Interoperability in petroleum production plants: A case study on the DEXPI P&ID specification

Rafael H. Petry[1], Nicolau O. Santos[1], Fabrício H. Rodrigues[1], Haroldo R. S. Silva[1], Régis K. Romeu[1], Haroldo R. S. Silva[1], David Cameron[2], Mara Abel[1] and João C. Netto[1]

[1]*Informatics Institute, Federal University of Rio Grande do Sul (INF-UFRGS). Av. Bento Gonçalves, 9090 - Agronomia, Porto Alegre - RS, 91540-000, Brazil.*

[2]*dScience Centre for Computational and Data Science, University of Oslo. Oslo, Norway*

Abstract

This work investigates semantic and syntactic interoperability capabilities on a Piping and Instrumentation Diagram (P&ID) model that supports the refinement steps of an engineering plant. By mapping P&ID classes of the DEXPI representation model to the Offshore Petroleum Production Plant Ontology (O3PO), we aim to clarify and regulate the semantics of P&ID entities and structuring relations. Our analysis extends to a case study involving the Data Exchange in the Process Industry (DEXPI) specification, emphasizing the need for well-founded ontological relationships. While technical interoperability between DEXPI P&IDs and BFO-based domain ontologies is achievable, we identify challenges in semantic interoperability of the standard, including issues related to clarity, conciseness, extensibility, consistency, and essence. These challenges could bottleneck seamless system integration and pose adoption barriers for the DEXPI P&ID specification beyond the Computer Aided Engineering (CAE) vendor's use cases.

Keywords

Semantic Interoperability, Ontology, Piping and Instrumentation Diagrams (P&IDs), Oil and Gas, Data EXchange in the Process Industry (DEXPI)

1. Introduction

Piping and Instrumentation Diagrams (P&IDs) are documents employed to exchange information concerning installations, equipment, and related elements [1]. P&IDs utilize predefined symbols to illustrate pipes, process equipment, and control systems without adhering to scale

13th International Workshop on Formal Ontologies Meet Industry (2024). Part of the Joint Ontology Workshops (JOWO 2024), July 15–19, 2024, Drienerlolaan 5, 7522 NB Enschede, Netherlands

*Corresponding author.

†These authors contributed equally.

‡Technical support on DEXPI

rhpetry&inf.ufrgs.br (R. H. Petry); nicolau.santos@inf.ufrgs.br (N. O. Santos); fabricio.rodrigues@inf.ufrgs.br (F. H. Rodrigues); hrssilva@inf.ufrgs.br (H. R. S. Silva); regisromeu@ufrgs.br (R. K. Romeu); hrssilva@inf.ufrgs.br (H. R. S. Silva); davidbc@ifi.uio.no (D. Cameron); marabel@inf.ufrgs.br (M. Abel); netto@inf.ufrgs.br (J. C. Netto)

https://www.inf.ufrgs.br/~rhpetry/ (R. H. Petry); https://www.inf.ufrgs.br/site/docente/mara-abel/ (M. Abel); https://www.inf.ufrgs.br/site/docente/joao-cesar-netto/ (J. C. Netto)

0000-0001-6023-0826 (R. H. Petry); 0000-0003-0901-2465 (N. O. Santos); 0000-0002-0615-8306 (F. H. Rodrigues); 0009-0003-0594-4852 (H. R. S. Silva); 0000-0001-7765-086X (R. K. Romeu); 0009-0003-0594-4852 (H. R. S. Silva); 0000-0002-1172-478X (D. Cameron); 0000-0002-9589-2616 (M. Abel); 0000-0002-5350-1728 (J. C. Netto)

or geographical orientation. They are the most important documents for the maintenance and evolution of any installation in the industry, providing information essential for equipment manufacturing, installation, commissioning, start-up, and ongoing plant operation.

P&IDs have a long and evolving technical history, initially as physical drawings and later in digital formats, for Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) applications. In complex installations, such as petroleum plants, several companies may be involved in the refinement and upgrades of the diagrams, struggling to exchange data between different pieces of software from different vendors. Indeed, one important related problem is interoperability. Interoperability refers to the effective exchange of information and understanding to collectively pursue common objectives.

A remarkable initiative of the industry to provide homogeneous data interchange between P&ID projects from diverse vendors refers to the Data EXchange in the Process Industry (DEXPI) Association, which develops and promotes a common data exchange standard for the process industry, covering all phases of the process-plant lifecycle. The DEXPI solution includes: a conceptual model (the P&ID specification), an exchange format (the ProteusXML schema) and links to the Posc Caesar ontology (based on the ISO 15926). The conceptual model intends to organize and represent the entities that build industrial plants, while ProteusXML is the language that allows the formal description of the diagrams, and the links to the ontology provide useful definitions. Many important P&ID CAD/CAE software vendors are currently offering an option to import or export their data following the DEXPI standard and it is still an ongoing work.

