions to support the DEXPI transfer, this project symbol legend is referred to as the NOAKA DEXPI symbol legend.

All symbols within this legend are mapped to a corresponding DEXPI class definition including any additional attributes to support the definition of the object.

A key feature of the NOAKA DEXPI symbol legend is the addition of a defined origo point for all symbols and the inclusion of the label as part of the overall graphic. A standard origo point definition across NOAKA will ensure correct placement of the symbol based on a single point element (x,y) for the shape in the transfer file.

The origo point placement is based on a few general rules:

* In-line components with 2 connectors the origo is placed on the center of the line transecting the symbol along the plane of the to-be connected pipeline to ensure ease of rotation on the pipeline and ease of substitution for intelligent attribute based symbols.
* In-line components with 3 or 4 connectors the origo is placed at the intersection point of the connectors
* In-line components with 1 connection point the origo is placed at the connection point

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | **Label Details** | **Label Attribute(s)** | **Options** |
| **DEXPI ID** | **Graphic** | **Description** | **Rotation** | **Mirroring** | **ResizingX** | **ResizingY** | **TR1970** | **Z004** | **Symbol** | **A** | **1** |
| [ND0011](https://equinor.github.io/NOAKADEXPI/Symbols/Detail/ND0011_Detail.svg) |  | Spring Actuated Safety Valve | Yes | Yes | No | No |  |  | IM005A |  |  |
| [PV019A](https://equinor.github.io/NOAKADEXPI/Symbols/Detail/PV019A_Detail.svg) |  | Valve Ball | Yes | Yes | No | No | STLV008 | PV019A |  | <ObjectDisplayName> | [ValvePosition = 'NC'](https://equinor.github.io/NOAKADEXPI/Symbols/PV019A_Option1.svg) |

#### Requirement Details:

* SymbolRegistrationNumber attribute within the Shape element shall be used to provide the reference to the NOAKA DEXPI symbol reference ID.
* Name attribute of the Shape element shall be used as an internal file identifier to link the Shape symbol reference of the object to the Graphical representation instance of the object. Name is implemented in Proteus using ComponentName attribute.
* Location of the instance shall be given by the Shape Usage (Position / Reference) attributes for X & Y axiis. The X & Y points are the location reference to the origo of the symbol as defined in the symbol library for the project.
* Project shall use Label type ‘TagNameLabel’ for any item labels that are represented by a symbol e.g. for SafetyReliefValve.
  + Note: for most symbols the label is defined as part of the overall graphic. Ref [7]
* Transformation definition for the symbol shall use the definitions and rules as defined within the DEXPI standard (This is described in Ref [6] p. 19). Although any rotation shall be allowed and possible to transfer in the DEXPI file it is preferrable that only 0,90,180 & 270 deg anti-clockwise rotation of the symbol is used.
  + X-axis mirroring is provided for within DEXPI 1.3 through the combination of attributes: ‘IsMirrored=True’ and ‘Rotation=180.0’ as per example (Ref: ANNEX D: Symbol Rotation and mirroring example)
* Target application shall apply functionality to ensure that any text given as part of the overall symbol is displayed from left-to-right or bottom-to-top based on the rotation of the graphic.
* Target application shall use the rotation values from the transfer file and interpret the information to ensure graphical best fit with a focus on ensuring any symbol rotation transferred can be imported and displayed.
* All connection points shall be on the grid. This will ensure correct placement and management of the drawing connections for import.
* Symbol scaling shall be given by the ScaleX and ScaleY attribute of the DEXPI ShapeUsage element (implemented by the Proteus Scale element X and Y attributes)

#### DEXPI Model Examples

##### DEXPI w/ Proteus implementation: Shape SymbolRegistrationNumber

Ref: [Proteus Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/LabelSymbolRef.xml)

|  |  |  |
| --- | --- | --- |
| DEXPI definition |  | Proteus Implementation |
| Figure 3: DEXPI Shape SymbolRegistrationNumber |  | Figure 4: Proteus Implementation SymbolRegistrationNumberAssignmentClass |

##### DEXPI Model: Shape with label symbol

|  |  |  |
| --- | --- | --- |
| DEXPI definition |  | Proteus Implementation |
| Figure 5: DEXPI model example Safety Valve Label Symbol Reference |  | Figure 6: Proteus Implementation SymbolRegistrationNumberAssignmentClass for label |

### Persistent ID

* The following cases have been discussed and agreed upon for this revision:

|  |  |
| --- | --- |
| **Case** | **Persistent ID required?** |
| ·         Plant Break Down Structure Items e.g. Plant, System, Area | ·         Yes |
| ·         Tagged Items (created within the Contractors system) | ·         Yes |
| ·         Tagged Items (bulk created and reserved within Equinor STID) | ·         Yes |
| ·         Untagged items that are part of the process and are graphically represented on the P&ID e.g. flange, nozzle | ·         Yes |
| ·         Untagged items that are **NOT** part of the process and are graphically reqresented on the P&ID e.g. property breaks, flow direction, elevation, labels. | ·         No for the examples given |
| ·         Untagged Items using commonly accepted rules to group: e.g. Loop | ·         Yes for the example given |
| ·         Untagged Items used to group based on internal rules (that are not common across systems) e.g. DEXPI PipingNetworkSegment | ·         Persistent ID for PipingNetworkSystem. |
| ·         No persistent ID for PipingNetworkSegment or Pipe (Centerline). Reassess after creating/applying update use cases. |
| ·         Items without graphical representation on the P&ID created ‘on the fly’ during export as required by DEXPI which contain process related information e.g. SignalConveyingFunction (used for alarms) | ·         Yes |
| ·         Items without graphical representation on the P&ID created ‘on the fly’ during export as required by DEXPI which contain **NO** process related information e.g. VirtualPipingConnector, ProcessSignalGeneratingFunction | ·         No reassess after update use cases. |
| ·         MeasuringLineFunction | ·         No not required |

* Using the definitions from the cases above persistent ID requirements for the DEXPI class types are defined as follows:

|  |  |  |
| --- | --- | --- |
| **DEXPI class type** | **Comment** | **Persistent ID** |
| PlantStructureItem | All subtypes which are included in the export | Required |
| Note |  | Required |
| PipeOffPageConnector |  | Required |
| SignalOffPageConnector |  | Required |
| Equipment | All subtypes which have a representation on the P&ID | Required |
| Nozzle | NOT included: items with IsVirtual=TRUE | Required |
| PipingNetworkSystem |  | Required |
| PipingComponent | All subtypes which have a representation on the P&ID. NOT included: VirtualPipingConector | Required |
| ControlledActuator |  | Required |
| ProcessInstrumentationFunction |  | Required |
| InstrumentationLoopFunction |  | Required |
| SignalLineFunction |  | Required |
| PipingNetworkSegment |  | Not Required |
| PipingConnection |  | Not Required |
| MeasuringLineFunction |  | Not Required |
| ActuatingElectricalFunction |  | Not Required |
| ActuatingFunction |  | Not Required |
| ProcessSignalGeneratingFunction |  | Not Required |
| ProcessSignalGeneratingSystem |  | Not Required |
| SensingLocation |  | Not Required |

### File & Drawing Metadata

File and Drawing Metadata are the high level data used to give information about the file itself and also drawing top level details. Many of the MetaData attribute are part of the drawing template as shown below:

