Ref. Ares(2023)7274242 - 25/10/2023

FACTLOG
www.factlog.eu

ENERGY-AWARE FACTORY ANALYTICS FOR PROCESS INDUSTRY

Deliverable D4.2

Knowledge Graph Modelling

Version 1.2	**Lead Partner** EPFL
Date 25/10/2023	**Project Name** FACTLOG – Energy-aware Factory Analytics for Process Industries

FACTLOG has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 869951

Call Identifier
H2020-NMBP-SPIRE-2019

Topic
DT-SPIRE-06-2019 - Digital technologies for improved performance in cognitive production plants

Project Reference
869951

Start date
November 1st, 2019

Type of Action
IA – Innovation Action

Duration
42 Months

Dissemination Level

X	**PU**	Public
	CO	Confidential, restricted under conditions set out in the Grant Agreement
	CI	Classified, information as referred in the Commission Decision 2001/844/EC

Disclaimer

This document reflects the opinion of the authors only.

While the information contained herein is believed to be accurate, neither the FACTLOG consortium as a whole, nor any of its members, their officers, employees or agents make no warranty that this material is capable of use, or that use of the information is free from risk and accept no liability for loss or damage suffered by any person in respect of any inaccuracy or omission.

Executive Summary

This document is the technical report about WP4 Knowledge Graph and Process Modelling. It presents the FACTLOG ontology developed based on BFO and IoF and knowledge graph modelling based on OWL. Furthermore, it introduces a proposed shop floor case, ontology and OWL models. Finally, reasoning and query of SQWRL and SPARQL are proposed to explain how to analyse the knowledge graph models using the defined rules.

Revision History

Revision	Date	Description	Organisation
0.1	18/12/2021	ToC	EPFL
0.2	10/01/2022	Initial draft	EPFL
0.3	20/01/2022	Contributions in Section 4 and 5	JSI, QLECTOR
0.4	25/01/2022	Further updates in Section 3 and 4	EPFL
1.0	30/01/2022	Final draft ready to be submitted	EPFL
1.2	25/10/2023	Addressing the comments from EC	EPFL

Contributors

Organisation	Author	E-Mail
EPFL	Jinzhi Lu	jinzhi.lu@epfl.ch
EPFL	Dimitris Kiritsis	dimitris.kiritsis@epfl.ch
JSI	Aljaž Košmerlj	aljaz.kosmerlj@ijs.si
JSI	Jože Rožanec	Joze.Rozanec@ijs.si
QLECTOR	Klemen Kenda	klemen.kenda@qlector.com

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose and Scope

This document is the technical report for ontology definition and knowledge graph modelling in Task 4.2 and providing a guidance for defining all the entities and relations in the context of the four pilots (i.e., TUPRAS, BRC, PIA, and CONT)[1]. Knowledge Graph Model (KGM) denotes a generic ontology representation and description with all related product and equipment elements. Moreover, in order to construct cognitive enhanced twins, AI methods, algorithms, mechanisms, services and tools are also described directly and integrated into an overall modelling application or platform. In any specialized FACTLOG use cases, the knowledge graph models require unified and high-level abstract ontology definition in order to describe the related products, methods, algorithms and mechanisms in a standardization way. In the whole FACTLOG platform, the knowledge graph models are used to describe the domain specific knowledge, platform services and their interrelationships. KGM interconnects and interoperates with external AI tools, Optimization tools, Analytics tools, data visualization tools, etc. These tools require to develop the data interfaces based on the developed ontology for importing and exporting knowledge graph models. According to specific cognition needs, the knowledge graph models can be developed using the unified ontology case by case.

The main content deals with the design and implementation of ontology definition and knowledge graph modelling. To summarize, main objectives of this report are:

- Develop a methodology and tools for ontology definition for analytic and dynamic models at various scales (machine, process, production line, factory, etc.) across FACTLOG pilot;
- Define an integrated FACTLOG ontology for technical supports and each pilot;
- Develop knowledge graph models based on the FACTLOG ontology;
- An example of ontology reasoning based on knowledge graph model for supporting decision-makings.
- An integration approach based on Neo4j is introduced to demonstrate how FACTLOG platform integrated with knowledge graph models.

1.2 Relation with other Deliverables

D2.1 is the input of this deliverable for pilot information.

D8.3 is the input of this deliverable for knowledge graph modelling specification.

1.3 Structure of the Document

- Section 2 introduces the background of ontology, semantic modelling and knowledge graph modelling.
- Section 3 introduces FACTLOG ontology based on BFO for knowledge graph modelling.

[1] JEMS pilot did not meet its objectives, especially with regards to the integration of the FACTLOG system to its plant since there is not yet an operative plant in Slovenia.

- Section 4 introduces the knowledge graph models which are developed based on FACTLOG ontology.
- Section 5 introduces a case study supporting reasoning based on the anomaly detection.
- Section 6 introduces integration from knowledge graph models and FACTLOG platform through Neo4j.

2 Background

2.1 Ontology Engineering and Semantic Modelling

2.1.1 Ontology engineering

Ontology engineering is the general term of methodologies and methods for building ontologies. Ontology engineering refers to "The set of activities that concern the ontology development and the ontology lifecycle, the methods and methodologies for building ontologies and the tool suites and languages that support them". The results of ontology engineering provide domain knowledge representation to be reused efficiently and prevent waste of time and money which are usually caused by non-shared knowledge. It helps Information Technology (IT) to operate with interoperability and standardization.

Ontology represents the nature of being, becoming, existence, and so on in the way of philosophy. One of the most well-known is: "ontology is an explicit, formal specification of a shared conceptualization of a domain of interest" [1].

Ontology represents the following ideas together [2]:

- Semantic modelling can help defining the data and the relationships between entities.
- An information model provides the ability to abstract different kind of data and provides an understanding of how the data elements are related.
- A semantic model is a type of information model that supports the modelling of entities and their relationships.
- The total set of entities in our semantic model comprises the taxonomy of classes we use in our model to represent the real world.

2.1.2 Semantic modelling

The main objective of semantic modelling techniques is to define the meaning of data within the context of its correlation, and to model the domain world in the abstract level. The benefits of exploiting semantic data models for business applications are mainly as follows:

- **Avoiding misunderstanding**: by providing a clear, accessible, agreed set of terms, relations as a trusted source and discussions, misunderstandings can easily be resolved.
- **Conduct reasoning**: by being machine understandable and through the usage of logic statements (rules), ontologies enable automatic reasoning and inference which leads to automatic generation of new and implicit knowledge.
- **Leverage resources**: by extending and relating an application ontology to external ontological resources, via manual or automatic mapping and merging processes, the need for repetition of entire design process for every application domain is eliminated.
- **Improve interoperability**: semantic models can serve as a basis for schema matching to support systems' interoperability in close environments where systems, tools and data sources have no common recognition of data type and relationships.

2.1.3 Basic ontology concepts

Ontologies provide formal models of domain knowledge exploited in different ways. Therefore, ontology plays a significant role for many knowledge-intensive applications. Depending on corresponding languages, a number of different knowledge representation formalisms exists. However, they share minimal set of components as follows:

- Classes represent concepts, which are taken in a broad sense. For instance, in the Product Lifecycle domain, concepts are: Life Cycle phase, Product, Activity, Resources, Even, and so on. Classes in ontology are usually organized in taxonomies through which inheritance mechanisms can be applied.
- Relations represent a type of association between concepts of the domain. They are formally defined as any subset of a product of n sets, that is: $R \subset C1 \times C2 \times ... \times Cn$. Ontologies usually contain binary relations. The first argument is known as the domain of the relation, and the second argument is the range.
- Formal axioms serve to model sentences that are always true. They are normally used to represent knowledge that cannot be formally defined by the other components. In addition, formal axioms are used to verify the consistency of the ontology itself or the consistency of the knowledge stored in a knowledge base. Formal axioms are very useful to infer new knowledge.

For instance, Energy Efficiency at Buildings domain could be that it is not possible to build a public building without a fire door (based on legal issues).

- Instances are used to represent elements or individuals in an ontology.

As a Design Rationale (DR), ontology can be used as follows [3]:

- Level 1: Used as a common vocabulary for communication among distributed agents.
- Level 2: Used as a conceptual schema of a relational database. Structural information of concepts and relations among them is used. Conceptualization in a database is nothing other than conceptual schema. Data retrieval from a database is easily done when there is an agreement on its conceptual schema.
- Level 3: Used as the backbone information for a user of a certain knowledge base. Levels higher than this plays role of the ontology, which has something to do with "content".
- Level 4: Used for answering competence questions.
- Level 5: Standardization
    - Standardization of terminology (at the same level of Level 1)
    - Standardization of meaning of concepts
    - Standardization of components of target objects (domain ontology).
    - Standardization of components of tasks (task ontology)
- Level 6: Used for transformation of databases considering the differences of the meaning of conceptual schema. This requires not only the structural transformation but also semantic transformation.
- Level 7: Used for reusing knowledge of a knowledge base using DR information.
- Level 8: Used for reorganizing a knowledge base based on DR information.

2.2 Semantic Modelling Languages

Several semantic modelling languages are developed to support ontology definition and semantic modelling.

- **XML-based Ontology Exchange Language:** The US bioinformatics community designed XOL for the exchange of ontology definitions among a heterogeneous set of software systems in their domain. Researchers developed it after studying the representational needs of experts in bioinformatics. They selected Ontolingua (a Tool for Collaborative Ontology Construction) and OML as the basis for creating XOL, merging the high expressiveness of OKBC-Lite, a subset of the Open Knowledge Based Connectivity protocol, and the syntax of OML, based on XML. There are no tools that allow the development of ontologies using XOL. However, since XOL files use XML syntax, we can use an XML editor to author XOL files.
- **Simple HTML Ontology Extension:** SHOE is a small extension to HTML which allows web page authors to annotate their web documents with machine-readable knowledge. SHOE makes real intelligent agent software on the web possible. HTML was never meant for computer consumption; its function is for displaying data for humans to read. The "knowledge" on a web page is in a human-readable language (usually English), laid out with tables and graphics and frames in ways that we as humans comprehend visually. Unfortunately, intelligent agents aren't human. Even with state-of-the-art natural language technology, getting a computer to read and understand web documents is very difficult. This makes it very difficult to create an intelligent agent that can wander the web on its own, reading and comprehending web pages as it goes. SHOE eliminates this problem by making it possible for web pages to include knowledge that intelligent agents can actually read.
- **Ontology Markup Language:** OML, developed at the University of Washington, is partially based on SHOE. In fact, it was first considered an XML serialization of SHOE. Hence, OML and SHOE share many features. Four different levels of OML exist: OML Core is related to logical aspects of the language and is included by the rest of the layers; Simple OML maps directly to RDF(S); Abbreviated OML includes conceptual graphs features; and Standard OML is the most expressive version of OML. We selected Simple OML, because the higher layers don't provide more components than the ones identified in our framework. These higher layers are tightly related to the representation of conceptual graphs. There are no other tools for authoring OML ontologies other than existing general-purpose XML edition tools.
- **Ontology Interchange Language**: OIL, developed in the OntoKnowledge project (www.ontoknowledge.org/OIL), permits semantic interoperability between Web resources. Its syntax and semantics are based on existing proposals (OKBC, XOL, and RDF(S)), providing modelling primitives commonly used in frame-based
- approaches to ontological engineering (concepts, taxonomies of concepts, relations, and so on), and formal semantics and reasoning support found in description logic approaches (a subset of first order logic that maintains a high expressive power, together with decidability and an efficient inference mechanism). OIL, built on top of RDF(S), has the following layers: Core OIL groups the OIL primitives that have a direct mapping to RDF(S) primitives; Standard OIL is the complete OIL model, using more primitives than the ones defined in RDF(S); Instance OIL adds instances of concepts and roles to the previous model; and Heavy OIL is the layer for future extensions of OIL. OILEd, Protégé2000, and

WebODE can be used to author OIL ontologies. OIL's syntax is not only expressed in XML but can also be presented in ASCII. We use ASCII for our examples.

- **DARPA Agent Markup Language+OIL**: DAML+OIL has been developed by a joint committee from the US and the European Union (IST) in the context of DAML, a DARPA project for allowing semantic interoperability in XML. Hence, DAML+OIL shares the same objective as OIL. DAML+OIL is built on RDF(S). Its name implicitly suggests that there is a tight relationship with OIL. It replaces the initial specification, which was called DAML-ONT, and was also based on the OIL language. OILEd, OntoEdit, Protégé2000, and WebODE are tools that can author DAML+OIL ontologies.
- **OWL**: OWL is the result of the work of the W3C Web Ontology Working Group. This language derived from DAML+OIL and, as the previous languages, is intended for publishing and sharing ontologies in the Web. OWL is built upon RDF(S), has a layered structure and is divided into three sublanguages: OWL Lite, OWL DL and OWL Full. OWL is grounded on Description Logics and its semantics are described in two different ways: as an extension of the RDF(S) model theory and as a direct model-theoretic semantics of OWL. Both of them have the same semantic consequences on OWL ontologies.
- **OWL 2**: OWL 2 is an extension and revision of OWL that adds new functionality with respect to OWL; some of the new features are syntactic sugar (e.g., disjoint union of classes) while others offer new expressivity. OWL 2 includes three different profiles (i.e., sublanguages) that offer important advantages in particular application scenarios, each trading off different aspects of OWL's expressive power in return for different computational and/or implementation benefits. These profiles are:
- **OWL 2 EL**: It is particularly suitable for applications where very large ontologies are needed, and where expressive power can be traded for performance guarantees.
- **OWL 2 QL**: It is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to access the data directly via relational queries (e.g., SQL).
- **OWL 2 RL**: It is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to operate directly on data in the form of RDF triples. OWL 2 ontologies: The Direct Semantics that assigns meaning directly to ontology structures and the RDF- Based Semantics that assigns meaning directly to RDF graphs.
- **Resource Description Framework and RDF Schema:** RDF, developed by the W3C for describing Web resources, allows the specification of the semantics of data based on XML in a standardized, interoperable manner. It also provides mechanisms to explicitly represent services, processes, and business models, while allowing recognition of nonexplicit information. The RDF data model is equivalent to the semantic networks formalism. It consists of three object types:
    - Resources are described by RDF expressions and are always named by URLs plus optional anchor IDs
    - Properties define specific aspects, characteristics, attributes, or relations used to describe a resource
    - Statements assign a value for a property in a specific resource (this value might be another RDF statement)

The RDF data model does not provide mechanisms for defining the relationships between properties (attributes) and resources—this is the role of RDFS. RDFS offers primitives for defining knowledge models that are closer to frame-based approaches. RDF(S) is widely used as a representation format in many tools and projects, such as Amaya, Protégé, Mozilla, SilRI, and so on.

According to W3C, RDF model has advantages as follows:

- The RDF model is made up of triples: as such, it can be efficiently implemented and stored; other models requiring variable-length fields would require a more cumbersome implementation
- The RDF model is essentially the canonicalization of a (directed) graph and has all the advantages (and generality) of structuring information using graphs
- The basic RDF model can be processed even in absence of detailed information (an "RDF schema") on the semantics: it already allows basic inferences to take place, since it can be logically seen as a fact basis
- The RDF model has the important property of being modular

The union of knowledge (directed graphs) is mapped into the union of the corresponding RDF structures. Since RDF is a standard model for data interchange and is a W3Crecommendation designed to standardize the definition and use of metadata-descriptions of Web-based resources, it is well suited to representing data. As knowledge representation, when it comes to semantic interoperability, RDF has significant advantages: The object-attribute structure provides natural semantic units because all objects are independent entities. A domain model – defining objects and relationships – can be represented naturally in RDF. To find mappings between two RDF descriptions, techniques from research in knowledge representation are directly applicable. Therefore, the Z-BRE4K ontology has been implemented in the RDF format.

2.3 Knowledge Graph Modelling

Several scholar, websites and companies have proposed many different definitions. To synthesise a coherent definition that helps frame the discussion about Knowledge Graphs (KGs), [4] [5] [6] [7] [8] [9] [10] [11] [12] were reviewed. They have the following common features:

1. A KG represents interrelationships. All the research studies specify this feature but in a different way.
2. A KG uses techniques to extract knowledge from one or more sources. The kinds of sources differ from one definition to another.
3. The organization is a graph, although the precise meaning of "graph" varies from one definition to another.
4. While a KG must have a schema, not all KG definitions mention it. Those that do mention it specify that the schema defines classes and relations.
5. The KG supports various graph-computing, search, and query interfaces. The supported operations and performance will vary, and the performance will depend on how trade-offs among scalability, performance, and maintainability are handled as well as on other technical issues.

From these features it is apparent that a KG is not simply another way to represent facts. It involves a software architecture that includes active capabilities for extracting and processing the facts. Jans Aasman [13] characterized the operations of a KG as follows:

- Generation:
  - Collection: Ingestion, web extraction, catalog extraction, ontology, ...
  - Processing: Schema mapping, entity resolution, cleaning, ...
- Storage
- Applications: Querying, graph mining, recommendation, search, question answering, ...
- Statistical and machine learning techniques are used for all of the above

Accordingly, these lead to the following proposal for a definition of a KG:

A KG is a representation of a set of statements in the form of a node- and edge- labelled directed multigraph allowing multiple, heterogenous edges at the same nodes. A collection of definitional statements specifying the meaning of KGs labels is called schema.

3 FACTLOG Ontology design based on BFO

3.1 Principles

Entities in the FACTLOG semantic framework have been arranged based on the Basic Formal Ontology (BFO) which is a formal ontology framework developed by Barry Smith and his associates [14]. In BFO, there are two varieties which are continuants comprehending continuant entities such as three-dimensional enduring objects and occurrent comprehending processes conceived as extended through (or as spanning) time. To adopt BFO framework will provide availability to merge the other CT domain ontology structured by BFO.

Originated from BFO, ontology design principles of FACTLOG are as follows:

- use single nouns (except data) and avoid acronyms
- ensure unicity of terms and relational expressions
- distinguish the general from particular
- provide all non-root terms with definitions
- use essential features in defining terms and avoid circularity
- start with the most general terms in the domain
- use simpler terms than the term you are defining (to ensure intelligibility)
- do not create terms for universals through logical combination
- structure ontology around is_a hierarchy and ensure is_a completeness
- single inheritance

3.2 Methodology

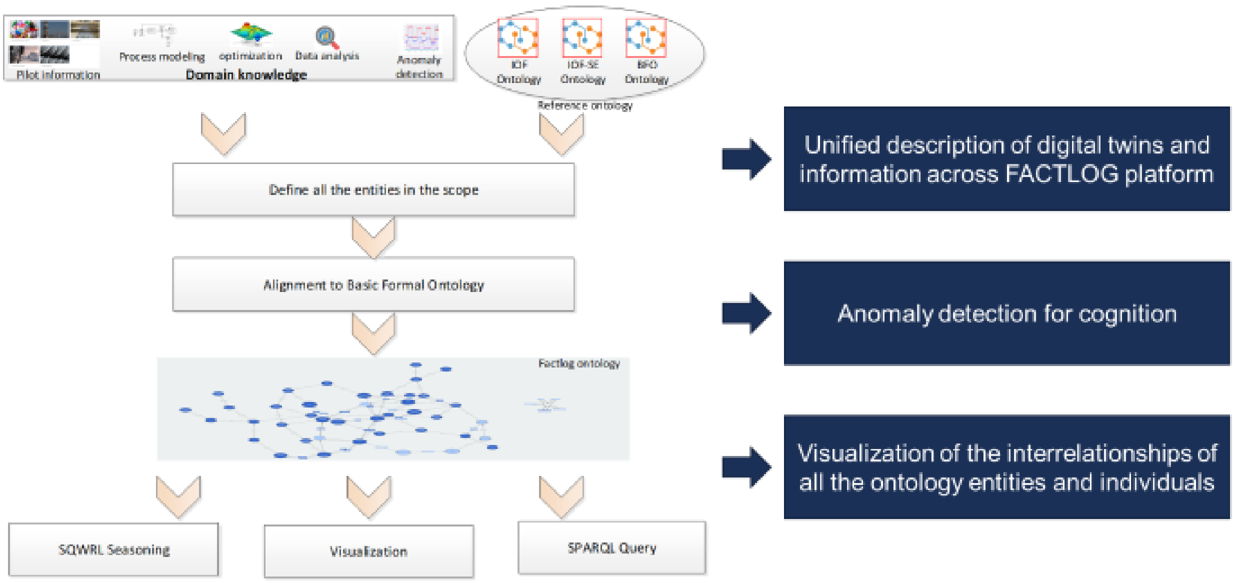

Figure 1:Ontology development workflow

In order to design the unified ontology for developing knowledge graph models supporting cognitive capabilities, the one of the well-known systems thinking development methodology (D8.3) through domain knowledge has been applied to define the domain knowledge including: (i) pilot information; (ii) process modelling; (iii) optimization; (iv) data analysis; (v) anomaly detection. IoF ontology, BFO ontology and IoF SE ontology are three main reference ontology. By composing a top-level overview, abstract concepts form domain specific knowledge from FACTLOG pilots and technical views. After the extraction of entities from FACTLOG pilots, the list of classes was updated in a comparison with existing ontology such as IOF-SE ontology, and IOF ontology. And then, all the entities were rearranged in the BFO structure. Finally, the SQWRL and SPARQL are used to support reasoning and query of the OWL models.

All the ontology concepts are manly used for three aspects:

1. Unified description of digital twin and information across the FACTLOG platform.
2. Ontology reasoning for anomaly detection.
3. Visualization of the interrelationships of all the ontology entities and individuals.

3.3 Ontology Framework

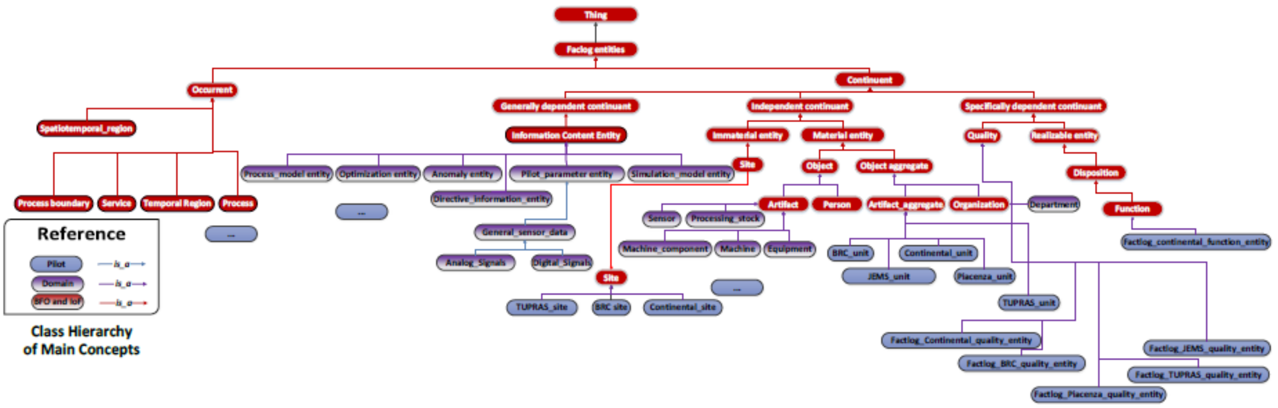

Figure 2: Ontology framework based on BFO

As shown in Figure 2, the FACTLOG ontology is developed based on basic formal ontology. The red blocks refer to BFO and IoF concepts. The purple blocks refer to domain concepts. The blue ones which are defined under BFO and domain concepts are used to define FACTLOG concepts. The whole entire FACTLOG entities include occurrent and continuant entity.

- A continuant is an entity that persists, endures, or continues to exist through time while maintaining its identity.
- An occurrent is an entity that unfolds itself in time or it is the instantaneous boundary of such an entity (for example a beginning or an ending) or it is a temporal or spatiotemporal region which such an entity occupies_temporal_region or occupies_spatiotemporal_region.

Under the occurrent entity, several concepts are defined:

- Process: an occurrent that has temporal proper parts and for some time t, p s-depends_on some material entity at t.

- Process_boundary: a temporal part of a process & p has no proper temporal parts.
- Service: Service is delivered when the service implements the system function.
- Spatiotemporal_region: an occurrent entity that is part of spacetime.
- Temporal_region: an occurrent entity that is part of time as defined relative to some reference frame.

Under the continuant entity, several concepts are defined:

- generically_dependent_continuant is a continuant that generally depends on one or more other entities.
- independent_continuant, a continuant which is such that there is no c and no t such that b s-depends_on c at t.
- specifically_dependent_continuant, a continuant & there is some independent continuant c which is not a spatial region and which is such that b s-depends_on c at every time t during the course of b's existence

Under generically_dependent_continuant entity, several concepts are defined:

- Information_content_entity, a generically dependent continuant that is about something.
    - Process_model_entity, a virtual concept used to define process model.
    - Optimization_entity, a virtual concept used to define optimization concept.
    - Simulation_model_entity, a virtual concept used to define simulation model concepts.
    - Pilot_parameter_concept, a virtual entity to define FACTLOG pilot parameters.
        - General sensor data, sensor data used for all the FACTLOG pilot
    - Directive_information_entity, a plan specification which describes the inputs and output of mathematical functions as well as workflow of execution for achieving a predefined objective. Algorithms are realized usually by means of implementation as computer programs for execution by automata.
    - Anomaly_entity, a virtual entity to support anomaly detection.
    - Data_analysis_entity, a virtual entity used for data analysis.
- Specifically_dependent_continuant, is a continuant & there is some independent continuant c which is not a spatial region and which is such that b s-depends_on c at every time t during the course of b's existence
    - Quality, a specifically dependent continuant that, in contrast to roles and dispositions, does not require any further process in order to be realized.
        - Factlog_BRC_quality_entity, quality used in the BRC pilot.
        - Factlog_CONT_quality_entity, quality used in the CONT pilot.
        - Factlog_JEMS_quality_entity, quality used in the PIA pilot.
        - Factlog_PIA_quality_entity, quality used in the PIA pilot.
        - Factlog_TUPRAS_quality_entity, quality used in the TUPRAS pilot.
    - realizable_entity, a specifically dependent continuant that inheres in some independent continuant which is not a spatial region and is of a type instance of which are realized in processes of a correlated type.
        - Disposition, a realizable entity & b's bearer is some material entity & b is such that if it ceases to exist, then its bearer is physically changed, & b's realization occurs when and because this bearer is in some

special physical circumstances, & this realization occurs in virtue of the bearer's physical make-up.

        - Function, a disposition that exists in virtue of the bearer's physical make-up and this physical make-up is something the bearer possesses because it came into being, either through evolution (in the case of natural biological entities) or through intentional design (in the case of artifacts), in order to realize processes of a certain sort.
            - Factlog_CONT_function_entity, functions used in CONT pilot.
    - Role, a realizable entity & b exists because there is some single bearer that is in some special physical, social, or institutional set of circumstances in which this bearer does not have to be& b is not such that, if it ceases to exist, then the physical make-up of the bearer is thereby changed.
- Independent_continuant, a continuant which is such that there is no c and not such that b s-depends_on c at t.
    - immaterial_entity, which are divided into two subgroups: boundaries and sites, which bound, or are demarcated in relation, to material entities, and which can thus change location, shape and size and as their material hosts move or change shape or size (for example: your nasal passage; the hold of a ship; the boundary of Wales.
        - Site, three-dimensional immaterial entity that is (partially or wholly) bounded by a material entity or it is a three-dimensional immaterial part thereof.
            - BRC_site, site used in BRC.
            - CONT_site, site used in CONT.
            - TUPRAS_site, site used in TUPRAS.
    - material_entity, which can preserve their identity even while gaining and losing material parts. Continuants are contrasted with occurrents, which unfold themselves in successive temporal parts or phases.
        - Object, a material entity which manifests causal unity of one or other of the types CUn listed above & is of a type (a material universal) instance of which are maximal relative to this criterion of causal unity.
        - Artifact, an Object that was designed by some Agent to realize a certain Function.
            - Sensor, a device that produces an output signal for the purpose of sensing of a physical phenomenon.
            - Processing stock, is an artifact in an industrial site corresponds to any material in the process of producing or manufacturing finished product.
            - Machine component, compositions for constructing machines.
            - Machine, a physical system using power to apply forces and control movement to perform an action.
            - Equipment, the set of physical resources serving to equip a person or thing implementing used in an operation or activity.
        - Person, an object that is a human being.
        - Object_aggregate, an object aggregates if and only if there is a mutually exhaustive and pairwise disjoint partition of into objects.

- Artifact_aggregate, a collection of artifacts that designed or arranged by some Agent to realize a certain Function.
    - BRC_unit, a company group generally equivalent in size and character to implement BRC services.
    - CONT_unit, a company group generally equivalent in size and character to implement CONT services.
    - JEMS_unit, a company group generally equivalent in size and character to implement JEMS services.
    - PIA_unit, a company group generally equivalent in size and character to implement PIA services.
    - TUPRAS_unit, a company group generally equivalent in size and character to implement TUPRAS services.
- Organization, an object aggregate that corresponds to social institutions such as companies, societies etc. that does something.
    - Department, an organizational unit in FACTLOG.

3.4 Ontology per Pilot

Using the ontology framework based on BFO, information for each pilot is formalized including process, pilot parameter, artefact and artefact aggregate and department.

3.4.1 BRC pilot ontology

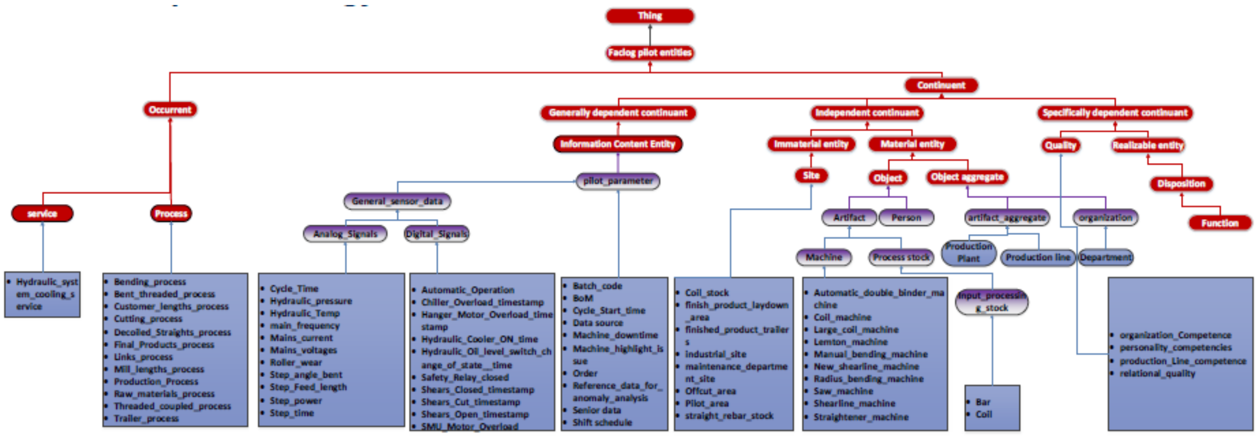

Figure 3: Domain specific ontology for BRC pilot

As shown in Figure 3, domain specific ontology is proposed for BRC pilot. Process, pilot parameter and machine are three important aspects to formalize the BRC scenarios.

BRC pilot has twelve processes:

1. Bending process
2. Bent threaded process
3. Customer length process
4. Cutting process
5. Decoiled straights process
6. Final products process

7. Links process
8. Mill length process
9. Production process
10. Raw material process
11. Threaded coupled process
12. Trailer process

In order to describe the pilot parameters, ten concepts are defined:

1. Batch code
2. BoM
3. Cycle start time
4. Data source
5. Machine downtime
6. Machine highlight issue
7. Order
8. Reference data for anomaly analysis
9. Senior data
10. Shift schedule

There are ten types of machines which are expected to describe.

1. Automatic double binder machine
2. Coil machine
3. Large coil machine
4. Lemton machine
5. Manual bending machine
6. New shearline machine
7. Radius bending machine
8. Saw machine
9. Shearline machine
10. Straightener machine

There are two types of input processing stocks:

1. Bar
2. Coil

There are two types of object aggregates:

1. Production plant
2. Production line

There are four types of domain specific quality for BRC pilot:

1. Organization competence
2. Personality competencies
3. Production Line competence
4. Relational quality

There are eight types of sites:

1. Coil_stock
2. finish_product_laydown_area
3. finished_product_trailers
4. industrial_site
5. maintenance_department_site
6. Offcut_area
7. Pilot_area
8. straight_rebar_stock

There are eleven types of analog signals:

1. Cycle_Time
2. Hydraulic_pressure
3. Hydraulic_Temp
4. main_frequency
5. Mains_current
6. Mains_voltages
7. Roller_wear
8. Step_angle_bent
9. Step_Feed_length
10. Step_power
11. Step_time

There are ten types of digital signals:

1. Automatic_Operation
2. Chiller_Overload_timestamp
3. Hanger_Motor_Overload_timestamp
4. Hydraulic_Cooler_ON_time
5. Hydraulic_Oil_level_switch_change_of_state__time
6. Safety_Relay_closed
7. Shears_Closed_timestamp
8. Shears_Cut_timestamp
9. Shears_Open_timestamp
10. SMU_Motor_Overload

There is one type of service:

1. Hydraulic_system_cooling_service

3.4.2 TUPRAS pilot ontology

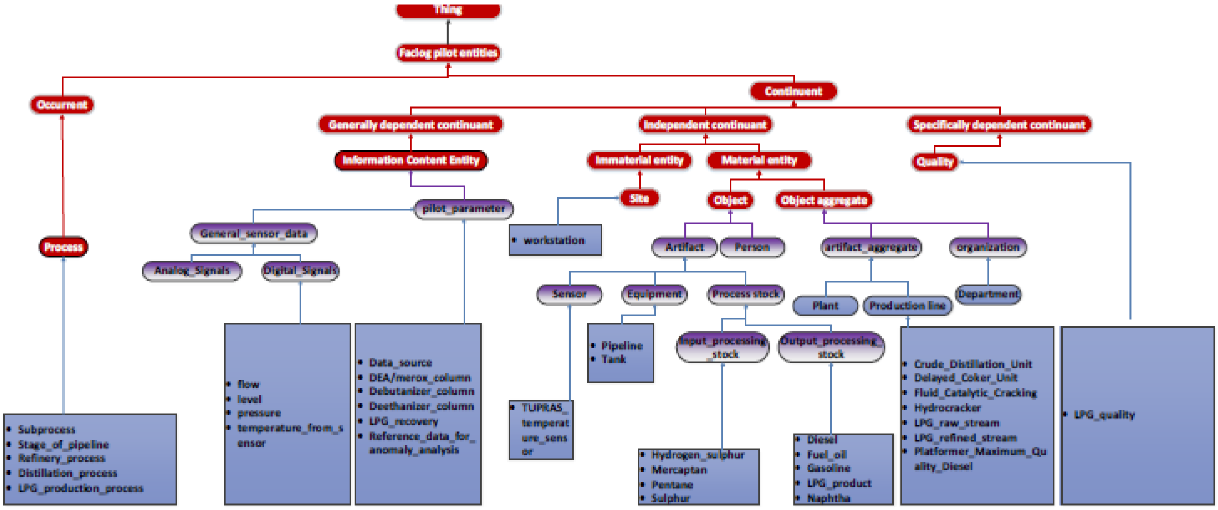

Figure 4: Domain specific ontology for TUPRAS pilot

As shown in Figure 4, domain specific ontology is proposed for TUPRAS pilot. Among them, process, pilot parameter and machine are three important aspects to formalize TUPRAS scenarios.

TUPRAS pilot has five processes:

1. Subprocess
2. Stage_of_pipeline
3. Refinery_process
4. Distillation_process
5. LPG_production_process

In order to describe the pilot parameters, six concepts are defined:

1. Data_source
2. DEA/merox_column
3. Debutanizer_column
4. Deethanizer_column
5. LPG_recovery
6. Reference_data_for_anomaly_analysis

There are totally two types of equipment which are expected to describe:

1. Pipeline
2. Tank

There is one type of sensor:

1. TUPRAS_temperature_sensor

There are four types of input processing stocks:

1. Hydrogen_sulphur

2. Mercaptan
3. Pentane
4. Sulphur

There are five types of output processing stocks:

1. Diesel
2. Fuel_oil
3. Gasoline
4. LPG_product
5. Naphtha

There are two types of object aggregates:

1. Production plant
2. Production line
    - Crude_Distillation_Unit
    - Delayed_Coker_Unit
    - Fluid_Catalytic_Cracking
    - Hydrocracker
    - LPG_raw_stream
    - LPG_refined_stream
    - Platformer_Maximum_Quality_Diesel

There is one type of domain specific quality:

1. LPG_quality

There is one type of site:

1. Work station

There are four types of digital signals:

1. flow
2. level
3. pressure
4. temperature_from_sensor

3.4.3 PIA pilot ontology

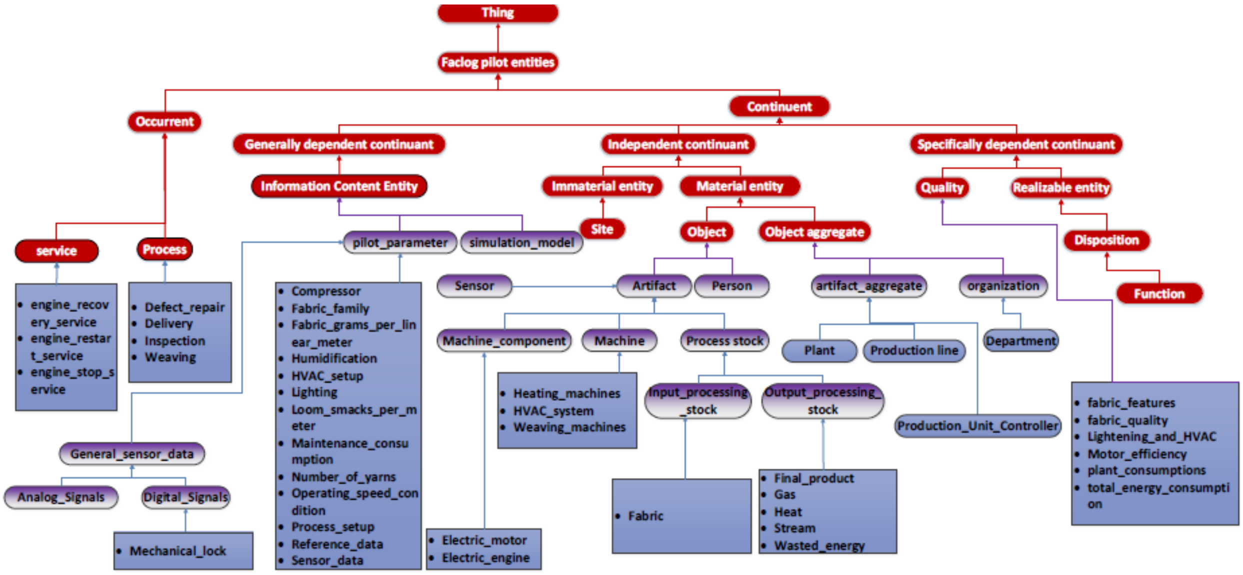

Figure 5: Domain specific ontology for PIA pilot

As shown in Figure 5, domain specific ontology is proposed for PIA pilot. Among them, process, pilot parameter and machine are three important aspects to formalize PIA scenarios.