Even if it contains some semantic capabilities, the DEXPI solution is basically a technical interoperability resource, focused on format data exchanging between commercial software. We recognize that the DEXPI solution conceptual model can be thought as a kind of weak ontology, but this is different from a well-founded ontology. The DEXPI solution also provides some linking for a third-party ontology, but only to provide some definitions. The DEXPI solution does not organically derive from this third-party ontology.

We have explored the DEXPI solution in the context of a Research and Development (R&D) project for a petroleum production digital twin[1]. More specifically, we have invested in trying to couple the DEXPI solution to a well-founded ontology developed in the project, named O3PO. After all, in our digital twin context, we had to be more ambitious about semantic interoperability.

So, we have mapped the DEXPI conceptual model classes to the ontological entities of the Offshore Petroleum Production Plant Ontology (O3PO), necessitating a thorough semantic analysis of the DEXPI model. Our primary objective is to present the findings of this analysis. This means to make explicit the gap between a mostly technical interoperability tool, good for some purposes, and a more complete semantic interoperability perspective, required for more advanced purposes. We observed some good matches, but also some challenges, like some vague, missing and overlapping definitions in the DEXPI model.

[1]More information at https://www.petwin.org/

2. Theoretical Background

2.1. Data Exchange, Data Integration, Data Interoperability and Semantic Interoperability

Interoperability is multidimensional, encompassing diverse perspectives and approaches across various application domains [2, 3, 4]. In the realm of interoperability between systems, it is commonly defined as the ability/capability of two (or more) systems or components to exchange information and to effectively use the information that has been exchanged [5, 6]. Other definitions emphasize capabilities such as sharing knowledge efficiently and safely [7], or that the exchange of data should occur with minimal loss of content and functionality [8].

Interoperability is also separated into different levels, distinguishing syntactic, technical and semantic interoperability:

- Syntactic interoperability refers to describing the exact representation of the information to be exchanged in terms of grammar and format [9].
- Technical interoperability is viewed as the capability of exchanging information [10]. It is associated with data integration services, data presentation, and exchange activities [9].
- Semantic interoperability is the capability of using the given information [10]. It guarantees that data exchange makes sense and that the parties involved have a shared understanding of the meanings of the data [11].

2.2. The Data EXchange in the Process Industry (DEXPI) Specification

The DEXPI initiative seeks to address interoperability challenges within the process industry by establishing standardized digital communication norms, facilitating seamless data exchange across the industry's life cycle, from development to maintenance [12]. One of it's key focuses is the standardized exchange of P&IDs across CAE tools from different vendors. The DEXPI P&ID specification is a well known format adopted by various CAE system providers. The objective of the specification is to allow vendors to exchange P&IDs, so that different systems can use the same data inside a company keeping data exchange costs to a minimal.

The design approach of separating specification (DEXPI) from implementation (ProteusXML) requires aligning P&ID files, initially in the ProteusXML format, with the DEXPI conceptual model. To effectively handle Proteus files within this framework, the P&ID Toolbox provides a comprehensive implementation of the DEXPI information model. Additionally, this toolbox streamlines the processes of importing, exporting, and visualizing DEXPI P&IDs as images. Figure 1 shows an example of a P&ID.

2.3. The Offshore Petroleum Production Plant Ontology (O3PO)

The Offshore Petroleum Production Plant Ontology (O3PO)[2] is a public reference domain ontology of petroleum plants that uses BFO[14] as a top-level ontology [15]. O3PO is the product

[2] Avalaible at https://github.com/BDI-UFRGS/O3POntology.

