Systems & Control Transactions

Original Research Article
Peer Reviewed Conference Proceeding
ESCAPE 36 - European Symposium on Computer Aided Process Engineering
Sheffield, UK. 21-24 June 2026

PSE PRESS

Semantic PEA Datasheets for digitalised modular plant documentation

Sascha Lamm[a]*, Sebastian Tecl[a], Ingo Dietrich[b], Sissy Sommer[b], Markus Heinbücher[a], Peter F. Pelz[a]

[a] Technical University of Darmstadt, Chair of Fluid Systems, Darmstadt, Germany
[b] Industrial Science GmbH, Darmstadt, Germany
* Corresponding Author: sascha.lamm@tu-darmstadt.de.

ABSTRACT

Modular plants emerged as the key solution for reducing time-to-market and increasing flexibility in the process industry by combining different modules known as Process Equipment Assemblies (PEAs). While PEA automation is standardised through the Module Type Package (MTP), comparable tools for their documentation remain absent. This work presents the Semantic PEA Datasheet (SPEAD) ontology, which represents PEA documentation as a machine-readable knowledge graph that adheres to the FAIR principles. SPEAD integrates established standards such as DEXPI and the VDI 2776 guidelines and ensures data quality through comprehensive annotations and constraint-based validation. The ontology was evaluated against twelve competency questions derived from a representative use case as well as competency questions from the literature using a continuous stirred-tank reactor PEA as well as a dosing PEA as example systems. SPEAD successfully covers operational and design parameters as well as interface definitions, and provides provenance tracking through rich metadata. Comparison with existing information models OntoCAPE and DEXPI shows that SPEAD provides enhanced provenance tracking essential for trust in engineering documentation. Future work includes enriching capability and safety descriptions and exploring integration with the Asset Administration Shell (AAS).

Keywords: Information Management, Modelling, Industry 4.0, Documentation, Modular Plants, Ontology, Knowledge Graphs

INTRODUCTION

In recent years, modular process plants have emerged as a key solution to tackle the growing need for reduced time-to-market and increasing flexibility in the process industry [1]. Modular plants are built by combining several modules, known as Process Equipment Assemblies (PEAs) [2]. PEAs perform single process steps, such as dosing or heating [2]. By combining and re-combining different PEAs, the Owner/Operator (O/O) of the plant can adapt their plant to fit the requirements of the product to be produced [2]. This process is called orchestration and replaces detailed equipment engineering with module selection [2].

Guidelines for the design of modules, their automation both on PEA and plant level, and the documentation of the modules are already established. The design of a PEA depends on the vendor, the operator, and general design requirements and is thus not unified across the industry [3]. The automation of a module is performed in a standardised manner using the Module Type Package (MTP) [4]. However, no comparable standard tool for the documentation of modules exists to date. In the guideline VDI 2776-1 [2], minimum requirements for the content of the documentation are given. The guideline further recommends the provision of documentation in a modular, machine-readable format. Guideline VDI 2770 provides such a format [5], which defines the minimum requirements for the exchange of digital manufacturer documentation. Neither guideline specifies how these requirements should be fulfilled, nor how information should be represented. This may lead to inconsistent data, metadata and contradictory interpretation of data. In order for the documentation to be usable across the lifecycle of a PEA, it must contain information on all aspects of the PEA and provide clear, unambiguous definitions of critical terminology and data.

Research Objective

To address this gap, we previously proposed the Semantic PEA Datasheet (SPEAD) [6]. SPEAD utilises semantic web technologies to represent the documentation of a PEA as a knowledge graph.

In this work, we present the SPEAD ontology, focusing primarily on the modelling of physical properties. This paper is structured as follows: First, we present current documentation standards and guidelines, semantic web technologies, and related models and standards. Next, we present our modelling methodology and the resulting ontology. Afterwards, we evaluate our model against competency questions developed during the modelling phase using a continuous stirred-tank reactor PEA and a dosing PEA as examples. Finally, the results are discussed and a conclusion is given.

Documentation Standards and Guidelines

Guideline VDI 2776-1 recommends that the documentation of a modular plant is hierarchically structured in the same way as its modular structure [2]. Documentation for PEAs, their contained functional equipment assemblies (FEAs), and additional components should be stored in a database that enables cross-referencing between corresponding entities. Updates performed on any element are thus inherited by higher level elements.

Guideline VDI 2776-2 recommends that PEA documentation contains at least (1) an overview of implemented process functions, (2) an overview of implemented services, (3) the interface requirements both towards adjacent PEAs as well as the (non-modular) environment, and (4) a presentation of the intended area of use. The PEA documentation should reflect its life cycle [3]. Services should be described according to the MTP [4], and a P&ID should be provided in an open data format such as DEXPI. Furthermore, it is recommended that documentation be supplied as modular digital information with an open data format, and that the exchange format described in guideline VDI 2770-1 be used. In this guideline, documents are classified according to their purpose and stored alongside relevant metadata such as ID and provenance in containers [5].