Diagram

Description automatically generated

Figure 7: DEXPI MetaData model example

##### Requirement Details:

* Only current revision information shall be transferred in the export file.
* File encoding shall be ‘uft-8’
* Each DEXPI transfer file shall include the following PlantInformation XML attributes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Comment** | **Value** |
| SchemaVersion | Proteus schema version | 4.1.1 |
| Date | Export UTC date format yyyy-mm-dd |  |
| Time | Export UTC time format HH:mm:ss |  |
| Units | Unit of measure reference for all measurement values (used in the Verificator) |  |
| ApplicationVersion | DEXPI standard | 1.3.1 |

* Each DEXPI transfer file shall include the following attributes as part of the drawing metadata element:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ApprovalDateRepresentation | http://sandbox.dexpi.org/rdl/ApprovalDateRepresentationAssignmentClass |  |  |
| ApprovalDescription | http://sandbox.dexpi.org/rdl/ApprovalDescriptionAssignmentClass |  |  |
| ApproverName | http://sandbox.dexpi.org/rdl/ApproverNameAssignmentClass |  |  |
| CheckerName | http://sandbox.dexpi.org/rdl/CheckerNameAssignmentClass |  |  |
| DrafterName | http://sandbox.dexpi.org/rdl/DrafterNameAssignmentClass |  | P&ID drafter user ID |
| DrawingName | http://data.posccaesar.org/rdl/RDS2102503531 | PIPING AND INSTRUMENT DIAGRAM |  |
| DrawingNumber | http://sandbox.dexpi.org/rdl/DrawingNumberAssignmentClass | PID.001 |  |
| DrawingSubTitle | http://sandbox.dexpi.org/rdl/DrawingSubTitleAssignmentClass | TEST P&ID |  |
| PlantAreaName | http://sandbox.dexpi.org/rdl/AreaIsa95NameAssignmentClass |  |  |
| PlantSystemIdentificationCode | http://sandbox.dexpi.org/rdl/PlantSystemIdentificationCodeAssignmentClass | 20 |  |
| ProcessPlantIdentificationCode | http://sandbox.dexpi.org/rdl/ProcessPlantIdentificationCodeAssignmentClass | D |  |
| ProcessPlantName | http://sandbox.dexpi.org/rdl/ProcessPlantNameAssignmentClass | KRAFLA |  |
| ProjectNumber | http://sandbox.dexpi.org/rdl/ProjectNumberAssignmentClass |  |  |
| RevisionNumber | http://sandbox.dexpi.org/rdl/RevisionNumberAssignmentClass | 01 |  |
| SheetFormat | http://sandbox.dexpi.org/rdl/SheetFormatAssignmentClass | NTS at A1 |  |
| CreatorName | http://sandbox.dexpi.org/rdl/CreatorNameAssignmentClass |  | User ID of person exporting P&ID to DEXPI |

### Process Plant, Plant System & Plant Area

System numbering codes and plant area codes are both given on the P&ID to provide information about the engineering elements. DEXPI provides for the transfer of these elements via the PlantSystem & PlantArea class definitions.

Diagram

Description automatically generated

Figure 8: DEXPI ProcesPlant, PlantSystem & PlantArea model example

#### Requirement Details:

* ProcessPlant class shall be transferred.
* Each ProcessPlant DEXPI transfer object shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ProcessPlantIdentificationCode | http://sandbox.dexpi.org/rdl/ProcessPlantIdentificationCodeAssignmentClass | D | Plant code |

* PlantSystem association shall be used where this information is available for an item.
* Each PlantSystem DEXPI transfer object shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| PlantSystemIdentificationCode | http://sandbox.dexpi.org/rdl/PlantSystemIdentificationCodeAssignmentClass | 20 | System code associated with the item |

* PlantArea association shall be used where this informationis available for an item.
* Each PlantArea DEXPI transfer object shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| PlantAreaIdentificationCode | http://sandbox.dexpi.org/rdl/PlantAreaIdentificationCodeAssignmentClass | B23 | Area code associated with the item |

### Equipment

Equipment is the main items shown in a P&ID to perform the process required treatment. The plant equipment is shown in the P&ID by an icon showing the equipment in basic manner. Equipment is usually identified by a name and unique tag (Unique identifier that is assigned to a field device, skid or equipment).

Diagram

Description automatically generated

Figure 9: DEXPI Equipment model example

#### Requirement Details:

* Direct connections between Equipment shall be identified within the DEXPI transfer file by transferring the secondary equipment (e.g. motor) unique ID value in the ConnectedEquipment attribute on the primary equipment (e.g. pump) element.
* Each Equipment DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ConnectedEquipmentID | http://noaka.org/rdl/ConnectedEquipmentIDAssignmentClass |  | Proteus XML ID value of the connected equipment element |
| ObjectDisplayName | http://noaka.org/rdl/ObjectDisplayNameAssignmentClass | D-20PA001 | Label text as displayed on the P&ID |
| Sequence | http://noaka.org/rdl/SequenceAssignmentClass | 001 | Sequence number which is part of the tag name. |
| TagType | http://noaka.org/rdl/TagTypeAssignmentClass | PA | Letter code indicating the function of the item. |
| EquipmentDescription | http://data.posccaesar.org/rdl/RDS2181987301 | GEAR PUMP | Functional service description of the tagged item. |
| TagName | http://sandbox.dexpi.org/rdl/TagNameAssignmentClass | D-20PA001 | Tag name as stored in the tag register system. |

* For each Equipment DEXPI transfer object represented by a symbol with an type code text component shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TypeCode | http://noaka.org/rdl/TypeCodeAssignmentClass | E | Text code to be displayed within the symbol. Ref: [7] ‘Label Attributes’ |

### Piping

A Piping System is an assembly of various components put together with a proper method of joints, functionally to transport fluid from its source to destination. The different components put together are defined as piping components. They are designed for withstanding the operating and design conditions specified in the process parameters. The following sub-sections detail specific subsets of requirements based on component group types within the piping system.

Diagram

Description automatically generated

Figure 10: DEXPI PipingNetworkSystem model example

#### Piping

This sub-section details the transfer requirements related to the PipingNetworkSystem, PipingNetWorkSegment and Pipe elements.

##### Requirement Details:

* Each pipeline shall be represented by a separate PipingNetworkSystem
* PipingNetworkSystems shall contain all PipingNetworkSegments that share the same pipeline name ‘ItemTag’ attribute value.
* Each pipeline shall contain one or more PipingNetworkSegments where the topology of the PipingNetworkSegments is defined in ANNEX C: Draft update ‘P&ID Profile file specification 3.3.3’
* Where heat tracing is utilized, this information shall be transferred via DEXPI using the HeatTracingType attribute on the PipingNetworkSegment. All elements within the PipingNetworkSegment shall inherit the heat tracing setting using one of the following values:
  + NULL: heat tracing system has not yet been evaluated
  + HeatTracingSystem: Heat tracing will be used on the piping network system
  + NoHeatTracingSystem: Heat tracing will not be used on the piping network system
* Piping connections shall be defined within the Proteus export file as per the Piping Connector details in ANNEX H: DEXPI Connection Points & Associations
* Each PipingNetworkSystem DEXPI transfer object shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | D-20L00001A | The pipeline tag number. |
| LineDescription | http://noaka.org/rdl/LineDescriptionAssignmentClass | D-20L00001A-1800PL-AD200- | Complete Line number as indicated on the P&ID. |
| LineSuffix | http://noaka.org/rdl/LineSuffixAssignmentClass | A | Size indicator suffix |
| ObjectDisplayName | http://noaka.org/rdl/ObjectDisplayNameAssignmentClass | D-20L00001A-1800PL-AD200- | Label text as displayed on the P&ID |
| ProductCode | http://noaka.org/rdl/ProductCodeAssignmentClass | PL | Product service code for the line |
| PipingClassCode | http://sandbox.dexpi.org/rdl/PipingClassCodeAssignmentClass | AD750 | Piping class for the line |
| HeatTracingRequirement | http://noaka.org/rdl/HeatTraceRequiredAssignmentClass | Y/N/S | Is heat tracing required for some/all of the pipeline: Y-Yes; N-No; S-Safety critical |

