

Verbundvorhaben ModuLA

Als Teil der Initiative
 des Projektverbundes ENPRO 2.0
 im Rahmen der Förderinitiative „Energieeffizienz in der Industrie“
 des Bundesministeriums für Wirtschaft und Klimaschutz

Schlussbericht

Zuwendungsempfänger: AixCAPE e.V. Peterstraße 2-4, 52062 Aachen	Förderkennzeichen: 03ET1594A
Vorhabensbezeichnung: Module im Lebenszyklus einer prozesstechnischen Anlage (ModuLA) Anwendungen für integrierte Modelle	
Laufzeit des Vorhabens: 1.9.2018 - 28.2.2022	
Berichtszeitraum: 1.9.2018 - 28.2.2022	Eingereicht am: 30.8.2022

Gefördert durch:



aufgrund eines Beschlusses
 des Deutschen Bundestages

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.



Dieser Schlussbericht wird von AixCAPE e.V. unter der Lizenz *Creative Commons - Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International (CC BY-SA 4.0)* bereitgestellt.

<https://creativecommons.org/licenses/by-sa/4.0>

Inhaltsverzeichnis

1. Projektübersicht	5
1.1. Kurze Darstellung der Aufgabenstellung	5
1.2. Kurze Darstellung zu den Voraussetzungen des Vorhabens	6
1.3. Kurze Darstellung zur Planung und Ablauf des Vorhabens	7
1.4. Kurze Darstellung zum wissenschaftlich-technischen Stand	7
1.5. Überblick über die wichtigsten Ergebnisse	9
1.6. Zusammenarbeit mit anderen Stellen	9
2. Eingehende Darstellung	11
2.1. Eingehende Darstellung der Ergebnisse	11
2.1.1. Konzept zur Datenintegration	11
2.1.2. Informationsmodell zur Lebenszyklus-Modellierung	14
2.1.3. Tutorial zur Lebenszyklus-Modellierung	27
2.1.4. Anwendungsbeispiel: Verfahrensentwicklung und Prozessoptimierung	37
2.1.5. Anwendungsbeispiel: Modulare Anlagenplanung und Modul-Automatisierung	46
2.1.6. Anwendungsbeispiel: Verknüpfung von Produktion und Intralogistik in einem Brownfield-Projekt	56
2.2. Eingehende Darstellung des voraussichtlichen Nutzens	65
2.3. Eingehende Darstellung der Veröffentlichung der Ergebnisse	66
A. Modelle	68
Literatur	74

1. Projektübersicht

1.1. Kurze Darstellung der Aufgabenstellung

Um die chemische Produktion flexibler zu gestalten, den Energiebedarf zu minimieren und die Markteinführungszeiten eines Produktes zu reduzieren, wächst der Wunsch nach modularen Anlagen, d.h. Anlagen, die aus standardisierten und konfigurierbaren modularen Prozesseinheiten¹ aufgebaut sind (vgl. ProcessNet, 2017). Eine wesentliche Herausforderung bei der modulbasierten Anlagenplanung oder bei der Durchführung von Brownfield-Projekten unter Einbeziehung von Modulen ist die Bereitstellung von Informationsmodellen für den gesamten Lebenszyklus² eines Moduls sowie deren informationstechnische Integration. Sind Module informationstechnisch erfasst und leicht in einen Anlagenlebenszyklus zu integrieren, lassen sich beispielsweise viele Prozessvarianten in kurzer Zeit miteinander vergleichen.

Den Schwerpunkt des Projektes *ModuLA* innerhalb des Projektverbundes *ENPRO 2.0* bildete deshalb die Spezifikation eines durchgängigen und konsistenten Informationsmodells, mit dem relevante Informationen im gesamten Lebenszyklus eines Moduls oder allgemeiner einer prozesstechnischen Komponente³ erfasst, integriert und bereitgestellt werden können.

Dabei waren existierende Standards zur Datenintegration sowie zur Beschreibung verschiedener (Modul-)Aspekte zu berücksichtigen.

Die Anforderungen an ein solches Modell, das im folgenden Bericht auch kurz als *Integrationsmodell* bezeichnet wird, sollten aus Anwendungsszenarien auf Basis konkreter Prozesse und Anlagen abgeleitet werden. Die Beziehungen zwischen den verschiedenen Aspekten (oder auch Phasen) des Lebenszyklus waren möglichst vollständig, Attribute innerhalb der Aspekte exemplarisch abzubilden. Bei der Ausarbeitung der Szenarien war außerdem darauf zu achten, dass aktuelle und industriell relevante Herausforderungen adressiert werden. Dies wurde durch die assoziierten Partner des Projektes, die Chemieunternehmen *BASF SE*, *Clariant Produkte (Deutschland) GmbH* und *Evonik Operations GmbH*, sichergestellt. Schließlich sollten das erarbeitete Integrationsmodell mithilfe von zu implementierenden prototypischen Software-Anwendungen validiert und einhergehende Potentiale demonstriert werden.

¹Modulare Prozesseinheiten oder *Process Equipment Assemblies* (PEAs) werden im allgemeinen Sprachgebrauch häufig als Module bezeichnet.

²Eine oftmals verwendete Bezeichnung für den Lebenszyklus in diesem Zusammenhang ist *Asset Life Cycle* (ALC).

³Eine prozesstechnische Komponente kann eine Anlage (konventionell, modular oder hybrid), ein Modul, ein Einzel-Equipment, etc. sein.

Die Ergebnisse des Projektes stellen Enabling-Technologien auf dem Weg zur *digitalen Anlage* und *Industrie 4.0* dar, mit denen eine modulbasierte Vorgehensweise, von der Planung bis zum Betrieb und bei der Optimierung einer prozesstechnischen Anlage, etabliert werden kann. Sie stellen insbesondere eine Grundlage dafür dar, dass die mit einer durchgängigen Informationsverfügbarkeit verbundenen Potentiale, wie z.B. zur Ressourcen- und Energieeinsparung, künftig realisiert werden können.

1.2. Kurze Darstellung zu den Voraussetzungen des Vorhabens

Eine prozesstechnische Anlage wird heute immer noch möglichst optimal für einen zu realisierenden Prozess ausgelegt. Eine Verwendung von konfigurierbaren und wiederverwendbaren Modulen soll etabliert werden. An der Planung, dem Bau und dem Betrieb einer Anlage sind verschiedene Gewerke in unterschiedlichen Phasen des Lebenszyklus beteiligt, z.B. in der Verfahrensauslegung, der Rohrleitungstechnik oder der Automatisierung. Die einzelnen Gewerke erzeugen Informationen und müssen in der Regel auf Informationen anderer Gewerke, innerhalb derselben Phase oder darüber hinaus, zugreifen. Spielen künftig auch Hersteller von Modulen eine immer größere Rolle, wird der Austausch von Informationen zwischen verschiedenen Gewerken noch intensiver. Vielleicht findet das Engineering eines Moduls durch einen Anlagenbetreiber statt, der Bau wird bei einem Hersteller beauftragt. Oder verschiedene Prozessalternativen mit verschiedenen Modultypen unterschiedlicher Hersteller sollen leicht und schnell im Engineering eines Anlagenbetreibers miteinander verglichen und bewertet werden, beispielsweise bzgl. der Energie- und Ressourceneffizienz. Voraussetzung dafür ist eine durchgängige Verfügbarkeit von Informationen aller Gewerke. Von einer durchgängigen Verfügbarkeit der Informationen im gesamten Lebenszyklus einer prozesstechnischen Anlage oder eines Moduls, und damit auch von der Realisierung eines digitalen Zwillings, ist die chemische Industrie heute jedoch noch weit entfernt.

Allerdings existieren zahlreiche Standards, die für bestimmte Anwendungsbereiche innerhalb des Lebenszyklus bestimmt sind. Durch domänenspezifische Standardisierung und Informationsmodelle, die auf diesen Standards basieren, können die Informationen eines Anwendungsbereiches verfügbar gemacht werden. Dies ist eine grundlegende Voraussetzung, um z.B. die Informationen zwischen Software-Werkzeugen verschiedener Anwendungsbereiche übertragen zu können (Schüller u. a., 2017). Außerdem leisten verschiedene Projekte innerhalb der Initiative *ENPRO*, aber auch Gremien und Arbeitskreise außerhalb der Initiative bereits Beiträge auf dem Weg zur modularen Produktion, beispielsweise in den Bereichen Prozesstechnik, Anlagenplanung und Automatisierung. Mit einem durchgängigen Integrationsmodell können Informationen aus diesen Domänen künftig in integrierter Form zur Verfügung gestellt und genutzt werden, z.B. in Software für die Planung von modularen Anlagen oder in Software für die Planung einer modularen Logistik.

1.3. Kurze Darstellung zur Planung und Ablauf des Vorhabens

Das Projekt wurde im Oktober 2018 rückwirkend zum 1. September 2018 mit einer Laufzeit von drei Jahren bewilligt.

Im Projekt waren die drei Arbeitspakete *Szenarien und Informationsbedarf*, *Modellierung* sowie *Anwendungen und Validierung des Gesamtmodells* vorgesehen. Die Arbeitspakete wurden in drei Iterationen mit Bezug zu jeweils einer konkreten Anwendung durchgeführt. Die Anwendungsszenarien wurden mit den assoziierten Partnern und in Abstimmung mit anderen Projekten der Initiative *ENPRO* detailliert. Dabei haben die assoziierten Partner unterschiedliche Schwerpunkte gesetzt und aktuelle sowie industriell relevante Herausforderungen auf Basis konkreter Prozesse und Anlagen in die Formulierung der Szenarien einfließen lassen. Während *BASF* den Schwerpunkt auf die Verknüpfung von Informationsmodellen für Produktions- und Logistikmodule gelegt hat, war für *Clariant* die durchgängige Verfügbarkeit von Informationen zur Verfahrensentwicklung und Prozessoptimierung, vom Labor über das Technikum bis hin zum realen Anlagenbetrieb, im Fokus. *Evonik* hat die Bereitstellung von Informationen für ein modulbasiertes Anlagen-Engineering sowie die Integration von Informationen aus dem Modul-Engineering und der Modul-Automatisierung thematisiert.

Der aktuelle Stand der Arbeiten wurde kontinuierlich mit den assoziierten Partnern diskutiert, gemeinsam mit allen Partnern und je nach Themenschwerpunkt auch in kleinerer Runde. Dies konnte zu Beginn des Projektes in Form von persönlichen Treffen stattfinden. Im späteren Verlauf des Projektes wurden die Treffen aufgrund der Coronavirus-Pandemie als virtuelle Treffen organisiert. Insgesamt wurden vier reguläre Gesamt-Projekttreffen und vier themenspezifische Workshops in persönlicher Form, fünf virtuelle Gesamt-Projekttreffen und zahlreiche weitere virtuelle Absprachen organisiert.

1.4. Kurze Darstellung zum wissenschaftlich-technischen Stand

In diesem Abschnitt gehen wir auf Arbeiten ein, die für die Durchführung des Projektes relevant waren bzw. an die angeknüpft wurde.

Wie in Abschnitt 1.2 bereits erwähnt, existieren zahlreiche Standards für die Beschreibung einzelner Lebenszyklus-Aspekte einer konventionellen Anlage. Diese lassen sich teilweise, ggf. mit entsprechenden Anpassungen bzw. Erweiterungen, auch zur Beschreibung von Lebenszyklus-Aspekten eines Moduls verwenden. Beispielsweise ermöglicht der *DEXPI*-Standard grundsätzlich den Austausch von *Rohrleitungs- und Instrumentierungsdiagrammen* (R&Is) sowohl von konventionellen Anlagen, von modularen Anlagen und von R&Is einzelner Module.

Andererseits sind Standards verfügbar, die konkret auf ein modulares Setting abzielen, wie beispielsweise das *MTP* für die automatisierungstechnischen Aspekte von Modulen. Oder solche Standards befinden sich in der Entwicklung, wie dies für die modulare Pro-

duktionslogistik im Projekt *ENPRO 2.0 MoProLog* der Fall ist.

Die folgende Liste enthält einige relevante Vorarbeiten zu derartigen *Domänenmodellen*, an die im Projekt *ModuLA* angeknüpft wurde, ohne Anspruch auf Vollständigkeit zu erheben.

- In *ENPRO 1.0 Modularisierung* (Schembecker, 2018) wurden Ansätze zur Unterstützung einer modularen Anlagenplanung realisiert. Für Wärmetauscher- und Pumpenmodule wurden u.a. relevante Equipment-Informationen in einer Datenbank implementiert.
- *DEXPI* (Data Exchange in the Process Industries) ist eine internationale Initiative von Anlagenbetreibern, Software-Herstellern und Forschungseinrichtungen zur Entwicklung und Verbreitung eines allgemeinen Standards in der Prozessindustrie (DEXPI Initiative, 2022). Der Fokus liegt zur Zeit auf dem Austausch von R&Is (DEXPI 1.3).
- *MTP* (Module Type Package): In Zusammenarbeit von Namur und ZVEI wird ein Domänenstandard für die *modulare Automatisierung* entwickelt. Dieser Standard bildet die Grundlage dafür, dass Module weitestgehend automatisiert in die Prozessführungsebene integriert werden können (Bernshausen u. a., 2016).
- **NE 150** ist eine NAMUR-Empfehlung, die eine standardisierte Schnittstelle zum Austausch von Engineering-Informationen zwischen CAE-Systemen⁴ und Werkzeugen für das PCS-Engineering⁵ beschreibt.
- **NE 159** ist eine NAMUR-Empfehlung, die eine standardisierte Schnittstelle für den Datenaustausch zwischen CAE-Systemen der Verfahrensauslegung und CAE-Systemen für die PLT-Hardware-Planung⁶ beschreibt.

Die **ISO 15926** ist ein internationaler Standard zur Integration von Informationen im Lebenszyklus prozesstechnischer Anlagen. Für das Projekt *ModuLA* ist insbesondere Teil 2 (**ISO 15926-2**), in dem ein Informationsmodell zur Datenintegration definiert wird, von Bedeutung. West (2011), einer der Autoren von **ISO 15926-2**, schlägt eine als *HQDM Framework* (*High-Quality Data Model Framework*) bezeichnete Variante dieses Datenmodells vor, mit dem Ziel, das Modell leichter verständlich und formal einheitlicher zu gestalten. Die grundlegenden Modellierungsprinzipien aus **ISO 15926-2** werden beibehalten.

Diese Modellierungsprinzipien wurden bereits in dem Projekt *ENPRO 1.0 Datenintegration* (von Wedel u. a., 2018) zur durchgängigen Modellierung von Informationen im Lebenszyklus einer chemischen Anlage angewandt. In *ModuLA* wurden die dort erzielten Ergebnisse formalisiert und insbesondere im Hinblick auf modulare Anlagen erweitert.

⁴CAE: Computer Aided Engineering

⁵PCS: Process Control System

⁶PLT: Prozessleittechnik

1.5. Überblick über die wichtigsten Ergebnisse

- Im Projekt wurden verschiedene **Anwendungsszenarien** ausgearbeitet. Die assoziierten Partner haben dabei sichergestellt, dass in den Szenarien aktuelle industrielle Herausforderungen von hoher Relevanz auf Basis konkreter Prozesse und Anlagen betrachtet wurden.
- Zu den Anwendungsszenarien wurden verschiedene **Instanzenmodelle** erstellt, um zu überprüfen, ob und wie sich die erforderlichen Informationen mit den im Projekt erarbeiteten bzw. erweiterten Modellierungskonzepten abbilden lassen.
- Auf Basis der identifizierten Anforderungen, die anhand der Szenarien und der zugehörigen Instanzmodelle abgeleitet wurden, wurde ein **Integrationsmodell** zur durchgängigen Modellierung des Lebenszyklus prozesstechnischer Anlagen und ihrer Komponenten **inklusive Dokumentation** erstellt.
- Um Anwenderinnen und Anwendern ganz unterschiedlicher Expertise und Zielsetzung einen möglichst einfachen Zugang zur Verwendung bzw. Implementierung des Integrationsmodells zu ermöglichen, enthält dieser Bericht auch ein einführendes **Tutorial**.
- Im Projekt wurden **Software-Prototypen** entwickelt, um das Integrationsmodell zu validieren aber auch um die Potentiale der Verwendung eines solchen Modells zu verdeutlichen.
- In Zusammenarbeit mit dem Projekt *ENPRO 2.0 ORCA* und weiteren Beteiligten wurden **nutzer-konfigurierbare Transformationsregeln für die automatische Erzeugung eines MTP-Grundgerüsts aus einem DEXPI-R&I** (und darüber hinaus) aufgestellt. Zu diesen Arbeiten ist ein Artikel verfügbar (Klose u. a., 2021).
- Im Sinne einer **Verbreitung der Ergebnisse** wurden die Arbeiten des Projektes in verschiedenen Standardisierungsgremien, Workshops und auf Konferenzen vorgestellt.
- Das entwickelte **Informationsmodell sowie Anwendungsbeispiele** werden **online**⁷ zur Verfügung gestellt.

1.6. Zusammenarbeit mit anderen Stellen

Während der Projektlaufzeit hat ein kontinuierlicher Austausch mit anderen Projekten aus der zweiten Phase der Initiative *ENPRO* stattgefunden, insbesondere mit den Projekten *ENPRO 2.0 SkaMPi*, *ENPRO 2.0 ORCA*, *ENPRO 2.0 MoProLog* und *ENPRO 2.0 VoPa*. Darüber hinaus wurde durch die aktive Mitarbeit der Projektbearbeiter

⁷<https://modula.aixcape.org>

in der *DEXPI*-Initiative sichergestellt, dass aktuelle Entwicklungen permanent ausgetauscht wurden. Im Projekt *ModuLA* wurden konkrete Empfehlungen bzgl. möglicher Erweiterungen des *DEXPI*-Standards für Module (vgl. dazu Abschnitt [2.1.5.7](#)) formuliert.

2. Eingehende Darstellung

2.1. Eingehende Darstellung der Ergebnisse

2.1.1. Konzept zur Datenintegration

Ziel der Datenintegration ist es, Daten und Informationen, die in unterschiedlichen Formaten und/oder Informationsmodellen (*Domänenmodellen*) vorliegen, zu einem Gesamtmodell zu vereinen. Dazu ist zunächst ein übergreifendes Informationsmodell erforderlich, dessen Ausdrucksstärke der Summe der vorliegenden Domänenmodelle (bzw. deren relevanter Teile) entspricht. Die vorhandenen Domäneninformationen, d.h. die Informationen, die mittels der Domänenmodelle abgebildet sind, sind in einem zweiten Schritt mittels des übergreifenden Informationsmodells abzubilden.

Abbildung 2.1 zeigt schematisch das Konzept aus *ENPRO 1.0 Datenintegration* (von Wedel u. a., 2018). Wesentlich ist, dass die vorhandenen Domänenmodelle und damit die Domäneninformationen *unverändert* übernommen werden (*Domänenlayer*). Die Integration erfolgt in einem separaten *Integrationslayer*, aus dem einzelne Elemente des Domänenlayers referenziert werden und in dem zusätzlich die Querverbindungen als wesentliche Integrationsinformationen abgebildet sind.

2.1.1.1. Domänenlayer

Im Domänenlayer werden Domäneninformationen, beispielsweise Simulationsmodelle, Labordaten, Grundfließbilder, Verfahrensfließbilder, R&Is, Apparatespezifikationen, *MTPs*, Wartungsinformationen oder Betriebsdaten abgelegt. Dazu können letztlich beliebige Informationsmodelle genutzt werden, im Interesse eines einfacheren Datenaustauschs aber vorzugsweise keine proprietären Werkzeugmodelle, sondern akzeptierte Standards (beispielsweise *DEXPI* für R&Is).¹

Wesentlich ist, dass einzelne Elemente im Domänenlayer, die beispielsweise einen technischen Platz *P4711* für eine Pumpe oder eine Pumpe mit der Seriennummer *XY123* repräsentieren, referenziert werden können.² Auf diese Weise wird mittels des Integrationslayers eine Verknüpfung über Domänengrenzen möglich, wie in Abbildung 2.1 veranschaulicht. Es sei darauf hingewiesen, dass keine physikalische Reproduktion der vollstän-

¹Als formaler Rahmen für die Domäneninformationsmodelle wird in dieser Arbeit die UML (*Unified Modeling Language*, UML 2.5.1) vorgeschlagen, s. Abschnitt 2.1.2.4. Informationsmodelle für Domänen liegen in der Regel entweder bereits in UML vor (*DEXPI*-R&I) oder können in einfacher und systematischer Weise auf diese abgebildet werden (Datenbankschemata, XML-Schemata, ...).

²Technisch handelt es sich bei diesen „Elementen“ um UML-Objekte und -Attribute bzw. Datensätze in einer Datenbank, XML-Elemente, PIMS-Datensätze, etc.

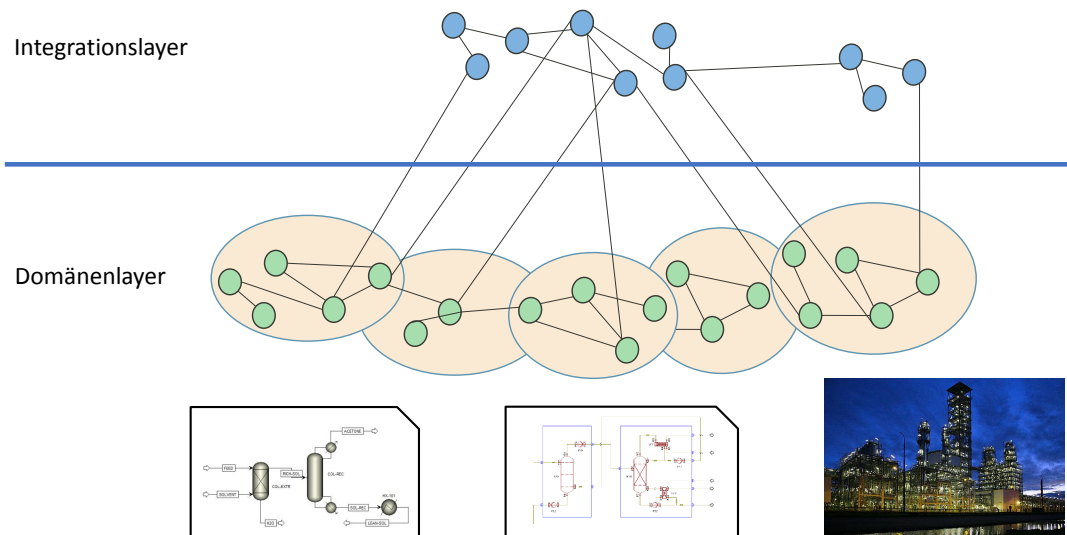


Abbildung 2.1.: Separation von Integrations- und Domänenlayer.³

digen Domäneninformationen erforderlich ist; die Integration kann über eine geeignete Referenzierung der vorhandenen Informationselemente erfolgen, was insbesondere bei großen Datenmengen angebracht ist, etwa Betriebsdaten in einem PIM-System (PIMS).

2.1.1.2. Integrationslayer

Das Schiff, auf welchem Theseus mit seinen Begleitern nach Creta geschift, und glücklich zurück gekommen war, hatte dreyßig Ruder, und wurde von den Atheniensern bis auf die Zeiten des Demetrius Phalereus zum Andenken erhalten, indem man das veraltete Holz wegnahm, und dafür neues und festes einzog. Es gab nachher so gar den Philosophen zu ihren Streitfragen über die Veränderung der Dinge im Wachsthume ein Beyspiel ab, und einige behaupten, es wäre eben dasselbe, und andre, es wäre ein andres Schiff.

— Plutarch, *Theseus* (Übersetzung nach von Schirach, 1777)

Offensichtlich ist die Beantwortung der Frage nach der Identität von Dingen⁴ nicht trivial. Die Realisierung des Integrationskonzeptes in Abbildung 2.1 setzt aber eine stringente Modellierung der Identitäten der Dinge im Integrationslayer voraus. Beispielsweise hat ein technischer Platz *P4711* für eine Pumpe eine andere Identität als eine Pumpe mit der Seriennummer *XY123*, die zu einem bestimmten Zeitpunkt den technischen Platz

³Lizenz zum Bild der Produktionsanlage unten rechts: Attribution: Mikulova, CC BY-SA 3.0 <https://creativecommons.org/licenses/by-sa/3.0>. Quelle: https://commons.wikimedia.org/wiki/File:Slovnaft_-_new_polypropylene_plant_PP3.JPG

⁴Wir verwenden hier den ungewohnt erscheinenden Begriff *Ding* im Interesse einer einheitlichen Terminologie, vgl. das in Abschnitt 2.1.2.3 in Anlehnung an ISO 15926 eingeführte Klassenmodell.

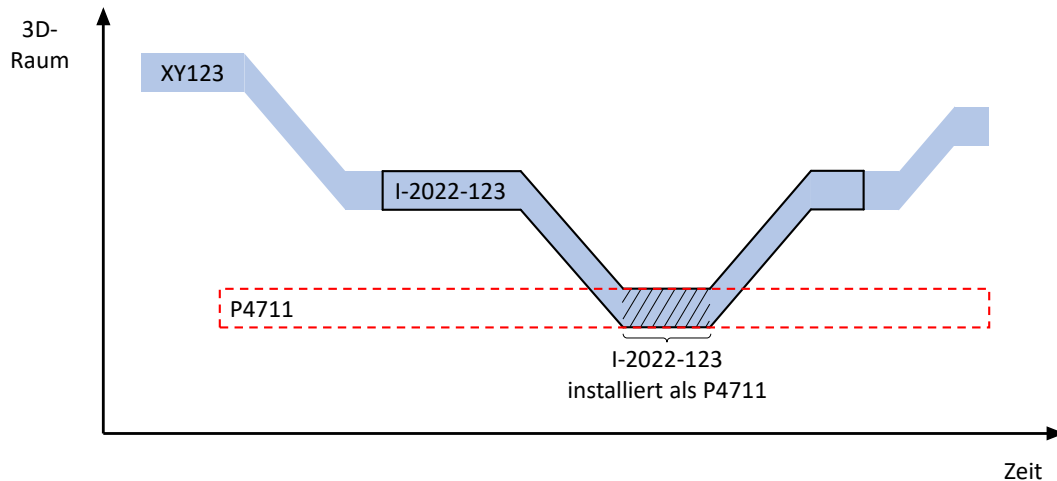


Abbildung 2.2.: Schematische Darstellung von *Dingen* in der Raumzeit.

einnimmt. Im Integrationslayer sind also zwei Identitäten vorzusehen, von denen eine etwa mit der Darstellung der Pumpe in einem R&I zu verknüpfen ist, die andere mit Herstellerinformationen zur Pumpe, etwa Baujahr und Pumpentyp. Die Pumpe mit der Seriennummer *XY123* mag auch eine Inventarnummer haben, z.B. *I-2022-123*, solange sie im Besitz eines Unternehmens ist. Es wäre zu entscheiden, ob diese inventarisierte Pumpe dieselbe Identität hat wie die Pumpe mit der Seriennummer *XY123*.

In der [ISO 15926](#) wird das Paradigma des *Vierdimensionalismus* angewandt. Konkrete Dinge werden als eingebettet in die vierdimensionale Raumzeit betrachtet. Dieses Paradigma liefert ein einfaches Kriterium für Identität: Zwei Dinge sind genau dann identisch, wenn sie demselben Ausschnitt der Raumzeit, kurz 4D-Ausschnitt, entsprechen (*Extensionalismus*, s. West (2011) für eine Einführung).

Zur Illustration des 4D-Paradigmas diene Abbildung 2.2. Die senkrechte Achse repräsentiert den 3D-Raum, die waagerechte Achse die Zeit als vierte Dimension. Die Pumpe mit der Seriennummer *XY123* wird durch den blau unterlegten Bereich dargestellt. Sie existiert in einem bestimmten Zeitbereich, und zu jedem Zeitpunkt ihrer Existenz nimmt sie einen Teil des 3D-Raums ein. Die Pumpe mit der Inventarnummer *I-2022-123* wird in der Abbildung durch den schwarz umrahmten Bereich repräsentiert. Dieser Bereich stellt eine echte Teilmenge des blau unterlegten Bereichs für die Pumpe mit der Seriennummer *XY123* dar.⁵ Gemäß dem Kriterium des Extensionalismus handelt es sich bei den „beiden“ Pumpen um unterschiedliche Dinge in der Raumzeit.

Der rot umrandete Bereich repräsentiert die mit *P4711* bezeichnete Funktion (im Sinne eines technischen Platzes) für eine Pumpe in einer Produktionsanlage. Im schwarz schraffierten Bereich überdecken sich *P4711* und *I-2022-123*; dieser Bereich entspricht dem Zustand, dass die Pumpe *I-2022-123* (bzw. die Pumpe *XY123*) in der Funktion *P4711* installiert ist (als *installiertes Ding*, s. [InstalledThing](#), Abschnitt 2.1.2.3.2). Sowohl

⁵Genauer: Die Pumpe mit der Inventarnummer ist ein *zeitlicher Teil* der Pumpe mit der Seriennummer, s. [TemporalAggregation](#), Abschnitt 2.1.2.3.5.

P4711 als auch das installierte Ding sind wiederum als Dinge mit eigener Identität zu betrachten.