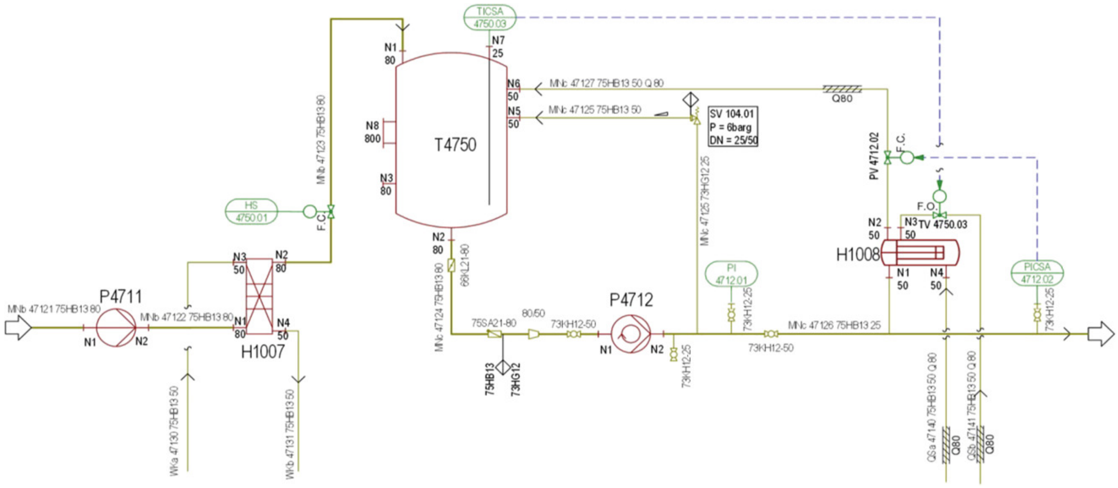

Figure 1: Example of a P&ID from [13]

of an industry-academia project in Brazil called PeTWIN[3]. The project focused on developing digital twins for offshore petroleum production plants, including the semantic infrastructure. Besides using BFO, O3PO imports the core ontology produced by Industrial Ontologies Foundry (IOF-Core) and the core ontology for Geosciences (GeoCore). These middle-level ontologies also use BFO as a top-level ontology.

There are three main uses for O3PO: (1) to define what are the main types of entities between the subsurface reservoir and the surface platform; (2) to determine the main properties that inhere on those entities; and (3) to follow the fluid path from the reservoir to the platform[15]. O3PO was initially created to deal with the substantial number of time-series data sources in an offshore oil plant, from surface facilities to subsea equipment and wells in Brazil's Pre-Salt. Since then, several uses for ontology have been discovered, and it has continually grown.

3. Interoperability Analysis of DEXPI P&ID Specification

In this section we will present how the different subdivisions of interoperability (i.e. syntactic, technical and semantic interoperability) manifest themselves in the DEXPI P&ID specification. Showcasing the tools, descriptions and situations in where the specification achieves the different interoperability and where it has some room for improvement.

3.1. Syntactic and Technical Interoperability Analysis

Syntactic interoperability requires an exact and well-defined representation format for information to be properly exchanged. The DEXPI P&ID standard currently adopts the ProteusXML[4] schema as it's exchange format. The standard's documentation also point direct mappings from

[3]More information at https://www.petwin.org/
[4]Available at: https://github.com/ProteusXML

the DEXPI Information Model to ProteusXML patterns [12]. The pnb Toolbox Library[5] can be used in order to map from the ProteusXML format to the DEXPI Information Model without the need of implementing the parser and mapping manually.

Technical interoperability, on the other hand, is associated with data integration services and data exchange activities. On that matter, many CAE systems provide automated services to import/export P&IDs using the DEXPI P&ID standard[6]. This brings the possibility of third party applications to exchange P&ID data with such CAE systems without the need for vendor-specific integration.

3.2. Semantic Interoperability Analysis

Semantic interoperability requires that the parties involved have a shared understanding on the meaning of the exchanged data. The DEXPI P&ID specification presents several definitions to contextualize the elements in a process plant, and many of them are based on definitions from POSC Caesar[7]. Despite that, there are flaws in the definitions of many other terms that compose the specification, which hinders its semantic interoperability capabilities. We analyze some of these issues below.

Vague definitions Certain elements within the DEXPI standard exhibit vague definitions and require a comprehensive grasp of the standard's structure and access to industry-specific glossaries for proper comprehension. Examples of this issue include the definitions of **DEXPI:Piping Component** as *"A piping component"*, and **DEXPI:ColumnSection**, defined as *"A column section"*, lacking contextualization or specific and explicit identity criteria for the identification of its instances.

With vague definitions, the interpretation is left to the reader of the model, which can lead to issues if the recipient of the information has a conceptualization different from the sender's. For instance, a pump is classified as 'equipment' in the DEXPI standard and is likely distinct from 'piping components', yet some consider it as a piping component. This raises questions about what exactly constitutes a piping component. If the definition is limited to only tubes, then valves would not be considered piping components; however, if it includes any item through which a fluid passes, then valves, tubes, and even the insulation of tubes would be classified as components. Conversely, thermal insulation would not qualify as a component if the definition restricts it to items where the fluid directly contacts their walls. This ambiguity in classification highlights the need for precise definitions to ensure uniform understanding and application across different contexts.