Lastly, Regulation 2023/1230 of the European Parliament and of the Council (commonly “Machinery Directive”) demands certain information to be present in the technical documentation in accordance with harmonised standards or common specifications, such as design and manufacturing drawings [7]. Where these standards and specifications have not been or have only been partially applied, descriptions of the utilised standards or specifications must be provided [7].

Semantic Web Technologies

Engineering documentation for modular plants should adhere to the FAIR principles to ensure data is Findable, Accessible, Interoperable, and Reusable throughout the plant lifecycle and across organisational boundaries. Semantic web technologies provide the standardised framework necessary to achieve these goals through structured, machine-readable knowledge representation. In the following paragraph we introduce the core W3C standards that form the technical foundation of our ontology.

The Resource Description Framework (RDF) is the core technology for the semantic web. It allows information about resources, e.g., documents, physical objects, etc., to be expressed in the form of triples [8]. Triples always have the form subject-predicate-object, where the subject and object are the resources being related, and the predicate is their relationship. A collection of these triples forms a directed labelled graph in which the subject and object are nodes and relationships are edges. RDF enables the creation of vocabularies and ontologies that provide reusable definitions of common terminology. The creation of ontologies is aided by the Web Ontology Language (OWL). OWL itself is an RDF vocabulary and supports the publication and sharing of ontologies by providing abstract constructs for the definition of classes, properties, restrictions, and logical axioms [9]. To ensure data quality and consistency, RDF graphs can be validated against a set of constraints using the Shapes Constraint Language (SHACL). Shapes are defined to specify how targeted class instances and properties can be constructed. These constraints include but are not limited to data type, cardinality, and logical constraints [10]. In this work we use the Terse RDF Triple Language (Turtle) syntax for serialising RDF data due to its human-readable format and widespread support.

Related vendor-independent standards

Koch et al. [11] evaluated existing vendor-independent standards in information modelling for process engineering, identifying three key standards: OntoCAPE, DEXPI, and IEC 61512.

1. OntoCAPE [12] is an ontology covering the domain of computer aided process engineering. It consists of multiple modules spread across six layers, each covering different topics across the process engineering domain.
2. The DEXPI (Data Exchange in the Process Industry) specifications [13] provide two complementing data models, DEXPI Plant and DEXPI Process. DEXPI Plant is a standardised data model for P&IDs, whereas DEXPI Process is a data model for process engineering information. Both specifications aim to enable consistent data exchange in the process industry. Both were recently unified into the DEXPI 2.0 model.
3. The IEC 61512 standard [14] deals with batch control and defines three models: a process model, a physical

model, and a procedural model for linking automation, operations and procedures to the other models. It defines three types of control and distinguishes between different recipe types.

Based on their evaluation, OntoCAPE and the DEXPI specifications were identified as the most suitable information models for modular plants [11]. However, neither model should be considered as a standalone solution. Instead, Koch et al. suggest the integration and extension of these models towards an ontology for modular plants [11]. The goal of this work is to lay a foundation for such an ontology, with the focus of applying the model to the documentation of modular plants.

METHODOLOGY

Development Approach

The SPEAD ontology was modelled using the use case- and competency question-based method by Kendall and McGuinness [15]. Competency questions (CQ) are derived from the use case and represent requirements for the knowledge graph that must be fulfilled by it. A representative, iteratively developed use case for the SPEAD ontology is given in the following.

Use case: Exchange of PEAs in a modular plant

> A service technician is charged with replacing the reactor PEA from an existing plant with a new one from a pool of available reactor PEAs. To do this, they require information about the operating conditions, structure, and available services of all PEAs in the pool.

The derived competency questions CQ1 – CQ8 are listed in Table 1. These CQs focus on the content of the information model. Based on the CQs, lists are created which contain the necessary terminology to answer the CQs. For example, to answer CQ1, terms such as “PEA” and “operating pressure” must be defined. Further, a representation for quantitative properties in the format name/symbol = value-unit, e.g., vessel diameter D = 300 mm, must be established. To further organise the domain knowledge, we group the information into five categories: (i) operating range, (ii) construction, (iii) interfaces, (iv) services, and (v) safety. In this work we focus on (i) and (ii) and only briefly introduce our concept for (iii), (iv), and (v).