Das 4D-Paradigma beantwortet nicht die Frage der *Athenienser*, ob sie noch immer stolze Besitzer von *Theseus' Schif* sind, und es beantwortet nicht die Frage eines Anlagenbetreibers, wo die Wartungsinformationen zur Pumpe *P4711* oder zur Pumpe mit der Seriennummer *XY123* zu finden sind. Es erlaubt aber, die Bedeutung von *Theseus' Schif* und die Bedeutung von Pumpe *P4711* genau zu definieren.

Das vorgeschlagene Konzept zur Datenintegration zeichnet sich dadurch aus, dass der Detaillierungsgrad und die Tiefe der Integration flexibel gehandhabt werden können. In einem minimalen Ansatz würde im Integrationslayer lediglich die Identität von Dingen abgebildet werden. Für alle weiteren Informationen würde auf den Domänenlayer verwiesen werden.

Dieser minimale Ansatz kann schrittweise erweitert werden, indem solche Domäneninformationen, die über Domänengrenzen relevant sind, auch im Integrationslayer abgebildet werden. Dies ist grundsätzlich mit zusätzlichem Aufwand und erhöhter Komplexität verbunden, da eine Abbildung von den Informationsmodellen der Domänen auf den Integrationslayer zu definieren und zu implementieren ist.⁶ Deshalb sollten derartige Ergänzungen im Integrationslayer nur bei Bedarf, d.h. in Abhängigkeit von den vorgesehenen Anwendungen, in Betracht gezogen werden.

Dieses Konzept unterscheidet sich vom Ansatz, der nach Kenntnisstand der Autoren zur Zeit schwerpunktmäßig in der Community zur [ISO 15926](#) verfolgt wird.⁷ Dort wird versucht, sämtliche Lebenszyklus-Informationen mit den Mitteln der ISO 15926 abzubilden. Aus Sicht der Autoren dieses Berichtes hat ein solcher „vollständiger“ Integrationslayer sicherlich Vorteile; insbesondere bietet er einen nicht nur formal, sondern auch semantisch einheitlichen Zugriff auf sämtliche Informationen. Eine Schwierigkeit dabei mag aber sein, dass die oben angesprochene Komplexität im Integrationslayer voll zur Geltung kommt. Mit dem in diesem Bericht vorgeschlagenen Konzept zur Datenintegration soll eine bzgl. der Komplexität schrittweise Implementierung in kommerziellen Software-Werkzeugen unterstützt werden.

2.1.2. Informationsmodell zur Lebenszyklus-Modellierung

Das *Informationsmodell zur Lebenszyklusmodellierung* stellt Modellierungselemente (Datentypen, Klassen, etc.) bereit, mit denen Lebenszyklusinformationen gemäß den Konzepten aus *ENPRO 1.0 Datenintegration* und Abschnitt 2.1.1 abgebildet werden können. Es ist eine Weiterentwicklung des *Object Integration Model for the Engineering Lifecycle* (OIMEL) aus *ENPRO 1.0 Datenintegration* unter Berücksichtigung der Anforderungen einer modulbasierten Vorgehensweise. Im Vergleich zum OIMEL wurden einzelne

⁶Ursächlich für diesen Aufwand sind die Unterschiede zwischen den Informationsmodellen der Domänen, die in der Regel unabhängig voneinander entstanden sind. In einfachen Fällen handelt es sich lediglich um terminologische Unterschiede, typisch sind aber wesentliche strukturelle Unterschiede, vgl. die Darstellung zu *DEXPI* und *MTP* in Abschnitt 2.1.5.

⁷vgl. <https://15926.org/home>

Aspekte, die zunächst nur konzeptionell behandelt wurden (etwa die Domänenmodelle), formalisiert.

Das Informationsmodell ist ein Klassenmodell gemäß der *Unified Modeling Language* (UML 2.5.1). Es ist in 5 Pakete aufgeteilt:

- **core**: **Item** als grundlegende Basisklasse;
- **primitive_types**: primitive Datentypen, z.B. für Zahlen und Zeichenketten;
- **integration**: modifizierte Teilmenge des Datenmodells aus [ISO 15926-2](#) (*Integrationlayer*);
- **domain**: modifizierte Teilmenge der [UML 2.5.1](#), erweitert um die Darstellung von Instanzen (*Domänenlayer*);
- **decisions**: Erweiterung des Integrationslayers zur Entscheidungsmodellierung, basierend auf der *Decision Representation Language* (Lee und Lai, 1991).

2.1.2.1. Paket core

Das Paket enthält die Klasse **Item** als Basisklasse für alle weiteren Klassen (im Folgenden nicht mehr explizit erwähnt).

2.1.2.2. Paket primitive_types

Im Paket **primitive_types** werden primitive Datentypen wie **string** und **double** definiert, die von den übrigen Paketen genutzt werden. Auf eine formale Definition der Datentypen (Wertebereich, ggf. Präzision, etc.) wurde in *ModuLA* als Projekt mit konzeptionellem Schwerpunkt verzichtet. Wir gehen davon aus, dass die Datentypen den gleichnamigen XML-Datentypen entsprechen.

2.1.2.3. Paket integration

Das Paket **integration** stellt eine modifizierte Teilmenge der Klassen aus [ISO 15926-2](#) bereit. Wir verzichten an dieser Stelle auf ausführliche Erläuterungen und elementare Anwendungsbeispiele. In diesem Abschnitt geben wir lediglich eine Übersicht über alle Klassen des Pakets **integration** sowie einige grundlegende Erläuterungen. Die Darstellung hier soll dem Leser ermöglichen, bei der Durchsicht der übergreifenden Anwendungsbeispiele, insbesondere in Abschnitt 2.1.3, die dort verwendeten Klassen in den Kontext einordnen zu können. In der online⁸ bereitgestellten Dokumentation des Pakets **integration** wird auf Korrespondenzen zu Entitätstypen in der [ISO 15926-2](#) verwiesen, insbesondere werden Erweiterungen bzw. Modifikationen beschrieben.

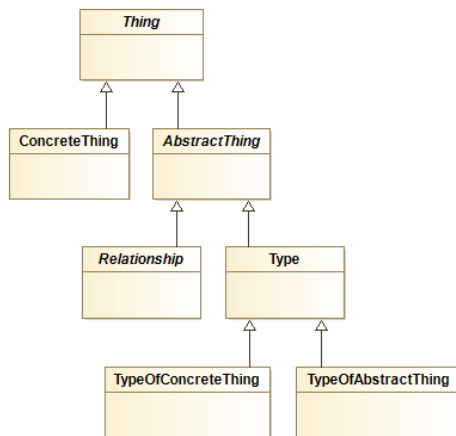


Abbildung 2.3.: Paket integration: *Thing* und weitere Top-Level-Klassen.

2.1.2.3.1. Klasse *Thing*

Thing (s. Abbildung 2.3) ist die abstrakte Basisklasse für alle weiteren Klassen des Pakets integration.⁹ *Thing* hat zwei Unterklassen:

- Ein konkretes Ding (*ConcreteThing*) repräsentiert einen Ausschnitt der vierdimensionalen Raumzeit (s. Abschnitt 2.1.1.2). In Abschnitt 2.1.2.3.2 werden Unterklassen für 4D-Ausschnitte mit besonderen Eigenschaften eingeführt.
- Alles andere sind abstrakte Dinge (*AbstractThings*).¹⁰ Wir unterscheiden dabei zwischen Beziehungen und Typen.
 - Eine Beziehung (*Relationship*, Abschnitt 2.1.2.3.5) beschreibt das Verhältnis zweier Dinge zueinander.
 - Ein Typ (*Type*) ist eine Menge von Dingen, wobei zwischen Typen konkreter Dinge (*TypeOfConcreteThing*) und Typen abstrakter Dinge (*TypeOfAbstractThing*) unterschieden wird.¹¹

2.1.2.3.2. Klasse *ConcreteThing*

Ein *ConcreteThing* (s. Abbildung 2.4) beschreibt einen beliebigen Ausschnitt der vierdimensionalen Raumzeit, wie sie in Abschnitt 2.1.1.2 eingeführt wurde.¹² Im Folgenden gehen wir auf Unterklassen von *ConcreteThing* ein, die ggf. zur genaueren Charakterisierung solcher Ausschnitte genutzt werden können. Es ist zu beachten, dass 4D-Ausschnitte

⁸<https://modula.aixcape.org>

⁹vgl. Entitätstyp `thing` der ISO 15926-2

¹⁰vgl. Entitätstyp `abstract_thing` der ISO 15926-2

¹¹vgl. Entitätstyp `class` der ISO 15926-2

¹²vgl. Entitätstyp `possible_individual` der ISO 15926-2

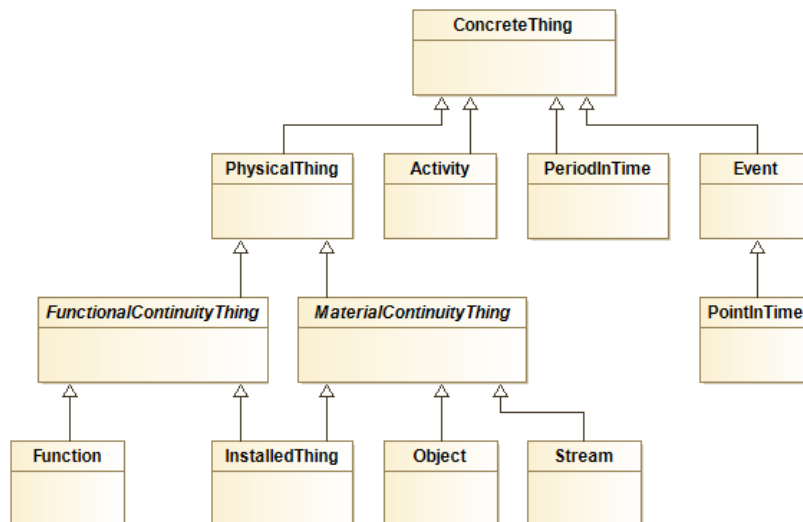


Abbildung 2.4.: Paket integration: **ConcreteThing** und Unterklassen.

denkbar sind, auf die keine dieser Charakterisierungen zutrifft. Die Klasse **ConcreteThing** kann in solchen Fällen direkt genutzt werden, da sie nicht abstrakt ist.

Ein **PhysicalThing** ist eine 4D-Verteilung von Materie oder Energie.¹³ Entsprechend der Unterscheidung von *funktionaler* und *materieller Identität* physikalischer Dinge werden zwei abstrakte Unterklassen *FunctionalContinuityThing* und *MaterialContinuityThing* eingeführt. Diese werden weiter spezialisiert zu vier konkreten Klassen:

- Eine **Function** ist ein physikalisches Ding mit funktionaler Identität.¹⁴
- Ein **Object** ist ein physikalisches Ding, dessen Identität auf kontinuierlicher Erhaltung wesentlicher Teile seiner Materie oder Energie beruht.¹⁵ Dabei ist es zulässig, dass einzelne Komponenten ersetzt werden.
- Ein **Stream** ist ein physikalisches Ding, dessen Materie einem kontinuierlichen oder wiederholten Ab- und Zugang unterliegt.¹⁶
- Ein **InstalledThing** vereint funktionale und materielle Identität.

Ein **Event** ist ein 4D-Ausschnitt ohne zeitliche Ausdehnung.¹⁷ Dabei ist es zulässig, dass das **Event** an unterschiedlichen räumlichen Position zu unterschiedlichen Zeitpunkten stattfindet. Die Spezialisierung **PointInTime** zeichnet sich dadurch aus, dass der gesamte 3D-Raum zu einem einzigen Zeitpunkt beschrieben wird.¹⁸

¹³vgl. Entitätstyp `physical_object` der [ISO 15926-2](#)

¹⁴vgl. Entitätstyp `functional_physical_object` der [ISO 15926-2](#)

¹⁵vgl. Entitätstyp `materialized_physical_object` der [ISO 15926-2](#)

¹⁶vgl. Entitätstyp `stream` der [ISO 15926-2](#)

¹⁷vgl. Entitätstyp `event` der [ISO 15926-2](#)

¹⁸vgl. Entitätstyp `point_in_time` der [ISO 15926-2](#)

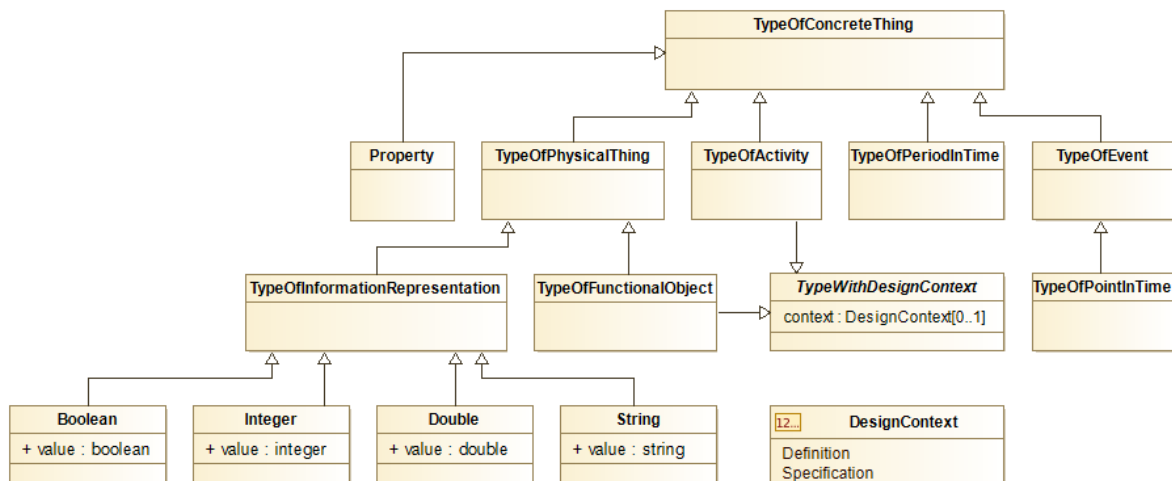


Abbildung 2.5.: Paket integration: [TypeOfConcreteThing](#) und Unterklassen.

Eine [PeriodInTime](#) umfasst den gesamten 3D-Raum zwischen zwei [PointsInTime](#).¹⁹
 Eine [Activity](#) ist ein 4D-Ausschnitt, der durch einen Vorgang gekennzeichnet ist.²⁰

2.1.2.3.3. Klasse [TypeOfConcreteThing](#)

Ein [TypeOfConcreteThing](#) (s. [Abbildung 2.5](#)) beschreibt eine Art oder Menge von [ConcreteThings](#), d.h. von 4D-Ausschnitten.²¹

Die Unterklassen von [TypeOfConcreteThing](#) sind teilweise analog zu den Unterklassen von [ConcreteThing](#) angelegt: Ein [TypeOfActivity](#) ist eine Art von [Activities](#), ein [TypeOfEvent](#) ist eine Art von [Events](#), und entsprechend für [TypeOfPointInTime](#), [TypeOfPeriodInTime](#) und [TypeOfPhysicalThing](#).

[TypeOfFunctionalObject](#), eine Spezialisierung von [TypeOfPhysicalThing](#), beschreibt eine Art von physikalischen Dingen, die im Hinblick auf eine erwünschte Funktion festgelegt wird.²²

Sowohl ein [TypeOfActivity](#) als auch ein [TypeOfFunctionalObject](#) können sich durch einen [DesignContext](#) auszeichnen (Attribut `context`, geerbt von [TypeWithDesignContext](#)). Der [DesignContext](#) kann einer der literalen Werte [Specification](#) und [Definition](#) sein. Die Klasse [DesignContext](#) wurde eingeführt, um eine klare Unterscheidung treffen zu können zwischen Eigenschaften, die ein Ding per Definition hat (z.B. eine Pumpe zu sein), und Eigenschaften, die es in der Regel *haben sollte* (z.B. pumpen zu können). Für ein Anwendungsbeispiel verweisen wir auf [Abschnitt 2.1.3.2](#).

[TypeOfInformationRepresentation](#) ist eine weitere Spezialisierung von [TypeOfPhysicalThing](#). Ein [TypeOfInformationRepresentation](#) ist eine Menge physikalischer Dinge, die ein

¹⁹vgl. Entitätstyp `period_in_time` der [ISO 15926-2](#)

²⁰vgl. Entitätstyp `activity` der [ISO 15926-2](#)

²¹vgl. Entitätstyp `class_of_individual` der [ISO 15926-2](#)

²²vgl. Entitätstyp `class_of_functional_object` der [ISO 15926-2](#)

bestimmtes Informationsmuster repräsentieren.²³ Unterklassen von [TypeOfInformationRepresentation](#) umfassen damit auch Werte von Datentypen, beispielsweise [String](#) und [Double](#).²⁴

Eine [Property](#) ist eine Menge von [ConcreteThings](#), die eine bestimmte Eigenschaft teilen.²⁵ Einfache Beispiele für eine [Property](#) sind *Height = 2m*, die Menge derjenigen Dinge mit einer Höhe von 2 m, und *Gelb*, die Menge der gelben (bzw. gelb erscheinenden) Dinge.

2.1.2.3.4. Klasse [TypeOfAbstractThing](#)

Ein [TypeOfAbstractThing](#) (s. Abbildung 2.6) repräsentiert eine Menge von [AbstractThings](#).²⁶

Ein [TypeOfType](#) ist eine Typisierung von [Types](#). Relevant ist vor allem die Unterklasse [TypeOfProperty](#), die genutzt werden kann, um [Properties](#) nach zugrundeliegenden Kriterien zu typisieren. Beispiele sind *Höhe* und *Farbe*. [TypeOfProperty](#) entspricht also der intuitiven Vorstellung einer Eigenschaft, und [Property](#) entspricht der Vorstellung eines Eigenschaftswertes.

Ein [TypeOfRelationship](#) ist eine Typisierung von [Relationships](#). Die Hierarchie der Unterklassen von [TypeOfRelationship](#) in Abbildung 2.6 folgt weitgehend der Hierarchie der Unterklassen von [Relationship](#) (Abschnitt 2.1.2.3.5). Diese Unterklassen werden im Wesentlichen zur Modellierung geplanter [Things](#) genutzt (s. Abschnitt 2.1.3.2).

2.1.2.3.5. Klasse [Relationship](#)

[Relationship](#) (s. Abbildung 2.7) ist die abstrakte Oberklasse für eine Reihe von Beziehungen, die zwischen [Things](#) bestehen können.²⁷

Eine [Typification](#) beschreibt die Zugehörigkeit eines [Thing](#) (Attribut [typed](#)) zu einem [Type](#) (Attribut [type](#)).²⁸ Eine [Specialization](#) zwischen zwei [Types](#) besagt, dass der [subtype](#) eine Teilmenge des [supertype](#) ist.²⁹

Eine [Aggregation](#) ist eine Teil-Ganzes-Beziehung zwischen einem [ConcreteThing](#) als Ganzem ([whole](#)) und einem weiteren [ConcreteThing](#) als Teil ([part](#)).³⁰ Die Semantik der [Aggregation](#) ist über eine Teilmengenbeziehung in der vierdimensionalen Raumzeit definiert: Die Menge der 4D-Punkte des Teils ist eine Teilmenge der 4D-Punkte des Ganzen. Für einige Sonderfälle werden Spezialisierungen der [Aggregation](#) eingeführt:

²³vgl. Entitätstyp [class_of_information_representation](#) der [ISO 15926-2](#)

²⁴Die [PhysicalThings](#), die einem [TypeOfInformationRepresentation](#) angehören, sind in realistischen Modellen häufig nicht von Bedeutung. Dem [String](#) mit `value="Abschlussbericht"` gehört beispielsweise das [Object](#) an, das den entsprechenden Bereich des Papiers oder Monitors repräsentiert, wenn Sie die Titelseite dieses Berichts lesen.

²⁵vgl. Entitätstyp [property](#) der [ISO 15926-2](#)

²⁶vgl. Entitätstyp [class_of_abstract_object](#) der [ISO 15926-2](#)

²⁷vgl. Entitätstyp [relationship](#) der [ISO 15926-2](#)

²⁸vgl. Entitätstyp [classification](#) der [ISO 15926-2](#)

²⁹vgl. Entitätstyp [specialization](#) der [ISO 15926-2](#)

³⁰vgl. Entitätstyp [composition_of_individual](#) der [ISO 15926-2](#)

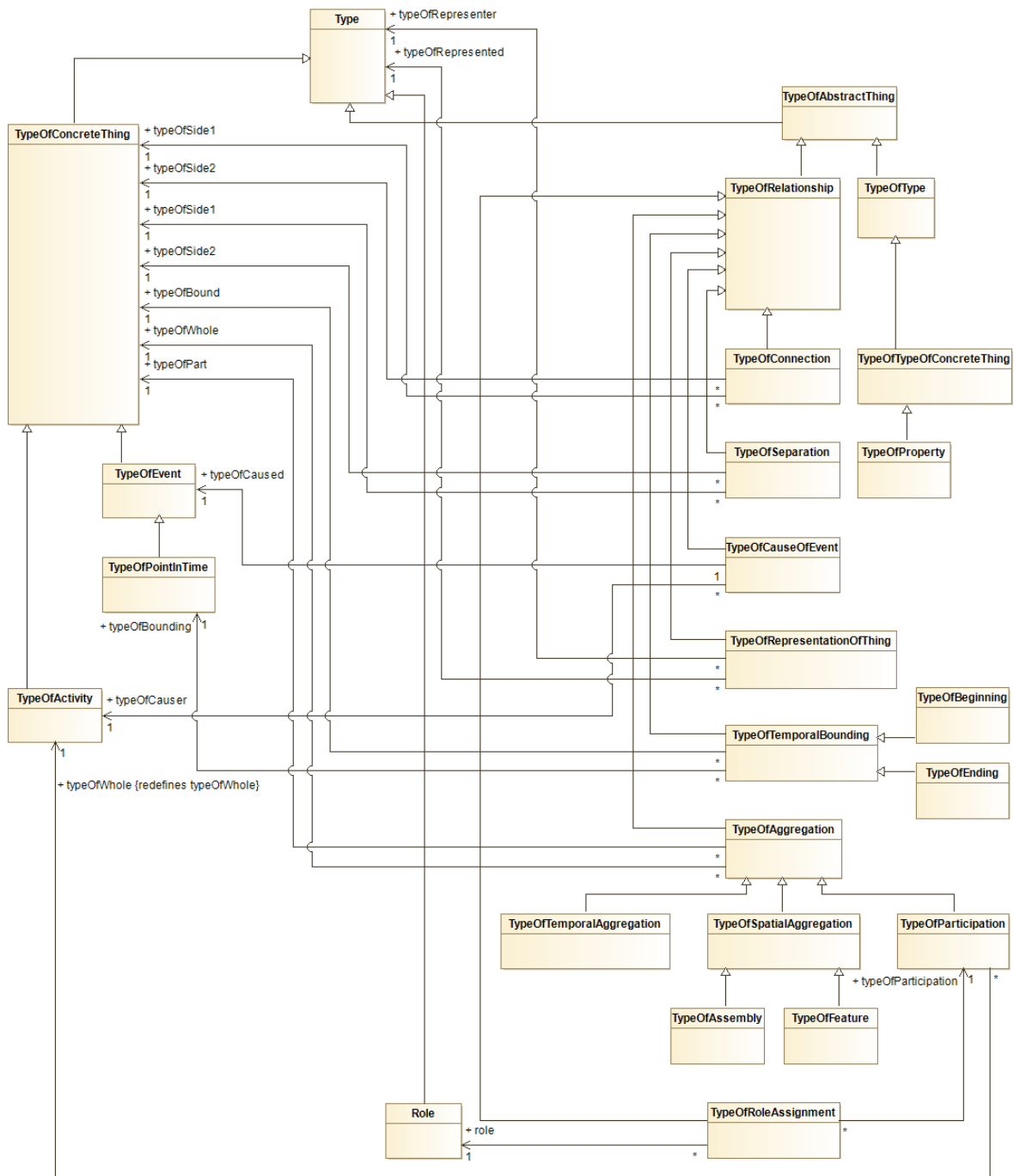


Abbildung 2.6.: Paket integration: [TypeOfAbstractThing](#) und Unterklassen.

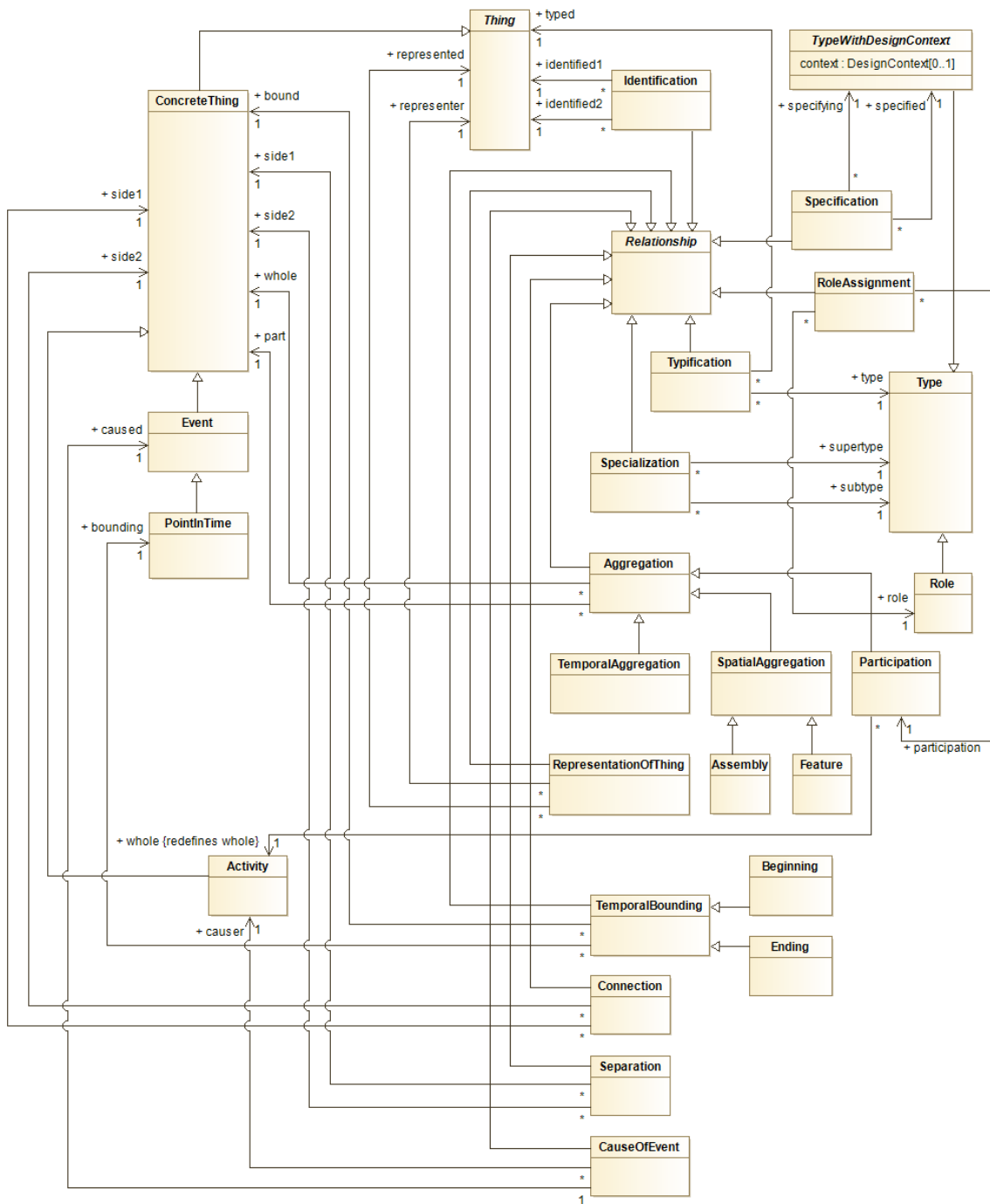


Abbildung 2.7.: Paket integration: *Relationship* und Unterklassen.

- Eine **TemporalAggregation** besagt, dass der Teil zu bestimmten Zeiten den gesamten 3D-Ausschnitt des Ganzen ausfüllt (*zeitlicher Teil*).³¹
- Eine **SpatialAggregation** besagt, dass zu allen Zeiten, in denen das Ganze besteht, der Teil innerhalb des Raumes des Ganzen liegt (*räumlicher Teil*).³² Bei der **SpatialAggregation** wird zudem zwischen einer **Assembly** und einem **Feature** unterschieden. Bei einer **Assembly**³³ ist es zumindest gedanklich möglich, das Teil vom Ganzen zu trennen, bei einem **Feature**³⁴ hingegen nicht.
- Eine **Participation** beschreibt die Teilnahme eines **ConcreteThing** an einer **Activity** (vgl. die Typ-Redefinition des Attributs `whole` bei **Participation**).³⁵

2.1.2.3.6. Notation für Relationships

Im Informationsmodell sind **Relationships** (und **TypesOfRelationship**) als Klassen im Sinne der UML definiert. Instanzen der **Relationships** sind damit vollwertige Objekte, man spricht von *Reifikation*. Auf diese Weise können Instanzen von **Relationship** durch andere Instanzen von **Relationship** referenziert werden.