PIA pilot has four processes:

1. Defect_repair
2. Delivery
3. Inspection
4. Weaving

In order to describe the pilot parameters, fourteen concepts are defined:

1. Compressor
2. Fabric_family
3. Fabric_grams_per_linear_meter
4. Humidification
5. HVAC_setup
6. Lighting
7. Loom_smacks_per_meter
8. Maintenance_consumption
9. Loom_smacks_per_meter
10. Number_of_yarns
11. Operating_speed_condition
12. Process_setup
13. Reference_data
14. Sensor_data

There are totally three types of machines which are expected to describe.

1. Heating_machines

2. HVAC_system
3. Weaving_machines

There are totally two types of machine components:

1. Electric_motor
2. Electric_engine

There are three types of object aggregates:

1. Production plant
2. Production line
3. Production_Unit_Controller

There is 1 type of input processing stocks:

1. Fabric

There are 5 types of output processing stocks:

1. Final_product
2. Gas
3. Heat
4. Stream
5. Wasted_energy

There are six types of domain specific quality:

1. fabric_features
2. fabric_quality
3. Lightening_and_HVAC
4. Motor_efficiency
5. plant_consumptions
6. total_energy_consumption

There are 3 types of service:

1. engine_recovery_service
2. engine_restart_service
3. engine_stop_service

There is 1 type of digital signal:

1. Mechanical_lock

3.4.4 JEMS pilot ontology

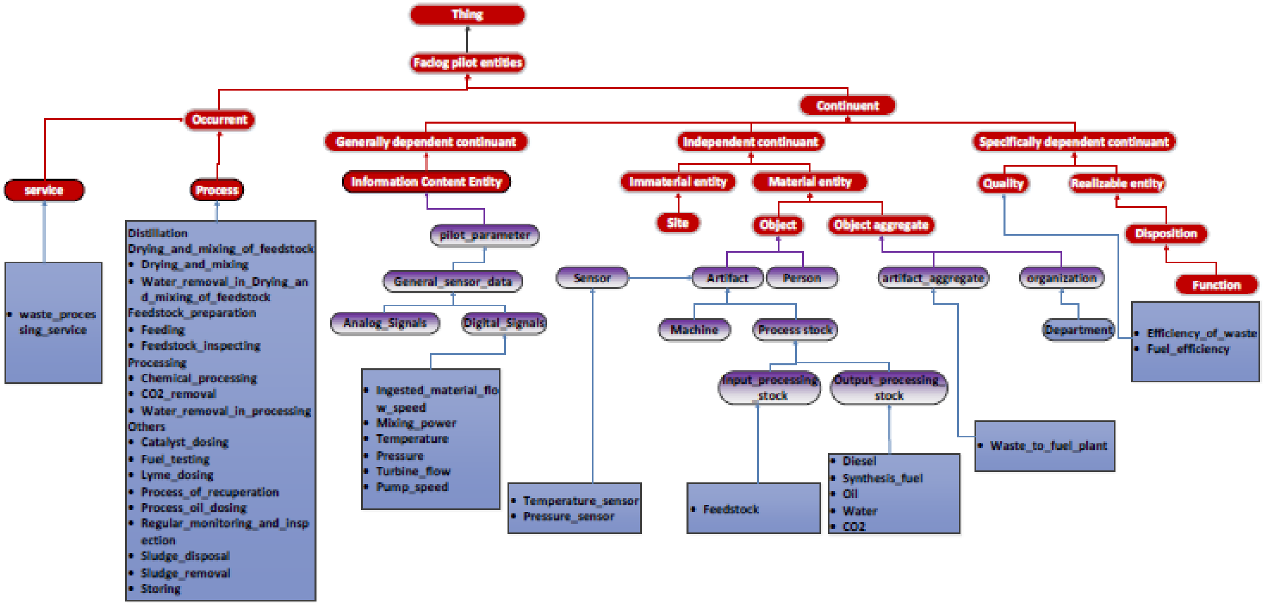

Figure 6: Domain specific ontology for JEMS pilot

As shown in Figure 6, domain specific ontology is proposed for JEMS pilot[2]. Among them, process, pilot parameter and machine are three important aspects to formalize JEMS scenarios.

Among the entire processes, JEMS pilot has several processes:

1. Distillation
2. Drying_and_mixing_of_feedstock
   - Drying_and_mixing
   - Water_removal_in_Drying_and_mixing_of_feedstock
3. Feedstock_preparation
   - Feeding
   - Feedstock_inspecting
4. Processing
   - Chemical_processing
   - CO2_removal
   - Water_removal_in_processing
5. Others
   - Catalyst_dosing
   - Fuel_testing
   - Lyme_dosing
   - Process_of_recuperation
   - Process_oil_dosing
   - Regular_monitoring_and_inspection
   - Sludge_disposal
   - Sludge_removal
   - Storing

[2] JEMS pilot did not meet its objectives, especially with regards to the integration of the FACTLOG system to its plant since there is not yet an operative plant in Slovenia.

In order to describe the pilot parameters, 6 concepts for general sensor data are defined:

1. Ingested_material_flow_speed
2. Mixing_power
3. Temperature
4. Pressure
5. Turbine_flow
6. Pump_speed

There are two types of sensors:
1. Temperature_sensor
2. Pressure_sensor

There are two types of domain specific quality for JEMS pilot:
1. Efficiency_of_waste
2. Fuel_efficiency

There are 1 type of input processing stocks:
1. Feedstock

There are 5 types of output processing stocks:
1. Diesel
2. Synthesis_fuel
3. Oil
4. Water
5. CO2

There are 1 type of service:
1. waste_processing_service

There are 1 type of object aggregates:
1. Waste_to_fuel_plant

There are 6 types of digital signals:
1. Ingested_material_flow_speed
2. Mixing_power
3. Temperature
4. Pressure
5. Turbine_flow
6. Pump_speed

3.4.5 CONT pilot ontology

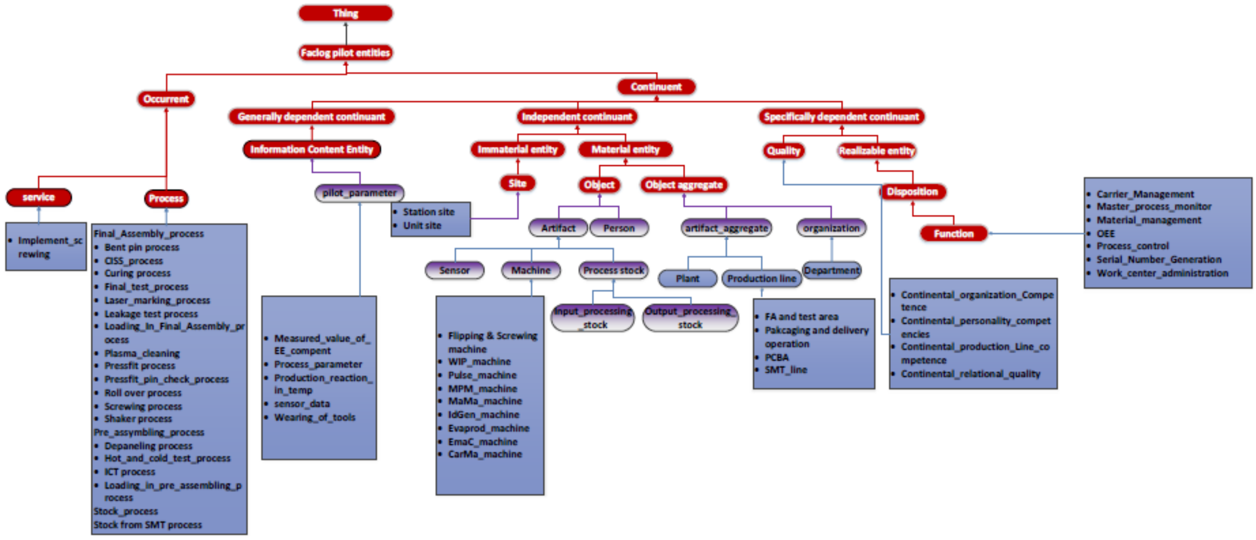

Figure 7: Domain specific ontology for CONT pilot

As shown in Figure 7, domain specific ontology is proposed for CONT pilot. Among them, process, pilot parameter and machine are three important aspects to formalize CONT scenarios.

Among the entire processes, CONT pilot has four processes:

7. Final_Assembly_process
   - Bent pin process
   - CISS_process
   - Curing process
   - Final_test_process
   - Laser_marking_process
   - Leakage test process
   - Loading_In_Final_Assembly_process
   - Plasma_cleaning
   - Pressfit process
   - Pressfit_pin_check_process
   - Roll over process
   - Screwing process
   - Shaker process
8. Pre_assymbling_process
   - Depaneling process
   - Hot_and_cold_test_process
   - ICT process
   - Loading_in_pre_assembling_process
9. Stock_process
10. Stock from SMT process

In order to describe the pilot parameters, five concepts are defined:

1. Measured_value_of_EE_compent
2. Process_parameter
3. Production_reaction_in_temp
4. sensor_data
5. Wearing_of_tools

There are totally nine types of machines which are expected to describe.

1. Flipping & Screwing machine
2. WIP_machine
3. Pulse_machine
4. MPM_machine
5. MaMa_machine
6. IdGen_machine
7. Evaprod_machine
8. EmaC_machine
9. CarMa_machine

There are two types of object aggregates:

1. Production plant
2. Production line
    - FA and test area
    - Pakcaging and delivery operation
    - PCBA
    - SMT_line

There are four types of domain specific quality:

1. CONT_organization_Competence
2. CONT_personality_competencies
3. CONT_production_Line_competence
4. CONT_relational_quality

There are seven types of domain specific function:

1. Carrier_Management
2. Master_process_monitor
3. Material_management
4. OEE
5. Process_control
6. Serial_Number_Generation
7. Work_center_administration

There are two types of sites:

1. Station site
2. Unit site

There is one type of service:

1. Implement_screwing

3.5 Ontology for Directive_information_entity

The ontology in the Directive_information_entity is used to represent the general information concepts which may be used in different FACTLOG pilots:

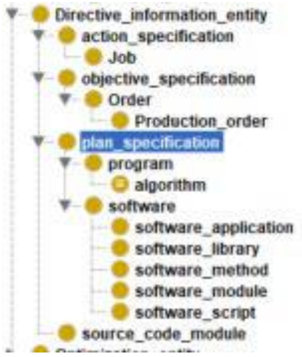

Figure 8: Ontology classes for Directive_information_entity

1. Action_specification, a directive information entity that describes an action the bearer will take.
    - Job, a piece of work.
2. Objective_specification, a directive information entity that describes an intended process endpoint. When part of a plan specification the concretization is realized in a planned process in which the bearer tries to affect the world so that the process endpoint is achieved.
    - Order, the announcement of an intended purchase.
        - Production_order
3. Plan_specification, a directive information entity with action specifications and objective specifications as parts that, when concretized, is realized in a process in which the bearer tries to achieve the objectives by taking the actions specified.
    - program
        - algorithm
    - software
        - software_application
        - software_library
        - software_method
        - software_module
        - software_script
    - source_code_module

3.6 Ontology for Optimization

The ontology in the optimization is used to represent the optimization concepts which may be used in different FACTLOG pilots:

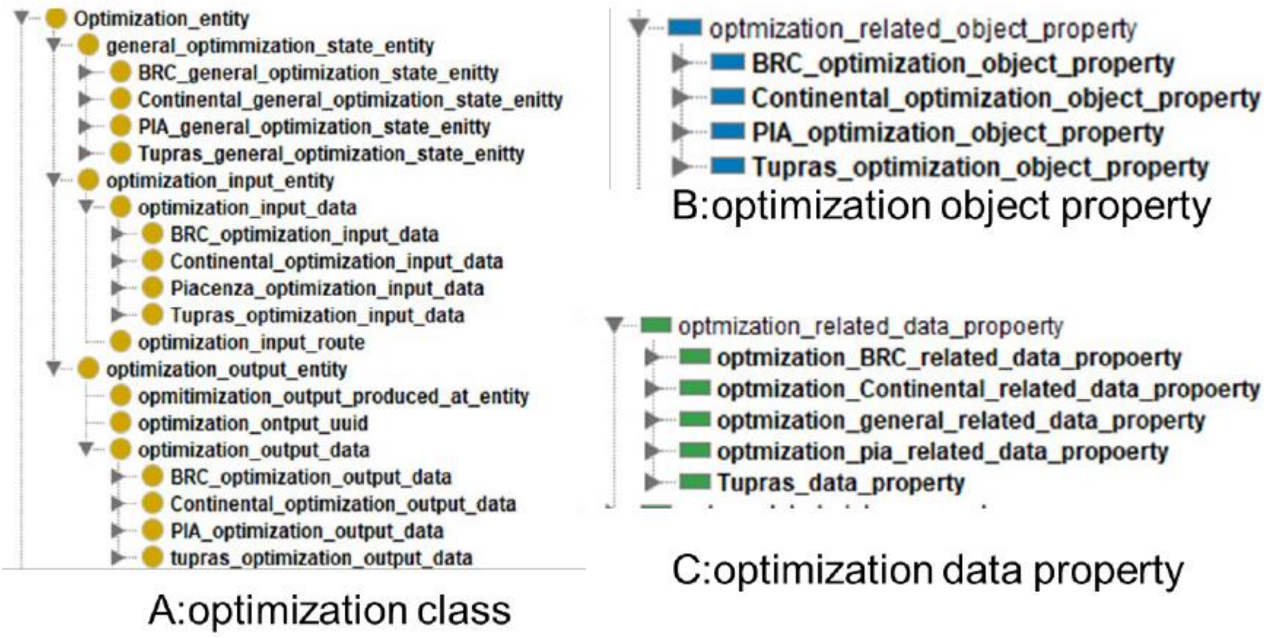

Figure 9: Ontology class, object property, and data property for optimisation

As shown in Figure 9, there are three key subclasses of optimization entity:

1. General optimization state entity
    - BRC general optimization state entity
        - BRC_order_related_entity
            - BRC_order_due_data
            - BRC_order_id
        - job_related_entity
            - job_id
            - job_parameter_id
            - job_parent_id
            - job_process_stage
            - job_process_time_duration
            - job_processing_times
        - machine_related_entity
            - completion_time
            - machine_id
            - machine_status
            - machine_type
            - parent_id
            - setup_time
            - start_time
            - status
    - CONT general optimization state entity
        - ProductBOM_Related_Entity
            - ProductBOMMultiplicity
        - LinesRelatedEntity
            - LineCode
            - LineDescription

    - LineId
  - LineTypesRelatedEntity
    - LineTypesCode
    - LineTypesDescription
    - LineTypesId
  - ProcessingTimesRelatedEntity
    - IdealProcessingTime
    - RealProcessingTime
  - ProductFamiliesRelatedEntity
    - ProductFamiliesDescription
    - ProductFamiliesId
    - ProductFamiliesName
  - ProductionOrdersRelatedEntity
    - ProductionOrderDueDate
    - ProductionOrderId
    - ProductionOrderMaxQuantity
    - ProductionOrderName
    - ProductionOrderPriority
    - ProductionOrderQuantity
  - productRelatedEntity
    - EfficiencyRate
    - productDescription
    - productionId
    - productName
  - ResourceBOMRelatedEntity
    - ResourceBOMMultiplicity
  - ResourcesRelatedEntity
    - ResourcesDescription
    - ResourcesId
    - ResourcesName
  - ScheduledMaintenanceActivitiesRelatedEntity
    - ScheduledMaintenanceActivitiesDuration
    - ScheduledMaintenanceActivitiesId
    - ScheduledMaintenanceActivitiesStart
  - ScheduleRelatedEntity
    - Scheduleend
    - Schedulestart
  - SetupTimesRelatedEntity
    - SetupTimeId
  - WorkplacesRelatedEntity
    - SequenceInLine
    - WorkplaceCode
    - WorkplaceDescription
    - WorkplaceId
  - WorkplaceTypesRelatedEntity
    - WorkplaceTypesCode
    - WorkplaceTypesDescription
    - WorkplaceTypesId
- PIA general optimization state entity

    - loom_order_sequence_related_entity
        - loom_order_sequence_order_sequence_entity
        - loom_order_sequnce_id
    - loom_related_entity
        - loom_ID
        - loom_speed
        - maintenance_end
        - maintenance_start
    - PIA_order_related_entity
        - PIA_order_ca
        - PIA_order_cc
        - PIA_order_chain_id
        - PIA_order_comb
        - PIA_order_comb_height
        - PIA_order_delivery_date
        - PIA_order_drawing
        - PIA_order_due_data
        - PIA_order_energy_consumption
        - PIA_order_fabric_type
        - PIA_order_id
        - PIA_order_incom
        - PIA_order_kStrokes
        - PIA_order_loom_id
        - PIA_order_part_id
        - PIA_order_priority_weight
        - PIA_order_process_end_time
        - PIA_order_process_start_time
        - PIA_order_processing_time
        - PIA_order_setup_end_time
        - PIA_order_setup_start_time
        - PIA_order_setup_time
        - PIA_order_status
        - PIA_order_strokesPerMt
        - PIA_order_tardiness
        - PIA_order_target_meters
        - PIA_order_type
        - PIA_order_variant
        - PIA_order_yarns
        - PIA_order_ybf
- Tupras general optimization state entity
    - InputFeeds_related_entity
        - IC2_i
        - IC5_i
        - IF_i
        - InputNodeID
        - ISU_i
    - linkScenarios_related_entity
        - CAP_ij
        - link_id_is

        - PC2_i_j_s
        - PC5_i_j_s
        - PF_i_j_s
        - PSU_i_j_s
    - optimization_input_Settings_related_entity
        - Horizon
        - PriceOfElectricity
        - PriceOfLPG
        - PriceOfNG
        - TimeToOptimize
    - optimization_input_specs_related_entity
        - C2
        - C2C5
        - C5
        - SU
    - optimization_output_SolutionScenariosRelatedEntity
        - OptID
        - Scenario_ID
        - SolutionScenariosNode_ID
    - Output_tanks_related_entity
        - OutputNodeID
        - Q_start_i
        - Q_total_i
        - QC2_start_i
        - QC5_start_i
        - QSU_start_i
    - UnitScenarios_related_entity
        - E_i_s
        - Unit_Scenarios_id_is
        - UnitScenarios_NodeID

2. Optimization input entity
    - optimization_input_data
        - BRC_optimization_input_data
            - optimization_input_lambda
        - CONT_optimization_input_data
            - CONT_optimization_input_dynamic_data
    - optimization_input_BreakDownEvents_data
    - optimization_input_ProductInitialInventory_data
    - optimization_input_ProductionOrders_data
    - optimization_input_ResourceInitialInventory_data
    - optimization_input_ScheduledMaintenanceActivities_data
    - optimization_input_UnScheduledMaintenanceActivities_data
            - CONT_optimization_input_static_data
    - optimization_input_Lines_data
    - optimization_input_LineTypes_data
    - optimization_input_ProductFamilies_data
    - optimization_input_Products_data
    - optimization_input_Resources_data
    - optimization_input_StorageZones_data

- optimization_input_Workplaces_data
- optimization_input_WorkplaceTypes_data
- optimization_ProcessingTimes_data
- optimization_ProductBOM_data
- optimization_ResourceBOM_data
- optimization_SetupTimes_data
    - PIA_optimization_input_data
        - optimization_input_currentTotalSetupTime_entity
        - optimization_input_historicalDataWeekly_entity
        - optimization_input_loom_entity
        - optimization_input_startDate_entity
        - optimization_input_workGroupsPerShift_entity
    - Tupras_optimization_input_data
        - optimization_input_OS
- linkedScenario
- UnitScenario
        - optimization_input_PI
- optimization_input_InputFeed_entity
- optimization_input_OutputTanks
- optimization_input_Settings
- optimization_input_Specs
        - optimization_input_PM
- optimization_input_route

3. Optimization output entity
- opmitimization_output_produced_at_entity
- optimization_ontput_uuid
- optimization_output_data
    - BRC_optimization_output_data
        - BRC_optimization_output_objective_values
- BRC_optimization_output_objective_Makespan
- BRC_optimization_output_objective_TotalLateness
- BRC_optimization_output_objective_TotalTardiness
        - BRC_optimization_output_output_job
        - BRC_optimization_output_output_job_completion_Time
        - BRC_optimization_output_output_job_start_Time
        - BRC_optimization_output_output_order
        - BRC_optimization_output_output_order_sequence
    - CONT_optimization_output_data
        - CONT_optimization_output_Metrics_data
- CONT_AvgWorkStationIdleTime
- CONT_Makespan
- CONT_TotalTardiness
        - CONT_optimization_output_Schedule
    - PIA_optimization_output_data
        - PIA_optimization_output_objective_values
- PIA_loom_order_sequence
- PIA_optimization_ouput_totalCompletionTimes
- PIA_optimization_ouput_totalEnergyCons

- PIA_optimization_ouput_totalTardiness
- PIA_optmization_output_makespan
    - Tupras_optimization_output_data
        - optimization_output_output_OutputKPIs
- optimization_output_output_C2C5perc
- optimization_output_output_C2perc
- optimization_output_output_C5perc
- optimization_output_output_OutputNodeID
- optimization_output_output_Quantity
- optimization_output_output_SUperc
        - optimization_output_output_SolKPIs
- optimization_output_output_found_solution
- optimization_output_output_TimeToInitializeMillisec
- optimization_output_output_TimeToSolveMillisec
        - optimization_output_output_total_energy
        - optimization_output_SolutionScenarios

Object properties are used to define the interrelationships among ontology classes of optimization:

1. BRC_optimization_object_property
    - optimization_input_include_machine
    - optimization_input_include_order
    - optimization_input_include_route
    - optimization_input_inlcude_lambda
    - optimization_ouput_have_job
    - optimization_ouput_have_output_order
    - optimization_output_machine_job_id
    - optimization_output_Objective
    - optimization_output_order_have_output_job
    - optimization_output_output_job_have_parent_id
    - optimization_output_output_job_have_start_time
    - optimization_output_output_machine
    - optimization_produced_at
    - optmization_output_job_have_machine
    - optmization_output_output_have_complete_time
2. CONT_optimization_object_property
    - optimization_cont_input_has_static_data
    - optimization_cont_input_include_rounte
    - optimization_cont_input_line_related_object_property
        - line_has_code
        - line_has_description
        - line_has_id
        - line_has_linetypeId
    - optimization_cont_input_line_type_related_object_property
        - linetype_has_code
        - linetype_has_description
        - linetype_has_id
    - optimization_cont_input_product_object_property

  - optimization_input_product_has__efficiency_rate
  - optimization_input_product_has_description
  - optimization_input_product_has_id
  - optimization_input_product_has_name
  - optimization_input_product_have_product_end_line_id
  - optimization_input_product_have_product_familiy_id
  - optimization_input_product_have_product_source_line_id
- optimization_cont_input_productBOM_related_entity
  - optimization_cont_input_productBOM_has_multiplicity
  - optimization_cont_input_productBOM_has_productid
  - optimization_cont_input_productBOM_has_requiredProductsId
  - optimization_cont_input_productBOM_has_workplacesid
- optimization_cont_input_resource_Bom_related_entity
  - optimization_input_resource_bom_has_Multiplicity
  - optimization_input_resource_bom_has_products_id
  - optimization_input_resource_bom_has_resource_id
  - optimization_input_resource_bom_has_workplaceid
- optimization_cont_input_scheduleMaintenanceActivity_has
- optimization_cont_input_workplace_type_related_object_property
  - workspace_type_has_code
  - workspace_type_has_description
  - workspace_type_has_id
- optimization_cont_input_workspace_related_entity
  - optimization_cont_input_workspace_has_code
  - optimization_cont_input_workspace_has_description
  - optimization_cont_input_workspace_has_id
  - optimization_cont_input_workspace_has_lineId
  - optimization_cont_input_workspace_has_SequnceLine
  - optimization_cont_input_workspace_has_worplacetypeId
- optimization_cont_output_has_metrics
- optimization_cont_output_has_schedule
- optmization_cont_input_has_dynamic_data
- optmization_cont_input_processing_time_related_object_property
  - optimization_input_processingtime_has_ideal_processing_time
  - optimization_input_processingtime_has_product_id
  - optimization_input_processingtime_has_real_processing_time
  - optimization_input_processingtime_has_workpalcetype
- optmization_cont_input_product_families_object_properety
  - product_families_has_description
  - product_families_has_id
  - product_families_has_name
- optmization_cont_input_production_order_has
- optmization_cont_input_resource_related_object_property
  - optimization_input_resource_has_description
  - optimization_input_resource_has_id
  - optimization_input_resource_has_name
- optmization_cont_input_setup_time_related_object_property
  - optmization_input_setup_time_has_linetypeid
  - optmization_input_setup_time_has_productfamiliesId

    - optmization_input_setup_time_has_setuptime_id
- optmization_cont_output_has_uuid
- optmization_cont_output_product_at
- optmization_cont_output_schedule_has

3. PIA_optimization_object_property
    - opimization_pia_output_include_looms_order_sequence
    - optimization_pia_input_include_currentTotalSetupTime
    - optimization_pia_input_include_historicalDataWeekly
    - optimization_pia_input_include_looms
    - optimization_pia_input_include_order
    - optimization_pia_input_include_route
    - optimization_pia_input_loom_include_entity
    - optimization_pia_output_include_order
    - optimization_pia_output_include_uuid
    - optimization_pia_output_loom_order_sequence_include_id
    - optimization_pia_output_loom_order_sequence_include_sequence_entity
    - optimization_pia_output_produce_at
    - optmization_pia_input_include_startDate
    - optmization_pia_input_include_workGroupsPerShift
    - optmization_pia_input_order_include_entity
    - optmization_pia_output_include_include_total_EnergyCons
    - optmization_pia_output_include_makespan
    - optmization_pia_output_include_total_compeletionTime
    - optmization_pia_output_include_totalTardiness
    - optmization_pia_output_order_include_entity
4. Tupras_optimization_object_property
    - optimization_input__unit_scenario_include_entity
    - optimization_input_include_OS
    - optimization_input_include_PM
    - optimization_input_InputFeeds_including_entity
    - optimization_input_likScenario_including_entity
    - optimization_input_OS_inlucde_unit_scenario
    - optimization_input_output_task_include_entity
    - optimization_input_PI_include_output_task
    - optimization_input_PI_include_settings
    - optimization_input_PI_include_specs
    - optimization_input_setting_include_entity
    - optimization_input_specification_include_entity
    - optimization_input_unit_scenario_including_linkScenario
    - optimization_output_including_output_kpi_entity
    - optimization_output_including_SolKPIs
    - optimization_output_including_solution_scenarios
    - optimization_output_including_total_energy
    - optmization_inpput_including_route
    - optmization_input_including_PI
    - optmization_input_PI_includ_inputFeed
    - optmization_output_including_produced_at_entity

- optmization_output_including_uuid
- optmization_output_solution_scenario_including_NodeID
- optmization_output_solution_scenario_including_OptID
- optmization_output_solution_scenario_including_ScenarioID

Data properties are used to define the interrelationships among ontology individuals and their values:

1. optmization_BRC_related_data_property
    - optimization_output_job_completion_time_sequence_id_value
    - optimization_output_job_completion_time_value
    - optimization_output_job_start_time_sequence_id_value
    - optimization_output_job_start_time_value
    - optimization_output_TotalLateness
    - optimization_output_TotalTardiness
    - optimzation_input_lambda_value
    - optmization_ouput_job_parennt_id_value
    - output_order_list_element_id
    - OutputOrder_type
2. optmization_CONT_related_data_property
    - AvgWorkStationIdleTime_value
    - producted_entity_value
    - schedule_end_value
    - schedule_start_value
    - ScheduledMaintenanceActivitiesRelatedDataProperty
        - ScheduledMaintenanceActivities_duration_value
        - ScheduledMaintenanceActivities_Id_value
        - ScheduledMaintenanceActivities_start_value
    - TotalTardiness_value
3. optmization_general_related_data_property
    - optimization_output_produced_at_value
    - optimization_output_uuid_value
    - optmization_input_route_value
    - optmization_output_Makespan
4. optmization_pia_related_data_property
    - optimization_input_start_data_value
    - optimization_output_llom_order_sequence_content_value
    - optimization_output_llom_order_sequence_entity_id_value
    - optimization_output_llom_order_sequence_id_value
    - optimization_output_makespan
    - optimization_output_total_completion_times
    - optimization_output_total_energycons_value
    - optimization_output_total_tardiness_value
    - optmization_input_currentTotalSetupTime_value
    - optmization_input_historicalDataWeekly_value
    - optmization_input_workGroupsPerShift_value
5. Optimization_Tupras_data_property
    - TotalEnergy_value

- SU_value
- ScenarioID_i_s_value
- QSU_start_i_value
- QC5_start_i_value
- QC2_start_i_value
- Q_total_i_value
- Q_start_i_value
- PSU_i_j_s_value
- PF_i_j_s_value
- PC5_i_j_s_value
- PC2_i_j_s_value
- OptID_value
- ISU_i_value
- LinkID_value
- InputNodeID_value
- IF_i_value
- IC5_i_value
- IC2_i_value
- Horizon_value
- FoundSolution_value
- E_i_s_value
- CAP_ij_value
- C5_value
- C2C5_value
- C2_value

3.7 Ontology for Process Modelling

There are two types of process models including PN models and discrete event simulation models.

3.7.1 PN Models

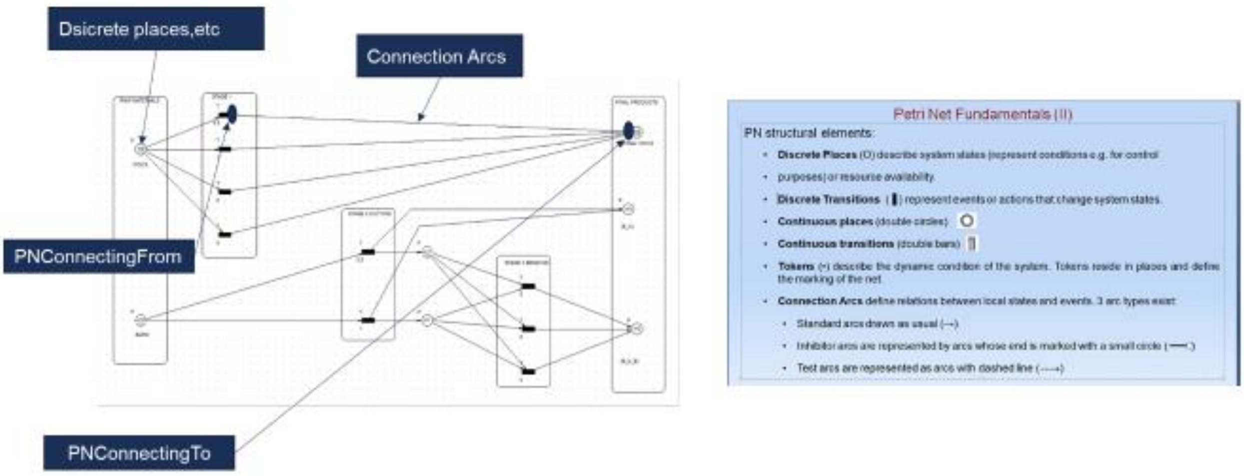

Figure 10: PN models for BRC pilot

An example of PN model is demonstrated in Figure 10. It includes several key nodes and connections:

1. Nodes
   - Discrete place
   - Discrete transitions
   - Continuous places
   - Continuous transitions
2. Connection Arcs
   - Standard arcs
   - Inhibitor arcs
   - Test arcs
3. General PN elements
   - Tokens
   - PN result

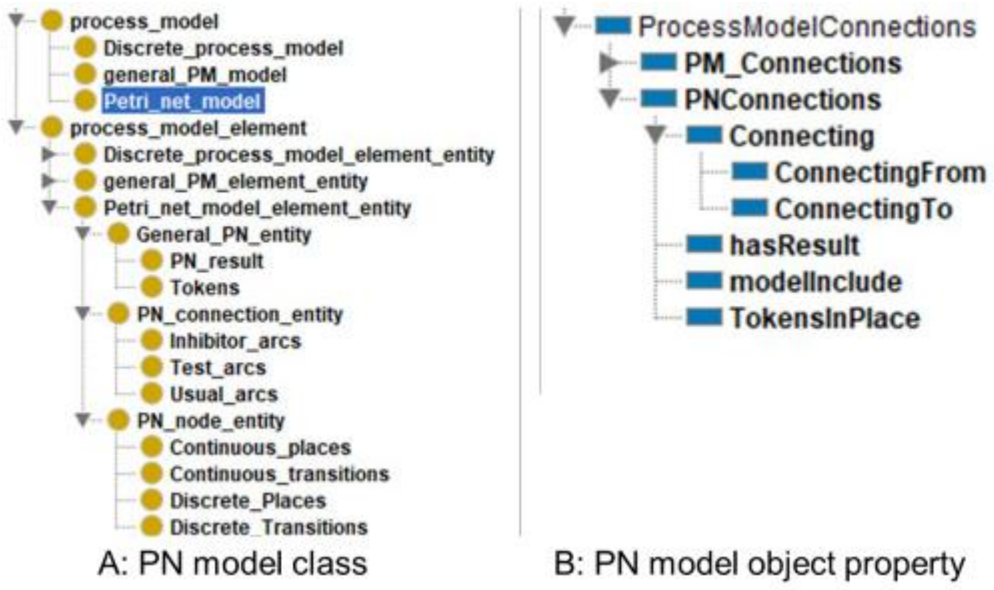

Figure 11: PN model class and properties

As shown in Figure 11, ontology class and object properties for PN models are:

1. Process_model_entity
   - Petri_net_model
2. Petri_net_model_element_entity
   - General_PN_entity
     - PN_result
     - Tokens
   - PN_connection_entity
     - Inhibitor_arcs
     - Test_arcs
     - Usual_arcs
   - PN_node_entity
     - Continuous_places
     - Continuous_transitions

- Discrete_Places
- Discrete_Transitions

Object properties are used to demonstrate the interrelationships among PN models classes:

1. ProcessModelConnections
   - PNConnections
     - PNConnecting
       - PNConnectingFrom
       - PNConnectingTo
     - hasResult
     - modelInclude
     - TokensInPlace

3.7.2 Discrete event simulation models

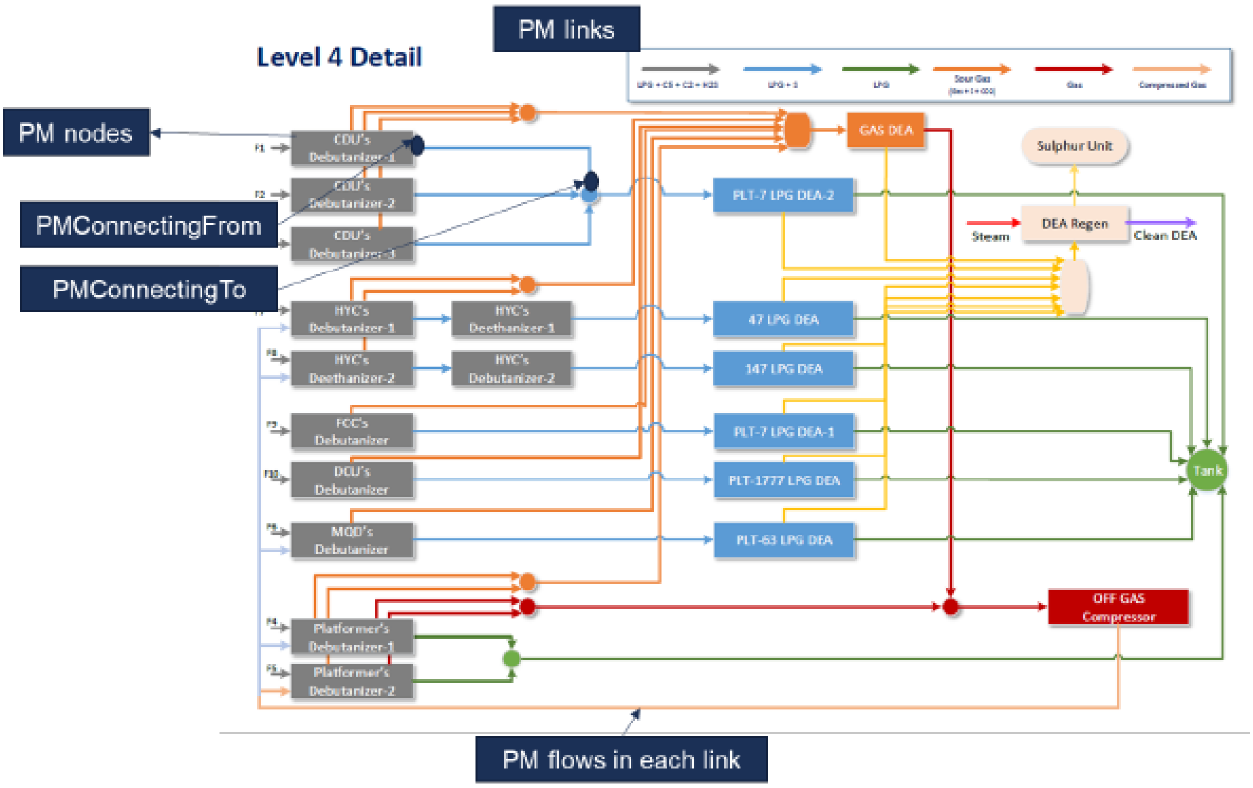

Figure 12: PM (Process Modelling) Discrete event simulation model

PM discrete event simulation model is used in the TUPRAS pilot for simulating the process for oil industry. In order to make of ontology to represent the discrete event simulation model which is considered as process model, as shown in Figure 12, several ontology classes, object property and data properties are introduced in Figure 13:

Figure 13: PM Discrete event simulation model class, object property and data property

1. Process_model_entity
    - process_model
        - Discrete_event_simulation_model
    - process_model_element
        - PM_element_entity
            - PM_flow_related_entity
                - flow_caculated
                - flow_factor
                - flow_formula
                - flow_manual
                - flow_name
                - flow_quantity
            - PM_link_related_entity
                - PM_link_description
                - PM_link_id
                - PM_link_name
                - PM_link_source
                - PM_link_target
            - PM_node_related_entity
                - PM_node_description
                - PM_node_id
                - PM_node_name
                - PM_node_one2one
                - PM_node_parameter_entity
                - PM_node_script_source
                - PM_node_solved
                - PM_node_specificationMethod

        - PM_node_stage
        - PM_node_stock
        - PM_node_type
      - PM_parameter_related_entity
        - PM_parameter_description
        - PM_parameter_Symbol
        - PM_parameter_unit
        - PM_parameter_value
      - PM_related_entity
        - PM_id
        - PM_Name
      - PM_resource_related_entity
        - PM_resource_description
        - PM_resource_id
        - PM_resource_name
        - PM_resource_unit
    - PM_element_related_entity
      - PM_link
        - Clean_DEA_flow
        - Compressed_Gas_flow
        - Gas_flow
        - LPG_C5_C2_H2S_flow
        - LPG_CS_C2_H2S
        - LPG_flow
        - LPG_S_flow
        - SourGas_flow
        - Steam
      - PM_node
        - DEA_Regen_component
        - Debutanizer_component
        - LPG_component
        - LPG_DEA_component
        - Merge_component
        - Off_Gas_compressor_component
        - SulphUnit_component
        - Sulphur_Unit_component
        - Tank_component
      - PM_parameter
      - PM_resource
      - PM_results
      - PM_Tolerance

Object properties are used to represent the interrelationships among PM classes:

1. ProcessModelConnections
    - PM_Connections
      - PM_link_connected_to_flow
      - PM_flow_including
      - PM_has_id
      - PM_has_links
      - PM_has_name

- PM_has_node
- PM_has_parameter
- PM_has_tolerance
- PM_link_includding
- PM_node_including
- PM_parameter_including
- PM_resource_including
- PMConnectingFrom
- PMConnectingTo
- PMhasResult

Data properties are used to define the values of each individual of the PM classes:

1. process_model_data_property
    - PM_data_property
        - Description_value
        - link_description_value
        - link_flow_calculated_value
        - link_flow_factor_value
        - link_flow_formula_value
        - link_flow_manual_value
        - link_flow_name_value
        - link_flow_quantity_value
        - link_id_value
        - link_name_value
        - link_source_value
        - link_target_value
        - node_onetoone_value
        - node_ScriptSource_value
        - node_stock_value
        - node_type_value
        - PM_Id_value
        - PM_name_value
        - PM_node_description_value
        - PM_node_id_value
        - PM_node_name_value
        - PM_node_stage_value
        - PM_resource_description__value
        - PM_resource_id__value
        - PM_resource_name__value
        - PM_resource_unit_value
        - PM_solved_value
        - PM_specMethod
        - Symbol_value
        - tolerance_value
        - Unit_value
        - Value_value

3.8 Ontology for Data Analysis

Data analysis ontology is used to define data analysis issues for TUPRAS pilot from partner JSI. The target of this ontology is to describe sensors (sensor id and sensor name), sensor data, pilot parameters (target if and control id), and a connection between units (unit A goes to Unit B). This ontology model is used for JSI data analysis module to understand the information related to the data analysis.

