Project Report

Concepts of Data Integration & the Asset Administration Shell

Case study on behalf of and in cooperation with
Boehringer Ingelheim Biopharmaceuticals GmbH

Authors	Marcus Soemers, Manfred Theißen
Project	BI iMMA PoC Case Study
Project Period	18th of November until 15th of December, 2022
Report Date	19th of December, 2022

Contents

1. Introduction

The goal of the case study was the development of an application example that shows the relevance of integrated information in the life cycle of a process plant or, more generally, of an asset. *Integrated information* refers to the explicit representation of cross-relationships between different domains such as planning information and information about a real plant, but also between different aspects of planning, e.g., between a Piping and Instrumentation Diagram (P&ID) and equipment specifications.

1.1. The Digital Thread

Within the ENPRO Initiative[1], a methodology and an information model for data integration[2] across different phases of the life cycle of a plant have been developed. Several owner-operators have been involved in order to validate applicability and relevance of the integration model. The name of this information model for data integration has evolved over time; in this report, we refer to it as the *Digital Thread*. As there is currently no summary of the *Digital Thread* approach in English language, an overview is provided in the appendix of this report (see Sec. A).

1.2. The Asset Administration Shell

The *Asset Administration Shell*[3] (AAS) is a highly promising approach for the representation of domain information that aims at a complete *Digital Twin* of the entire life cycle of production facilities. Various submodels of the AAS are developed within the Platform Industry 4.0 by IDTA in different working groups[4]. Thus, the AAS is predestined as one basis for the case study.

1.3. Goals of the case study

The goal of the case study was to answer the following questions:

1. For a given use case, is it possible to model all relevant information by means of the *Digital Thread* or the AAS, respectively?

[1] http://enpro-initiative.de/
[2] https://modula.aixcape.org/information_model.html
[3] https://industrialdigitaltwin.org/en/
[4] https://industrialdigitaltwin.org/en/content-hub/submodels

2. If applicable: What is missing?

3. Are there potential synergies between the *Digital Thread* and AAS?

To this end, both a *Digital Thread* and an AAS model of a use case have been elaborated.

1.4. Use case

As a specific and deliberately simple use case, we consider the final planning information for the plant *X2*; according to the DEXPI life cycle model, this planning information is called an *asset specification*[5]. The asset specification comprises a P&ID of the plant to be built. In particular, a centrifugal pump with tag name *P4711* is planned with a required volumetric flow rate of 420 m^3/h. Further, the asset specification includes the information that for *P4711* a certain pump model *PressureMaker 400* by the manufacturer *Pump23, Inc.* should be used. According to the manufacturer, this pump type comes with maximum capacity of 500 m^3/h.

Here, the *relevant information* (cf. question 1 above) is the entire "information chain" that establishes a relationship between the required and maximum capacity. If this information is explicit, it is available for automatic plausibility checks, documentation, etc.

[5]The four life cycle phases defined by DEXPI are *functional requirements*, *functional design*, *asset specification*, and *asset in operation*. The latter corresponds to the physical, i.e., built plant, whereas the three other phases correspond to different planning phases. Note that in DEXPI terminology, the usage of the term *asset* is narrower than in an AAS context.

2. Modeling the Use Case

2.1. Modeling with Digital Thread approach

Figure 2.1 shows a *Digital Thread* model that covers the aforementioned information. The *TypeOfFunctionalObject* Ⓐ corresponds to the asset specification of the plant *X2*. It is represented by the *ConceptualModel* Ⓑ, i.e., a DEXPI P&ID. The *TypeOfFunctionalObject* Ⓒ encapsulates the details of the plant; here, only the fact that *X2* should contain a component *P4711* is modeled Ⓓ. This component is specified Ⓔ to be a *CentrifugalPump* Ⓕ and a *PressureMaker 400* Ⓖ. The *PressureMaker 400* type refers to some external domain object Ⓗ that carries the information of a maximum volume flow rate of 500 m^3/h.