Ein Nachteil der Reifikation ist die erhöhte Komplexität bei der graphischen Darstellung von Modellbeispielen in Objektdiagrammen. Wir führen deshalb eine vereinfachte Notation ein, die eine Instanz einer **Relationship** durch eine Verbindungslinie repräsentiert (s. Abbildung 2.8). Die genutzte konkrete **Relationship**-Klasse geht aus der graphischen Gestaltung der Verbindungslinie hervor. In einigen Fällen enthält die Verbindungslinie zusätzlich eine Beschriftung; beispielsweise gibt bei der **Aggregation** (unten links in der Abbildung) die Beschriftung ggf. eine Unterklasse der **Aggregation** an.

2.1.2.3.7. Templates

Es wurden einige *Templates* definiert, die als abkürzende Notation für wiederkehrende Muster aus Modellierungselementen genutzt werden können (vgl. ISO 15926-7). Diese sind in Abbildung 2.9 dargestellt.

2.1.2.4. Paket domain

Das Paket `domain` dient dazu, Informationen im Domänenlayer in der Weise abzubilden, in der sie in Domänenmodellen (z.B. Toolmodellen, idealerweise aber Standards wie z.B. DEXPI 1.3) vorliegen. Um beispielsweise abzubilden, dass ein Objekt in einer DEXPI-Datei ein *DEXPI-Tank* ist, ist es zwangsläufig erforderlich, sich auf die DEXPI-Klasse *Tank* zu beziehen. Das Paket `domain` muss deshalb die Abbildung sowohl der Informationsmodelle für Domänen als auch der Domänen-Informationen unterstützen.

³¹vgl. Entitätstyp `temporal_whole_part` der ISO 15926-2

³²vgl. Entitätstyp `arrangement_of_individual` der ISO 15926-2

³³vgl. Entitätstyp `assembly_of_individual` der ISO 15926-2

³⁴vgl. Entitätstyp `feature_whole_part` der ISO 15926-2

³⁵vgl. Entitätstyp `participation` der ISO 15926-2

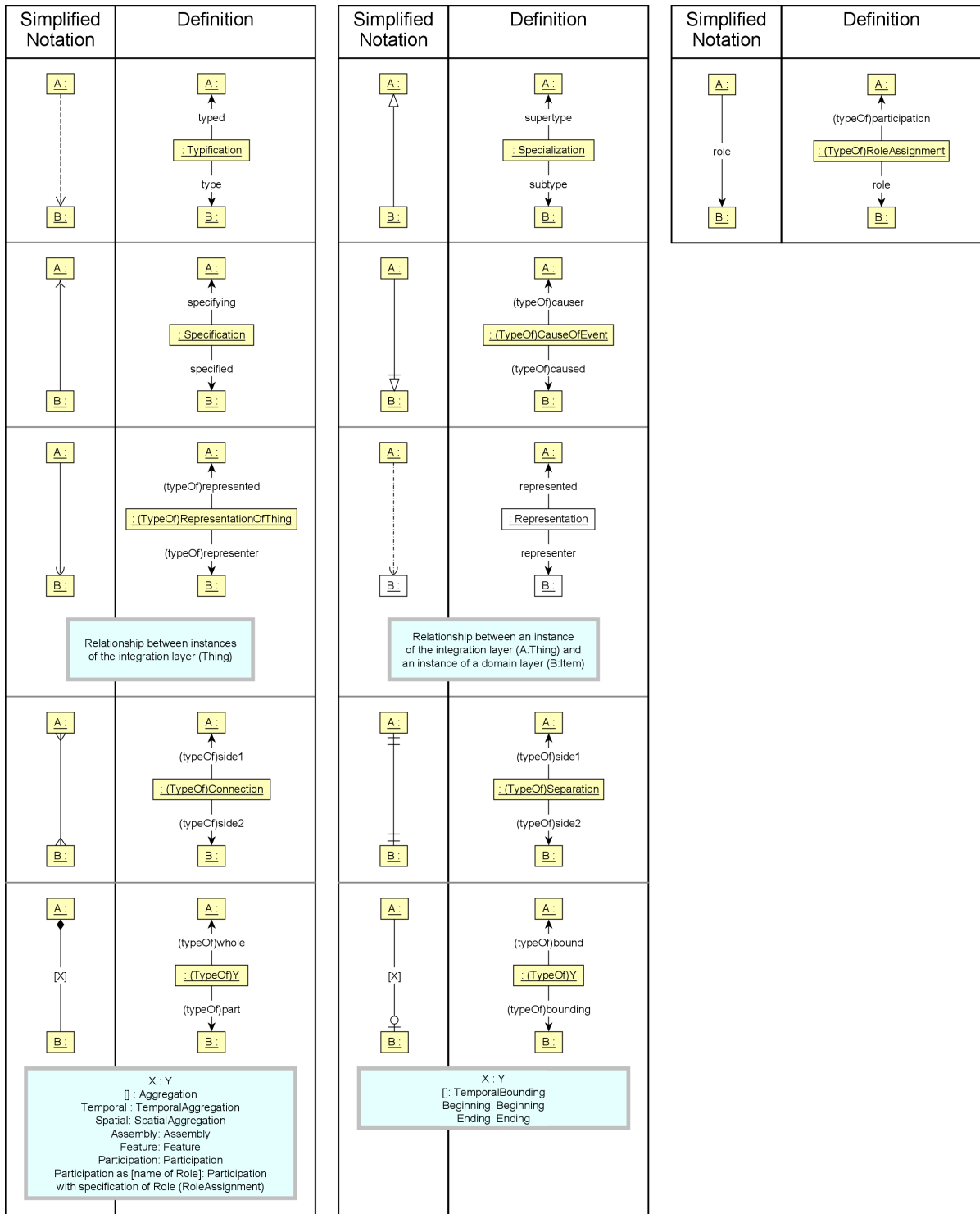


Abbildung 2.8.: Vereinfachte Notation für häufig wiederkehrende *Relationships* und *TypesOfRelationship*.

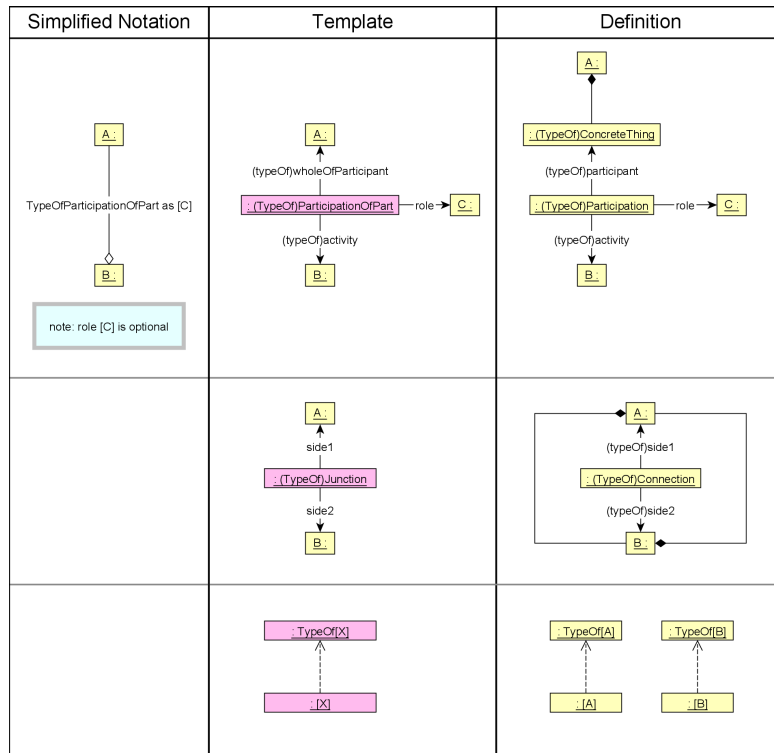


Abbildung 2.9.: Definition der in den Beispielen genutzten Templates.

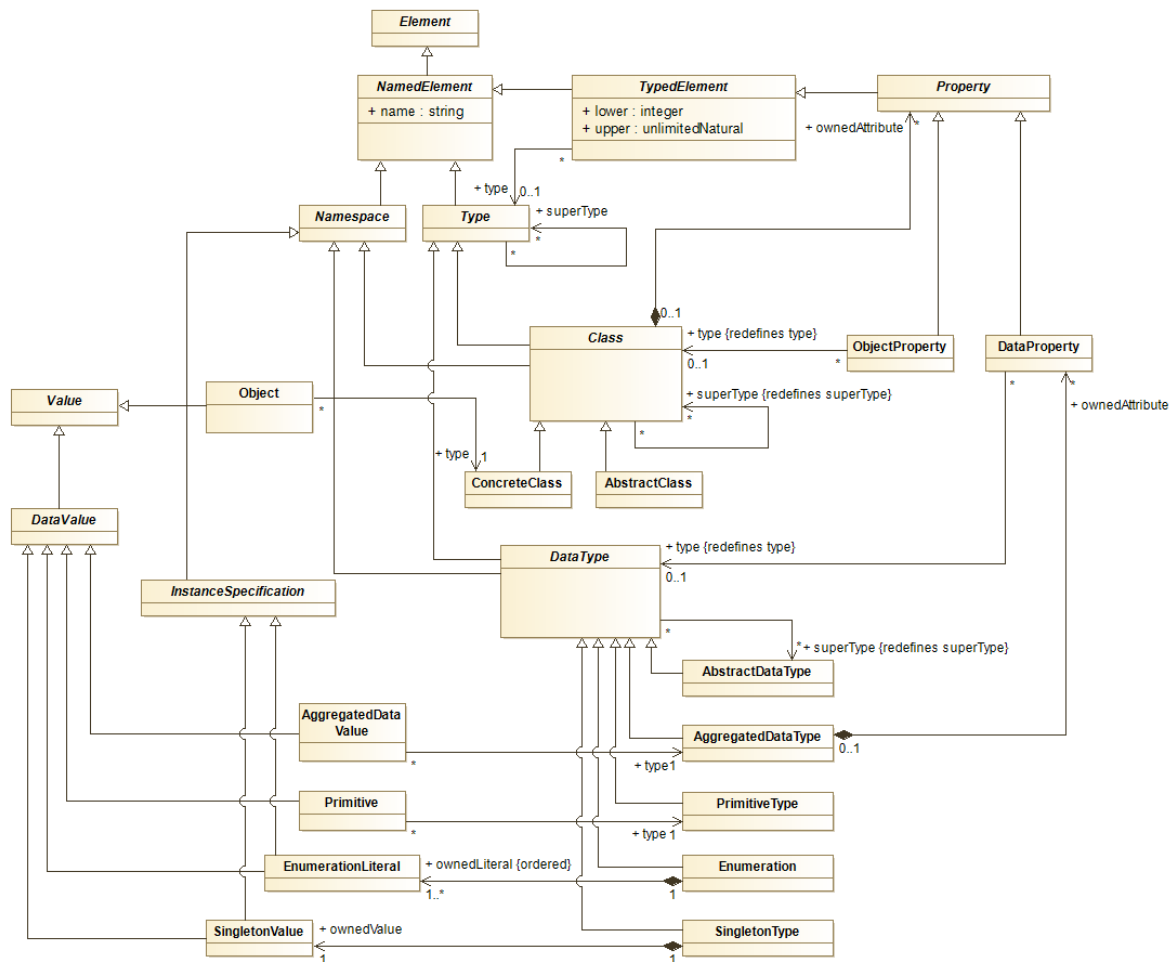


Abbildung 2.10.: Paket domain (basierend auf UML 2.5.1).

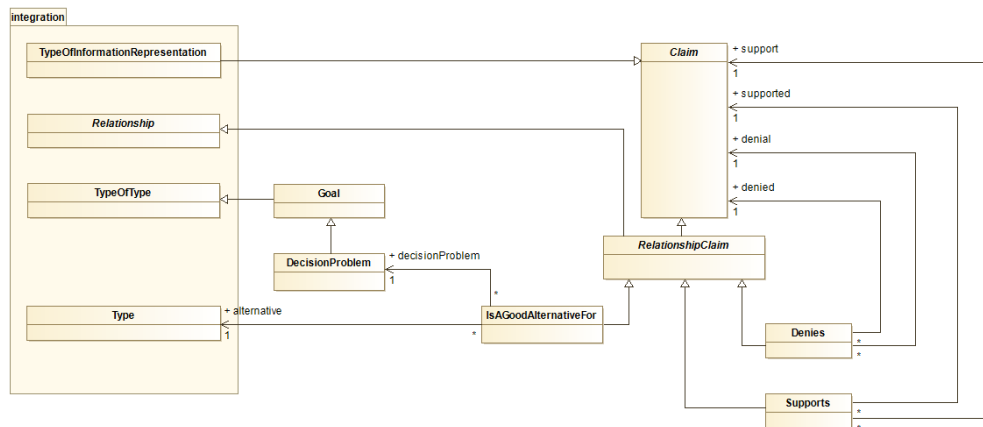


Abbildung 2.11.: Paket decisions.

Diese Anforderung wird durch die [UML 2.5.1](#) erfüllt. Deshalb wurde eine geeignete Teilmenge der UML bestimmt und in das Paket `domain` (s. [Abbildung 2.10](#)) übernommen. Wir verzichten deshalb hier auf eine Beschreibung der Klassen und verweisen auf [UML 2.5.1](#).

Im Interesse einer einfacheren Implementierung wurden einige formale Modifikationen im Vergleich zur UML vorgenommen, beispielsweise:

- die Angabe des `name` eines `NamedObjects` ist erforderlich;
- auf die `Generalization` der UML wurde verzichtet, stattdessen wird ein Attribut `superType` für `Type` genutzt;
- einige abstrakte Basisklassen (z.B. `EncapsulatedClassifier` und `BehavioredClassifier` für `Class`) wurden nicht berücksichtigt, da sie hier keinen Mehrwert bieten.

2.1.2.5. Paket decisions

Das Paket `decisions` ist eine Erweiterung des Pakets `integration`. Basierend auf der *Decision Representation Language* (DRL) von Lee und Lai (1991) ermöglicht es, eine Begründung für bestimmte Sachverhalte anzugeben, insbesondere für Design-Entscheidungen wie die Auswahl eines bestimmten Equipment-Typen zur Realisierung einer Grundoperation. Für Anwendungsbeispiele verweisen wir nicht nur auf [Abschnitt 2.1.3](#) im vorliegenden Bericht, sondern auch auf den Abschlussbericht von *ENPRO 2.0 SkaMPi* (Grünwald, 2021).

[Abbildung 2.11](#) zeigt die wesentlichen Klassen des Pakets `decisions` mit Fokus auf dem Zusammenspiel mit Klassen aus dem Paket `integration`. Ein `Claim` repräsentiert eine Behauptung im Zusammenhang mit einem Entscheidungsproblem (`DecisionProblem`). Ein `Claim` kann einerseits eine Information beliebiger Art sein, zum Beispiel ein String mit einem erläuternden Text oder ein als Black-Box betrachtetes Dokument. Diese Fälle

werden durch die Generalisierung von `TypeOfInformationRepresentation` aus dem Paket `integration` zu `Claim` erfasst. Andererseits kann ein `Claim` ein `RelationshipClaim` sein, eine spezielle Art einer `Relationship`. Beispielsweise repräsentiert ein `Supports`, dass ein `Claim` einen anderen unterstützt, und ein `Denies`, dass ein `Claim` einen anderen bestreitet. Auf diese Weise können bei Bedarf komplexe Argumentationen abgebildet werden. `IsAGoodAlternativeFor` schließlich dient der Zuordnung einer Alternative zu einem `DecisionProblem`. Es ist zu beachten, dass der Typ von `alternative` gerade `Type` ist. Dies liegt in der Modellierung *geplanter* Dinge mittels `Types` begründet. Entsprechend ist `DecisionProblem` indirekt über `Goal` eine Spezialisierung von `TypeOfType`; ein `Type` kann auf diese Weise als `DecisionProblem` typisiert werden, d.h. als ein `Type`, der weiter auszuarbeiten ist.

2.1.3. Tutorial zur Lebenszyklus-Modellierung

In diesem Abschnitt zeigen wir anhand eines einfachen übergreifenden Beispiels, wie das Informationsmodell gemäß Abschnitt 2.1.2 genutzt werden kann, um Informationen zum Lebenszyklus einer Anlage darzustellen. Dieser Lebenszyklus umfasst die Planung eines Prozesses, die Planung einer Anlage zur Realisierung des geplanten Prozesses, die reale Anlage im Betrieb und schließlich den real ablaufenden Prozess in der Anlage. Diesem Zyklus von der Prozessplanung bis hin zum real ablaufenden Prozess in einer Anlage können die Aspekte des *Asset Life Cycles*, wie sie im Projekt *ENPRO 1.0 Datenintegration* definiert wurden (von Wedel u. a., 2018), wie folgt zugeordnet werden:

- geplanter Prozess: funktionale Anforderungen (*functional requirements, FR*)
- geplante Anlage: funktionales Design (*functional design, FD*) und Spezifikation von Assets (*asset specification, AS*)
- reale Anlage und realer Prozess: Asset im Betrieb (*asset in operation, AiO*)

Um die möglichst allgemein gehaltene Beschreibung zu veranschaulichen, betrachten wir exemplarisch einen Prozess *Pulvertrocknung* von der Planung bis zum realen Ablauf in einer Anlage.

Wir zeigen im Folgenden Ausschnitte aus einem Gesamtmodell³⁶ (s.a. Abbildung A.1).

Das Farbschema (s. Abbildung 2.12) für RDL-*Things* (blau), Katalog-*Things* (grün) und Projekt-*Things* (gelb) wird auch in den Modellen zu den Szenarien genutzt.

2.1.3.1. Modellierung von Planungsinformationen

Zur Modellierung von Planungsinformationen dienen Unterklassen der Klasse `Type`. Es werden lediglich die Typen, das heißt Mengen von Dingen, die bestimmte Kriterien erfüllen, festgelegt. Im Wesentlichen sind es die Typen von konkreten Dingen und die Typen der Beziehungen zwischen diesen konkreten Dingen. Typen von konkreten Dingen,

³⁶<https://modula.aixcape.org/Tutorial-Life-Cycle-Modeling.svg>

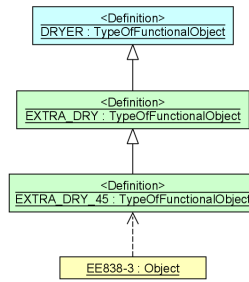


Abbildung 2.12.: Farbschema für Dinge (*Thing*).

die im Rahmen von Planungsinformationen auftreten, sind in der Regel Spezialisierungen von RDL-Einträgen.

Für eine Auswahl von Typen konkreter Dinge, die wohl am häufigsten bei der Modellierung von Planungsinformationen auftreten, wird der **DesignContext** unterschieden. Neben allgemeingültigen Typeninformationen, die im Design-Kontext **Definition** beschrieben sind, werden spezifische Typeninformationen im Kontext **Specification** angegeben. Es können unterschiedliche Spezifikationen zu ein und derselben Definition existieren. Der Design-Kontext wird zur Vereinfachung in der folgenden Beschreibung nicht bei jedem Vorkommen explizit genannt.

2.1.3.2. Modellierung eines geplanten Prozesses (FR)

Siehe dazu das Instanzbeispiel in Abbildung 2.13.

- Prozessschritte sind Typen von Aktivitäten (**TypeOfActivity**). Sie sind Spezialisierungen von RDL-Einträgen, wie z.B. das Trocknen *A1_Drying* im geplanten Prozess *P1_PowderDrying*.
- Zustände im Prozess (z.B. *S2_DryPowder*) sind Typen funktionaler Objekte (**TypeOfFunctionalObject**).
- Zustände können in Beziehung zu einzelnen Prozessschritten stehen. Sie können an einem Prozessschritt vollständig (**TypeOfParticipation**) oder teilweise (**TypeOfParticipationOfPart**) bzgl. ihrer Raumzeit-Ausdehnung teilnehmen und dies in einer bestimmten Rolle (**Role**), z.B. als Edukt-, Produkt- oder Prozessstrom. Die Rollen werden durch Verweis auf RDL-Einträge spezifiziert.
- Ein geplanter Prozess ist eine Aggregation (**TypeOfAggregation**) von einzelnen Prozessschritten.

2.1.3.3. Modellierung einer geplanten Anlage (FD)

Im funktionalen Design werden die geplanten Funktionen der zu bauenden Anlage festgelegt. Siehe dazu das Instanzbeispiel in Abbildung 2.14, in dem die geplante Anlage aus

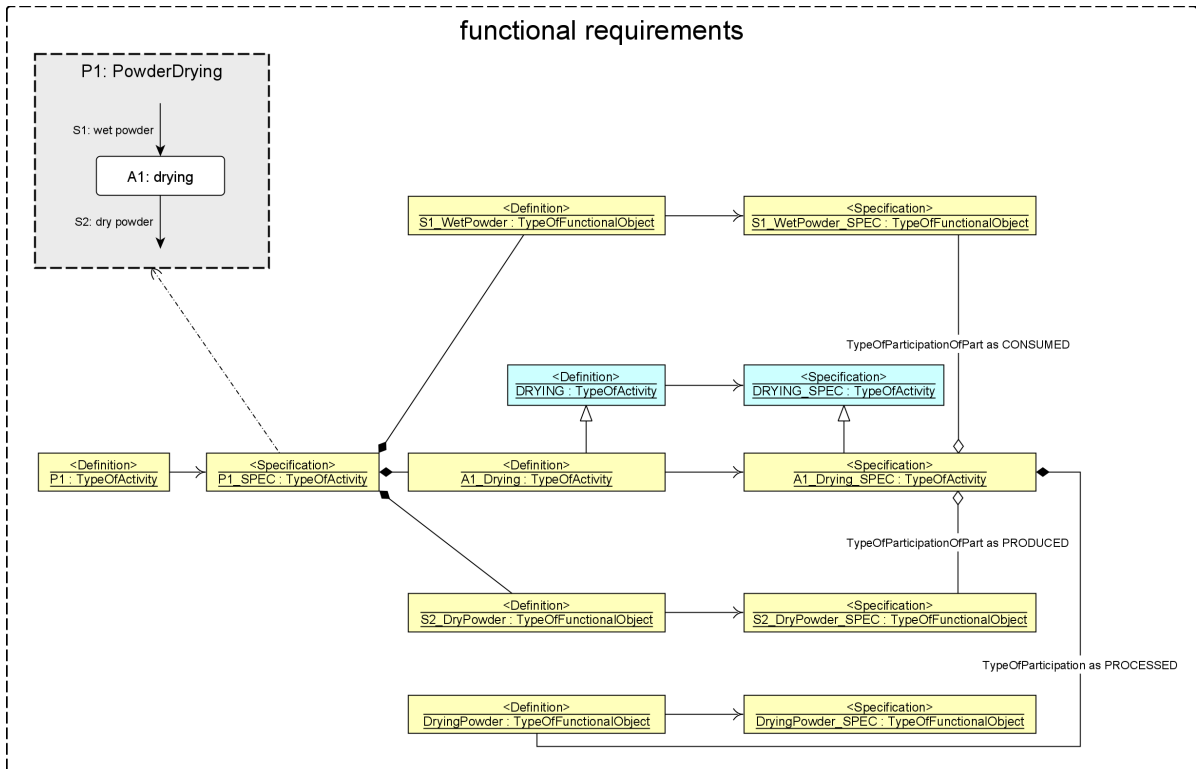


Abbildung 2.13.: Instanzbeispiel eines geplanten Prozesses zur Pulvertrocknung.

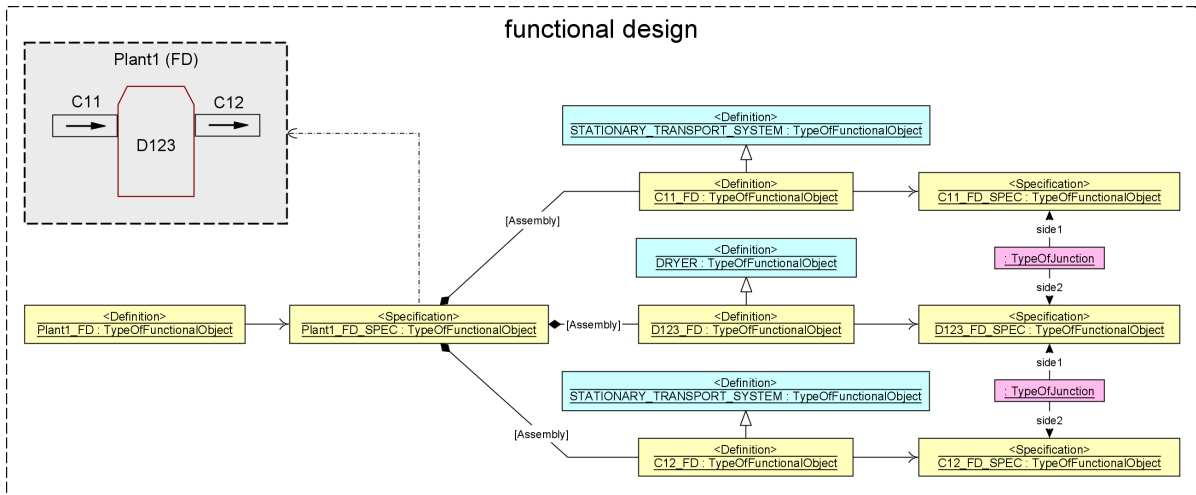


Abbildung 2.14.: Instanzbeispiel einer geplanten Anlage zur Pulvertrocknung (funktionales Design).

einem Trockner und zwei mit dem Trockner verbundenen stationären Transportsystemen besteht.

- Geplante Funktionen ([TypeOfFunctionalObject](#)) einer zu bauenden Anlage sind in der Regel Spezialisierungen von RDL-Einträgen. Die RDL-Einträge, wie z.B. *DRYER* für einen Trockner, beschreiben noch keine konkrete technische Ausführung.
- Die Beziehungen zwischen den geplanten Funktionen sind Typen von Verbindungen ([TypeOfJunction](#)).
- Die geplante Anlage ist räumlich zusammengesetzt ([TypeOfAssembly](#), eine Unterklasse von [TypeOfSpatialAggregation](#)) aus den geplanten Funktionen.

2.1.3.4. Modellierung von Querbeziehungen zwischen FR und FD

Die Funktionen einer zu bauenden Anlage werden im funktionalen Design derart geplant, dass die Komponenten einer zugehörigen realen Anlage an der Realisierung eines bestimmten Prozessschrittes beteiligt sind. Die geplanten Funktionen nehmen somit an einem geplanten Prozessschritt ([TypeOfActivity](#)) in der Rolle ([Role](#)) eines *PROCESS EQUIPMENT* teil ([TypeOfParticipationOfPart](#)).

In unserem Beispiel nimmt der geplante Trockner an dem geplanten Prozessschritt zum Trocknen teil, siehe Abbildung 2.15.

Die Modellierung erfolgt somit analog zu der von Beziehungen zwischen Prozessschritten und Zuständen (Strömen) bei den funktionalen Anforderungen des geplanten Prozesses, lediglich die Rollen sind unterschiedlich.

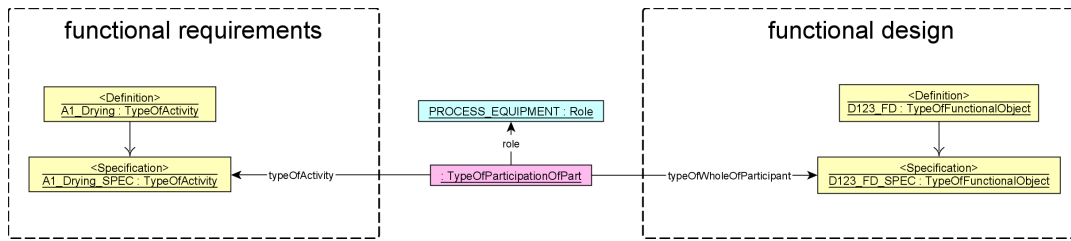


Abbildung 2.15.: Instanzbeispiel einer Querbeziehung zwischen FR und FD.