* Each PipingNetworkSegment DEXPI transfer object shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | D-20L00001A | The pipeline tag number. |
| SegmentLineTypeRepresentation | http://noaka.org/rdl/ SegmentLineTypeRepresentationAssignmentClass | Primary/Secondary/Utility | Indicate line type to use |
| HeatTracingType | http://sandbox.dexpi.org/rdl/HeatTracingTypeSpecialization | NoHeatTracingSystem | Indicates that no heat tracing is used. |
| NominalDiameterNumericalValueRepresentation | http://sandbox.dexpi.org/rdl/ NominalDiameterNumericalValueRepresentationAssignmentClass | 800 | Nominal diameter for the line |

#### Piping Components

Piping components are those components that are connected in-line with the pipe to support the transport of fluid from its source to destination. Piping components referred to in this section can be (but are not limited to) one of the following types:

* Pipe fittings
* Flanges
* Gaskets
* Manually Operated Valves
* Special Items
* Nozzles
* Reducers

The Piping Components group as discussed here does not include pipes, safety valves or actuated valves. Safety valves and actuated valves are detailed in Instrumentation section.

##### Requirement Details:

* NozzleTee symbol shall be used to represent all nozzles on the P&ID
* Where a nozzle has not been included on the drawing between the equipment and the pipe/measuring line the transfer shall include a ‘virtual’ nozzle to ensure compliance with DEXPI. A ‘virtual’ nozzle shall be identified using the Nozzle custom attribute ‘IsVirtual= true’ (Ref: ANNEX B: Custom attribute defintions)
* BlindFlange DEXPI class mapping shall be used in the transfer file for the flange where there is one PipingNode connection
* Flange DEXPI class mapping shall be used in the transfer file for the flange where there are two PipingNode connections.
* VirtualPipingConnector custom class shall be used to provide for direct pipe to pipe connections i.e., where there is no graphical representation of a connecting piping compontent between the two pipes shown on the P&ID. (Ref: Virtual Piping Connector)
* Each PipingComponent DEXPI transfer object shall include the following attributes when available/applicable:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | D-VB20-0002 | Tag name as stored in the tag register |
| ObjectDisplayName | http://noaka.org/rdl/ObjectDisplayNameAssignmentClass | D-VB20-0002 | Label text as displayed on the P&ID |
| Sequence | http://noaka.org/rdl/SequenceAssignmentClass | 0002 | Sequence number which is part of the tag number. |
| TagType | http://noaka.org/rdl/TagTypeAssignmentClass | VB | Letter code indicating the function of the item. |

* For each PipingComponent DEXPI transfer object that is of type OperatedValve (or sub-type) shall include the following attributes in addition to the attributes defined in the table above when available. In the main these attributes will be used within the label for the Valve (Ref: ANNEX F: Valve Label Details)

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ValvePosition | http://noaka.org/rdl/ValvePositionAssignmentClass | NC | Valve Normal Position: Used to define how the graphics for the valve shall be displayed (e.g. black/closed or white/open)  Symbol options provided in the symbol library show how this will be represented. Ref [7] |
| VDS | <http://noaka.org/rdl/VDSAssignmentClass> | BMBD101R | Valve Data Sheet |
| NominalDiameter | <http://noaka.org/rdl/NominalDiameterAssignmentClass> | 0200 | Nominal diameter of the valve |
| TrimType | <http://noaka.org/rdl/TrimTypeAssignmentClass> | FB | Trim Type: e.g. Full Bore / Reduced Bore |
| LockMechanism | <http://noaka.org/rdl/LockMechanismAssignmentClass> | CSO | Lock Mechanism code |
| BarrierInformationFlowIn | http://noaka.org/rdl/BarrierInformationFlowInAssignmentClass | DP | Barrier Information e.g. DP (double piston) for the flow in side of the valve |
| BarrierInformationFlowOut | http://noaka.org/rdl/BarrierInformationFlowOutAssignmentClass | SR | Barrier Information e.g. SR (Self Relief) for the flow out side of the valve |
| TypicalReference | http://sandbox.dexpi.org/rdl/TypicalReferenceAssignmentClass |  | Code identifying the associated Typical if part of an actuated system. |

* Each Nozzle DEXPI transfer object shall include the following attributes when available/applicable:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| IsVirtual | http://noaka.org/rdl/IsVirtualAssignmentClass | TRUE | Set to ‘true’ if the Nozzle is included in the DEXPI modelling but does not have a graphical representation on the drawing. |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | M01 | Tag name as stored in the tag register |
| ObjectDisplayName | http://noaka.org/rdl/ObjectDisplayNameAssignmentClass | M01 | Label text as displayed on the P&ID |
| Sequence | http://noaka.org/rdl/SequenceAssignmentClass | 01 | Sequence number which is part of the tag number. |
| TagType | http://noaka.org/rdl/TagTypeAssignmentClass | M | Letter code indicating the function of the item. |

* For each PipingComponent DEXPI transfer object that is defined as a special item shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| SpecialItemNumber | http://noaka.org/rdl/SpecialItemNumberAssignmentClass | 3029 | Text for special item identification to be shown in SpecialItem label. |

* For each PipeReducer DEXPI transfer object shall include the following attributes when available/applicable:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| NominalDiameterIn | http://noaka.org/rdl/NominalDiameterInAssignmentClass | 0100 | Nominal diameter into the reducer |
| NominalDiameterOut | http://noaka.org/rdl/NominalDiameterOutAssignmentClass | 0200 | Nominal diameter out of the reducer |

### Instrumentation

Instrumentation is the items shown in a P&ID required to run, monitor and control a specific process. E.g. Indicators, Recorders, Controllers , including: pressure, temperature and flow instruments, control valves, pressure safety valves, meters etc.

#### Instrumentation (Off-Line Instrumentation)

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/NOAKA_Instrument_OffLine.xml)

Diagram

Description automatically generated

Figure 11:DEXPI Instrumentation model example within connection to PipingComponent (Flange)

Diagram

Description automatically generated

Figure 12:DEXPI Instrumentation model example within connection to PipingNetworkSegment and graphical connection nodes

##### Requirement Details:

* All instruments (not incl. actuated and safety valves) shall be represented by a ProcessInstrumentFunction (PIF) class element in DEXPI.
* InstrumentationLoopFunction class must be used as a 'grouping' mechanism for ProcessInstrumentFunction (PIF) elements where the loop identifier for those PIF items is known.
* InstrumentationLoopFunction class shall only be created in the export if there is an associated ProcessInstrumentationFunction class.
* Signal conveying lines shall be represented using DEXPI SignalConveyingFunction class using the SignalConveyingFunctionTypeRepresentation to define the line style Ref: [7] for valid type values.
* SignalConveyingFunction Parent shall be the ProcessInstrumentationFunction item associated with the ‘sending’ PIF. \*Ensures consistency with cases where the PIF is associated with an ActuatingFunction.
* Measure lines between instrument and piping component shall be represented using MeasuringLineFunction class.
* DEXPI MeasuringLineFunction shall have an ProcessSignalGeneratingFunction as its Source
* DEXPI ProcessInstrumentationFunction reference shall contain a ProcessInstrumentationFunctionType descriptor attribute where both a field device and shared display/shared control element exists for the object and the tag name for these objects is the same.
* Instrumentation connections shall be defined within the Proteus export file as per the Instrument Connector details in ANNEX H: DEXPI Connection Points & Associations
* The DEXPI SensingLocation for the DEXPI ProcessSignalGeneratingFunction shall be either Nozzle, PipingComponent or PipingNetworkSegment as per the DEXPI standard. In the case where an off-line instrument is drawn with a connection direct to a pipe the SensingLocation shall be set to PipingNetworkSegment.
* Each InstrumentationLoopFunction DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| InstrumentationLoopFunctionNumber | http://sandbox.dexpi.org/rdl/InstrumentationLoopFunctionNumberAssignmentClass | T0003 | Loop number use to group associated instruments. |