2.2. Modeling with Asset Administration Shells

The upper part of Figure 2.2 shows an AAS Ⓐ of the asset specification of the plant *X2* Ⓑ. The AAS contains two submodels.

Submodel Ⓒ describes the P&ID of the plant; it is based on the IDTA submodel template 02012 for DEXPI that is currently developed[1]. The figure only shows a subset of the information covered by the submodel template. For example, the property *ProcessPlantName* (that corresponds to the metadata property *ProcessPlantName* in DEXPI[2]) has the value *X2* Ⓓ. The submodel also provides the tagged objects within the underlying DEXPI P&ID, for example pieces of equipment such as the pump *P4711* Ⓔ. The DEXPI property *TagName* of the pump[3] is reproduced as a property *DEXPI name* with value *P4711* Ⓕ.

However, the reproduction of more detailed content of DEXPI P&IDs in terms of the AAS metamodel is beyond the scope of the submodel template 02012. The submodel template rather proposes to include the underlying DEXPI XML file as a further submodel element Ⓖ. Where applicable, relationships between the *Tags* in the submodel

[1] https://github.com/admin-shell-io/submodel-templates/tree/main/development/DEXPI/1/0, version from July 8, 2022. Minor modifications have been applied to Figure 2.2 after personal communication of the authors of this report with one of the developers of the submodel template. These modifications are expected to be included in a future version of the submodel template.

[2] https://dexpi.plants-and-bytes.de/reference/MetaData/MetaData.html#dexpi-metadata-metadata-processplantname

[3] https://dexpi.plants-and-bytes.de/reference/Equipment/TaggedPlantItem.html#dexpi-equipment-taggedplantitem-tagname