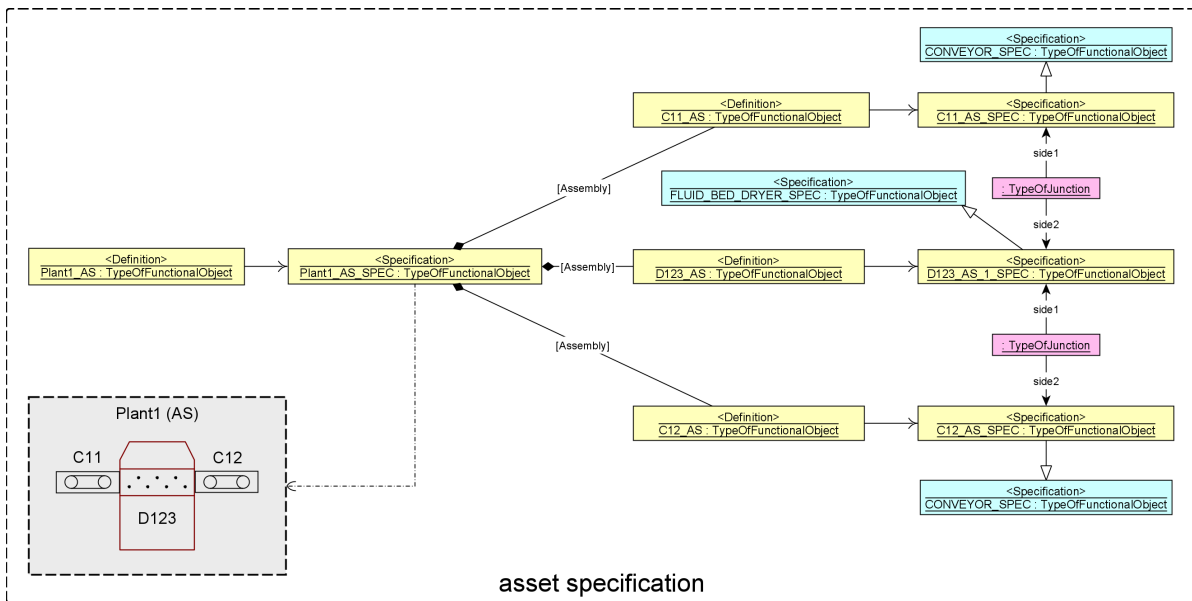


Abbildung 2.16.: Instanzbeispiel einer geplanten Anlage zur Pulvertrocknung (Spezifikation von Assets).

2.1.3.5. Modellierung einer geplanten Anlage (AS)

Bei der Spezifikation von Assets werden die geplanten Funktionen aus dem funktionalen Design spezialisiert. In unserem Beispiel werden die stationären Transportsysteme als Förderbänder und der Trockner als Flüssigbett-Trockner spezialisiert, siehe Abbildung 2.16.

2.1.3.6. Modellierung von Querbeziehungen zwischen FD und AS

Wie bereits erläutert, werden beim Übergang vom funktionalen Design zur Assetspezifikation die geplanten Funktionen und damit auch die räumliche Aggregation dieser Komponenten, d.h. die gesamte geplante Anlage, spezialisiert. Wir verzichten auf eine Abbildung dieser elementaren Assoziationen (**Specialization**).

Allerdings sei erwähnt, dass der Planungsprozess grundsätzlich sehr kreativ ist und der Übergang vom funktionalen Design zur Assetspezifikation in der Realität fließender

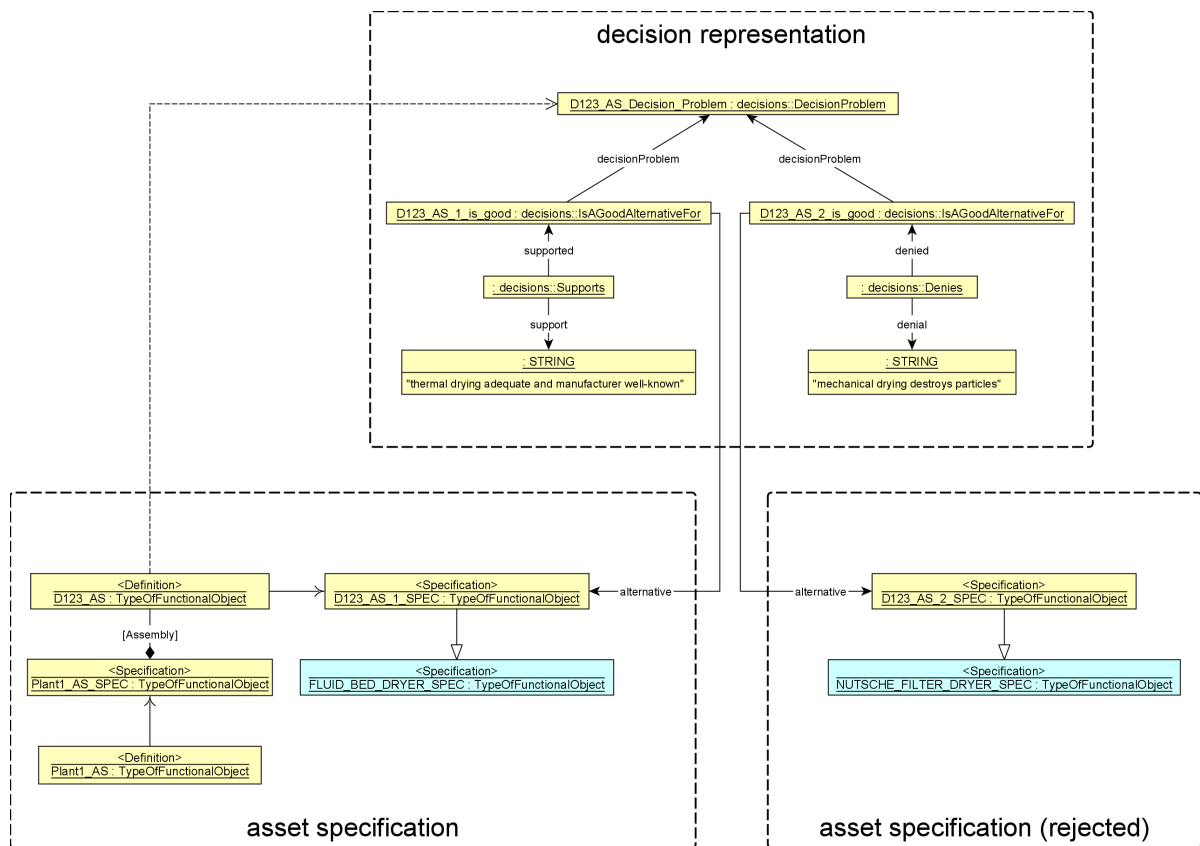


Abbildung 2.17.: Instanzbeispiel zur Bewertung von alternativen Trocknertypen für das Pulvertrocknen.

sein mag, vielleicht findet das funktionale Design mit einer grundlegenden Planung von Funktionen gar nicht statt und die Assets werden direkt spezifiziert. Oder es werden mehrere Spezifikationen einer geplanten Anlage im Sinne von zu bewertenden Alternativen erstellt.

Die Auswahl eines Equipments ist ein Entscheidungsproblem, in dem unterschiedliche Alternativen zu bewerten sind. In unserem Beispiel ist die Auswahl eines konkreten Bautyps für den Trockner ein solches Entscheidungsproblem. Der Flüssigbettrockner ist nur eine Alternative neben einem Nutsche-Filtertrockner und anderen. Mit der DRL, der Grundlage des Pakets `decisions` (s. Abschnitt 2.1.2.5), kann beispielsweise abgebildet werden, dass die Anlagenalternative mit dem Nutsche-Filtertrockner verworfen wird, da die zugrundeliegende Technologie, das mechanische Trocknen, die Partikel des Mediums zerstört. Es kann analog abgebildet werden, dass die Anlagenalternative mit dem Flüssigbettrockner ausgewählt wird, da das thermische Trocknen zerstörungsfrei wirkt und ein Hersteller von solchen Trocknern bereits bewährter Zulieferer ist, siehe Abbildung 2.17.

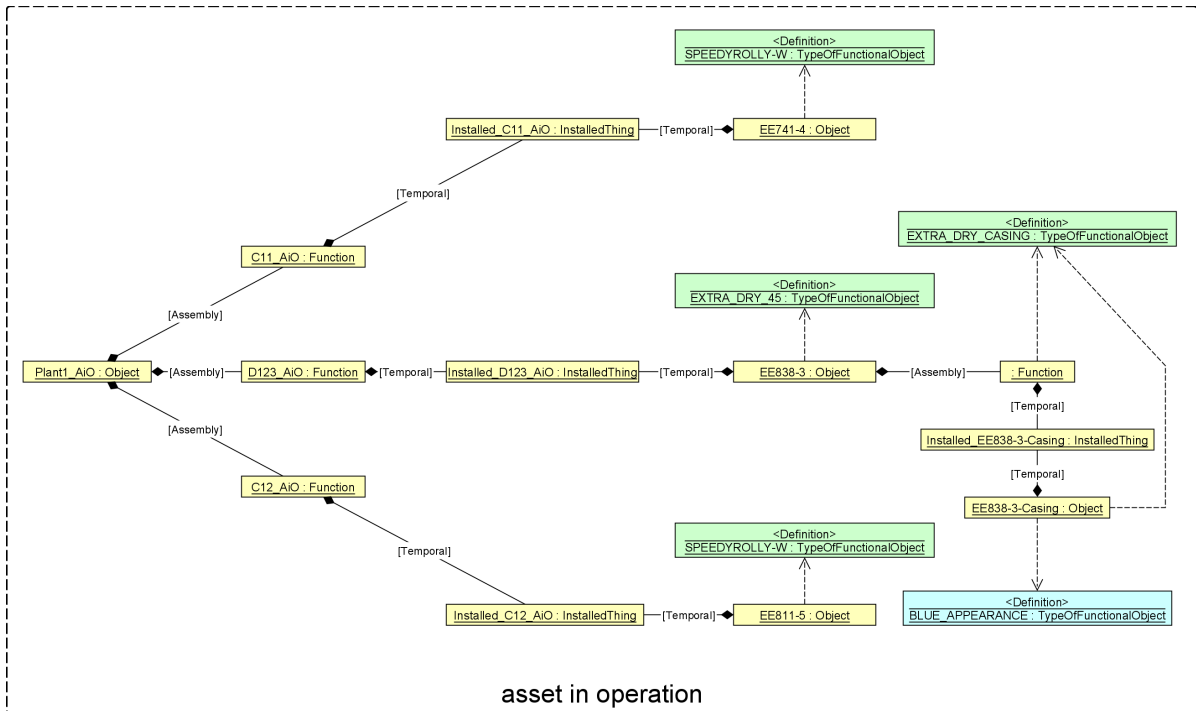


Abbildung 2.18.: Instanzbeispiel einer realen Anlage zur Pulvertrocknung.

2.1.3.7. Modellierung von Informationen im Betrieb (AiO)

Zur Modellierung von Informationen einer Anlage im Betrieb oder eines real ablaufenden Prozesses dienen Unterklassen von *ConcreteThing* und *Relationship*. Es werden konkrete Dinge zu den in der Planung festgelegten Typen und deren Beziehungen modelliert.

2.1.3.8. Modellierung einer realen Anlage

Gemäß der Planung besteht die Anlage *Plant1* aus drei Komponenten: einem Trockner (Tagname *D123*) und zwei Förderbändern (Tagnames *C11* und *C12*). Der linke Bereich der Abbildung 2.18 zeigt das entsprechende Modell:

- Die Anlage, in der Abbildung mit *Plant1_AiO* bezeichnet, ist ein Objekt (*Object*, Abschnitt 2.1.2.3.2).
- Die Angabe eines Tagname für den Trockner deutet bereits darauf hin, dass seine Identität durch seine Funktion definiert ist, nämlich innerhalb der Anlage die Trocknung durchzuführen. Der Trockner *D123* ist deshalb als *Function* (Abschnitt 2.1.2.3.2) *D123_AiO* modelliert. In gleicher Weise repräsentieren die *Function* *C11_AiO* und die *Function* *C12_AiO* die beiden Förderbänder.
- Die hier betrachtete Zerlegung der Anlage ist rein räumlicher Natur, d.h. die An-

lage besteht immer aus den genannten *funktionalen* Komponenten.³⁷ *Plant1_AiO* ist deshalb jeweils mit einer **Assembly** (Abschnitt 2.1.2.3.5), der räumlichen Spezialisierung der allgemeinen **Aggregation** (Abschnitt 2.1.2.3.5), mit den Komponenten verknüpft.

Wir haben bisher lediglich die funktionale Identität der Anlagenkomponenten betrachtet. Der mittlere Bereich von Abbildung 2.18 zeigt deren Verknüpfung mit den Dingen, die zu einer bestimmten Zeit die Funktion erfüllen, nämlich einen Trockner mit der Inventarnummer *EE838-3* und zwei Förderbänder mit den Inventarnummern *EE741-4* und *EE811-5*.

- Die Inventarnummern deuten darauf hin, dass die Identität dieser Dinge stofflich begründet ist. Sie sind deshalb als **Objects** *EE838-3*, *EE741-4* und *EE811-5* modelliert.
- **InstalledThing** dient dazu, Dinge mit funktionaler Identität (**Functions**), etwa den Trockner *D123_AiO* oder das Förderband *C11_AiO*, mit Dingen zu verknüpfen, deren Identität auf ihrer materiellen Kontinuität beruht (**Objects**), etwa mit dem Trockner vom Typ *EXTRA_DRY_45* mit der Seriennummer *EE838-3* oder mit dem Förderband vom Typ *SPEEDYROLLY-W* mit der Seriennummer *EE811-5*. Ein **InstalledThing** ist ein temporärer Teil sowohl der **Function** als auch des **Objects**. In der Raumzeit entspricht ein **InstalledThing** dem Zeitbereich, in dem die **Function** und das **Object** räumlich zusammenfallen.

In unserem Instanzbeispiel besteht der installierte Trockner vom Typ *EXTRA_DRY_45* mit der Seriennummer *EE838-3* insbesondere aus einem Gehäuse, für welches das Konzept eines installierten Dings ebenfalls Anwendung findet. Das ursprüngliche Gehäuse ist vom Typ *EXTRA_DRY_CASING* und besitzt laut Hersteller eine gelbe Farbe (vgl. dazu die Angaben des Herstellers zu einem Trockner³⁸ vom Typ *EXTRA_DRY_45* in Abbildung 2.19). Das Gehäuse wurde aber beispielsweise aufgrund von Sicherheitsvorschriften während einer Wartung umlackiert und hat aktuell eine blaue Farbe.

Wir gehen nun auf die Verbindungen zwischen installierten Dingen ein, die Details sind in Abbildung 2.20 dargestellt.

- Installierte Dinge können miteinander verbunden sein (**Connection**) oder sie sind voneinander getrennt (**Separation**). In unserem Instanzbeispiel sind die beiden installierten Dinge *Installed_C11_AiO* und *Installed_D123_AiO* in einem bestimmten Zeitintervall miteinander über eine direkte Verbindung (**Junction** ist eine spezielle Art einer allgemeinen **Connection**) verbunden. Analog kann abgebildet werden, dass die beiden installierten Dinge in einem bestimmten Zeitintervall, z.B. aus Wartungsgründen, voneinander getrennt sind.

³⁷Dies bedeutet nicht, dass zu jedem Zeitpunkt *materialisierte* Trockner und Förderbänder in der Anlage vorhanden sind.

³⁸Es sei an dieser Stelle darauf hingewiesen, dass die **Objects**, die in einer Anlage installiert werden, einen eigenen *Asset Life Cycle* besitzen, der mit den selben Konzepten beschrieben werden kann.

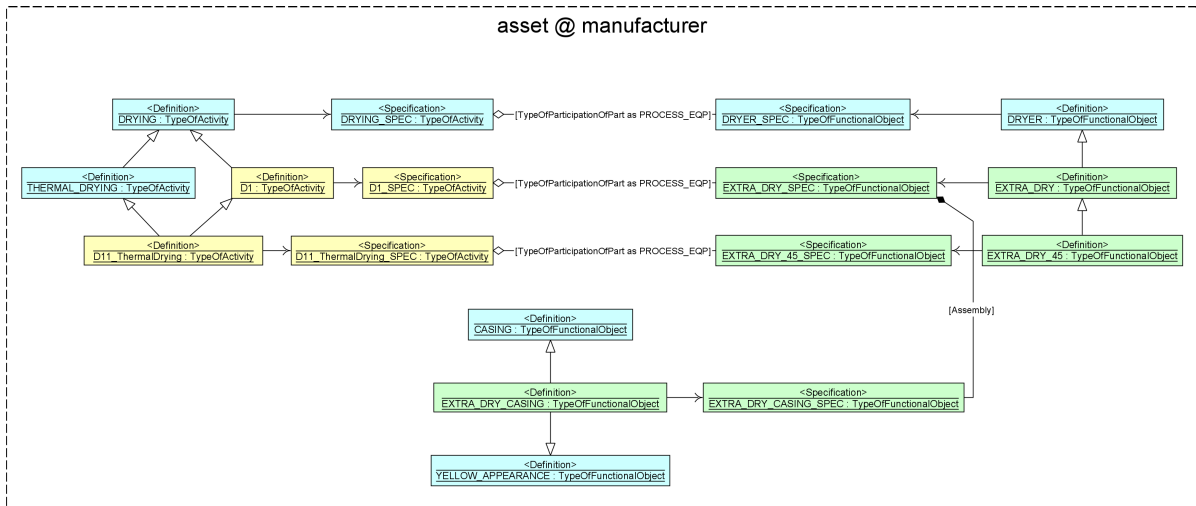


Abbildung 2.19.: Instanzbeispiel von Herstellerangaben zu einem Trockner.

- Für die **Junctions** ist eine im Vergleich zu den **TypeOfJunctions** detailliertere Darstellungsform erforderlich. Da Verbindungselemente, wie z.B. die Stutzen einer Pumpe, in der Regel keine eigenständige Identifizierung besitzen, werden sie als untrennbarer Teil (**Feature** ist eine Spezialisierung der **Aggregation**) eines konkreten Dings modelliert.

In unserem Instanzbeispiel in Abbildung 2.20 wurde modelliert, dass eine Wartungsaktivität zwei Ereignisse (**Event**) verursacht. Ein Trennungseignis findet zum Zeitpunkt 2021-08-21T02:32:45 statt, die Verbindung zwischen den installierten Dingen *Installed_C11_AiO* und *Installed_D123_AiO* wird getrennt. Das bedeutet, dass die verbundenen installierten Teile zeitlich enden (**Ending**), während die getrennten installierten Dinge zeitlich beginnen (**Beginning**). Analog wurde ein erneutes Verbindungsereignis nach Abschluss der Wartungsarbeiten zum Zeitpunkt 2021-08-22T09:45:20 modelliert.

2.1.3.9. Modellierung eines realen Prozesses

Siehe dazu das Instanzbeispiel in Abbildung 2.21.

- Ein realer Prozess besteht aus Aktivitäten (**Activity**), wie beispielsweise der Aktivität *P1_Drying_ForOrder4711*, die für eine bestimmte Bestellung durchgeführt wird.
- Eine Aktivität ist eine Aggregation (**Aggregation**) von Teilaktivitäten und von materialisierten Dingen mit einer kontinuierlichen Funktion (**Function**). In unserem Instanzbeispiel sind es eine Aktivität zum Trocknen und die Ströme aus feuchtem, gerade trocknendem und bereits getrocknetem Pulver, die zu der zuvor genannten Bestellung gehören. Die materialisierten Dinge mit kontinuierlicher Funktion nehmen an der Aktivität (teilweise oder ganz, **Participation** oder **ParticipationOfPart**)

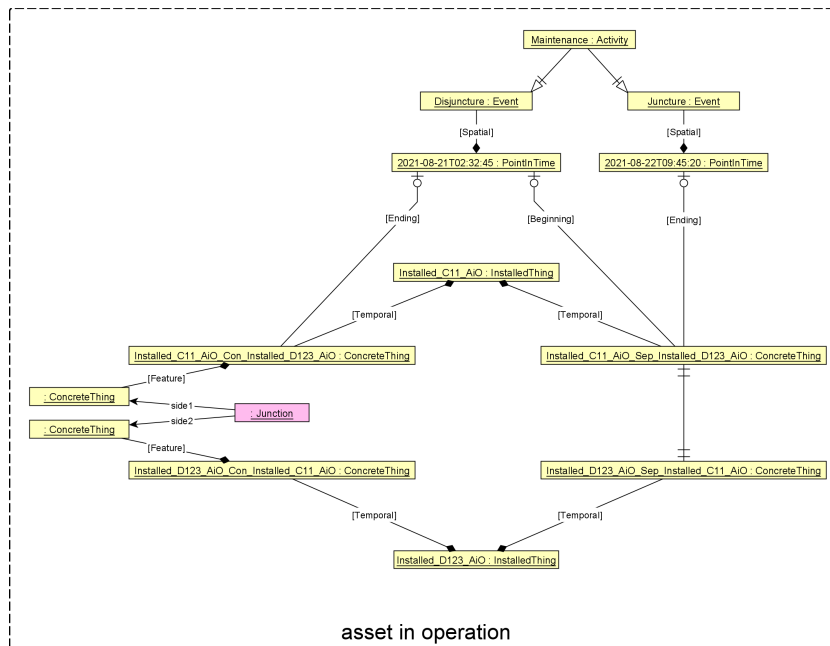


Abbildung 2.20.: Instanzbeispiel für verbundene und getrennte installierte Dinge.

in einer bestimmten Rolle (**Role**) teil. In unserem Beispiel sind es die Rollen eines Edukt-, Prozess- und Produktstroms.

- Die Edukt-, Prozess-, und Produktströme eines Gesamtprozesses werden zur Realisierung einzelner Aktivitäten des Prozesses benötigt. Sie werden mit dem bereits beschriebenen Konzept eines installierten Dings (**InstalledThing**) mit einzelnen Aktivitäten des Prozesses verbunden. In unserem Beispiel sind temporäre Teile der Edukt-, Prozess-, und Produktströme des Gesamtprozesses installiert, um die Aktivität *A1_Drying_ForOrder4711*, die für eine bestimmte Bestellung durchgeführt wird, zu realisieren.

2.1.3.10. Modellierung von Querbeziehungen zwischen und innerhalb Planung und Betrieb

Insgesamt lassen sich die folgenden Querbeziehungen zwischen und innerhalb der Planung und dem Betrieb eines Prozesses bzw. einer Anlage formulieren, siehe dazu auch das Instanzbeispiel in Abbildung 2.22.

- Während der Planung eines Prozesses bzw. einer Anlage werden Typen von konkreten Dingen und deren Beziehungen festgelegt. Dies sind beispielsweise der Typ einer Aktivität (**TypeOfActivity**) zum Trocknen eines Mediums oder die geplante Funktion eines Trockners, d.h. der Typ eines funktionalen Objektes (**TypeOfFunctionalObject**).

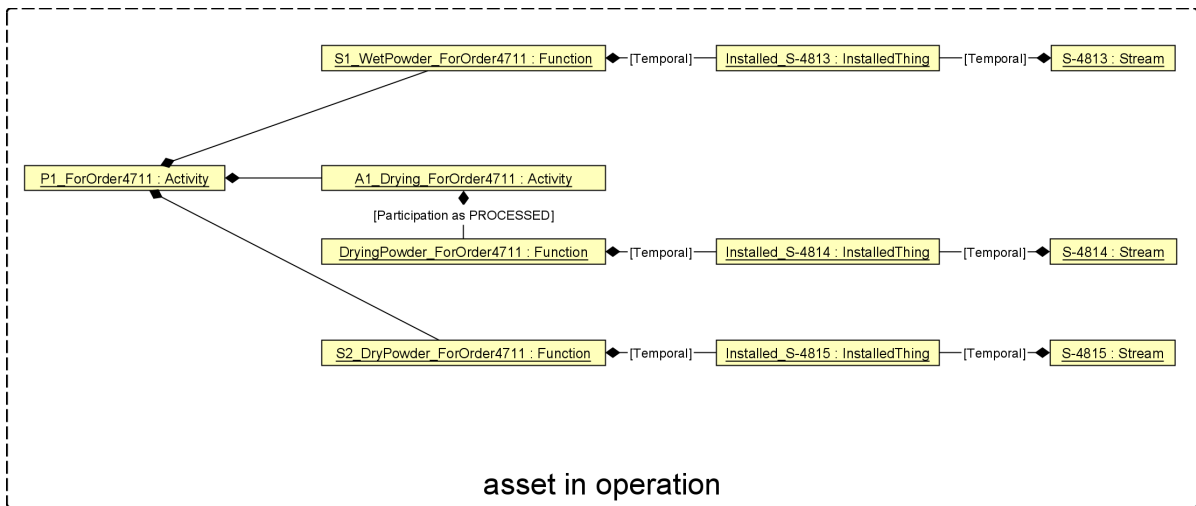


Abbildung 2.21.: Instanzbeispiel eines realen Prozesses zur Pulvertrocknung.

- Eine geplante Anlagenkomponente soll dabei zur Realisierung einer geplanten Prozessaktivität beitragen (**TypeOfParticipationOfPart**) und zwar in einer bestimmten Rolle (**Role**).
- Während der Planung eines Prozesses bzw. einer Anlage wird der Design-Kontext unterschieden. Neben allgemeingültigen Typeninformationen, die im **DesignContext Definition** beschrieben sind, werden spezifische Typeninformationen im Kontext **Specification** angegeben.
- Im Betrieb eines Prozesses oder einer Anlage sind konkrete Dinge und deren Beziehungen zu den in der Planung festgelegten Typen abzubilden. Die konkreten Dinge und deren Beziehungen sollen also die zuvor festgelegten Typen haben (**Typification**).
- Dinge mit funktionaler Identität (**Functions**) werden durch installierte Dinge (**InstalledThings**) mit Dingen verknüpft, deren Identität auf ihrer materiellen Kontinuität beruht (**Objects**).
- Eine Anlagenkomponente im Betrieb, wie z.B. eine Funktion, ist schließlich an der Realisierung einer bestimmten Prozessaktivität beteiligt (**ParticipationOfPart**), und zwar in der geplanten Rolle (**Role**).

2.1.4. Anwendungsbeispiel: Verfahrensentwicklung und Prozessoptimierung

Das erste Szenario wurde mit den assoziierten Partnern des Projektes *ModuLA* (Schwerpunkt der *Clariant*) unter Einbindung von Ergebnissen aus dem Projekt *ENPRO 2.0 SkaMPi* ausgearbeitet.

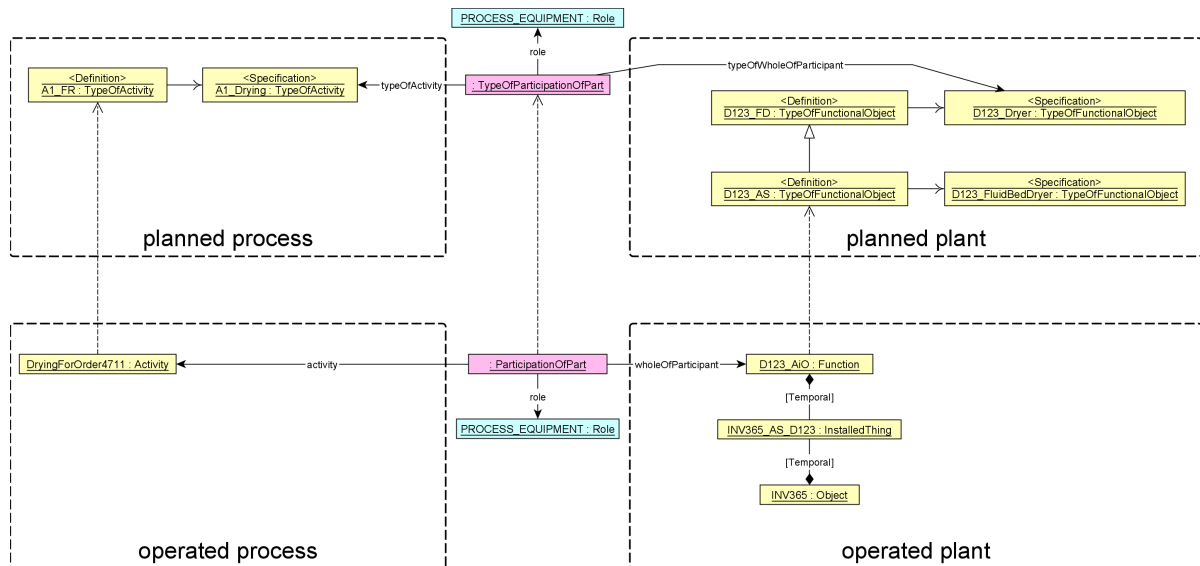


Abbildung 2.22.: Instanzbeispiel der Querbeziehungen zwischen Planung und Betrieb.

2.1.4.1. Herausforderungen

Forschung und Verfahrensentwicklung, Betrieb und externe Partner befinden sich oft an unterschiedlichen Standorten. Sie müssen die Arbeiten untereinander koordinieren und benötigen zunächst Zugriff auf vorhandene Lebenszyklus-Informationen, zu einem bestehenden Prozess und einer Bestandsanlage. Dies sind z.B. Dokumente zur Verfahrensentwicklung, aus dem Labor und dem Pilotbetrieb, Modelle für die Apparateauslegung oder für die Automatisierung sowie Betriebsdaten. Veränderungen am Prozess oder der Bestandsanlage und die Entscheidungsgrundlagen dafür müssen nachvollziehbar dokumentiert werden.

Eine wesentliche Herausforderung ist somit die Verknüpfung von Informationen über verschiedene Grenzen hinweg. Diese Grenzen sind einerseits die Phasen im Lebenszyklus von der Planung einer (modularen) Anlage bis hin zur Optimierung der Anlage im Betrieb, andererseits die Skalierung vom Labor über das Technikum bis hin zum Weltmaßstab. Erworbenes Wissen und Erfahrungen bei der Verfahrensentwicklung müssen archiviert und für künftige Entwicklungsaufgaben verfügbar gemacht werden.

2.1.4.2. Ausgangspunkt

In den folgenden Abschnitten stellen wir das Szenario und die zugehörigen Software-Prototypen vor. Technische Details zur Modellierung sind in Abschnitt 2.1.4.5 zusammengefasst.