In addition, we derived questions regarding the provenance and terminology of the model, which is shown in Table 2. The provenance-related questions enable the model to capture information that is critical for regulatory compliance, change management, and establishing accountability in collaborative engineering environments. The terminology-related questions ensure that domain concepts are consistently defined, thus enabling unambiguous communication between stakeholders and interoperability between systems. Together these CQs address the metadata layer essential for trust, traceability, and quality assurance in engineering documents.

Table 1: Competency questions and the expected answers regarding the content of the information model.

CQ	Competency Question	Expected answer
1	What is the maximum operating pressure of this PEA?	The maximum operating pressure of this PEA is X bara.
2	List the boundaries of the operating temperature of this PEA.	A table with the desired boundaries.
3	Find all PEAs with a maximum operating pressure larger than 20 bar and a minimum operating temperature below 10°C.	A list of IDs of PEAs that fulfil the criteria.
4	What is the total volume of the reactor vessel of this PEA, and how much of it is heated?	The total volume and the heated volume of the reactor vessel.
5	Which FEAs and components does this PEA contain?	A list of FEAs and components of this PEA.
6	How many inlet interfaces does this PEA have that can be used for educts or flushing?	The count of inlet interfaces with the specified function.
7	What is the IP rating of this PEA?	The IP rating of this PEA.
8	Which MTP services are available at this PEA?	A list of services this PEA provides.

Table 2: Competency questions and the expected answers regarding the terminology and provenance of information inside the model.

CQ	Competency Question	Expected answer
9	Which document is the source of this value?	Reference to the source document.
10	When was this value added to the documentation of this PEA?	The date on which the value was added.
11	What is the definition of the class *spead:ProcessEquipmentAssembly*?	The definition text from the ontology.
12	List all physical properties of this PEA whose *rdfs:label* property contains the substring “flow”.	A list of physical properties with labels such as “volume flow” or “flow rate”.

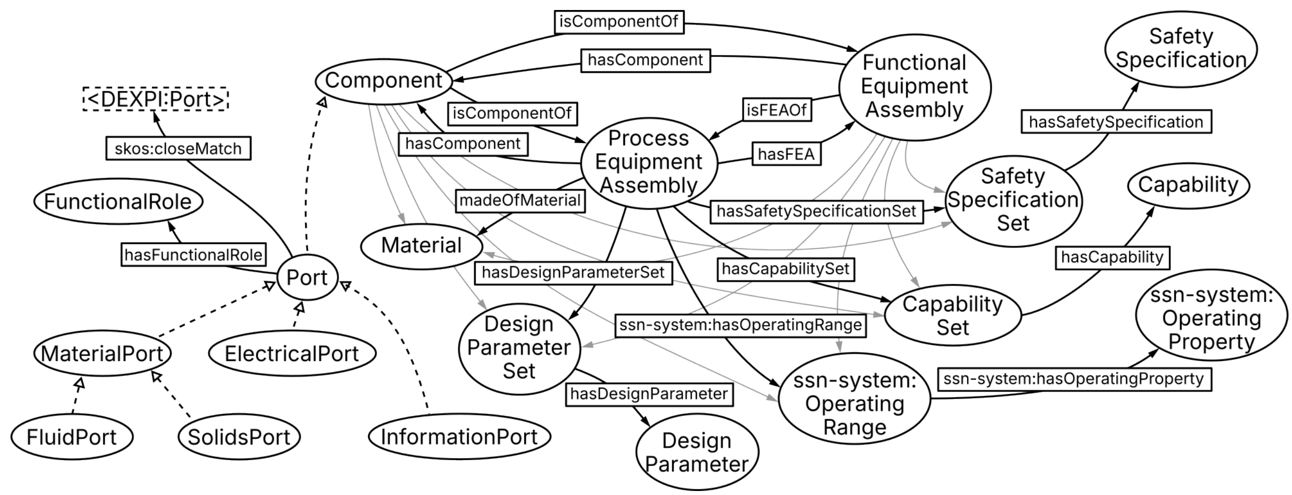

Figure 1. Core classes and properties from the SPEAD ontology. The thin grey lines depict relationships between the Component class and FEA class and other core classes that are only labeld between the PEA class and other core classes. Dashed lines indicate that the subject is a subclass of the object. The *spead:Port* class closely matches the definition of a port from the DEXPI 2.0 specification.

SPEAD Ontology

The core classes and properties are shown in Figure 1. Unless otherwise stated, all definitions shown are our own, and labelled with the prefix *spead:*. Components, FEAs, and PEAs are specialisations of OntoCAPE's plant item. Their aggregation relationships are derived from the VDI 2776-1 guideline [2]. We introduce intermediate grouping classes (e.g., *spead:SafetySpecificationSet* and *spead:DesignParameterSet*) to aggregate related property specifications from four of our five categories, drawing on the system capabilities module of the Semantic Sensor Network (SSN) ontology (prefix *ssn-system:*) [16] for the operating range. While these add verbosity, the grouping allows logically related specifications to be directly referenced. Further, sets can carry their own metadata and can be shared across different PEA instances, which simplifies the instance graph when a PEA pool contains multiple copies of the same type of PEA. Components and FEAs may also carry these sets, as indicated by the thin grey lines in Figure 1.