* Each Instrument (PIF) shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | D-20TI-0003 | Tag name as stored in the tag register system. |
| ProcessInstrumentationFunctionLocation | http://noaka.org/rdl/ ProcessInstrumentationFunctionLocationAssignmentClass | Field / Primary / Auxiliary / Inaccessable | Location information for the instrument. Ref: Figure 13 for details regarding use of attribute for instrument symbols. |
| ProcessInstrumentationFunctionType | http://noaka.org/rdl/ ProcessInstrumentationFunctionTypeAssignmentClass | Discrete / SharedDisplaySharedControl | Differentiate between field device and control function block. Ref: Figure 13 for details regarding use of attribute for instrument symbols. |
| TagSuffix | http://noaka.org/rdl/TagSuffixAssignmentClass | A | Suffix code |
| ProcessInstrumentationFunctionCategory | http://sandbox.dexpi.org/rdl/ ProcessInstrumentationFunctionCategoryAssignmentClass | T | Function category |
| ProcessInstrumentationFunctionNumber | http://sandbox.dexpi.org/rdl/ ProcessInstrumentationFunctionNumberAssignmentClass | 0003 | Sequence number |
| ProcessInstrumentationFunctions | http://sandbox.dexpi.org/rdl/ ProcessInstrumentationFunctionsAssignmentClass | I | Additional functions |
| SetPointTemperature | http://noaka.org/rdl/SetPointTemperatureAssignmentClass | 100.0 DegreeCelsius | Temperature SetPoint value and UoM |
| SetPointGaugePressure | http://noaka.org/rdl/SetPointGaugePressureAssignmentClass | 17.0 Bar | Gauge Pressure SetPoint value and UoM |
| SetPointLength | http://noaka.org/rdl/SetPointLengthAssignmentClass | 20.0 Centimetre | Length SetPoint value and UoM |

\*\* The engineering naming system can sometimes allow for two instrument symbols (elements) to be given the same tag name even though they represent different functions. It is therefore key to include the above attributes, particularly the ProcessInstrumentationFunctionType attribute.

* For each ProcessInsrumentationFunction DEXPI transfer object represented by a symbol with an type code text component shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TypeCode | http://noaka.org/rdl/TypeCodeAssignmentClass | PSD | Text code to be displayed within the symbol. Ref: [7] ‘Label Attributes’ and Ref: Figure 13 for details regarding use of attribute for instrument symbols. |

* Each SignalConveyingFunction DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| SignalConveyingFunctionTypeRepresentation | http://noaka.org/rdl/SignalConveyingFunctionTypeRepresentationAssignmentClass | SignalConveying | Linestyle type representation as defined in the Symbols legend for ‘SignalAndLineStyles’ |

Table

Description automatically generated

Figure 13: Instrumentation symbols with reference to ProcessInstrumentationFunctionType, ProcessInstrumentationFunctionLocation & TypeCode

##### Instrumentation (Off-Line instrument alarm and shut-down ‘cause’)

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/NOAKA_Instrument_PSDAlarm_Cause.xml)

The following section details additional modelling and transfer requirements for instrument control functions that have alarm points and/or associated shutdown function(s).

Graphical user interface, application, Word

Description automatically generated

Figure 14: DEXPI Instrumentation Alarm & Shut-down ‘cause’ model

###### Requirement Details:

* Main requirement details are as per Ref: Instrumentation (Off-Line Instrumentation)
* Alarm levels shall be represented by DEXPI SignalConveyingFunction class type and shall include the following attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| PortStatus | http://sandbox.dexpi.org/rdl/PortStatusSpecialization | StatusLowLowPort | Alarm type |

##### Instrumentation (Off-line Instrument shut-down ‘effect’ with multiple signal lines)

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/NOAKA_Instrument_PSDAlarm_Effect.xml)

The following section details additional modelling and transfer requirements for instrument shutdown function(s) ‘effect’ elements.

Note: each shutdown ‘cause’ object may have one or more ‘effect’ objects.

Diagram

Description automatically generated

Figure 15: DEXPI Instrumentation Shut-down ‘effect’ model with multiple signal lines

###### Requirement Details:

* Main requirement details are as per Ref: Instrumentation (Off-Line Instrumentation)
* Where more than one signal is connected to an actuator the export system shall ensure that there is 1 ‘owner’ of the ActuatingFunction and that this shall be a SharedDisplaySharedControl type PIF if available.

##### Instrumentation Electrical Powered Equipment

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/NOAKA_Instrument_ElectricalPowered.xml)

This is a special grouping for electrical powered objects that receive/send signals from other electrical powered objects, controllers, shutdowns and equipment.

Diagram

Description automatically generated

Figure 16: DEXPI Instrumentation Electrical Powered Equpment model

###### Requirement Details:

* Main requirement details are as per Ref: Instrumentation (Off-Line Instrumentation)
* Electrical signal lines between ‘instrument electrical powered equipment’ object and equipment shall be represented using SignalConveyingFunction class which has a DEXPI ActuatingElectricalFunction class as its Target.

#### Instrumentation (In-Line Instrumentation)

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/NOAKA_Instrument_InLine.xml)

\*\*Does not include instrument safety valves or actuated valves.

In-line instrumentation are those elements that are part of the piping network and provide an instrumentation function.

Graphical user interface, table

Description automatically generated

Figure 17: Example In-line thermowell with associated transmitter.

##### Requirement Details:

* In-line instrument on the P&ID shall be represented as both an in-line instrument **and** a separate off-line instrument as a ‘short-cut’ to allow for signal connections. In the case where a signal connection is required this shall be via a separate off-line instrument connected to the in-line instrument.
* All in-line instruments (not incl. actuated and safety valves) shall be represented by an InlinePrimaryElement (or subtype) class element in DEXPI.
* Measure lines between in-line instrument and off-line instrument shall be represented using MeasuringLineFunction class.
* DEXPI MeasuringLineFunction shall have an ProcessSignalGeneratingFunction as its Source
* Each DEXPI InlinePrimaryElement (or subtype) shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| InlinePrimaryElementFunctionCategory | http://noaka.org/rdl/InlinePrimaryElementFunctionCategoryAssignmentClass | T | Function category |
| InlinePrimaryElementFunctions | http://noaka.org/rdl/InlinePrimaryElementFunctionsAssignmentClass | W | Additional functions |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | D-20TI-0003 | Tag name as stored in the tag register system. |
| TagType | http://noaka.org/rdl/TagTypeAssignmentClass | TI | Letter code indicating the function of the item. |
| InstrumentLoopNumber | http://noaka.org/rdl/InstrumentLoopNumberAssignmentClass | T0003 | Loop number use to group associated instruments. |

* For each InlinePrimaryElement DEXPI transfer object that is defined as a special item shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| SpecialItemNumber | http://noaka.org/rdl/SpecialItemNumberAssignmentClass | 5068 | Text for special item identification to be shown in SpecialItem label. |

#### Instrumentation (Safety and Self Acting Valves)

Instrumentation (Safety and Self-Acting Valves) shall be transferred as a DEXPI PipingComponent suptype within the DEXPI PipingNetworkSystem.