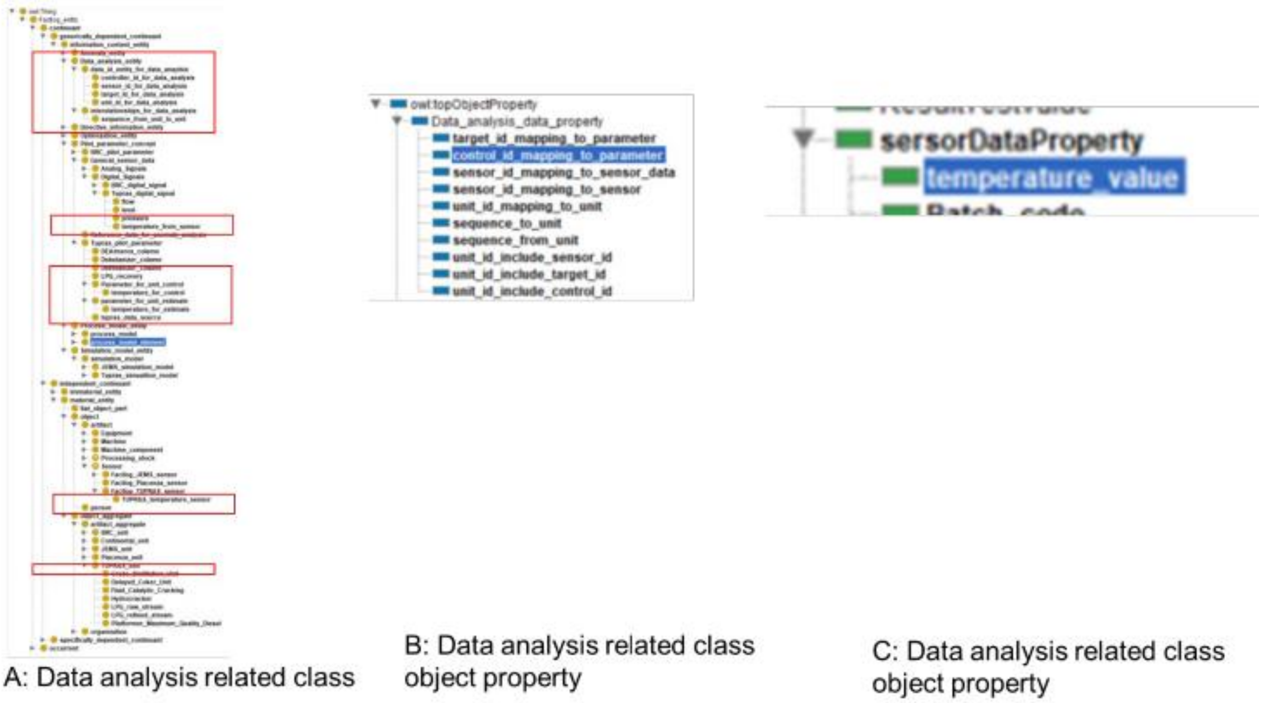

Figure 14: Data analysis class, object property and data property

Combined with ontology for pilot description, data analysis class enables to represent the requirements from JSI for describing the data analysis issues. Thus, the specific data analysis classes include:

1. Data_analysis_entity
    - data_id_entity_for_data_anaylsis
        - controller_id_for_data_analysis
        - sensor_id_for_data_analysis
        - target_Id_for_data_analysis
        - unit_Id_for_data_analysis
    - interelationships_for_data_analysis
        - sequence_from_unit_to_unit

3.9 Ontology for Anomaly Detection

The ontology concepts are defined to support anomaly detection in FACTLOG pilots.

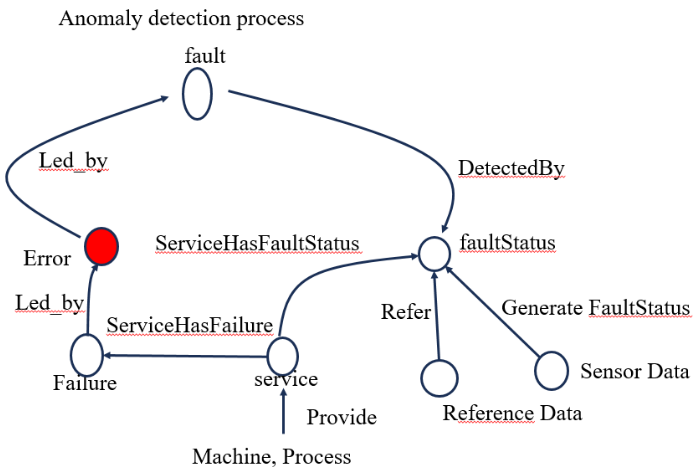

Figure 15: Ontology for describing anomaly detection processes

As shown in Figure 15, several concepts are defined to describe anomaly detection process.

1. A service is defined as which is delivered when the service implements the system function.
2. A service failure, often abbreviated here to failure, is an event that occurs when the delivered service deviates from correct service.
3. An error is the deviation that at least one (or more) external state of the system deviates from the correct service state, since a service is a sequence of the system's external states.
4. A fault is an adjudged or hypothesized cause of an error.
5. An observation of a fault is fault status.
6. Sensor data refers to the data directly generated from machine or sensor.
7. Reference data refers to the data from simulation models for decision-makings.

Figure 16: Ontology classes for describing anomaly detection processes

As shown in Figure 16, ontology classes are defined based on the Figure 15.

3.10 Ontology for Simulation Model

The ontology in the simulation model entity is used to represent the simulation models used in the FACTLOG pilots.

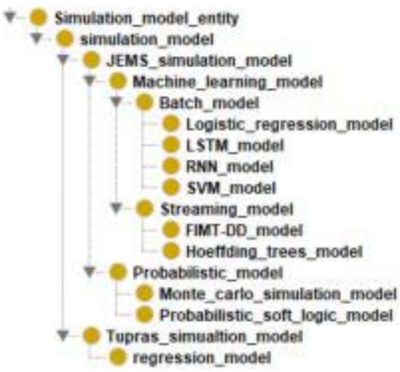

Figure 17: Ontology class for simulation model

As shown in Figure 17, ontology classes are defined as following:

1. Probabilistic_model
    - Probabilistic_soft_logic_model
    - Monte_carlo_simulation_model
2. Machine_learning_model
    - Streaming_model
    - FIMT-DD_model
    - Hoeffding_trees_model
3. Batch_model

  - LSTM_model
  - RNN_model
  - Logistic_regression_model
4. SVM_model
5. Regression model

4 Knowledge Graphs Models for Unified Pilot Description

4.1 BRC pilot

BRC is the largest supplier of steel reinforcement in the UK, producing bespoke products for the construction industry. They produce batches of steel bars made to customer specification. Each job requires a batch of steel to be produced in a certain shape – everything from simple straight bar to complex 3D shapes. The process involves cutting and shaping various diameters of steel reinforcing bar using various manual or automatic operations. The bending machines work with great forces and under great stress and do break down. Having a system that would supervise its operation and detect upcoming potential malfunctions would help avoid unnecessary down-time. Besides that, there is a need to improve management of the batches of steel in the production floor. There are two scenarios in the BRC pilot.

4.1.1 Pilot scenarios

Scenario #1: Machine monitoring

<u>Description</u>: The hydraulic bending machine works with immense pressure and is prone to breakage. The problem is, there is no way of foreseeing the upcoming mechanical failure until it occurs as the machine offers little insight into its operational status. The plan to overcome this is to install a sensory layer between the machine control panel and the machine itself to be able to capture the signals going to and from the machine. A system is needed that will predict the likelihood of breakage from the dynamics of these values.

<u>Analytics approach</u>: Similar as in the other cases this is an anomaly detection problem. Normal operational values need to be identified for different machine jobs and settings so that alerts may be raised when machine status deviates significantly. The challenge in this pilot lies in the fact that the machine sensor array is still being assembled and there is no existing data to learn from. Therefore, more attention will need to be given whether to proceed with the more general approach of modelling the machine state with a generative model or perhaps to use more targeted stream classification models by incorporating more expert knowledge into the models. Nevertheless, this does not change the technical requirements of the pilot.

Scenario #2: Production scheduling and crane operation

<u>Description</u>: The steel rebar is processed in the BRC factory floor in batches. The batches are moved around using cranes when loading/unloading the materials to/from the machines. To achieve optimal plant operation, the jobs on the machines have to be planned together with the crane movements. This way, there is minimal waiting and new materials are provided to the machines when they finish previous jobs and the crane is available to move the processed batches away from the machines for storage or shipment.

<u>Analytics approach</u>: This is an optimisation problem where the space of possible plans needs to be searched to find the optimal one. The optimization component for the BRC case, depends upon parameters that are input data to be derived from analytics. As discussed in D1.1, the role of optimization in this pilot case is to provide solutions for BRC's complex multistage flowshop problem. In order for optimization to be able to derive to an optimal production schedule that takes under consideration raw materials, crane

movement and machine maintenance, the analytics should provide indications with respect to, productions times, anomalies detection relevant to the machines' availability and schedules of maintenance. More precisely, the analytics should provide inputs with respect to operation and set up times for each product type in every production step. Such estimations could be easily derived for some products (for instance products with shape code C1-98) but much trickier for others (for instance products with shape code C99). Regression models predicting the operation times based on the machine state and product specifications (materials, shape…) can be used.

Additionally, when operating, different detected anomalies in the involved machines will have to be able to inform the optimization module for a potential problem (e.g. underperformance based on currently produced batch). Lastly and in relation to the cranes detected anomalies in operation (e.g. availability, movement based on production schedule etc.) will also have to be identified in order to also inform the optimizer. These anomalies are outputs of the anomaly detection system described in scenario #1.

4.1.2 Data types and sources

Currently, the BRC production floor is not fully equipped for the application of digital twin technology for the scenarios described above. The process still needs to be equipped with sensors and some of it needs to be digitalised. These adaptations are being performed now in the scope of the project, including installing a sensory layer into the machine control board.

There are several different data sources in the BRC pilot:

- **Production data**: Production data that is stored is stored and utilised in the MES system however reports can be derived in excel
- **Planning data**: Planning data from the MES system using its existing optimisation based on machine providers production estimates and rules of optimisation we set. It is exported then to excel
- **Barmark data**: Data generated once entered from customer schedules into MES and exported by search query into excel
- **Transport data**: Mostly generated from the planning sheet however is then moved into a separate excel and the data is then moved in excel to generate loads
- **Machine capability**: Produced from experience and machine handbooks
- **Stock data**: Currently done by stock processor who records in excel
- **Machine sensor system**: Produced form new monitoring systems on the machine and data fed into a PLC then database
- **Crane sensor system**: Newly fitted system to measure distance of long travel and cross travel that's fed into a database
- **Scan data**: Dependant on existing MES System doing extra scanning and timestamping or if new app system for scanners needs to be created to work alongside
- **Machine PPM schedule**: Currently done manually and in existing Microsoft applications new system needs creating to work with FACTLOG system

The data parameters with their meta-information, including data types, data availability and data sources, are listed in D2.1. Note that though the majority of the values are numeric or binary, some of the parameters are less structured. A representative example of these is the parameter "Instructions for transport", collected from the planning sheet. Such values will need to be manually inspected by a data science expert to extract features appropriate for processing in machine learning algorithms. Most likely, manual transformations will be sufficient as there are only a few such values. Automatic feature generation approaches will be used if they prove necessary. For example, term frequency measures such as TF-IDF can be used to identify important phrases in the instruction texts and indicator features can be generated for those.

4.1.3 Knowledge Graph modelling

4.1.3.1 Scenario description

In the BRC pilot, four main concerns are first considered when knowledge graph models are developed to formalize the BRC pilot: 1) scenario description; 2) PN model formalism; 3) optimization formalism; 4) anomaly detection (introduced in Section 5.1.2).

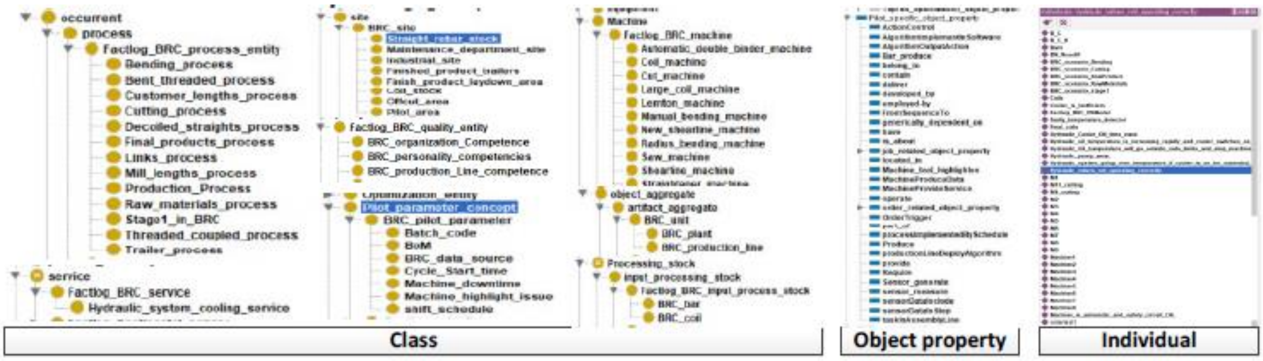

Figure 18: Ontology class, object property and individuals for BRC pilot

As shown in Figure 18, process entities, services, sites, pilot parameter, machine, unit and input process stock are the main ontology concepts defined in the knowledge graph models. General specific object properties are defined in order to support all the pilot description. Individuals are defined to describe an example of the BRC scenario.

4.1.3.2 PN model formalism

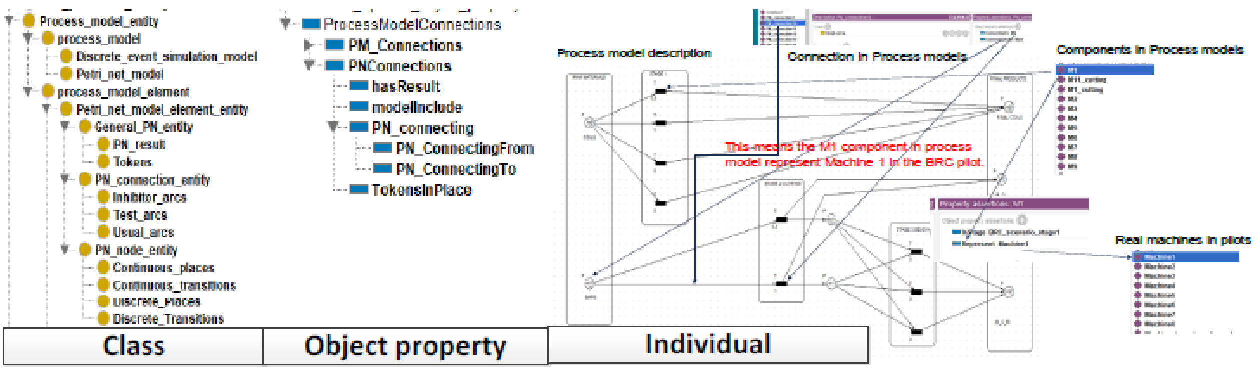

Figure 19: Ontology class, object property and individuals for simulation in BRC pilot

As shown in Figure 19, classes of petri net model entities and object properties of petri net model connections are defined to describe the petri net model used in the BRC pilot. The PN node entities are used to represent the "model compositions" in the petri net model. The PN connection entities are used to represent the model connections in the petri net model. PN_connectingFrom and PN_connectingTo are used to connect the PN connections and PN nodes. Through this way, the entire PN model is described.

4.1.3.3 Optimization formalism

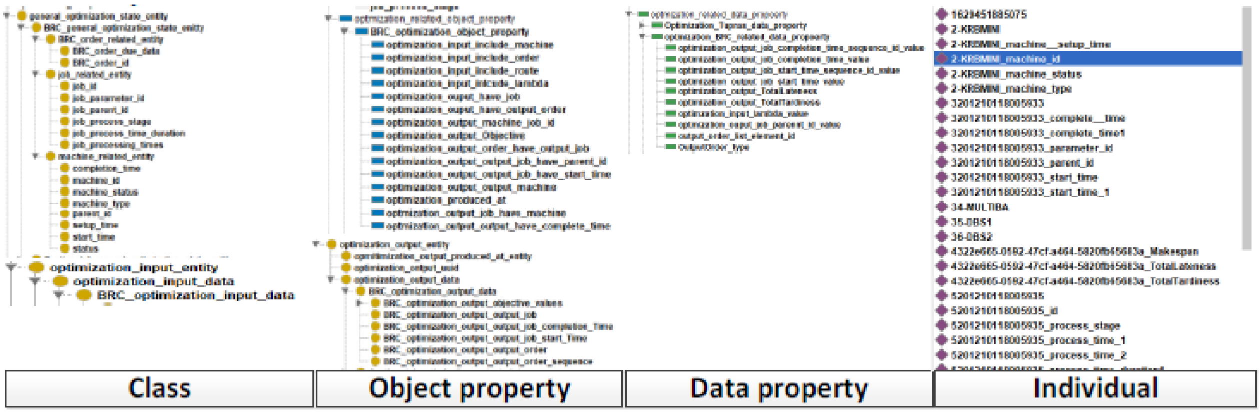

Figure 20: Ontology class, object property, data property and individuals for optimization for BRC pilot

As shown in Figure 20, optimization classes, object properties and data properties are defined to represent optimization scenario using individuals. In Figure 21, the individuals are used to represent the optimization input data structure. The BRC optimization input has a machine named “2KRBMINI” who has a machine id as 2-KRBMINI.

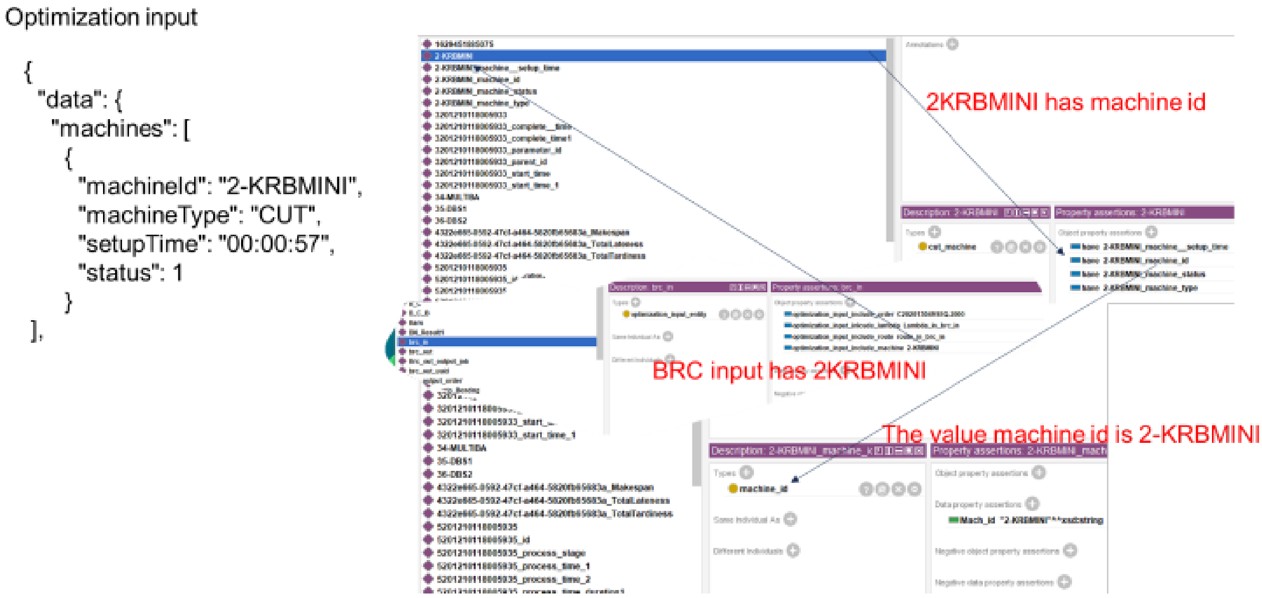

Figure 21: Individuals for describing the optimization scenario for BRC pilot

4.2 TUPRAS pilot

The TUPRAS oil refinery processes raw oil into several petroleum products such as Liquid Petroleum Gas (LPG), naphtha, gasoline, diesel and fuel oil. Within FACTLOG, the focus is on production of LPG, which is the final output of a complex set of interconnected processes. The main problem is how to achieve the proper ingredient constitution of the gas, making sure the impurities and individual components are within legally set limits (chief among them being the sulphur content). The core idea is to detect possible trends and anomalies of the ingredient constitution in the early phases to minimise the impact in the final output tank.

4.2.1 Pilot scenarios

Scenario #1: New Data Streams Monitoring and Storing

Description: LPG is the output of several different subsystems in the oil refinery, which means the system needs to collect, store and process a wide variety of data. While this is primarily a knowledge management problem where data needs to be transformed into a common format and validated, analytics can have a role in the cleaning and pre-processing.

Analytics approach: This scenario covers the very initial stages of analytics work, including data cleaning and preparation. The key requirement is to provide quality input for later processing. This includes data resampling and interpolation when sensors fail to report readings with reliable frequency. Some feature selection could also be performed at this point, using standard machine learning feature evaluation metrics to determine which data streams to focus on.

Scenario #2: Anomaly detection

Description: As the LPG production process is running, LPG is produced in several different sub-processes in subsystems of the refinery and is then collected into a common output pool. LPG quality is tested on samples from this final pool. Since this is a laboratory test which takes time, it would be highly beneficial if bad trends and anomalies would be detected earlier in the pipeline (and in time).

Analytics approach: This is an anomaly detection problem which can be tackled with a general approach of building a generative model of the process from historic data and then comparing the sensor readings to the model predicted values. When significant discrepancies are detected, an alert is raised. The generative model can either be based on typical value distributions of the sensor values; a set of regression models for individual stages of the pipeline (e.g. neural networks); process models built by experts or a combination of all three. Since the allowed limit values are known for all the impurities, a classification model could be built for each, predicting for some time horizon ahead that the limit value is going to be exceeded.

Scenario #3: Impact assessment and optimized intervention (LPG quality)

Description: As already described above in the description of scenario #2, the final LPG product is a mix of several different subsystem outputs. Once an anomaly is detected or off-specs product is predicted, the best way of remedying the problem needs to be determined. This includes identifying the relevant parts of the process/refinery and estimating the impact of these parts. The optimal intervention can then be found and performed.

Analytics approach: This is a root-cause identification and an optimisation problem. The generative model of the pipeline, introduced in the approach to scenario #2, can be used to explore the impact of interaction with different parts of the pipelines. By exploring the space of possible interactions, the most impactful and effective interactions can be identified.

4.2.2 Data types and sources

The Tupras data comes from their oil refinery in Izmit, Turkey. The refinery is equipped with a wide array of sensors with different properties. Here we provide an overview over the main parts of the pipeline and the different types of sensors.

The crude oil is fed into crude distillation units (CDUs), which have the following main elements (described in detail in D2.1):

- Debutanizer column: removes butane
- Deetanizer column: removes ethane
- DEA/merox column: removes sulphur and mercaptans
- LPG recovery: recovers leftover LPG from by-products

The main types of sensors/values for these elements are:

- Temperature
    - Description (nature of data): temperature of the related stream
    - Measurement device: Temperature Transmitter / Thermocouple
    - Frequency: 1 data point / sec
    - Unit: °C
    - Range: 20 - 200 °C
- Flow
    - Description (nature of data): flow of the related stream
    - Measurement device: Flowmeter
    - Frequency: 1 data point / sec
    - Unit: m3/h
    - Range: 0 - 3000 m3/h
- Pressure
    - Description (nature of data): pressure of the related stream/ specific location of the column
    - Measurement device: Pressure Transmitter
    - Frequency: 1 data point / sec
    - Unit: kg/cm2
    - Range: 0 - 20 kg/cm2
- Level
    - Description (nature of data): level of capacity reached
    - Measurement device: Level sensor
    - Frequency: 1 data point / sec
    - Unit: %
    - Range: 0 - 100%

The LPG is then collected in the collection tanks which are equipped with the same sensors but the typical values have different ranges:

- Temperature: 0 - 40 °C
- Flow: -500 - +500 m3/h

- Pressure: 0 - 8.5 kg/cm2
- Level: 0 - 20 m

The chemical composition of the LPG and the levels of various impurities are measured by online analysers at different points in the process and in a laboratory using a gas chromatograph at the collection tank. The ranges of these readings differ depending on what impurity they are testing for:

- Sulphur: 0 - 250 mg/kg
- Butane: 0 - 200 %(mol/mol)
- Ethane: 0 - 300 %(mol/mol)
- Diene: 0 - 6 %(mol/mol)

The frequency of lab tests differs between the units from weekly tests in the final tank to daily in sulphur related units but are typically not performed more than once per day at best.

At least two years of historic data is available at the frequency of 1 data point/sec (for non-laboratory values). All values are real-valued numbers which are well suited for machine learning algorithms. The volume of data is very large and should be sufficient for analyses. In case the volume should prove hard to process the data can be down-sampled.

4.2.3 Knowledge Graph modelling

In the TUPRAS pilot, four main concerns are first considered when knowledge graph models are developed to formalize the TUPRAS pilot: 1) scenario description; 2) PM model formalism; 3) optimization formalism; 4) Data analysis formalism.

4.2.3.1 Scenario description

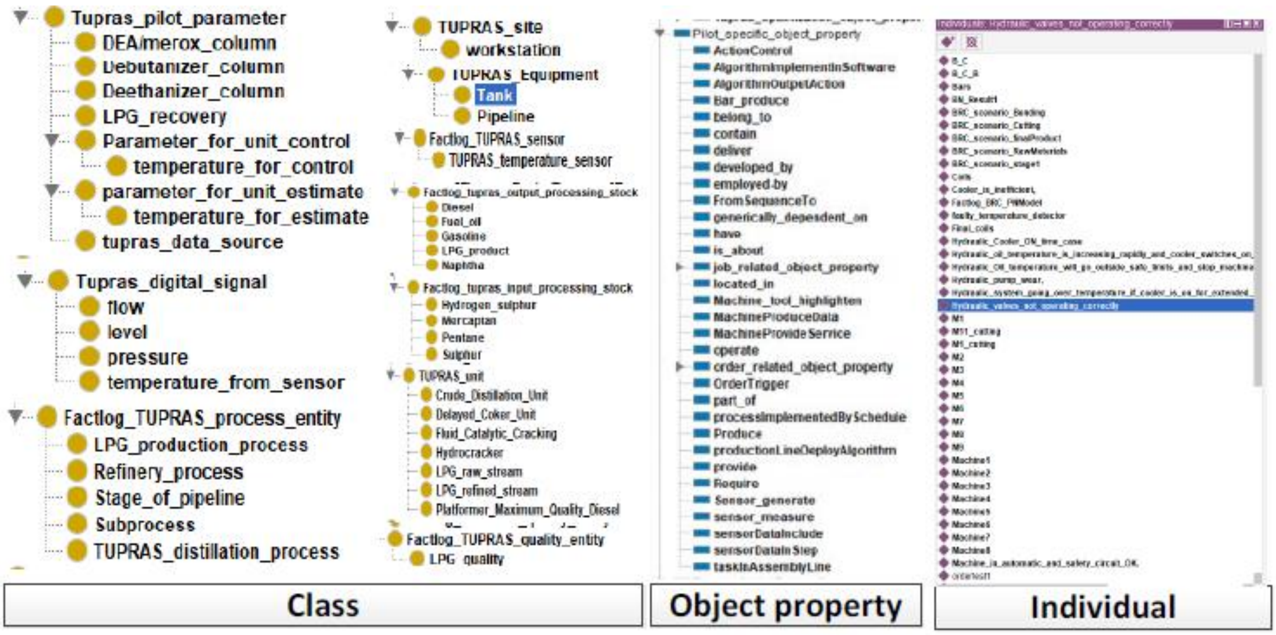

Figure 22: Ontology class, object property, data property and individuals for scenario description in TUPRAS pilot

As shown in Figure 22, TUPRAS process entities, services, pilot parameter, unit, TUPRAS quality, digital signal, site, input process stock and output process stock are the main ontology concepts defined in the knowledge graph models. General specific object properties are defined in order to support all the pilot description. Individuals are defined to describe an example of the TUPRAS scenario.

4.2.3.2 PN model formalism

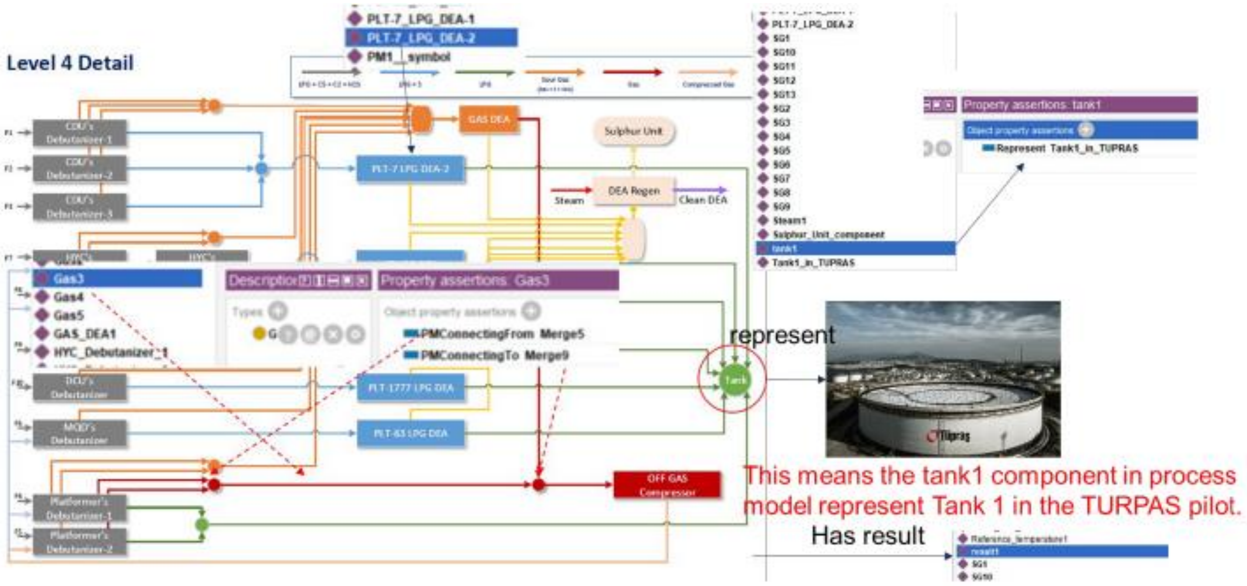

Figure 23: Individuals for process model description for TUPRAS pilot

Using the ontology in Section 3.7, the Figure 23 shows the individuals which are used to represent model structure for TUPRAS. For example, the individuals of PM_links are used to represent the connections among PM_nodes. The individuals of PM_nodes are used to represent each model composition in the model. The model composition is defined to represent the real equipment in the TUPRAS pilot.

4.2.3.3 Optimization formalism

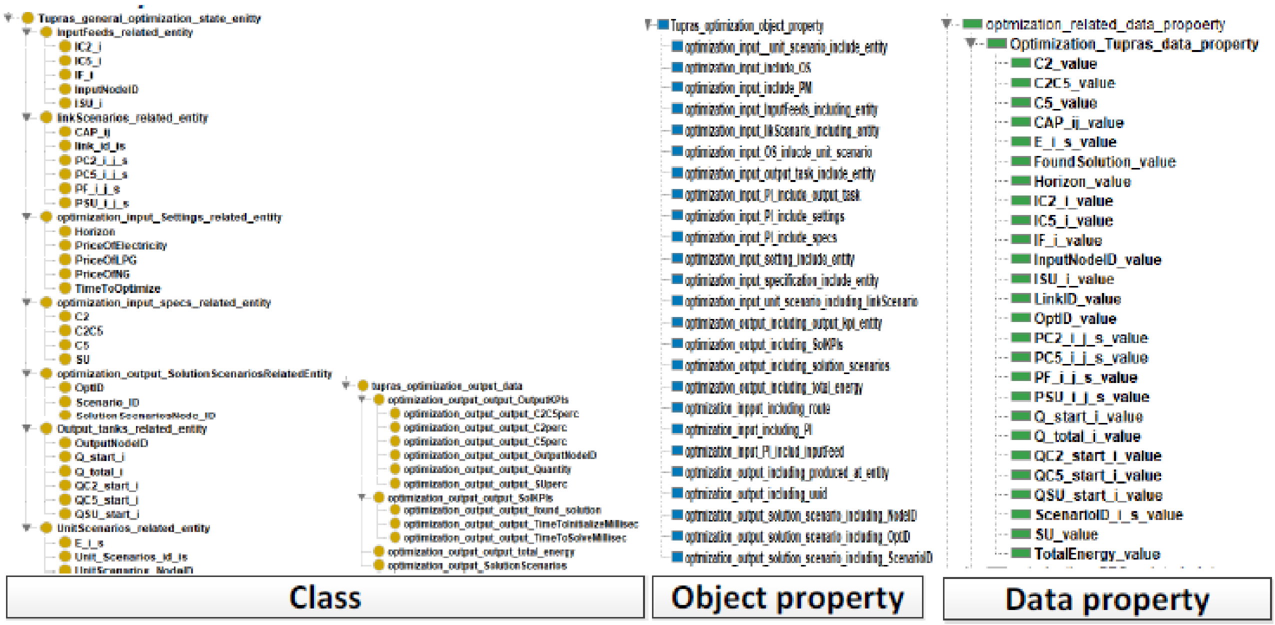

Figure 24: Ontology class, object property, and data property for optimization for TUPRAS pilot

As shown in Figure 24, optimization classes, object properties and data properties are defined to represent optimization scenario using individuals. In Figure 25, the individuals are used to represent the optimization output data structure. The TUPRAS optimization output has a solKPIs, solution scenario, outputKPIs and etc.

