MONSOON

MOdel based coNtrol framework for Site-wide OptimizatiON of data-intensive processes

D2.8– Final Cross-sectorial Domain Model

Deliverable ID	D2.8
Deliverable Title	Final Cross-Sectorial Domain Model
Work Package	WP2 – Requirements Engineering and Reference Architecture
Dissemination Level	PUBLIC
Version	1.1
Date	2018-12-13
Status	final
Lead Editor	
Main Contributors	Martin Sarnovsky, Peter Bednar (TUK)

Published by the MONSOON Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723650.

Document History

Version	Date	Author(s)	Description
0.1	2018-11-15	Martin Sarnovsky(TUK)	First Draft with TOC, Introduction draft, Scope and description of DMOP's Predictive algorithm and model concepts.
0.2	2018-11-19	Peter Bednar (TUK)	Added metamodel for domain concept.
0.3	2018-11-20	Peter Bednar (TUK)	Added graphical notation for process modelling.
0.4	2018-11-22	Peter Bednar (TUK)	Added description of fields for the process industry domain concepts (Production Process, Production Resource, etc.), added introduction and description of the overall structure
0.5	2018-11-24	Martin Sarnovsky (TUK)	Updated description of data mining concepts, added graphical notation to the sub-chapter, example added.
0.6	2018-11-30	Peter Bednar, Martin Sarnovsky (TUK)	Case study from aluminium domain section added.
0.7	2018-12-03	Peter Bednar	Added KPI elements, added motivation and main objectives
1.0	2018-12-08	Peter Bednar	Final editing
1.1	2018-12-13	Martin Sarnovsky (TUK)	Minor revisions after internal reviews.

Internal Review History

Version	Review Date	Reviewed by	Summary of comments
1.0	2018-12-10	Marco Dias (GLN)	Accepted with minor layout corrections and acronym addition
1.0	2018-12-11	Vincent Bonnivard (PROB)	Accepted with minor corrections

Table of Contents

1 Introduction

This deliverable describes the final specification of the Cross-sectorial domain model. The goal of the deliverable is to introduce the extensions of the initial version of Cross-sectorial domain model presented in deliverable D2.6 Initial Cross-sectorial domain model. The initial model specification presentedthe main concepts required for modelling of production processes, data elements and predictive functions. The idea was to provide the basic concepts to describe the production processes and related elements. Initial model served as a basis for modelling of application domains in MONSOON project. This deliverable presents the extension of the initial model with the new properties and relations, which were identified and incorporatedfrom the experiences gained during the modelling of pilot aluminium and plastic domains.

The deliverable is organized as follows:
Second chapter provides overall motivation and objectives of cross-sectorial model. Following chapter describes the final model structure. We focus mostly on areas which were improved or extended from the initial version of the model. Particular sub-sections are aimed at process modelling concepts, domain and data mining concepts. Following chapter presents a use case of model application in aluminium domain of the MONSOON project.

1.1 Scope

This document presents the final cross-sectorial domain model used in MONSOON platform. The document is a part of the *"Task 2.5 – Cross-sectorial domain model"* which is meant to develop the model definition. The main objective is to develop a model, that will unify existing standards for data analysis with standards related to manufacturing processes and their linking with ERPs. Model will be supported and maintained by Semantic framework for dynamic multi-scale industry modelling developed within Task 4.1, which will provide the tools to model the concrete production processes, etc. Initial model was described in deliverable *"D2.7 - Initial Cross-sectorial Domain Model"*and was extended and refined into the final model presented in deliverable "*D2.8 – Final Cross-sectorial Domain Model"*.

1.2 Terminology remark

The term **model** and **modelling** in this document is used in the following different meanings:

a) as the **semantic model**, i.e. formal specification of the concepts from the given domain
b) as the **data analysis model**, i.e. result of applying of the data modelling techniques (e.g. predictive function).
c) as the **process model**, i.e. model which specifies decomposition of the process to steps and describes their data and execution dependencies (usually based on graphical notation in the form of flow charts).

2 Motivation and main objectives

The main objectives of the Cross-sectorial domain model were already defined in the first version of the model described in deliverable D2.7 Initial Cross-sectorial domain model. The objectives can be summarized as to:

- Improve communication between the data scientists and domain experts in phases of Problem understanding, Data understanding and Evaluation by creation of the shared knowledge base.
- Allow within and cross-sectorial sharing of knowledge about the optimization of the production processes by data analytics methods and predictive functions.
- Automatize Evaluation of predictive functions by automatic computation of impact on the key-performance indicators from evaluation statistics.
- Allow "what-if" simulation scenarios and automatic optimization of production process during the Modelling and Evaluation phases.
- Automatize validation of data dependencies during the Deployment of the predictive function in the operation environment.

In order to achieve these objectives, Cross-sectorial domain model support:

a) Presentation in the human-readable form using the Semantic framework tools developed in work package 4. Semantic framework tools allow creation, visualization, querying, filtering and navigation of the Cross-sectorial domain model elements by data scientists and domain experts.
b) Machine-readable format with the formally defined semantics, which support automatic validation of deployment, quality evaluation of the predictive functions and simulation and automatic optimization of the production processes.

Deliverable D2.7 further describes the overall process of the data mining and analysis process and defines how the phases of the process can be mapped to the concepts of the Cross-sectorial domain model.The following table provides revised summary of the concepts' mapping.

Data analysis phase	Cross-sectorial domain modules	Involved roles
Problem understanding	Production process modelling, Data and KPI modelling, Predictive function modelling	Domain experts, Data scientists
Data understanding	Data modelling	Domain experts, Data scientists, IT experts
Modelling	Predictive function modelling	Data scientists
Evaluation	Data modelling, KPI modelling, Predictive Function modelling	Domain experts, data scientists
Deployment	Data modelling, Predictive function modelling	IT experts

Table 1 – Mapping of the Cross-sectorial domain model concepts to the data analysis process.

3 FinalCross-sectorial domain model

This chapter provides formal specification of enhancements and extensions of the initial version of Cross-sectorial domain model specified in deliverable D2.7. Most extensions introduced in the final version of the model are aimed at better and more detailed description of the predictive functions. While initial model enabled to specify the predictive function, input data, output data and evaluation results, the idea was to extend the predictive functions description in more detailed fashion including model structure and performance metrics. For that purposes, we decided to evaluate existing ontologies used for modelling of data mining (both, process and data mining models). Another major extension is the definition of the meta-data model for knowledge organization system (KOS). All concepts of the Cross-sectorial Domain model can be now organized in hierarchies and can be described as the entries of the controlled vocabulary using the multi-lingual labels and scope definitions.

The main tool introduced by the Cross-sectorial Domain model to improve communication between the users is the shared vocabulary collaboratively created by the domain experts and data scientists. The domain concepts are described as the vocabulary entries which unambiguously define their meanings. Each entry has assigned a primary label presented as the unique title representing the concepts for the user. Optionally, each entry can also list additional labels such as synonyms, acronyms or abbreviations representing the same concept in the particular natural language(s). Besides the definition of the meaning, vocabulary can be used as the knowledge organization system (KOS), i.e. the entries can be used as the indexing keywords for the organization of any documents or other resources (e.g. datasets, scripts, etc.), which contain the relevant information about the domain or solved business problem. The concepts in the vocabularyare further organized in the hierarchical structure (taxonomy) from broader concepts to the narrower. Taxonomy structure allows efficient navigation in the domain knowledge and retrieval of relevant information.

Structure of the Cross-sectorial Domain model vocabulary is based on Simple Knowledge Organization System (SKOS) standard with the extensions defined in the JSKOS standard, which define JSON (JavaScript Object Notation) format for the knowledge organization systems such as taxonomies. All domain concepts are represented as the objects of the JSKOS Concept type which formally defines fields of the vocabulary entries and relations between them (e.g. narrower/broader, previous/next, related, etc.).

The structure of the controlled vocabulary is domain independent and the proposed JSKOS format can be used to describe concepts in any domain. For the application in the process industry domain, Cross-sectorial Domain model defines core of the vocabulary with the basic top concepts for the representation of the main entities in the process industry. The top concepts can be divided to:

- **Production process modelling** – specifies decomposition of production processes to segments (production phases) and describes resources required for the production.
- **Data modelling** – specifies concepts for description of the data elements and key-performance indicators.
- **Predictive function modelling** – specifies concepts for description of the data analytics models.

Besides the shared vocabulary, execution workflow of the production processes and their decomposition to the activities and sub-processes are usually modelled using the graphical notation in the form of various schemas or workflow diagrams. Structure of the vocabulary specified by Cross-sectorial Domain model allows to link each graphical object from the diagram to the definition entry in the shared vocabulary, i.e. it is possible to navigate from the process model to the definition of the related concepts or from the concept (e.g. data element, KPI or predictive function) to the relevant part of the graphical process model. As the reference graphical notation for the modelling of the production processes the Cross-sectorial Domain model adoptedthe subset of the BPMN 2.0 standard, which is also implemented in the Semantic framework modelling tools in deliverables D4.1 and D4.2.

To summarize, the overall structure of the proposed Cross-sectorial Domain model is presented on the following schema (Figure 1). Bottom layer provides interoperability with the linked data and semantic web technologies based on the RDF and JSON-LD standards. Over this standard knowledge model, we have defined the meta-model for Knowledge Organization System based on the SKOS and JSKOS standards respectively. Using this meta-model, we have defined common Vocabulary for the Process Industry domain which describes Production Processes and Production Resources. The domain concepts are complemented by the general data analytics concepts adopted from the DMOP (Data Mining OPtimization) ontology, which are used for the description of the Data Elements, KPIs and Predictive Functions. Finally, definition of the production processes is extended with the graphical notation based on the simplified BPMN diagrams.