Diagram

Description automatically generated

Figure 18: Instrumentation (Safety Valve) DEXPI model example

##### Requirement Details:

* Label symbol reference shall be transferred as a separate reference on the label element as per the example given in **Error! Reference source not found.**
* Each DEXPI SafetyValveOrFitting (or subtype) shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ItemTag | http://noaka.org/rdl/ItemTagAssignmentClass | D-20PSV-0002 | Tag name as stored in the tag register system. |
| Sequence | http://noaka.org/rdl/SequenceAssignmentClass | 0002 | Sequence number which is part of the tag number. |
| TagType | http://noaka.org/rdl/TagTypeAssignmentClass | PSV | Letter code indicating the function of the item. |
| InstrumentLoopNumber | http://noaka.org/rdl/InstrumentLoopNumberAssignmentClass | P0002 | Loop number use to group associated instruments. |
| SetPointTemperature | http://noaka.org/rdl/SetPointTemperatureAssignmentClass | 100.0 DegreeCelsius | Temperature SetPoint value and UoM |
| SetPointGaugePressure | http://noaka.org/rdl/SetPointGaugePressureAssignmentClass | 17.0 Bar | Gauge Pressure SetPoint value and UoM |
| SetPointLength | http://noaka.org/rdl/SetPointLengthAssignmentClass | 20.0 Centimetre | Length SetPoint value and UoM |

#### Instrumentation (Actuated Valves)

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/ActuatedValve_Example.xml)

Actuated Valves are complex DEXPI structures as the valve is part of the piping package while instrumentation provides the automation. The special DEXPI modelling is shown below. Note that for actuated valves there are a number of RepresentationGroups required: including the representation for the valve, the actuator and the instrument bubble symbol.

Diagram

Description automatically generated

Figure 19: DEXPI Actuated Valve model example

A screenshot of a computer

Description automatically generated with medium confidence

Figure 20: Proteus Implementation Actuated Valve (Ref: included example file ‘ActuatedValve\_Example.xml’)

##### Requirement Details:

* InstrumentationLoopFunction class association for an actuated valve shall be via the associated ProcessInstrumentationFunction object Ref: Figure 19: DEXPI Actuated Valve model example.
* DEXPI ActuatingFunction shall have an associated SignalLineFunction. The ActuatingFunction shall be the Target of the SignalLineFunction.
* The OperatedValve (or subtype) class shall contain the information relevant for the tag.
* The OperatedValve (or subtype) class element shall ‘own’ the link to the label for this item type.
* For each Actuator (ControlledActuator) DEXPI transfer object represented by a symbol with an type code text component shall include the following attribute:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TypeCode | http://noaka.org/rdl/TypeCodeAssignmentClass | M | Text code to be displayed within the symbol. Ref: [7] ‘Label Attributes’ |
| FailAction | http://sandbox.dexpi.org/rdl/FailActionSpecialization | FC | Fail Action code associated with the actuator. |

### Equipment Box

**Equipment box shall not be transferred**.

This section has been left as a reference in the event that the project determines it is necessary to transfer equipment box information.

Equipment Box/Table is the addition of a tabulated list of attribute name and value pairs that are associated with an equipment shown on the P&ID. The extent of the requirements for this list of attributes shall be as per the project requirements. The Equipment box transfer will be managed similarly to the symbol transfer via a reference ID.

Graphical user interface, application, table

Description automatically generated

Figure 21: DEXPI Equipment box model example

#### Requirement Details:

* Any attributes that are shown in the equipment box shall be transferred with the associated item in the transfer file.
* During the project only Tag Name and Description are required to be transferred using the DEXPI attribute definitions below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| TagName | http://sandbox.dexpi.org/rdl/TagNameAssignmentClass | D-20PA001 | Tag name as stored in the tag register system. |
| EquipmentDescription | <http://data.posccaesar.org/rdl/RDS2181987301> | GEAR PUMP | Functional service description of the tagged item. |

### Equipment Internals

**Equipment internals shall not be transferred**.

### Labels

NOAKA DEXPI requirements define that for most object symbols the label will form part of the overall graphical representation Ref: [7] ‘Label Attributes’. For other object types the information for that object may be displayed in an associated label symbol e.g. for Off-line Instruments. For this type of label the DEXPI and Proteus representation is provided: Ref: (Figure 5: DEXPI model example Safety Valve Label Symbol Reference) where the link between the object and the label is via the ComponentName attribute.

In addition to these types, information for the object may also be provided as a text string naming label related to the object e.g. for piping ‘PipingNetworkSystem’. In this case the label exported as a direct child object of the piping class element and refers by default to ‘ObjectDisplayName’ as the default attribute used for label text.

Graphical user interface, text, application, email

Description automatically generated

Figure 22: Proteus implementation for the PipingNetworkSystem naming label

### Note

Note is the text added to the P&ID to provide additional information about an item or items on the drawing. Each note shall be linked to at least one item on the drawing, however, the modelling described below will allow for notes that are not connected. An item on the drawing can have more than one note.

DEXPI standard does not currently contain a note class element, this has been identified as a gap within DEXPI. A suggestion has been provided to the DEXPI content group regarding the introduction of a new ‘Note’ class element as a direct child of the Conceptual Model, this suggestion has been approved for release in the next version of the DEXPI standard. It has also been approved for use within the NOAKA DEXPI project and adjustments to the DEXPI verificator will be made accordingly to ensure that this new class type does not produce verification errors.

Ref: ANNEX G: NOAKA DEXPI Notes Implementation

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/Annotations.xml)

Diagram

Description automatically generated

Figure 23: NOAKA DEXPI Notes model example

Graphical user interface, application

Description automatically generated

Figure 24: Proteus Implementation NOAKA DEXPI Notes example (Ref: included example file ‘NoteDEXPISuggestion.xml’)

#### Requirement Details:

* Each note shall have a PersistentID
* For each Note DEXPI transfer object shall include the following attributes (which will be considered as Dexpi attributes and not custom attributes in this project):

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| NoteIdentifier | http://sandbox.dexpi.org/rdl/NoteIdentifierAssignmentClass | 1 | Note reference ID |
| NoteText | http://sandbox.dexpi.org/rdl/NoteTextAssignmentClass | This is a note text | Note text |
| NoteClassification | http://sandbox.dexpi.org/rdl/NoteClassificationAssignmentClass | Technical Requirement | Note classification |

### Miscellaneous Graphics with Symbol Reference

#### Graphics Modelled within Dexpi

The following section contains details for those miscellaneous elements that are modelled within DEXPI 1.3 standard and are defined with a symbol reference.

##### PropertyBreak

Property break graphic shall be used to represent the break of ‘one’ attribute only i.e. each break type shall be represented by a separate property break graphic.

The type of break being represented by the property break graphic shall be identified by setting the associated property break attribute as defined below. The graphic shall display the ‘break’ attribute from the pipeline on either side of the property break graphic.