In einer existierenden Flüssig-Flüssig-Extraktion soll das Lösemittel ausgetauscht werden, da es die vorgeschriebenen Ökotox-Kriterien nicht erfüllt. Dies ist ein Beispiel, das die Anpassung bzw. Optimierung eines existierenden Prozesses erfordert. Es gibt vielfältige andere Motivationen, wie z.B. die Verbesserung der Energieeffizienz durch Nutzung

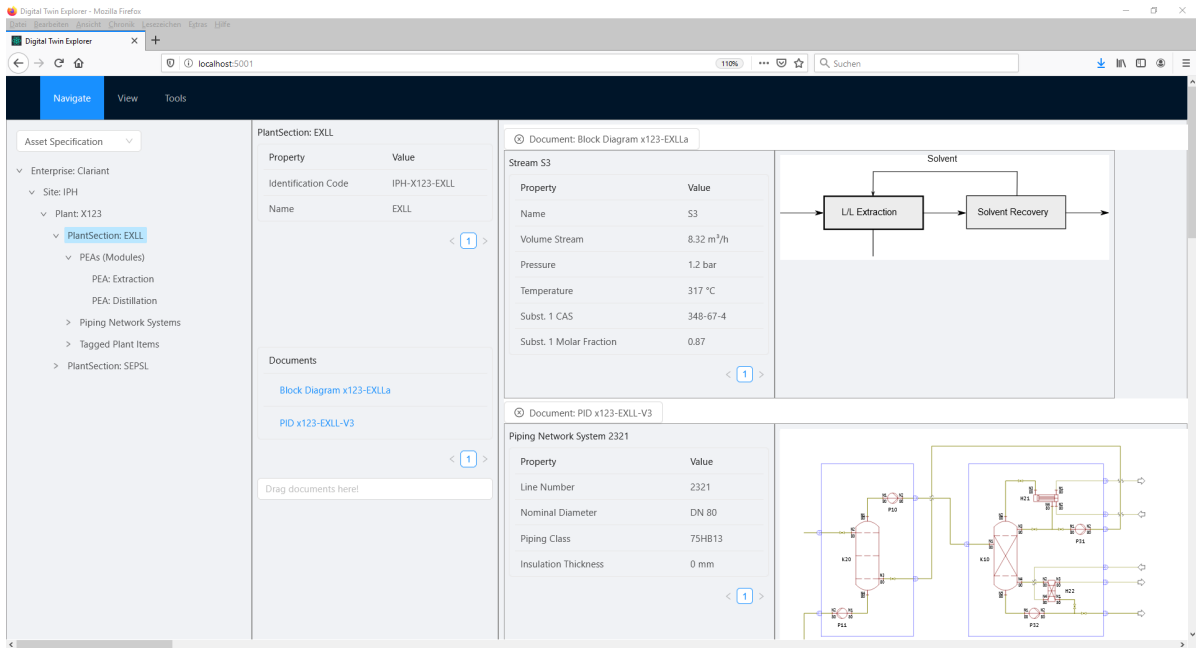


Abbildung 2.23.: Browsen von Lebenszyklus-Informationen im *Digital Twin Explorer*.

innovativer Apparatetechnologien oder geänderte Vorschriften zur Abwasserbehandlung bzw. Regulierungen im Allgemeinen. Insgesamt wird deutlich, dass ein Anlagenbetreiber in die Lage versetzt werden muss, auf veränderte Anforderungen zeitnah reagieren zu können.

Folgende Aktionen müssen zum Wechsel des Lösemittels durchgeführt werden: die Auswahl eines neuen Lösemittels, die Anpassung des geplanten Prozesses (d.h. der Simulation) und schließlich die Anpassung der geplanten Anlage (durch Apparateauswahl und Anpassung des zugehörigen R&Is). Im Detail sind dazu die folgenden Schritte und Informationen notwendig.

2.1.4.3. Analyse

1. Zunächst müssen Informationen zum existierenden Prozess gesichtet und bereitgestellt werden.

Der *Digital Twin Explorer* (s. Abbildung 2.23) bietet genau diese Funktionalität: Lebenszyklus-Informationen, die der Software-Prototyp *Digital Twin Integrator* über eine einheitliche Schnittstelle bereitstellt³⁹, können vom Nutzer im *Digital Twin Explorer* abgerufen und gepflegt werden.⁴⁰ Dies beinhaltet z.B. In-

³⁹Der *Digital Twin Integrator* ist realisiert als RDF Triple Store mit einer SPARQL-Schnittstelle. Das genutzte RDF-Vokabular entspricht exakt dem Vokabular des Integrationsmodells gemäß Abschnitt 2.1.2.

⁴⁰Im Folgenden beziehen wir uns im Interesse besserer Lesbarkeit bei der Beschreibung der Funktionalität einheitlich auf den *Digital Twin Explorer*, auch wenn im Hintergrund der *Digital Twin Integrator*

formationen zu einem Unternehmen, zu verschiedenen Standorten, den zugehörigen Anlagen und den Anlagenteilen bis hin zu den modularen Prozesseinheiten (PEAs) eines Anlagenteils. Im Szenario besteht der Anlagenteil, der die Flüssig-Flüssig-Extraktion realisiert, aus zwei PEAs, nämlich einer Extraktions- und einer Destillations-PEA. Zu beiden PEAs sind die zugehörigen Apparatetechnologien hinterlegt, die Extraktions-PEA ist im existierenden Prozess mit einer Siebbodenkolonne realisiert.

Zum Anlagenteil sind weitere Informationen verfügbar. Das Verfahrensfließbild enthält die beiden Unit Operations *LL-Extraction* und *Solvent Recovery* sowie den verbindenden Lösemittelstrom. Zu diesem Strom können Stoff-, Simulations- und Betriebsdaten abgefragt werden, wie z.B. der Name des zugehörigen Stroms in der Simulation, die CAS-Nummer des Lösemittels oder die Temperatur des Stoffstroms im Betrieb. Außerdem ist das zugehörige R&I des Anlagenteils verfügbar. Da die Informationen aus beiden Dokumenten miteinander verknüpft (integriert) sind, kann zu dem Lösemittelstrom im Verfahrensfließbild auch direkt die zugehörige Rohrleitung im R&I identifiziert werden.⁴¹ Zur Rohrleitung sind weitere Informationen verfügbar, wie z.B. der Nenndurchmesser und die Materialklasse.

2. Nun wird ein neues Lösemittel ausgewählt, das die Ökotox- und weitere Kriterien erfüllt (beispielsweise muss sichergestellt werden, dass das Material der Rohrleitungen für das neue Lösemittel geeignet ist). Der Verteilungskoeffizient zu diesem neuen Lösemittel ist im Szenario nicht bekannt, wird aber für die Entscheidung, ob die momentan betriebene Apparatetechnologie (Siebbodenkolonne) noch geeignet ist, benötigt. Deshalb werden die entsprechenden Experimente durchgeführt und mit einem üblichen Labordaten-Werkzeug erfasst. Insbesondere werden Messdaten in einer Excel-Tabelle dokumentiert.

Zur Demonstration wird hier das freie Werkzeug *eLabFTW*⁴² genutzt. Abbildung 2.24 zeigt links eine Liste der in *eLabFTW* hinterlegten Experimente und rechts Details zum Experiment *Distribution Coefficient X123-EXLL*, insbesondere eine Download-Möglichkeit für die Excel-Tabelle *data.xlsx*.

Im *Digital Twin Explorer* werden die Daten aus dem Laborexperiment direkt als Information zu dem Anlagenteil hinterlegt, der die Flüssig-Flüssig-Extraktion realisiert; dazu kann das Experiment *Distribution Coefficient X123-EXLL* aus der Liste in Abbildung 2.24 direkt in den Bereich *Documents* im *Digital Twin Explorer* geschoben werden. Neue Informationen lassen sich so leicht integrieren, so dass sie für die Zukunft verfügbar und systematisch abrufbar werden.⁴³

einen wesentlichen Anteil daran hat.

⁴¹Die Verknüpfung zwischen Lösemittelstrom und Rohrleitung wird im *Digital Twin Explorer* durch Blinken beider Elemente visualisiert und ist deshalb in Abbildung 2.23 nicht zu erkennen.

⁴²<https://www.elabftw.net>

⁴³Im Rahmen des Forschungsprojekts *ModuLA* war keine vollständige Integration mit *eLabFTW* möglich. Es werden nur die Excel-Dateien als Domäneninformationen mit den **Objects** verknüpft, wie in **Anmerkung 1 zu Modell *Scenario-1-Experiments-to-Real-World*** beschrieben, nicht aber die Datensätze aus *eLabFTW*.

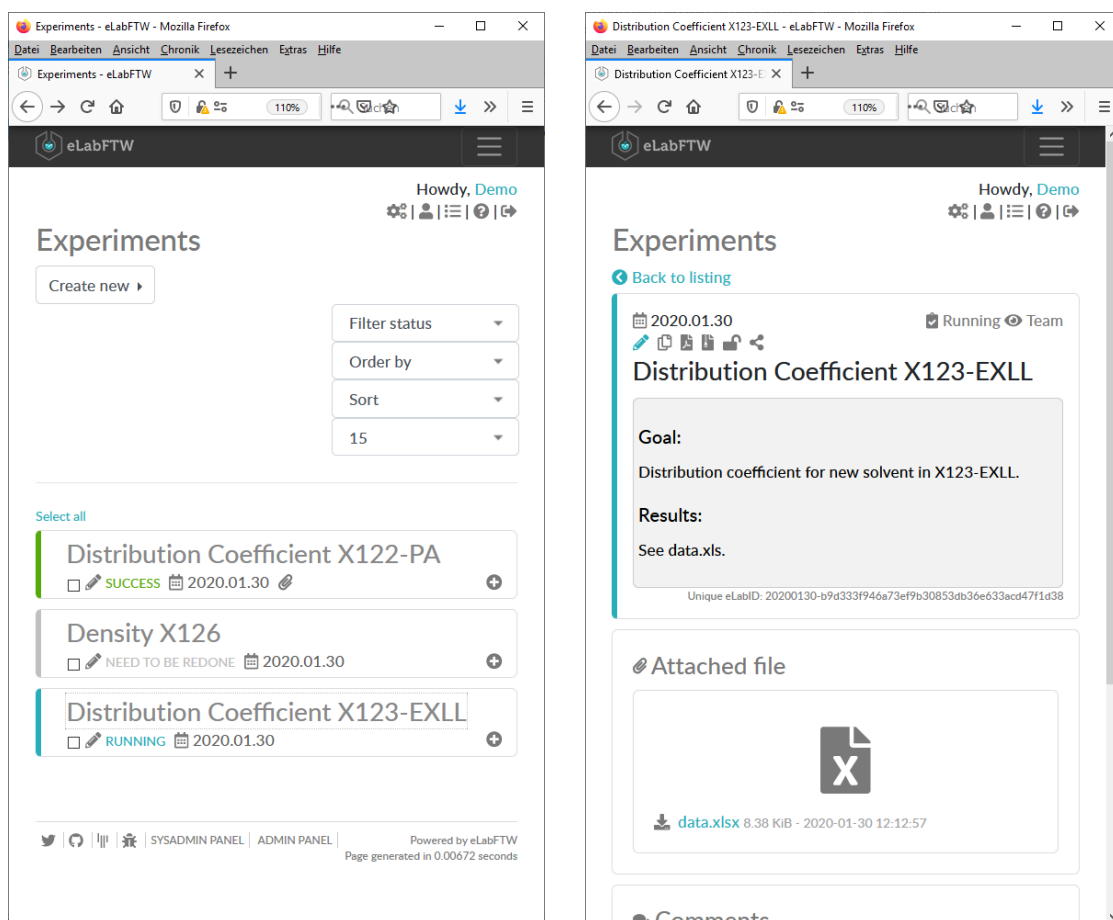


Abbildung 2.24.: Verwaltung von Labordaten in *eLabFTW*.

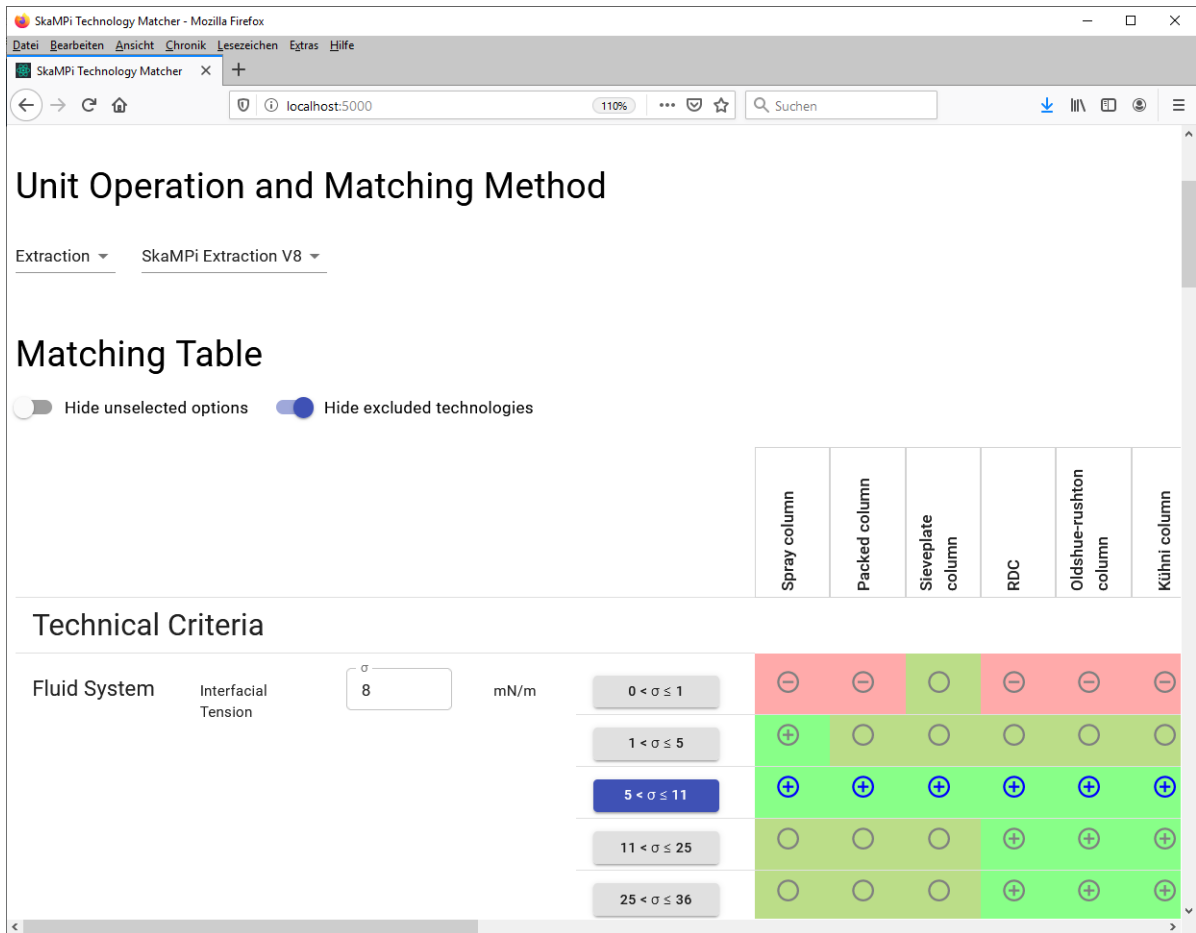


Abbildung 2.25.: Bewertung von Extraktionstechnologien im *SkaMPi Technology Matcher*.

- Mithilfe der bekannten Stoffdaten etc. wird die Simulation des Teilprozesses erneut durchgeführt. Eine Integration der Simulation kann ähnlich wie bei den Labordaten erfolgen. Dieser Schritt wird im Software-Prototypen zur Zeit nicht unterstützt.
- Im nächsten Schritt wird überprüft, ob die aktuell verwendete Apparate-technologie noch in Kombination mit dem neuen Lösemittel betrieben werden kann und ob evtl. bereits bessere (innovativere) Technologien, z.B. unter dem Aspekt der Energieeffizienz, zur Verfügung stehen.

Dazu kann die im Projekt *ENPRO 2.0 SkaMPi* entwickelte Methode angewendet werden, die auf Grundlage einiger Kenngrößen wie z.B. der Oberflächenspannung und der Dichtedifferenz des neuen Lösemittels (Eigenschaften des Fluidsystems) oder des Volumenstroms (Prozessparameter) eine Bewertung von Technologien ermöglicht. Im Szenario ergibt sich, dass die aktuell genutzte Apparate-technologie nicht mehr adäquat ist und die Packungskolonnen am besten geeignet ist, da sie mit vergleichsweise geringem Aufwand aus der aktuell genutzten Technologie durch

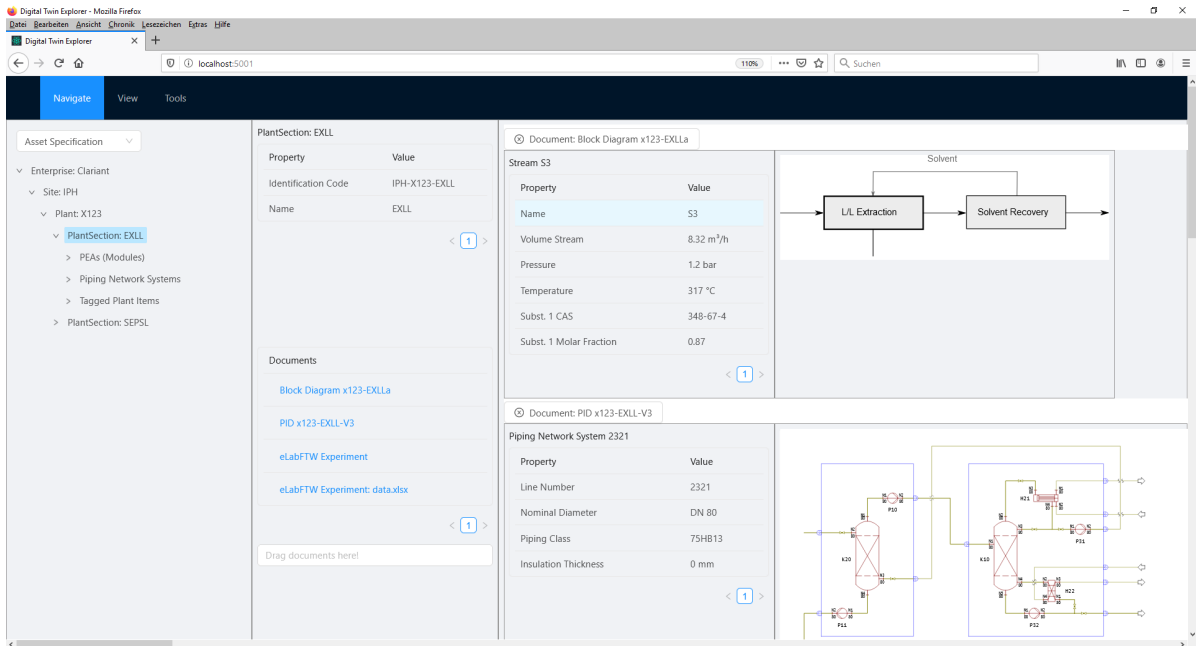


Abbildung 2.26.: Aktualisiertes R&I mit Packungskolonnen-PEA visualisiert im *Digital Twin Explorer*.

Umbau realisiert werden kann.

Im Projekt *ENPRO 2.0 SkaMPi* hat *AixCAPE* die beschriebene Methode im *SkaMPi Technology Matcher* (Grünwald, 2021; s. Abbildung 2.25) implementiert. Der *SkaMPi Technology Matcher* wurde bewusst als eigenständiges Werkzeug unabhängig vom *Digital Twin Explorer* realisiert und *a posteriori* mit ihm integriert. Stoff- und Prozessgrößen, die für die Technologiebewertung relevant sind, können automatisiert aus dem *Digital Twin Explorer* in den *SkaMPi Technology Matcher* übernommen werden.

2.1.4.4. Änderungsdokumentation und Vorbereitung der physikalischen Anpassungen

5. Nach Auswahl einer neuen Extraktions-PEA, z.B. aus einem Hersteller-Katalog, wird das zugehörige R&I aktualisiert.

Im vorliegenden Fall (keine komplexen strukturellen Änderungen) kann das R&I im *Digital Twin Explorer* automatisiert aktualisiert werden (s. Abbildung 2.26). In anderen Fällen würde das bisherige R&I als *veraltet* markiert werden; die eigentliche Bearbeitung würde in einem üblichen CAE-Werkzeug erfolgen.

6. Zu einem späteren Zeitpunkt wird die ganze Historie der Teilanlage für die Flüssig-Flüssig-Extraktion abgefragt, inklusive aller Änderungen.

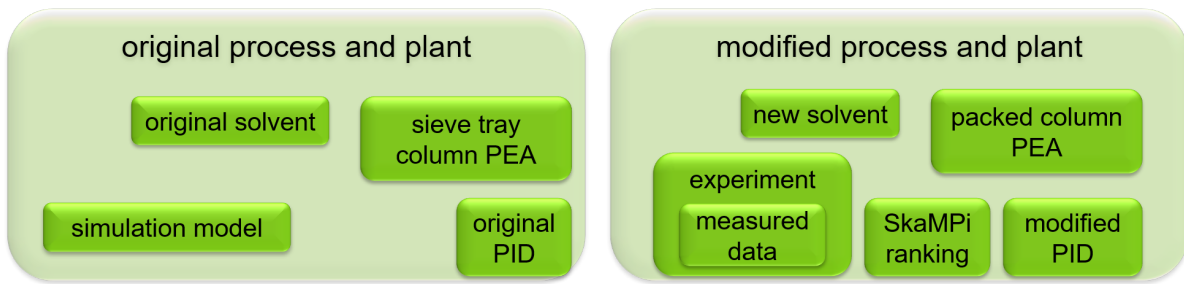


Abbildung 2.27.: Historische und aktuelle Informationen im *Digital Twin Explorer*.

Die Informationen im *Digital Twin Explorer* umfassen nicht nur den aktuellen Zustand der Teilanlage wie z.B. die Extraktions-PEA mit der Packungskolonne. Auch die Information, dass diese PEA ursprünglich mit einer Siebbodenkolonne realisiert war und im Rahmen neuer Anforderungen an das Lösemittel basierend auf der *SkaMPi Methodik* ersetzt wurde, ist verfügbar. Abbildung 2.27 zeigt schematisch die insgesamt verfügbaren Informationen.

Das beschriebene Szenario wurde auf der PAAT 2019 vorgestellt. Das dort gezeigte Demonstrationsvideo ist online verfügbar.⁴⁴

Mit den Projektpartnern sowie mit Teilnehmern der PAAT 2019 wurden weitere Ideen zum Szenario diskutiert:

- Bei der Auswahl des neuen Lösemittels ist eine Übersicht über Projekte eines Unternehmens hilfreich, in denen ein Lösemittel bereits ausgetauscht wurde. So lassen sich z.B. wiederholte Experimente an verschiedenen Standorten vermeiden und Erfahrungen bzw. bereits vorhandenes Wissen standortübergreifend verfügbar machen.
- Die erforderlichen Informationen, um ein Experiment in Auftrag zu geben, lassen sich automatisiert zusammenstellen. Die Übergabe von Informationen aus der Planung an das Labor und zurück kann automatisiert werden.

Die tatsächliche Realisierung dieser Ideen im Prototypen hätte den Rahmen des Projektes gesprengt. Wir gehen aber davon aus, dass sich dabei keine weiteren Anforderungen an das Informationsmodell ergeben hätten.

2.1.4.5. Modellierung

Das Modell *Scenario-1-Experiments-to-Real-World*⁴⁵ (s.a. Abbildung A.2) zeigt die Abbildung einzelner Aspekte des Szenarios mit dem Integrationsmodell.

⁴⁴<https://downloads.aixcape.org/PAAT2019.mp4>

⁴⁵<https://modula.aixcape.org/Scenario-1-Experiments-to-Real-World.svg>

Anmerkung 1 zu Modell *Scenario-1-Experiments-to-Real-World*

↪ <https://modula.aixcape.org/Scenario-1-Experiments-to-Real-World.svg#note1>

Der Modellausschnitt zeigt die Festlegung des Wertes von $0,77 \text{ g/J K m}^2$ für die spezifische Trocknungsrate des Apparatetyps *MiniExtraDry6* einschließlich der dafür durchgeführten Laborexperimente und Berechnungen.

Der **TypeOfFunctionalObject** *MiniExtraDry6* mit **DesignContext** Definition repräsentiert den Apparatetyp, der über eine **Specification**-Beziehung verknüpfte **TypeOfFunctionalObject** *MiniExtraDry6* mit **DesignContext** Specification die erwarteten oder gewünschten Eigenschaften des Apparatetyps. Dies schließt die Eigenschaft ein, eine spezifische Trocknungsrate von $0,77 \text{ g/J K m}^2$ zu haben, was durch **Specialization** von einer entsprechend definierten **Property** abgebildet wird.

Die Beziehung zu den Laborexperimenten und Berechnungen wird dadurch modelliert, dass der **TypeOfFunctionalObject** *MiniExtraDry6* mit **DesignContext** Definition als ein **DecisionProblem** typisiert wird, für den die eingangs genannte **Specification** eine gute Alternative darstellt (**IsAGoodAlternativeFor**). Dies wird durch die *CalculationInLab* gestützt (**Supports**), d.h. eine Information (**TypeOfInformationRepresentation**), die in einer Excel-Datei abgelegt ist, auf die hier in einem einfachen Domänenmodell verwiesen wird. Diese **Supports**-Beziehung wird ihrerseits durch die gemessenen Daten aus den Experimenten gestützt (**Supports**).

Anmerkung 2 zu Modell *Scenario-1-Experiments-to-Real-World*

↪ <https://modula.aixcape.org/Scenario-1-Experiments-to-Real-World.svg#note2>

Die gemessenen Daten sind indirekt durch einen Verweis auf einen Tag in einem PIM-System abgebildet. Der Tag ist eine **Representation** des Stroms (**Stream**) *DriedProductInLabExp1*, der während der Durchführung des Laborexperiments existierte.

Anmerkung 3 zu Modell *Scenario-1-Experiments-to-Real-World*

↪ <https://modula.aixcape.org/Scenario-1-Experiments-to-Real-World.svg#note3>

Im Szenario wird davon ausgegangen, dass mittels der Laborexperimente und einer Berechnung die spezifische Trocknungsrate $0,77 \text{ g/J K m}^2$ für den Laborapparat vom Typ *MiniExtraDry6* bestimmt wird. In einer weiteren Berechnung auf Grundlage einer Ähnlichkeitsanalyse kann daraus die spezifische Trocknungsrate $0,6 \text{ g/J K m}^2$ für den größeren *MaxiExtraDry3* bestimmt werden.

Die Modellierung dieses Wertes der Trocknungsrate und seiner Begründung erfolgt weitgehend analog zur Modellierung der Trocknungsrate für den Laborapparat. Der einzige Unterschied besteht darin, dass die Bewertung dieser Trocknungsrate als *gut* (**IsAGoodAlternativeFor**) nicht direkt durch die Laborexperimente gestützt wird, sondern durch die Aussage, dass die Trocknungsrate für den Laborapparat *gut* ist. Insgesamt ist aber die Argumentationskette von den Laborexperimenten über die beiden Berechnungen zur spezifischen Trocknungsrate für den *MaxiExtraDry3* abgebildet und kann nachvollzogen werden, zumal die externen Informationen (Excel-Dateien, PIMS-Tag) verlinkt

sind. Umgekehrt können die „Folgefehler“ automatisch identifiziert werden, wenn beispielsweise ein Fehler in einer der externen Informationseinheiten festgestellt wird, z.B. eine falsche Formel in einer Excel-Datei.

2.1.5. Anwendungsbeispiel: Modulare Anlagenplanung und Modul-Automatisierung

Das zweite Szenario wurde mit den assoziierten Partnern des Projektes *ModuLA* (Schwerpunkt der *Evonik*) in Abstimmung mit dem Projekt *ENPRO 2.0 ORCA* ausgearbeitet.