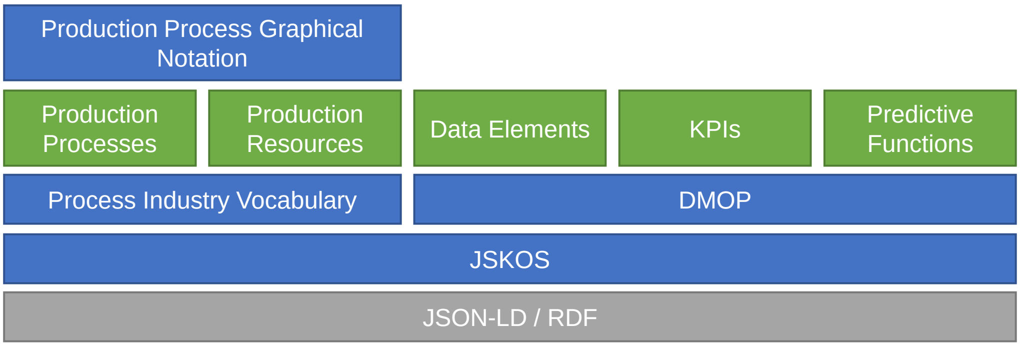
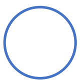
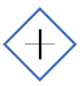


Figure 1 – Overall structure of the Cross-sectorial Domain model.

3.1 Process graphical notation

The suitable form for the modelling of the knowledge about the production processes is to use diagrams where the process steps, their execution dependencies and involved resources are represented as the interconnected graphical objects. The Cross-sectorial Domain Model is not limited to the particular graphical notation and can be used to describe processes using the various kind of schemas, workflow diagrams or dynamic visualizations presented in the process visualization software (e.g. InTouch screens).

As the reference graphical notation for the implementation of the MONSOON Semantic framework, we have adopted subset of BPMN 2.0 standard[1]. The list of selected elements is presented in the following table.

Element	Description
	Start event.
	End event.
	Sequence flow, which defines execution order of activities.
	Default flowis the default branch to be chosen if all other conditions evaluate to false.
	Exclusive gatewayroutes the sequence flow to exactly one of the outgoing branches.
	Parallel gateway routes execution flow to all outgoing branches and all branches are activated simultaneously.

Task	A Task is a unit of work, the job to be performed.
Subprocess ⊞	Sub-Process is an activity that can be refined.

Table 2 – Process elements description.

3.2 Domain concepts

In order to define unified model for the various domains, all domain concepts are modelled using the meta-concepts specified in the generic knowledge organization system (KOS). As the meta-model for the domain concepts, we have adopted JSKOS which defines a JavaScript Object Notation (JSON) structure for the various types of the knowledge organization systems such as classifications, thesauri, and authority files. JSKOS supports encoding of the concepts, concept schemes, concept occurrences and concept mappings. The main part of JSKOS is compatible with Simple Knowledge Organisation System (SKOS) and JavaScript Object Notation for Linked Data (JSON-LD). A simple JSKOS document can be mapped to SKOS expressed in the Resource Description Framework (RDF), and vice versa.

JSKOS is based on JSON which consists of objects with pairs of fields and values, arrays with members, strings, numbers, and the special values true, false, and null. All strings and fields of a JSKOS document must be stored in Unicode format. Besides the types defined by JSON, JSKOS further defines the following types:

- **URI** – is syntactically correct URI
- **URL** – is syntactically correct URL with HTTPS or HTTP scheme.
- **non-negative integer** – is a JSON number without preceding minus part, without fraction part, or exponent
- **percentage** – is a JSON number with value between zero (0%) and one (100 %)
- **date** – is a date or datetime as defined by XML Schema datetime, date, gYearMonth or gYear
- **list** – is a possibly empty array of strings and an optional last member null
- **set** – is a possible empty array where all members are JSKOS resources, except the last member optionally being null, and have distinct values in field uri (i.e. members must not be the same resources)
- **language range** – is either the string "-" or a string that conforms to the syntax of RFC 3066 language tags, limited to lowercase, followed by the character "-"
- **language map** – is a JSON object in which every key is either RFC 3066 language tag in lowercase or a language range and which has only string values.

The meta-model of domain concepts is defined by the following types:

- Resources for all kind of entities
    - Items for named entities
        - Concepts for entities from a knowledge organization system
        - Concept schemes for compiled collections of concepts

3.2.1 Resource

A Resource is a JSON object with the following optional fields:

Field	Type	Description
uri	URI	Primary globally unique identifier.
identifier	list	Additional identifiers.

type	list of URIs	URIs of types.
@context	URI	Reference to a JSON-LD context document.
created	date	Date of creation.
issued	date	Date of publication.
modified	date	Date of last modification.
creator	set	Agent primarily responsible for creation of resource (i.e. user identity of the domain expert or data scientist).
contributor	set	Agent responsible for making contributions to the resource.
publisher	set	Agent responsible for making the resources available.
partOf	set	Resources which this resource is part of (if no other field applies).

Table 3 – Resource element description.

Core of the Resource fields are based on the Dublin Core metadata properties and are used to document vocabulary concepts itself. For the compatibility with the JSON-LD, it is recommended that all resources will include at least the fields uri, type, and @context. The value of field @context should be set to https://gbv.github.io/jskos/context.json.

General types and their relations are presented on the following diagram.

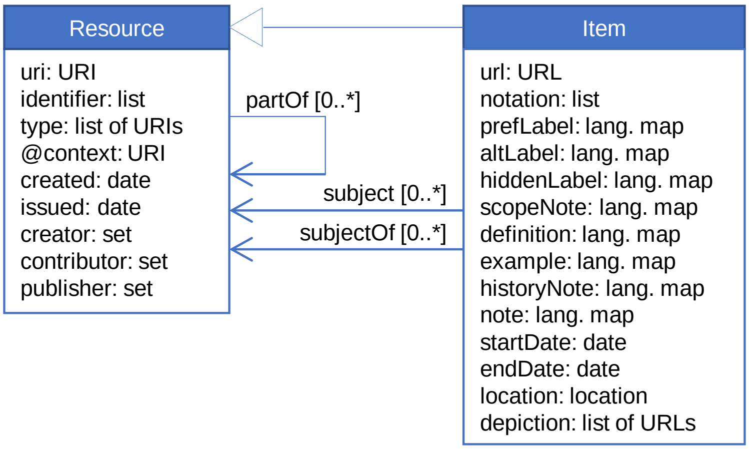


Figure 2 – General types and relations.

Types used to define controlled vocabulary, i.e. Concept and Concept Scheme are presented on the following diagram.

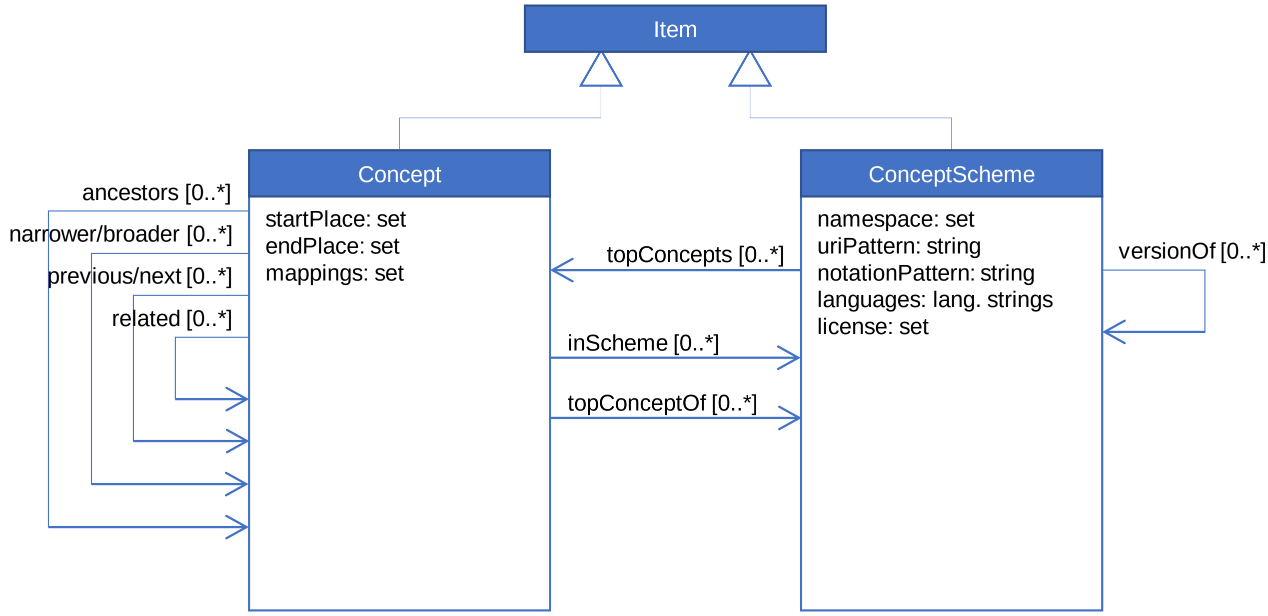


Figure 3 – Concept and Concept Scheme relations.