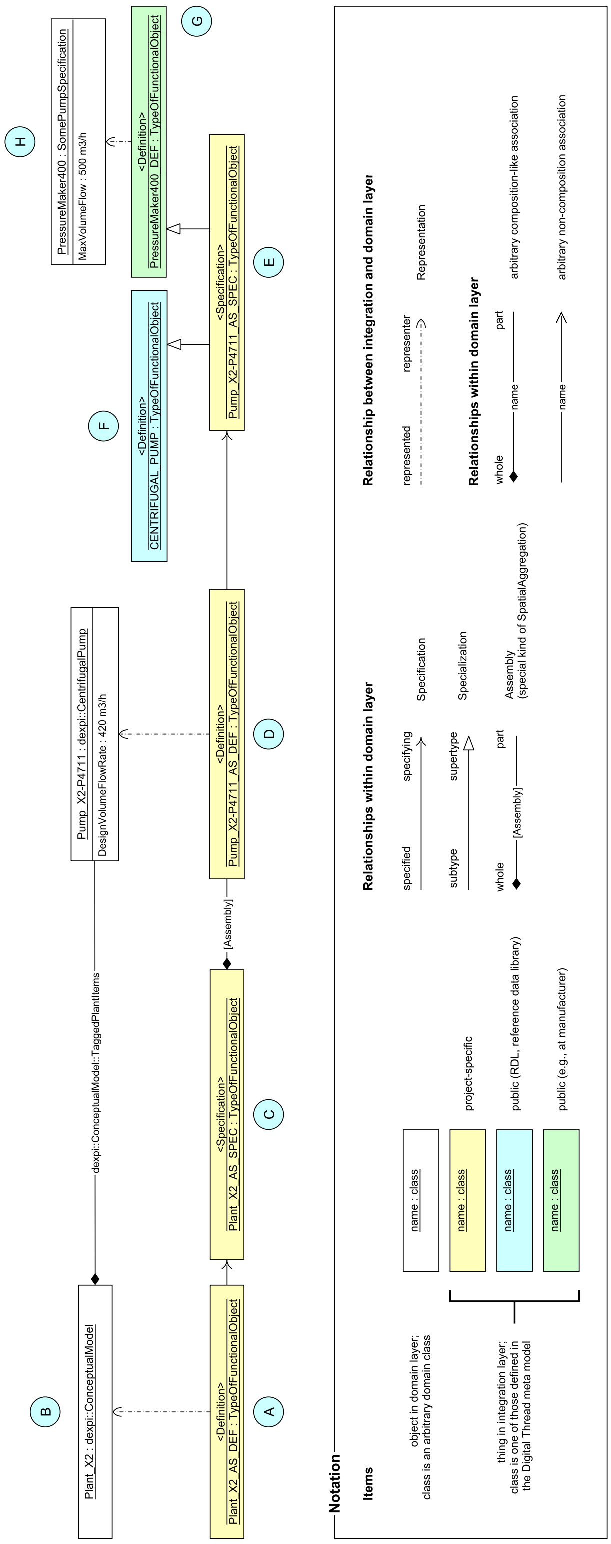

Figure 2.1.: *Digital Thread* model of the asset specification of a chemical plant and of a pump type offered by a manufacturer.