2.1.5.1. Herausforderungen

Für Planung, Bau und Betrieb einer Anlage mit Modulen, sogenannten *Process Equipment Assemblies* (PEAs) und deren funktionalen Komponenten, den sogenannten *Functional Equipment Assemblies* (FEAs), wurden folgende informationstechnische Herausforderungen und Ziele identifiziert:

1. Relevante PEA-Daten (Simulation, R&I, Services, Bedienbild (HMI), Abmessungen, Anschlusspunkte, etc.) werden in unterschiedlichen Standards beschrieben, wie z.B. dem *DEXPI*-Standard für R&Is oder dem *MTP* (Module Type Package) für automatisierungstechnische Aspekte. Die Querverbindungen sind häufig nicht verfügbar und schwer ersichtlich, wie z.B. $m : n$ -Verknüpfungen von Elementen eines R&I mit Elementen eines HMI (siehe dazu das Beispiel in Abbildung 2.28, in dem eine Struktur für Rohrleitungen und Instrumentierungsinformationen im R&I auf eine im Vergleich weniger detaillierte Struktur zur Visualisierung im HMI abgebildet ist). Künftig sollen diese Querverbindungen verfügbar gemacht werden.
2. Bislang werden Änderungen an PEAs oder modularen Anlagen im Betrieb, z.B. Revisionen von PEAs sowie Kapazitäts-, Produkt- oder Funktionsänderungen, in der Regel nicht mit den Planungsdokumenten abgeglichen. Mithilfe von explizit verfügbaren Querverbindungen können Konsistenzprüfungen von Informationen der verschiedenen Aspekte im *Asset Life Cycle* künftig automatisiert durchgeführt werden, z.B. als Grundlage für ein Änderungsmanagement.
3. Lassen sich konkrete Regeln für Querverbindungen zwischen verschiedenen Aspekten des *Asset Life Cycles* angeben, so ist auch die initiale Erstellung oder Aktualisierung von Informationen des einen Aspekts mithilfe der (geänderten) Informationen des anderen Aspekts automatisierbar. Beispielsweise kann ein *MTP*-Grundgerüst regelbasiert aus Informationen eines zugrundeliegenden *DEXPI*-R&Is erstellt werden.
4. Bei der Übergabe von Informationen, z.B. aus dem Engineering bei einem PEA-Hersteller an das Anlagen-Engineering bei einem EPC oder OO, sind Konzepte zur Integration von separaten Lebenszyklen erforderlich (Integration des PEA-Lebenszyklus mit dem Lebenszyklus der Anlage, in die die PEA zeitweise eingebaut wird).

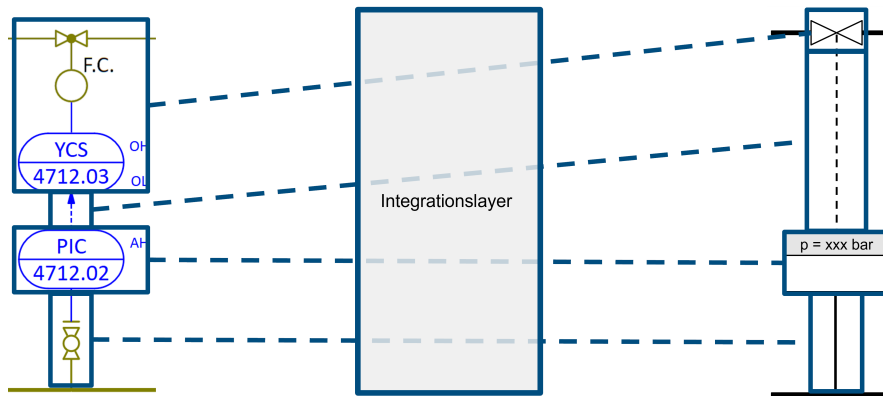


Abbildung 2.28.: Unterschiedlicher Detaillierungsgrad und strukturelle Unterschiede zwischen einem *DEXPI*-R&I und einem *MTP*-HMI.

2.1.5.2. Ausgangspunkt

Im Szenario wird durch einen OO (oder EPC) die Neuplanung oder Anpassung einer Anlage vorgenommen. Der in den funktionalen Anforderungen geplante (Teil-)Prozess besteht aus einer Reaktion und einem Prozessschritt zum Abfüllen des Produktes.

2.1.5.3. Modulbasiertes Engineering

Im funktionalen Design wird zunächst festgelegt, dass die Reaktion mithilfe einer Reaktor-PEA eines speziellen Typs realisiert werden soll. Dies geschieht in einem Engineering-Werkzeug, das bereits auf Basis des Informationsmodells zur Lebenszyklusmodellierung (s. Abschnitt 2.1.2) eine Schnittstelle zu Informationen von PEA-Herstellern (z.B. aus Hersteller-Katalogen) besitzt. So werden Informationen zu verschiedenen Typen von verfügbaren Reaktor-PEAs mit relevanten Eckdaten (KPIs, Abmessungen, etc.) direkt in dem Engineering-Werkzeug zur Auswahl gestellt.

Darüber hinaus wird im funktionalen Design die Verknüpfung der Reaktor-PEA mit dem Rest der Anlage geplant. Die Anschlusspunkte der Reaktor-PEA sind in der zugehörigen Schnittstelle (auch *Interface* genannt) zusammengefasst. Im Szenario besteht die gewählte Reaktor-PEA aus einem Tank und einer Durchflusseinheit, die das Produkt vom Tank abzieht und für weitere Prozessschritte zur Verfügung stellt. Die Durchflusseinheit ist eine FEA der Reaktor-PEA. Im Engineering-Werkzeug werden nicht nur Informationen des Interface der Reaktor-PEA nach extern angezeigt sondern auch Informationen zum PEA-Interface nach intern. So kann während der Spezifikation der Assets direkt im Engineering-Werkzeug ein konkreter FEA-Typ, vielleicht eines anderen Herstellers oder aber aus dem Bestand beim OO selbst, ausgewählt werden.

2.1.5.4. Modulautomatisierung

Die im Szenario für die Anlage spezifizierte Reaktor-PEA (und ebenso die FEA) hat einen eigenen Lebenszyklus. Sie wird beispielsweise von einem Hersteller geplant, gebaut und am Markt angeboten. Es gibt aber auch andere Optionen. Die PEA könnte auch vom Engineering eines OOs oder EPCs geplant und dann bei einem Hersteller in Auftrag gegeben werden. Sie könnte auch von einem OO selbst geplant, gebaut und in einer Anlage betrieben werden. Im Szenario liegen Planungsinformationen zu der Reaktor-PEA unter anderem als R&I in [DEXPI 1.3](#) vor. Während der Durchführung von *ModuLA* haben sich einige Anforderungen an *DEXPI* bzgl. seiner Verwendung im modulbasierten Engineering ergeben; diese sind in Abschnitt [2.1.5.7](#) zusammengefasst.

Weitere Informationen zur Reaktor-PEA sind in anderen Standards, wie z.B. [NE 150](#) oder [NE 159](#) beschrieben. Sie werden im Szenario automatisiert durch nutzerkonfigurierbare Regeln in ein Grundgerüst für ein *MTP*-HMI (Bernshausen u. a., [2016](#)) transformiert. Zu diesem Zweck wurde in *ModuLA* ein Software-Prototyp entwickelt (s. Abschnitt [2.1.5.6](#)). Dieses *MTP*-Grundgerüst wird anschließend von einem Werkzeug zum Automation-Engineering importiert und weiterbearbeitet, bevor das fertig spezifizierte *MTP* in eine POL⁴⁶ für den Betrieb der Reaktor-PEA eingelesen wird.

2.1.5.5. Modellierung

Das Modell *Scenario-2-DEXPI-MTP*⁴⁷ (s.a. Abbildung [A.3](#)) zeigt die Abbildung einzelner Aspekte des Szenarios mit dem Integrationsmodell.

Anmerkung 1 zu Modell *Scenario-2-DEXPI-MTP*

↔ <https://modula.aixcape.org/Scenario-2-DEXPI-MTP.svg#note1>

Der Hersteller der PEA vom Typ *Reaktoflex3000* ([TypeOfFunctionalObject](#) *Reactoflex3000_FD* mit [DesignContext](#) Definition) legt im *functional design* fest ([TypeOfFunctionalObject](#) *Reactoflex3000_FD_SPEC* mit [DesignContext](#) Specification), dass dieser PEA-Typ u.a. aus einem Behälter (*Vessel*) besteht. Dazu wird ein [TypeOfAssembly](#) genutzt, der auf einen [TypeOfFunctionalObject](#) mit [DesignContext](#) Specification verweist, der wiederum eine [Specialization](#) des [TypeOfFunctionalObject](#) *Vessel* ist.

Für die ebenfalls dargestellten Piping-Komponenten und Rohrleitungen im PEA-Typen wurde auf eine Darstellung der [TypeOfAssemblies](#) verzichtet, da sie in Anlehnung an [DEXPI 1.3](#) über mehrere Ebenen (*PipingNetworkSystem*, *PipingNetworkSegment*) erfolgen und die graphische Darstellung überfrachten würde.

⁴⁶Process Orchestration Layer

⁴⁷<https://modula.aixcape.org/Scenario-2-DEXPI-MTP.svg>

Anmerkung 2 zu Modell *Scenario-2-DEXPI-MTP*

↔ <https://modula.aixcape.org/Scenario-2-DEXPI-MTP.svg#note2>

Die PEA wird in der *asset specification* seitens des Herstellers weiter spezifiziert. Es wird ein Einbauplatz für eine FEA vorgesehen (*Flowunit*), die aus einer *Pump* und zwei *Nozzles* besteht. Es werden *Interfaces* mittels zweier `TypesOfFunctionalObject` definiert, die jeweils `Specializations` einer RDL-Klasse *Interface* sind und mittels üblicher `Aggregations` ihre Komponenten referenzieren. Ein „inneres“ Interface, bestehend aus zwei *Nozzles*, beschreibt den Einbauplatz der FEA, ein „äußeres“ Interface, bestehend aus fünf *Pipes*, die äußeren Anschlusspunkte der PEA.

Anmerkung 3 zu Modell *Scenario-2-DEXPI-MTP*

↔ <https://modula.aixcape.org/Scenario-2-DEXPI-MTP.svg#note3>

Der `TypeOfFunctionalObject` *Reactoflex3000_AS* mit `DesignContext` Definition repräsentiert die geplante PEA im Zustand der *asset specification*. Über eine `Specification` ist ihm der `TypeOfFunctionalObject` *Reactoflex3000_AS_SPEC* zugeordnet, der die gewünschten Eigenschaften festlegt, im Beispiel im wesentlichen die Equipment- und Rohrleitungs-Topologie der PEA. *Reactoflex3000_AS_SPEC* wird durch zwei Domänenmodelle repräsentiert: ein *DEXPI-R&I* und ein *MTP-HMI*. Die `domain::Objects` der Domänenmodelle (d.h. im Domänenlayer) lassen sich den *Things* im Integrationslayer zuordnen. Die Zuordnung aus dem *DEXPI-R&I* ist weitgehend trivial, da die Dekomposition der *Things* im Integrationslayer derjenigen im *DEXPI-R&I* entspricht. Es sind also im Wesentlichen 1:1-Beziehungen, die im Interesse der Übersichtlichkeit nicht dargestellt sind.

Anmerkung 4 zu Modell *Scenario-2-DEXPI-MTP*

↔ <https://modula.aixcape.org/Scenario-2-DEXPI-MTP.svg#note4>

Im Gegensatz dazu ist die Zuordnung der `domain::Objects` aus dem *MTP-HMI* zu den *Things* im Integrationslayer komplexer, da das *MTP-HMI* eine Visualisierung der Anlage ist, die auf Inhalte fokussiert, die für den Anlagenfahrer relevant sind, andere aber ausblendet. Wie in Abbildung 2.28 veranschaulicht, können diese Ausblendungen orthogonal zur Objektdekomposition im *DEXPI-R&I* und damit orthogonal zur Dekomposition im Integrationslayer erfolgen. Ein `domain::Object` im HMI kann also eine beliebige Menge von *Things* im Integrationslayer repräsentieren. Im Modell wird dies dadurch erfasst, dass diese Menge zu einem (im Beispiel) `TypeOfFunctionalObject` aggregiert wird, das durch ein `domain::Object` im HMI repräsentiert wird. Konkret ist im Modell dargestellt, dass die *Flowunit* im *MTP* ausgeblendet wird und deshalb eine *MTP-Pipe* eine Komposition aus einer *Pipe*, einem *Flange*, einer *Pump*, einem weiteren *Flange* und einer weiteren *Pipe* repräsentiert.

Anmerkung 5 zu Modell *Scenario-2-DEXPI-MTP*

↪ <https://modula.aixcape.org/Scenario-2-DEXPI-MTP.svg#note5>

Der `TypeOfFunctionalObject` *RotoQuick15926* beschreibt eine mögliche FEA, die für die *Flowunit* geeignet wäre. Die Modellierung erfolgt in üblicher Weise. Das „äußere“ Interface des *RotoQuick15926* (zwei *Nozzles*) wird wie bei der PEA angegeben.

Anmerkung 6 zu Modell *Scenario-2-DEXPI-MTP*

↪ <https://modula.aixcape.org/Scenario-2-DEXPI-MTP.svg#note6>

Im *functional design* für eine Anlage wird festgelegt, dass ein Prozessschritt `TypeOfActivity`) *A1* durch eine modulare Prozesseinheit vom Typ *Reactoflex3000* mit dem Tagnamen *XP4* ausgeführt werden soll: Der `TypeOfFunctionalObject` *XP4* mit `DesignContext` Definition ist eine `Specialization` des `TypeOfFunctionalObject` *Reactoflex3000_AS*. Man beachte, dass diese Modellierung formal der bei konventionellem Vorgehen entspricht, bei der beispielsweise eine `Specialization` des `TypeOfFunctionalObject` *Pump* genutzt werden würde.

Im Modell wird ebenso dargestellt, dass die aus *XP4* (im Fließbild nach unten) austretende *Pipe* eine `Specialization` der entsprechenden *Pipe* des PEA-Typen ist; damit ist exemplarisch die Verknüpfung des Interface des PEA-Typen mit der Umgebung von *XP4* in der geplanten Anlage gezeigt.

Anmerkung 7 zu Modell *Scenario-2-DEXPI-MTP*

↪ <https://modula.aixcape.org/Scenario-2-DEXPI-MTP.svg#note7>

In der *asset specification* wird für *XP4* zusätzlich festgelegt, dass eine PEA vom Typ *Reaktoflex3000_AS* mit einer FEA vom Typ *RotoQuick15926* genutzt werden soll. Dazu wird für den `TypeOfFunctionalObject` *XP4_with_FEA32_SPEC* zusätzlich mittels einer `TypeOfAssembly` auf eine entsprechende Komponente verwiesen, die einem *RotoQuick15926* entspricht, der an der betreffenden Stelle im Interface der PEA (nämlich dem „Platzhalter“ für die *Flowunit*) einzubauen ist.

2.1.5.6. Details zum DEXPI-MTP-Tool

Das *DEXPI-MTP*-Tool ist ein in *ModuLA* entwickelter Software-Prototyp zur Erzeugung eines *MTP*-HMI aus einem *DEXPI*-R&I. Er dient dazu, den Aufwand beim Engineering von PEAs zu reduzieren (s. Abschnitt 2.1.5.4). Abbildung 2.29 gibt eine Übersicht über das Tool.

- Ein R&I, das beispielsweise aus einem geeigneten Engineering-Tool stammt, wird im *DEXPI*-Format (Proteus XML) in das *DEXPI-MTP*-Tool importiert. Die interne Darstellung des R&I erfolgt durch `domain::Objects`. Für diesen Schritt wird die *PID Toolbox* der *pnb plants & bytes GmbH*, Aachen, genutzt.

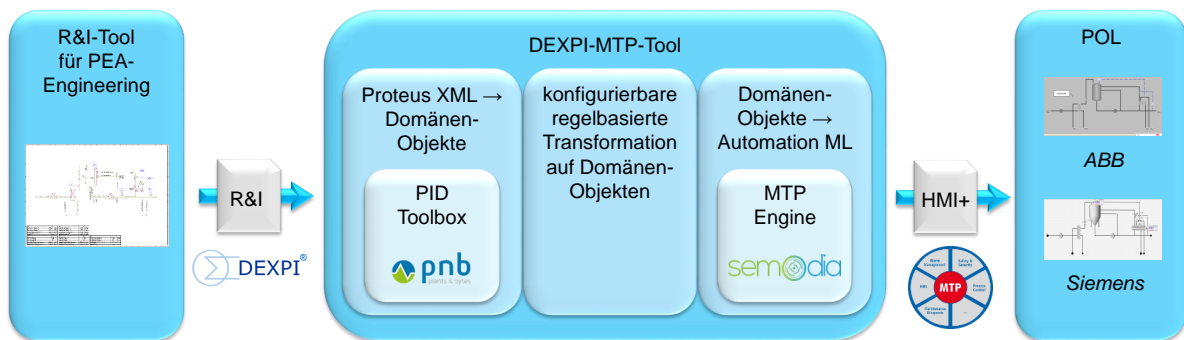


Abbildung 2.29.: *DEXPI-MTP-Tool*.

- Mittels einer konfigurierbaren und regelbasierten Transformation werden Elemente des *MTP*, mit Fokus auf dem HMI, erzeugt.
- Das erzeugte *MTP*-HMI wird in einem XML-basierten Format (Automation ML) exportiert. Dazu wird die *MTP Engine* der *Semodia GmbH*, Dresden, genutzt.

2.1.5.6.1. Konfigurierbare Transformationsregeln

Die Erstellung eines HMI aus einem R&I ist keine bloße Format- oder Datentransformation, d.h. ein Übergang von einer Darstellung in eine andere, inhaltlich gleichartige Darstellung. Vielmehr handelt es sich um eine Engineering-Aufgabe, für die es keine eindeutige, einzig richtige Lösung gibt, sondern die einer Reihe von Einflussgrößen unterliegt:

- Darstellung im Ausgangs-R&I
 - genutzte Normen (insbesondere für Instrumentierung, z.B. DIN 19227, ANSI/ISA 5.1 oder IEC 62424)
 - Werknormen
 - Grad der Detaillierung und Attributierung
 - ...
- Darstellung im HMI
 - angestrebter Detaillierungsgrad
 - strukturelle Vereinfachungen im Vergleich zum R&I (z.B. Ausblenden bestimmter Rohrleitungs-elemente oder Topologieabschnitte)
 - graphische Anordnung
 - Konventionen im Unternehmen
 - ...

DEXPI und *MTP* sind Austauschformate, die im Interesse einer breiten Anwendbarkeit kaum Vorgaben bezüglich der aufgeführten Freiheitsgrade machen. Ein Transformationstool muss deshalb hinreichend konfigurierbar sein, um den Anforderungen der Nutzer zu genügen. Dazu wurde im *DEXPI-MTP*-Tool ein regelbasierter Ansatz realisiert.

Es existieren zwei Arten von Regeln: *Elementare Regeln* haben die Form *wenn-dann*, z.B.: „Wenn eine *PipingComponent* Ziel einer *Pipe* und Start einer *Pipe* ist und nicht mit weiteren *Pipes* oder *SignalConveyingFunctions* verbunden ist, dann sollen die *PipingComponent* entfernt und die beiden *Pipes* zu einer vereinigt werden.“⁴⁸ In der Konfigurationsdatei des *DEXPI-MTP*-Tools wird diese Regel in folgender Form angegeben:

```
REMOVE_INNER_PIPING_COMPONENTS = MergeEdges(
    condition=lambda node, incoming_edge, outgoing_edge:
        isinstance(node.base, dexpi.PipingComponent)
        and incoming_edge.edge_type is EdgeType.piping
        and outgoing_edge.edge_type is EdgeType.piping,
    get_merged_edge_type=lambda: EdgeType.piping)
```

Die zweite Art von Regeln erlaubt es, einzelne Regeln miteinander zu kombinieren: Eine *Sequenz* umfasst mehrere Regeln, die in der vorgegebenen Reihenfolge anzuwenden sind. Eine *Schleife* besagt, dass eine Regel so oft wie möglich angewendet werden soll.

Die Gesamt-Transformation kann damit in einer übergeordneten Regel zusammengefasst werden:

```
DEXPI_TO_MTP = Sequence(
    # vorbereitende Schritte
    Sequence(
        # ...
    ),
    # eigentliche Transformation
    Loop(
        Sequence(
            # Siehe oben.
            REMOVE_INNER_PIPING_COMPONENTS,
            # Entferne PipingComponents, die nur mit einer
            # Rohrleitung verbunden sind (z.B. Blindstutzen).
            REMOVE_DANGLING_PIPING_COMPONENTS,
            # weitere Regeln ...
        )),
    # nachbereitende Schritte
    Sequence(
        # ...
    ))
```

⁴⁸*PipingComponent*, *Pipe*, etc. sind Klassen des *DEXPI*-Informationsmodells. *PipingComponent* umfasst beispielsweise Ventile, Klappen, etc.

Details zu den einzelnen Regeln wurden von den Projektpartnern *AixCAPE* und *Evo-nik* gemeinsam mit der *TU Dresden* und der *Semodia GmbH* festgelegt, vgl. dazu auch (Klose u. a., 2021).

Die Transformationsregeln sind so gestaltet, dass es möglich wäre, eine graphische Nutzeroberfläche zu ihrer Bearbeitung zu erstellen. Auf diese Weise könnten Endnutzer ohne Programmierkenntnisse die Regeln einfacher an ihre jeweiligen Bedürfnisse anpassen.

2.1.5.6.2. HMI-Layout

Neben den abstrakten Informationsobjekten erfordert ein HMI gemäß *MTP* zwingend Layoutinformationen, z.B. zur Platzierung der graphischen Darstellung der Objekte und zum Verlauf der Konnektoren (*Pipes*, etc.). Diese Layoutinformationen sind in den Domänenmodellen gemäß *DEXPI* und *MTP* jeweils enthalten und könnten deshalb grundsätzlich auch mit dem beschriebenen regelbasierten Ansatz behandelt werden. Es hat sich aber als zweckmäßig herausgestellt, nur die Transformation der abstrakten Informationsobjekte regelbasiert durchzuführen, um die Regeln möglichst einfach zu halten.

Die Erstellung des HMI-Layouts erfolgt in einem nachgelagerten Schritt, in dem HMI-Objekte gemäß der Positionierung der entsprechenden R&I-Objekte platziert werden. Die Informationen zur Verknüpfung von R&I- und HMI-`domain::Objects` wird während der regelbasierten Transformation nachgehalten und final im Integrationslayer abgelegt, s. Abbildung 2.28 und Anmerkung 4 zu Modell *Scenario-2-DEXPI-MTP*. Oft entspricht ein HMI-Objekt einer Gruppe von R&I-Objekten. Solche Fälle werden mit Heuristiken behandelt. Wenn beispielsweise für zwei *Pipes* und eine *PipingComponent* im R&I nur eine *Pipe* im HMI dargestellt wird, d.h. die *PipingComponent* im HMI ausgeblendet wird, so wird der Verlauf der *Pipe* im HMI wie die beiden ursprünglichen *Pipes* dargestellt, die ggf. bis zu ihrem Schnittpunkt verlängert oder (bei annähernd parallelem Verlauf) direkt verbunden werden. Ergeben sich dabei unnötig komplizierte Verläufe von *Pipes*, so werden diese “geglättet”.

2.1.5.6.3. Anwendungsbeispiel

Das erweiterte *DEXPI-MTP*-Tool wurde am *DEXPI*-Referenz-R&I getestet (s. Abbildung 2.30). Abbildung 2.31 zeigt das Layout des generierten *MTP*-HMI. Man beachte, dass ein *MTP* nur die Platzierung und Größe der Visualisierungsobjekte enthält. Bei einem Import des *MTP* in ein PLS⁴⁹ würden die grauen Rechtecke durch im PLS hinterlegte Symbole ersetzt werden.

Das erzeugte *MTP* konnte sowohl von *ABB AG* als auch von *Siemens AG* erfolgreich in ihre jeweiligen Leitsysteme importiert werden (Klose u. a., 2021).

⁴⁹Prozessleitsystem

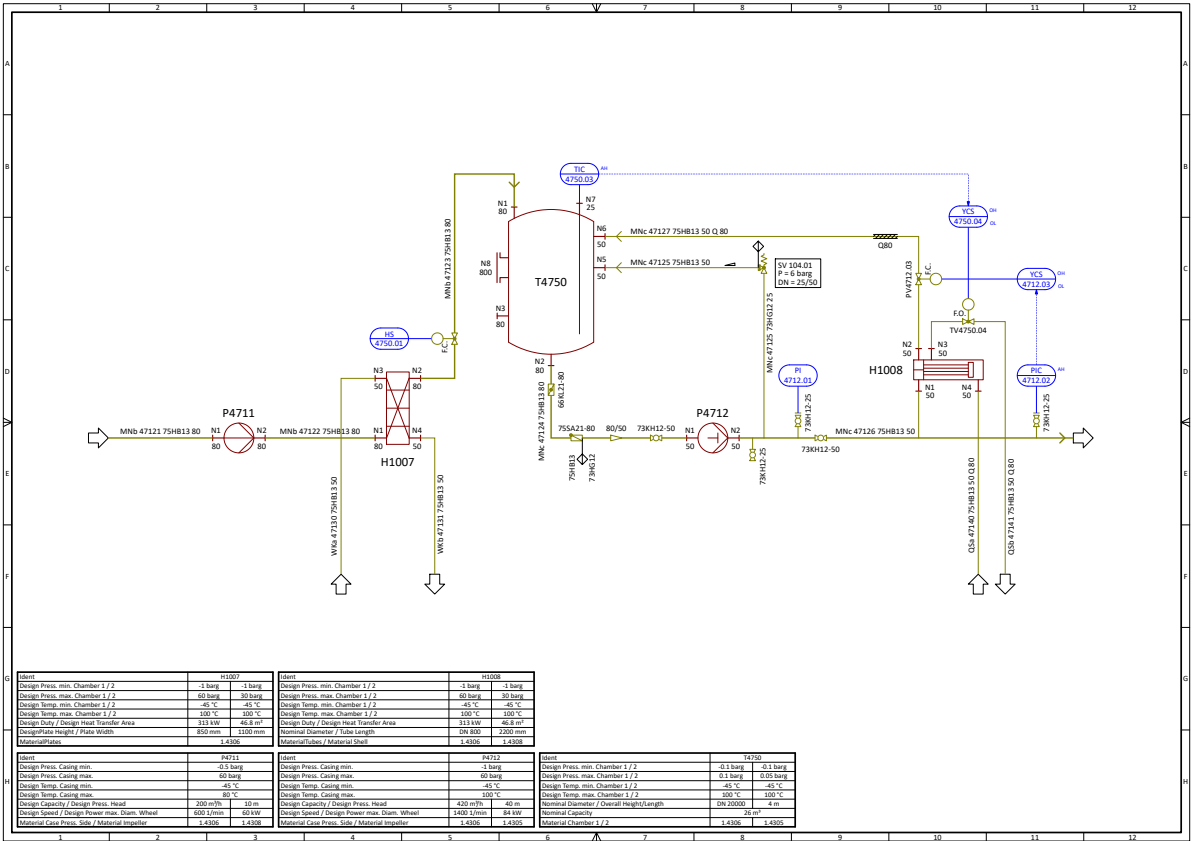


Abbildung 2.30.: Modifiziertes *DEXPI*-Referenz-R&I.

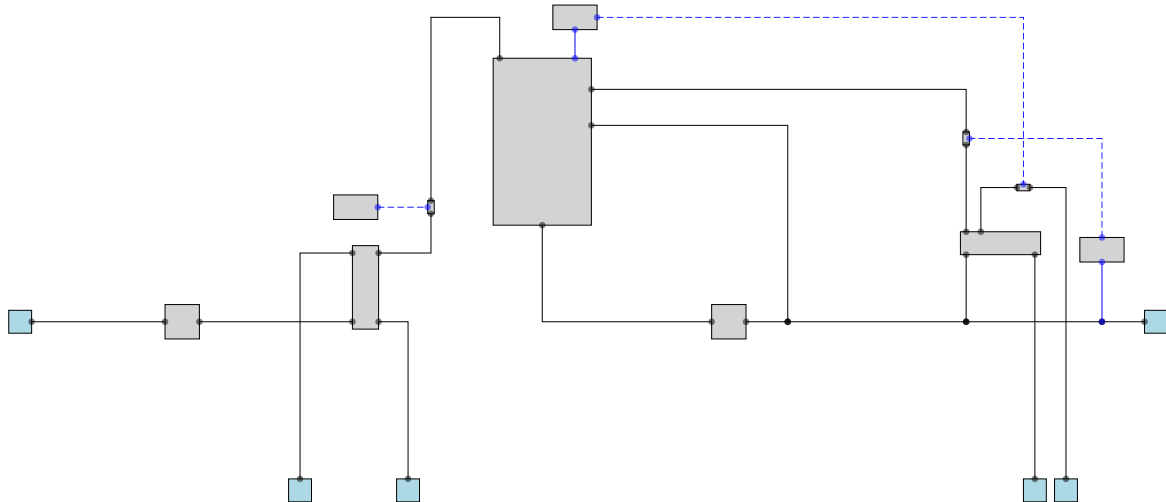


Abbildung 2.31.: Layout des generierten *MTP*-HMI.

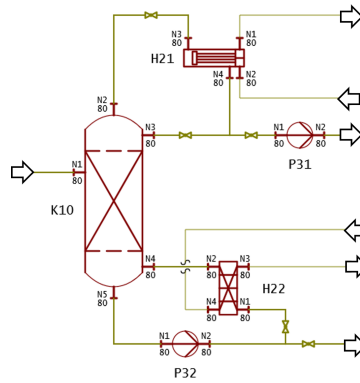


Abbildung 2.32.: R&I einer PEA.

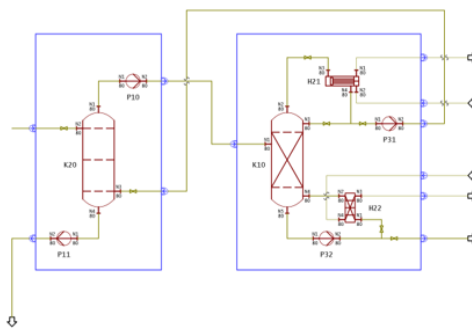


Abbildung 2.33.: R&I einer modularen Anlage mit innerem Aufbau der PEAs.