Missing definitions Despite the large number of classes in the DEXPI specification, such as **DEXPI:PipingComponent**, **DEXPI:PipingNetworkSegment**, and **DEXPI:PipingNode**, which define elements in terms of being part of a piping, the standard lacks a clear definition of what constitutes a "piping". This absence of definition poses challenges when attempting to state,

[5] Available at: https://www.plants-and-bytes.de/en/p-id-and-dexpi
[6] Available at: https://dexpi.org/software/
[7] Available at: https://data.posccaesar.org/rdl/

for instance, that **DEXPI:PipingComponent** is a piping component, as the concept of a piping system is not explicitly defined.

Consequently, one must resort to terms like **O3PO:Pipeline** as an approximation to the implicit DEXPI notion of what a piping system entails.

Another case of missing definitions seems to be the **DEXPI:PipeFitting**, defined as *"A pipe fitting"*, without resorting to any external definition. By analyzing its subclasses, the class seems to refer to passive elements assembled between two pipes, like strainers, fittings, and line blinds, that we could categorize as valves.

Collapsed definitions An illustrative example of a lack of definition and collapsed terms is **DEXPI:SafetyValveOrFitting**, denoted in the specification as *"A safety valve or fitting"*. From an ontological viewpoint, this definition muddies the distinction between different types of entities: pipe fittings and valves, both serving the role of ensuring safety. Within the O3PO model, the sole common class encompassing pipe fittings and valves is **IOF:MaterialArtifact**. Had the standard implemented separate classes for each, it would have enabled the differentiation between the elements that define their identity and those that specify their utility. The absence of a valve class raises further questions, particularly regarding how one might quantify the number of valves within a plant. Despite the seemingly straightforward nature of this query, it necessitates a manual count of instances, encompassing **DEXPI:OperatedValves**, **DEXPI:Check Valve**, and non-safety fitting instances of **DEXPI:SafetyValveOrFitting**. Given that these instances are distributed among various subclasses of **DEXPI:PipingComponent** and no clear criteria exist within the **DEXPI:CustomSafetyValveOrFitting** class to differentiate between valves and fittings, this tallying process necessitates human intervention.

Contradictory definitions In DEXPI specification, a Piping Network Segment is defined in POSC Caesar as something composed of inanimate physical objects. In DEXPI, a Piping Network Segment Item is defined as something that can be part of a Piping Network Segment. However, in DEXPI, Piping Network Segment Item is subclassed as PipeOffPageConnector, which is an "Abstract Class" defined as "A connector that indicates that a piping network segment is continued elsewhere, either on the same PID or on another PID. Graphically, it is usually represented as an arrow"

Custom Object Role One parent class present across the DEXPI specification is the **DEXPI: CustomObject** (defined as *"The abstract base class of all custom classes"*), which differentiates classes initially defined in the standard from user-defined ones. The standard currently counts 57 custom classes, all sharing the same pattern.

The issue with custom classes is that instead of relying on the entity responsible for defining a class like "this class is defined in DEXPI 1.3" or "this class was defined by organization X," its members are defined in a case-by-case manner, using the same pattern of "a **DEXPI:CustomY** is a Y and is not covered by any of the other subclasses of Y." This means that to support the representation of any of the custom classes in DEXPI, one needs to be able to represent and differentiate all other subclasses of the parent class.

Differentiability of equipment As far as redundancy goes, there are also classes like **DEXPI:TaggedColumnSection** that exist to convey that an item is a **DEXPI:ColumnSection** and is tagged. In the same way, as in previous examples, it is quite interesting to note that a column section is still the same whether it is tagged or not, and the only difference is this relation with a tag that does not affect the identity of the column. The precision and clarity of a model presuppose that the entity has a single representation and that non-specialization properties, like "having a tag," should not define new entities but be inserted as dependent properties of the entity.

The same definitions in different places Within the DEXPI specification, the class **DEXPI: InlinePrimaryElement** (defined as *"An inline primary element"*) encompasses ten distinct subclasses. Two of these subclasses stand out: the **DEXPI:FlowMeasuringElement** and the **DEXPI: ElectromagneticFlowMeter**. The former is defined as as *"A FLOW MEASURING ELEMENT is a MEASURING ELEMENT that measures a FLOW RATE"*. In contrast, the latter is *"A velocity flow meter that measures the flow rate of a conductive fluid running through a magnetic field by measuring the charge created when fluid interacts with the field"*.