3.2.2 Item

An Item is a Resource with the following optional fields (in addition to the optional fields defined for the Resource):

Field	Type	Description
url	URL	URL of a page with information about the item.
notation	list	List of notations. Some Resources can be recognized by *notations* (or *captions*) which represent various codes for the classification of the Resources. Notations are symbols which are not normally recognizable as words or sequences of words in any natural language and are thus usable independently of natural-language contexts. They are typically composed of digits, complemented with punctuation signs and other characters.
prefLabel	language map of strings	Preferred lexical labels indexed by language. It is recommended that no two concepts in the same Concept Schema be given the same preferred lexical label for any given language tag.
altLabel	language map of lists	Alternative lexical labels indexed by language. This property can be used to assign synonyms, acronyms or abbreviations, which can be alternatively used to express the concept besides the preferred label.
hiddenLabel	language map of lists	Hidden labels indexed by language. Hidden labels can be used to express labels which are indexed and are available for text-based search operations but would not be visible otherwise. Hidden labels may be for instance used to include misspelled variants of other lexical labels.

scopeNote	language map of lists	Supplies some, possibly partial, information about the intended meaning of a concept.
definition	language map of lists	Supplies a complete explanation of the intended meaning of a concept.
example	language map of lists	Supplies an example of the use of a concept.
historyNote	language map of lists	Describes significant changes to the meaning or the form of a concept.
changeNote	language map of lists	Documents fine-grained changes to a concept, for the purposes of administration and maintenance.
note	language map of lists	Property for general documentation purposes which can be further specialized to scopeNote, definition, example, historyNote.
startDate	date	Date of creation or establishment of the item.
endDate	date	Date of resolution of the item.
relatedDate	date	Other date related to the item.
location	location	Geographic location(s) of the item.
subject	set	What this item is about (i.e. topic).
subjectOf	set	Resources about this item (e.g. documentation).
depiction	list of URLs	List of URLs of images or image elements depicting the item.

Table 4 – Item element description.

3.2.3 Concept

A Concept is an Item with the following optional fields (in addition to the optional fields of Item and Resource):

Field	Type	Description
narrower	set	Narrow concepts.
broader	set	Broader concepts.
related	set	Generally related concepts.
previous	set	List of concepts ordered somewhere before the concept.
next	set	List of concepts ordered somewhere after the concept.
ancestors	set	List of ancestors, possible up to a top concept.
inScheme	set	Concept schemes or URI of the concept scheme.
topConceptOf	set	Concept schemes or URI of the concept schemes.
startPlace	set	Where a concept started (e.g. place of establishment).

endPlace	set	Where a concept ended (e.g. place of resolution).

Table 5 – Concept element description.

The first element of the type field must be URI http://www.w3.org/2004/02/skos/core#Concept indicating that each JSKOS concept is SKOS concept.

3.2.4 Concept Scheme

A Concept Scheme is an Item with the following optional fields (in addition to the fields defined for Item):

Field	Type	Description
topConcepts	set of concepts	Top concepts of the scheme.
versionOf	set of concept schemes	Concept scheme which this scheme is a version or edition of.
namespace	URI	URI namespace that all concepts URIs are expected to match.
uriPattern	string	Regular expression that all concept URIs are expected to match.
notationPattern	string	Regular expression that all primary notations should follow.
extend	string	Size of the concept scheme.
languages	list of language strings	Supported languages.
license	set	Licenses which the full scheme can be used under.

Table 6 – Concept scheme description.

The first element of the type field must be URI http://www.w3.org/2004/02/skos/core#ConceptScheme indicating that each JSKOS concept schema is SKOS concept schema.

3.3 Vocabulary for the process industry

The vocabulary for the description of the process industry applications is based on the B2MML standard for the modelling of the manufacturing processes [2]. It defines the following main types, which all extend Item type:

- Production process and Production segment
- Production resource
    - Equipment
    - Personnel role

The main types specified for the process industry are presented on the following diagram:

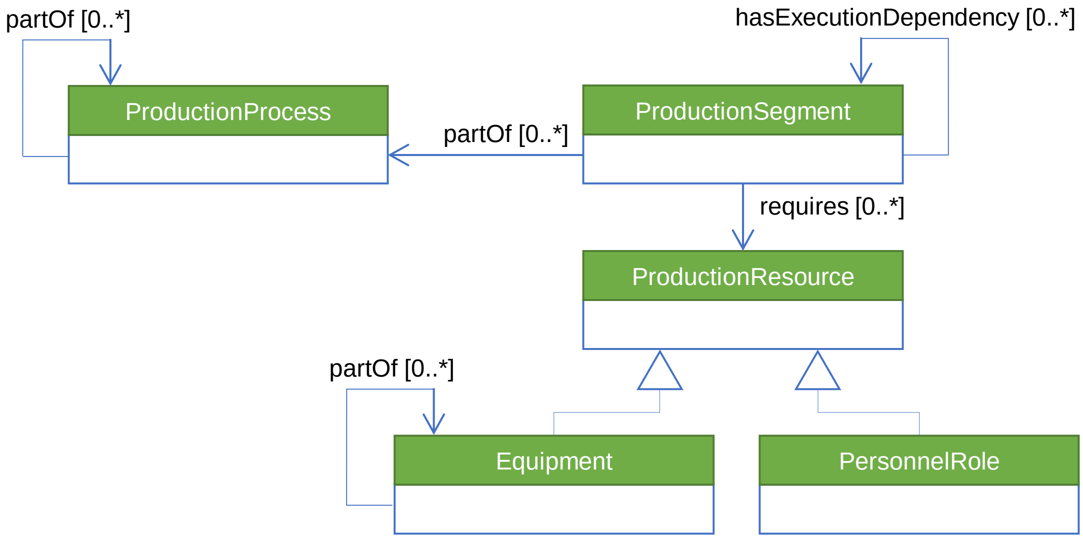


Figure 4 – Concepts for description of process industry domain.

3.3.1 Production Process

Production process is a container composed of Production Segments (steps) ordered in the execution workflow. Production process can be further decomposed to sub-processes. This is represented by the compound Production Segment which delegates the execution to the underlying sub-process. The Production Process is a Concept with the following optional fields:

Field	Type	Description
partOf	set	Refers to the Process for which this Process is a sub-process.
broader	set	The taxonomy of Production Process concepts corresponds to the decomposition of the production processes to sub-processes, i.e. broader field is equivalent to partOf field.
narrower	set	The narrower field is inverse to the partOf field.

Table 7 – Production process elements description.

3.3.2 Production Segment

Process Segment is a logical grouping of personnel and equipment resources required to carry out a production step. It defines what types of personnel and equipment are needed and it can describe specific resources (e.g. some specific type of equipment for particular segment). Process Segments are ordered in the execution workflow by execution dependency relation. The Production Segment is a Concept with the following optional fields:

Field	Type	Description
partOf	set	Refers to the Production Process which consists of this Production Segment.
broader	set	The taxonomy of Production Segment concepts corresponds to the decomposition of the processes to the Production Segments, i.e. broader field is equivalent to thepartOf field.

hasExecutionDependency	set	Represents execution workflow order between the Production segments (steps), i.e. segment 2 has execution dependency with segment 1 if segment 2 can be executed only after the segment 1 is finished.
previous	set	Ordering of the Production Segment concepts corresponds to the execution workflow order, i.e. previous field is equivalent to the hasExecutionDependency field.
next	set	The next field is inverse to the hasExecutionDependency field.
requires	set	List of Production Resources (Equipment or Personnel Role concepts) required to perform this Production Segment.

Table 8 – Production Segment element description.

3.3.3 Production Resource

Production Resource class is a common super-type for the concepts representing resources required to carry out a production step. Grouping of resources with similar characteristics and purposes can be described by Resource Classes. Any piece of production resource may be a member of zero or more classes.

3.3.4 Equipment

Equipment concept covers sites, areas, production units, production lines, work cells, process cells, or units. Equipment may consist of other equipment, e.g. production line may be made up of work cells. Each may be defined as a separate equipment element with separate properties and capabilities.

Field	Type	Description
partOf	set	Represents hierarchical decomposition of the Equipment to sub-pieces. Decomposition relations must be constrained to acyclic hierarchy.
narrower	set	The taxonomy of Equipment concepts corresponds to the decomposition of the equipment to sub-pieces, i.e. narrower field is equivalent to partOf field.
broader	set	Broader field is inverse to the partOf field and points to the Equipment which consists of this sub-piece.

Table 9 – Equipment element description.

3.3.5 Personnel Role

Personnel Role concept covers any human roles involved in the Production Segment and describes their capabilities.

Besides the types for the description of the properties of concepts from the production industry domain, Cross-sectorial Domain model also provides core vocabulary which consists of the main top categories dividing all domain concepts Production Processes and Production Resources. The vocabulary is represented as the JSKOS Concept Scheme and each category is represented as the JSKOS Concept. Note that it is necessary to distinguish types for the domain concepts (Production Process, Production Resource, etc.) and category concepts (Production Process Concept, Production Resource Concept, etc.). All specific concepts of the given type are linked as the narrower concepts of the corresponding category, e.g. each specific process concepts (e.g. "Green Anode Production") has type Production Process and are linked as the narrower concept to Production Process Concept category. The structure of the core concept scheme is presented on the following diagram.

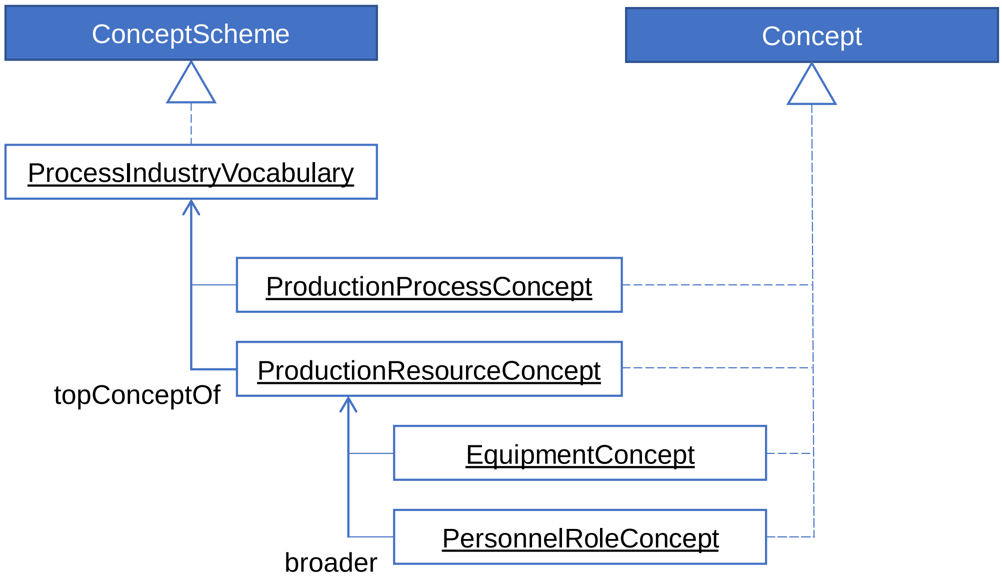


Figure 5 – Structure of core concept scheme.