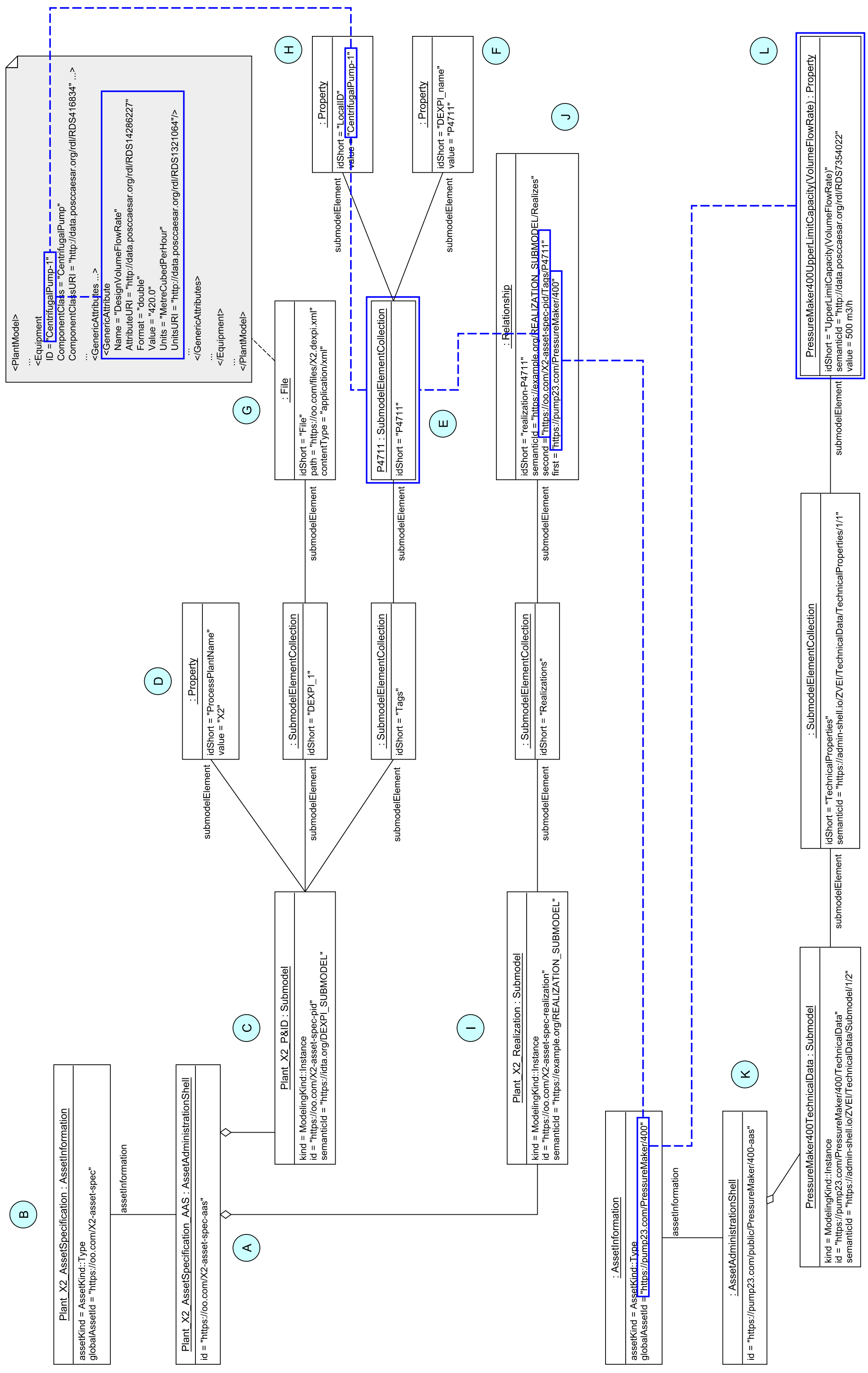

Figure 2.2.: *Asset Administration Shells* of the asset specification of a chemical plant and of a pump type offered by a manufacturer.

and the corresponding sections of the XML file can be established using the IDs of the XML file (e.g., *CentrifugalPump-1*).[4]

The second submodel Ⓘ of the AAS Ⓐ provides information about the realization of the plant *X2*. It is derived from a hypothetical submodel template *Realization* that enables to give *Realizes* relationships that link, e.g., the *Tags* of a DEXPI P&ID to the specific equipment types that shall be used in the plant. In the figure, the *Realizes* relationship Ⓙ states that a *PressureMaker 400* pump by *Pump23, Inc.* shall realize the pump *P4711* from the P&ID.

For the pump type *PressureMaker 400*, a separate AAS Ⓚ is shown in the lower part of the figure. It contains a submodel according to the IDTA submodel template 02003-1-2 *Generic Frame for Technical Data for Industrial Equipment in Manufacturing*[5]. A single exemplary property Ⓛ states that the maximum flow rate of the pump type is 500 m^3/h. The property is identified (*semanticId*) via the ISO 15926 RDL entity *UpperLimitCapacity(VolumeFlowRate)*[6].

[4]The depiction in the figure is simplified in the interest of readability. Rather than a simple property, the submodel proposes a relationship.

[5]https://github.com/admin-shell-io/submodel-templates/tree/main/published/Technical_Data/1/1

[6]http://data.posccaesar.org/rdl/RDS7354022

3. Analysis

The following analysis is based on the questions posed in Sec. 1.3.

1. *For a given use case, is it possible to model all relevant information by means of the* Digital Thread *or the AAS model, respectively?*

Thanks to their sufficiently generic and flexible design, both models enable the representation of the relevant information. In particular, the relationship between the required pump capacity in the plant and the maximum capacity of the chosen pump type is covered.

2. *If applicable: What is missing?*

3. *Are there potential synergies between AAS and the* Digital Thread*?*

The *Digital Thread* is a conceptual model, whose focus is on a sufficiently flexible and expressive foundation of the integration layer, independently from the internal structure of the information models used in the domain layer. However, it is not an implementation model, and it does not prescribe a technical realization, in particular with respect to the requirements of a distributed environment.

Vice versa, the AAS does not yet provide a semantically rich submodel to describe cross-domain relationships in a standardized way, whereas its technical realization has already reached an advanced stage.

Thus, by defining an AAS submodel template for the integration layer of the *Digital Thread*, the strengths of the two approaches can be combined.

4. Conclusion

The application example was modeled based on the concepts of data integration that were developed in two publicly funded projects (see Sec. A) and which are based on ISO 15926. This was compared with the modeling according to the current status of the specification for the AAS. In principle, it can be stated that the modeling of the information required in the example, including the cross-relationships, is possible or will be possible in the future by means of the AAS. As part of the case study, we investigated how relationships between different submodels of an AAS and between different AAS can be modeled. However, there is still no predefined framework with clearly defined semantics of basic relationships. Such a framework could be given with the help of the concepts from data integration. Therefore, a new submodel for the AAS is proposed in this case study.