For interfaces, we define the abstract *spead:Port* class, which is specialised further into material, electrical, and information ports. This split was derived from the DEXPI 2.0 process model [13], and is mapped to the respective classes using *skos:closeMatch* as shown in Figure 1.

We define the abstract class *spead:QuantifiableProperty* for all equipment-related properties which are typically expressed as a name, a value, and a unit. In some cases, these kinds of properties occur as single values, e.g., the volume of a tank vessel. In other cases, values are a range or set, e.g., the maximum and minimum operating pressure of a tank vessel. These properties are also categorised differently, e.g., design parameters for attributes such as tank volume or procedure parameter for properties related to an MTP service [4]. In the instance graph, these properties can also be instances of the *spead:QuantifiableProperty* class, providing them with a shared structure. We choose this approach over subclassing to allow qualitative properties (categorisations, colours, etc.) to be described using the same classes as quantitative properties.

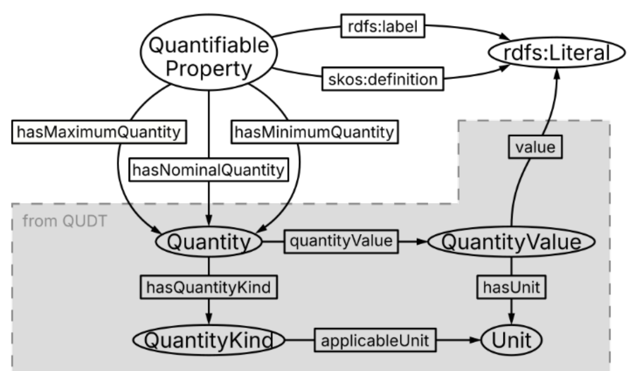

Figure 2. Schema for describing quantifiable properties. Classes and properties outside the gray box are own definitions, excluding SKOS and RDFS terms.

A detailed schema for the *spead:QuantifableProperty* class is shown in Figure 2. We utilise the structure of the Quantities, Units, Dimensions and Data Types Ontologies (QUDT), which provides a schema for the description of quantities (prefixed with *qudt:*) as well as vocabularies that define a large number of quantity kinds (prefix *quantitykind:*), units (*prefix unit:*), dimension vectors, and constants [17]. Using these kinds of vocabularies ensures that every stakeholder in the PEA lifecycle has the same, unambiguous understanding of the described quantities. This is relevant in cases where vendors and operators use different internal vocabulary. For example, one party might call the rate at which a fluid flows through a dosing module "flow rate", while the other calls it "volume flow". In our model, both parties link to the QUDT instance of *quantitykind:VolumeFlowRate*, maintaining their internal naming schema whilst preserving the interoperability of the data. We deviate from the structure of physical quantities in OntoCAPE, since QUDT provides an actively maintained, cross-domain, standardised structure for the representation of quantities, dimensions and units. Additionally, it contains an extensive vocabulary of kinds of quantities and units. These include 2896 unit descriptions, whereas OntoCAPE's 41 unit descriptions miss crucial units for process systems, such as those for volume flow. By leveraging QUDT, we eliminate the need of creating additional unit and quantity descriptions, thus removing a potential source of inconsistencies.

For provenance tracking, we transform the data model from the VDI 2770 guideline [5] into an RDF format. As a base we use the PROV Ontology due to its extensive vocabulary for tracking entities, their derivations, involved agents, and activities [18]. Figure 3 shows a small part of the transformed data model and how it relates to the components, FEAs, and PEAs.

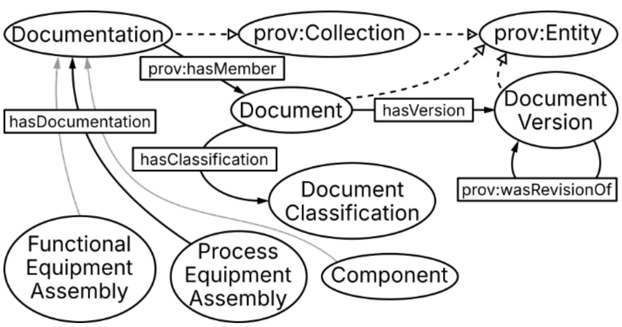

Figure 3. Classes and properties for tracking provenance. The thin grey lines depict the hasDocumentation relationship between FEAs and Components and their documentation.