2.1.5.7. Anforderungen an DEXPI für modulbasiertes Engineering

Beim modulbasierten Engineering lassen sich mindestens drei Arten von R&Is unterscheiden:

- ein R&I einer PEA, das beispielsweise ein PEA-Hersteller erstellt und als Teil der Dokumentation mit der gebauten PEA ausliefert (s. Abbildung 2.32);
- ein R&I einer modularen Anlage, in dem der innere Aufbau der PEAs dargestellt wird (s. Abbildung 2.33);
- ein R&I einer modularen Anlage, in dem der innere Aufbau der PEAs *nicht* dargestellt wird (s. Abbildung 2.34).

Bezüglich der grafischen Darstellung ergeben sich keine besonderen Anforderungen an *DEXPI*, da *DEXPI* das Aussehen von R&Is grundsätzlich nicht vorschreibt. Stattdessen wird eine Vektorgrafik übertragen, deren Elemente den Objekten im *ConceptualModel*, dem semantischen Teilmodell in *DEXPI* (vgl. Spezifikation [DEXPI 1.3](#)) zugeordnet sind.

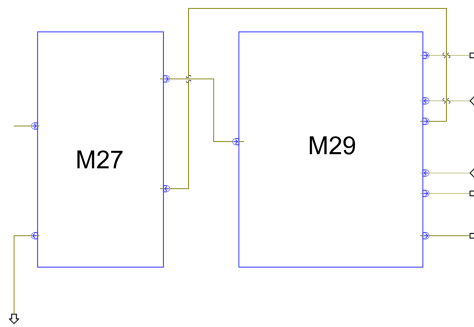


Abbildung 2.34.: R&I einer modularen Anlage ohne inneren Aufbau der PEAs.

Bezüglich des *ConceptualModel* wird empfohlen, eine Möglichkeit einzuführen, diejenigen Objekte, die zu einer PEA gehören, zu gruppieren. Dies kann durch eine zusätzliche Unterklasse von *PlantStructure* erfolgen.

Wünschenswert wäre ebenso eine Möglichkeit, von einem R&I (beispielsweise dem einer modularen Anlage) auf andere R&Is (beispielsweise solche von PEAs) verweisen zu können, um Fälle wie in Abbildung 2.33 übertragen zu können, ohne die vollständigen PEA-R&Is mit übertragen zu müssen. CAE-Tools könnten zudem Visualisierungen wie in Abbildung 2.33 und Abbildung 2.34 als Ansichtsvarianten des selben Modells unterstützen, zwischen denen der Benutzer wählen kann. Angesichts der Einschränkungen des gegenwärtigen Austauschformats auf Basis von Proteus Schema kann ein solcher Verweis nur informal, beispielsweise durch Angabe einer R&I-Nummer oder PEA-Typbezeichnung erfolgen.

2.1.6. Anwendungsbeispiel: Verknüpfung von Produktion und Intralogistik in einem Brownfield-Projekt

Der Ausgangspunkt des dritten Szenarios wurde so gewählt, dass das Potential eines Integrationsmodells auch bei Brownfield-Projekten deutlich wird. Zudem werden die Arbeiten zu den beiden bereits beschriebenen Szenarien eingebunden, so dass dieses Szenario den Charakter eines Gesamtszenarios hat. Es wurde mit den assoziierten Partnern des Projektes *ModuLA* (Schwerpunkt Intralogistik der *BASF*) und in Abstimmung mit den Projekten *ENPRO 2.0 VoPa* und *ENPRO 2.0 MoProLog* ausgearbeitet.

2.1.6.1. Herausforderungen

Die Herausforderungen der ersten beiden Szenarien bestehen auch in diesem Gesamtszenario. Zusätzliche Herausforderungen ergeben sich durch die Betrachtung der Gesamtanlage im *Betrieb* und die *Berücksichtigung der Intralogistik*. Das Integrationsmodell muss beispielsweise folgende Fragen beantworten können:

- *Was ist die Ursache für eine abweichende Produktqualität, die in der Logistik beobachtet wird?*

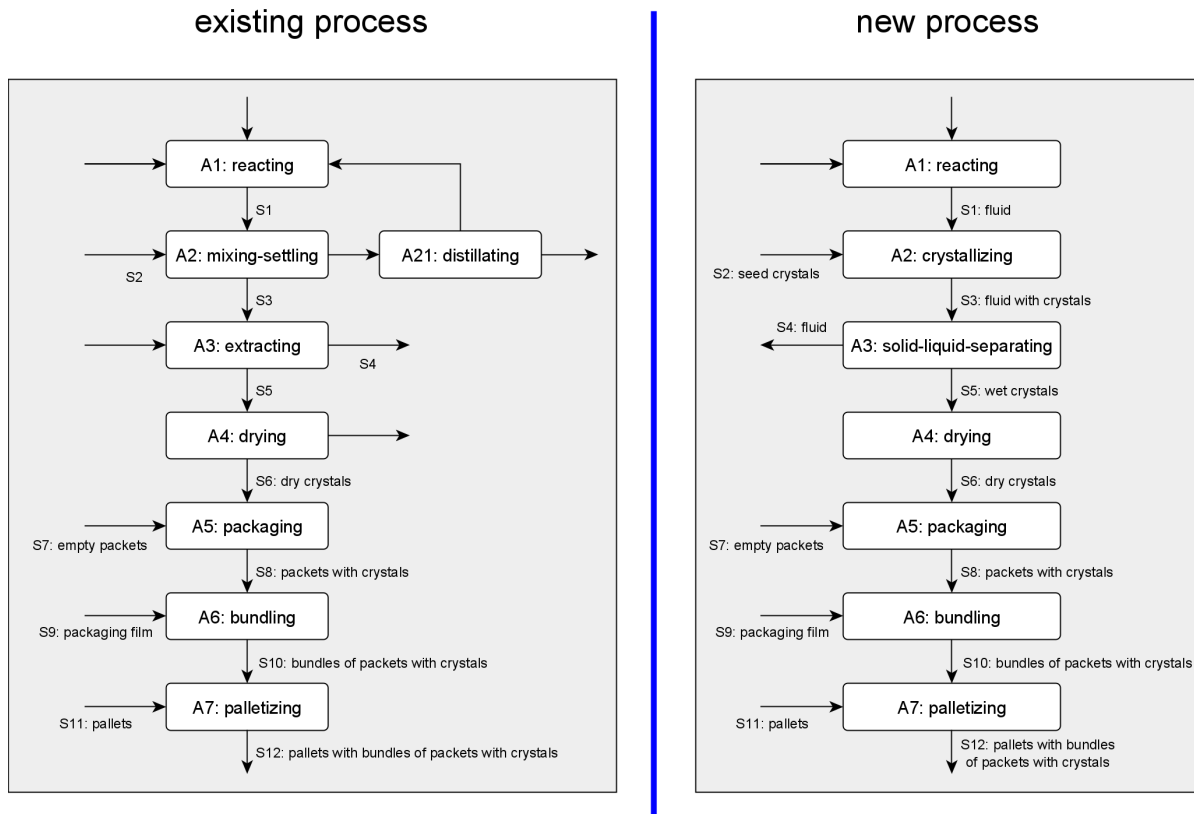


Abbildung 2.35.: Existierender und neuer Prozess im Gesamtszenario Brownfield.

- *In welchen Verpackungseinheiten befindet sich Produkt, dessen Qualität außerhalb der geforderten Toleranzen liegt?*

Die Logistik soll also in die Lage versetzt werden, künftig auf Basis des Integrationsmodells eine Ursachenanalyse für auftretende Probleme auch direkt mithilfe von Informationen aus der Produktion durchführen zu können. Verpackungseinheiten, in die Ausschussware zu Betriebszeiten außerhalb eines Normalbetriebs gelangt ist, sollen leicht identifiziert und lokalisiert werden können, z.B. in einem Gebinde auf einer Palette an einer bestimmten Stelle im Lager.

2.1.6.2. Ausgangspunkt

Das derzeitige Feststoffprodukt eines Unternehmens soll durch eine Produktinnovation am Markt ersetzt werden. Die bestehende Produktionsanlage ist konventionell aufgebaut, in einem Teilprozess sind die Verfahrensschritte zur Reaktion und Abtrennung wie in Abbildung 2.35 auf der linken Seite realisiert. Dabei besteht die Abtrennung aus Einzel-equipments (Mixer-Settler, Destillation, Extraktion und Trocknung) entsprechend den dargestellten Grundoperationen.

Die zu klärende Frage bei der Einführung der Produktinnovation lautet bzgl. der Produktionsanlage:

- *Kann die bestehende Produktionsanlage, z.B. durch Umbau weniger Unit Operations, zur Herstellung der Produktinnovation genutzt werden?*

Im Downstream-Bereich befindet sich die Intralogistik, die im Gegensatz zur Produktionsanlage modular aufgebaut ist, d.h. sie besteht aus modularen logistischen Einheiten.

Diese modularen logistischen Einheiten sind dabei feingranularer als die Grundoperationen im Fließbild. Die *Consumer Unit (packaging)* besteht aus der Produktabfüllung, d.h. der Befüllung von Paketen mit Pulver, der Primärverpackung, d.h. dem Verschluss der Pakete, und deren Kennzeichnung. Die nachgelagerte *Trade Unit (bundling)* besteht aus der Sekundärverpackung, d.h. der Bündelung von Paketen zu Gebinden, und deren Kennzeichnung. Schließlich besteht die *Logistics Unit (palletizing)* aus der Palettierung, d.h. der Ablage von Gebinden auf Paletten, und deren Kennzeichnung.

2.1.6.3. Analyse der bestehenden Produktionsanlage

Es soll die Machbarkeit in der Bestandsanlage geprüft werden. Dazu werden die folgenden Schritte durchgeführt.

2.1.6.3.1. Informationen zusammenstellen

Für die Machbarkeitsanalyse ist zunächst der Zugriff auf vorhandene Lebenszyklus-Informationen zum bestehenden Prozess und der Bestandsanlage erforderlich. Dies sind z.B. Dokumente zur Verfahrensentwicklung, aus dem Labor und dem Pilotbetrieb, Modelle für die Apparateauslegung oder für die Automatisierung sowie Betriebsdaten. Im Rahmen der Arbeiten zum ersten Szenario (s. Abschnitt 2.1.4) wurde bereits erläutert, wie diese Informationen in einem Integrationsmodell abgebildet werden können, so dass sie beispielsweise mit dem *Digital Twin Explorer* abgerufen und gepflegt werden können.

2.1.6.3.2. Laborversuche

Es werden Laborversuche durchgeführt, um das innovative Produkt zu charakterisieren, insbesondere im Hinblick auf Eigenschaften, die für seine Herstellung relevant sind.

2.1.6.3.3. Optionen identifizieren

Für die Herstellung des neuen Produkts ergeben sich zwei Optionen. Option (A): Der Teilprozess bestehend aus Reaktion und Abtrennung bleibt mit den Unit Operations wie auf der linken Seite von Abbildung 2.35 bestehen, und es werden lediglich einzelne Apparate ausgetauscht. Option (B): Die gesamte Abtrennung wird ersetzt durch einen *VoPa*-Apparat, bestehend aus Einheiten zur Kristallisation, Filtration, und Trocknung. Für Details zu einem modularen, wandlungsfähigen und kontinuierlich betriebenen Bänderapparat zur vollintegrierten Partikelsynthese (*VoPa*-Apparat) wird auf die Arbeiten aus dem Projekt *ENPRO 2.0 VoPa* (vgl. Dobler u. a., 2021) verwiesen.

2.1.6.3.4. Optionen bewerten

Die beiden Optionen müssen bewertet werden unter ganz unterschiedlichen Kriterien, wie z.B. Machbarkeit, Energieeffizienz und Anlageninvestment. Insbesondere soll die Downstream-Logistik zur Bewertung beider Optionen bereits einbezogen werden, denn die bestehenden modularen Logistikeinheiten sollen nach Möglichkeit weiter genutzt werden, ggf. mit einer Anpassung der Betriebsparameter.

Im Szenario sind die Bestandsapparate für Mixer-Settler, Destillation und Trocknung zur Realisierung von Option (A) geeignet. Allerdings lässt sich anhand der Matching-Matrix für die Extraktion aus dem Projekt *ENPRO 2.0 SkaMPi* (Grünwald, 2021) feststellen, dass der Bestandsapparat für die Extraktion nicht verwendet werden kann. Es kommen aber andere innovative Extraktions-Apparate in Betracht. Zur Bewertung von Technologien auf Basis von Matching-Matrizen wurde im Projekt *ENPRO 2.0 SkaMPi* von *AixCAPE* der *SkaMPi Technology Matcher*⁵⁰ entwickelt. Ein Demonstrationsvideo⁵¹ zu diesem prototypischen Werkzeug ist auf den Seiten der Austauschplattform *ENPRO Connect* verfügbar. Es wird auch im Gesamtszenario eingesetzt, um ein Ranking für die geeigneten innovativen Technologien zur Extraktion zu erstellen. Zur Bewertung von Option (A) wird abschließend eine Gesamt-Simulation des Prozesses für die meistversprechenden Technologien laut *SkaMPi*-Ranking mit dem Simulationswerkzeug *INOSIM*⁵² durchgeführt.

Zur Bewertung von Option (B) wird im Szenario der *VoPa*-Apparat dimensioniert. Dies kann teilautomatisiert mit Informationen aus den Laborversuchen und ggf. einer Simulation geschehen. Schließlich werden alle Informationen zur Prozessführung des *VoPa*-Apparates in einer Excel-Tabelle, wie sie bislang im Projekt *ENPRO 2.0 VoPa* spezifiziert ist, festgehalten (s. [Anmerkung 4 zu Modell *Scenario-3-VoPa*](#)). Es gibt allgemeine Parameter, wie beispielsweise das Stoffsystem, das Filtermedium und die Taktzeit. Darüber hinaus gibt es Parameter, die spezifisch sind für die verschiedenen Segmente des Apparates (Kristallisation, Fest-flüssig-Trennung und Trocknung). Dies sind beispielsweise die Anzahl der Funktionseinheiten oder Temperaturen bzw. Drücke.

Die Bewertung führt zu folgendem Fazit:

- Die Bestandsanlage kann zur Produktion der Innovation sowohl durch Austausch der Technologie für die Extraktion als auch durch einen Umbau der Abtrennung und den Einsatz des *VoPa*-Apparates verwendet werden.
- Eine Überprüfung der Betriebsbereiche und -parameter für die verschiedenen modularen Einheiten der Intralogistik (wie z.B. die Korngröße, Rieselfähigkeit, maximale Bandgeschwindigkeit, Verpackungsmaterial) ergibt für beide Optionen, dass die modularen Logistikeinheiten der Bestandsanlage ohne bauliche Veränderungen genutzt werden können. Da mehr Produkt pro Zeit hergestellt werden kann, ist allerdings eine Anpassung der Geschwindigkeit erforderlich.

⁵⁰<https://skampi.aixcape.org>

⁵¹<http://www.enpro-initiative.de/ModuLA.html>

⁵²<https://www.inosim.com>

- Eine Kosten-Nutzen-Abschätzung sowie eine hohe Gewichtung von Flexibilität und Energieeffizienz führen im Szenario zu der Entscheidung, dass Option (B) realisiert wird.

2.1.6.3.5. Änderungen dokumentieren

Nachdem die Entscheidung für Option (B) gefallen ist, ist das neue Prozessdesign mit der Prozessführungstabelle des *VoPa*-Apparates und weiteren Informationen zu dokumentieren, die zur Entscheidungsfindung beigetragen haben (*SkaMPi*-Ranking, *INOSIM*-Simulation, etc.). In [Anmerkung 3 zu Modell *Scenario-3-VoPa*](#) wird die Dokumentation der getroffenen Entscheidung im Integrationsmodell beschrieben. In der Realität könnte zur Erstellung dieser Dokumentation der *Digital Twin Explorer* als graphische Schnittstelle zum *Digital Twin Integrator* genutzt werden.

2.1.6.4. Umbau der Anlage

2.1.6.4.1. Vorbereitung

Die erforderlichen Planungsinformationen für den Umbau sind zu erstellen. Insbesondere wird ein R&I zum zuvor dimensionierten *VoPa*-Apparat benötigt.

In der oben erwähnten Excel-Tabelle zur Prozessführung des *VoPa*-Apparates sind die nötigen Informationen zur Erstellung des R&I bereits enthalten, insbesondere die Anzahl und Art der zu nutzenden Segmente im *VoPa*-Apparat (Kristallisation, Filtration oder Trocknung). Um eine manuelle Erstellung des R&I zu vermeiden, wurde ein Prototyp erstellt, der das R&I auf Grundlage der vorhandenen Eingangsinformationen automatisch im *DEXPI*-Format erstellt. [Abbildung 2.36](#) zeigt exemplarisch das erzeugte R&I für einen *VoPa*-Apparat mit drei Segmenten für die Kristallisation, anschließend einem Segment für die Filtration und schließlich zwei Segmenten für die Trocknung.

Darüber hinaus wird die Automatisierung des *VoPa*-Apparates mittels *MTP* vorgenommen. Dabei kann das in [Abschnitt 2.1.5.6](#) beschriebene prototypische Tool zur Erstellung eines *MTP*-Grundgerüsts aus einem *DEXPI*-R&I genutzt werden.

2.1.6.4.2. Umbau

Neben dem Umbau der realen Anlage ist die Modellierung des *asset in operation* im Integrationsmodell anzupassen, s. dazu [Anmerkung 2 zu Modell *Scenario-3-VoPa*](#).

2.1.6.5. Optimierung im Betrieb

Ausgangspunkt für die Ursachenanalyse bei einer Betriebsstörung und die anschließende Identifikation von Ausschussware kann beispielsweise ein Alarm in der Logistikeinheit zur Produktabfüllung sein, d.h. in der Einheit, in der Pakete mit Pulver befüllt werden. Werden Betriebsdaten etwa aus einem PIMS mit dem bisher beschriebenen Integrationsmodell verknüpft, so stehen die notwendigen Informationen bereit, um eine Ursachenanalyse zu betreiben (s. [Anmerkung 2 zu Modell *Scenario-3-Logistics*](#)).

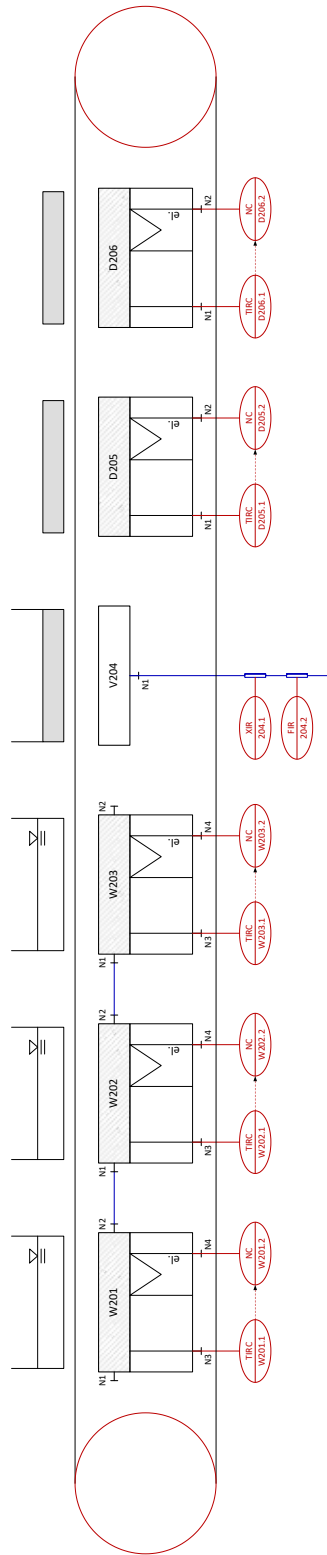


Abbildung 2.36.: Grafik eines erzeugten *DEXPI*-R&Is für einen *VoPa*-Apparat.

Im Falle eines Alarms dürfte sich allerdings die Identifikation von Ausschussware leicht gestalten, da die Logistikeinheiten die Kennzeichnungen der Pakete, die während eines Alarms befüllt werden, markieren (nachhalten). Im Szenario wird deshalb auch der Fall betrachtet, dass kein Alarm einer Produktions- oder Logistikeinheit ausgelöst wird, sondern eine Offline-Analyse des Pulvers zur Sicherstellung der Produktqualität durchgeführt wird. Dabei ergibt sich eine verminderte Rieselfähigkeit des Produktes, die außerhalb der Toleranzen liegt.

Die Realisierung eines Software-Prototypen mit Anbindung an ein echtes PIM-Systeme hätte den Rahmen von *ModuLA* gesprengt. Stattdessen wurden Excel-Tabellen erstellt, die Daten enthalten, die aus einem echten PIM-System hätten stammen können. Diese Daten wurden zusammen mit zusätzlichen Informationen aus dem Integrationsmodell in einem RDF-Graphen abgebildet und die Verfügbarkeit der benötigten Informationen mittels exemplarischer SPARQL-Abfragen geprüft. Es ist anzumerken, dass für eine realistische Anwendung weitere Aspekte zu prüfen sind. Insbesondere verbieten Skalierungseffekte (Menge an PIMS-Daten, Menge an zu berücksichtigenden Paketen, etc.) eine direkte Persistierung der Daten etwa in einem RDF Triple Store. Das Informationsmodell kann hier nur als Fassade genutzt werden, mittels derer auf Daten etwa in einem PIM-System zugegriffen werden kann.

2.1.6.5.1. Ursachenanalyse

Ausgehend von der Kennzeichnung des untersuchten Pakets mit unzureichender Produktqualität kann aufgrund der Integration von Informationen zu einzelnen Paketen (s. [Anmerkung 3 zu Modell *Scenario-3-Logistics*](#)) auf die Zeiten geschlossen werden, zu denen sich der Paketinhalt in bestimmten Teilen der Produktions- und Logistikanlage befand⁵³.

Die beteiligten Produktions- und Logistikeinheiten werden auf fehlerfreien Betrieb geprüft. Aus den erfassten Betriebsdaten geht hervor, dass kein technisches Problem (Statusabfrage) vorliegt. Daraufhin wird analysiert, ob die Betriebsparameter der Produktionseinheiten Veränderungen aufweisen. Im Szenario erreicht der neu eingebaute *VoPa*-Apparat im Trocknungs-Segment nicht die notwendige Mindesttemperatur. Die Soll-Werte für die Temperatur wurden absichtlich im Betrieb des *VoPa*-Apparates herabgesetzt, um das Anbacken von Kristallen zu reduzieren.

Um die Ausschussware zu lokalisieren, wird das Zeitintervall bestimmt, in dem der *VoPa*-Apparat mit zu niedriger Soll-Temperatur betrieben wurde. Anhand dieses Zeitintervalls lässt sich nachvollziehen, in welchen Paketen sich das Pulver mit der verminderten Rieselfähigkeit befindet. Damit können dann auch die entsprechenden Gebinde und die zugehörigen Paletten sowie deren Stellplatz im Lager bestimmt werden.

⁵³Dazu ist insbesondere die Abbildung von Transportgeschwindigkeiten, mittleren Verweilzeiten, etc. erforderlich. Dies ist im Modell nicht dargestellt; die Modellierung würde analog zur Modellierung von Labordaten in Szenario 1 erfolgen.

2.1.6.5.2. Problembhebung

Um das Problem zu beheben, wird die Prozessführung des *VoPa*-Apparates angepasst, es werden zwei Trocknungseinheiten hinzugefügt, so dass das Produkt schonender erhitzt (getrocknet) werden kann. Damit ist auch eine Anpassung des R&Is sowie der Automatisierung erforderlich. Nach Anpassung der Prozessführung ergibt eine erneute Überprüfung, dass die Produktqualität den Anforderungen genügt.

2.1.6.6. Modellierung

Aufgrund der Größe des Szenarios wird das zugehörige Integrationsmodell in zwei Teilen bereitgestellt:

- *Scenario-3-VoPa*⁵⁴ (s.a. Abbildung A.4) legt den Schwerpunkt auf den Umbau der Bestandsanlage, d.h. die Ersetzung des konventionellen Equipments für die Trennung durch einen *VoPa*-Apparat.
- *Scenario-3-Logistics*⁵⁵ (s.a. Abbildung A.5) legt den Schwerpunkt auf den Betrieb in der Intralogistik.

Anmerkung 1 zu Modell *Scenario-3-VoPa*

↪ <https://modula.aixcape.org/Scenario-3-VoPa.svg#note1>

Es wird der Zustand der Bestandsanlage vor dem Umbau dargestellt. Eine modulare Verpackungseinheit mit der Seriennummer *PS9483* (ein `Object`) ist installiert (`InstalledThing`), um eine Funktion (`Function`) vom Typ (`TypeOfFunctionalObject`) *PACKAGING_SYSTEM* zu erfüllen. Die modulare Verpackungseinheit mit der Seriennummer *PS9483* ist vom Typ (`TypeOfFunctionalObject`) *McPackXXL*.

Anmerkung 2 zu Modell *Scenario-3-VoPa*

↪ <https://modula.aixcape.org/Scenario-3-VoPa.svg#note2>

Nach dem Umbau wird dieselbe Verpackungseinheit *PS9483* verwendet (`InstalledThing`), jetzt für eine modifizierte Funktion (`Function`) vom Typ (`TypeOfFunctionalObject`) *PACKAGING_SYSTEM*.

Anmerkung 3 zu Modell *Scenario-3-VoPa*

↪ <https://modula.aixcape.org/Scenario-3-VoPa.svg#note3>

Es werden einige Aspekte dargestellt, die zur Design-Entscheidung beigetragen haben, in der modernisierten Anlage einen *VoPa*-Apparat zu nutzen.

⁵⁴<https://modula.aixcape.org/Scenario-3-VoPa.svg>

⁵⁵<https://modula.aixcape.org/Scenario-3-Logistics.svg>

Der **TypeOfFunctionalObject** *FunctionalDesignOfProduction* repräsentiert das *functional design* der Produktionsanlage. Es ist typisiert als ein **DecisionProblem**. Zwei Alternativen sind aufgeführt: der **TypeOfFunctionalObject** *FunctionalDesignOfProduction_FD_SPEC1* für die Variante mit konventionellem Equipment und der **TypeOfFunctionalObject** *FunctionalDesignOfProduction_FD_SPEC2* für die Variante mit einem *VoPa*-Apparat. Die Bewertungen dieser Alternativen als *gut* (**IsAGoodAlternativeFor**) werden jeweils durch einige exemplarische **Claims** gestützt bzw. in Frage gestellt. Die zweite Variante wird schließlich ausgewählt und als **Specification** dem **TypeOfFunctionalObject** *FunctionalDesignOfProduction* zugeordnet.

Anmerkung 4 zu Modell Scenario-3-VoPa

↪ <https://modula.aixcape.org/Scenario-3-VoPa.svg#note4>

Die ausgewählte Alternative mit einem *VoPa*-Apparat wird mithilfe eines einfachen Domänenmodells (s. Abschnitt 2.1.2.4) auf Grundlage eines im Projekt *ENPRO 2.0 VoPa* genutzten Excel-Templates spezifiziert.

Anmerkung 1 zu Modell Scenario-3-Logistics

↪ <https://modula.aixcape.org/Scenario-3-Logistics.svg#note1>

Die **Activity** *P201029_ForOrder4711* stellt die Ausführung des geplanten Gesamtprozesses **TypeOfActivity** *P201029* für einen bestimmten Auftrag dar. Diese **Activity** besteht aus einzelnen **Activities** (*A5_PackagingForOrder4711*, etc.) entsprechend Grundoperationen im geplanten Gesamtprozess. Für *A5_PackagingForOrder4711* ist exemplarisch die Verknüpfung mit dem **Object** mit der Seriennummer *PS9483* (einem *McPackXXL*) dargestellt (mittels **ParticipationOfPart**-, **Function**- und **InstalledThing**-Dingen).

Anmerkung 2 zu Modell Scenario-3-Logistics

↪ <https://modula.aixcape.org/Scenario-3-Logistics.svg#note2>

Für detaillierte Informationen zur **Activity** *A5_PackagingForOrder4711* wird auf ein Domänenmodell verwiesen. Das Domänenmodell repräsentiert übliche PIMS-Daten. Im Beispiel sind Daten angezeigt, die in Zusammenhang mit dem Unterschreiten des minimalen Füllstands in einzelnen Paketen stehen.

Anmerkung 3 zu Modell Scenario-3-Logistics

↪ <https://modula.aixcape.org/Scenario-3-Logistics.svg#note3>

Es wird die Modellierung einzelner Pakete demonstriert. Das Befüllen des Pakets 42 für den Auftrag 4711 (**Object** *Packet_4711_42*) wurde beispielsweise zum **PointInTime** *2021-05-15T08:30:21* beendet. Später wird dieses Paket Teil des Bündels **Object** *Bundle_4711_3*.