3.4 Data mining concepts

The majority of the ontological models used to describe the data mining models are aimed to support the overall data mining workflows. Several other models are available, which provide more complete frameworks for data mining e.g. OntoDM, DMOP or Exposé ontologies [3,4,5]. Those provide both, rather general concepts related to the data mining tasks, datasets, etc., as well as more in-depth and detailed entities related to algorithms or experiments. OntoDM is aligned with upper-level ontologies, which makes it more general. Regarding the description of the data mining models, DMOP covers detailed description of the data mining algorithms and its internal structure. Exposé then combines both approaches and leverages on OntoDM more general structure combined with DMOP algorithms description. The core concepts of the DMOP ontology are:

- *DM-Task* – specification of any activity that is a part of data mining process
- *DM-Algorithm* – specification of realization of the particular *DM-Task*
- *DM-Model* – specification of DM-Algorithm output – the model
- *DM-Operator* – specifies a program which implements the *DM-Algorithm*
- *DM-Workflow* – specifies the structure of the *DM-Operators*

3.4.1 Data mining tasks

Data mining tasks are specified by inputs and outputs. Both are represented by *IOClass*. This may include both *DM-Data* or *DM-Models*, e.g. input to data pre-processing task is data, as well as output. Predictive model training task transforms input data to an output model. Tasks are then addressed by an algorithm (*DM-Algorithm*).

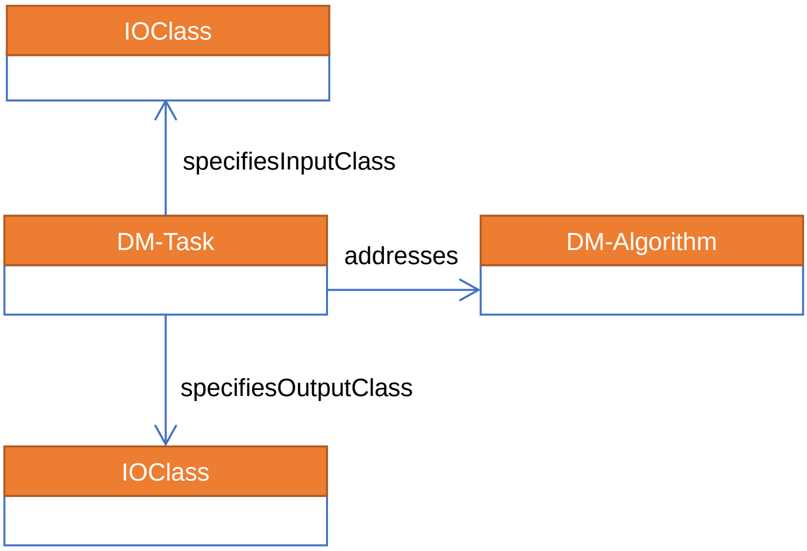


Figure 6 – Task and its relation to inputs/outputs and algorithms.

Core-DM-Task concept cover the following hierarchy of the most common data mining tasks:

- DataProcessingTask
    - DataTransformationTask
        - FeatureExtractionTask
        - ClassBalancingTask
        - FeatureTransformationTask
        - FeatureDiscretizationTask
        - FeatureWeightingTask
        - FeatureRankingTask
    - DataCleaningTask
        - NoiseReductionTask
        - MissingValueImputationTask
    - DataReductionTask
        - CardinalityReductionTask
            - SubsamplingTask
        - DimensionalityReductionTask
            - FeatureSelectionTask
    - DataAbstractionTask
- HypothesisProcessingTask
    - PatternSetProcessingTask
        - PatternPruningTask
    - ModelProcessingTask
        - ModelAggregationTask
        - ModelPruningTask
- InductionTask
    - PatternDiscoveryTask
        - AssociationDiscoveryTask
        - DeviationDetectionTask

        - DissociationDiscoveryTask
        - SubgroupDiscoveryTask
    - ModelingTask
        - PredictiveModelingTask
            - StructurePredictiveTask
            - ClassificationModelingTask
            - RegressionModelingTask
        - Descriptive Modeling Task
            - ProbabilityEstimationTask
            - DependencyModelingTask
            - ClusteringModelingTask
- HypothesisEvaluationTask
    - ModelEvaluationTask
    - PatternSetEvaluationTask
- HypothesisApplicationTask
    - ModelApplicationTask
        - DescriptionTask
            - ClusteringTask
        - PredictionTask
            - StructurePredictionTask
            - RegressionTask
            - ClassificationTask
    - PatternSetApplicationTask

ModelingTask concept then can be divided into the sub-classes, referring to the particular data mining task types such as *DescriptiveModellingTask* for description of descriptive tasks (e.g. clustering, generalization), *PatternDiscoveryTask* (e.g. sequence pattern mining, frequent items discovery) or predictive tasks. From the perspective of predictive functions in MONSOON project, modelling of predictive data mining is important. *PredictiveModellingTask* recognizes three different predictive sub-tasks:

- *ClassificationModellingTask* for tasks when predicting the nominal target attribute
- *RegressionModellingTask* for tasks when predicting the numeric target attribute

3.4.2 Data mining algorithm and models

Each task specifies a particular step in data mining process and it is realized by an algorithm. *DM-Algorithm* concept (or its sub-classes) describes the particular learning algorithm. Algorithm characteristic and parameters then specify various aspects of the model induction algorithm, e.g. type of the solved problem, tolerance to noise in the input data, tolerance to missing values, etc. and are represented by *hasQuality* and *hasParameter* properties. *DM-Model* then represents the output of a learning algorithm – a model. *DM-Model* class (or its subclasses) then have specified *ModelStructure* and *ModelParameter*. *ModelStructure* describes the nature of the model, e.g. logical structure (tree, rules, etc.) or mathematical expression (e.g. kernel functions, etc.). *ModelParameter* on the other hand specifies the various parameters for a given model type. Each specific model type has a specific set of model parameters. Such parameters can include e.g. weights in neural network models, threshold splits in tree models, etc. *ModelComplexityMeasure* class then contains concepts to quantify the model complexity. In similar fashion, specific model types with specific parameters have assigned particular complexity measures. In case of tree classification model, it could be depth of a tree, or number of leaves, in case of SVM model number of support vectors or sum of their weights. Simplified conceptualization of the ClassificationModel structure is depicted in Fig.7.

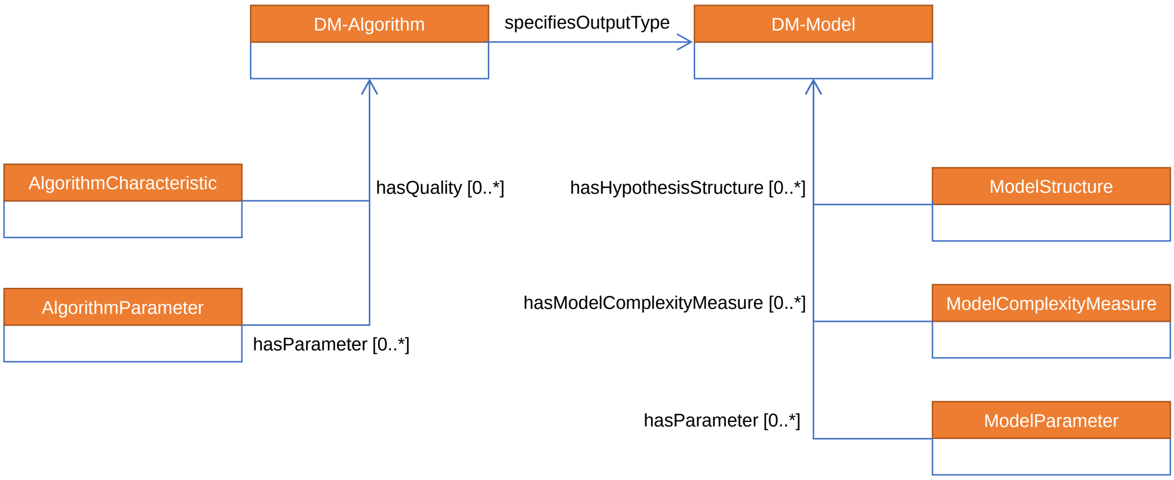


Figure 7 – DM-Algorithm and DM-Model concepts relation.

AlgorithmParameter concept then provides a hierarchy of different parameters specific to concrete algorithms. Compound model parameters can be specified using Operators concept from DMOP, e.g. model combinationoperators (in bagging or boosting).

- AlgorithmParameter
  - K-MeansParameter
  - KNN-Parameter
  - NNParameter
  - SVC-Parameter
    - KernelTypeParameter
  - TreeInductionParameter
    - LeafSizeParameter
    - SplitTypeParameter
    - TreePruningParameter
- ModelStructure
  - ProbabilisticModelStructure
    - JointProbDistribution
    - PosteriorProbDistribution
  - NonprobabilisticModelStructure
    - LogicalStructure
      - DisjunctionOfConjunctions
    - MathematicalExpression

ModelParameter then covers theparameters of a learned model – contrary to runtime parameters of a model learning algorithms.