Data Quality Assurance

Data quality is ensured at two levels. Our OWL ontology is intentionally kept lightweight, defining only class hierarchy and property semantics and avoiding complex class restrictions. Structural validation is instead delegated to SHACL shapes. In an industrial context, the Closed World Assumption SHACL operates under is important: unlike OWL's Open World Assumption, which treats missing information as merely unknown, SHACL treats absence as a validation error.

At the ontology level, we utilise ontologies for metadata and provenance descriptions such as Simple Knowledge Organization System (SKOS, prefix *skos:*) [19] and DCMI Metadata Terms (DC Terms, prefix *dcterms:*) [20]. All class and property definitions were annotated with labels, definitions and examples, as well as common alternative labels and sources where appropriate in both English and German. External resources (e.g., definitions from DEXPI) were related using SKOS mapping properties. Finally, we utilise the Basic Formal Ontology (BFO, prefix *bfo:*), as a foundational ontology. Foundational ontologies supply generic categorisation of the kind of things that an ontology describes, domain-independent relations, and means to represent attributes of the described things. The use of foundational ontologies provides a framework of fundamental distinctions, and facilitates the integration of different domain-specific ontologies based on the same foundational ontology. In our ontology, we use the OWL implementation of BFO, and declare our classes and properties as specialisations of the appropriate BFO terms. For example, a PEA is an object aggregate, i.e., a material object that consists of multiple parts that work together as a functional entity. We document the alignment rationale with the *skos:scopeNote* property. The serialisation of this example is as follows, where the object aggregate term is called *bfo:BFO_0000027*:

```
spead:ProcessEquipmentAssembly a owl:Class ;
   rdfs:subClassOf bfo:BFO_0000027 ;
   skos:scopeNote "Aligned to BFO 'object
aggregate', as PEAs are aggregations of FEAs
and components that together form a
unit."@en .
```

At the instance level, SHACL (prefix *sh:*) shapes are used to validate structural conformance. Amongst others, these include shapes that validate the aggregation of components and FEAs to PEAs, which is comprised of validating the domain and range of the properties shown in Figure 1, as well as ensuring that components and FEAs are only part of one PEA, excluding transitive relationships (e.g., a component that is part of a FEA which is part of a PEA is by the transitive characteristic of their relationships also part of that PEA). Further, we introduced shapes to ensure that all instances have an identifier, expressed as *dcterms:identifier*, and a source expressed as *dcterms:source*.

Since we retained the structure of QUDT, the SHACL shapes provided by QUDT can be reused. We expand on them, adding cardinality restrictions and further structural requirements. For example, QUDT allows for a quantity to have zero or more quantity kinds. Since the data in a documentation must be clear and leave as little room for faulty interpretation as possible, we enforce that every quantity has exactly one quantity kind, as shown in the following.

```
spead:QuantityShape a sh:NodeShape ;
```