This scenario perfectly illustrates a case where an electromagnetic flow meter inherently functions as a flow-measuring element due to its ability to measure flow. However, in the DEXPI specification, both classes are categorized as direct subclasses of **DEXPI:InlinePrimaryElement**, thus implying that we can not classify one entity under both classes.

A similar scenario arises when considering **DEXPI:HeatExchanger** and **DEXPI:CoolingTower**. The former is *"An apparatus or machine that has the capability of heat exchanging"*. In contrast, the latter is defined as *"A cooler and an air-cooled heat exchanger that is a tall structure through which air circulates by convection"*. Notably, the definition of **DEXPI:CoolingTower** explicitly states that it is an *"air-cooled heat exchanger"*. Despite this clear relationship, the DEXPI standard does not classify **DEXPI:CoolingTower** as a subclass of **DEXPI:HeatExchanger**, showcasing a discrepancy between the definitions and the classification within the standard.

4. Experimenting with DEXPI P&ID data importing

As demonstrated, DEXPI facilitates syntactic and technical interoperability but falls short in providing semantic interoperability. To address this gap, an experiment was conducted to showcase potential solutions. The experiment utilized the O3PO ontology alongside a subset of DEXPI packages, specifically those whose terms overlap with O3PO categories. This approach allowed for a deeper integration of data and systems by aligning terms from different standards under a unified semantic framework.

One of our goals is assessing to which degree one can achieve semantic interoperability between process plant descriptions adhering to DEXPI and O3PO. In the context of offshore oil and gas production, both models depict some aspects of reality in common. Also, the capacity of converting elements described in the DEXPI standard to an ontological model provides us access to tools such as query languages and tools common to the ontology community, such as graph databases and query languages, reasoning, fact checking and so on.

Both the standard data model and O3PO cover, to some degree, physical quantities, component types, and connection topology. If these data *could* be seamlessly integrated, one could easily make a query for pressure data (with the aid of the Information Artifact Ontology (IAO) alongside O3PO) of all elements along the piping (piping and connections represented in DEXPI).

Within the framework of the DEXPI standard, distinct packages focus on various facets of a production plant integration possibilities within specific packages. Package Piping stands out for its depiction of connection topology, while both Package Equipment and Package Piping contribute by outlining diverse equipment types within the plant. Additionally, Package PhysicalQuantities proves valuable in elucidating the array of physical quantities associated with production elements, such as surface areas of frequencies.

To assess the overlap of intended models between the conceptualizations of DEXPI and O3PO, a classification process is proposed to establish a depiction of DEXPI classes with O3PO classes and relationships, using elements from its imported ontologies when necessary. We selected the DEXPI P&ID specification packages based on their apparent overlap with represented elements in O3PO. Package piping and equipment were chosen for this study, aligning with the component types and relations in O3PO.

After a more in-depth analysis of the selected DEXPI packages, the most representative classes identified were PipingComponent and Equipment. From there, their subclasses were analyzed down the taxonomy, stopping whenever new terms brought out the same meaning as their parent classes. In addition, other classes help to explain dependent concepts. While some of the established correspondences rely on concepts directly defined in O3PO, many require definitions from O3PO's imported ontologies, such as IOF-Core for more general terms like assemblies and systems or BFO for capabilities and functions. It is essential to note that we considered only classes and relations defined in these important ontologies and imported by O3PO for the analysis.

By applying this methodology, we have formulated O3PO classifications for some DEXPI piping and equipment classes (exemplified by Table 1), while simultaneously pinpointing the specific challenges for the cases that render the classification unfeasible.

Consider the case of the **DEXPI:PipingComponent**, as outlined in the DEXPI specification as *"A piping component"*. Given the vague nature of its definition, by taking into consideration its subtypes and the understanding that P&IDs primarily illustrate assemblies rather than individual components, one can classify **DEXPI:PipingComponent** as a **BFO:MaterialEntity** and BFO:bearer_of some **BFO:Function**, and IOF:prescribedBy some **IOF:DesignSpecification** and O3PO:component_of some **O3PO:Pipeline**.

Based on that, we can also specify **DEXPI:InlinePrimaryElement**, defined in the DEXPI specification as *"An inline primary element"*. Since the definition is also lacking specificity, the fact that it is a subclass of **DEXPI:PipingComponent** and all its subclasses are used to perform some measurement, we can say that a **DEXPI:InlinePrimaryElement** is a **BFO:MaterialEntity**, and a BFO:bearer_of some **BFO:Function**, and is IOF:prescribedBy some **IOF:DesignSpecification**, and a O3PO:component_of some **O3PO:Pipeline**, and IOF:hasCapability some **IOF:MeasurementCapability**.