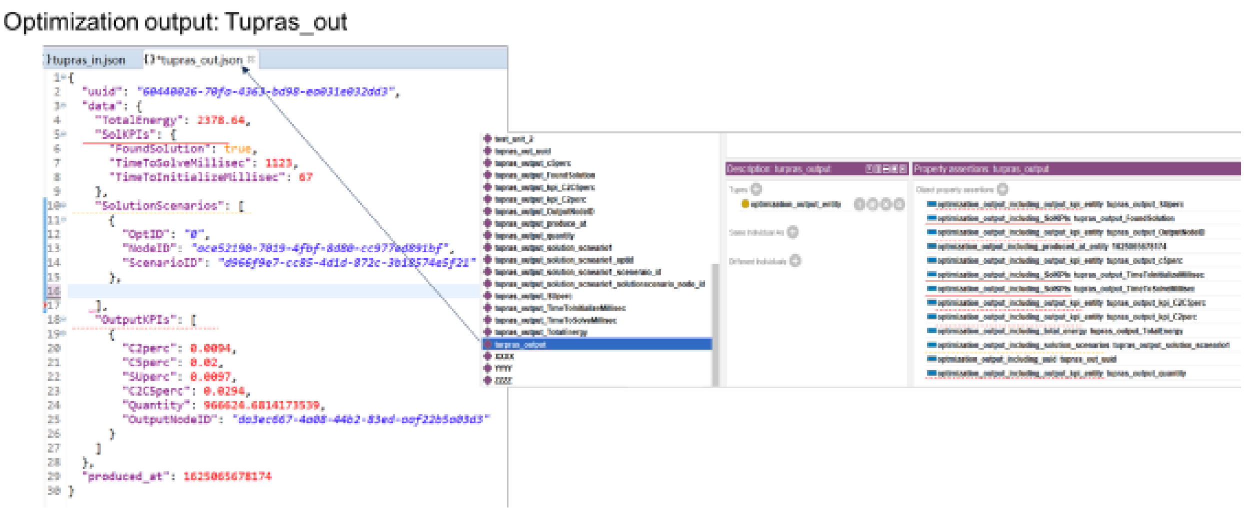

Figure 25: Individuals for optimization in TUPRAS pilot

4.2.3.4 Data analysis formalism

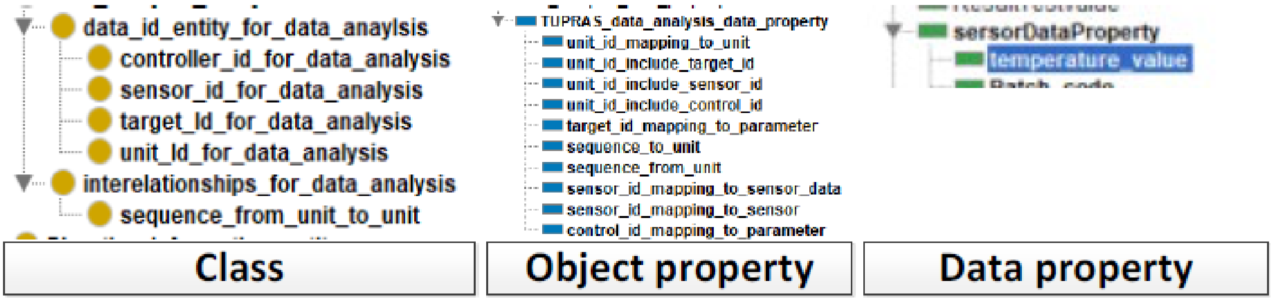

Figure 26: Ontology class, object property, and data property for data analysis in TUPRAS pilot

As shown in Figure 26, optimization classes, object properties and data properties are defined to represent data analysis scenario using individuals. In Figure 27, the individuals are used to represent the data analysis scenario. The TUPRAS data analysis scenario has a uuid which is mapping to the unit “crude distillation unit”.

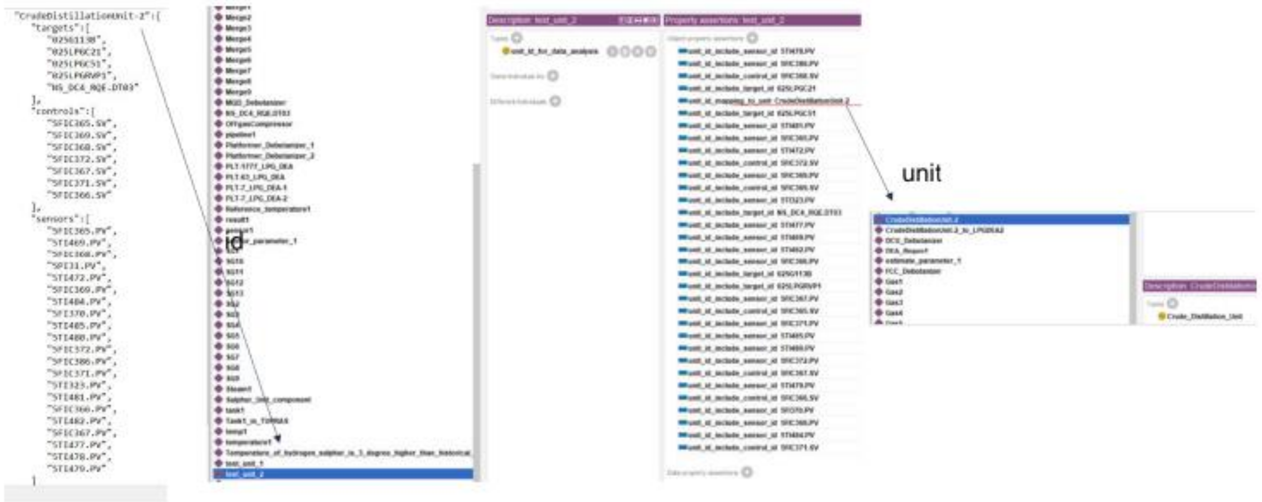

Figure 27: Individuals for data analysis in TUPRAS pilot

4.3 JEMS pilot

JEMS[3] is developing waste-to-fuel transformer plants. The plants transform hydrocarbon-based waste into synthetic diesel fuel through a chemical de-polymerization process. The plants are designed for continuous operation and run the process in a multi-stage pipeline from input to output. The challenge in the JEMS pilot is to use cognitive digital twin technology to ensure the optimal operation of the plant and to avoid malfunctions. The plant is equipped with a large number of sensors monitoring its operation in real time and provide the data to facilitate the cognitive twin.

4.3.1 Pilot scenarios

Scenario #1: Clogging of pipes

---

[3] JEMS pilot did not meet its objectives, especially with regards to the integration of the FACTLOG system to its plant since there is not yet an operative plant in Slovenia.

Description: The material being processed in the plant passes through the plant pipes from the feeding stage at the start all the way to the output after the final distillation. At input the waste is ground down into small pieces and mixed with processing oil so it can pass through the pipes. Since the material is still quite dense and non-homogeneous, the pipes can get clogged. Cleaning the pipes can stop production for up to a day and is therefore costly. If clogging is detected beforehand, there are several actions available to prevent it (e.g. filtering or adding more oil) depending on the cause of the clogging.

Analytics approach: This scenario represents an anomaly detection problem. When a pipe is getting clogged the values reported by the plant sensors deviate from those recorded during normal operation. This can be detected by observing discrepancies between the sensor values and those generated by the simulation based on historic values. The root cause of the anomaly, which in this case means which pipe is getting clogged, can be identified either by using process models built by experts or by using a classification model which finds malfunctions of the same type in historic data.

This is a general anomaly detection approach that can detect any kind of anomaly, not just clogging of the pipes. If its performance would prove to not be satisfactory, a targeted classification model could be built using stream learning methodology by labelling the target clogging situations in the data. The general and targeted approach can run side by side with the targeted model catching the known critical failures, while the general detector covers any other problems.

Scenario #2: New input materials

Description: The plant can process any type of hydrocarbon-based waste, which includes for example old wood, garden trimmings or even plastic garbage. These types of waste differ significantly in their properties such as calorific value, water content, chunk size etc. The plant operating parameters need to be set appropriately to ensure optimal processing of the input materials and determining the best parameter set can be a slow process that takes days. Since the plant is designed for continuous operation, it would ideally be able to automatically adapt to the new material or even variations in the same input material batch.

Analytics approach: To explore the space of possible parameters without actually running the plant we need to be able to simulate the plant operation. We can achieve this by modelling the plant stages taking the parameters and sensor values on the inbound pipes as input data and predicting the sensor values on the outbound pipes. A streaming regression model or an artificial neural network are models that can achieve this purpose. The exploration of the parameter space can be formulated as an optimisation problem for the optimisation component of the FACTLOG platform, however a reinforcement learning approach is also applicable as a solution from the field of machine learning.

4.3.2 Data Types and Sources

The chief source of data in the JEMS case are the sensors from the waste processing plant. The entire process is monitored from input to output and the data includes machine state values such as motor speeds or valve open/closed status as well as operations measurements such as temperature or pressure. Besides being used for monitoring the values are also stored in a historian database. The list of values along with their properties is given in D2.1.

4.3.3 Knowledge Graph Modelling

In the JEMS pilot, two concerns are first considered when knowledge graph models are developed to formalize the JEMS pilot: 1) scenario description; 2) anomaly detection (introduced in Section 5.1.3).

4.3.3.1 Scenario description

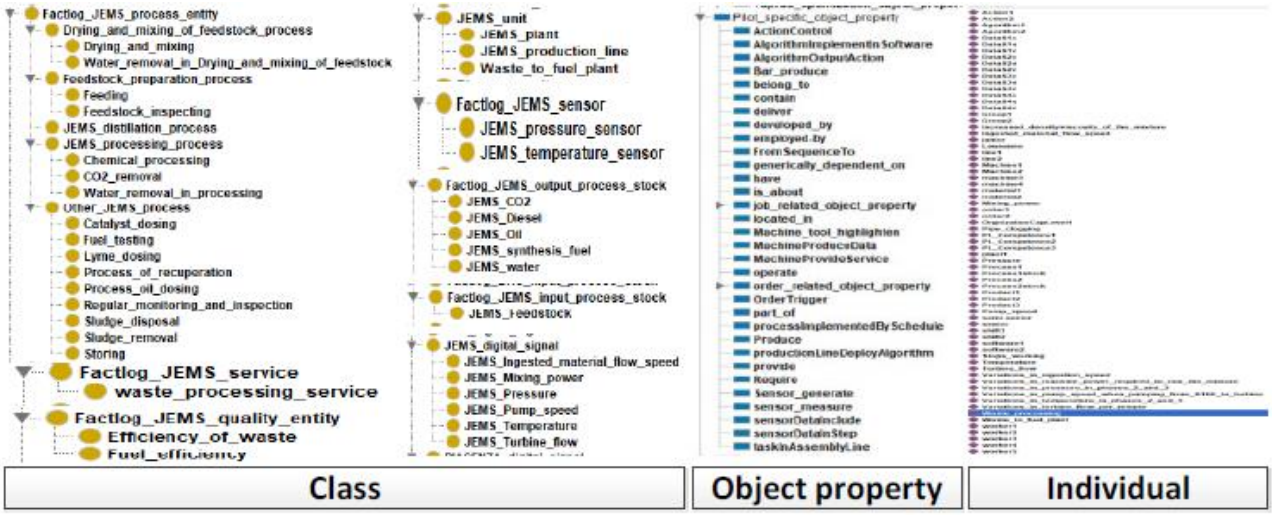

Figure 28: Ontology class, object property, data property and individuals for optimisation for JEMS pilot

As shown in Figure 28, JEMS process entities, services, pilot parameter, sensor, JEMS unit, quality, digital signal, unit, input process stock and output process stock are the main ontology concepts defined in the knowledge graph models. General specific object properties are defined in order to support all the pilot description. Individuals are defined to describe an example of the JEMS scenario.

4.4 PIA pilot

PIA manufactures woollen fabrics and is the leader in their market segment. Their plant receives the wool already cleaned and spun into yarns from a supplier and they perform the weaving into fabric on their machine looms. The looms are massive machines which must be set up with appropriately warping the yarns – a process that can take considerable time. The looms then run the weaving process during which a yarn may break. In case of such breakage, the process must be stopped and the yarn mended, slowing down production.

As a supplier to a very dynamic and demanding market, PIA continuously struggles with meeting the demand for their products. Optimally planning the production orders to minimise delays due to loom setup and adapting the plans to incoming high-priority orders is key to their business. Besides that, due to the demand there is a constant push to set the weaving process on the looms to go as fast as possible, but that increases the likelihood of yarn breakages. Predicting a yarn is likely to break during operation so that it may be avoided would help the process.

4.4.1 Pilot scenarios

Scenario #1: New data streams and storing

Description: The PIA plant already has a data management and collection system that stores the data from the existing sensors, the Manufacturing Execution System (MES) data, the Enterprise Resource Planning (ERP) data and the production schedule. In order to support the improvements aimed for in the project, the collected data needs to be extended with new data sources, in particular with regards to the quality of input materials (e.g., yarn for weaving) and from inside sources, including incremental output (e.g., fabric quality) and performance data (e.g., machine speed).

Analytics approach: Analytics can help evaluate and identify relevant data streams among the new ones. By using methodology for feature selection, the predictive and explanatory value of individual data streams for the target events can be estimated and those that prove to not be useful in the pilot, can be dropped for the full deployment.

Scenario #2: Anomaly detection and new production plan formulation

Description: The plant constantly needs to plan how to process the work orders to meet the demand. Since the loom setup can take a long time, this is a crucial factor for planning. Some setups are more similar any it may be beneficial to plan them one after the other on the same machine. The two main challenges to plan effectively are newly incoming orders of high-priority and the breakages on the looms. Both disrupt the regular operation and require the plant to modify the plans on-the-fly.

Analytics approach: The key contribution of analytics is to provide a prediction for when a yarn is likely to break. This is an anomaly detection problem, which can again be solved with a general anomaly detection methodology or using a targeted stream classification model if there is labelled data available. Since in the PIA case, the target anomaly is known in advance (breaking of the yarn), a targeted classification seems a more likely approach. An additional challenge is identifying the root cause, so that the loom operating parameters can be modified appropriately. The predictions may be probabilistic but should be reliable enough so that when they are fed to the optimisation component as input, good plans can be produced.

4.4.2 Data types and sources

As mentioned above, PIA already collects some operational data in their plant, but plans to expand the set with new sensors. The existing and planned features along with their meta-information are listed in D2.1.

The historic data contains the past orders and related planning from MES and ERP for looms and some sensor readings for energy consumption and water. Energy consumption is an important feature. In weaving, for example, given that all other parameters are constant, an increase in energy use reveals a wearing of components which can lead to an expected stop of production. Currently the past energy consumption is available at aggregated level for the whole weaving department and there are ongoing efforts to obtain this data at the machine level by installing further sensors. All the data relevant for the anomaly detection problem is numeric and well-suited for processing with machine learning algorithms.

At current stage of the project (September 2020) all the above-mentioned data are collected except the ones related with machinery consumption: they are still to be selected. This delay is related with the closure of the company due to the COVID-19 emergency and to the period of mandatory holidays and layoff of the employees caused by the lack of orders. These provisions involved all the division of the company to grant equal economic treatment. It is expected to be able to run the selection and installation of the sensors by the end of 2020. In order to avoid any slowdown in FACTLOG activities a contingency plan has been formed: the involved partners have agreed to define the format of the expected data related with energy in advance and, eventually, to work on estimated dummy data until the real ones will be available.

4.4.3 Knowledge Graph modelling

4.4.3.1 Scenario description

In the PIA pilot, three concerns are first considered when knowledge graph models are developed to formalize the PIA pilot: 1) scenario description; 2) optimization formalism; 3) anomaly detection (introduced in Section 5.1.4).

Figure 29: Ontology class, object property and individuals for PIA pilot description

As shown in Figure 29, PIA process entities, services, pilot parameter, machine, unit, PIA input process stock and output process stock are the main ontology concepts defined in the knowledge graph models. General specific object properties are defined in order to support all the pilot description. Individuals are defined to describe an example of the PIA scenario.

4.4.3.2 Optimization formalism

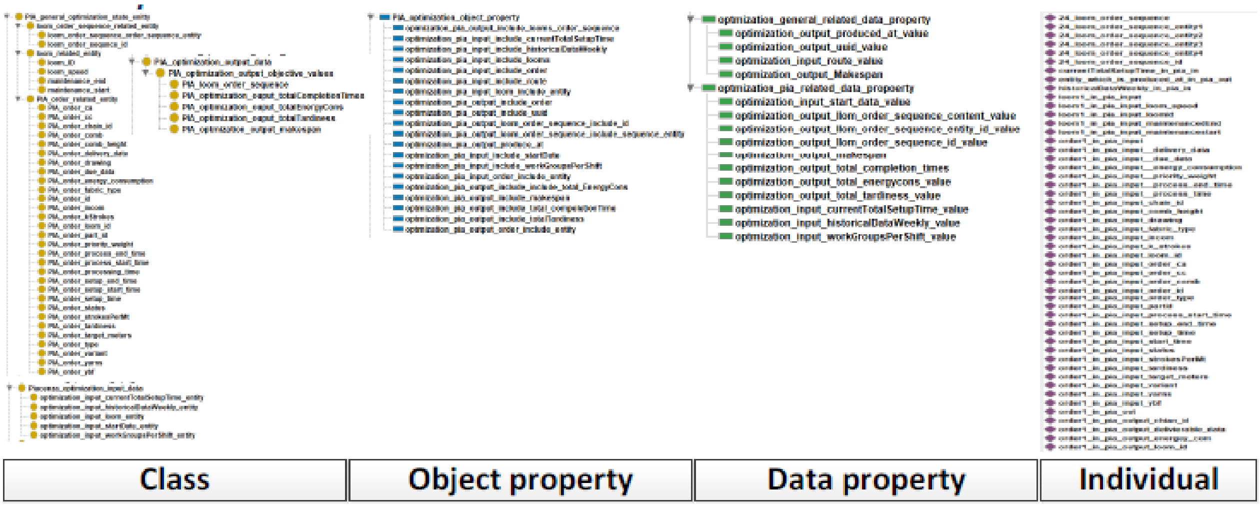

Figure 30: Ontology class, object property, data property and individuals for optimization for PIA pilot

As shown in Figure 30, optimization classes, object properties and data properties are defined to represent optimization scenario using individuals. In Figure 31, the individuals are used to represent the optimization input data structure. The PIA optimization output has an order1 named “order1” who has a processing time and its value is 1991.79.

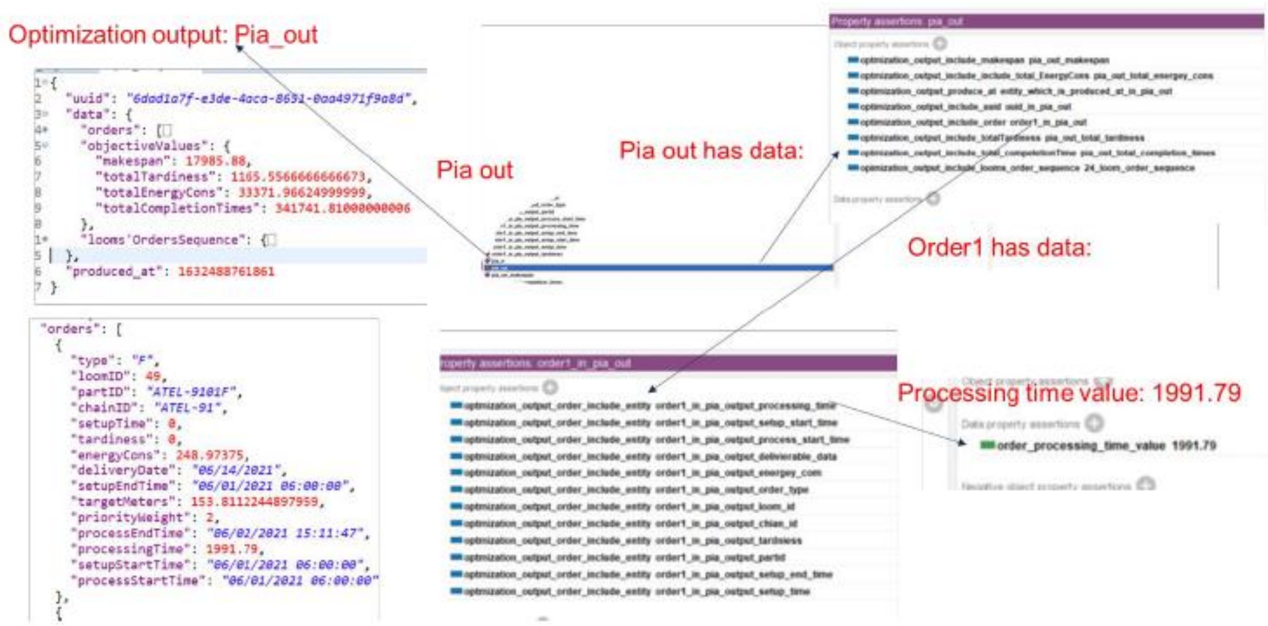

Figure 31: Individuals for describing the optimization scenario for PIA pilot

4.5 CONT pilot

CONT is among the top worldwide electronics manufacturers. Its products are manufactured in electronics plants such as the plant in Timisoara where the pilot line is located. This plant produces high electronic products designed by different CONT developers worldwide. The products are customized for the final customer (i.e. automotive original equipment manufacturers) from the design phase onwards. Although these products (e.g. airbag control units, chaises controllers, hand brake controllers etc.) have a high complexity degree, their manufacturing process can be described (in brief) as follows:

4.5.1 Pilot scenarios

Scenario #1: Machine downtime caused by breakdown

Description: In the Timisoara CONT plant all production lines run non-stop 24/7. Every unplanned downtime affects the plant because it could cause the situation of not delivering the needed quantity in time to the customers. Foreseeing malfunctions in advance would greatly alleviate this problem and improve the Overall Equipment Efficiency (OEE) by increasing the availability and increased quality of the Final Assembly Line. An example of a process where such monitoring is needed is the screwing process:

- The process is implemented with state-of-the-art technology components (e.g. screwdriver, screwdriver controller, axes systems for positioning, PLC for controlling the station).
- HMI interface with operator (permits the operator to know the status of the machine and the step sequence of the process).
- Specific communication with MES system for traceability and monitor performance of the line and specific process parameters.
- Issue handling process is manual: the operator sees an issue in HMI, tries to correct it by interaction with the machine. And in case of no solution the machine is set in breakdown.

Analytics approach: This is again a predictive maintenance problem solvable through either general anomaly detection or targeted (stream) classification models. By modelling the mechanical assembly area of the Final Assembly Line from the sensor readings we predict the machine malfunctions and improve (OEE). Our expectation is the reduction of down time caused by breakdown due to the possibility to forecast issues and plan them in preventive maintenance.

Scenario #2: Machine maintenance cost in % of total operational cost

Description: There are two different types of maintenance done in the plant: Preventive and Corrective/Reactive. By doing this, we can have two types of costs correlated with the number of failures. In case of preventive maintenance, we can have high costs with a low number of failures, while on the other hand, if we do not make preventive maintenance and wait until we have a big number of failures, we will, definitely, have high costs too and also unplanned downtime, which also produces a lot of costs. Finding a balance between the two is crucial.

Analytics approach: The challenge is finding the right balance in terms of cost of preventing malfunction vs number of failures in the Final Assembly Line which is an optimisation problem (illustrated by the graph in Figure 1). Another possible optimisation parameter is the reduction of maintenance cost in terms of head count time used for Predictive, Preventive and Corrective maintenance in the Final Assembly Line. Analytics support this optimisation problem by providing a reliable estimate of the likelihood of malfunction from the models from scenario #1.

Scenario #3: Energy consumption of machines

Description: There is currently no correlation between the line ON/OFF state and the production scheduling at the plant. This way a lot of energy is consumed while the line is not running and the equipment is not operational/running/producing. By taking this into account at the schedule planning stage a significant amount of energy could be saved.

Analytics approach: This is an optimisation problem focused on the reduction of energy consumption/equipment by interacting with the planning in the production and also on a different implementation in the equipment used in production in terms of energy consumption control and monitoring. This includes:

- automatic setup of the Final Assembly Line status (plan / no plan) and Final Assembly Line energy consumption.
- automatic safe mode “shut down” and “wake up” based on production scheduling.

Analytics support this by modelling the energy consumption of machines using regressive machine learning models where this is not possible to estimate otherwise.

4.5.2 Data types and sources

The data in the CONT pilot is coming from the sensors in the production equipment. A Manufacturing Execution System (MES) which collects data from the machines is already in place. Each piece of equipment is connected to a MES client and they exchange data which is then stored in the MES database as illustrated in Figure 32.

Figure 32: The data collection process from the machines through MES clients

The format of the data and its other meta-information is presented in D2.1. The sensors return binary or real-valued reading well suited for the kind of predictive and regression models needed by the pilot. CONT stores over ten years of historic data with a high resolution, so data volume may be a challenge. Feature selection and sub-sampling techniques will be used if necessary.

4.5.3 Knowledge Graph modelling

In the CONT pilot, three main concerns are first considered when knowledge graph models are developed to formalize the CONT pilot: 1) scenario description; 2) optimization formalism; 3) anomaly detection (introduced in Section 5.1.1).

4.5.3.1 *Scenario description*

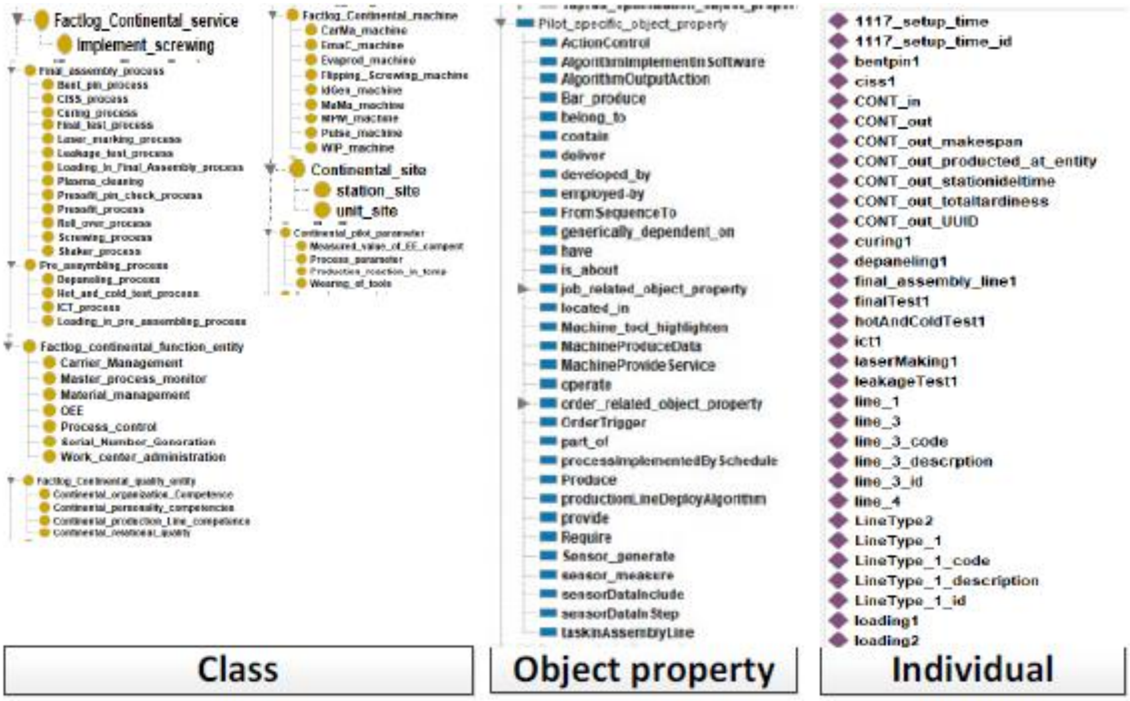

Figure 33: Ontology class, object property, data property and individuals for CONT pilot

As shown in Figure 33, CONT process entities, services, pilot parameter, site, machine, and function are the main ontology concepts defined in the knowledge graph models. General specific object properties are defined in order to support all the pilot description. Individuals are defined to describe an example of the CONT scenario.

4.5.3.2 *Optimization formalism*

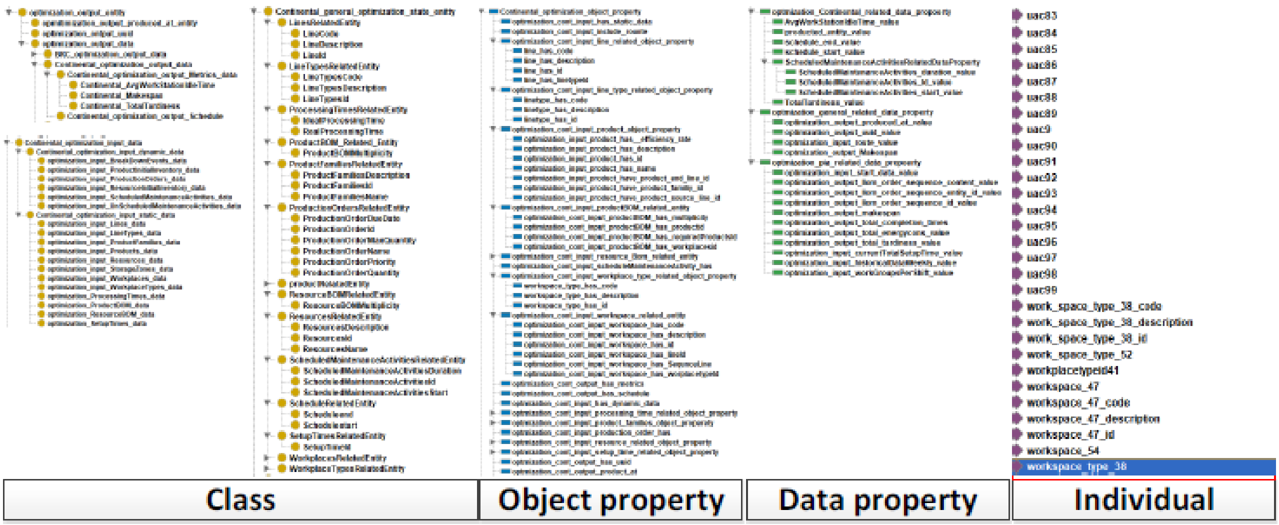

Figure 34: Ontology class, object property, data property and individuals for optimization for CONT pilot

As shown in Figure 34, optimization classes, object properties and data properties are defined to represent optimization scenario using individuals. In Figure 35, the individuals are used to represent the optimization input data structure. The CONT optimization input has a route named “prod-and-maint-sched” whose value is “prod-and-maint-sched”.

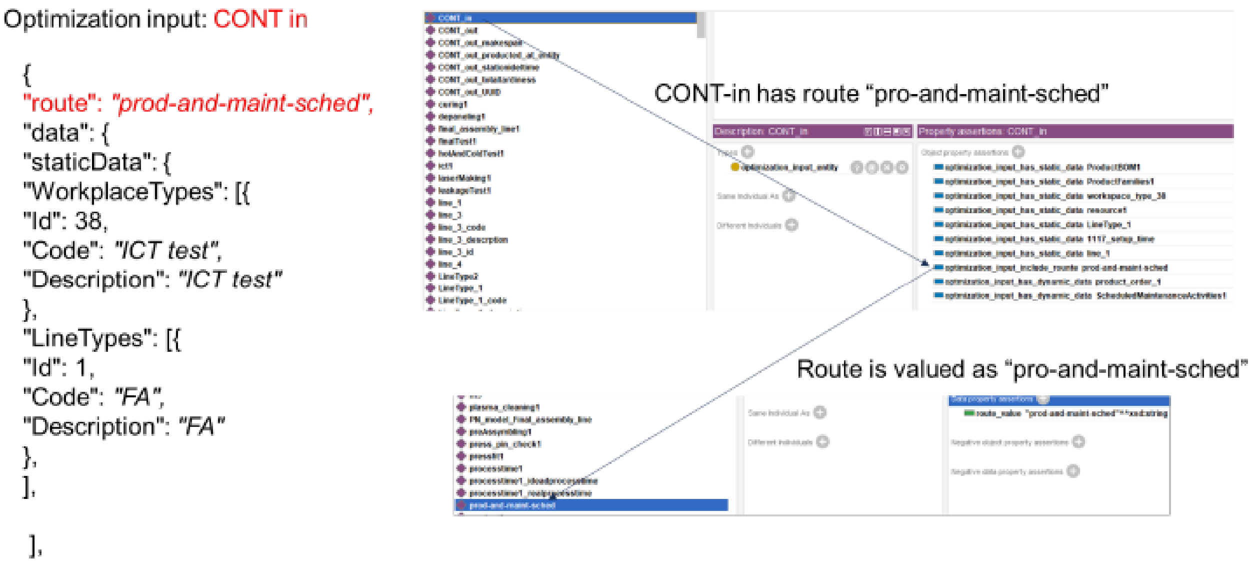

Figure 35: Individuals for describing the optimization scenario for CONT pilot

5 Reasoning for Anomaly Detection based Knowledge Graphs

5.1 Anomaly detection for BRC pilot

Based on the ontology for anomaly detection, a knowledge graph model is defined to represent the rational of anomaly detection for the BRC pilot.

Table 1: Anomaly detection scenario for BRC pilot

Sensor Data **Digital Signals**	Possible Error detection (**Error**)	Required additional data inputs (**Reference data**)	Drift Measure (**Fault status**) extral (Ontology for natural data）	Possible Causes (**Fault**)	Possible outcomes (**Failure**)	**Service**
Hydraulic Cooler ON time	Hydraulic system going over temperature if cooler is on for extended periods	Machine in automatic and safety circuit OK.	The time the cooler is on when the machine is running is varying outside normal operational parameters so system is less efficient	1) Hydraulic pump wear 2) Hydraulic valves not operating correctly. 3) Cooler is inefficient. 4) faulty temperature detector	Hydraulic Oil temperature will go outside safe limits and stop machine	Cooling for Hydraulic system is running

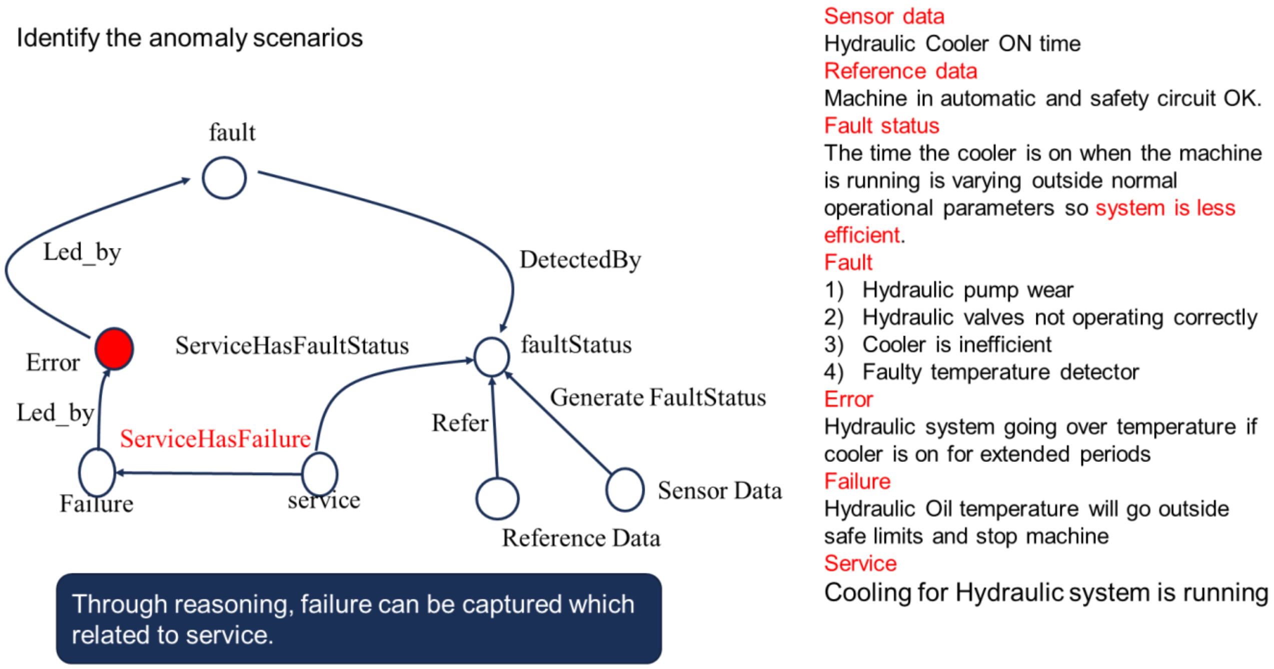

Figure 36: Anomaly scenario for BRC pilot

As shown in Figure 36, based on Table 1, an anomaly scenario is defined. The knowledge graph model is built based on the given data and anomaly ontology. The relationships between failure and two other concepts including service are not defined in the knowledge graph. Thus, a reasoning is executed based on the developed knowledge graph models in order to capture a service has a failure.