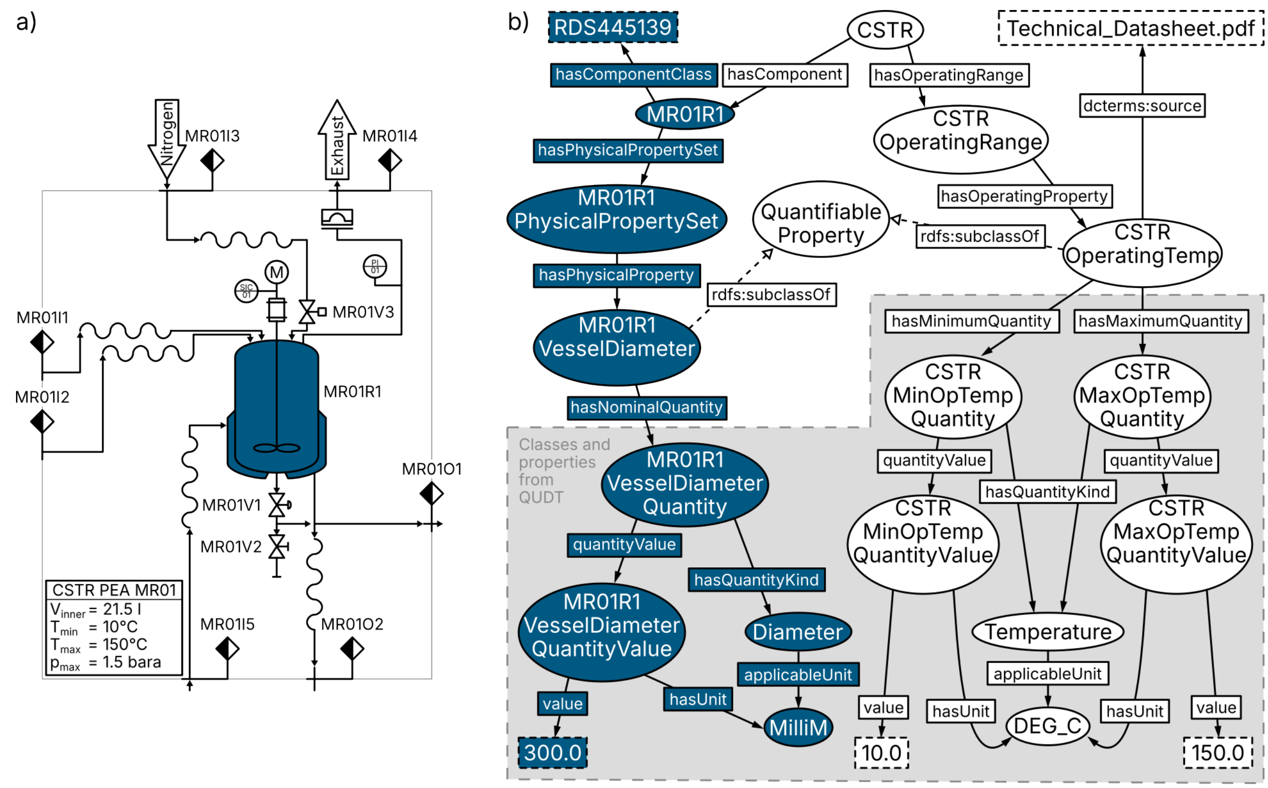

Figure 4. a) Simplified P&ID of a Continuous Stirred-Tank Reactor (CSTR) PEA developed within the project REUNION. The reactor component is highlighted in blue. b) Excerpt from the SPEAD representing the CSTR PEA. Highlighted in blue are nodes and properties relating to the reactor component in a). Boxes with dashed outlines indicate literal values, and the properties they are the subject of are datatype properties. Unless otherwise stated, all classes and properties are own definitions.

```
sh:targetClass qudt:Quantity ;
sh:property [
   sh:path qudt:hasQuantityKind ;
   sh:class qudt:QuantityKind ;
   sh:minCount 1 ;
   sh:maxCount 1 ;
] .
```

Further, the minimum and maximum quantity of a property must link to the same quantity kind using the *qudt:hasQuantityKind* property. The quantity values of those quantities must also be associated with at least one unit that is applicable to that specific quantity kind.

EVALUATION

Case study: CSTR PEA & Dosing PEA

Two PEAs are used as case studies for the evaluation of our ontology: a continuous stirred-tank reactor (CSTR) PEA as well as a dosing PEA. Figure 4a) shows a simplified P&ID of the CSTR PEA, which is developed as part of the REUNION project. The reactor component is highlighted in blue. The documentation of both PEAs was evaluated and information relevant to the competency questions was transformed into an instance graph of the SPEAD ontology, an excerpt of which is shown in Figure 4b) for the CSTR PEA. Nodes, relationships and values that directly relate to the reactor component are highlighted in blue. Figure 4b) shows how the diameter of the reactor vessel as well as the operating temperature of the CSTR PEA are modelled using our ontology. The vessel component is identified as a vessel-class component through a reference to the same Reference Data Library (RDL) used by DEXPI via the datatype property *spead:hasComponentClass*. Provenance of the operating temperature property is given with the *dcterms:source* property. In total, the SPEAD of the CSTR PEA contains 358 triples, while the SPEAD of the dosing PEA includes 214 triples, excluding any inferred triples.

We validated the ontology against the CQs shown in Table 1 and Table 2 through SPARQL queries. As an example, the following query was used to validate CQ1:

```
SELECT ?val ?unitLbl
WHERE {
   ?pea dcterms:identifier "MR01" .
   ?pea ssn-system:hasOperatingRange/
ssn-system:hasOperatingProperty ?prop .
```

```
   ?prop spead:hasMaximumQuantity/qudt:hasQuantityKind quantitykind:Pressure .
   ?prop spead:hasMaximumQuantity/qudt:quantityValue ?valNode .
   ?valNode qudt:value ?val .
   ?valNode qudt:unit/rdfs:label ?unitLbl .
}
```

For the content-related questions, the instance graph of both PEAs successfully covers operating parameters (CQs 1-3, see also Figure 4b)), as well as design parameters and PEA structure details (CQs 4-6). Certification and service details (CQs 7 and 8) are technically answerable, though the current implementation has limited expressiveness compared to the structure of quantifiable properties, which is discussed below. Integration of metadata properties from VDI 2770 enables the instance graph to answer provenance-related questions (CQs 9 and 10). CQ11 is covered through our strict policy to include labels, alternative labels, definitions, and sources for every newly introduced class and property. By validating the instance graph with our SHACL shapes we ensured that CQ12 was also successfully met.