Even though all the elements represented in a P&ID diagram are part of piping, that seems to be the only difference between **DEXPI:Equipment**, defined as *"An apparatus or machine"*, and **DEXPI:PipingComponent**.

Through this classification, we clarified the semantics of the entities represented in a P&ID and provided support for the operator to understand the meaning and intrinsic restriction of

Table 1
DEXPI to O3PO Classification Examples

Concept name	DEXPI Definitions	Description
Piping Component	A piping component.	**DEXPI:PipingComponent** is_a **IOF:MaterialArtifact** or **IOF:EngineeredSystem** and (O3PO:component_of some **O3PO:Pipeline**).
Piping Network Segment Item	An item that can be part of a PipingNetworkSegment.	**DEXPI:PipingNetworkSegmentItem** is_a **IAO:Identifier** or **IAO:Symbol** or (**BFO:MaterialEntity** and O3PO:component_of some **O3PO:Pipeline**).
Pipe Fitting	A pipe fitting.	**DEXPI:PipeFitting** is_a **IOF:MaterialArtifact** and O3PO:component_of some **O3PO:Pipeline**).
Inline Primary Element	An inline primary element.	**DEXPI:InlinePrimaryElement** is_a **IOF:MaterialArtifact** and BFO:bearer_of some **BFO:function**) and (IOF:prescribedBy some **IOF:DesignSpecification**) and (O3PO:component_of some **O3PO:Pipeline**).
Equipment	An apparatus or machine.	**DEXPI:Equipment** is_a (**IOF:MaterialArtifact** or **IOF:EngineeredSystem**) and (BFO:bearer_of some **BFO:function**) and (IOF:prescribedBy some **IOF:DesignSpecification**).
Operated Valve.	A valve that includes an external means of operation. (E.g. handwheel / lever / actuator.) (from http://data.posccaesar.org/rdl/RDS11141590).	**DEXPI:OperatedValve** is_a **O3PO:Valve** and BFO:bearer_of some **BFO:function**) and (**IOF:prescribedBy** some IOF:DesignSpecification) and (O3PO:component_of some **O3PO:Pipeline**) and (IOF:hasCapability some **O3PO:BeingOperatedCapability**).

the modeled entities. Also, O3PO supports a description of the DEXPI entities while utilizing domain-specific vocabulary tailored to the Oil and Gas industry. Moreover, this approach equips us with a foundational framework built upon ontological principles, fostering open accessibility, collaborative development, and enhanced interoperability.

Further possibilities remain open for mapping the two models, such as establishing a correspondence between DEXPI and O3PO topology models. DEXPI's topology is structured around nodes associated with elements and their interconnections, whereas O3PO presents its topology through connections and fluid supply properties.

Another possibility is the use of O3PO for the creation of a Comprehensive Information Base (CIB), integrating data from various sources, including DEXPI P&IDs. By allowing the complementing of the data in the P&ID with other information sources, we could bridge the gap between classifications. This enhancement could help accurately categorize valve types such as GasLiftValve, ChokeValve, or InflowControlValve.

Integrating the DEXPI standard with the O3PO presented several challenges, including imprecise term definitions, potentially redundant or overly vague definitions that rely on previous understanding of those familiar with the subject matter, the absence of a superclass in definitions, the complexities of multiple inheritances and collapsed entities or relations. These limitations lead us to search the correspondence between the entities primarily based on the entity name, a common practice for mapping models that results in false agreements. This section will elaborate on these challenges to offer a more comprehensive understanding.

Representing elements from the DEXPI standard in OWL using the O3PO domain ontology affords a dual advantage. From the technical interoperability perspective, it enables the execution