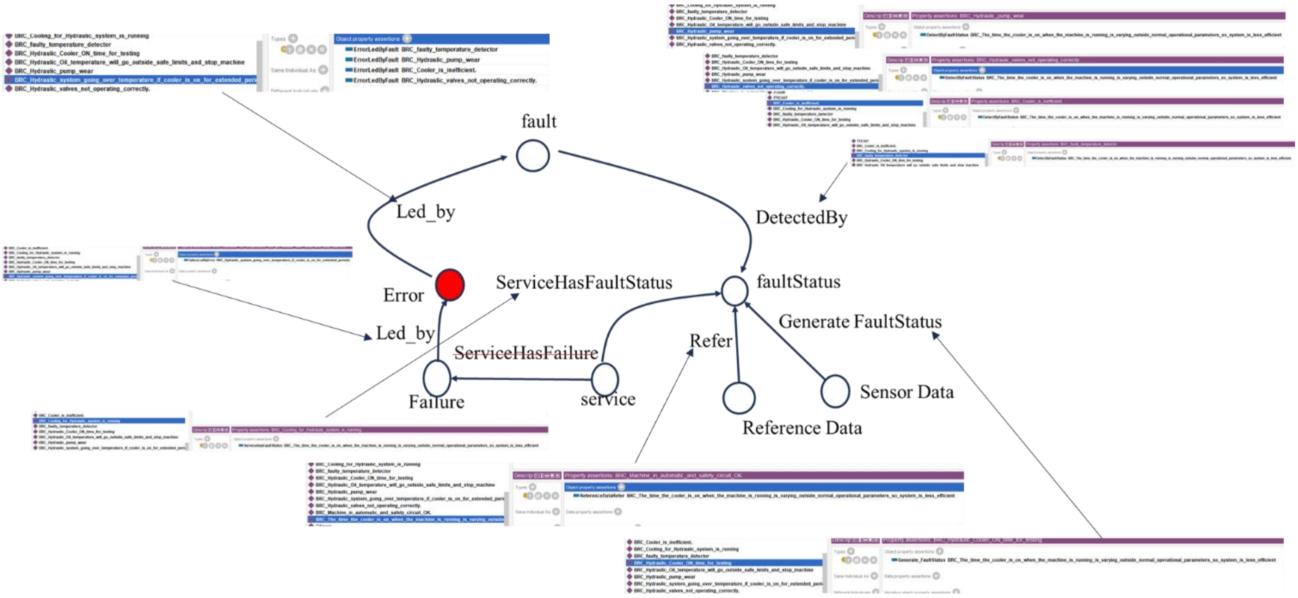

Figure 37: Knowledge Graphs models for BRC pilot

As shown in Figure 37, knowledge graph models are defined to define the anomaly scenario. Through SQWRL rule, a reasoning is implemented:

FactLog_BFO:Hydraulic_system_cooling_service(?servi) ^ FactLog_BFO:ServiceHasFaultStatus(?servi, ?fsta) ^ FactLog_BFO:Hydraulic_Cooler_ON_time(?senData) ^ FactLog_BFO:Generate_FaultStatus(?senData, ?fsta) ^ FactLog_BFO:fault_status(?fsta) ^ FactLog_BFO:DetectByFaultStatus(?fau, ?fsta) ^ FactLog_BFO:fault(?fau) ^ FactLog_BFO:ErrorLedByFault(?err, ?fau) ^ FactLog_BFO:error(?err) ^ FactLog_BFO:FailureLedByError(?imser, ?fal) ^ FactLog_BFO:failure(?fail) -> sqwrl:select(?servi, ?fail)

Finally, a reasoning result is obtained. From the result, we can understand BRC_Cooling_for_Hydraulic_system_is_running service has a failure: BRC_Hydraulic_Oil_temperature_will_go_outside_safe_limits_and_stop_machine.

service	failure
FactLog_BFO:BRC_Cooling_for_Hydraulic_system_is_running	FactLog_BFO:BRC_Hydraulic_Oil_temperature_will_go_outside_safe_limits_and_stop_machine
FactLog_BFO:BRC_Cooling_for_Hydraulic_system_is_running	FactLog_BFO:BRC_Hydraulic_Oil_temperature_will_go_outside_safe_limits_and_stop_machine
FactLog_BFO:BRC_Cooling_for_Hydraulic_system_is_running	FactLog_BFO:BRC_Hydraulic_Oil_temperature_will_go_outside_safe_limits_and_stop_machine

Figure 38: Reasoning results for BRC pilot

5.2 Anomaly detection for JEMS pilot

Based on the ontology for anomaly detection, a knowledge graph model is defined to represent the rationale of anomaly detection for the JEMS pilot[4].

Table 2 – Anomaly detection scenario for JEMS pilot

Sensor Data	Possible Error detection (**Error**)	Required additional data inputs （**Reference data**)	Drift Measure （**Fault status**) extra (ontology for natural data）	Possible Causes (**Fault**)	Possible outcomes (**Failure**)	**Service**	Machine
• Ingested material flow speed • Mixing power • Temperature • Pressure • Turbine flow • Pump speed	**clogged pipe**	Machine in automatic and safety circuit OK.	• Ingested material flow speed: decreasing ingestion speed • Mixing power: increasing machine power required to mix the mixture • Temperature: temperature above 160C or 300C in phases 2 and 3 • Pressure: increased pressure in phases 2 and 3 • Turbine flow: decreased turbine flow per minute • Pump speed: decreased pump speed when pumping from B100 to turbine	The fault status signals highly increased density/viscosity of the mixture preventing normal operation	The machine stops working / must be stopped .	The machine provides the waste processing.	Waste to fuel plant

As shown in Figure 39, based on Table 2, an anomaly scenario is defined. The knowledge graph model is built based on the given data and anomaly ontology. The relationships between failure and two other concepts including service are not defined in the knowledge graph. Thus, a reasoning is executed based on the developed knowledge graph models in order to capture a machine has a failure.

[4] JEMS pilot did not meet its objectives, especially with regards to the integration of the FACTLOG system to its plant since there is not yet an operative plant in Slovenia.

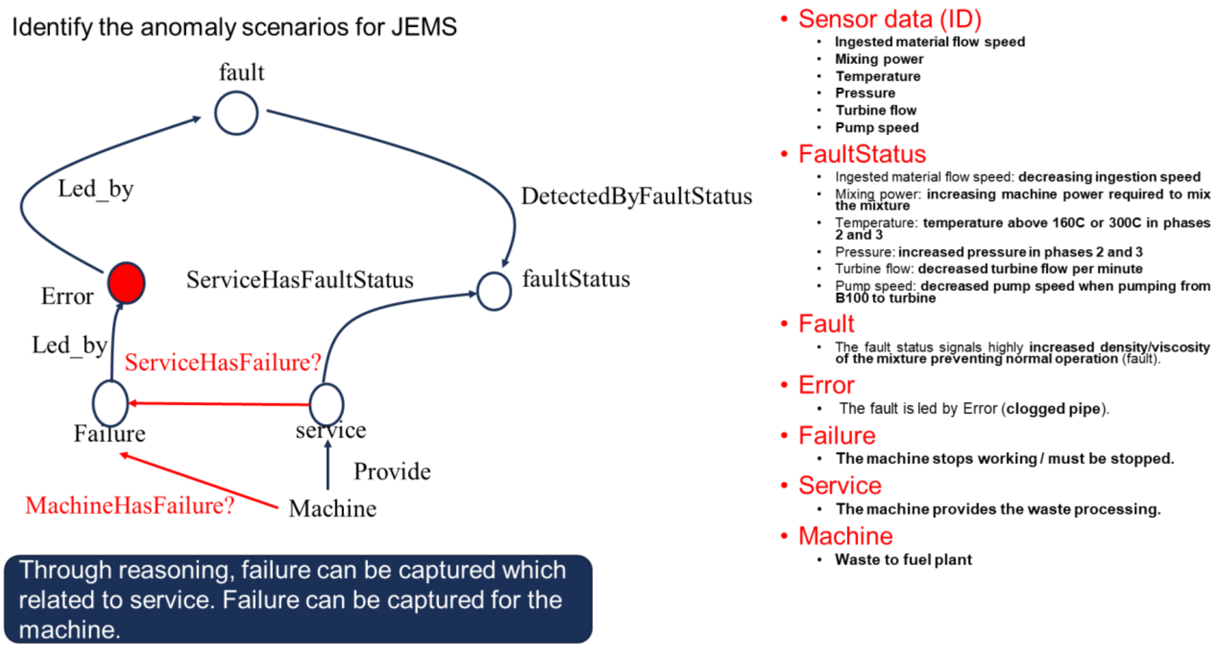

Figure 39: Anomaly scenario for JEMS pilot

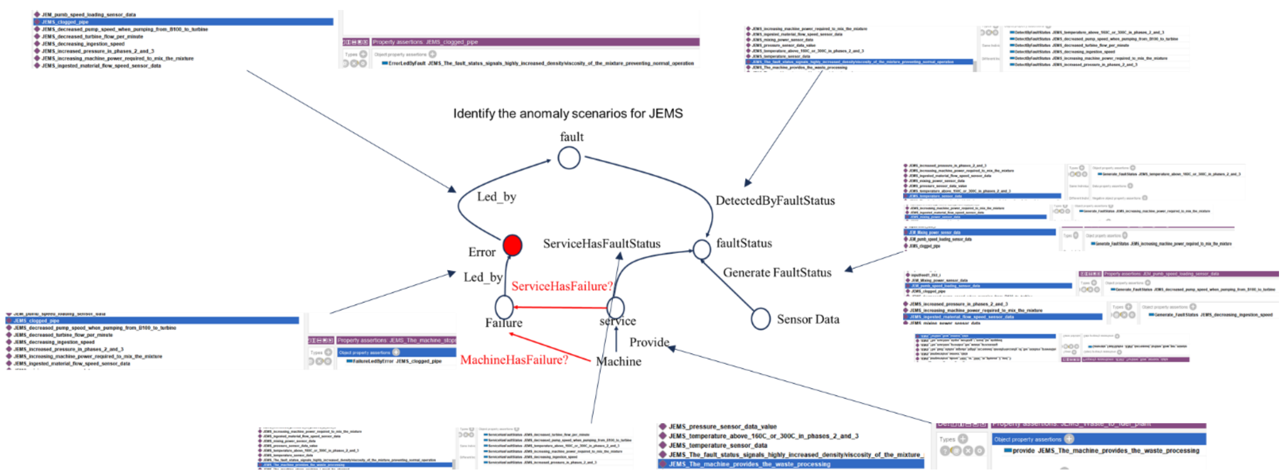

Figure 40: Knowledge graph models for JEMS pilot

As shown in Figure 40, knowledge graph models are defined to define the anomaly scenario. Through SQWRL rule, a reasoning is implemented:

FACTLOG_BFO:Waste_to_fuel_plant (?mac) ^ FACTLOG_BFO:provide(?mac, ?servi) ^ FACTLOG_BFO:waste_processing_service(?servi) ^ FACTLOG_BFO:ServiceHasFaultStatus(?servi, ?fsta) ^ FACTLOG_BFO:fault_status(?fsta) ^ FACTLOG_BFO:DetectByFaultStatus(?fau, ?fsta) ^ FACTLOG_BFO:fault(?fau) ^ FACTLOG_BFO:ErrorLedByFault(?err, ?fau) ^ FACTLOG_BFO:error(?err) ^ FACTLOG_BFO:FailureLedByError(?imser, ?fal) ^ FACTLOG_BFO:failure(?fail) -> sqwrl:select(?mac, ?servi, ?fail)

Finally, a reasoning result is obtained. From the result, we can understand the machine JEMS_Waste_to_fuel_plant provides a service (JEMS_The_machine_provides_the_waste_processing) has a failure: JEMS_The_machine_must_be_stopped.

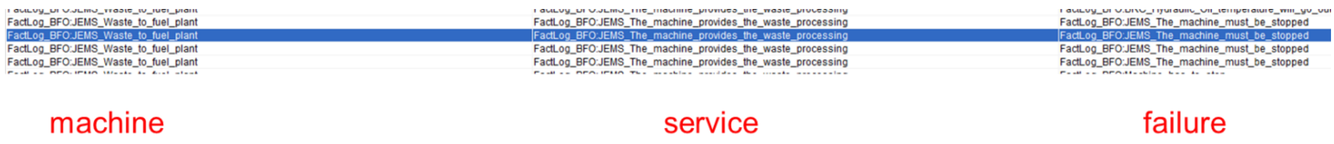

Figure 41: Reasoning result for JEMS pilot

5.3 Anomaly detection for PIA pilot

Based on the ontology for anomaly detection, a knowledge graph model is defined to represent the rational of anomaly detection for the PIA pilot.

Table 3: Anomaly detection scenario for PIA pilot

Sensor Data	Possible Error detection (**Error**)	Required additional data inputs (**Reference data**)	Drift Measure (**Fault status**) extra (ontology for natural data)	Possible Causes (**Fault**)	Possible outcomes (**Failure**)
Mechanical lock	The error is detected by the PLC. Then the yarn brokerage data is sent to MES	Failure cause type is inputted manually by the human after the visual inspection of the stopped loom	The error is detected by the PLC. Then the yarn brokerage data is sent to MES	Digital 0 (stopped)	The error is detected by the PLC. Then the yarn brokerage data is sent to MES

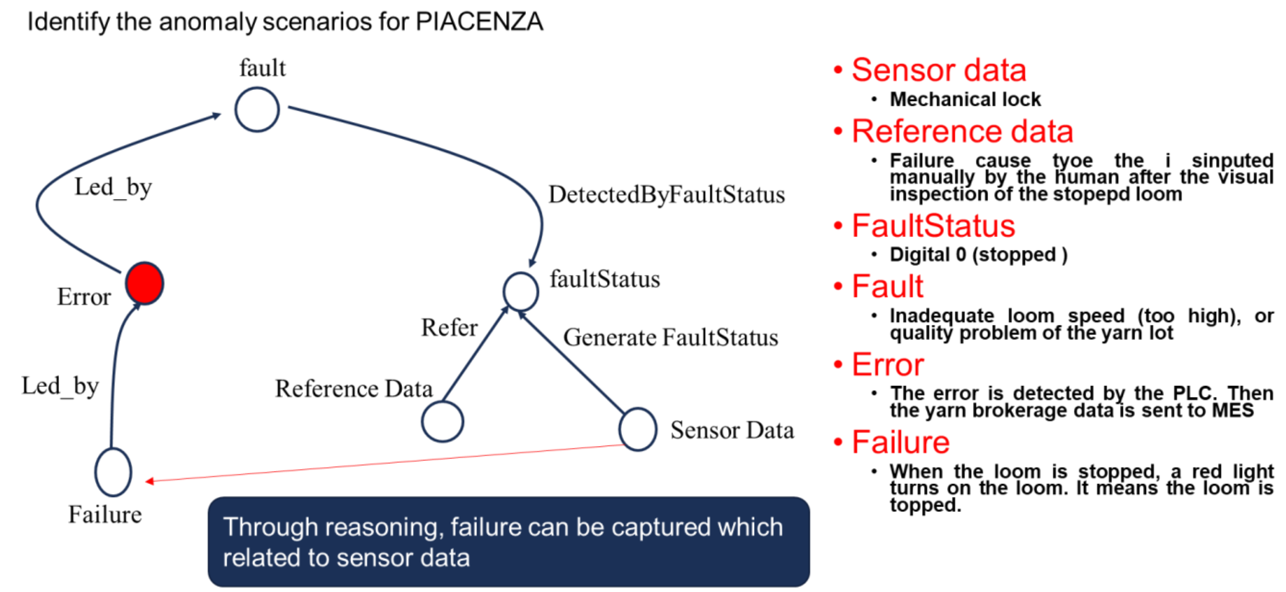

Figure 42: Anomaly detection for PIA pilot

As shown in Figure 42, based on Table 3, an anomaly scenario is defined. The knowledge graph model is built based on the given data and anomaly ontology. The relationships between failure and sensor data are not defined in the knowledge graph. Thus, a reasoning is executed based on the developed knowledge graph models in order to capture a sensor data has a failure.

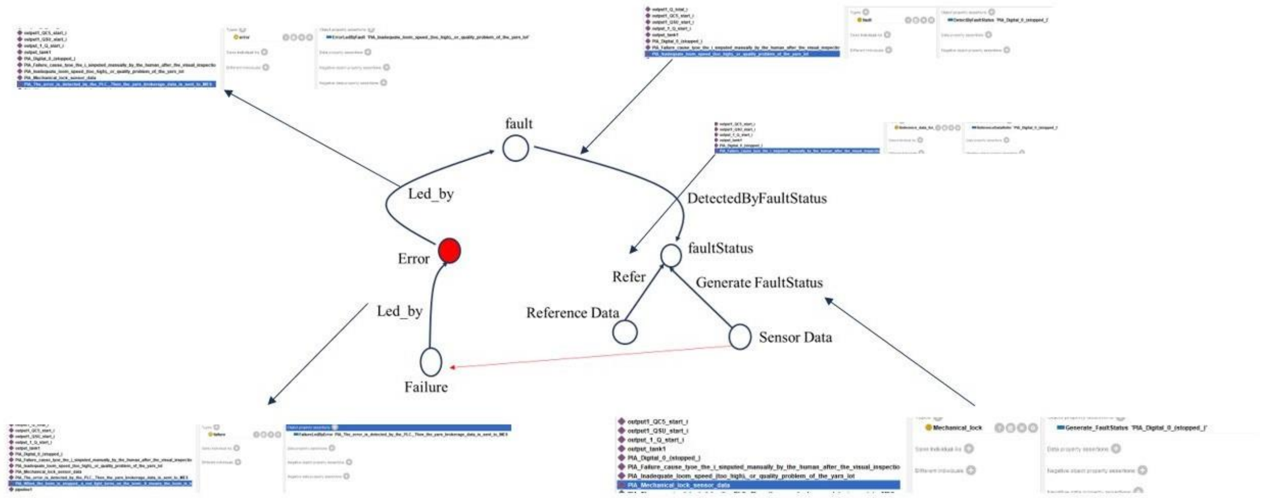

Figure 43: Knowledge Graphs models for PIA pilot

As shown in Figure 43, knowledge graph models are defined to define the anomaly scenario. Through SQWRL rule, a reasoning is implemented:

FactLog_BFO:Mechanical_lock (?sendata) ^ FactLog_BFO:Generate_FaultStatus (?sendata, ?fsta) ^ Reference_data_for_anomaly_analysis (?refedata) ^ FactLog_BFO:ReferenceDataRefer (?refedata, ?fsta) ^ FactLog_BFO:fault_status(?fsta) ^ FactLog_BFO:DetectByFaultStatus(?fau, ?fsta) ^ FactLog_BFO:fault(?fau) ^ FactLog_BFO:ErrorLedByFault(?err, ?fau) ^ FactLog_BFO:error(?err) ^ FactLog_BFO:FailureLedByError(?imser, ?fal) ^ FactLog_BFO:failure(?fail) -> sqwrl:select(?sendata,?fail)

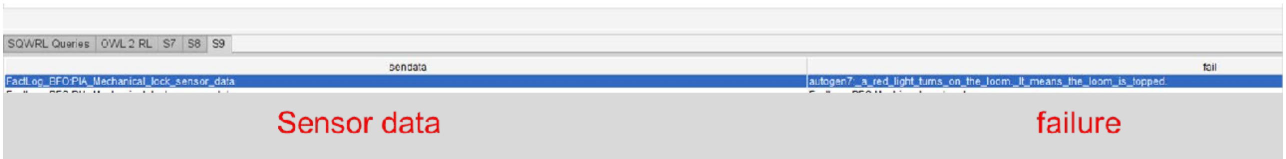

Figure 44: Reasoning result for PIA pilot

Finally, a reasoning result is obtained. From the result, we can understand the sensor data (Mechanical_lock) has a failure: "When the loom is stopped, a red light turns on the loom. It means the loom is topped.

5.4 Anomaly detection for CONT pilot

Based on ontology for anomaly detection, a knowledge graph model is defined to represent the rational of anomaly detection for the CONT pilot.

Table 4: Anomaly detection scenario for CONT pilot

Anomaly								
Sensor Data	Possible Error detection (**Error**)	Required additional data inputs (**Reference data**)	Drift Measure (**Fault status**) extral (ontology for natural data)	Possible Causes (**Fault**)	Possible outcomes (**Failure**)	**Service**	Machine	Machine type

-	Some impurities in the machine	-	The screw height is out of limit	Screw is too high	Machine has to stop	Implement screwing	VW20	Flipping & Screwing

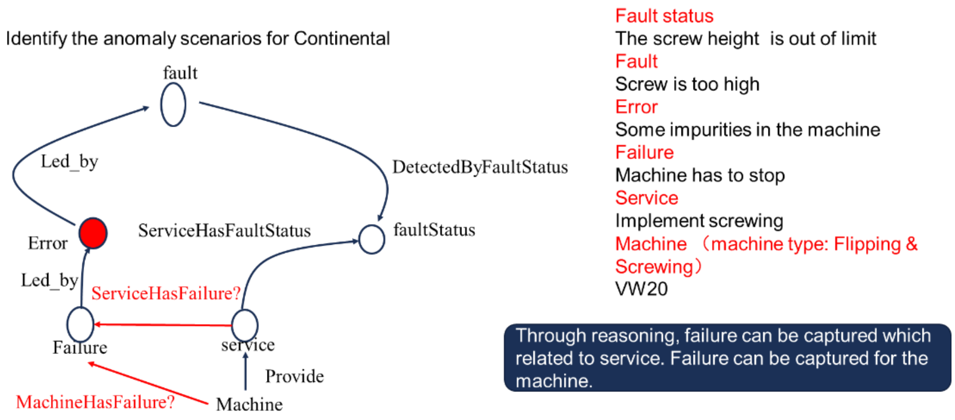

Figure 45: Anomaly scenario for CONT pilot

As shown in Figure 45, based on Table 4, an anomaly scenario is defined. The knowledge graph model is built based on the given data and anomaly ontology. The relationships between failure and two other concepts including service and machine are not defined in the knowledge graph. Thus, a reasoning is executed based on the developed knowledge graph models in order to capture a machine or service has a failure.

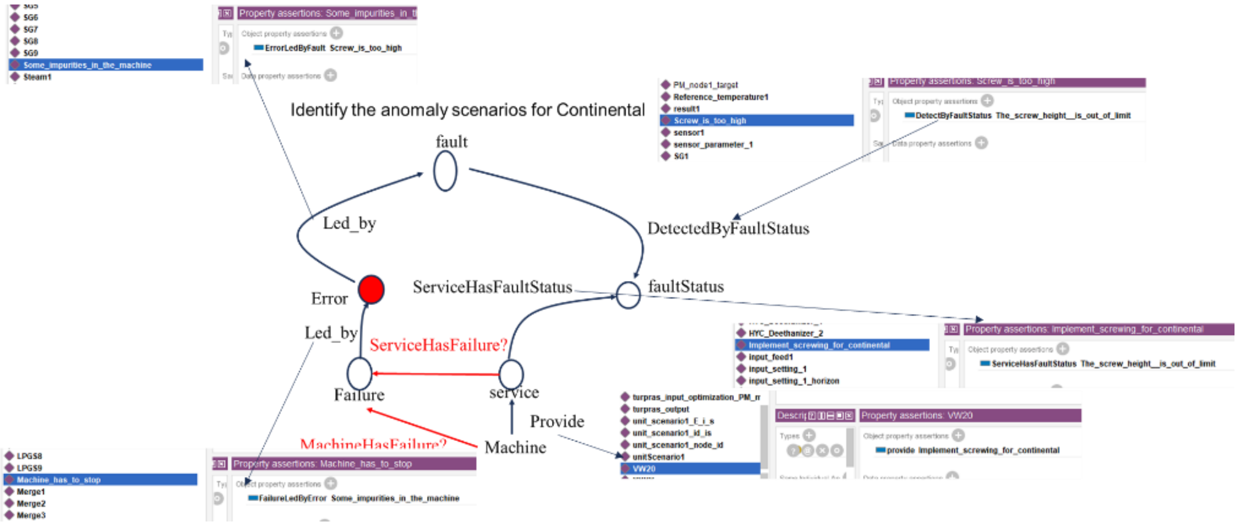

Figure 46: Knowledge graph models for CONT pilot

As shown in Figure 46, knowledge graph models are defined to define the anomaly scenario. Through SQWRL rule, a reasoning is implemented:

FactLog_BFO:Flipping_Screwing_machine(?mac) ^ FactLog_BFO:provide(?mac, ?servi) ^ FactLog_BFO:Implement_screwing(?servi) ^ FactLog_BFO:ServiceHasFaultStatus(?servi, ?fsta) ^ FactLog_BFO:fault_status(?fsta) ^ FactLog_BFO:DetectByFaultStatus(?fau, ?fsta) ^ FactLog_BFO:fault(?fau) ^ FactLog_BFO:ErrorLedByFault(?err, ?fau) ^ FactLog_BFO:error(?err) ^ FactLog_BFO:FailureLedByError(?imser, ?fal) ^ FactLog_BFO:failure(?fail) -> sqwrl:select(?mac, ?servi, ?fail)

SQWRL Queries | OWL 2 RL | S4 | S5 | S6

mac	servi	
FactLog_BFO:VW20	FactLog_BFO:Implement_screwing_for_continental	FactLog_BFO:Machine_has_to_stop
FactLog_BFO:VW20	FactLog_BFO:Implement_screwing_for_continental	FactLog_BFO:Machine_has_to_stop
machine	service	failure

Figure 47: Reasoning result for CONT pilot

Finally, a reasoning result is obtained. From the result, we can understand VW20 machine has a failure: Machine has to stop.

6 Data Visualization for Knowledge Graph Models

As shown in Figure 48, a React and java project is developed based on Jena Lib in order to visualize a knowledge graph model which is built for the FACTLOG project. Totally, there are three steps to realize the visualization:

1. OWL models are developed in Protégé based on the developed ontology.
2. Through the java project we proposed, which details are shown in ANNEX III.
3. The OWL files are shown in the web

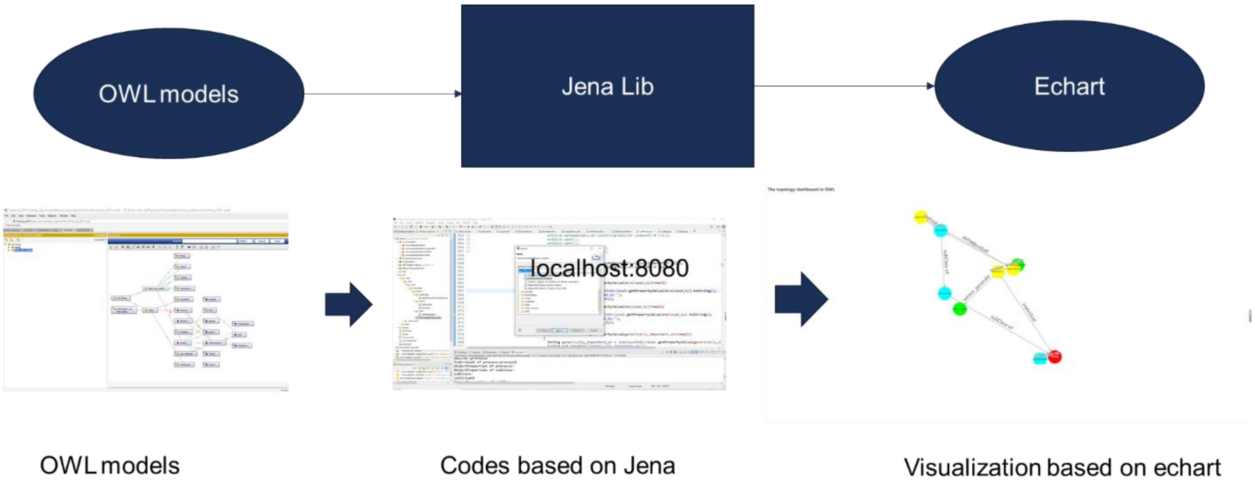

Figure 48: workflow for visualizing knowledge graph models

Conclusions

This report demonstrates basic background of ontology, ontology engineering and knowledge graph modelling first. Then ontology for the entire FACTLOG project is introduced including pilot description, optimization, process modelling, etc. Moreover, knowledge graph models for each pilot are introduced. Finally, reasoning for anomaly detection, data visualization and integration with FACTLOG platform are reported.

References

[1] Okoye K, Islam S, Naeem U, Sharif MS (2020) Semantic-based process mining technique for annotation and modelling of domain processes. Int J Innov Comput Inf Control 16:899–921. doi: 10.24507/ijicic.16.03.899.

[2] Лобанов ГЮ (2013) Basic Formal Ontology как средство построения онтологии в системной модели аргументации. РациоRu 126–143.

[3] Chandrasekaran B, Josephson JR, Benjamins VR (1999) What aro ontologies, and why do we need them? IEEE Intell Syst Their Appl 14:20–26. doi: 10.1109/5254.747902

[4] Krötzsch M, Thost V (2016) Ontologies for Knowledge Graphs: Breaking the Rules. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp 376–392.

[5] Paulheim H (2016) Knowledge graph refinement: A survey of approaches and evaluation methods. Semant Web 8:489–508. doi: 10.3233/SW-160218.

[6] Blumauer A (2014) From Taxonomies over Ontologies to Knowledge Graphs. Semant Web Co 1–3.

[7] Färber M, Ell B, Menne C, Rettinger A, Bartscherer F (2017) Linked Data Quality of DBpedia, Freebase. Semant Web 0:1–53.

[8] Pujara J, Miao H, Getoor L, Cohen W (2013) Knowledge Graph Identification. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp 542–557

[9] (2020) Knowledge Graphs and Machine Learning - Towards Data Science. https://towardsdatascience.com/knowledge-graphs-and-machine-learning-3939b504c7bc

[10] Aijal J (2020) What is a knowledge graph and how does one work? In: https://thenextweb.com/podium/2019/06/11/what-is-a-knowledgegraph-and-how-does-one-work/.

[11] Mike Bergman (2019) A Common Sense View of Knowledge Graphs. In: http://bit.ly/307PEBs and http://bit.ly/2RAbE6X. https://www.mkbergman.com/2244/a-common-sense-view-of-knowledge-graphs/#kg18

[12] Baclawski K, Bennett M, Berg-Cross G, Schneider T, Sharma R, Singer J, Sriram RD (2021) Ontology summit 2020 communiqué: Knowledge graphs. Appl Ontol 16:229–247. doi: 10.3233/AO-210249.

[13] Aasman J (2020) Why Knowledge Graphs Hit the Hype Cycle and What they have in common. In: https://ontologforum.s3.amazonaws.com/OntologySummit2020/Introduction/Why-Knowledge-Graphs-Now--JansAasman_20190904.pdf

[14] Arp R, Smith B, Spear AD (2016) Building Ontologies with Basic Formal Ontology. The MIT Press, Cambridge.

Annex I

Protégé

Protégé is supported by a strong community of academic, government, and corporate users, who use Protégé to build knowledge-based solutions in areas as diverse as biomedicine, e-commerce, and organizational modelling https://protege.stanford.edu/
In our project, we make use of it to model knowledge graph models using OWL.

Unzip the Protege-5.5.0-win and open Protege.exe. Then start your knowledge graph modelling journey.

Twinkle

Twinkle is a simple GUI interface that wraps the ARQ SPARQL query engine. The tool should be useful both for people wanting to learn the SPARQL query language, as well as those doing Semantic Web development http://www.ldodds.com/projects/twinkle/.
In this project, we make use of it to implement SPARQL query.

Unzip the twinkle-2.0-src and open twinkle.jar. Then start your query journey.

Annex II – Knowledge Graph visualization

We have made a project for the visualization for BFO. In this project, two resource classes (Entity and Line) were constructed for the parse of BFO model. The OWL file was read by *Jena* ontology API. The objects of Entity Class and Line Class stored the topological relation of BFO model. The open-source JavaScript visualization library ECHART was used for visualization in this project. Due to the large number of properties of BFO, we only show the properties needed in this particular OWL file in this demo. The details of the project are shown as follow.

HOW TO RUN THE DEMO IN ECLIPSE

(1) Import project

In Eclipse, select the File -> Import. Then the dialog box is shown as Figure 49. Select the Existing Maven Project and choose the demo directory and the maven project is import in the Eclipse. It may take some time to load the libraries.

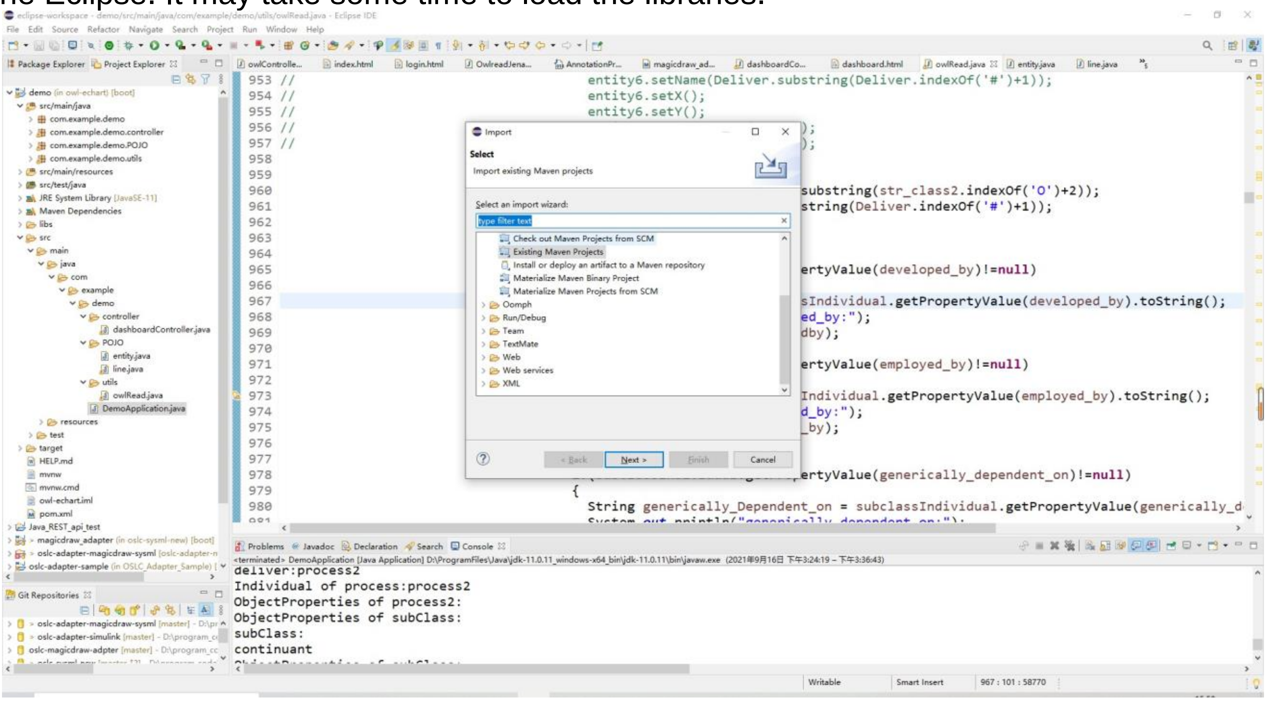

Figure 49: Eclipse project

(2) Modify the file path

Modify the OWL file path in line 33 and line 1828 in owlRead.java. Change file path to your own file path of the OWL file.

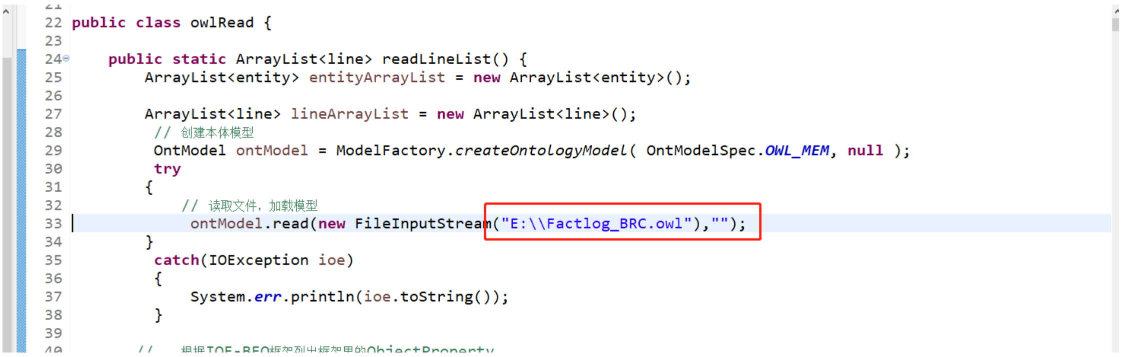

Figure 50: Change the path for loading OWL files

(3) Run the application

Run the springboot project as Java Application, shown as Figure 51.

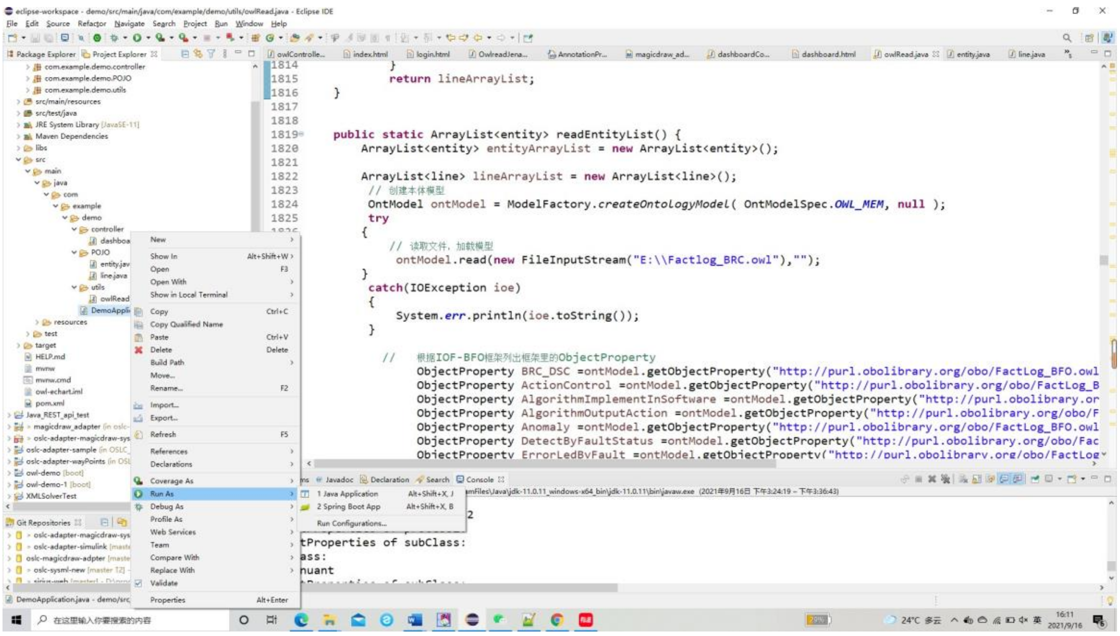

Figure 51: Compile the whole project

(4) Result browser

Open a browser and type in the URL localhost:8080. Then the result is shown as follow which is localhost:8080.