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/PropertyBreakExample.xml)



Figure 25: DEXPI PropertyBreak model example

###### Requirement Details:

* \*\*Note: PropertyBreak class is a PipingNetworkSegmentItem subtype in DEXPI 1.3
* A PropertyBreak element is required to have the same x,y ordinate values for both process nodes as the PropertyBreak represents a single point.
* Each PropertyBeak DEXPI transfer object shall use the following attributes to identify the type of property break:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Transfer Value** | **Value RDS** |
| AreaBreak | http://noaka.org/rdl/AreaBreakAssignmentClass | AreaBreak | NA |
| HeatTracingBreak | http://noaka.org/rdl/HeatTracingBreakAssignmentClass | HeatTracingBreak | NA |
| InsulationBreak | http://sandbox.dexpi.org/rdl/InsulationBreakSpecialization | InsulationBreak | http://sandbox.dexpi.org/rdl/InsulationBreak |
| NominalDiameterBreak | http://sandbox.dexpi.org/rdl/NominalDiameterBreakSpecialization | NominalDiameterBreak | http://sandbox.dexpi.org/rdl/NominalDiameterBreak |
| PipingClassBreak | http://sandbox.dexpi.org/rdl/PipingClassBreakSpecialization | PipingClassBreak | http://sandbox.dexpi.org/rdl/PipingClassBreak |
| ContractorBreak | http://noaka.org/rdl/ContractorBreakAssignmentClass | ContractorBreak | NA |
| CommissioningBreak | http://noaka.org/rdl/CommissioningBreakAssignmentClass | CommissioningBreak | NA |
| PipingInstrumentBreak | http://noaka.org/rdl/PipingInstrumentBreakAssignmentClass | PipingInstrumentBreak | NA |
| LineIDBreak | http://noaka.org/rdl/LineIDBreakAssignmentClass | LineIDBreak | NA |
| BreakValue1 | http://noaka.org/rdl/BreakValue1AssignmentClass | Text for downstream or first break value |  |
| BreakValue2 | http://noaka.org/rdl/BreakValue2AssignmentClass | Text for upstream or second break value |  |

##### Piping Off Page Connector (Out)

Piping Off Page Connector (OPC) graphic shall be used to represent a pipeline that continues elsewhere either on the same drawing or on another drawing with the flow direction ‘out’.

Diagram

Description automatically generated

Figure 26: DEXPI FlowOutPipeOffPageConnector model example

###### Requirement Details:

* The piping off page connector (Flow Out) shall use the correct symbol to indicate flow direction ‘out’ to the side dependant on which side of the drawing the graphic is placed.
* Each FlowOutPipeOffPageConnector shall be transferred with an associated PipeOffPageConnectorReferenceByNumber DEXPI transfer object as per the figure above.
* Each PipeOffPageConnectorReferenceByNumber DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ReferencedConnectorNumber | http://sandbox.dexpi.org/rdl/ReferencedConnectorNumberAssignmentClass | TBD | The unique key used to match this connector with its matching counterpart. |
| ReferencedDrawingNumber | http://sandbox.dexpi.org/rdl/ReferencedDrawingNumberAssignmentClass | TBD | The Name attribute of the Drawing that the matching ConnectorSymbol is on. |

##### Piping Off Page Connector (In)

Piping Off Page Connector (OPC) graphic shall be used to represent a pipeline that continues elsewhere either on the same drawing or on another drawing with the flow direction ‘in’.

Diagram

Description automatically generated

Figure 27: DEXPI FlowInPipeOffPageConnector model example

###### Requirement Details:

* The piping off page connector (Flow In) shall use the correct symbol to indicate flow direction ‘in’ from the side dependant on which side of the drawing the graphic is placed.
* Each FlowInPipeOffPageConnector shall be transferred with an associated PipeOffPageConnectorReferenceByNumber DEXPI transfer object as per the figure above.
* Each PipeOffPageConnectorReferenceByNumber DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ReferencedConnectorNumber | http://sandbox.dexpi.org/rdl/ReferencedConnectorNumberAssignmentClass | TBD | The unique key used to match this connector with its matching counterpart. |
| ReferencedDrawingNumber | http://sandbox.dexpi.org/rdl/ReferencedDrawingNumberAssignmentClass | TBD | The Name attribute of the Drawing that the matching ConnectorSymbol is on. |

##### Piping Off Page Connector where flow is bi-directional or unknown

Where the flow is bi-directional or unknown the export tool shall manage the paired connectors for the pipe such that one connector is defined as ‘Flow Out’ and the other connector is defined as ‘Flow In’ as per the definitions above (Ref: Piping Off Page Connector (Out) and Piping Off Page Connector (In)). The symbol used to show the connector in this case is not related to flow direction.

##### Signal Off Page Connector (Out)

Signal Off Page Connector (OPC) graphic shall be used to represent a signal that continues elsewhere either on the same drawing or on another drawing with the flow direction ‘out’.

Diagram

Description automatically generated

Figure 28: DEXPI FlowOutSignalOffPageConnector model example

###### Requirement Details:

* The signal off page connector (Flow Out) shall use the correct symbol to indicate flow direction out dependant on which side of the drawing the graphic is placed.
* Each FlowOutSignalOffPageConnector shall be transferred with an associated SignalOffPageConnectorReferenceByNumber DEXPI transfer object as per the figure above.
* Each SignalOffPageConnectorReferenceByNumber DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ReferencedConnectorNumber | http://sandbox.dexpi.org/rdl/ReferencedConnectorNumberAssignmentClass | TBD | The unique key used to match this connector with its matching counterpart. |
| ReferencedDrawingNumber | http://sandbox.dexpi.org/rdl/ReferencedDrawingNumberAssignmentClass | TBD | The Name attribute of the Drawing that the matching ConnectorSymbol is on. |

##### Signal Off Page Connector (In)

Signal Off Page Connector (OPC) graphic shall be used to represent a signal that continues elsewhere either on the same drawing or on another drawing with the flow direction ‘in’.

Diagram

Description automatically generated

Figure 29: DEXPI FlowInSignalOffPageConnector model example

###### Requirement Details:

* The signal off page connector (Flow In) shall use the correct symbol to indicate flow direction in dependant on which side of the drawing the graphic is placed.
* Each FlowInSignalOffPageConnector shall be transferred with an associated SignalOffPageConnectorReferenceByNumber DEXPI transfer object as per the figure above.
* Each SignalOffPageConnectorReferenceByNumber DEXPI transfer object shall include the following attributes when available:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Value Example** | **Comment** |
| ReferencedConnectorNumber | http://sandbox.dexpi.org/rdl/ReferencedConnectorNumberAssignmentClass | TBD | The unique key used to match this connector with its matching counterpart. |
| ReferencedDrawingNumber | http://sandbox.dexpi.org/rdl/ReferencedDrawingNumberAssignmentClass | TBD | The Name attribute of the Drawing that the matching ConnectorSymbol is on. |

##### Signal Off Page Connector unknown flow direction

Signal Off Page Connectors typically do not store a flow direction and for this reason the tool shall manage the paired connectors for the signal such that one connector is defined as ‘Flow Out’ and the other connector is defined as ‘Flow In’ as per the definitions above (Ref: Signal Off Page Connector (Out) and Signal Off Page Connector (In)). The symbol used to show the connector in this case is not related to flow direction.

##### Slope

Pipeline slope graphic shall be used to represent the slope for the pipeline.

Graphical user interface

Description automatically generated

Figure 30: DEXPI Slope model example

###### Requirement Details:

* The pipeline slope graphic shall use the ‘PipeSlopeSymbol’ type as per the model example above.
* The pipeline slope graphic shall reference the SymbolRegistrationNumber ‘STPL008’ within the transfer file as per the model example above.
* The association between the PipingNetworkSegment and the pipeline slope graphic shall be provided in the DEXPI transfer file following the model example above.

##### Piping Flow Direction Arrow

Pipeline flow direction arrow graphic shall be used to represent the flow direction for the pipe.