- ModelParameter
  - ClassBasedModelParameter
    - ClassPriorVector
  - FeatureBasedModelParameter
    - ClassCondMeanMatrix
    - SetOfSplitThresholds
    - SetOfWeightVectors
    - ...
  - InstanceBasedModelParameters
    - InstanceWeightVector
- ModelComplexityMeasure
  - ModelParameterCount
    - NumInternalNodes

- NumLeaves
- TreeDepth
- NumProbabilities
- NumWeights
- ...

- ModelParameterMagnitude
    - SumSquaredWeights
    - SumWeights

3.4.3 Data mining algorithm class hierarchy

DM-Algorithm concept provides a hierarchy of data mining algorithms. The algorithms are grouped and mapped to the corresponding data-mining task e.g. *ClassificationModellingTask* refers to the *ClassificationModelAlgorithm*. The classification models are grouped to three different types corresponding to the Generative Models, Discriminative Models and Discriminant Function Algorithms and then respective algorithms or algorithm groups follows such structure. Class hierarchy for Classification algorithms is depicted on Fig.8.

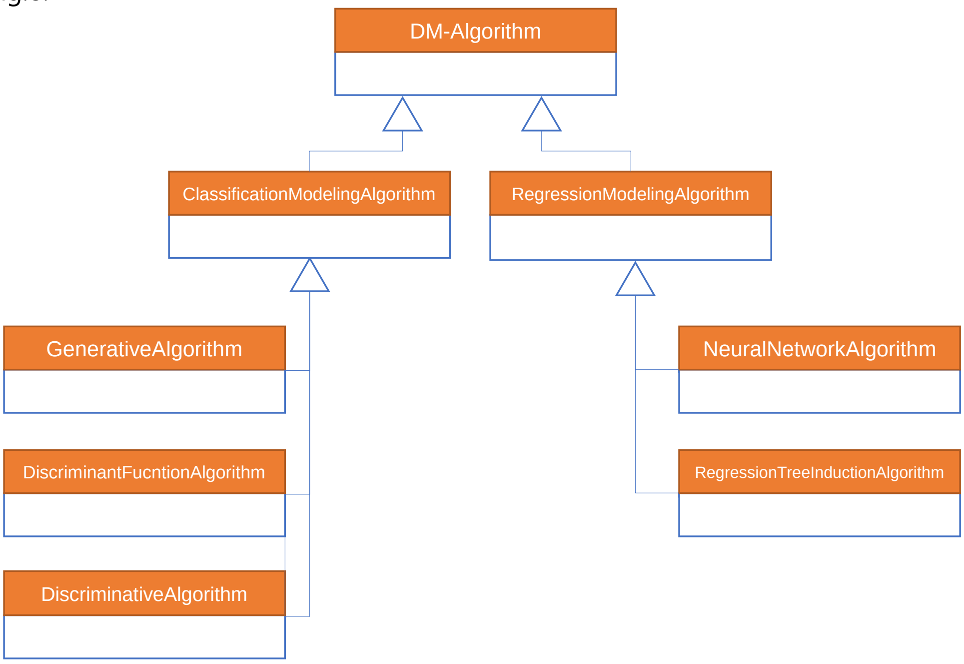


Figure 8 – DM-Algorithm class hierarchy.

Hierarchy then recognizes different types of the models. For example, class hierarchy for classification models:

- ClassificationModelingAlgorithm
    - GenerativeAlgorithm
        - NormalLinearDiscriminant
        - NormalQuadDiscriminant
        - NaiveBayes
            - NaiveBayesNormal
            - NaiveBayesKernel
            - NaiveBayesMultinomial
            - ...
    - DiscriminativeAlgorithm
        - KNearestNeighbors
        - LogisticRegression

    - DiscriminantFunctionAlgorithm
        - SupportVectorClassifier
            - LinearSVC
            - SoftMarginSVC
            - ...
        - NeuralNetworks
            - MLP-Backprop
        - SetCovering
        - RecursivePartitioning
            - CART
            - C4.5
            - ...

3.4.4 Data mining model class hierarchy

Each particular algorithm when applied on the training data produces a model. For example, *ClassificationModellingAlgorithm* produces as an output a *ClassificationModel*, e.g. *C4.5Algorithm* produces an output *C4.5Model*. DMOP contains the concepts to describe various characteristics of these models, namely structure of the model and its parameters. DM-Model class then presents similar class hierarchy as the DM-Algorithm class (see Fig.9).

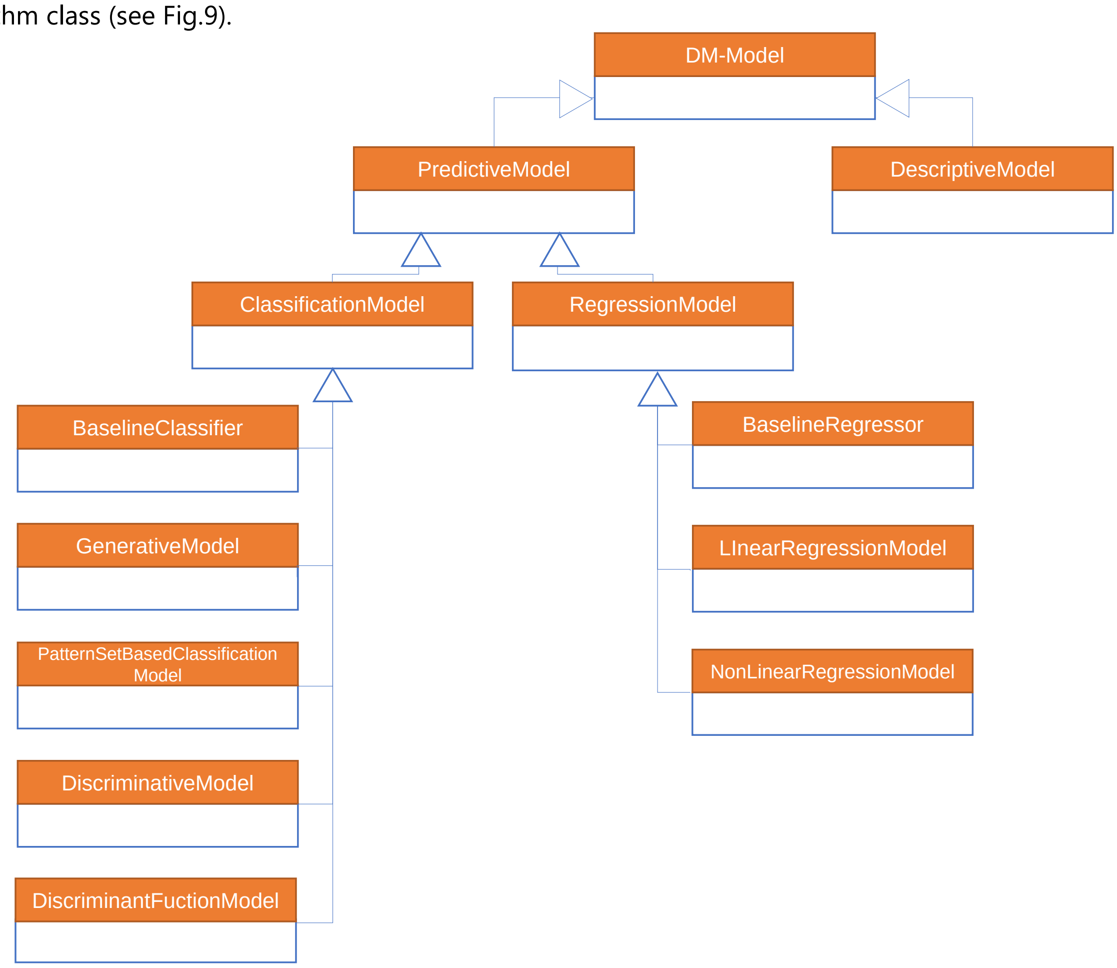


Figure 9 – Predictive models class hierarchy.

Hierarchy then follows similar structure as algorithm structure. *ClassificationModel* class hierarchy:

- ClassificationModel
    - BaselineClassifier
    - DiscriminativeModel

      - KNearestNeighborModel
      - LogisticRegressionModel
  - DiscriminantFunctionModel
      - C4.5CrispTreeModel
      - CARTClassificatioModel
      - ID3Model
      - ...
  - GenerativeModel
      - BayesianModel
          - BayesNetModel
          - NaiveBayesModel
              - NaiveBayesNormalModel
              - NaiveBayesKernelModel
              - ...
      - LinearDiscriminantModel
      - QuadraticDiscriminantModel
  - PatternSetBasedClassificationModel

3.4.5 Example

To present, how a concrete predictive function could be described using concepts presented in previous paragraphs, take as an example a C4.5 tree classification model (see Fig. 10). In this case *C4.5* is a concrete instance of *C4.5Algorithm* class. *C4.5Algorithm* class has specified a property *specifiesOutputType* which describes the input training data for the algorithm. When trained on these data, *specifiesOutputType* property defines, that the output of this algorithm is a *C4.5Model*, a tree classification model. Such model can be then described from the perspective of its structure (*hasModelStructure* property), parameters (*hasModelParameter* property) or complexity metrics (*hasComplexityMetric* property). In this case, model structure is in form of disjunction of conjunctions, parameters describe *SetOfSplitThresholds* (describing the concrete split threesholds in the tree model). The complexity of the model can be expressed via number of nodes and leaves.