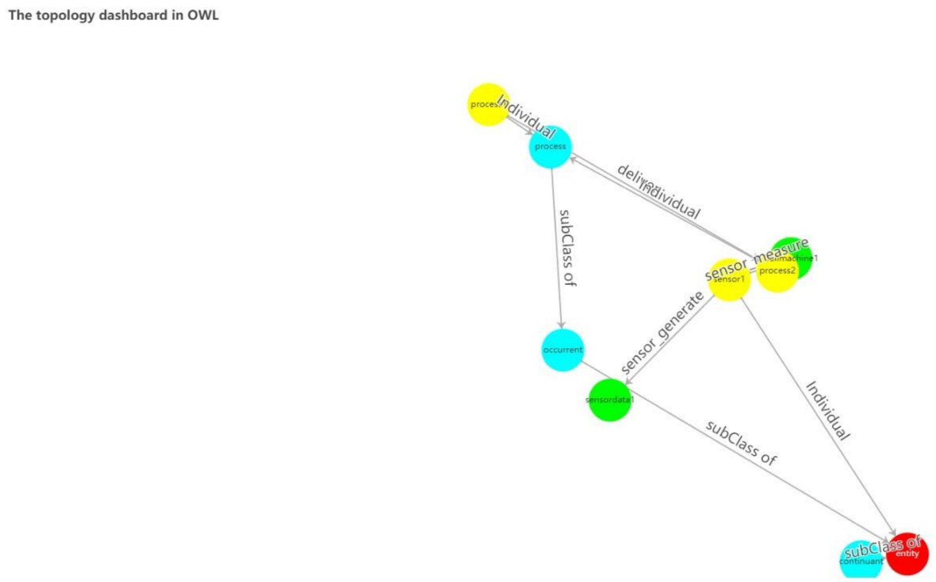

Figure 52: Visualize the OWL information

Annex III – Integrated Ontologies for FACTLOG

The classes and object properties for the FACTLOG ontology are demonstrated as follow:

Table 5: Class of FACTLOG ontology

Id	Ontology class	Description
C 1	Factlog_entity	The FACTLOG root ontology entity
C 1.1	continuant	IAO_0000600 "A continuant is an entity that persists, endures, or continues to exist through time while maintaining its identity.
C 1.1.1	generically_dependent_continuant	A continuant which is such that there is no c and not such that b s-depends_on c at t.
C 1.1.1.1	information_content_entity	A generically dependent continuant that is about something.
C 1.1.1.1.1	Anomaly_entity	a virtual entity used for anomaly detection.
C 1.1.1.1.1.1	error	FactlogDefintion "Since a service is a sequence of the system's external states, a service failure means that at least one (or more) external state of the system deviates from the correct service state. The deviation is called an error."
C 1.1.1.1.1.2	failure	FactlogDefintion "A service failure, often abbreviated here to failure, is an event that occurs when the delivered service deviates from correct service. A"
C 1.1.1.1.1.3	fault	FactlogDefintion "The adjudged or hypothesized cause of an error is called a fault"
C 1.1.1.1.1.4	fault_status	The adjudged or hypothesized cause (its observation is fault status) of an error is called a fault.
C 1.1.1.1.1.4.1	drift_measurement	Fault status observed by drift measurement.
C 1.1.1.1.1.4.2	step_change_measurement	fault status observed by step change measurement.
C 1.1.1.1.2	Data_analysis_entity	FactlogDefintion "a virtual entity used for data analysis."
C 1.1.1.1.2.1	interelationships_for_data_analysis	Relationships between data analysis concept
C 1.1.1.1.2.1.1	sequence_from_unit_to_unit	FactlogDefintion "a directional relationship from one unit to another unit"
C 1.1.1.1.2.2	data_id_entity_for_data_anaylsis	data id of FACTLOG concepts used for data analysis
C 1.1.1.1.2.2.1	controller_id_for_data_analysis	controller_id used for data analysis
C 1.1.1.1.2.2.2	sensor_id_for_data_analysis	sensor_id used for data analysis
C 1.1.1.1.2.2.3	target_Id_for_data_analysis	target_Id used for data analysis
C 1.1.1.1.2.2.4	unit_Id_for_data_analysis	unit_Id used for data analysis
C 1.1.1.1.3	Directive_information_entity	Basic information entity used to represent FACTLOG concepts
C 1.1.1.1.3.1	action_specification	IAO_0000115 "a directive information entity that describes an action the bearer will take"
C 1.1.1.1.3.1.1	Job	A task or piece of work proposed by a service or a machine.

Id	Ontology class	Description
C 1.1.1.1.3.2	objective_specification	IAO_0000115 "a directive information entity that describes an intended process endpoint. When part of a plan specification the concretization is realized in a planned process in which the bearer tries to affect the world so that the process endpoint is achieved."
C 1.1.1.1.3.2.1	Order	an authoritative command or instruction.
C 1.1.1.1.3.2.2	Production order	A production order is an order issued within a company to produce a specific quantity of material within a certain timeframe.
C 1.1.1.1.3.3	plan_specification	IAO_0000115 "A directive information entity with action specifications and objective specifications as parts that, when concretized, is realized in a process in which the bearer tries to achieve the objectives by taking the actions specified."
C 1.1.1.1.3.3.1	program	IAO_0000115 "A language in which source code is written that is intended to be executed/run by a software interpreter. Programming languages are ways to write instructions that specify what to do, and sometimes, how to do it."
C 1.1.1.1.3.3.2	software	IAO_0000115 "Software is a plan specification composed of a series of instructions that can be interpreted by or directly executed by a processing unit."
C 1.1.1.1.3.3.2.1	software_application	IAO_0000115 "A software application is software that can be directly executed by some processing unit."
C 1.1.1.1.3.3.2.2	software_library	IAO_0000115 "A software library is software composed of a collection of software modules and/or software methods in a form that can be statically or dynamically linked to some software application."
C 1.1.1.1.3.3.2.3	software_method	IAO_0000115 "A software method (also called subroutine, subprogram, procedure, method, function, or routine) is software designed to execute a specific task."
C 1.1.1.1.3.3.2.4	software_module	The module of software component
C 1.1.1.1.3.3.2.5	software_script	IAO_0000115 "A software script is software whose instructions can be executed using a software interpreter."
C 1.1.1.1.3.4	source_code_module	IAO_0000115 "A source code module is a directive information entity that specifies, using a programming language, some algorithm."
C 1.1.1.1.4	Optimization_entity	FactlogDefintion "A virtual concepts used to define optimization concept."
C 1.1.1.1.4.1	general_optimization_state_entity	general entity for optimization state
C 1.1.1.1.4.1.1	BRC_general_optimization_state_enitty	general entity for BRC optimization state
C 1.1.1.1.4.1.1.1	BRC_order_related_entity	BRC order related entity for BRC optimization state
C 1.1.1.1.4.1.1.1.1	BRC_order_due_data	order due data for BRC optimization state
C 1.1.1.1.4.1.1.1.2	BRC_order_id	order id for BRC optimization state

Id	Ontology class	Description
C 1.1.1.1.4.1.1.2	job_related_entity	Job related entity for BRC optimization state
C 1.1.1.1.4.1.1.2.1	job_id	job_id for BRC optimization state
C 1.1.1.1.4.1.1.2.2	job_parameter_id	job_parameter_id for BRC optimization state
C 1.1.1.1.4.1.1.2.3	job_parent_id	job_parent_id for BRC optimization state
C 1.1.1.1.4.1.1.2.4	job_process_stage	job_process_stage for BRC optimization state
C 1.1.1.1.4.1.1.2.5	job_process_time_duration	job_process_time_duration for BRC optimization state
C 1.1.1.1.4.1.1.2.6	job_processing_times	job_processing_times for BRC optimization state
C 1.1.1.1.4.1.1.3	machine_related_entity	Machine related entity for BRC optimization state
C 1.1.1.1.4.1.1.3.1	completion_time	Machine completion_time for BRC optimization state
C 1.1.1.1.4.1.1.3.2	machine_id	machine_id for BRC optimization state
C 1.1.1.1.4.1.1.3.3	machine_status	machine_status for BRC optimization state
C 1.1.1.1.4.1.1.3.4	machine_type	machine_type for BRC optimization state
C 1.1.1.1.4.1.1.3.5	parent_id	Machine parent_id for BRC optimization state
C 1.1.1.1.4.1.1.3.6	setup_time	Machine setup_time for BRC optimization state
C 1.1.1.1.4.1.1.3.7	start_time	Machine start_time for BRC optimization state
C 1.1.1.1.4.1.1.3.8	status	Machine Status for BRC optimization state
C 1.1.1.1.4.1.2	CONT_general_optimization_state_entity	General entity for CONT optimization state
C 1.1.1.1.4.1.2.1	LinesRelatedEntity	LinesRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.1.1	LineCode	FactlogDefintion "LineCode for Lines in CONT optimization state"
C 1.1.1.1.4.1.2.1.2	LineDescription	FactlogDefintion "LineDescription for Lines in CONT optimization state"
C 1.1.1.1.4.1.2.1.3	LineId	FactlogDefintion "LineId for Lines in CONT optimization state"
C 1.1.1.1.4.1.2.2	LineTypesRelatedEntity	LineTypesRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.2.1	LineTypesCode	FactlogDefintion "LineTypesCode for CONT optimization state"
C 1.1.1.1.4.1.2.2.2	LineTypesDescription	FactlogDefintion "LineTypesDescription for CONT optimization state"
C 1.1.1.1.4.1.2.2.3	LineTypesId	FactlogDefintion "LineTypesId for CONT optimization state"
C 1.1.1.1.4.1.2.3	ProcessingTimesRelatedEntity	ProcessingTimesRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.3.1	IdealProcessingTime	FactlogDefintion "IdealProcessingTime for CONT optimization state"
C 1.1.1.1.4.1.2.3.2	RealProcessingTime	RealProcessingTime for CONT optimization state"
C 1.1.1.1.4.1.2.4	ProductBOM_Related_Entity	ProductBOM_Related_Entity for CONT optimization state

Id	Ontology class	Description
C 1.1.1.1.4.1.2.4.1	ProductBOMMultiplicity	FactlogDefintion "ProductBOMMultiplicity for CONT optimization state"
C 1.1.1.1.4.1.2.5	ProductFamiliesRelatedEntity	ProductFamiliesRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.5.1	ProductFamiliesDescription	FactlogDefintion "ProductFamiliesDescription for CONT optimization state"
C 1.1.1.1.4.1.2.5.2	ProductFamiliesId	FactlogDefintion "ProductFamiliesId for CONT optimization state"
C 1.1.1.1.4.1.2.5.3	ProductFamiliesName	FactlogDefintion "ProductFamiliesName for CONT optimization state"
C 1.1.1.1.4.1.2.6	ProductionOrdersRelatedEntity	ProductionOrdersRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.6.1	ProductionOrderDueDate	FactlogDefintion "ProductionOrderDueDate for CONT optimization state"
C 1.1.1.1.4.1.2.6.2	ProductionOrderId	FactlogDefintion "ProductionOrderId for CONT optimization state"
C 1.1.1.1.4.1.2.6.3	ProductionOrderMaxQuantity	FactlogDefintion "ProductionOrderMaxQuantity for CONT optimization state"
C 1.1.1.1.4.1.2.6.4	ProductionOrderName	FactlogDefintion "ProductionOrderName for CONT optimization state"
C 1.1.1.1.4.1.2.6.5	ProductionOrderPriority	FactlogDefintion "ProductionOrderPriority for CONT optimization state"
C 1.1.1.1.4.1.2.6.6	ProductionOrderQuantity	FactlogDefintion "ProductionOrderQuantity for CONT optimization state"
C 1.1.1.1.4.1.2.7	productRelatedEntity	productRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.7.1	EfficiencyRate	FactlogDefintion "EfficiencyRate for CONT optimization state"
C 1.1.1.1.4.1.2.7.2	productDescription	FactlogDefintion "productDescription for CONT optimization state"
C 1.1.1.1.4.1.2.7.3	productionId	FactlogDefintion "productionId for CONT optimization state"
C 1.1.1.1.4.1.2.7.4	productName	FactlogDefintion "productName for CONT optimization state"
C 1.1.1.1.4.1.2.8	ResourceBOMRelatedEntity	ResourceBOMRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.8.1	ResourceBOMMultiplicity	FactlogDefintion "ResourceBOMMultiplicity for CONT optimization state"
C 1.1.1.1.4.1.2.9	ResourcesRelatedEntity	ResourcesRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.9.1	ResourcesDescription	ResourcesDescription for CONT optimization state
C 1.1.1.1.4.1.2.9.2	ResourcesId	ResourcesId for CONT optimization state
C 1.1.1.1.4.1.2.9.3	ResourcesName	ResourcesName for CONT optimization state
C 1.1.1.1.4.1.2.10	ScheduledMaintenanceActivitiesRelatedEntity	ScheduledMaintenanceActivitiesRelat edEntity for CONT optimization state
C 1.1.1.1.4.1.2.10.1	ScheduledMaintenanceActivitiesDuration	ScheduledMaintenanceActivitiesDurat ion for CONT optimization state

Id	Ontology class	Description
C 1.1.1.1.4.1.2.10.2	ScheduledMaintenanceActivitiesId	ScheduledMaintenanceActivitiesId for CONT optimization state
C 1.1.1.1.4.1.2.10.3	ScheduledMaintenanceActivitiesStart	ScheduledMaintenanceActivitiesStart for CONT optimization state
C 1.1.1.1.4.1.2.11	ScheduleRelatedEntity	ScheduleRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.11.1	Scheduleend	Scheduleend for CONT optimization state
C 1.1.1.1.4.1.2.11.2	Schedulestart	Schedulestart for CONT optimization state
C 1.1.1.1.4.1.2.12	SetupTimesRelatedEntity	SetupTimesRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.12.1	SetupTimeId	SetupTimeId for CONT optimization state
C 1.1.1.1.4.1.2.13	WorkplacesRelatedEntity	WorkplacesRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.13.1	SequenceInLine	SequenceInLine for CONT optimization state
C 1.1.1.1.4.1.2.13.2	WorkplaceCode	WorkplaceCode for CONT optimization state
C 1.1.1.1.4.1.2.13.3	WorkplaceDescription	WorkplaceDescription for CONT optimization state
C 1.1.1.1.4.1.2.13.4	WorkplaceId	WorkplaceId for CONT optimization state
C 1.1.1.1.4.1.2.14	WorkplaceTypesRelatedEntity	WorkplaceTypesRelatedEntity for CONT optimization state
C 1.1.1.1.4.1.2.14.1	WorkplaceTypesCode	WorkplaceTypesCode for CONT optimization state
C 1.1.1.1.4.1.2.14.2	WorkplaceTypesDescription	WorkplaceTypesDescription for CONT optimization state
C 1.1.1.1.4.1.2.14.3	WorkplaceTypesId	WorkplaceTypesId for CONT optimization state
C 1.1.1.1.4.1.3	PIA_general_optimization_state_enitty	general entity for PIA optimization state
C 1.1.1.1.4.1.3.1	loom_order_sequence_related_entity	loom_order_sequence_related_entity for PIA optimization state
C 1.1.1.1.4.1.3.1.1	loom_order_sequence_order_sequence_entity	loom_order_sequence_order_sequence_entity for PIA optimization state
C 1.1.1.1.4.1.3.1.2	loom_order_sequnce_id	loom_order_sequnce_id for PIA optimization state
C 1.1.1.1.4.1.3.2	loom_related_entity	loom_related_entity for PIA optimization state
C 1.1.1.1.4.1.3.2.1	loom_ID	loom_ID for PIA optimization state
C 1.1.1.1.4.1.3.2.2	loom_speed	loom_speed for PIA optimization state
C 1.1.1.1.4.1.3.2.3	maintenance_end	maintenance_end for PIA optimization state
C 1.1.1.1.4.1.3.2.4	maintenance_start	maintenance_start for PIA optimization state
C 1.1.1.1.4.1.3.3	PIA_order_related_entity	PIA_order_related_entity for PIA optimization state
C 1.1.1.1.4.1.3.3.1	PIA_order_ca	PIA_order_ca for PIA optimization state
C 1.1.1.1.4.1.3.3.2	PIA_order_cc	PIA_order_cc for PIA optimization state
C 1.1.1.1.4.1.3.3.3	PIA_order_chain_id	PIA_order_chain_id for PIA optimization state
C 1.1.1.1.4.1.3.3.4	PIA_order_comb	PIA_order_comb for PIA optimization state
C 1.1.1.1.4.1.3.3.5	PIA_order_comb_height	PIA_order_comb_height for PIA optimization state
C 1.1.1.1.4.1.3.3.6	PIA_order_delivery_date	PIA_order_delivery_date for PIA optimization state

Id	Ontology class	Description
C 1.1.1.1.4.1.3.3.7	PIA_order_drawing	PIA_order_drawing for PIA optimization state
C 1.1.1.1.4.1.3.3.8	PIA_order_due_data	PIA_order_due_data for PIA optimization state
C 1.1.1.1.4.1.3.3.9	PIA_order_energy_consumption	PIA_order_energy_consumption for PIA optimization state
C 1.1.1.1.4.1.3.3.10	PIA_order_fabric_type	PIA_order_fabric_type for PIA optimization state
C 1.1.1.1.4.1.3.3.11	PIA_order_id	PIA_order_id for PIA optimization state
C 1.1.1.1.4.1.3.3.12	PIA_order_incom	PIA_order_incom for PIA optimization state
C 1.1.1.1.4.1.3.3.13	PIA_order_kStrokes	PIA_order_kStrokes for PIA optimization state
C 1.1.1.1.4.1.3.3.14	PIA_order_loom_id	PIA_order_loom_id for PIA optimization state
C 1.1.1.1.4.1.3.3.15	PIA_order_part_id	PIA_order_part_id for PIA optimization state
C 1.1.1.1.4.1.3.3.16	PIA_order_priority_weight	PIA_order_priority_weight for PIA optimization state
C 1.1.1.1.4.1.3.3.17	PIA_order_process_end_time	PIA_order_process_end_time for PIA optimization state
C 1.1.1.1.4.1.3.3.18	PIA_order_process_start_time	PIA_order_process_start_time for PIA optimization state
C 1.1.1.1.4.1.3.3.19	PIA_order_processing_time	PIA_order_processing_time for PIA optimization state
C 1.1.1.1.4.1.3.3.20	PIA_order_setup_end_time	PIA_order_setup_end_time for PIA optimization state
C 1.1.1.1.4.1.3.3.21	PIA_order_setup_start_time	PIA_order_setup_start_time for PIA optimization state
C 1.1.1.1.4.1.3.3.22	PIA_order_setup_time	PIA_order_setup_time for PIA optimization state
C 1.1.1.1.4.1.3.3.23	PIA_order_status	PIA_order_status for PIA optimization state
C 1.1.1.1.4.1.3.3.24	PIA_order_strokesPerMt	PIA_order_strokesPerMt for PIA optimization state
C 1.1.1.1.4.1.3.3.25	PIA_order_tardiness	PIA_order_tardiness for PIA optimization state
C 1.1.1.1.4.1.3.3.26	PIA_order_target_meters	PIA_order_target_meters for PIA optimization state
C 1.1.1.1.4.1.3.3.27	PIA_order_type	PIA_order_type for PIA optimization state
C 1.1.1.1.4.1.3.3.28	PIA_order_variant	PIA_order_variant for PIA optimization state
C 1.1.1.1.4.1.3.3.29	PIA_order_yarns	PIA_order_yarns for PIA optimization state
C 1.1.1.1.4.1.3.3.30	PIA_order_ybf	PIA_order_ybf for PIA optimization state
C 1.1.1.1.4.1.4	Tupras_general_optimization_state_enitty	general entity for Tupras optimization state
C 1.1.1.1.4.1.4.1	InputFeeds_related_entity	InputFeeds_related_entity for Tupras optimization state
C 1.1.1.1.4.1.4.1.1	IC2_i	IC2_i for Tupras optimization state
C 1.1.1.1.4.1.4.1.2	IC5_i	IC5_i for Tupras optimization state
C 1.1.1.1.4.1.4.1.3	IF_i	IF_i for Tupras optimization state
C 1.1.1.1.4.1.4.1.4	InputNodeID	InputNodeID for Tupras optimization state
C 1.1.1.1.4.1.4.1.5	ISU_i	ISU_i for Tupras optimization state

Id	Ontology class	Description
C 1.1.1.1.4.1.4.2	linkScenarios_related_entity	linkScenarios_related_entity for Tupras optimization state
C 1.1.1.1.4.1.4.2.1	CAP_ij	CAP_ij for Tupras optimization state
C 1.1.1.1.4.1.4.2.2	link_id_is	link_id_is for Tupras optimization state
C 1.1.1.1.4.1.4.2.3	PC2_i_j_s	PC2_i_j_s for Tupras optimization state
C 1.1.1.1.4.1.4.2.4	PC5_i_j_s	PC5_i_j_s for Tupras optimization state
C 1.1.1.1.4.1.4.2.5	PF_i_j_s	PF_i_j_s for Tupras optimization state
C 1.1.1.1.4.1.4.2.6	PSU_i_j_s	PSU_i_j_s for Tupras optimization state
C 1.1.1.1.4.1.4.3	optimization_input_Settings_related_entity	optimization_input_Settings_related_e ntity for Tupras optimization state
C 1.1.1.1.4.1.4.3.1	Horizon	Horizon for Tupras optimization state
C 1.1.1.1.4.1.4.3.2	PriceOfElectricity	PriceOfElectricity for Tupras optimization state
C 1.1.1.1.4.1.4.3.3	PriceOfLPG	PriceOfLPG for Tupras optimization state
C 1.1.1.1.4.1.4.3.4	PriceOfNG	PriceOfNG for Tupras optimization state
C 1.1.1.1.4.1.4.3.5	TimeToOptimize	TimeToOptimize for Tupras optimization state
C 1.1.1.1.4.1.4.4	optimization_input_specs_related_entity	optimization_input_specs_related_entity for Tupras optimization state
C 1.1.1.1.4.1.4.4.1	C2	C2 for Tupras optimization state
C 1.1.1.1.4.1.4.4.2	C2C5	C2C5 for Tupras optimization state
C 1.1.1.1.4.1.4.4.3	C5	C5 for Tupras optimization state
C 1.1.1.1.4.1.4.4.4	SU	SU for Tupras optimization state
C 1.1.1.1.4.1.4.5	optimization_output_SolutionScenariosRelatedEntity	optimization_output_SolutionScenariosRelatedEntity for Tupras optimization state
C 1.1.1.1.4.1.4.5.1	OptID	OptID for Tupras optimization state
C 1.1.1.1.4.1.4.5.2	Scenario_ID	Scenario_ID for Tupras optimization state
C 1.1.1.1.4.1.4.5.3	SolutionScenariosNode_ID	SolutionScenariosNode_ID for Tupras optimization state
C 1.1.1.1.4.1.4.6	Output_tanks_related_entity	Output_tanks_related_entity for Tupras optimization state
C 1.1.1.1.4.1.4.6.1	OutputNodeID	OutputNodeID for Tupras optimization state
C 1.1.1.1.4.1.4.6.2	Q_start_i	Q_start_i for Tupras optimization state
C 1.1.1.1.4.1.4.6.3	Q_total_i	Q_total_i for Tupras optimization state
C 1.1.1.1.4.1.4.6.4	QC2_start_i	QC2_start_i for Tupras optimization state
C 1.1.1.1.4.1.4.6.5	QC5_start_i	QC5_start_i for Tupras optimization state
C 1.1.1.1.4.1.4.6.6	QSU_start_i	QSU_start_i for Tupras optimization state
C 1.1.1.1.4.1.4.7	UnitScenarios_related_entity	UnitScenarios_related_entity for Tupras optimization state
C 1.1.1.1.4.1.4.7.1	E_i_s	E_i_s for Tupras optimization state
C 1.1.1.1.4.1.4.7.2	Unit_Scenarios_id_is	Unit_Scenarios_id_is for Tupras optimization state
C 1.1.1.1.4.1.4.7.3	UnitScenarios_NodeID	UnitScenarios_NodeID for Tupras optimization state
C 1.1.1.1.4.2	optimization_input_entity	general entity for optimization input
C 1.1.1.1.4.2.1	optimization_input_data	optimization_input_data for optimization input

Id	Ontology class	Description
C 1.1.1.1.4.2.1.1	BRC_optimization_input_data	optimization_input_data for BRC
C 1.1.1.1.4.2.1.1.1	optimization_input_lambda	optimization_input_lambda for BRC
C 1.1.1.1.4.2.1.2	CONT_optimization_input_data	optimization_input_data for CONT
C 1.1.1.1.4.2.1.2.1	CONT_optimization_input_dynamic_data	CONT optimization input dynamic data
C 1.1.1.1.4.2.1.2.1.1	optimization_input_BreakDownEvents_data	BreakDownEvents_data in CONT Optimization input
C 1.1.1.1.4.2.1.2.1.2	optimization_input_ProductInitialInventory_data	ProductInitialInventory_data in CONT Optimization input
C 1.1.1.1.4.2.1.2.1.3	optimization_input_ProductionOrders_data	ProductionOrders_data in CONT Optimization input
C 1.1.1.1.4.2.1.2.1.4	optimization_input_ResourceInitialInventory_data	ResourceInitialInventory_data in CONT Optimization input
C 1.1.1.1.4.2.1.2.1.5	optimization_input_ScheduledMaintenanceActivities_data	ScheduledMaintenanceActivities_data in CONT Optimization input
C 1.1.1.1.4.2.1.2.1.6	optimization_input_UnScheduledMaintenanceActivities_data	UnScheduledMaintenanceActivities_data in CONT Optimization input
C 1.1.1.1.4.2.1.2.2	CONT_optimization_input_static_data	CONT optimization input static data
C 1.1.1.1.4.2.1.2.2.1	optimization_input_Lines_data	optimization_input_Lines_data for CONT
C 1.1.1.1.4.2.1.2.2.2	optimization_input_LineTypes_data	optimization_input_LineTypes_data for CONT
C 1.1.1.1.4.2.1.2.2.3	optimization_input_ProductFamilies_data	optimization_input_ProductFamilies_d ata for CONT
C 1.1.1.1.4.2.1.2.2.4	optimization_input_Products_data	optimization_input_Products_data for CONT
C 1.1.1.1.4.2.1.2.2.5	optimization_input_Resources_data	optimization_input_Resources_data for CONT
C 1.1.1.1.4.2.1.2.2.6	optimization_input_StorageZones_data	optimization_input_StorageZones_dat a for CONT
C 1.1.1.1.4.2.1.2.2.7	optimization_input_Workplaces_data	optimization_input_Workplaces_data for CONT
C 1.1.1.1.4.2.1.2.2.8	optimization_input_WorkplaceTypes_data	optimization_input_WorkplaceTypes_ data for CONT
C 1.1.1.1.4.2.1.2.2.9	optimization_ProcessingTimes_data	optimization_ProcessingTimes_data for CONT
C 1.1.1.1.4.2.1.2.2.10	optimization_ProductBOM_data	optimization_ProductBOM_data for CONT
C 1.1.1.1.4.2.1.2.2.11	optimization_ResourceBOM_data	optimization_ResourceBOM_data for CONT
C 1.1.1.1.4.2.1.2.2.12	optimization_SetupTimes_data	optimization_SetupTimes_data for CONT
C 1.1.1.1.4.2.1.3	PIA_optimization_input_data	optimization_input_data for PIA
C 1.1.1.1.4.2.1.3.1	optimization_input_currentTotalSetupTime_entity	Current total setup time for optimization input
C 1.1.1.1.4.2.1.3.2	optimization_input_historicalDataWeekly_ entity	Historical Data Weekly for optimization input
C 1.1.1.1.4.2.1.3.3	optimization_input_loom_entity	PIA Loom for optimization input
C 1.1.1.1.4.2.1.3.4	optimization_input_startDate_entity	Start Date for optimization input
C 1.1.1.1.4.2.1.3.5	optimization_input_workGroupsPerShift_entity	Work Groups Per Shift for optimization input
C 1.1.1.1.4.2.1.4	Tupras_optimization_input_data	optimization_input_data for Tupras
C 1.1.1.1.4.2.1.4.1	optimization_input_OS	optimization_input_OS for Tupras optimization input
C 1.1.1.1.4.2.1.4.1.1	linkedScenario	linkedScenario for Tupras optimization

Id	Ontology class	Description
		input
C 1.1.1.1.4.2.1.4.1.2	UnitScenario	UnitScenario for Tupras optimization input
C 1.1.1.1.4.2.1.4.2	optimization_input_PI	optimization_input_PI for Tupras optimization input
C 1.1.1.1.4.2.1.4.2.1	optimization_input_InputFeed_entity	InputFeed for Tupras optimization input
C 1.1.1.1.4.2.1.4.2.2	optimization_input_OutputTanks	OutputTanks for Tupras optimization input
C 1.1.1.1.4.2.1.4.2.3	optimization_input_Settings	Settings for Tupras optimization input
C 1.1.1.1.4.2.1.4.2.4	optimization_input_Specs	Specs for Tupras optimization input
C 1.1.1.1.4.2.1.4.3	optimization_input_PM	optimization_input_PM for Tupras optimization input
C 1.1.1.1.4.2.2	optimization_input_route	optimization_input_route for optimization input
C 1.1.1.1.4.3	optimization_output_entity	general entity for optimization output
C 1.1.1.1.4.3.1	opmitimization_output_produced_at_entity	produced_at_entity parameter for opmitimization output
C 1.1.1.1.4.3.2	optimization_ontput_uuid	Uuid in optimization ontput file
C 1.1.1.1.4.3.3	optimization_output_data	Optimization output related data
C 1.1.1.1.4.3.3.1	BRC_optimization_output_data	Optimization output related data for BRC
C 1.1.1.1.4.3.3.1.1	BRC_optimization_output_objective_values	BRC_optimization_output_objective_value s for optimization output
C 1.1.1.1.4.3.3.1.1.1	BRC_optimization_output_objective_Makespa n	BRC_optimization_output_objective_Makes pan parameter for optimization output
C 1.1.1.1.4.3.3.1.1.2	BRC_optimization_output_objective_TotalLat eness	BRC_optimization_output_objective_TotalL ateness parameter for optimization output
C 1.1.1.1.4.3.3.1.1.3	BRC_optimization_output_objective_TotalTar diness	BRC_optimization_output_objective_TotalT ardiness parameter for optimization output
C 1.1.1.1.4.3.3.1.2	BRC_optimization_output_output_job	BRC_optimization_output_output_job parameter for optimization output
C 1.1.1.1.4.3.3.1.3	BRC_optimization_output_output_job_co mpletion_Time	BRC_optimization_output_output_job_comp letion_Time parameter for optimization output
C 1.1.1.1.4.3.3.1.4	BRC_optimization_output_output_job_start_Ti me	BRC_optimization_output_output_job_start_ Time parameter for optimization output
C 1.1.1.1.4.3.3.1.5	BRC_optimization_output_output_order	BRC_optimization_output_output_ord er parameter for optimization output
C 1.1.1.1.4.3.3.1.6	BRC_optimization_output_output_order_sequ ence	BRC_optimization_output_output_ord er_sequence parameter for optimization output
C 1.1.1.1.4.3.3.2	CONT_optimization_output_data	Optimization output related data for CONT
C 1.1.1.1.4.3.3.2.1	CONT_optimization_output_Metrics_data	CONT_optimization_output_Metrics_data for CONT optimization output
C 1.1.1.1.4.3.3.2.1.1	CONT_AvgWorkStationIdleTime	CONT_AvgWorkStationIdleTime
C 1.1.1.1.4.3.3.2.1.2	CONT_Makespan	CONT_Makespan for CONT optimization output
C 1.1.1.1.4.3.3.2.1.3	CONT_TotalTardiness	CONT_TotalTardiness for CONT optimization output
C 1.1.1.1.4.3.3.2.2	CONT_optimization_output_Schedule	CONT_optimization_output_Schedule for CONT
C 1.1.1.1.4.3.3.3	PIA_optimization_output_data	Optimization output related data for PIA

Id	Ontology class	Description
C 1.1.1.1.4.3.3.3.1	PIA_optimization_output_objective_values	PIA optimization output objective values
C 1.1.1.1.4.3.3.3.1.1	PIA_loom_order_sequence	PIA loom order sequence
C 1.1.1.1.4.3.3.3.1.2	PIA_optimization_ouput_totalCompletionTimes	PIA optimization ouput totalCompletionTimes
C 1.1.1.1.4.3.3.3.1.3	PIA_optimization_ouput_totalEnergyCons	PIA optimization ouput totalEnergyCons
C 1.1.1.1.4.3.3.3.1.4	PIA_optimization_ouput_totalTardiness	PIA optimization ouput totalTardiness
C 1.1.1.1.4.3.3.3.1.5	PIA_optmization_output_makespan	PIA optmization output makespan
C 1.1.1.1.4.3.3.4	tupras_optimization_output_data	Optimization output related data for Tupras
C 1.1.1.1.4.3.3.4.1	optimization_output_output_OutputKPIs	Optimization output OutputKPIs
C 1.1.1.1.4.3.3.4.1.1	optimization_output_output_C2C5perc	C2C5perc for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.1.2	optimization_output_output_C2perc	C2perc for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.1.3	optimization_output_output_C5perc	C5perc for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.1.4	optimization_output_output_OutputNodeID	OutputNodeID for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.1.5	optimization_output_output_Quantity	Quantity for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.1.6	optimization_output_output_SUperc	SUperc for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.2	optimization_output_output_SolKPIs	Optimization output SolKPIs
C 1.1.1.1.4.3.3.4.2.1	optimization_output_output_found_solution	found_solution for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.2.2	optimization_output_output_TimeToInitialize Millisec	TimeToInitializeMillisec for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.2.3	optimization_output_output_TimeToSolveMilli sec	TimeToSolveMillisec for TUPRAS optimization output
C 1.1.1.1.4.3.3.4.3	optimization_output_output_total_energy	optimization output total energy
C 1.1.1.1.4.3.3.4.4	optimization_output_SolutionScenarios	Optimization output SolutionScenarios
C 1.1.1.1.5	Pilot_parameter_concept	FactlogDefintion "a virtual entity to define Factlog pilot parameters."
C 1.1.1.1.5.1	BRC_pilot_parameter	Pilot specific parameter for BRC
C 1.1.1.1.5.1.1	Batch_code	Batch_code parameter for BRC
C 1.1.1.1.5.1.2	BoM	BoM parameter for BRC
C 1.1.1.1.5.1.3	BRC_data_source	BRC_data_source parameter for BRC
C 1.1.1.1.5.1.4	Cycle_Start_time	Cycle_Start_time parameter for BRC
C 1.1.1.1.5.1.5	Machine_downtime	Machine_downtime parameter for BRC
C 1.1.1.1.5.1.6	Machine_highlight_issue	Machine_highlight_issue parameter for BRC
C 1.1.1.1.5.1.7	shift_schedule	shift_schedule parameter for BRC
C 1.1.1.1.5.2	CONT_pilot_parameter	Pilot specific parameter for CONT
C 1.1.1.1.5.2.1	Measured_value_of_EE_compent	Measured_value_of_EE_compent parameter for CONT
C 1.1.1.1.5.2.2	Process_parameter	Process_parameter for CONT
C 1.1.1.1.5.2.3	Production_reaction_in_temp	Production_reaction_in_temp for CONT
C 1.1.1.1.5.2.4	Wearing_of_tools	Wearing_of_tools for CONT
C 1.1.1.1.5.3	General_sensor_data	Sensor data for all the pilots
C 1.1.1.1.5.3.1	Analog_Signals	Analog_Signals from sensor
C 1.1.1.1.5.3.1.1	BRC_analog_signal	Analog_signal from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.1	Cycle_Time	Cycle_Time from sensors in the BRC pilot

Id	Ontology class	Description
C 1.1.1.1.5.3.1.1.2	Hydraulic_pressure	Hydraulic_pressure from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.3	Hydraulic_Temp	Hydraulic_Temp from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.4	main_frequency	main_frequency from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.5	Mains_current	Mains_current from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.6	Mains_voltages	Mains_voltages from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.7	Roller_wear	Roller_wear from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.8	Step_angle_bent	Step_angle_bent from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.9	Step_Feed_length	Step_Feed_length from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.10	Step_power	Step_power from sensors in the BRC pilot
C 1.1.1.1.5.3.1.1.11	Step_time	Step_time from sensors in the BRC pilot
C 1.1.1.1.5.3.2	Digital_Signals	Digital_Signals from sensor
C 1.1.1.1.5.3.2.1	BRC_digital_signal	Digital_Signals from sensor in BRC
C 1.1.1.1.5.3.2.1.1	Automatic_Operation	Automatic_Operation from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.2	Chiller_Overload_timestamp	Chiller_Overload_timestamp from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.3	Hanger_Motor_Overload_timestamp	Hanger_Motor_Overload_timestamp from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.4	Hydraulic_Cooler_ON_time	Hydraulic_Cooler_ON_time from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.5	Hydraulic_Oil_level_switch_change_of_state time	Hydraulic_Oil_level_switch_change_of_state_time from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.6	Safety_Relay_closed	Safety_Relay_closed from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.7	Shears_Closed_timestamp	Shears_Closed_timestamp from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.8	Shears_Cut_timestamp	Shears_Cut_timestamp from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.9	Shears_Open_timestamp	Shears_Open_timestamp from sensors in the BRC pilot
C 1.1.1.1.5.3.2.1.10	SMU_Motor_Overload	SMU_Motor_Overload from sensors in the BRC pilot
C 1.1.1.1.5.3.2.2	JEMS_digital_signal	Digital_Signals from sensor in JEMS
C 1.1.1.1.5.3.2.2.1	JEMS_Ingested_material_flow_speed	JEMS_Ingested_material_flow_speed from sensors in the JEMS pilot
C 1.1.1.1.5.3.2.2.2	JEMS_Mixing_power	JEMS_Mixing_power from sensors in the JEMS pilot
C 1.1.1.1.5.3.2.2.3	JEMS_Pressure	JEMS_Pressure from sensors in the JEMS pilot
C 1.1.1.1.5.3.2.2.4	JEMS_Pump_speed	JEMS_Pump_speed from sensors in the JEMS pilot
C 1.1.1.1.5.3.2.2.5	JEMS_Temperature	JEMS_Temperature from sensors in the JEMS pilot
C 1.1.1.1.5.3.2.2.6	JEMS_Turbine_flow	JEMS_Turbine_flow from sensors in the JEMS pilot
C 1.1.1.1.5.3.2.3	PIA_digital_signal	Digital_Signals from sensor in PIA