In addition, further submodels may be useful in the future. For example, the decision representation model that was also proposed in the data integration projects could be introduced as a submodel of the AAS. In this way, decisions could be tracked and documented. To give an example, we refer again to the plant considered in this case study. In order to be able to create a predictive maintenance plan for the pumps of the plant, the plant operator would like to access the history of all installed pumps, including the information as to why these pumps were replaced in the past (wear, defects, energy optimization, etc.). Such information could be documented (modeled) using the aforementioned decision representation model.

In addition to the results documented in this report, the case study workers gained valuable insights and were able to hold exciting discussions with people from the working groups on the AAS.

Acknowledgements

This case study would not have been possible without many fruitful discussions with Sten Grüner (ABB AG), Andreas Orzelski (PHOENIX CONTACT Deutschland GmbH), and Thomas Tauchnitz (TAUTOMATION.consulting).

A. Information model for life cycle modeling – basis of the Digital Thread

The *information model for life cycle modeling*[1] provides modeling elements (data types, classes, etc.) with which life cycle information is modeled according to the results of the publicly funded project *ENPRO 2.0 ModuLA*. It is a further development of the *Object Integration Model for the Engineering Lifecycle* (OIMEL) from *ENPRO 1.0 Data Integration*, taking into account the requirements of a module-based approach. Compared to OIMEL, individual aspects that were initially only dealt with conceptually (such as the domain models) have been formalized. The information model is a class model according to the *Unified Modeling Language* (UML 2.5.1).

Here we only reproduce an excerpt from the project report of the ModuLA project in order to explain essential aspects of the *information model for life cycle modeling*. For more details, we refer to the project report, which is **online**[2] available (in German language).

A.1. Concepts

The concept of data integration aims at joining data and information that is contained in different formats and/or information models (*domain models*) in a single overall model. For this, in a first step an overall information model is required, the expressiveness of which is equal to the sum of the considered domain models (or relevant parts of these domain models). The existing domain information, i.e., information that is given by means of the domain models, has to be modeled in a second step by means of the overall information model.

Figure A.1 shows the concept from *ENPRO 1.0 Data Integration* (von Wedel et al., 2018) in a schematic way. An essential approach is that the existing domain models und hence the corresponding domain information are reused without any modification (*domain layer*). This way, additional domain models can easily be included with moderate effort.

Integration of the information in the domain layer is realized in a separate *integration layer*, out of which single elements of the domain layer are referenced and in which cross-relationships as essential integration information are modeled additionally.

[1] https://modula.aixcape.org

[2] https://modula.aixcape.org/project_report.html

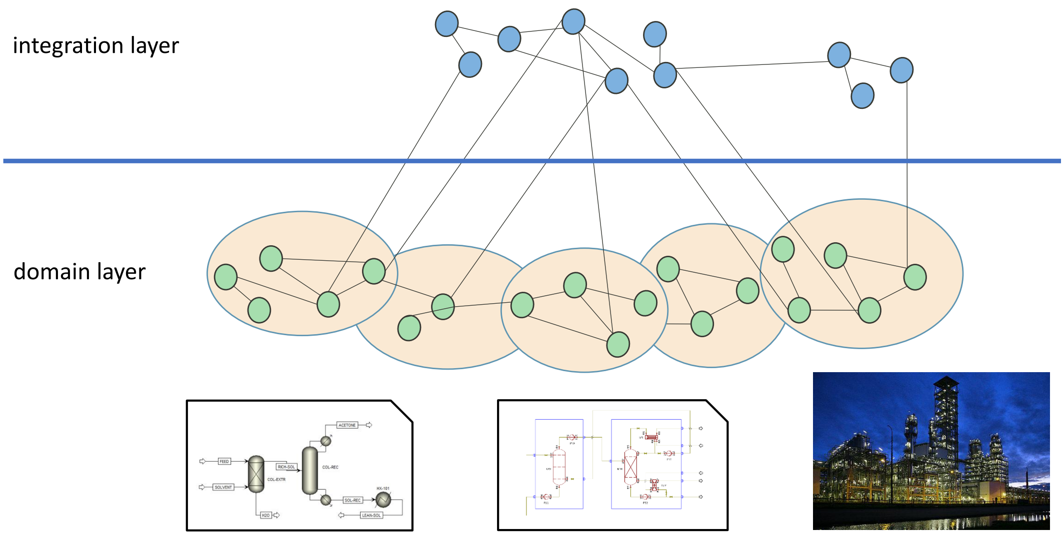

Figure A.1.: Separation of integration and domain layer.[3]