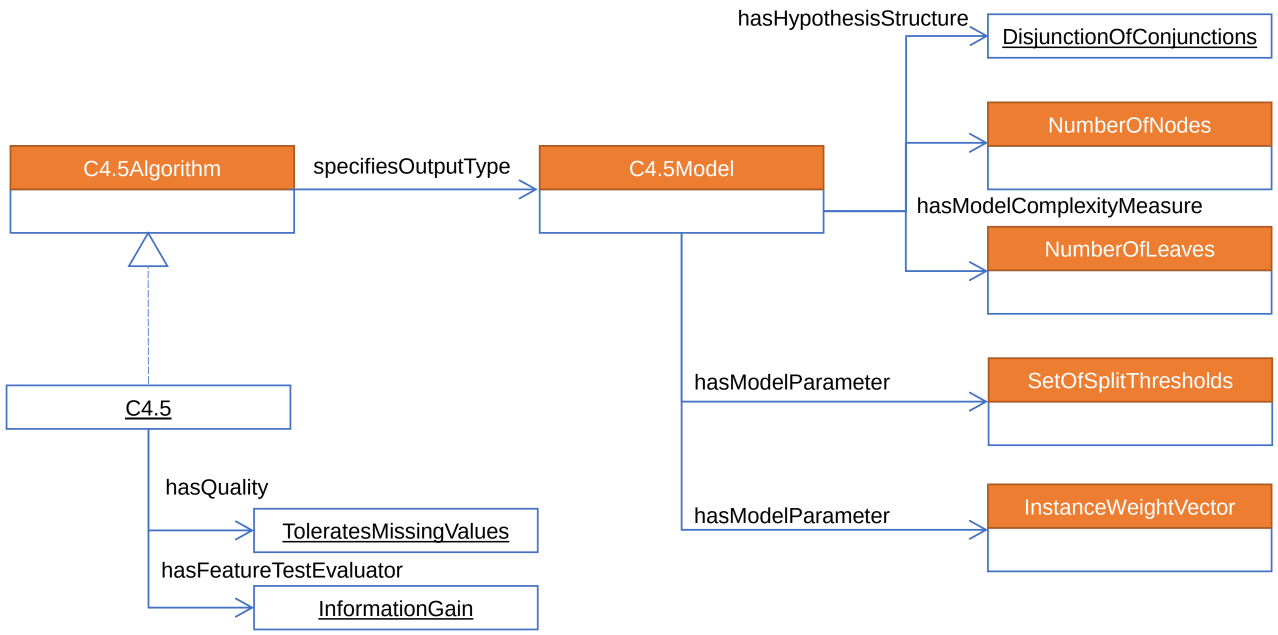


Figure 10 – Example of C4.5 tree algorithm and model representation.

4 Case study – Aluminium domain

This section presents the application of MONSOON semantic model on one of the project's application domains. In this case we present the aluminium domain models from the electrolysis use case. Following sub-sections demonstrate the JSON-LD formalization of process models including main production process and sub-processes, data elements, KPIs and predictive functions. Models were created using MONSOON Semantic Modeller. Initial version of the tool was described in deliverable D4.1*InitialSemantic Framework* [6]. Fig. 11 illustrates the MONSOON Semantic Modeller process view and presents the top-level process model in electrolysis use case. Complete demonstration of the Semantic Modeller tool will be contained in deliverable D4.2*Final Semantic Framework*.

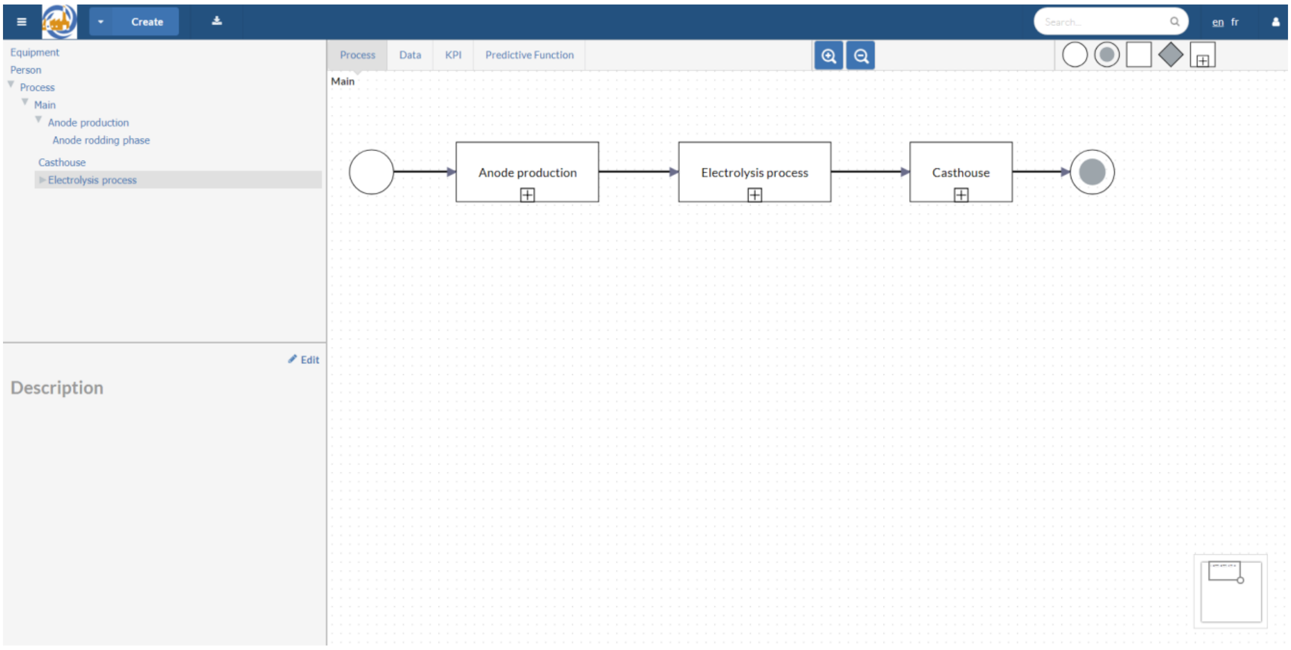


Figure 111 – MONSOON Semantic Modeller process view in electrolysis domain.

4.1 Production process models

Following examples presents the elements related to the production processes. This includes modelling using tasks (or sub-processes) and process flow-related objects.

This example illustrates the main process (top level process), in electrolysis use case. In this case, main process consists of three sub-processes, which are subsequently modelled in similar fashion. Each element represents the particular sub-process. *HasExecutionDependency* property then specifies previous process steps (previous activity). While *narrower/broader* properties specify the position within the domain concept taxonomy and relation to more specific/general concepts. Fig. 10 then depicts the overall process diagram using simplified BPMN process-like model elements described in Tab. 1.


```
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "",
```

```
"partOf": "",
"narrower": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"broader": "",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-5e00b8d2-6548-431c-824f-9431e930f164",
"previous": "",
"definition": {
"en": ""
    },
"identifier": "99f09f55-7935-400c-9b0e-cfd02a4175f1",
"prefLabel": {
"en": "Anodeproduction"
    }
}

{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"partOf": "",
"narrower": "https://monsoon.ekf.tuke.sk/modeller/process-5e00b8d2-6548-431c-824f-9431e930f164",
"broader": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"next": "",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-5e00b8d2-6548-431c-824f-9431e930f164",
"definition": {
"en": ""
    },
"identifier": "344e039c-3209-4818-99e0-eb985ad5ff7b",
"prefLabel": {
"en": "Electrolysisprocess"
    }
}
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-5e00b8d2-6548-431c-824f-9431e930f164",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"partOf": "",
"narrower": "",
"broader": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"definition": {
"en": ""
    },
"identifier": "5e00b8d2-6548-431c-824f-9431e930f164",
"prefLabel": {
"en": "Casthouse"
    }
}
```

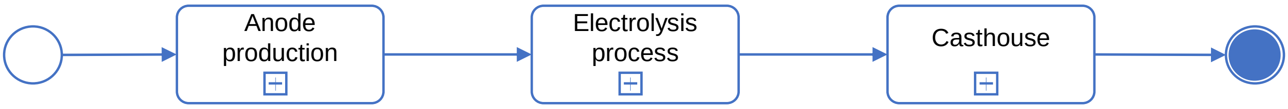


Figure 12 – Top process in aluminium domain.

Followingsectionpresentstheformalization of theAnodeProductionsub-processfromthe top process model. AnodeProductionsub-processismodelled as a separateprocess model consisting of separateactivities, in thiscase, sub-processes. Overallstructure of theAnodeProductionsub-processisdepicted in Fig. 11. Eachtask or sub-processismodelled in a samefashion as a top level process. *PartOf*propertyspecifiestheURI of thesub-process on a top level, whichrepresents a parenttaskfor a givenprocess model element. In followingcase, allprocesselements (each of sub-processes) are a part of Anodeproductionsub-process.


```
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-89c6355f-2ed2-4a76-aab9-d8fe8b5f5b67",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "",
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"narrower": "https://monsoon.ekf.tuke.sk/modeller/process-b332a33c-4001-4062-85d5-1ac05a82db3a",
"broader": "",
"next": "",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-b332a33c-4001-4062-85d5-1ac05a82db3a",
"definition": {
"en": ""
    },
"identifier": "89c6355f-2ed2-4a76-aab9-d8fe8b5f5b67",
"prefLabel": {
"en": "Greenanode\nproductionprocess"
    }
},
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-b332a33c-4001-4062-85d5-1ac05a82db3a",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "https://monsoon.ekf.tuke.sk/modeller/process-89c6355f-2ed2-4a76-aab9-
d8fe8b5f5b67",
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"narrower": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"broader": "https://monsoon.ekf.tuke.sk/modeller/process-89c6355f-2ed2-4a76-aab9-d8fe8b5f5b67",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-89c6355f-2ed2-4a76-aab9-d8fe8b5f5b67",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"definition": {
"en": ""
    },
"identifier": "b332a33c-4001-4062-85d5-1ac05a82db3a",
"prefLabel": {
"en": "Bakingprocess"
    }
},
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"@context": "https://gbv.github.io/jskos/context.json",
```

"hasExecutionDependency": "https://monsoon.ekf.tuke.sk/modeller/process-b332a33c-4001-4062-85d5-1ac05a82db3a",
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-8d25544b-b6a9-4de3-8272-0140869a89d2",
"narrower": "",
"broader": "https://monsoon.ekf.tuke.sk/modeller/process-b332a33c-4001-4062-85d5-1ac05a82db3a",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-b332a33c-4001-4062-85d5-1ac05a82db3a",
"previous": "",
"definition": {
"en": ""
    },
"identifier": "8d25544b-b6a9-4de3-8272-0140869a89d2",
"prefLabel": {
"en": "Anoderoddingphase"
    }
}

Figure 13 – Anode production sub-process model.