Id	Ontology class	Description
C 1.1.1.1.5.3.2.3.1	Mechanical_lock	Mechanical_lock from sensor in PIA
C 1.1.1.1.5.3.2.4	Tupras _digital_signal	Digital_Signals from sensor in Tupras
C 1.1.1.1.5.3.2.4.1	flow	Flow from sensor in Tupras
C 1.1.1.1.5.3.2.4.2	level	Level from sensor in Tupras
C 1.1.1.1.5.3.2.4.3	pressure	Pressure from sensor in Tupras
C 1.1.1.1.5.3.2.4.4	temperature_from_sensor	temperature_from_sensor from sensor in Tupras
C 1.1.1.1.5.4	JEMS_pilot_parameter	Pilot specific parameter for JEMS
C 1.1.1.1.5.4.1	Ingested_material_flow_speed	Ingested_material_flow_speed parameter for JEMS
C 1.1.1.1.5.4.2	JEMS_temperature	JEMS_temperature parameter for JEMS
C 1.1.1.1.5.4.3	Mixing_power	Mixing_power parameter for JEMS
C 1.1.1.1.5.4.4	Pressure	Pressure parameter for JEMS
C 1.1.1.1.5.4.5	Pump_speed	Pump_speed parameter for JEMS
C 1.1.1.1.5.4.6	Turbine_flow	Turbine_flow parameter for JEMS
C 1.1.1.1.5.5	PIA_pilot_parameter	Pilot specific parameter for PIA
C 1.1.1.1.5.5.1	Compressor	Compressor parameter for PIA
C 1.1.1.1.5.5.2	Fabric_family	Fabric_family parameter for PIA
C 1.1.1.1.5.5.3	Fabric_grams_per_linear_meter	Fabric_grams_per_linear_meter parameter for PIA
C 1.1.1.1.5.5.4	Humidification	Humidification parameter for PIA
C 1.1.1.1.5.5.5	HVAC_setup	HVAC_setup parameter for PIA
C 1.1.1.1.5.5.6	Lighting	Lighting parameter for PIA
C 1.1.1.1.5.5.7	Loom_smacks_per_meter	Loom_smacks_per_meter parameter for PIA
C 1.1.1.1.5.5.8	Maintenance_consumption	Maintenance_consumption parameter for PIA
C 1.1.1.1.5.5.9	Number_of_yarns	Number_of_yarns parameter for PIA
C 1.1.1.1.5.5.10	Operating_speed_condition	Operating_speed_condition parameter for PIA
C 1.1.1.1.5.5.11	Process_setup	Process_setup parameter for PIA
C 1.1.1.1.5.6	Reference_data_for_anomaly_analysis	Reference data for anomaly analysis
C 1.1.1.1.5.7	Tupras_pilot_parameter	Pilot specific parameter for Tupras
C 1.1.1.1.5.7.1	DEA/merox_column	DEA/merox_column parameter for Tupras
C 1.1.1.1.5.7.2	Debutanizer_column	Debutanizer_column parameter for Tupras
C 1.1.1.1.5.7.3	Deethanizer_column	Deethanizer_column parameter for Tupras
C 1.1.1.1.5.7.4	LPG_recovery	LPG_recovery parameter for Tupras
C 1.1.1.1.5.7.5	Parameter_for_unit_control	Parameter_for_unit_control parameter for Tupras
C 1.1.1.1.5.7.5.1	temperature_for_control	temperature_for_control parameter for Tupras
C 1.1.1.1.5.7.6	parameter_for_unit_estimate	parameter_for_unit_estimate parameter for Tupras
C 1.1.1.1.5.7.6.1	temperature_for_estimate	temperature_for_estimate parameter for Tupras
C 1.1.1.1.5.7.7	tupras_data_source	tupras_data_source parameter for Tupras
C 1.1.1.1.6	Process_model_entity	FactlogDefintion "A virtual concepts used to define process model."
C 1.1.1.1.6.1	process_model	FactlogDefintion "process model concepts used in Factlog"

Id	Ontology class	Description
C 1.1.1.1.6.1.1	Discrete_event_simulation_model	Process modeling model for Factlog
C 1.1.1.1.6.1.2	Petri_net_model	PN model used for Factlog
C 1.1.1.1.6.2	process_model_element	FactlogDefintion "process model element concepts used in Factlog"
C 1.1.1.1.7	Simulation_model_entity	FactlogDefintion "a virtual concepts used to define simulation model concepts"
C 1.1.1.1.7.1	Petri_net_model_element_entity	Model elements for PN models
C 1.1.1.1.7.1.1	General_PN_entity	General model information for PN
C 1.1.1.1.7.1.1.1	PN_result	PN simulation result
C 1.1.1.1.7.1.1.2	Tokens	tokens in PN models
C 1.1.1.1.7.1.2	PN_connection_entity	General model element connection for PN
C 1.1.1.1.7.1.2.1	Inhibitor_arcs	Inhibitor_arcs connection for PN
C 1.1.1.1.7.1.2.2	Test_arcs	Test_arcs connection for PN
C 1.1.1.1.7.1.2.3	Usual_arcs	Usual_arcs connection for PN
C 1.1.1.1.7.1.3	PN_node_entity	General model element for PN
C 1.1.1.1.7.1.3.1	Continuous_places	Continuous_places node in PN models
C 1.1.1.1.7.1.3.2	Continuous_transitions	Continuous_transitions node in PN models
C 1.1.1.1.7.1.3.3	Discrete_Places	Discrete_Places node in PN models
C 1.1.1.1.7.1.3.4	Discrete_Transitions	Discrete_Transitions node in PN models
C 1.1.1.1.7.2	PM_element_entity	Model elements for discrete event simulation models
C 1.1.1.1.7.2.1	PM_link	Connections in PM models
C 1.1.1.1.7.2.1.1	Clean_DEA_flow	Clean_DEA_flow connections in PM models
C 1.1.1.1.7.2.1.2	Compressed_Gas_flow	Compressed_Gas_flow connections in PM models
C 1.1.1.1.7.2.1.3	Gas_flow	Gas_flow connections in PM models
C 1.1.1.1.7.2.1.4	LPG_C5_C2_H2S_flow	LPG_C5_C2_H2S_flow connections in PM models
C 1.1.1.1.7.2.1.5	LPG_CS_C2_H2S	LPG_CS_C2_H2S connections in PM models
C 1.1.1.1.7.2.1.6	LPG_flow	LPG_flow connections in PM models
C 1.1.1.1.7.2.1.7	LPG_S_flow	LPG_S_flow connections in PM models
C 1.1.1.1.7.2.1.8	SourGas_flow	SourGas_flow connections in PM models
C 1.1.1.1.7.2.1.9	Steam	Steam connections in PM models
C 1.1.1.1.7.2.2	PM_node	Nodes in PM models
C 1.1.1.1.7.2.2.1	DEA_Regen_component	DEA_Regen_component Nodes in PM models
C 1.1.1.1.7.2.2.2	Debutanizer_component	Debutanizer_component Nodes in PM models
C 1.1.1.1.7.2.2.3	LPG_component	LPG_component Nodes in PM models
C 1.1.1.1.7.2.2.4	LPG_DEA_component	LPG_DEA_component Nodes in PM models
C 1.1.1.1.7.2.2.5	Merge_component	Merge_component Nodes in PM models
C 1.1.1.1.7.2.2.6	Off_Gas_compressor_component	Off_Gas_compressor_component Nodes in PM models
C 1.1.1.1.7.2.2.7	SulphUnit_component	SulphUnit_component Nodes in PM models
C 1.1.1.1.7.2.2.8	Sulphur_Unit_component	Sulphur_Unit_component Nodes in PM models
C 1.1.1.1.7.2.2.9	Tank_component	Tank_component Nodes in PM models
C 1.1.1.1.7.2.3	PM_parameter	Parameter in PM models

Id	Ontology class	Description
C 1.1.1.1.7.2.4	PM_resource	Resource in PM models
C 1.1.1.1.7.2.5	PM_results	Results generated by PM models
C 1.1.1.1.7.2.6	PM_Tolerance	Tolerance set for PM simulation
C 1.1.1.1.7.3	PM_element_related_entity	Related information concepts for model elements for PM models
C 1.1.1.1.7.3.1	PM_flow_related_entity	Information concept related to PM flow
C 1.1.1.1.7.3.1.1	flow_caculated	flow_caculated set for PM flow
C 1.1.1.1.7.3.1.2	flow_factor	flow_factor set for PM flow
C 1.1.1.1.7.3.1.3	flow_formula	flow_formula set for PM flow
C 1.1.1.1.7.3.1.4	flow_manual	flow_manual set for PM flow
C 1.1.1.1.7.3.1.5	flow_name	flow_name set for PM flow
C 1.1.1.1.7.3.1.6	flow_quantity	flow_quantity set for PM flow
C 1.1.1.1.7.3.2	PM_link_related_entity	Information concept related to PM link
C 1.1.1.1.7.3.2.1	PM_link_description	PM_link_description set for PM link
C 1.1.1.1.7.3.2.2	PM_link_id	PM_link_id set for PM link
C 1.1.1.1.7.3.2.3	PM_link_name	PM_link_name set for PM link
C 1.1.1.1.7.3.2.4	PM_link_source	PM_link_source set for PM link
C 1.1.1.1.7.3.2.5	PM_link_target	PM_link_target set for PM link
C 1.1.1.1.7.3.3	PM_node_related_entity	Information concept related to PM node
C 1.1.1.1.7.3.3.1	PM_node_description	PM_node_description set for PM node
C 1.1.1.1.7.3.3.2	PM_node_id	PM_node_id set for PM node
C 1.1.1.1.7.3.3.3	PM_node_name	PM_node_name set for PM node
C 1.1.1.1.7.3.3.4	PM_node_one2one	PM_node_one2one set for PM node
C 1.1.1.1.7.3.3.5	PM_node_parameter_entity	PM_node_parameter_entity set for PM node
C 1.1.1.1.7.3.3.6	PM_node_script_source	PM_node_script_source set for PM node
C 1.1.1.1.7.3.3.7	PM_node_solved	PM_node_solved set for PM node
C 1.1.1.1.7.3.3.8	PM_node_specificationMethod	PM_node_specificationMethod set for PM node
C 1.1.1.1.7.3.3.9	PM_node_stage	PM_node_stage set for PM node
C 1.1.1.1.7.3.3.10	PM_node_stock	PM_node_stock set for PM node
C 1.1.1.1.7.3.3.11	PM_node_type	PM_node_type set for PM node
C 1.1.1.1.7.3.4	PM_parameter_related_entity	Information concept related to PM parameter
C 1.1.1.1.7.3.4.1	PM_parameter_description	PM_parameter_description set for PM parameter
C 1.1.1.1.7.3.4.2	PM_parameter_Symbol	PM_parameter_Symbol set for PM parameter
C 1.1.1.1.7.3.4.3	PM_parameter_unit	PM_parameter_unit set for PM parameter
C 1.1.1.1.7.3.4.4	PM_parameter_value	PM_parameter_value set for PM parameter
C 1.1.1.1.7.3.5	PM_related_entity	Information concept related to PM model
C 1.1.1.1.7.3.5.1	PM_id	PM_id set for PM model
C 1.1.1.1.7.3.5.2	PM_Name	PM_Name set for PM model
C 1.1.1.1.7.3.6	PM_resource_related_entity	Information concept related to PM resource
C 1.1.1.1.7.3.6.1	PM_resource_description	PM_resource_description set for PM resoource
C 1.1.1.1.7.3.6.2	PM_resource_id	PM_resource_id set for PM resoource

Id	Ontology class	Description
C 1.1.1.1.7.3.6.3	PM_resource_name	PM_resource_name set for PM resoource
C 1.1.1.1.7.3.6.4	PM_resource_unit	PM_resource_unit set for PM resoource
C 1.1.1.1.7.4	simulation_model	Factlog simulation model concept
C 1.1.1.1.7.4.1	JEMS_simulation_model	Simulation model concept used for JEMS
C 1.1.1.1.7.4.1.1	Machine_learning_model	Machine_learning_model used in JEMS
C 1.1.1.1.7.4.1.1.1	Batch_model	Batch_model used in JEMS
C 1.1.1.1.7.4.1.1.1.1	Logistic_regression_model	Logistic_regression_model used in JEMS
C 1.1.1.1.7.4.1.1.1.2	LSTM_model	LSTM_model used in JEMS
C 1.1.1.1.7.4.1.1.1.3	RNN_model	RNN_model used in JEMS
C 1.1.1.1.7.4.1.1.1.4	SVM_model	SVM_model used in JEMS
C 1.1.1.1.7.4.1.1.2	Streaming_model	Streaming_model used in JEMS
C 1.1.1.1.7.4.1.1.2.1	FIMT-DD_model	FIMT-DD_model used in JEMS
C 1.1.1.1.7.4.1.1.2.2	Hoeffding_trees_model	Hoeffding_trees_model used in JEMS
C 1.1.1.1.7.4.1.2	Probabilistic_model	Probabilistic_model used in JEMS
C 1.1.1.1.7.4.1.2.1	Monte_carlo_simulation_model	Monte_carlo_simulation_model used in JEMS
C 1.1.1.1.7.4.1.2.2	Probabilistic_soft_logic_model	Probabilistic_soft_logic_model used in JEMS
C 1.1.1.1.7.4.2	Tupras_simualtion_model	Simulation model concept used for TUPRAS
C 1.1.1.1.7.4.2.1	regression_model	Regression_model used in TUPRAS
C 1.1.2	independent_continuant	A continuant which is such that there is no c and not such that b s-depends_on c at t
C 1.1.2.1	immaterial_entity	Immaterial entities are divided into two subgroups:boundaries and sites, which bound, or are demarcated in relation, to material entities, and which can thus change location, shape and size and as their material hosts move or change shape or size (for example: your nasal passage; the hold of a ship; the boundary of Wales (which moves with the rotation of the Earth)
C 1.1.2.1.1	continuant_fiat_boundary	A continuant fiat boundary is a boundary of some material entity
C 1.1.2.1.1.1	one-dimensional_continuant_fiat_boundary	IAO_0000600 "a one-dimensional continuant fiat boundary is a continuous fiat line whose location is defined in relation to some material entity. (axiom label in BFO2 Reference: [032-001])"@en
C 1.1.2.1.1.2	two-dimensional_continuant_fiat_boundary	IAO_0000600 "a two-dimensional continuant fiat boundary (surface) is a self-connected fiat surface whose location is defined in relation to some material entity. (axiom label in BFO2 Reference: [033-001])"@en

Id	Ontology class	Description
C 1.1.2.1.1.3	zero-dimensional_continuant_fiat_boundary	IAO_0000116 "zero dimension continuant fiat boundaries are not spatial points. Considering the example 'the quadripoint where the boundaries of Colorado, Utah, New Mexico, and Arizona meet': There are many frames in which that point is zooming through many points in space. Whereas, no matter what the frame, the quadripoint is always in the same relation to the boundaries of Colorado, Utah, New Mexico, and Arizona."@en
C 1.1.2.1.2	site	A three-dimensional immaterial entity that is (partially or wholly) bounded by a material entity or it is a three-dimensional immaterial part thereof.
C 1.1.2.1.2.1	BRC_site	Site used in TUPRAS
C 1.1.2.1.2.1.1	Coil_stock	Site which the BRC Coil_stock locates in
C 1.1.2.1.2.1.2	Finish_product_laydown_area	The place which the BRC Finish_product_laydown_area locates in
C 1.1.2.1.2.1.3	Finished_product_trailers	Site which the BRC Finished_product_trailers locate in
C 1.1.2.1.2.1.4	Industrial_site	the BRC Industrial_site locates in
C 1.1.2.1.2.1.5	Maintenance_department_site	Site which the BRC Maintenance_department locates in
C 1.1.2.1.2.1.6	Offcut_area	Site which the BRC Offcut_area locates in
C 1.1.2.1.2.1.7	Pilot_area	Site which the BRC Pilot_area locates in
C 1.1.2.1.2.1.8	Straight_rebar_stock	Site which the BRC Straight_rebar_stock locates in
C 1.1.2.1.2.2	CONT_site	Site used in CONT
C 1.1.2.1.2.2.1	station_site	Site which the CONT station locates in
C 1.1.2.1.2.2.2	unit_site	Site which the CONT unit locates in
C 1.1.2.1.2.3	TUPRAS_site	Site used in TUPRAS
C 1.1.2.1.2.3.1	workstation	Site which the TUPRAS workstation locates in
C 1.1.2.1.3	spatial_region	All continuant parts of spatial regions are spatial regions.
C 1.1.2.1.3.1	one-dimensional_spatial_region	IAO_0000600 "A one-dimensional spatial region is a line or aggregate of lines stretching from one point in space to another. (axiom label in BFO2 Reference: [038-001])"@en
C 1.1.2.1.3.2	three-dimensional_spatial_region	IAO_0000600 "A three-dimensional spatial region is a spatial region that is of three dimensions. (axiom label in BFO2 Reference: [040-001])"@en
C 1.1.2.1.3.3	two-dimensional_spatial_region	IAO_0000600 "A two-dimensional spatial region is a spatial region that is of two dimensions. (axiom label in BFO2 Reference: [039-001])"@en
C 1.1.2.1.3.4	zero-dimensional_spatial_region	IAO_0000600 "A zero-dimensional spatial region is a point in space. (axiom label in BFO2 Reference: [037-001])"@en
C 1.1.2.2	material_entity	An independent continuant that has some portion of matter as proper or improper continuant part.

Id	Ontology class	Description
C 1.1.2.2.1	fiat_object_part	A material entity which is such that for all times t, if b exists at t then there is some object c such that b proper continuant_part of c at t and c is demarcated from the remainder of c by a two-dimensional continuant fiat boundary.
C 1.1.2.2.2	object	A material entity which manifests causal unity of one or other of the types CUn listed
		above & is of a type (a material universal) instance of which are maximal relative to this criterion of causal unity.
C 1.1.2.2.2.1	artifact	An Object that was designed by some Agent to realize a certain Function.
C 1.1.2.2.2.1.1	Equipment	The set of physical resources serving to equip a person or thing implementing used in an operation or activity"
C 1.1.2.2.2.1.1.1	TUPRAS_Equipment	A physical system using power to apply forces and control movement to perform an action in TUPRAS.
C 1.1.2.2.2.1.1.1.1	Pipeline	A pipeline transports crude oil or natural gas liquids.
C 1.1.2.2.2.1.1.1.2	Tank	A tank for oil storing.
C 1.1.2.2.2.1.2	Machine	A physical system using power to apply forces and control movement to perform an action.
C 1.1.2.2.2.1.2.1	Factlog_BRC_machine	A physical system using power to apply forces and control movement to perform an action for BRC.
C 1.1.2.2.2.1.2.1.1	Automatic_double_binder_machine	Automatic double binder machine for BRC
C 1.1.2.2.2.1.2.1.2	Coil_machine	Coil machine for BRC
C 1.1.2.2.2.1.2.1.3	Cut_machine	Cut machine for BRC
C 1.1.2.2.2.1.2.1.4	Large_coil_machine	Large coil machine for BRC
C 1.1.2.2.2.1.2.1.5	Lemton_machine	Lemton machine for BRC
C 1.1.2.2.2.1.2.1.6	Manual_bending_machine	Manual bending machine for BRC
C 1.1.2.2.2.1.2.1.7	New_shearline_machine	New shearline machine for BRC
C 1.1.2.2.2.1.2.1.8	Radius_bending_machine	Radius bending machine for BRC
C 1.1.2.2.2.1.2.1.9	Saw_machine	Saw machine for BRC
C 1.1.2.2.2.1.2.1.10	Shearline_machine	Shearline machine for BRC
C 1.1.2.2.2.1.2.1.11	Straightener_machine	Straightener machine for BRC
C 1.1.2.2.2.1.2.2	Factlog_CONT_machine	A physical system using power to apply forces and control movement to perform an action for CONT.
C 1.1.2.2.2.1.2.2.1	CarMa_machine	CarMa machine for CONT
C 1.1.2.2.2.1.2.2.2	EmaC_machine	EmaC machine for CONT
C 1.1.2.2.2.1.2.2.3	Evaprod_machine	Evaprod machine for CONT
C 1.1.2.2.2.1.2.2.4	Flipping_Screwing_machine	Flipping Screwing machine for CONT
C 1.1.2.2.2.1.2.2.5	IdGen_machine	IdGen machine for CONT
C 1.1.2.2.2.1.2.2.6	MaMa_machine	MaMa machine for CONT
C 1.1.2.2.2.1.2.2,7	MPM_machine	MPM machine for CONT
C 1.1.2.2.2.1.2.2.8	Pulse_machine	Pulse machine for CONT
C 1.1.2.2.2.1.2.2.9	WIP_machine	WIP machine for CONT
C 1.1.2.2.2.1.2.3	Factlog_PIA_machine	A physical system using power to apply

Id	Ontology class	Description
		forces and control movement to perform an action for PIA.
C 1.1.2.2.2.1.2.3.1	Heating_machine	Heating machine for PIA
C 1.1.2.2.2.1.2.3.2	HVAC_system	HVAC system for PIA
C 1.1.2.2.2.1.2.3.3	Weaving_machine	Weaving _machine for PIA
C 1.1.2.2.2.1.3	Machine_component	Compositions for constructing machines.
C 1.1.2.2.2.1.3.1	Factlog_PIA_machine_component	Compositions for constructing machines in PIA.
C 1.1.2.2.2.1.3.1.1	Electric_engine	Electric_engine used in PIA machine
C 1.1.2.2.2.1.3.1.2	Electric_motor	Electric_ motor used in PIA machine
C 1.1.2.2.2.1.4	Processing_stock	An artifact in an industrial site corresponds to any material in the process of producing or manufacturing finished product.
C 1.1.2.2.2.1.4.1	input_processing_stock	IAO_0000115 "is a processing stock that is a result of production or production or manufacturing process."
C 1.1.2.2.2.1.4.1.1	Factlog_BRC_input_process_stock	A processing stock that is a result of production or production or manufacturing process in BRC.
C 1.1.2.2.2.1.4.1.1.1	BRC_bar	Bar in BRC
C 1.1.2.2.2.1.4.1.1.2	BRC_coil	Coil in BRC
C 1.1.2.2.2.1.4.1.2	Factlog_JEMS_input_process_stock	A processing stock that is a result of production or production or manufacturing process in JEMS.
C 1.1.2.2.2.1.4.1.2	JEMS_Feedstock	Feedstock in JEMS
C 1.1.2.2.2.1.4.1.3	Factlog_PIA_input_processing_stock	A processing stock that is a result of production or production or manufacturing process in PIA.
C 1.1.2.2.2.1.4.1.3.1	Fabric	Fabric in PIA
C 1.1.2.2.2.1.4.1.4	Factlog_tupras_input_processing_stock	A processing stock that is a result of production or production or manufacturing process in TUPRAS.
C 1.1.2.2.2.1.4.1.4.1	Hydrogen_sulphur	Hydrogen_sulphur for TUPRAS process
C 1.1.2.2.2.1.4.1.4.2	Mercaptan	Mercaptan for TUPRAS process
C 1.1.2.2.2.1.4.1.4.3	Pentane	Pentane for TUPRAS process
C 1.1.2.2.2.1.4.1.4.4	Sulphur	Sulphur for TUPRAS process
C 1.1.2.2.2.1.4.2	output_processing_stock	IAO_0000115 "is a processing stock to be fed into a production or manufacturing process."
C 1.1.2.2.2.1.4.2.1	Factlog_JEMS_output_processing_stock	Output stock generated by PIA process
C 1.1.2.2.2.1.4.2.1.1	JEMS_CO2	CO2 generated by JEMS process
C 1.1.2.2.2.1.4.2.1.2	JEMS_Diesel	Diesel generated by JEMS process
C 1.1.2.2.2.1.4.2.1.3	JEMS_Oil	Oil generated by JEMS process
C 1.1.2.2.2.1.4.2.1.4	JEMS_synthesis_fuel	Synthesis fuel generated by JEMS process
C 1.1.2.2.2.1.4.2.1.5	JEMS_water	Water generated by JEMS process
C 1.1.2.2.2.1.4.2.2	Factlog_PIA_output_processing_stock	Output stock generated by PIA process
C 1.1.2.2.2.1.4.2.2.1	Pia_final_product	Final_product generated by PIA process
C 1.1.2.2.2.1.4.2.2.2	PIA_gas	Gas generated by PIA process
C 1.1.2.2.2.1.4.2.2.3	PIA_heat	Heat generated by PIA process
C 1.1.2.2.2.1.4.2.2.4	PIA_stream	Stream generated by PIA process
C 1.1.2.2.2.1.4.2.2.5	PIA_wasted_energy	Wasted_energy generated by PIA process

Id	Ontology class	Description
C 1.1.2.2.2.1.4.2.3	Factlog_tupras_output_processing_stock	Output stock generated by TUPRAS process
C 1.1.2.2.2.1.4.2.3.1	Diesel	Diesel generated by TUPRAS process
C 1.1.2.2.2.1.4.2.3.2	Fuel_oil	Fuel_oil generated by TUPRAS process
C 1.1.2.2.2.1.4.2.3.3	Gasoline	Gasoline generated by TUPRAS process
C 1.1.2.2.2.1.4.2.3.4	LPG_product	LPG_product generated by TUPRAS process
C 1.1.2.2.2.1.4.2.3.5	Naphtha	Naphtha generated by TUPRAS process
C 1.1.2.2.2.1.5	Sensor	A device that produces an output signal for the purpose of sensing of a physical phenomenon.
C 1.1.2.2.2.1.5.1	Factlog_JEMS_sensor	Sensors used in JEMS
C 1.1.2.2.2.1.5.1.1	JEMS_pressure_sensor	Pressure sensor used in JEMS
C 1.1.2.2.2.1.5.1.2	JEMS_temperature_sensor	Temperature sensor used in JEMS
C 1.1.2.2.2.1.5.2	Factlog_PIA_sensor	Sensors used in PIA
C 1.1.2.2.2.1.5.3	Factlog_TUPRAS_sensor	Sensors used in TUPRAS
C 1.1.2.2.2.1.5.2.1	TUPRAS_temperature_sensor	Temperature Sensors used in TUPRAS
C 1.1.2.2.2.2	person	An object that is a human being.
C 1.1.2.2.3	object_aggregate	A material entity consisting exactly of a plurality of objects as member_parts at all times at which b exists.
C 1.1.2.2.3.1	artifact_aggregate	A collection of artifacts that designed or arranged by some Agent to realize a certain Function.
C 1.1.2.2.3.1.1	BRC_unit	A company group generally equivalent in size and character to implement BRC services.
C 1.1.2.2.3.1.1.1	BRC_plant	Production plant used in BRC
C 1.1.2.2.3.1.1.2	BRC_production_line	Production line used in BRC
C 1.1.2.2.3.1.2	CONT_unit	A company group generally equivalent in size and character to implement CONT services.
C 1.1.2.2.3.1.2.1	CONT_plant	Production plant used in CONT
C 1.1.2.2.3.1.2.2	CONT_production_line	Production line used in CONT
C 1.1.2.2.3.1.2.2.1	FA_and_test_area	FA_and_test_area in CONT
C 1.1.2.2.3.1.2.2.2	Pakcaging_and_delivery_operation	Packaging and delivery operation line used in CONT
C 1.1.2.2.3.1.2.2.3	PCBA	PCBA line used in CONT
C 1.1.2.2.3.1.2.2.4	SMT_line	SMT line used in CONT
C 1.1.2.2.3.1.3	JEMS_unit	A company group generally equivalent in size and character to implement JEMS services.
C 1.1.2.2.3.1.3.1	JEMS_plant	Production plant used in JEMS
C 1.1.2.2.3.1.3.2	JEMS_production_line	Production line used in JEMS
C 1.1.2.2.3.1.4	PIA_unit	A company group generally equivalent in size and character to implement PIA services.
C 1.1.2.2.3.1.4.1	Pia_production_line	Production plant used in PIA
C 1.1.2.2.3.1.4.2	Pia_production_plant	Production line used in PIA
C 1.1.2.2.3.1.4.3	Pia_production_unit_controller	Production_unit_controller plant used in PIA

Id	Ontology class	Description
C 1.1.2.2.3.1.5	TUPRAS_unit	A company group generally equivalent in size and character to implement TUPRAS services.
C 1.1.2.2.3.1.5.1	Crude_Distillation_Unit	Crude Distillation Unit in TUPRAS
C 1.1.2.2.3.1.5.2	Delayed_Coker_Unit	Delayed Coker Unit in TUPRAS
C 1.1.2.2.3.1.5.3	Fluid_Catalytic_Cracking	Fluid Catalytic Cracking in TUPRAS
C 1.1.2.2.3.1.5.4	Hydrocracker	Hydrocracker in TUPRAS
C 1.1.2.2.3.1.5.5	LPG_raw_stream	LPG raw stream in TUPRAS
C 1.1.2.2.3.1.5.6	LPG_refined_stream	LPG refined stream in TUPRAS
C 1.1.2.2.3.1.5.7	Platformer_Maximum_Quality_Diesel	Platformer Maximum Quality Diesel in TUPRAS
C 1.1.2.2.3.2	organization	An object aggregate that corresponds to social institutions such as companies, societies etc. that does something
C 1.1.2.2.3.2.1	Department	An organizational unit in FACTLOG
C 1.1.3	specifically_dependent_continuant	A continuant & there is some independent continuant c which is not a spatial region and which is such that b s-depends_on c at every time t during the course of b's existence.
C 1.1.3.1	quality	A specifically dependent continuant that, in contrast to roles and dispositions, does not require any further process in order to be realized.
C 1.1.3.1.1	Factlog_BRC_quality_entity	BRC quality
C 1.1.3.1.1.1	BRC_organization_Competence	BRC organization Competence
C 1.1.3.1.1.2	BRC_personality_competencies	BRC personality competencies
C 1.1.3.1.1.3	BRC_production_Line_competence	BRC production Line competence
C 1.1.3.1.1.4	BRC_relational_quality	BRC relational quality
C 1.1.3.1.2.1	Factlog_CONT_quality_entity	CONT quality
C 1.1.3.1.2.1	CONT_organization_Competence	CONT organization Competence
C 1.1.3.1.2.2	CONT_personality_competencies	CONT personality competencies
C 1.1.3.1.2.3	CONT_production_Line_competence	CONT production Line competence
C 1.1.3.1.2.4	CONT_relational_quality	CONT relational quality
C 1.1.3.1.3	Factlog_JEMS_quality_entity	JEMS quality
C 1.1.3.1.3.1	Efficiency_of_waste	JEMS efficiency of waste
C 1.1.3.1.3.2	Fuel_efficiency	JEMS fuel efficiency
C 1.1.3.1.4	Factlog_PIA_quality_entity	PIA quality
C 1.1.3.1.4.1	fabric_features	Fabric features for PIA
C 1.1.3.1.4.2	fabric_quality	Fabric quality for PIA
C 1.1.3.1.4.3	Lightening_and_HVAC	Lightening and HVAC for PIA
C 1.1.3.1.4.4	Motor_efficiency	Motor efficiency for PIA
C 1.1.3.1.4.5	plant_consumptions	Plant consumptions for PIA
C 1.1.3.1.4.6	total_energy_consumption	Total energy consumption for PIA
C 1.1.3.1.5	Factlog_TUPRAS_quality_entity	TUPRAS quality
C 1.1.3.1.5.1	LPG_quality	LPG quality for TUPRAS
C 1.1.3.2	realizable_entity	A specifically dependent continuant that inheres in some independent continuant which is not a spatial region and is of a type instance of which are realized in processes of a correlated type.

Id	Ontology class	Description
C 1.1.3.2.1	disposition	A realizable entity & b's bearer is some material entity & b is such that if it ceases to exist, then its bearer is physically changed, & b's realization occurs when and because this bearer is in some special physical circumstances, & this realization occurs in virtue of the bearer's physical make-up.
C 1.1.3.2.1.1	function	A disposition that exists in virtue of the bearer's physical make-up and this physical make-up is something the bearer possesses because it came into being, either through evolution (in the case of natural biological entities) or through intentional design (in the case of artifacts), in order to realize processes of a certain sort.
C 1.1.3.2.1.1.1	Factlog_CONT_function_entity	CONT function
C 1.1.3.2.1.1.1.1	Carrier_Management	Carrier management function for CONT
C 1.1.3.2.1.1.1.2	Master_process_monitor	Master process monitor function for CONT
C 1.1.3.2.1.1.1.3	Material_management	Material management function for CONT
C 1.1.3.2.1.1.1.4	OEE	OEE function for CONT
C 1.1.3.2.1.1.1.5	Process_control	Process control function for CONT
C 1.1.3.2.1.1.1.6	Serial_Number_Generation	Serial number generation function for CONT
C 1.1.3.2.1.1.1.7	Work_center_administration	Work center administration function for CONT
C 1.1.3.2.2	role	A realizable entity & b exists because there is some single bearer that is in some special physical, social, or institutional set of circumstances in which this bearer does not have to be& b is not such that, if it ceases to exist, then the physical make-up of the bearer is thereby changed.
C 1.2	occurrent	An occurrent is an entity that unfolds itself in time or it is the instantaneous boundary of such an entity (for example a beginning or an ending) or it is a temporal or spatiotemporal region which such an entity occupies_temporal_region or occupies_spatiotemporal_region.
C 1.2.1	process	an occurrent that has temporal proper parts and for some time t, p s-depends_on some material entity at t.
C 1.2.1.1	Factlog_BRC_process_entity	Process entity proposed in the BRC pilot
C 1.2.1.1.1	Bending_process	BRC process for bending
C 1.2.1.1.2	Bent_threaded_process	BRC process for bent threaded
C 1.2.1.1.3	Customer_lengths_process	BRC process for customer lengths
C 1.2.1.1.4	Cutting_process	BRC process for cutting
C 1.2.1.1.5	Decoiled_straights_process	BRC process for decoiled straights
C 1.2.1.1.6	Final_products_process	BRC process for final production
C 1.2.1.1.7	Links_process	BRC process for linking
C 1.2.1.1.8	Mill_lengths_process	BRC process for mill lengths
C 1.2.1.1.9	Production_Process	BRC process for production
C 1.2.1.1.10	Raw_materials_process	BRC process for processing raw materials
C 1.2.1.1.11	Stage1_in_BRC	BRC process for stage 1

Id	Ontology class	Description
C 1.2.1.1.12	Threaded_coupled_process	BRC process for threaded_coupled
C 1.2.1.1.13	Trailer_process	BRC process for trailering
C 1.2.1.2	Factlog_CONT_process_entity	Process entity proposed in the CONT pilot
C 1.2.1.2.1	Final_assembly_process	CONT final assembly process
C 1.2.1.2.1.1	Bent_pin_process	CONT bent pin process
C 1.2.1.2.1.2	CISS_process	CONT CISS process
C 1.2.1.2.1.3	Curing_process	CONT curing process
C 1.2.1.2.1.4	Final_test_process	CONT final testing process
C 1.2.1.2.1.5	Laser_marking_process	CONT laser marking process
C 1.2.1.2.1.6	Leakage_test_process	CONT leakage testing process
C 1.2.1.2.1.7	Loading_In_Final_Assembly_process	CONT loading in final assembly process
C 1.2.1.2.1.8	Plasma_cleaning	CONT plasma cleaning process
C 1.2.1.2.1.9	Pressfit_pin_check_process	CONT pressfit pin check process
C 1.2.1.2.1.10	Pressfit_process	CONT pressfit process
C 1.2.1.2.1.11	Roll_over_process	CONT rolling over process
C 1.2.1.2.1.12	Screwing_process	CONT screwing process
C 1.2.1.2.1.13	Shaker_process	CONT shakering process
C 1.2.1.2.2	Pre_assymbling_process	CONT pre assembling process
C 1.2.1.2.2.1	Depaneling_process	CONT depaneling process
C 1.2.1.2.2.2	Hot_and_cold_test_process	CONT hot and cold testing process
C 1.2.1.2.2.3	ICT_process	CONT ICT process
C 1.2.1.2.2.4	Loading_in_pre_assembling_process	CONT loading process in pre assembling
C 1.2.1.2.3	Stock_from_SMT_process	CONT stock process from SMT
C 1.2.1.2.4	Stock_process	CONT stock process
C 1.2.1.3	Factlog_JEMS_process_entity	Process entity proposed in the JEMS pilot
C 1.2.1.3.1	Drying_and_mixing_of_feedstock_process	JEMS drying and mixing of feedstock process
C 1.2.1.3.1.1	Drying_and_mixing	JEMS drying and mixing process
C 1.2.1.3.1.2	Water_removal_in_Drying_and_mixing_of_feedstock	JEMS water removal in Drying and mixing of feedstock process
C 1.2.1.3.2	Feedstock_preparation_process	JEMS feedstock preparation process
C 1.2.1.3.2.1	Feeding	JEMS feeding process
C 1.2.1.3.2.2	Feedstock_inspecting	JEMS feedstock inspecting process
C 1.2.1.3.3	JEMS_distillation_process	JEMS distillation process
C 1.2.1.3.4	JEMS_processing_process	JEMS processing process
C 1.2.1.3.4.1	Chemical_processing	JEMS chemical processing
C 1.2.1.3.4.2	CO2_removal	JEMS CO2 removal process
C 1.2.1.3.4.3	Water_removal_in_processing	JEMS water removal in processing
C 1.2.1.3.5	Other_JEMS_process	Other process for JEMS
C 1.2.1.3.5.1	Catalyst_dosing	JEMS catalyst dosing process
C 1.2.1.3.5.2	Fuel_testing	JEMS fuel testing process
C 1.2.1.3.5.3	Lyme_dosing	JEMS lyme dosing process
C 1.2.1.3.5.4	Process_of_recuperation	JEMS recuperation process
C 1.2.1.3.5.5	Process_oil_dosing	JEMS oil dosing process
C 1.2.1.3.5.6	Regular_monitoring_and_inspection	JEMS regular monitoring and inspection process
C 1.2.1.3.5.7	Sludge_disposal	JEMS sludge disposal process
C 1.2.1.3.5.8	Sludge_removal	JEMS sludge removal process

Id	Ontology class	Description
C 1.2.1.3.5.9	Storing	JEMS storing process
C 1.2.1.4	Factlog_PIA_process_entity	Process entity proposed in the PIA pilot
C 1.2.1.4.1	Defect_repair_process	PIA repair process
C 1.2.1.4.2	PIA_delivery_process	PIA delivery process
C 1.2.1.4.3	PIA_inspection_process	PIA inspection process
C 1.2.1.4.4	PIA_weaving_process	PIA weaving process
C 1.2.1.5	Factlog_TUPRAS_process_entity	Process entity proposed in the TUPRAS pilot
C 1.2.1.5.1	LPG_production_process	TUPRAS process for LPG production
C 1.2.1.5.2	Refinery_process	TUPRAS process for refinery
C 1.2.1.5.3	Stage_of_pipeline	TUPRAS process for pipeline
C 1.2.1.5.4	Subprocess	TUPRAS subprocess
C 1.2.1.5.5	TUPRAS_distillation_process	TUPRAS process for distillation
C 1.2.2	process_boundary	a temporal part of a process & p has no proper temporal parts.
C 1.2.2.1	ending_instant	a process boundary when a process may start"
C 1.2.2.2	Starting_instant	a process boundary when a process may end"
C 1.2.3	service	Service is delivered when the service implements the system function
C 1.2.3.1	Factlog_BRC_service	Services proposed in BRC
C 1.2.3.1.1	Hydraulic_system_cooling_service	BRC service for cooling the hydraulic system
C 1.2.3.2	Factlog_CONT_service	Services proposed in CONT
C 1.2.3.2.1	Implement_screwing	CONT service for screwing
C 1.2.3.3	Factlog_JEMS_service	Services proposed in JEMS
C 1.2.3.3.1	waste_processing_service	JEMS service for processing the waste
C 1.2.3.4	Factlog_PIA_service	Services proposed in PIA
C 1.2.3.4.1	engine_recovery_service	PIA service for recovering the engine
C 1.2.3.4.2	engine_restart_service	PIA service for restarting the engine
C 1.2.3.4.3	engine_stop_service	PIA service for stopping the engine
C 1.2.4	spatiotemporal_region	A spatiotemporal region is an occurrent entity that is part of spacetime.
C 1.2.5	temporal_region	An occurrent entity that is part of time as defined relative to some reference frame.
C 1.2.5.1	one-dimensional_temporal_region	A temporal interval is a special kind of one-dimensional temporal region, namely one that is self-connected (is without gaps or breaks).
C 1.2.5.2	zero-dimensional_temporal_region	the temporal region during which a process occurs.