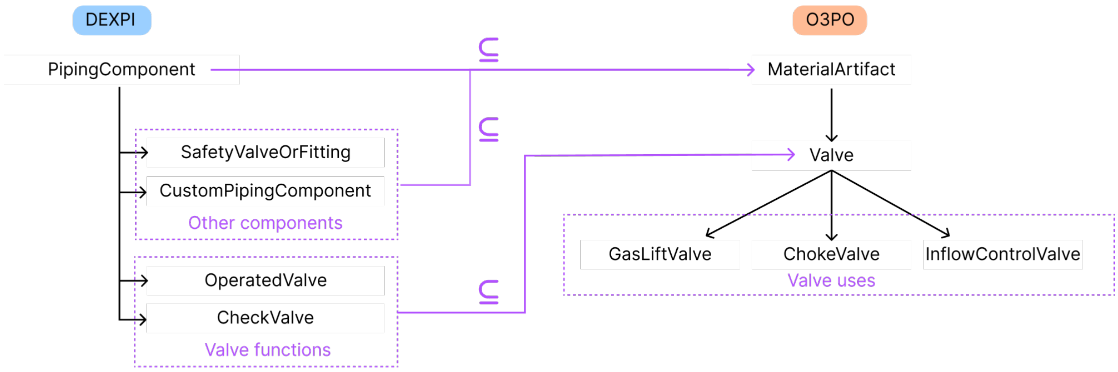

Figure 2: Gaps in direct mapping between DEXPI and O3PO valves

of queries leveraging languages such as SPARQL, facilitating information retrieval from process plants structured in alignment with the DEXPI standard. Simultaneously, from a semantic standpoint, it empowers the utilization of a domain-specific vocabulary and conceptual framework to encapsulate plant-related data. Beyond these advantages, it facilitates seamless integration and aggregation of information from diverse sources and supports fact-checking and reasoning for instances within the O3PO context. Figure 3 portrays the diverse information transitioned from the initial P&ID to the final O3PO instances, showcasing the potential outcomes achievable by mapping topological correlations and equipment between O3PO and DEXPI.

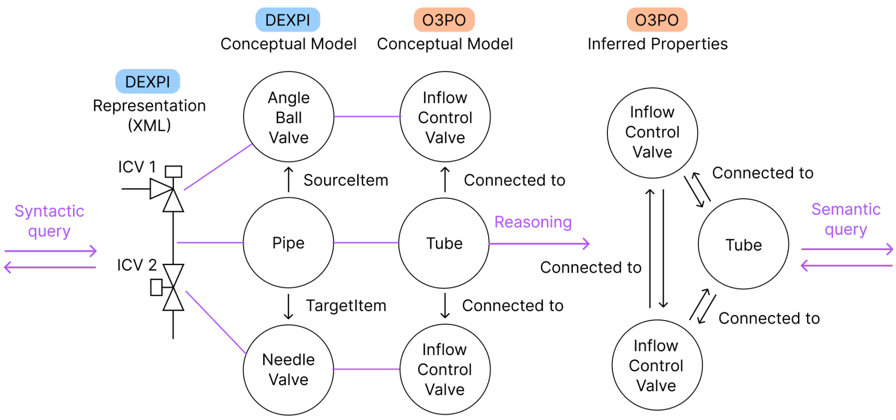

Figure 3: Enabling domain-specific semantic queries in P&IDs

To address the translation of data from the DEXPI file format (ProteusXML) to a conceptual model the pnb Toolbox Library, offered by pnb plants & bytes[8], was used, simplifying the complexities of the format. This data was then converted into an N-Triples file using specific mapping rules, focusing on elements from the DEXPI equipment and piping specifications. Using the Hermit reasoner, we processed this data to identify inconsistencies and generate new information.

[8]Available at: https://www.plants-and-bytes.de/en/

An application was developed to validate this process, taking a ProteusXML file and converting it into triples that were further mapped to the O3PO model using parser. The conversion and reasoning capabilities were enhanced with a custom module developed using the Owlready2 Python library[9], which supports sophisticated ontology-based applications [16]. This setup effectively maps and categorizes plant components such as pipes and valves, facilitating detailed information extraction and validation of the mapped data.

Upon receiving a ProteusXML file adhering to the DEXPI specification that delineates a plant featuring components like pipes, valves, separators, and nozzles, the application undertakes the task of mapping these elements to the O3PO model and subsequently infers their inherent attributes.

Subsequently, following the described procedures, the process of categorizing DEXPI classes into corresponding O3PO classes is executed, followed by reasoning to extract comprehensive information associated with the identified elements.

5. Concluding Remarks

This study approaches interoperability from multiple perspectives by facilitating technical interoperability between the P&ID representation standard (DEXPI) utilized by diverse software vendors and domain ontologies like O3PO tailored to represent the oil and gas production plant environment. Additionally, the study highlights instances where semantic interoperability between DEXPI and O3PO conceptualizations is feasible and instances where it is not.

While various tools are available to handle standards and technologies, the practical feasibility of achieving technical interoperability between DEXPI P&IDs and ontologies represented in OWL becomes evident. However, the predominant challenges remain within the realm of semantic interoperability. The presence of ambiguous and often vague definitions poses a significant hurdle to seamless systems integration.