A.2. Domain layer

Within the domain layer, domain specific information is represented, e.g., simulation models, laboratory data, basic flow diagrams, process flow diagrams, process and instrumentation diagrams (P&IDs), apparatus specifications, automation and maintenance information or operational data. For this purpose, in principle arbitrary information models can serve, however, in the interest of a simpler data exchange preferably no proprietary tool models should be used, but accepted standards (such as DEXPI with respect to P&IDs). As meta model for domain models, a subset of UML (*Unified Modeling Language*, UML 2.5.1) is used.[4] Information models for domains often already exist in UML (DEXPI-P&ID), or they can be mapped to UML in a relatively easy and systematic manner (data base schema, XML schema, ...).

It is essential that individual elements in the domain layer, which represent, e.g., a technical location *P4711* for a pump or a pump with the serial number *XY123*, can be referenced.[5] In this way, a relationship across domain boundaries can be established within the integration layer, as shown in Figure A.1. It should be emphasized that no physical reproduction of the complete domain information is required; integration can be realized by adequate referencing of existing information elements, which especially is appropriate with respect to huge amounts of data, e.g., operational data in a PIMS.

[3] License to the picture of the production plant bottom right: Attribution: Mikulova, CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0. Source: https://commons.wikimedia.org/wiki/File:Slovnaft_-_new_polypropylene_plant_PP3.JPG

[4] https://modula.aixcape.org/oimel/domain/index.html

[5] Technically, these “elements” can be UML objects, records in a database, XML elements, data records in a PIMS (Process Information Management System), etc.

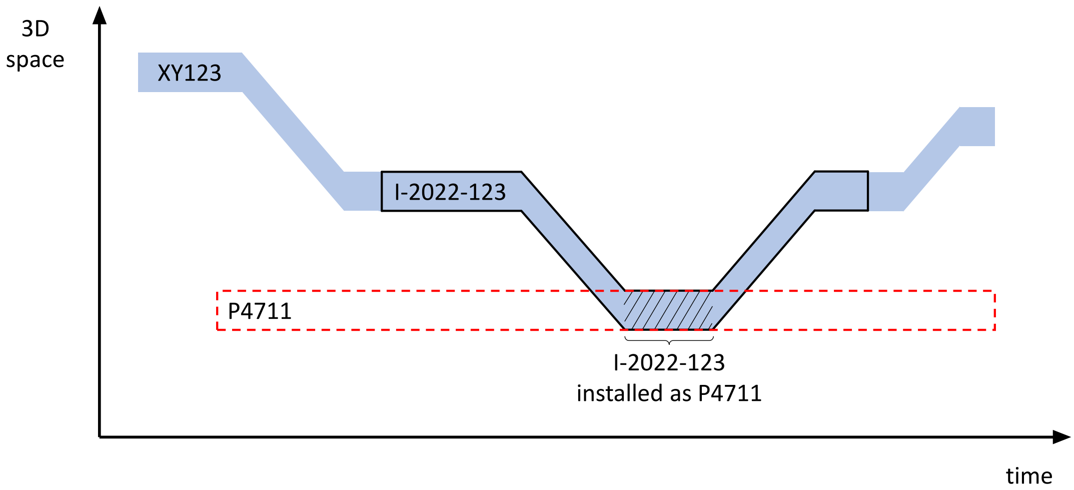

Figure A.2.: Schematic representation of *things* in space-time.