In addition to our own CQs, we evaluated the ontology against 8 of the 11 competency questions derived by Koch et al. based on the selection of PEAs for a neutralisation process [11]. The remaining 3 CQs cover the entire modular plant, which exceeds the scope of our ontology. SPEAD performs well in answering their questions related to asset specification. For example, the question “How many centrifugal pumps does the PEA module have?” can be answered by counting the number of component nodes directly or indirectly linked to the PEA node through the *spead:hasComponent* property which are the subject of the property *spead:hasComponentClass* with the value set to the appropriate component class from the DEXPI RDL. However, the questions regarding the functional design cannot be answered, since SPEAD does not include mathematical models for the process design. This could be addressed in the future by further integrating the relevant submodels of OntoCAPE.

DISCUSSION

The SPEAD ontology sufficiently addresses CQs 1-6 by modelling the structure of PEAs and their quantifiable properties in a consistent, verifiable manner. However, in its current state, the implementation of the *spead:Capability* and *spead:SafetySpecification* classes limit the quality of the stored information. This stems from a deliberate design choice. In automation terms, the capabilities of a PEA are reflected through the implemented MTP services. While enriching the SPEAD ontology with service descriptions would improve human readability, the operational information already exists in the MTP files. The current implementation only adds metadata, avoiding the duplication of information.

We did not focus on the safety specifications, only implementing and validating them in a string-based format. In the future, especially in cases such as ATEX certifications or the IP rating of a PEA, small controlled vocabularies could be used to express safety related properties of a PEA instead.

Compared to OntoCAPE and DEXPI, SPEAD defines terminology definitions more precisely (*rdfs:comment* vs. *skos:definition* and *dcterms:source*), though all three models are capable of answering CQ11. However, neither OntoCAPE nor DEXPI can answer CQ9 and CQ10, which are crucial for traceability and trust, especially in documentation. These questions are addressed by SPEAD through its utilisation of PROV-O and its implementation of metadata properties from VDI 2770.

The evaluated CSTR and dosing PEAs in our study consisted of 358 and 214 triples respectively. Information was primarily drawn from P&IDs and technical data sheets, which were the only available documents due to the ongoing development of the PEAs. We expect a single SPEAD to be of the order of 10^4 triples, and a modular plant to be of the order of 10^5 triples.

At this scale, both reasoning and validation remain tractable. The OWL constructs used in SPEAD are intentionally conservative, avoiding complex role chains and nominals. This keeps reasoning complexity polynomial with respect to graph size, meaning that a tenfold increase in graph size does not lead to a disproportionate increase in reasoning time. SHACL validation performance is similarly predictable: most shapes are node- or property-scoped and scale linearly with the number of instances. One exception is the path-based constraint excluding transitive aggregation relationships, which could become costly for deeply nested structures. In practice, nesting depth is bounded by the four-level component-FEA-PEA-MP hierarchy defined in VDI 2776-1.

CONCLUSION & OUTLOOK

We introduced the SPEAD ontology for the documentation of Process Equipment Assemblies that collects and categorises information on PEAs and enriches this information with documentation metadata. We evaluated the ontology against competency questions that cover both content of the information model as well as terminology and provenance-related information, which we derived from an exemplary use case. We also evaluated the ontology against competency questions from the literature. The results show that the ontology works well for the considered use case of exchanging PEAs in an existing plant. Compared to existing data models, it improves the tracking of provenance related metadata.

The next steps in the development of SPEAD include the improvement of the description of module capabilities and safety-related properties. Further, additional use cases for SPEAD will be investigated. For example, we aim to investigate how SPEAD can be used in the matching of PEAs during the plant engineering phase. SPEAD could also be used as a source of static information for process simulation models, e.g., stirrer speed or wall thickness. Further, we aim to extend the ontology to also cover the non-modular environment. This is based on the VDI 2776-1 guideline, which states that the non-modular environment can be considered as a PEA in the context of automation [2]. We extend this assumption to the documentation. The resulting "Infrastructure SPEAD" can be used to perform matching queries with the SPEADs of a pool of PEAs to select PEAs based on their compatibility with the infrastructure.

Since the creation of instance graphs can be a time-consuming task, an automated pipeline to convert existing documentation such as technical data sheets into SPEADs should also be explored.

Lastly, due to the emergence of the Asset Administration Shell (AAS) as a key digital twin technology, integration of SPEAD either as a source for semantic identifiers or as its own submodel should also be considered.