2.1.6.6.1. Fazit

Die exemplarischen Instanzbeispiele zeigen, dass mit dem vorgeschlagenen Informationsmodell (Abschnitt 2.1.2) alle Informationen zur Beantwortung der eingangs gestellten Fragen abgebildet werden können.

2.2. Eingehende Darstellung des voraussichtlichen Nutzens

Die Ergebnisse des Projektes *ModuLA* stellen eine Grundlage dar, um intelligente modulare Prozesseinheiten informationstechnisch miteinander verknüpfen bzw. diese in einen Anlagenlebenszyklus integrieren zu können, d.h. sie leisten einen Beitrag auf dem Weg zu *plug-and-produce-fähigen Modulen*. Wie in Abschnitt 2.1.2 ausführlich beschrieben, sind die Ergebnisse aber nicht auf ein modulares Setting beschränkt sondern können auch zur durchgängigen Abbildung und Bereitstellung von Lebenszyklus-Informationen bei konventionellen und hybriden Anlagen dienen.

Auf Basis eines durchgängigen Integrationsmodells können insbesondere die Planung von modularen Prozesseinheiten, beispielsweise bei Herstellern, und die Planung von modularen Anlagen, z.B. bei Engineering-Firmen, miteinander verzahnt werden.

Die Ergebnisse des Projektes stellen somit Enabling-Technologien auf dem Weg zu einer *modulbasierten Vorgehensweise*, von der Planung bis zum Betrieb und der Analyse einer prozesstechnischen Anlage, und zum *digitalen Zwilling* dar. Die Grundlagen für die Energieeffizienz einer Anlage, ob nun konventionell, modular oder hybrid, werden während der Planung und der Auslegung festgelegt. Mithilfe des ausgearbeiteten durchgängigen Integrationsmodells können energetisch relevante Eigenschaften von Modulen (oder ganzen Teilanlagen) bereits in der Planung zur Verfügung gestellt werden, z.B. durch eine automatisierte Informationsübertragung von Modul-Herstellern auf Basis einer Modulspezifikation. Somit lassen sich künftig mehr Prozessvarianten in immer kürzeren Planungszeiträumen miteinander vergleichen.

Kann in der modulbasierten Planung eine hohe Energieeffizienz erreicht werden, soll diese über den Lebenszyklus erhalten und weiter optimiert werden. Dabei ist der Anlagenlebenszyklus durch Umbauten und Wartung inklusive dem Austausch von einzelnen Maschinen und Apparaten oder ganzen Modulen gekennzeichnet. Liegen nun die Planungsdaten aller vorhergehenden Änderungen und der ursprünglichen Auslegung jederzeit auf Basis des durchgängigen Integrationsmodells vor, lässt sich die Energieeffizienz durch Wahl von passenden Einzel-Komponenten oder Modulen erhalten und im Fall von umfangreichen Umbauten auch schneller optimieren.

Der Nutzen eines durchgängigen Integrationsmodells lässt sich exemplarisch an den verschiedenen Anwendungsszenarien (vgl. Abschnitt 2.1) verdeutlichen. Heute ist die Durchführung dieser Szenarien sehr aufwändig, fehleranfällig und wenig automatisiert. Viele der genannten Informationen liegen häufig nicht in digitaler Form vor und der Zusammenhang zwischen den Informationen muss manuell hergestellt werden. Beispielsweise existiert heute keine Möglichkeit, im Prozessleitsystem (PLS) direkt auf Informationen

aus dem R&I einer (modularen) Anlage zuzugreifen. Das Gesamt-R&I einer modifizierten Anlage, in die ein Modul integriert werden soll, müsste zudem manuell (mit einem CAE-Werkzeug) aus den einzelnen R&Is der Anlage und des Moduls, die vielleicht in unterschiedlichen Formaten vorliegen, erzeugt werden. In Zukunft kann ein Gesamt-R&I automatisch aus den R&I-Informationen der Anlage und eines Moduls generiert werden. Werden die Verknüpfungen zwischen Planungs- sowie PLS-Informationen, d.h. zwischen Domänenstandards wie *DEXPI*, *MTP*, etc., mit dem durchgängigen Integrationsmodell abgebildet, kann künftig z.B. per Klick im PLS direkt das R&I eines zugehörigen Moduls angezeigt werden.

Es gibt vielfältige weitere Anwendungen, in denen die Ergebnisse einen erheblichen Mehrwert darstellen. Ein wichtiger nächster Schritt ist dabei deren Implementierung in industriell nutzbaren Software-Lösungen.

2.3. Eingehende Darstellung der Veröffentlichung der Ergebnisse

Die Arbeiten des Projektes wurden in verschiedenen Standardisierungsgremien, Workshops und auf Konferenzen vorgestellt. Die folgende Liste enthält die Veröffentlichungen in chronologischer Reihenfolge.

- 2019 – Vortrag; *Der digitale Zwilling – Anwendungen für die Prozess- und Anlagenplanung*; Dreiser, C.; Soemers, M.; Theißen M.; PAAT
- Vortrag; *ModuLA: Module im Lebenszyklus einer prozesstechnischen Anlage – Anwendungen für integrierte Modelle*; Soemers, M.; Theißen, M.; 4. ENPRO-Tag
- Poster; *Module im Lebenszyklus einer prozesstechnischen Anlage (ModuLA) – Anwendungen für integrierte Modelle*; Soemers, M.; Theißen, M.; Dreiser C.; Khbeis C.; Wiedau, M.; 4. ENPRO-Tag
- 2020 – Vortrag; *Der Weg zum Digital Twin im modularen Anlagenbau mittels MTP und DEXPI Standard*; Schäfer, C.; Rahm, J.; Klose, A.; Urbas, L.; Soemers, M.; Theißen, M.; 10. ProcessNet-Jahrestagung
- Vortrag; *Modulbasierte Anlagenplanung und Automatisierung – Anwendungen eines digitalen Zwillings basierend auf DEXPI & MTP*; Soemers, M.; Theißen, M.; Rahm, J.; Klose, A.; Urbas, L.; PAAT
- Vortrag; *Modulbasierte Anlagenplanung und Automatisierung – Anwendungen eines digitalen Zwillings basierend auf DEXPI & MTP*; Soemers, M.; Theißen, M.; Rahm, J.; Klose, A.; Urbas, L.; Sitzung der DEXPI-Arbeitsgruppe
- Vortrag; *Towards the Digital Twin - Milestones reached in ENPRO Datenintegration and ModuLA*; Soemers, M.; Theißen, M.; Sitzung des GMA Ausschusses 6.12

- Demomaterial; *Video zu einem prototypischen Software-Werkzeug und Beschreibung des realisierten Szenarios (Lösemittelaustausch)*; Soemers, M.; Theißen, M.; Internetseite der ENPRO-Initiative: <http://www.enpro-initiative.de/ModuLA.html>
- 2021
- Vortrag; *ModuLA: Module im Lebenszyklus einer prozesstechnischen Anlage – Anwendungen für integrierte Modelle*; Dreiser, C.; Soemers, M.; Theißen M.; 5. ENPRO-Tage
 - Artikel; *Efficient Automation Engineering of Modular Process Equipment Assemblies Using the Digital Twin*; Rahm, J.; Theißen, M.; Klose, A.; Soemers, M.; Temmen, H.; Schäfer, C.; Urbas, L.; Journal CIT: <https://doi.org/10.1002/cite.202100100>
 - Vortrag; *Der digitale Zwilling – Anwendungen in einem Brownfield-Szenario von der Produktherstellung bis zur Logistik*; Soemers, M.; Theißen, M.; Weigel, O.; PAAT
- 2022
- Vortrag; *ENPRO 2.0 ModuLA: Module im Lebenszyklus einer prozesstechnischen Anlage – Anwendungen für integrierte Modelle*; Soemers, M.; Theißen, M.; 6. ENPRO-Tag

A. Modelle

In Abschnitt [2.1](#) werden mehrere Modelle beschrieben. Wir empfehlen dem Leser, die online bereitgestellten Versionen der Modelle zu nutzen, die in Abschnitt [2.1](#) verlinkt sind.

Ersatzweise stellen wir die Modelle in diesem Anhang bereit. Den Autoren ist bewusst, dass der Anhang nur in der elektronischen Version dieses Berichts sinnvoll genutzt werden kann. Die gedruckte Version kann aufgrund der Größe der Modelle nur einen ersten Eindruck vermitteln.

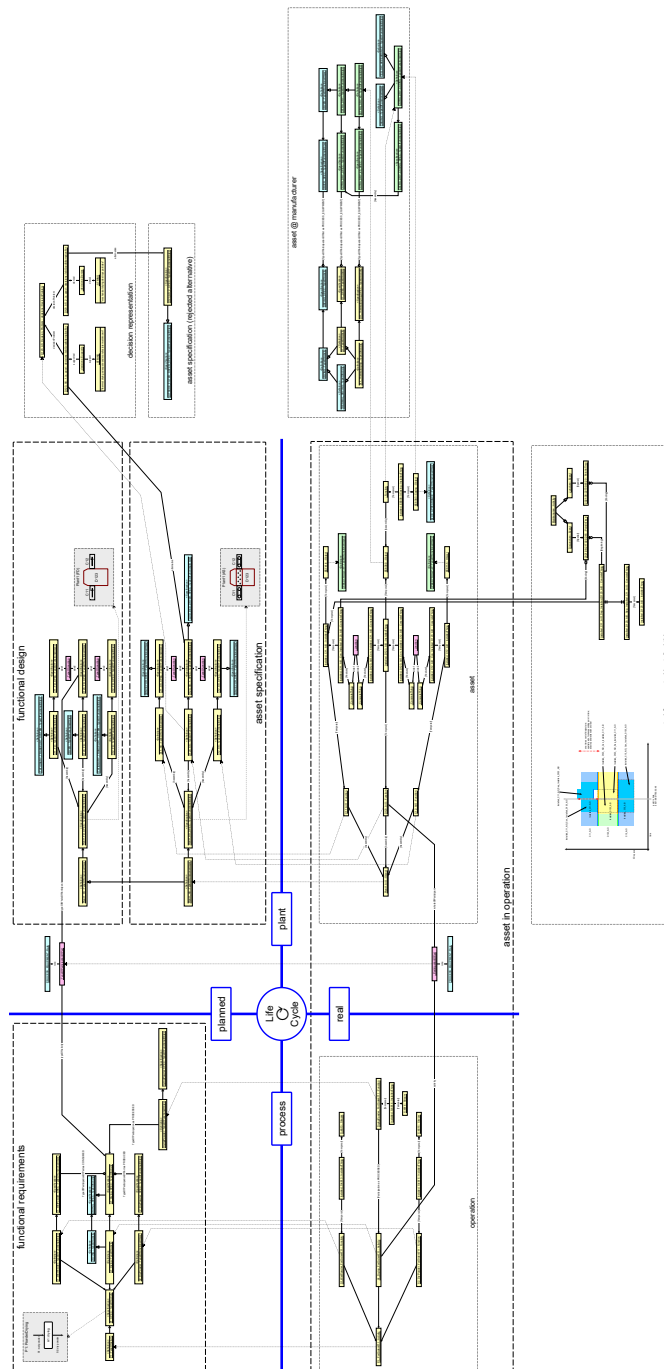


Abbildung A.1.: Modell zu Abschnitt 2.1.3.

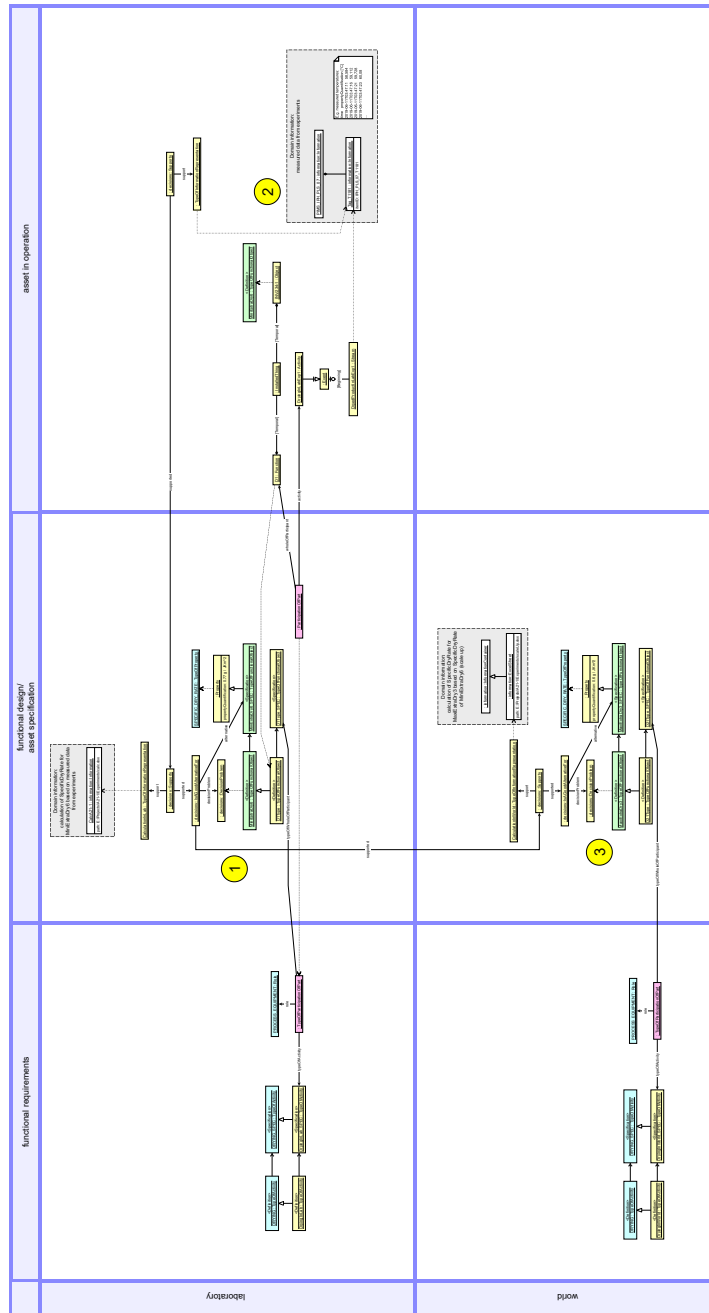


Abbildung A.2.: Modell zu Abschnitt 2.1.4.

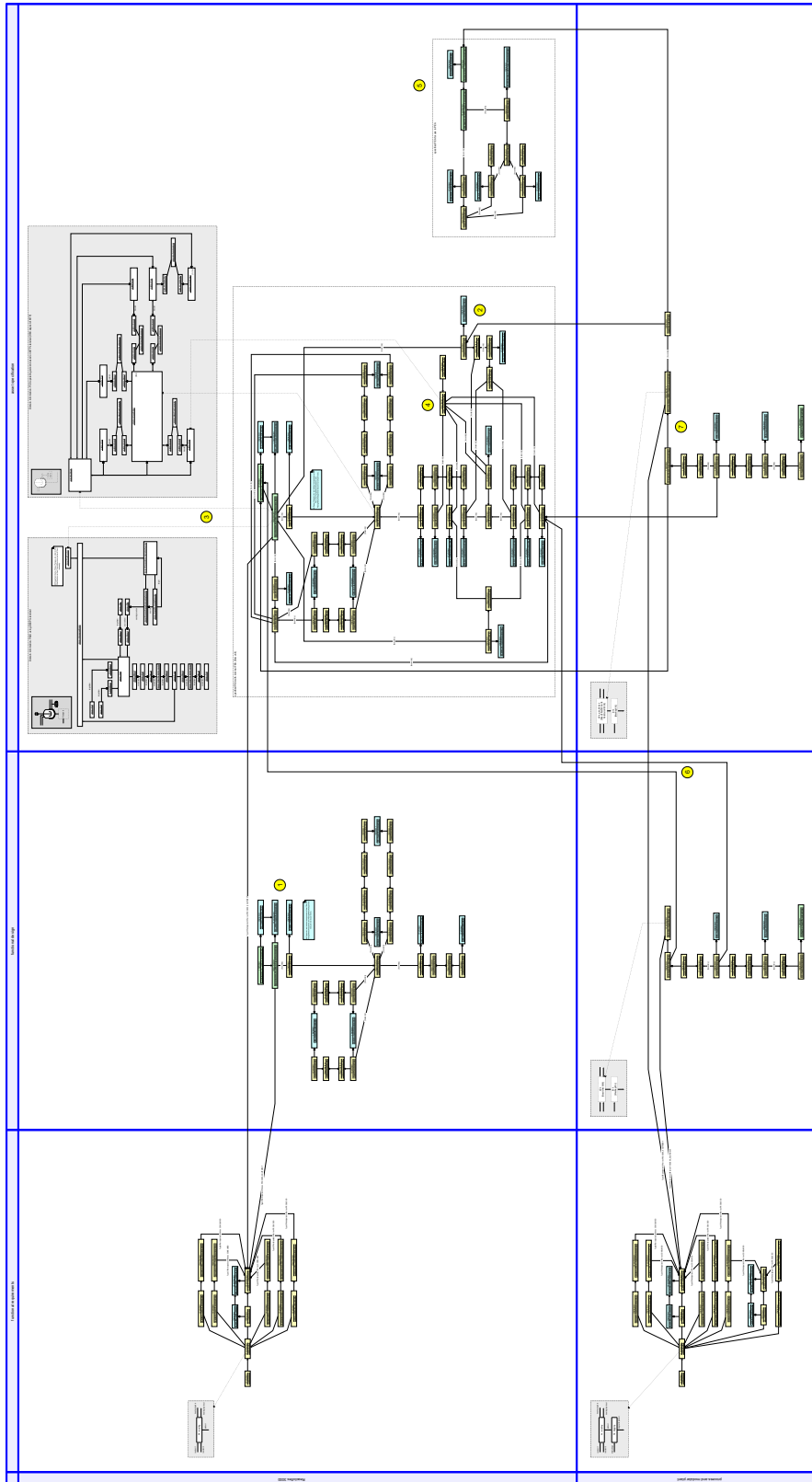


Abbildung A.3.: Modell zu Abschnitt 2.1.5.

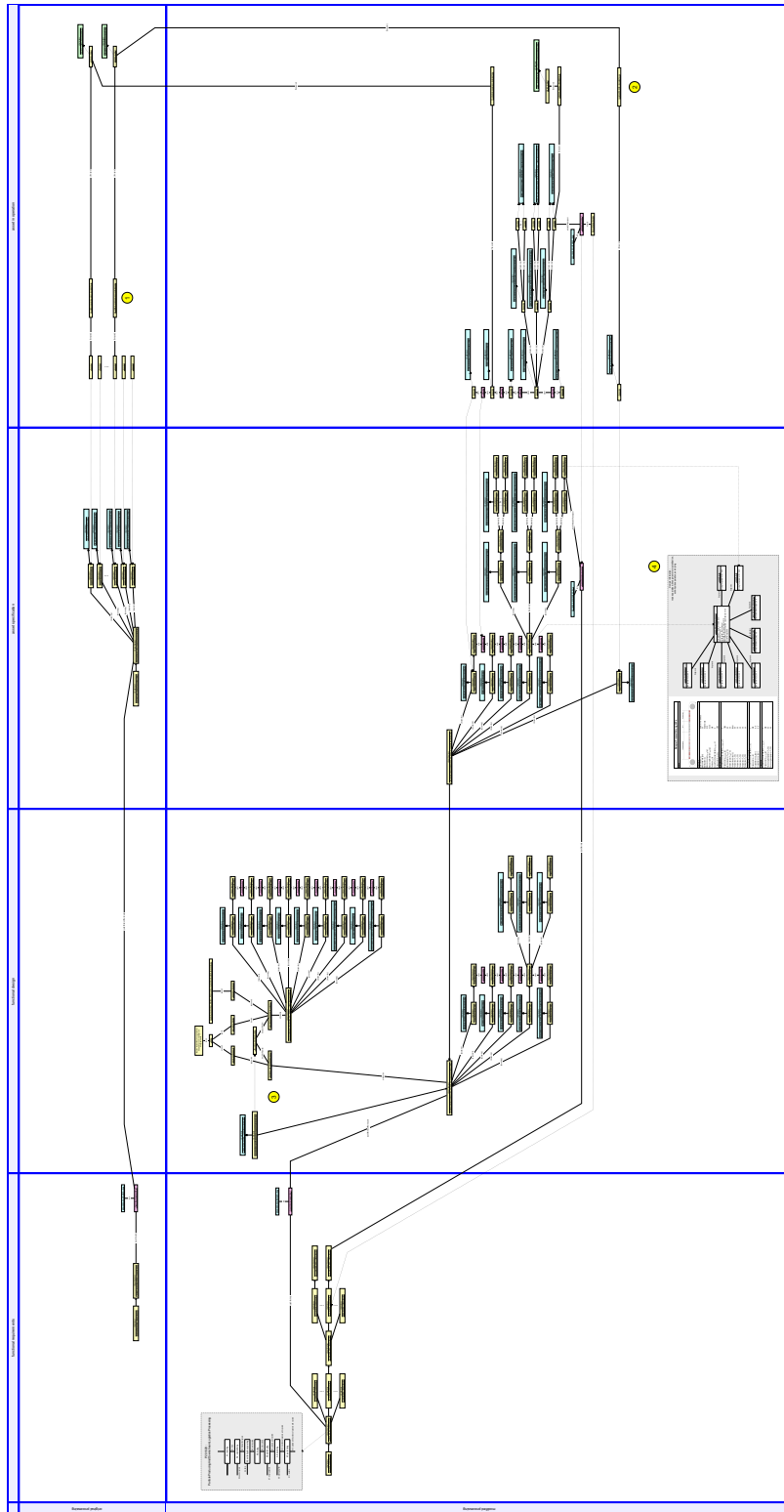


Abbildung A.4.: Modell 1 zu Abschnitt 2.1.6.

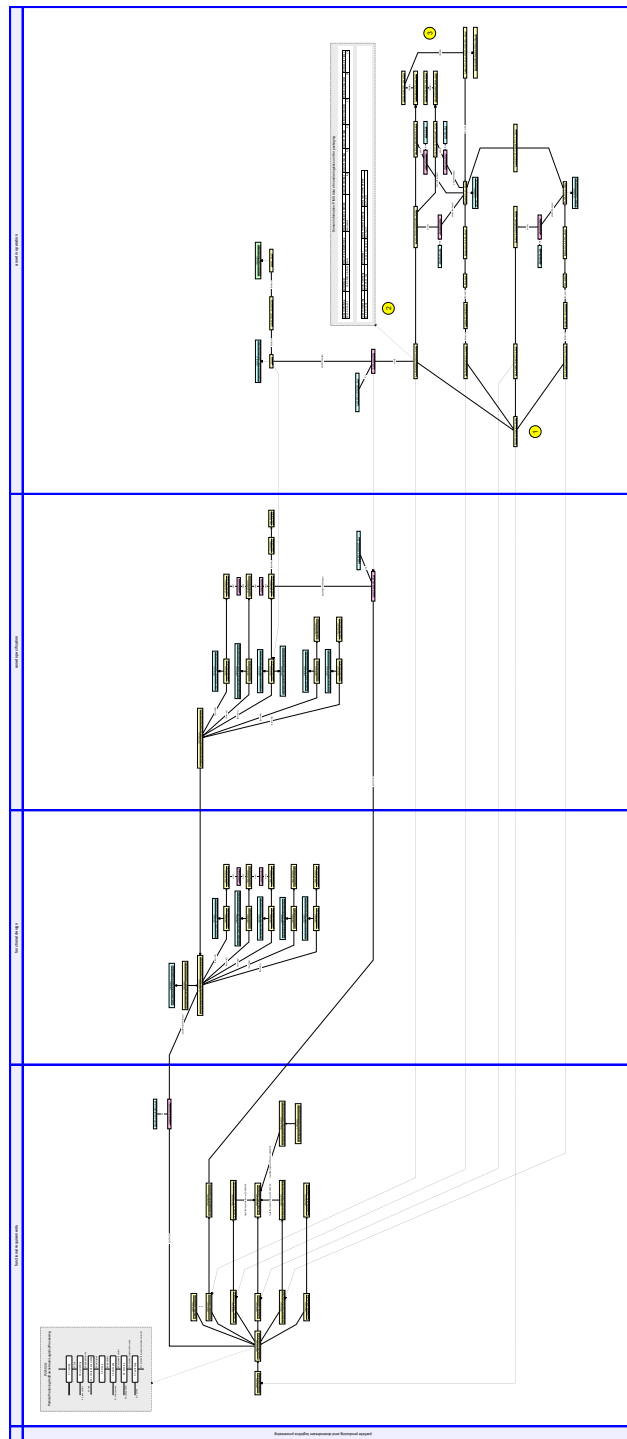


Abbildung A.5.: Modell 2 zu Abschnitt 2.1.6.

Literatur

- Bernshausen, J., Haller, A., Holm, T., Hoernicke, M., Obst, M. und Ladiges, J. (2016). „Namur Modul Type Package – Definition. Beschreibungsmittel für die Automation modularer Anlagen“. In: *atp edition* 58.1-2, S. 72–81. ISSN: 2364-3137. DOI: [10.17560/atp.v58i01-02.554](https://doi.org/10.17560/atp.v58i01-02.554).
- DEXPI Initiative (2022). *DEXPI – Data Exchange in the Process Industry*. Website. URL: <https://www.dexpi.org>.
- DEXPI 1.3 (2021). *DEXPI PID Specification*. Version 1.3. URL: <https://dexpi.org/wp-content/uploads/2020/09/DEXPI-PID-Specification-1.3.pdf>.
- Dobler, T., Buchheiser, S., Gleiß, M. und Nirschl, H. (2021). „Development and Commissioning of a Small-Scale, Modular and Integrated Plant for the Quasi-Continuous Production of Crystalline Particles“. In: *Processes* 9.4. DOI: [10.3390/pr9040663](https://doi.org/10.3390/pr9040663).
- Grünewald, M., Hrsg. (2021). *ENPRO 2.0 – SkaMPi. Abschlussbericht*. URL: <https://www.tib.eu/de/suchen/id/TIBKAT%3A1775577740>.
- ISO 15926. *Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities*. Internationaler Standard.
- ISO 15926-2 (2003). *Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 2: Data model*. Internationaler Standard.
- ISO 15926-7 (2011). *Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 7: Implementation methods for the integration of distributed systems: Template methodology*. Internationaler Standard.
- Klose, A., Rahm, J., Schäfer, C., Soemers, M., Temmen, H., Theißen, M. und Urbas, L. (2021). „Efficient Automation Engineering of Modular Process Equipment Assemblies Using the Digital Twin“. In: *Chemie Ingenieur Technik* 93.12, S. 2081–2091. DOI: [10.1002/cite.202100100](https://doi.org/10.1002/cite.202100100).
- Lee, J. und Lai, K.-Y. (1991). „What’s in design rationale?“ In: *Human-Computer Interaction* 6.3–4, S. 251–280. DOI: [10.1207/s15327051hci0603&4_3](https://doi.org/10.1207/s15327051hci0603&4_3).
- 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>.
- ProcessNet (Okt. 2017). *Modulare Anlagen. Flexible chemische Produktion durch Modularisierung und Standardisierung – Status quo und zukünftige Trends*. White Paper. URL: <http://dechema.de/2017+2+White+Paper+Modular+Plants-path-1,20071550.html>.

- Schembecker, G., Hrsg. (2018). *Modulares Equipment für die energieeffiziente Produktion (ENPRO Modularisierung). Gemeinsamer Abschlussbericht*. DOI: [10.2314/GBV:1027384188](https://doi.org/10.2314/GBV:1027384188).
- Schüller, A., Brendelberger, M., Li, F., Temmen, H. und Zgorzelski, P. (2017). „Durchgängiges PLT Engineering. Die Interaktion von Datenmodellen im PLT-Engineering“. In: *atp edition* 59.10, S. 16–33. ISSN: 2364-3137. DOI: [10.17560/atp.v59i10.1887](https://doi.org/10.17560/atp.v59i10.1887).
- NE 159 (2018). *Standardisierte NAMUR-Schnittstelle für den Datenaustausch zwischen CAE-Systemen der Verfahrensauslegung und der PLT-Hardware-Planung*. NAMUR-Empfehlung. URL: <https://www.namur.net/de/publikationen/news-archiv/ne-159-ist-neu-erschienen.html>.
- NE 150 (2014). *Standardisierte NAMUR-Schnittstelle zum Austausch von Engineering-Daten zwischen CAE-System und PCS-Engineering-Werkzeugen*. NAMUR-Empfehlung. URL: <https://www.namur.net/de/publikationen/news-archiv/die-ne-150-ist-neu-erschienen.html>.
- von Schirach, G. B. (1777). *Biographien des Plutarchs*. Erster Theil. Berlin und Leipzig: George Jacob Decker.
- von Wedel, L., Welke, R., Richert, H., Theißen, M. und Mitsos, A., Hrsg. (2018). *Verbundprojekt „ENPRO“-Datenintegration. Schlussbericht*. DOI: [10.2314/GBV:1037632206](https://doi.org/10.2314/GBV:1037632206).
- West, M. (2011). *Developing High Quality Data Models*. Burlington, MA, USA: Morgan Kaufmann. DOI: [10.1016/C2009-0-30508-5](https://doi.org/10.1016/C2009-0-30508-5).