A.3. Integration layer

To answer the question about the identity of things[6] is not trivial. However, the realization of the integration concept in Figure A.1 requires a stringent modeling of the identities of the things in the integration layer. For example, a technical location *P4711* for a pump has a different identity than a pump with serial number *XY123* being installed in the technical location at a given point in time. Two identities must therefore be provided in the integration layer, one of which is to be linked to the representation of the pump in a P&ID, the other one to manufacturer information on the pump, such as year of construction and pump type. The pump with the serial number *XY123* may also have an inventory number, e.g. *I-2022-123*, as long as it is owned by a company. It would have to be decided whether this inventoried pump has the same identity as the pump with serial number *XY123*.

In ISO 15926 the paradigm of *four-dimensionalism* is applied. Concrete things are viewed as embedded in four-dimensional space-time. This paradigm provides a simple criterion for identity: two things are identical if and only if they correspond to the same piece of space-time, or 4D-piece for short (*extensionalism*, see West (2011) for an introduction).

Figure A.2 serves to illustrate the 4D-paradigm. The vertical axis represents 3D-space, the horizontal axis represents time as the fourth dimension. The pump with the serial number *XY123* is represented by the area with a blue background. It exists in a specific time domain, and at every point in its existence it occupies a piece of 3D-space. The pump with the inventory number *I-2022-123* is represented in the figure by the area framed in black. This area represents a true subset of the area highlighted in blue for the pump with the serial number *XY123*.[7] According to the criterion of extensionalism,

[6] The term *thing* is used with respect to the class model that is introduced in ISO 15926.

[7] More precisely: the pump with the inventory number is a *temporal part*[8] of the pump with the serial

the "two" pumps are different things in space-time.

The area framed in red represents the function labeled *P4711* (in the sense of a technical location) for a pump in a production plant. *P4711* and *I-2022-123* overlap in the black shaded area; this area corresponds to the state that the pump *I-2022-123* (or the pump *XY123*) is installed in the function *P4711* (as an *installed thing*[9]). Both, *P4711* and the installed thing are in turn to be regarded as things with their own identity.

The 4D-paradigm allows to define the meaning of pump *P4711* exactly.

The proposed concept for data integration is characterized by the fact that the level of detail and the depth of the integration can be handled flexibly. In a minimal approach, only the identity of things would be mapped in the integration layer. For any further information, reference would be made to the domain layer.

This minimal approach can be gradually expanded by mapping such domain information that is relevant across domain boundaries in the integration layer. This is fundamentally associated with additional effort and increased complexity, since a mapping from the information models of the domains to the integration layer has to be defined and implemented.[10] Therefore, such additions in the integration layer should only be considered if necessary, i.e., depending on the intended applications.

This concept differs from the approach which, to the best of the authors' knowledge, is currently being pursued primarily in the ISO 15926 community.[11] There, an attempt is made to map all life cycle information using ISO 15926. From the point of view of the authors of this report, such a "complete" integration layer certainly has advantages; in particular, it offers access to all information, not only in a formally but also in a semantically uniform way. However, one difficulty may be that the complexity mentioned above will be reflected in the integration layer. The concept for data integration proposed in the Digital Thread is intended to support a step-by-step implementation in commercial software tools with regard to complexity.

number.

[8] https://modula.aixcape.org/oimel/integration/TemporalAggregation/index.html

[9] https://modula.aixcape.org/oimel/integration/InstalledThing/index.html

[10] The reasons for this effort are the differences between the information models of the domains, which usually have arisen independently from each other. In simple cases, only terminological differences have to accounted for, but typically there are significant structural differences.

[11] https://15926.org/home

Bibliography

ISO 15926. *Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities*. Internationaler Standard.

UML 2.5.1 (2017). *OMG Unified Modeling Language (OMG UML)*. Version 2.5.1. URL: https://www.omg.org/spec/UML/2.5.1.

von Wedel, L., Welke, R., Richert, H., Theißen, M., and Mitsos, A., eds. (2018). *Verbundprojekt „ENPRO“-Datenintegration. Schlussbericht*. DOI: 10.2314/GBV:1037632206.

West, M. (2011). *Developing High Quality Data Models*. Burlington, MA, USA: Morgan Kaufmann. DOI: 10.1016/C2009-0-30508-5.