Followingsectionpresentstheformalization of theElectrolysissub-processfromthe top process model. Thissub-processismodelled as a separateprocess model consisting of separateactivities, in thiscase, both, tasks and sub-processes. Overallstructure of theElectrolysissub-processisdepicted in Fig. 13. In similarfashion as Anodeproductionsub-process, sameproperties are used to specifythe top-level process element.

{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-e9d5785d-8692-4076-9e7c-577721ce5711",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "",
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"narrower": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"broader": "",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"definition": {
"en": "Theanode are consumedcontinuously by theelectrolysisreaction and mustbechanged on a regularbasis. Thisoperationiscalledanode change and is done withthe Pot tendingmachine (cranes in theelectrolysishalls). Thisoperationismonitoredviatheprocesscontrolsystem (ALPSYS)."
    },
"identifier": "e9d5785d-8692-4076-9e7c-577721ce5711",
"prefLabel": {
"en": "Anode change"
    }
},
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "https://monsoon.ekf.tuke.sk/modeller/process-e9d5785d-8692-4076-9e7c-577721ce5711",

```
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"narrower": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"broader": "https://monsoon.ekf.tuke.sk/modeller/process-e9d5785d-8692-4076-9e7c-577721ce5711",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-aa74e9bd-36ce-40d0-a27f-d06a95474ac1",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-e9d5785d-8692-4076-9e7c-577721ce5711",
"definition": {
"en": "Theanodecanbeoxidentifierized by theoxygenthusconsuming a part of theanodevia a sidentifierereaction. Thisnon-desiredreactionisavoidentifieredthanks to a goodcovering of theanode. Furthermore, theanodecovering has animportantimpact on the pot thermalbalance/isolation. Thisoperationiscalledanodecovering and is done withthe Pot tendingmachine (cranes in theelectrolysishalls). Thisoperationismonitoredviatheprocesscontrolsystem (ALPSYS).\n\nIftheanodeisexposed to theair, wecanexpectanincrease of theanoderesistance, impactingtheaverage pot resistance, leading to more downward orders of theanodebeam, reducingtheanode-cathodedistance and consequentlythecurrentefficiency.\n\nIfthesolidentifierbath (crust) collapses in theliquidentifierbath, itmightincreasethealuminaconcentration in thebath, decreasethe pot temperature and lead to pot instability by accumulation of \"sludge\" at thebottom of the pot (thesolidentifierbathcoolerthantheliquidentifierbathcanfalldirectly at thebottom of pot withoutbeingdissolved)."
    },
"identifier": "e9f3a32c-80a0-4549-addd-de159bc6355c",
"prefLabel": {
"en": "Anodecovering"
    }
},
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"narrower": "https://monsoon.ekf.tuke.sk/modeller/process-5e1025b0-87c9-4554-bc2f-3ed62e8153cc",
"broader": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-e9d5785d-8692-4076-9e7c-577721ce5711",
"previous": "",
"definition": {
"en": "Theanodes are linkedviatheanodebeam to the pot superstructure to allowthecurrentflowingthroughtheanode. Theanodes are consumedcontinuously by theelectrolysisreaction. As they are consumedtheanodeplan and thustheanodebeamgoesdownwardstillitreachesthelower limit of thebeam. Theanodebeammustthenberaised to theupper limit for a new cycle. Thisoperationiscalledanodebeamraising and is done in most caseswiththe Pot tendingmachine (cranes in theelectrolysishalls). Thisoperationismonitoredviatheprocesscontrolsystem (ALPSYS)."
    },
"identifier": "fad4c1bd-862b-43da-b142-76bd3d5a7c0f",
"prefLabel": {
"en": "Anodebeamraising"
    }
},
{
```

"uri": "https://monsoon.ekf.tuke.sk/modeller/process-5e1025b0-87c9-4554-bc2f-3ed62e8153cc",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"narrower": "",
"broader": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"next": "",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-0cccfbe1-6147-42f5-8f3f-f24dc6714f31",
"definition": {
"en": "Thisoperationcorrespondseither to the metal tapping or thebathtapping"
    },
"identifier": "5e1025b0-87c9-4554-bc2f-3ed62e8153cc",
"prefLabel": {
"en": "Tapping"
    }
},
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-0cccfbe1-6147-42f5-8f3f-f24dc6714f31",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "",
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"narrower": "https://monsoon.ekf.tuke.sk/modeller/process-aa74e9bd-36ce-40d0-a27f-d06a95474ac1",
"broader": "",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-5e1025b0-87c9-4554-bc2f-3ed62e8153cc",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-aa74e9bd-36ce-40d0-a27f-d06a95474ac1",
"definition": {
"en": "Theoperatorsmustcheckregularlyseveralpoints on pots (state of thehoods, state of theanode, state of themeasurementtools, CAFDholes, bath/crustsweeping,...). Thisinspectioniscalled Pot tending and is done manuallyfromtheshopfloor. Itismonitoredviatheprocesscontrolsystem (ALPSYS). Someadditionaloperations are linked to thisoverallcategory in theinterest of simplificationbutcanbeconsidentifierered as specificoperations (Tapping hole opening/closing, anoderecovering)."
    },
"identifier": "0cccfbe1-6147-42f5-8f3f-f24dc6714f31",
"prefLabel": {
"en": "Pot tending"
    }
},
{
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-aa74e9bd-36ce-40d0-a27f-d06a95474ac1",
"@context": "https://gbv.github.io/jskos/context.json",
"hasExecutionDependency": "https://monsoon.ekf.tuke.sk/modeller/process-0cccfbe1-6147-42f5-8f3f-f24dc6714f31",
"partOf": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b",
"narrower": "",
"broader": "https://monsoon.ekf.tuke.sk/modeller/process-0cccfbe1-6147-42f5-8f3f-f24dc6714f31",
"next": "https://monsoon.ekf.tuke.sk/modeller/process-0cccfbe1-6147-42f5-8f3f-f24dc6714f31",
"previous": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c",
"definition": {

"en": "Severalmeasurements are performedregularly (every32h), amongstthemtheliquidentifierheights, thesampling, raw materialdosemass, temperature, anodiccurrent drop... Thesevalues are saved in theprocesscontrolsystem (ALPSYS) fortheautomatic pot regulation."
    },
"identifier": "aa74e9bd-36ce-40d0-a27f-d06a95474ac1",
"prefLabel": {
"en": "Measurements"
    }
}

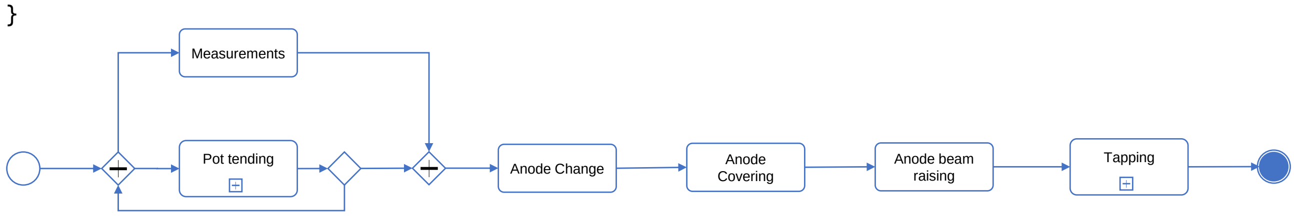


Figure 14 – Electrolysis sub-process model.

4.2 KPI models

Following sections contains the examples of formalizations of various KPIs related to the electrolysis scenario. Each KPI is provided with description and using *related*property linked with related process step (task or sub-process). One or more process steps can be provided. On the other hand, tags are provided as well for better searchability of the KPI in the MONSOON Semantic Modeller.