ACKNOWLEDGEMENTS

This work is conducted within the project REUNION. The authors thank all project partners and the German Federal Ministry of Economic Affairs and Energy for the funding (funding codes #03EN2100A & #03EN2100J)

AUTHOR IDENTIFIERS

Author ORCIDs:
Lamm S: 0000-0001-5912-1948
Tecl S: 0009-0004-3194-3199
Dietrich I: 0009-0008-9691-972X
Sommer S: 0009-0001-5287-2682
Heinbücher M: 0009-0003-5097-791X
Pelz PF: 0000-0002-0195-627X

REFERENCES

1. DECHEMA e.V.. Modular Plants: Flexible chemical production by modularization and standardization – status quo and future trends. (2016) https://dechema.de/dechema_media/modularplants.pdf
2. Verein Deutscher Ingenieure e.V. (VDI). VDI 2776-1, Process engineering plants – Modular plants – Fundamentals and planning modular plants. DIN Media (2020)
3. Verein Deutscher Ingenieure e.V. (VDI). VDI 2776-2, Process engineering plants – Modular plants – Designing modular plants. DIN Media (2024)
4. Verein Deutscher Ingenieure e.V. (VDI). VDI/VDE/NAMUR 2658-1, Automation engineering of modular systems in the process industry – General concept and interfaces. DIN Media (2022)
5. Verein Deutscher Ingenieure e.V. (VDI). VDI 2770, Operation of process engineering plants – Minimum requirements for digital manufacturer information for the process industry – Fundamentals. DIN Media (2020)
6. Lamm S, Wetterich P, Dietrich I, Sommer S and Pelz PF. Semantic PEA Data sheet (SPEAD) – enhancing modular plant documentation with the Asset Administration Shell (AAS). Zenodo (2025) https://doi.org/10.5281/zenodo.15187664
7. European Parliament and Council. Regulation (EU) 2023/1230 of the European Parliament and of the Council of 14 June 2023 on machinery and repealing Directive 2006/42/EC of the European Parliament and of the Council and Council Directive 73/361/EEC (Text with EEA relevance). Official Journal of the European Union (2023) http://data.europa.eu/eli/reg/2023/1230/oj
8. World Wide Web Consortium (W3C). RDF 1.1 Concepts and abstract syntax. W3C (2014) https://www.w3.org/TR/rdf11-concepts/
9. World Wide Web Consortium (W3C). OWL 2 Web Ontology Language Document Overview (Second Edition). W3C (2012) https://www.w3.org/TR/owl2-overview/
10. World Wide Web Consortium (W3C). Shapes Constraint Language (SHACL). W3C (2017) https://www.w3.org/TR/shacl/
11. Koch A, Hamedi N, Furtner L, Kock T, Klose A, Mädler J. Standards for information models considering knowledge distribution in modular plants. 2023 IEEE 21st International Conference on Industrial Informatics (INDIN) :1-7 (2023). https://doi.org/10.1109/indin51400.2023.10218218
12. Marquardt W, Morbach J, Wiesner A, Yang A. OntoCAPE – A Re-Useable ontology for Chemical Process Engineering. Springer Berlin Heidelberg (2010) https://doi.org/10.1007/978-3-642-04655-1
13. DEXPI e.V.. DEXPI Specification 2.0.0 (2025) https://dexpi.gitlab.io/-/Specification/-/jobs/11676485644/artifacts/src/.build/html/html/index.html
14. International Electrotechnical Commission (IEC). IEC 61512-1, Batch control – Part 1: Models and terminology. IEC (1997)
15. Kendall EF and McGuinness DL. Ontology Engineering. Springer International Publishing (2019) https://doi.org/10.1007/978-3-031-79486-5
16. World Wide Web Consortium (W3C). Semantic Sensor Network Ontology. W3C (2017)

https://www.w3.org/TR/vocab-ssn/
17. QUDT.org. Quantities, Units, Dimensions and Types (2014) https://doi.org/10.25504/FAIRsharing.d3pqw7
18. World Wide Web Consortium (W3C). PROV-O: The PROV Ontology. W3C (2013) https://www.w3.org/TR/prov-o/
19. World Wide Web Consortium (W3C). SKOS Simple Knowledge Organization System Reference. W3C (2009) https://www.w3.org/TR/skos-reference/
20. Dublin Core Metadata Initiative (DCMI). DCMI Metadata Terms. DCMI (2020) https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

© 2026 by the authors. Licensed to PSEcommunity.org and PSE Press. This is an open access article under the creative commons CC-BY-SA licensing terms. Credit must be given to creator and adaptations must be shared under the same terms. See https://creativecommons.org/licenses/by-sa/4.0/