Table 6: Object property of FACTLOG ontology

Id	Object property	Description
OP 1.1	Anomaly_object_property	Interrelationship for defining anomaly detection
OP 1.1.1	DetectByFaultStatus	Detect By Fault Status
OP 1.1.2	ErrorLedByFault	Error Led By Fault

Id	Object property	Description
OP 1.1.3	FailureLedByError	Failure Led By Error
OP 1.1.4	Generate_FaultStatus	Generate Fault Status
OP 1.1.5	ReferenceDataRefer	Reference Data Refer xxx
OP 1.1.6	serviceHasFailure	Service Has Failure
OP 1.1.7	ServiceHasFaultStatus	Service Has FaultStatus
OP 1.2	Data_analysis_data_property	Interrelationship for defining data analysis
OP 1.2.1	TUPRAS_data_analysis_data_property	Interrelationship for TUPRAS data analysis
OP 1.2.1.1	control_id_mapping_to_parameter	control_id mapping to parameter
OP 1.2.1.2	sensor_id_mapping_to_sensor	sensor_id mapping to sensor
OP 1.2.1.3	sensor_id_mapping_to_sensor_data	sensor_id mapping to sensor_data
OP 1.2.1.4	sequence_from_unit	Sequence from unit
OP 1.2.1.5	sequence_to_unit	Sequence to unit
OP 1.2.1.6	target_id_mapping_to_parameter	target_id mapping to parameter
OP 1.2.1.7	unit_id_include_control_id	unit_id include control_id
OP 1.2.1.8	unit_id_include_sensor_id	unit_id include sensor_id
OP 1.2.1.9	unit_id_include_target_id	unit_id include target_id
OP 1.2.1.10	unit_id_mapping_to_unit	unit_id mapping to unit
OP 1.3	interrelationship	Interrelationship among concepts from different perspectives
OP 1.3.1	FaultstatusRelatedSenserData	Fault status Related Senser Data
OP 1.3.2	In_stage	XXX In stage
OP 1.3.3	model_represent_physical_event	Model represents physical event
OP 1.3.4	Optimization_input_PM_mapping_to_process_model	Optimization input PM mapping to process model
OP 1.3.5	Represent	XXX Represent BBB
OP 1.4	job_related_data_property	Interrelationship for defining jobs
OP 1.5	optmization_related_object_property	Interrelationship for defining optimization
OP 1.5.1	BRC_optimization_object_property	Interrelationship for defining optimization in BRC
OP 1.5.1.1	optimization_input_include_machine	optimization_input include machine
OP 1.5.1.2	optimization_input_include_order	optimization_input include order
OP 1.5.1.3	optimization_input_include_route	optimization_input include route
OP 1.5.1.4	optimization_input_inlcude_lambda	optimization_input include lambda
OP 1.5.1.5	optimization_ouput_have_job	optimization_ouput have job
OP 1.5.1.6	optimization_ouput_have_output_order	optimization_ouput have output order
OP 1.5.1.7	optimization_output_machine_job_id	optimization_output machine_job_id
OP 1.5.1.8	optimization_output_Objective	optimization_output Objective
OP 1.5.1.9	optimization_output_order_have_output_job	optimization_output_order have output_job
OP 1.5.1.10	optimization_output_output_job_have_parent_id	optimization_output_output_job have parent_id
OP 1.5.1.11	optimization_output_output_job_have_start_time	optimization_output_output_job have start_time
OP 1.5.1.12	optimization_output_output_machine	optimization_output output machine
OP 1.5.1.13	optimization_produced_at	optimization_produced_at
OP 1.5.1.14	optmization_output_job_have_machine	optmization_output_job have machine
OP 1.5.1.15	optmization_output_output_have_complet e_time	optmization_output_output have complete_time

Id	Object property	Description
OP 1.5.2	CONT_optimization_object_property	Interrelationship for defining optimization in CONT
OP 1.5.2.1	optimization_cont_input_has_static_data	optimization_cont_input has static_data
OP 1.5.2.2	optimization_cont_input_include_rounte	optimization_cont_input include rounte
OP 1.5.2.3	optimization_cont_input_line_related_object_property	Interrelationships among optimization_cont_input_line
OP 1.5.2.3.1	line_has_code	Line has code
OP 1.5.2.3.2	line_has_description	Line has description
OP 1.5.2.3.3	line_has_id	Line has id
OP 1.5.2.3.4	line_has_linetypeId	Line has linetypeId
OP 1.5.2.4	optimization_cont_input_line_type_related_object_prop erty	Interrelationships among optimization_cont_input_line_type
OP 1.5.2.4.1	linetype_has_code	Linetype has code
OP 1.5.2.4.2	linetype_has_description	Linetype has description
OP 1.5.2.4.3	linetype_has_id	Linetype has id
OP 1.5.2.5	optimization_cont_input_product_object_property	Interrelationships among optimization_cont_input_product
OP 1.5.2.5.1	optimization_input_product_has_efficiency_rate	optimization_input_product has efficiency_rate
OP 1.5.2.5.2	optimization_input_product_has_description	optimization_input_product has description
OP 1.5.2.5.3	optimization_input_product_has_id	optimization_input_product has id
OP 1.5.2.5.4	optimization_input_product_has_name	optimization_input_product has name
OP 1.5.2.5.5	optimization_input_product_have_product_end_line_id	optimization_input_product have product_end_line_id
OP 1.5.2.5.6	optimization_input_product_have_product_familiy_id	optimization_input_product have product_familiy_id
OP 1.5.2.5.7	optimization_input_product_have_product_source_line _id	optimization_input_product have product_source_line_id
OP 1.5.2.6	optimization_cont_input_productBOM_related_entity	Interrelationships among optimization_cont_input_productBOM
OP 1.5.2.7	optimization_cont_input_resource_Bom_related_entity	Interrelationships among optimization_cont_input_resource_Bom
OP 1.5.2.7.1	optimization_input_resource_bom_has_M ultiplicity	optimization_input_resource_bom has Multiplicity
OP 1.5.2.7.2	optimization_input_resource_bom_has_pr oducts_id	optimization_input_resource_bom has products_id
OP 1.5.2.7.3	optimization_input_resource_bom_has_resource_id	optimization_input_resource_bom has resource_id
OP 1.5.2.7.4	optimization_input_resource_bom_has_workplaceid	optimization_input_resource_bom has workplaceid
OP 1.5.2.8	optimization_cont_input_scheduleMainten anceActivity_has	optimization_cont_input_scheduleMai ntenanceActivity has
OP 1.5.2.9	optimization_cont_input_workplace_type_related_obje ct_property	Interrelationships among optimization_cont_input_workplace_type
OP 1.5.2.9.1	workspace_type_has_code	workspace_type has code
OP 1.5.2.9.2	workspace_type_has_description	workspace_type has description
OP 1.5.2.9.3	workspace_type_has_id	workspace_type has id
OP 1.5.2.10	optimization_cont_input_workspace_related_entity	optimization_cont_input_workspace
OP 1.5.2.10.1	optimization_cont_input_workspace_has_code	optimization_cont_input_workspace has code
OP 1.5.2.10.2	optimization_cont_input_workspace_has_description	optimization_cont_input_workspace has

Id	Object property	Description
		description
OP 1.5.2.10.3	optimization_cont_input_workspace_has_id	optimization_cont_input_workspace has id
OP 1.5.2.10.4	optimization_cont_input_workspace_has_lineId	optimization_cont_input_workspace has lineId
OP 1.5.2.10.5	optimization_cont_input_workspace_has_SequnceLine	optimization_cont_input_workspace has SequnceLine
OP 1.5.2.10.6	optimization_cont_input_workspace_has_worplacetypeId	optimization_cont_input_workspace has worplacetypeId
OP 1.5.2.11	optimization_cont_output_has_metrics	optimization_cont_output has metrics
OP 1.5.2.12	optimization_cont_output_has_schedule	optimization_cont_output has schedule
OP 1.5.2.13	optmization_cont_input_has_dynamic_data	optmization_cont_input has dynamic_data
OP 1.5.2.14	optmization_cont_input_processing_time_related_object_property	Interrelationships among optmization_cont_input_processin_time
OP 1.5.2.14.1	optimization_input_processingtime_has_ideal_processing_time	optimization_input_processingtime has ideal_processing_time
OP 1.5.2.14.2	optimization_input_processingtime_has_product_id	optimization_input_processingtime has product_id
OP 1.5.2.14.3	optimization_input_processingtime_has_real_processing_time	optimization_input_processingtime has real_processing_time
OP 1.5.2.14.4	optimization_input_processingtime_has_workpalcetype	optimization_input_processingtime has workpalcetype
OP 1.5.2.15	optmization_cont_input_product_families_object_properety	Interelationships among optmization_cont_input_product_families
OP 1.5.2.15.1	product_families_has_description	product_families has description
OP 1.5.2.15.2	product_families_has_id	product_families has id
OP 1.5.2.15.3	product_families_has_name	product_families has name
OP 1.5.2.16	optmization_cont_input_production_order_has	optmization_cont_input_production_order has
OP 1.5.2.17	optmization_cont_input_resource_related_object_property	Interelationships among optmization_cont_input_resource
OP 1.5.2.17.1	optimization_input_resource_has_description	optimization_input_resource has description
OP 1.5.2.17.2	optimization_input_resource_has_id	optimization_input_resource has id
OP 1.5.2.17.3	optimization_input_resource_has_name	optimization_input_resource has name
OP 1.5.2.18	optmization_cont_input_setup_time_related_object_property	Interelationships among optmization_cont_input_setup_time
OP 1.5.2.18.1	optmization_input_setup_time_has_linetypeid	optmization_input_setup_time has linetypeid
OP 1.5.2.18.2	optmization_input_setup_time_has_productfamiliesId	optmization_input_setup_time has productfamiliesId
OP 1.5.2.18.3	optmization_input_setup_time_has_setuptime_id	optmization_input_setup_time has setuptime_id
OP 1.5.2.19	optmization_cont_output_has_uuid	optmization_cont_output has uuid
OP 1.5.2.20	optmization_cont_output_product_at	optmization_cont_output product_at
OP 1.5.2.21	optmization_cont_output_schedule_has	optmization_cont_output_schedule has
OP 1.5.3	PIA_optimization_object_property	Interrelationship for defining optimization in PIA
OP 1.5.3.1	opimization_pia_output_include_looms_order_sequence	opimization_pia_output include looms_order_sequence
OP 1.5.3.2	optimization_pia_input_include_currentTotalSetupTime	optimization_pia_input include currentTotalSetupTime
OP 1.5.3.3	optimization_pia_input_include_historicalDataWeekly	optimization_pia_input include

Id	Object property	Description
		historicalDataWeekly
OP 1.5.34	optimization_pia_input_include_looms	optimization_pia_input include looms
OP 1.5.3.5	optimization_pia_input_include_order	optimization_pia_input include order
OP 1.5.3.6	optimization_pia_input_include_route	optimization_pia_input include route
OP 1.5.3.7	optimization_pia_input_loom_include_entit y	optimization_pia_input_loom include entity
OP 1.5.3.8	optimization_pia_output_include_order	optimization_pia_output include order
OP 1.5.3.9	optimization_pia_output_include_uuid	optimization_pia_output include uuid
OP 1.5.3.10	optimization_pia_output_loom_order_sequ ence_include_id	optimization_pia_output_loom_order_sequ ence include id
OP 1.5.3.11	optimization_pia_output_loom_order_sequ ence_include_sequence_entity	optimization_pia_output_loom_order_sequ ence include sequence_entity
OP 1.5.3.12	optimization_pia_output_produce_at	optimization_pia_output produce at
OP 1.5.3.13	optmization_pia_input_include_startDate	optmization_pia_input include startDate
OP 1.5.3.14	optmization_pia_input_include_workGroupsPerShift	optmization_pia_input include workGroupsPerShift
OP 1.5.3.15	optmization_pia_input_order_include_entity	optmization_pia_input_order include entity
OP 1.5.3.16	optmization_pia_output_include_include_total_EnergyC ons	optmization_pia_output include include_total_EnergyCons
OP 1.5.3.17	optmization_pia_output_include_makespan	optmization_pia_output include makespan
OP 1.5.3.18	optmization_pia_output_include_total_compeletionTime	optmization_pia_output include total_compeletionTime
OP 1.5.3.19	optmization_pia_output_include_totalTardiness	optmization_pia_output include totalTardiness
OP 1.5.3.20	optmization_pia_output_order_include_entity	optmization_pia_output_order include entity
OP 1.5.4	Tupras_optimization_object_property	Interrelationship for defining optimization in Tupras
OP 1.5.4.1	optimization_input_unit_scenario_include_entity	optimization_input_unit_scenario include entity
OP 1.5.4.2	optimization_input_include_OS	optimization_input include OS
OP 1.5.4.3	optimization_input_include_PM	optimization_input include PM
OP 1.5.4.4	optimization_input_InputFeeds_including_entity	optimization_input_InputFeeds including entity
OP 1.5.4.5	optimization_input_likScenario_including_entity	optimization_input_likScenario including entity
OP 1.5.4.6	optimization_input_OS_inlucde_unit_scenario	optimization_input_OS inlucde unit_scenario
OP 1.5.4.7	optimization_input_output_task_include_entity	optimization_input_output_task include entity
OP 1.5.4.8	optimization_input_PI_include_output_task	optimization_input_PI include output_task
OP 1.5.4.9	optimization_input_PI_include_settings	optimization_input_PI include settings
OP 1.5.4.10	optimization_input_PI_include_specs	optimization_input_PI include specs
OP 1.5.4.11	optimization_input_setting_include_entity	optimization_input_setting include entity
OP 1.5.4.12	optimization_input_specification_include_entity	optimization_input_specification include entity
OP 1.5.4.13	optimization_input_unit_scenario_including_linkScenario	optimization_input_unit_scenario including linkScenario
OP 1.5.4.14	optimization_output_including_output_kpi_entity	optimization_output including output_kpi_entity
OP 1.5.4.15	optimization_output_including_SolKPIs	optimization_output including SolKPIs
OP 1.5.4.16	optimization_output_including_solution_scenarios	optimization_output including solution_scenarios

Id	Object property	Description
OP 1.5.4.17	optimization_output_including_total_energy	optimization_output including total_energy
OP 1.5.4.18	optmization_inpput_including_route	optmization_inpput including route
OP 1.5.4.19	optmization_input_including_PI	optmization_input including PI
OP 1.5.4.20	optmization_input_PI_includ_inputFeed	optmization_input_PI include inputFeed
OP 1.5.4.21	optmization_output_including_produced_at_entity	optmization_output including produced_at_entity
OP 1.5.4.22	optmization_output_including_uuid	optmization_output including uuid
OP 1.5.4.23	optmization_output_solution_scenario_including_NodeID	optmization_output_solution_scenario_including_NodeID
OP 1.5.4.24	optmization_output_solution_scenario_including_OptID	optmization_output_solution_scenario_including_OptID
OP 1.5.4.25	optmization_output_solution_scenario_including_ScenarioID	optmization_output_solution_scenario_including_ScenarioID
OP 1.6	Pilot_specific_object_property	Interrelationship for defining specific pilot
OP 1.6.1	ActionControl	AAAA ActionControl BBB
OP 1.6.2	AlgorithmImplementInSoftware	Algorithm Implement In Software
OP 1.6.3	AlgorithmOutputAction	Algorithm Output Action
OP 1.6.4	Bar_produce	Bar produce XXX
OP 1.6.5	belong_to	AAA Belong _to BBB
OP 1.6.6	contain	XXX contain BBB
OP 1.6.7	deliver	XXX deliver BBB
OP 1.6.8	developed_by	XXX Developed by BBB
OP 1.6.9	employed-by	XXX Employed by BBB

Table 7: Data property of FACTLOG ontology

Id	Data property	Description
DP 1.1	human_resource_related_data_property	Data_property of human resource
DP 1.1.2	human_resource_competency_ID	Value of human_resource_competency_ID
DP 1.1.2	human_resource_competency_title	Value of human_resource_competency_title
DP 1.2	industrial_process_related_data_property	Data_property of industrial_process
DP 1.2.1	indsutrial_process_title	Value of indsutrial_process_title
DP 1.2.2	industrial_process_workplace	Value of industrial_process_workplace
DP 1.2.3	industrial_process_ID	Value of industrial_process_ID
DP 1.2.4	industrial_process_num	Value of industrial_process_num
DP 1.2.5	industrial_process_peron_per_shift	Value of industrial_process_peron_per_shift
DP 1.3	job_related_data_property	Data_property of job
DP 1.3.1	job_process_stage	FactlogDefintion "job_process_stage for BRC optimization state"
DP 1.3.2	job_Id	Value of job_Id
DP 1.4	line_related_data_property	Data_property of line_related entity
DP 1.4.1	line_code_value	Value of line_code
DP 1.4.2	line_description_value	Value of line_description
DP 1.4.3	line_id_value	Value of line_id
DP 1.5	line_type_data_property	Data property of line_type
DP 1.5.1	linetyp_description_value	Value of linetyp_description

Id	Data property	Description
DP 1.5.2	linetype_code_value	Value of linetype_code
DP 1.5.3	linetype_id_value	Value of linetype_id
DP 1.6	loom_related_data_property	Data property of loom related entity
DP 1.6.1	loom_id_value	Value of loom_id
DP 1.6.2	loom_speed_value	Value of loom_speed
DP 1.6.3	maintenance_end_value	Value of maintenance_end
DP 1.6.4	maintenance_start_value	Value of maintenance_start
DP 1.7	machine_related_data_property	Data property of machine related entity
DP 1.7.1	Mach_id	Value of Mach_id
DP 1.7.2	Mach_set_up	Value of Mach_set_up
DP 1.7.3	Mach_status	Value of Mach_status
DP 1.7.4	Machine_downtime	Value of Machine_downtime
DP 1.7.5	Machine_highlight_issue	Value of Machine_highlight_issue
DP 1.7.6	machine_type_value	Value of machine_type_value
DP 1.8	machine_tool_related_data_property	Data property of machine tool related entity
DP 1.8.1	machining_tool_ID	Value of machining_tool_ID
DP 1.8.2	machining_tool_title	Value of machining_tool_title
DP 1.9	Node_related_data_property	Data property of node related entity
DP 1.9.1	NodeID	Value of NodeID
DP 1.9.2	NodeID_value	Value of NodeID
DP 1.9.3	OutputNodeID_value	Value of OutputNodeID
DP 1.10	optmization_related_data_property	Data property of optimization related entity
DP 1.10.1	Optimization_Tupras_data_property	Data property of Tupras optmization entity
DP 1.10.1.1	C2_value	Value of C2
DP 1.10.1.2	C2C5_value	Value of C2C5
DP 1.10.1.3	C5_value	Value of C5
DP 1.10.1.4	CAP_ij_value	Value of CAP_ij
DP 1.10.1.5	E_i_s_value	Value of E_i_s
DP 1.10.1.6	FoundSolution_value	Value of FoundSolution
DP 1.10.1.7	Horizon_value	Value of Horizon
DP 1.10.1.8	IC2_i_value	Value of IC2_i
DP 1.10.1.9	IC5_i_value	Value of IC5_i
DP 1.10.1.10	IF_i_value	Value of IF_i
DP 1.10.1.11	InputNodeID_value	Value of InputNodeID
DP 1.10.1.12	ISU_i_value	Value of ISU_i
DP 1.10.1.13	LinkID_value	Value of LinkID
DP 1.10.1.14	OptID_value	Value of OptID
DP 1.10.1.15	PC2_i_j_s_value	Value of PC2_i_j_s
DP 1.10.1.16	PC5_i_j_s_value	Value of PC5_i_j_s
DP 1.10.1.17	PF_i_j_s_value	Value of PF_i_j_s
DP 1.10.1.18	PSU_i_j_s_value	Value of PSU_i_j_s
DP 1.10.1.19	Q_start_i_value	Value of Q_start_i
DP 1.10.1.20	Q_total_i_value	Value of Q_total_i
DP 1.10.1.21	QC2_start_i_value	Value of QC2_start_i

Id	Data property	Description
DP 1.10.1.22	QC5_start_i_value	Value of QC5_start_i
DP 1.10.1.23	QSU_start_i_value	Value of QSU_start_i
DP 1.10.1.24	ScenarioID_i_s_value	Value of ScenarioID_i_s
DP 1.10.1.25	SU_value	Value of SU
DP 1.10.1.26	TotalEnergy_value	Value of TotalEnergy
DP 1.10.2	optmization_BRC_related_data_property	Data property of BRC optmization entity
DP 1.10.2.1	optimization_output_job_completion_time_sequence_i d_value	Value of optimization_output_job_completion_time_ sequence_id_value
DP 1.10.2.2	optimization_output_job_completion_time_value	Value of optimization_output_job_completion_time_ value
DP 1.10.2.3	optimization_output_job_start_time_seque nce_id_value	Value of optimization_output_job_start_time_seque nce_id_value
DP 1.10.2.4	optimization_output_job_start_time_value	Value of optimization_output_job_start_time_value
DP 1.10.2.5	optimization_output_TotalLateness	Value of optimization_output_TotalLateness
DP 1.10.2.6	optimization_output_TotalTardiness	Value of optimization_output_TotalTardiness
DP 1.10.2.7	optimzation_input_lambda_value	Value of optimzation_input_lambda_value
DP 1.10.2.8	optmization_ouput_job_parennt_id_value	Value of optmization_ouput_job_parennt_id_va lue
DP 1.10.2.9	output_order_list_element_id	Value of output_order_list_element_id
DP 1.10.2.10	OutputOrder_type	Value of OutputOrder_type
DP 1.10.3	optmization_CONT_related_data_property	Data property of CONT optmization entity
DP 1.10.3.1	AvgWorkStationIdleTime_value	Value of AvgWorkStationIdleTime
DP 1.10.3.2	producted_entity_value	Value of producted_entity
DP 1.10.3.3	schedule_end_value	Value of schedule_end
DP 1.10.3.4	schedule_start_value	Value of schedule_start
DP 1.10.3.5	ScheduledMaintenanceActivitiesRelatedDataProperty	Value of ScheduledMaintenanceActivitiesRelatedD ataProperty
DP 1.10.3.5.1	ScheduledMaintenanceActivities_duration_value	Value of ScheduledMaintenanceActivities_duration
DP 1.10.3.5.2	ScheduledMaintenanceActivities_Id_value	Value of cheduledMaintenanceActivities_Id
DP 1.10.3.5.3	ScheduledMaintenanceActivities_start_value	Value of ScheduledMaintenanceActivities_start
DP 1.10.3.6	TotalTardiness_value	Value of TotalTardiness
DP 1.10.4	optmization_general_related_data_property	Data property of _general optmization entity
DP 1.10.4.1	optimization_output_produced_at_value	Value of optimization_output_produced
DP 1.10.4.2	optimization_output_uuid_value	Value of optimization_output_uuid
DP 1.10.4.3	optmization_input_route_value	Value of optmization_input_route
DP 1.10.4.4	optmization_output_Makespan	Value of optmization_output_Makespan
DP 1.10.5	optmization_pia_related_data_property	Data property of PIA optmization entity
DP 1.10.5.1	optimization_input_start_data_value	Value of optimization_input_start_data
DP 1.10.5.2	optimization_output_llom_order_sequence_content_valu	Value of optimization_output_llom_order_sequence

Id	Data property	Description
	e	_content
DP 1.10.5.3	optimization_output_llom_order_sequence_entity_id_value	Value of optimization_output_llom_order_sequence_entity_id
DP 1.10.5.4	optimization_output_llom_order_sequence_id_value	Value of optimization_output_llom_order_sequence_id
DP 1.10.5.5	optimization_output_makespan	Value of optimization_output_makespan
DP 1.10.5.6	optimization_output_total_completion_times	Value of optimization_output_total_completion
DP 1.10.5.7	optimization_output_total_energycons_value	Value of optimization_output_total_energycons
DP 1.10.5.8	optimization_output_total_tardiness_value	Value of optimization_output_total_tardiness
DP 1.10.5.9	optmization_input_currentTotalSetupTime_value	Value of optmization_input_currentTotalSetupTime
DP 1.10.5.10	optmization_input_historicalDataWeekly_value	Value of optmization_input_historicalDataWeekly
DP 1.10.5.11	optmization_input_workGroupsPerShift_value	Value of optmization_input_workGroupsPerShft
DP 1.11	order_related_data_property	Data property of order related entity
DP 1.11.1	order_ca_value	Value of order_ca
DP 1.11.2	order_cc_value	Value of order_cc
DP 1.11.3	order_chainID_value	Value of order_chainID
DP 1.11.4	order_comb_value	Value of order_comb
DP 1.11.5	order_combheight_value	Value of order_combheight
DP 1.11.6	order_creation_time	Value of order_creation_time
DP 1.11.7	order_deliveryDate_value	Value of order_deliveryDate
DP 1.11.8	order_drawing_value	Value of order_drawing
DP 1.11.9	order_due_data	Value of order_due_data
DP 1.11.10	order_energyCons_value	Value of order_energyCons
DP 1.11.11	order_external_ID	Value of order_external_ID
DP 1.11.12	order_fabricType_value	Value of order_fabricType
DP 1.11.13	order_id	Value of order_id
DP 1.11.14	order_incom_value	Value of order_incom
DP 1.11.15	order_kStrokes_value	Value of order_kStrokes
DP 1.11.16	order_loom_id	Value of order_loom_id
DP 1.11.17	order_partID_value	Value of order_partID
DP 1.11.18	order_priorityWeight_value	Value of order_priorityWeight
DP 1.11.19	order_process_start_time	Value of order_process_start_time
DP 1.11.20	order_processing_time_value	Value of order_processing_time
DP 1.11.21	order_sequence_weight	Value of order_sequence_weight
DP 1.11.22	order_setup_end_time	Value of order_setup_end_time
DP 1.11.23	order_setup_start_time	Value of order_setup_start_time
DP 1.11.24	order_setup_time_value	Value of order_setup_time
DP 1.11.25	order_status_value	Value of order_status
DP 1.11.26	order_strokesPerMt_value	Value of order_strokesPerMt
DP 1.11.27	order_targetMeters_value	Value of order_targetMeters
DP 1.11.28	order_total_quantity	Value of order_total_quantity

Id	Data property	Description
DP 1.11.29	order_type_value	Value of order_type
DP 1.11.30	order_variant_value	Value of order_variant
DP 1.11.31	order_yarns_value	Value of order_yarns
DP 1.11.32	order_ybf_value	Value of order_ybf
DP 1.12	organization_related_data_property	Data property of related organization entity
DP 1.12.1	organization_ID	Value of organization_ID
DP 1.13	price_related_data_property	Data property of price related entity
DP 1.13.1	PriceOfElectricity_value	Value of PriceOfElectricity
DP 1.13.2	PriceOfLPG_value	Value of PriceOfLPG
DP 1.13.3	PriceOfNG_value	Value of PriceOfNG
DP 1.14	process_model_data_property	Data property of process related entity
DP 1.14.1	PM_data_property	Data property of PM_data related entity
DP 1.14.1.1	Description_value	Value of Description
DP 1.14.1.2	link_description_value	Value of link_description
DP 1.14.1.3	link_flow_calculated_value	Value of link_flow_calculated
DP 1.14.1.4	link_flow_factor_value	Value of link_flow_factor
DP 1.14.1.5	link_flow_formula_value	Value of link_flow_formula
DP 1.14.1.6	link_flow_manual_value	Value of link_flow_manual
DP 1.14.1.7	link_flow_name_value	Value of link_flow_name
DP 1.14.1.8	link_flow_quantity_value	Value of link_flow_quantity
DP 1.14.1.9	link_id_value	Value of link_id
DP 1.14.1.10	link_name_value	Value of link_name
DP 1.14.1.11	link_source_value	Value of link_source
DP 1.14.1.12	link_target_value	Value of link_target
DP 1.14.1.13	node_onetoone_value	Value of node_onetoone
DP 1.14.1.14	node_ScriptSource_value	Value of node_ScriptSource
DP 1.14.1.15	node_stock_value	Value of node_stock
DP 1.14.1.16	node_type_value	Value of node_type
DP 1.14.1.17	PM_Id_value	Value of PM_Id
DP 1.14.1.18	PM_name_value	Value of PM_name
DP 1.14.1.19	PM_node_description_value	Value of PM_node_description
DP 1.14.1.20	PM_node_id_value	Value of PM_node_id
DP 1.14.1.21	PM_node_name_value	Value of PM_node_name
DP 1.14.1.22	PM_node_stage_value	Value of PM_node_stage
DP 1.14.1.23	PM_resource_description_value	Value of PM_resource_description
DP 1.14.1.24	PM_resource_id_value	Value of PM_resource_id
DP 1.14.1.25	PM_resource_name_value	Value of PM_resource_name
DP 1.14.1.26	PM_resource_unit_value	Value of PM_resource_unit
DP 1.14.1.27	PM_solved_value	Value of PM_solved
DP 1.14.1.28	PM_specMethod	Value of PM_specMethod
DP 1.14.1.29	Symbol_value	Value of Symbol
DP 1.14.1.30	tolerance_value	Value of tolerance
DP 1.14.1.31	Unit_value	Value of Unit
DP 1.14.1.31	Value_value	Value refers to
DP 1.15	process_stock_material_related_data_property	Data property of process_stock_material related entity

Id	Data property	Description
DP 1.15.1	processing_stock_material_external_ID	Value of processing_stock_material_external_I D
DP 1.15.2	processing_stock_material_ID	Value of processing_stock_material_ID
DP 1.15.3	processing_stock_material_title	Value of processing_stock_material_title
DP 1.16	product_bom_related_data_property	Data property of product_bom related entity
DP 1.16.1	productBOM_multiplicity_value	Value of productBOM_multiplicity
DP 1.17	product_family_related_data_property	Data property of product_family related entity
DP 1.17.1	product_families_description	Value of product_families_description
DP 1.17.2	product_families_id_value	Value of product_families_id
DP 1.17.3	product_families_name	Value of product_families_name
DP 1.18	product_order_related_data_property	Data property of product_order related entity
DP 1.18.1	product_order_DueDate_value	Value of product_order_DueDate
DP 1.18.2	product_order_id_value	Value of product_order_id
DP 1.18.3	product_order_MaxQuantity_value	Value of product_order_MaxQuantity
DP 1.18.4	product_order_name_value	Value of product_order_name
DP 1.18.5	product_order_Priority_value	Value of product_order_Priority
DP 1.18.6	product_order_Quantity_value	Value of product_order_Quantity
DP 1.19	product_related_data_property	Data property of product related entity
DP 1.19.1	product_efficiencyRate	Value of product_efficiencyRate
DP 1.19.2	ProductDescription	Value of ProductDescription
DP 1.19.3	productId	Value of productId
DP 1.19.4	ProductionShape	Value of ProductionShape
DP 1.19.5	productName	Value of productName
DP 1.19.6	ProductSize	Value of ProductSize
DP 1.20	production_plant_related_data_property	Data property of production_plant related entity
DP 1.20.1	production_plant_external_ID	Value of production_plant_external_ID
DP 1.20.2	production_plant_ID	Value of production_plant_ID
DP 1.20.3	production_plant_title	Value of production_plant_title
DP 1.21	productline_related_data_property	Data property of productline related entity
DP 1.21.1	production_line_ID	Value of production_line_ID
DP 1.21.2	production_line_title	Value of production_line_title
DP 1.22	resource_related_data_property	Data property of resource related entity
DP 1.22.1	resource_name_value	Value of resource_name
DP 1.22.2	resource_bom_multiplicity_value	Value of resource_bom_multiplicity
DP 1.22.3	resource_description_value	Value of resource_description
DP 1.22.4	resource_id_value	Value of resource_id
DP 1.23	ResultTestValue	Value of Result Test
DP 1.24	route_value	Value of route
DP 1.25	ScenarioID	Value of ScenarioID
DP 1.26	sersorDataProperty	Data property of sersorData
DP 1.26.1	Batch_code	FactlogDefintion "Batch_code parameter for BRC"
DP 1.26.2	Cycle_Start_time	FactlogDefintion "Cycle_Start_time parameter for BRC"

Id	Data property	Description
DP 1.26.3	Hydraulic_pressure	FactlogDefintion "Hydraulic_pressure from sensors in the BRC pilot"
DP 1.26.4	Hydraulic_Temp	FactlogDefintion "Hydraulic_Temp from sensors in the BRC pilot"
DP 1.26.5	Mains_current	FactlogDefintion "Mains_current from sensors in the BRC pilot"
DP 1.26.6	Mains_frequency	Value of Mains_frequency
DP 1.26.7	Mains_voltages	FactlogDefintion "Mains_voltages from sensors in the BRC pilot"
DP 1.26.8	Roller_wear	FactlogDefintion "Roller_wear from sensors in the BRC pilot"
DP 1.26.9	Step_angle_bent	FactlogDefintion "Step_angle_bent from sensors in the BRC pilot"
DP 1.26.10	step_Feed_length	Value of step_Feed_length
DP 1.26.11	Step_power	Value of Step_power
DP 1.26.12	Step_time	Value of Step_time
DP 1.26.13	temperature_value_from_sensor	Value of temperature_from_sensor
DP 1.27	shift_schedule_related_data_property	Data property of shift_schedule related entity
DP 1.27.1	shift_schedule_delay_time	Value of shift_schedule_delay_time
DP 1.27.2	shift_schedule_delay_type	Value of shift_schedule_delay_type
DP 1.27.3	shift_schedule_ID	Value of shift_schedule_ID
DP 1.27.4	shift_schedule_num	Value of shift_schedule_num
DP 1.27.5	shift_schedule_week	Value of shift_schedule_week
DP 1.27.6	shift_schedule_year	Value of shift_schedule_year
DP 1.28	TemperatureValue	Value of Temperature
DP 1.29	time_related_data_property	Data property of time related entity
DP 1.29.1	duration_value	Value of duration
DP 1.29.2	IdealProcessingTime_value	Value of IdealProcessingTime
DP 1.29.3	realProcessingTime_value	Value of realProcessingTime
DP 1.29.4	TimeToInitializeMillisec_value	Value of TimeToInitializeMillisec
DP 1.29.5	TimeToOptimize_value	Value of TimeToOptimize
DP 1.29.6	TimeToSolveMillisec_value	Value of TimeToSolveMillisec
DP 1.30	uuid_value	Value of uuid_value
DP 1.31	workforce_related_data_property	Data property of workforce related entity
DP 1.31.1	workforce_ID	Value of workforce_ID
DP 1.32	workplace_related_data_property	Data property of workplace related entity
DP 1.32.1	workplace_type_code_value	Value of workplace_type_code
DP 1.32.2	workplace_type_description_value	Value of workplace_type_description
DP 1.32.3	workplace_type_id_value	Value of workplace_type_id
DP 1.32.4	workspace_code_value	Value of workspace_code
DP 1.32.5	workspace_description_value	Value of workspace_description
DP 1.32.6	workspace_id_value	Value of workspace_id