As a potential avenue for further contribution, future work could extend this approach of categorizing instances into well-founded ontologies to encompass the entire DEXPI standard by taking advantage of different ontologies to represent various aspects of the standard, thus serving as a valuable guide for future standards development and patching. Although the primary focus of the DEXPI revolves around enhancing interoperability between CAE tools, it is increasingly observed in the literature that various other applications adopt the description as a standard. These applications leverage the available tooling and the standard's publicly available documentation. However, if the DEXPI standard were to evolve into a de facto standard for representing P&IDs, the highlighted issues could potentially bottleneck its adoption and hinder the ease of integration into different systems.

References

[1] M. Toghraei, Piping and instrumentation diagram development, John Wiley & Sons, 2019.

[2] T. C. Ford, J. M. Colombi, S. R. Graham, D. R. Jacques, A survey on interoperability measurement, Gateways 2 (2007) 3.

[9] More information at: https://owlready2.readthedocs.io/en/latest/

[3] D. Gürdür, F. Asplund, A systematic review to merge discourses: Interoperability, integration and cyber-physical systems, Journal of Industrial Information Integration 9 (2018) 14–23. URL: https://www.sciencedirect.com/science/article/pii/S2452414X17300687. doi:https://doi.org/10.1016/j.jii.2017.12.001.
[4] Žiga Turk, Interoperability in construction – mission impossible?, Developments in the Built Environment 4 (2020). URL: https://www.sciencedirect.com/science/article/pii/S2666165920300144. doi:https://doi.org/10.1016/j.dibe.2020.100018.
[5] IEEE, Ieee standard computer dictionary: A compilation of ieee standard computer glossaries, IEEE Std 610 (1991) 1–217. doi:10.1109/IEEESTD.1991.106963.
[6] ISO, ISO 25964-2:2013: Information and Documentation : Thesauri and Interoperability with Other Vocabularies. Interoperability with other vocabularies, ISO, 2013. URL: https://books.google.com.br/books?id=703ooAEACAAJ.
[7] L. F. Garcia, V. Graciolli, L. F. D. Ros, M. Abel, A conceptual framework for rock data integration in reservoir models based on ontologies, Int. J. Monit. Surveill. Technol. Res. 5 (2017) 71–82. URL: https://doi.org/10.4018/IJMSTR.2017010104. doi:10.4018/IJMSTR.2017010104.
[8] J. Riley, Understanding metadata, Washington DC, United States: National Information Standards Organization (http://www. niso. org/publications/press/UnderstandingMetadata.pdf) 23 (2017) 7–10.
[9] E. Commission, D.-G. for Informatics, New European interoperability framework : promoting seamless services and data flows for European public administrations, Publications Office, 2017. doi:doi/10.2799/78681.
[10] T. Benson, G. Grieve, Why Interoperability Is Hard, Springer International Publishing, Cham, 2021, pp. 21–40. URL: https://doi.org/10.1007/978-3-030-56883-2_2. doi:10.1007/978-3-030-56883-2_2.
[11] S. Heiler, Semantic interoperability, ACM Comput. Surv. 27 (1995) 271–273. URL: https://doi.org/10.1145/210376.210392. doi:10.1145/210376.210392.
[12] D. Initiative, Dexpi p&id specification, Version 1.3 (2021).
[13] M. Wiedau, L. von Wedel, H. Temmen, R. Welke, N. Papakonstantinou, Enpro data integration: Extending dexpi towards the asset lifecycle, Chemie Ingenieur Technik 91 (2019) 240–255. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cite.201800112. doi:https://doi.org/10.1002/cite.201800112.
[14] J. N. Otte, J. Beverley, A. Ruttenberg, Bfo: Basic formal ontology, Applied Ontology 17 (2022) 17–43. URL: https://doi.org/10.3233/AO-220262. doi:10.3233/AO-220262, 1.
[15] N. O. Santos, F. H. Rodrigues, D. Schmidt, R. K. Romeu, G. Nascimento, M. Abel, O3po: A domain ontology for offshore petroleum production plants, Expert Systems with Applications 238 (2024) 122104. URL: https://www.sciencedirect.com/science/article/pii/S0957417423026064. doi:https://doi.org/10.1016/j.eswa.2023.122104.
[16] L. Jean-Baptiste, Ontologies with Python: Programming OWL 2.0 Ontologies with Python and Owlready2, Springer, 2021.