Graphical user interface, application

Description automatically generated

Figure 31: DEXPI Slope model example

###### Requirement Details:

* The pipeline flow arrow graphic shall use the ‘PipeFlowArrow’ type as per the model example above.
* The pipeline flow arrow graphic shall reference the SymbolRegistrationNumber ‘ND0010’ within the transfer file as per the model example above.
* The association between the PipingNetworkSegment and the pipeline flow arrow graphic shall be provided in the DEXPI transfer file following the model example above.

##### Radio-active target/source identification symbol

Radioactive symbol is used to identify a radioactive source or target object.

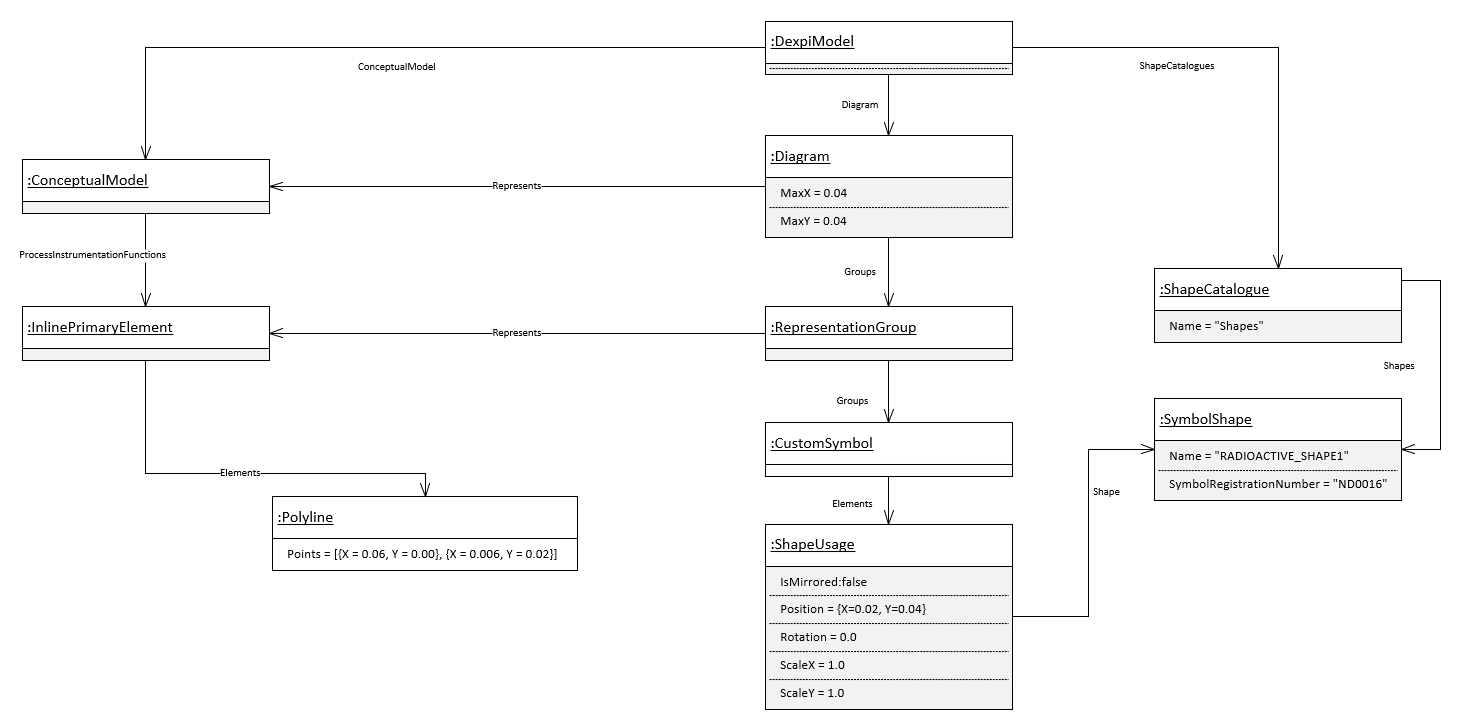


Figure 32: Radioactive symbol DEXPI model example

###### Requirement Details:

* The association between the ConceptualObject and the radioactive graphic shall be provided in the DEXPI transfer file following the model example above.

#### Graphics Items Without Symbol Reference

The following section details any special requirements regarding graphical elements that are not represented by a symbol.

##### Pipe

Pipe elements are represented in DEXPI as a connection between points: ConnectorLine InnerPoints and/or PipingNodePosition Position points.

Diagram

Description automatically generated

Figure 33: DEXPI Pipe with angle model example

###### Requirement Details:

* The PipingNodePosition Position point of the connected PipingNode shall represent a point on the PipingNodeOwner e.g. GateValve / Nozzle

##### Leader Lines

Leader lines are represented in DEXPI as a polyline with an ordered set of points depicting the start, end and any intermediate points of the line.

Diagram

Description automatically generated

Figure 34: Leader line graphic representation example

###### Requirement Details:

* Leader lines shall be ‘owned’ by the label as shown above.

# ANNEX A: Custom class definitions - Non Graphical Elements

The following section details the special custom class definitions without graphics that shall be available within the NOAKA DEXPI profile for verification within the project.

Additional custom class definitions for graphically represented objects can be found in the Symbol legend Ref: [7]

## Virtual Piping Connector

VirtualPipingConnector type is defined as a special piping connector type. This connector type is required in the case where the P&ID shows two pipes connecting directly with each other i.e., there is no graphical representation of a connecting piping compontent between the two pipes shown on the P&ID.

This type has a Supertype ‘PipingNodeOwner’ and thus provides the necessary PipingNode connection points between PipingNetworkSegments.

This type has no graphical representation.

### Requirement Details:

* Each CustomPipingComponet (VirtualPipingConnector) DEXPI transfer object shall use the following attributes to identify the type of property break:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Transfer Value** | **Value RDS** |
| TypeName | http://sandbox.dexpi.org/rdl/TypeNameAssignmentClass | VirtualPipingConnector |  |
| TypeURI | http://sandbox.dexpi.org/rdl/TypeURIAssignmentClass | http://sandbox.dexpi.org/rdl/VirtualPipingConnector | NA |

## Virtual Instrumentation Connector

VirtualInstrumentationConnector type is defined as a special instrument/piping connector type. This connector type is required in the case where the P&ID shows an instrument connecting directly with a pipe i.e., there is no graphical representation of a connecting piping compontent between the instrument and the pipe shown on the P&ID.

This type has a Supertype ‘PipingNodeOwner’ and thus provides the necessary PipingNode connection points between PipingNetworkSegments.

This type has no graphical representation.

### Requirement Details:

* Each CustomPipingComponet (VirtualInstrumentationConnector) DEXPI transfer object shall use the following attributes to identify the type of property break:

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **RDS** | **Transfer Value** | **Value RDS** |
| TypeName | http://sandbox.dexpi.org/rdl/TypeNameAssignmentClass | VirtualInstrumentationConnector |  |
| TypeURI | http://sandbox.dexpi.org/rdl/TypeURIAssignmentClass | http://sandbox.dexpi.org/rdl/VirtualInstrumentationConnector | NA |

# ANNEX B: Custom attribute defintions

Refer to NOAKA DEXPI Symbol legend file Ref: [7]

# ANNEX C: Draft update ‘P&ID Profile file specification 3.3.3’

**(Ref: [8]) Section 2.2.1 for DEXPI 1.3)**

**PipingNetworkSegment Topology (Connection element)**

Components within a PipingNetworkSegment are considered to be implicitly connected, by their main flow in and flow out connections points, in the order that they are represented in the PipingNetworkSegment. This ordering may differ between a P&ID and 3D model (see **Error! Reference source not found.**)

Each PipingNetworkSegment is a collection of PipingNetworkSegmentItems (e.g., PipingComponents such as Valves) and PipingConnections (e.g., Pipes) with common engineering properties that define a single process flow. Where there is a junction in the flow or a change of specification (e.g., piping class or nominal diameter), the PipingNetworkSegment will terminate.

A PipingNetworkSegment, as its SourceItem, will reference a Nozzle, PipingComponent, or PropertyBreak that it doesn’t contain or it will reference a FlowInPipeOffPageConnector that it contains.

Connection from :

Nozzle

PipingComponent

PropertyBreak

PipingNetworkSegment

…

PipingNetworkSegment

FlowInPipeOffPage­Connector

…

Connection from :

FlowInPipeOffPage­Connector

A PipingNetworkSegment, as its TargetItem, will reference a Nozzle, merging component (a PipingComponent such as a Tee) that it doesn’t contain or it will reference a PipingComponent, PropertyBreak, or FlowOutPipeOffPageConnector that it contains as its last component.

PipingNetworkSegment

PipingNetworkSegment

Connection to :

Nozzle

Merging component

…

PipeConnectorSymbol

Reducer

Splitting Component

…

Connection to :

PipingComponent, PropertyBreak  
FlowOutPipeOffPage­Connector

If the TargetItem of a PipingNetworkSegment is a contained PipingNetworkSegmentItem, the TargetNode of the PipingNetworkSegment is the main downstream PipingNode of this PipingNetworkSegmentItem, if applicable (i.e. the main flow out of the segment).

# ANNEX D: Symbol Rotation and mirroring example

The following table shows the rotation and mirroring of an example symbol and demonstrates that mirroring the base symbol (shown in RED) on the Y-axis and then rotating 180o provides the same result as if the base symbol had been mirrored on the X-Axis. \*\* **scaling before rotation**

Ref: [Protues Example](https://github.com/equinor/NOAKADEXPI/blob/main/ProteusExamples/mirroring%20and%20rotation.xml)

Table

Description automatically generated

Figure 35: Symbol Rotation Attribute Example

## Rotation and Mirroring in DEXPI and Proteus

### DEXPI

* geometry unit: m (because SI base unit)
* axes and rotation like in SVG:
  + horizontal axis (x): left to right
  + vertical axis (y): top to bottom
  + rotation: in degrees, clockwise
* meaning of ShapeUsage attributes (**note the order:** **in particular: scaling before rotation**):
  + ScaleX, ScaleY: shape is scaled along x and y axis w.r.t. its origo (0, 0)
  + if IsMirrored: shape is mirrored at y axis (x=0)
  + Rotation: around (0, 0), in degrees, clockwise
  + Position: shape is translated such that its origo (0, 0) is translated to Position

### Proteus

* geometry unit: given in <PlantInformation>, e.g., m or mm
* axes and rotation:
  + horizontal axis (x): left to right
  + vertical axis (y): bottom to top
  + rotation: in degrees, counterclockwise
* <Position> contains
  + <Location> = Position in DEXPI
  + <Axis> and <Reference> correspond to DEXPI Rotation and IsMirrored
    - <Axis> is rotation axis  
      we allow only rotation around third (z) axis:
      * <Axis X="0" Y="0" Z="1"/> means “not mirrorred”
      * <Axis X="0" Y="0" Z="-1"/> means “mirrored”
    - <Reference> is unit vector describing rotation of α around rotation axis:
      * <Reference X=”cos α”, Y=”sin α”, Z=0>
      * note that rotation axis has opposite direction if “mirrored”!

# ANNEX E: Signal conveying Line Types

Refer to NOAKA DEXPI Symbol legend file Ref: [7]

# ANNEX F: Valve Label Details

The attributes available within valve labels shall be determined based on whether the valve symbol is used to show a manual valve or an on/off actuated or controlled valve. Attribute details for the valve label is defined in the symbol legend details (Ref: [7]) using the attribute formats below.

* Manual Valves

A space character shall be used between the trim and lock mechanism attributes as shown.

**<ObjectDisplayName>**

**<NominalDiameter><VDS>**

**<TrimType> <LockMechanism>**

* On/Off Actuated or Controlled Valves

A space character shall be used between the trim and lock mechanism attributes as shown.

**<NominalDiameter><VDS>**

**<TrimType> <LockMechanism>**

Additional Notes:

* Valve Normal Position is no longer required to be shown in the label. Normal Position will be indicated by the symbol used e.g. black ⬄ closed , white ⬄ open
* Failure action is required to be shown in the label for the actuator if available Ref: [7]
* “Barrier information is not required in the label as this will be shown as part of the graphic” referring to DIB valves (examples below) and “Barrier information” the DP and SR notations. The following DIB valves will be added to the NOAKA DEXPI symbol legend soon.

**Table

Description automatically generated**

Figure 36: Valve Barrier variants

# ANNEX G: NOAKA DEXPI Notes Implementation

Within the NOAKA DEXPI project approval has been granted to use a non-DEXPI class type for Notes. This class type has been put forward to the DEXPI content group for approval in an upcoming version of DEXPI to provide support for drawing notations connected to conceptual model objects.

Diagram

Description automatically generated

Figure 37: Note class as a child of the ConceptualModel

Diagram

Description automatically generated

Figure 38: Note class with associaton to ConceptualObejct

# ANNEX H: DEXPI Connection Points & Associations

## Piping Connector

Within the piping system the piping connectors as defined by the conceptual DEXPI model have a 1:1 mapping with the graphical model. This allows the standard to simplify the export and NOT require the use of Associations to define relationships between conceptual elements, as this is already provided in Proteus export via graphical connections where:

* Node Type=’Process’ for piping ConnectionPoints Nodes
* Proteus XML Node element belonging to the ConnectionPoints XML node for the connected element represents the PipingNode conceptual element of the graphically represented element (has a SymbolReferenceID)
* The Proteus PipingNetworkSegment CenterLine child coordinates together with the coordinates of the nodes of the graphically connected elements provides the graphical connection information for the piping connectors.

A picture containing text

Description automatically generated Timeline

Description automatically generated

* Link between PipingNetworkSegment Connection From/To\* Node & PipingNodeOwner: ConnectionPoints / Node\*\*

\* From/ToNode is not required where the connection is to a Nozzle

\*\* based on node (zero-based) index of PipingNodeOwner / Connection Points

**Calendar

Description automatically generated with low confidence** **Timeline

Description automatically generated**

## Instrument Connector

For Instrumentation connections the conceptual model as defined by DEXPI is not a 1:1 mapping to the graphical representation of the connection and therefor the standard requires the use of Associations to define relationships between conceptual elements in addition to the graphical representation of the connections where:

* Node Type=’Signal’ for instrument ConnectionPoints Nodes
* Proteus XML Node element belonging to the ConnectionPoints XML node for the connected element represents the graphical connection points for the graphically represented element (has a SymbolReferenceID)
* Proteus InformationFlow has associations ‘has logical end’ & ‘has logical start’ to define the conceptual model element connections.
  + Where the connected conceptual element will have the association ‘is logical end of’ or ‘is logical start of’
* The Proteus InformationFlow CenterLine child coordinates together with the coordinates of the nodes of the graphically connected elements provides the graphical connection information for the instrumentation connectors.

A picture containing text

Description automatically generated Timeline

Description automatically generated with low confidence

* When the Proteus InformationFlow has ComponentClass=’MeasuringLineFunction’ it is required that the related ProcessSignalGeneratingFunction class element shall have an association ‘is located in’ to define the conceptual model element sensing connection.
  + Where the SensingLocation conceptual element (Nozzle, PipingComponent or PipingNetworkSegmenet) will have the association ‘is the location of’