 {
"uri": "https://monsoon.ekf.tuke.sk/modeller/kpi-cjm7vve11000785mtw0lx8llu",
"@context": "https://gbv.github.io/jskos/context.json",
"identifier": "cjm7vve11000785mtw0lx8llu",
"prefLabel": {
"en": "Anode Effect Duration"
    },
"definition": {
"en": "The duration of the anode effect is directly linked to the amount of hydrocarbons produced. Less the anode effect lasts, less hydrocarbons are released into the atmosphere."
    },
"related": [
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b"
      }
    ],
"tags": []
  },

{
"uri": "https://monsoon.ekf.tuke.sk/modeller/kpi-cjikbenbl00067vpbnzxhix0y",
"@context": "https://gbv.github.io/jskos/context.json",
"identifier": "cjikbenbl00067vpbnzxhix0y",
"prefLabel": {
"en": "Anode effects"

```
    },
"definition": {
"en": "One of the consequences of the pot thermal balance instability is to increase the quantity of solidified bath at the top of the liquids and as a result close the opening made by the chisel in the crust, preventing the alumina to be delivered to the pot correctly. This underfeeding of alumina can create the anode effects that disturb the pot and its performances."
    },
"related": [
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-e9d5785d-8692-4076-9e7c-577721ce5711"
      },
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-0cccfbe1-6147-42f5-8f3f-f24dc6714f31"
      },
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-aa74e9bd-36ce-40d0-a27f-d06a95474ac1"
      },
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-e9f3a32c-80a0-4549-addd-de159bc6355c"
      },
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-0eeb4d02-905d-4285-a147-73ed53c47dab"
      }
    ],
"tags": [
      {
"prefLabel": "alumina underfeeding"
      },
      {
"prefLabel": "Pot instability"
      },
      {
"prefLabel": "CAFD hole opening"
      },
      {
"prefLabel": "Pot thermal balance"
      },
      {
"prefLabel": "bath height"
      }
    ]
  },
  {
"uri": "https://monsoon.ekf.tuke.sk/modeller/kpi-cjm7ukaog000285mt7jv56esd",
"@context": "https://gbv.github.io/jskos/context.json",
"identifier": "cjm7ukaog000285mt7jv56esd",
"prefLabel": {
"en": "Hazardous wastes"
```

```
    },
"definition": {
"en": "The electrolysis process can produce hazardous wastes at different phase of the pot
lifecycle (delining/relining, production...). Linked to the global KPI regarding the wastes."
    },
"related": [
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b"
      }
    ],
"tags": []
  },
  {
"uri": "https://monsoon.ekf.tuke.sk/modeller/kpi-cjik7wp8200017vpb86gs284o",
"@context": "https://gbv.github.io/jskos/context.json",
"identifier": "cjik7wp8200017vpb86gs284o",
"prefLabel": {
"en": "Liquid heights"
    },
"definition": {
"en": "Standard deviation of the liquids height - Gap of the bath+metal height between pots
belonging to the same working group (33 pots), every 8h or every 32h.\nIt is well known by
electrolysis process experts that the bath volume impacts the alumina dissolution. If the bath
height is too low the alumina dissolution will not be optimized that may lead to technical
performance decrease (current efficiency).\nIf the liquid heights are not well controlled, the
bath may completely submerse the anode and may attack the pins of the anode, leading to metal
pollution (with iron).\nThe metal height can influence the thermal balance (The metal volume
takes part in heat dissipation and the height variability can impact this phenomenon) and the
electric stability (orientation of the electromagnetic forces creating liquid
movements).\nFinally if the liquid heights decrease, it may provoke the covering bath to collapse
into the liquid bath and so modify the bath chemistry and the insulation properties."
    },
"related": [
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-fad4c1bd-862b-43da-b142-76bd3d5a7c0f"
      },
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-0cccfbe1-6147-42f5-8f3f-f24dc6714f31"
      },
      {
"uri": "https://monsoon.ekf.tuke.sk/modeller/process-532b452a-2b12-4644-9b47-2c38efbe230e"
      }
    ],
"tags": [
      {
"prefLabel": "tapping"
      },
      {
```

```
"prefLabel": "pot tending"
      },
      {
"prefLabel": "anode beam raising"
      }
    ]
  }
```

4.3 Data elements models

Following examples demonstrates the selected data elements from aluminium domain formalized using MONSOON semantic model. Data elements correspond to attributes ALF3C, ALF3D and AMALF3, from electrolysis scenario. Each data element has specified measurements units (*unit* property), its role (input or output) and are described using human-readable description and notation. Using *related* property, relation to other model elements are specified, e.g. which process step (task), equipment, or predictive function is related to the data element.

```
{
       "unit":"%",
       "role":"Input",
"tags":[],
"uri":"https://monsoon.ekf.tuke.sk/modeller/data-cjk433her000r85pg9dwp9kta",
"@context":"https://gbv.github.io/jskos/context.json",
"prefLabel":{
"en":"Target for the percentage of ALF3 in the pot"
      },
"notation":"ALF3C",
"definition":{
       "en":"Target for the percentage of ALF3 in the pot - Information given by the process
       control system (Shift variable)"
      },
"identifier":"cjk433her000r85pg9dwp9kta",
"related":[
"https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b"
      ]
},
{
"unit":"%",
"role":"Input",
"tags":[],
"uri":"https://monsoon.ekf.tuke.sk/modeller/data-cjk434zf7000s85pgsu6o6i35",
"@context":"https://gbv.github.io/jskos/context.json",
"prefLabel":{
       "en":"Latest percentage of AlF3 in the pot at the end of the shift"
               },
"notation":"ALF3D",
"definition":{
               "en":"Latest percentage of AlF3 in the pot at the end of the shift - Information
               given by the process control system"
      },
"identifier":"cjk434zf7000s85pgsu6o6i35",
"related":[
"https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b"
      ]
},
{
"unit":"kg",
"role":"Input",
"tags":[],
```

```
"uri":"https://monsoon.ekf.tuke.sk/modeller/data-cjk439zk6000t85pg8t101ds0",
"@context":"https://gbv.github.io/jskos/context.json",
"prefLabel":{
                "en":"ALF3 manual addition"
      },
"notation":"AMALF3",
"definition":{
                "en":"ALF3 manual addition - Information given the process control system (Shift
                variable)"
      },
"identifier":"cjk439zk6000t85pg8t101ds0",
"related":[
"https://monsoon.ekf.tuke.sk/modeller/process-344e039c-3209-4818-99e0-eb985ad5ff7b"
      ]
},
```


4.4 Predictive function models

Followingsectionpresentstheformalization of predictivefunctions. FollowingexamplesdescribetheRandomForestclassification model. Thisparticular model is a classification model, of UnivariateDecisionTree model type. Itisan output of a RandomForestsalgorithmwhichisunivariatedecisiontreealgorithmspecified by a set of differentsettings, e.g. maximum depth of trees in forest, minimalnumber of records in a split or leaves, and has severalqualityattributese.g. isable to handlecontinuousfeatures, or performmulticlassclassification.


```
<!-- http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#RandomForestModel -->
<owl:NamedIndividualrdf:about="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#RandomForest
Model">
<rdf:typerdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#RandomTreeModel"/>
<DMOP:hasHypothesisStructurerdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMOP.owl
#UnivariateDecisionTree"/>
<DMOP:hasModelComplexityMeasurerdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/
DMOP.owl#NumberOfLeaves"/>
<DMOP:hasParameterrdf:resource="http://www.e-
lico.eu/ontologies/dmo/DMOP/DMKB.owl#NumberOfTrees"/>
</owl:NamedIndividual>

<!-- http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#RandomForestsAlgorithm -->
<owl:NamedIndividualrdf:about="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#RandomForests
Algorithm">
<rdf:typerdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#
UnivariateTreeInductionAlgorithm"/>
<DOLCE-Lite:has-qualityrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/
DMOP.owl#HandlesContinuousFeatures"/>
<DOLCE-Lite:has-qualityrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/
DMOP.owl#HandlesMulticlassClassification"/>
<DOLCE-Lite:has-qualityrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/
DMOP.owl#HighVarianceProfile"/>
<DMOP:hasFeatureTestEvaluatorrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/
DMOP.owl#GiniIndex"/>
<DMOP:hasParameterrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl #Max_Depth"/>
```

```
<DMOP:hasParameterrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl
#Min_samples_leaf"/>
<DMOP:hasParameterrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl
#Min_samples_split"/>
<DMOP:hasParameterrdf:resource="http://www.e-
lico.eu/ontologies/dmo/DMOP/DMKB.owl#Number_of_estimators"/>
<DMOP:hasParameterrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMOP.owl
#InformationGainRatio"/>
<DMOP:specifiesOutputClassrdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl #
RandomTreeModel"/>
</owl:NamedIndividual>
```

The algorithm was applied using concrete values of the algorithm parameters, which are specified as individuals of the Parameter class. In this case, *Min_samples_leaf* parameter instance is set to a integer value of 1, *Min_samples_split* parameter set to value 5 and *Max_depth* set to none (which specify, that the parameter was not considered in model training).

```
<!-- http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#Min_samples_leaf -->
<owl:NamedIndividualrdf:about="http://www.e-
lico.eu/ontologies/dmo/DMOP/DMKB.owl#Min_samples_leaf">

<rdf:typerdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMOP.owl#AlgorithmParameter"/>
<DMOP:hasDataValuerdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</DMOP:hasD
ataValue>
</owl:NamedIndividual>

<!-- http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#Min_samples_split -->
<owl:NamedIndividualrdf:about="http://www.e-
lico.eu/ontologies/dmo/DMOP/DMKB.owl#Min_samples_split">
<rdf:typerdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMOP.owl#AlgorithmParameter"/>
<DMOP:hasDataTyperdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMOP.owl#Integer"/>
<DMOP:hasDataValuerdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">5</DMOP:hasD
ataValue>
</owl:NamedIndividual>

<!-- http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#Max_Depth -->
<owl:NamedIndividualrdf:about="http://www.e-lico.eu/ontologies/dmo/DMOP/DMKB.owl#Max_Depth">
<rdf:typerdf:resource="http://www.e-lico.eu/ontologies/dmo/DMOP/DMOP.owl#AlgorithmParameter"/>
<DMOP:hasDataValuerdf:datatype="http://www.w3.org/2001/XMLSchema#string">None</DMOP:hasDataValue>
</owl:NamedIndividual>
```

5 Conclusions

This deliverable presented the final version of the Cross-sectorial domain model specification. The main section of the deliverableis dedicated to specification of the extensions of the initial Cross sectorial domain model as it was presented in deliverable D2.7. This document then describes in more detail the concepts for different model parts e.g. domain concepts or concepts used for predictive functions description. Chapter 4 then presents the machine-readable form of domain model for selected MONSOON application domain scenario (Electrolysis use case). The example demonstrates usage of MONSOON domain model for process, data elements, KPI and predictive function description from the selected domain. Human-readable process models were created using the MONSOON Semantic Modeller.

Acronyms

Acronym	Explanation
B2MML	Business To ManufacturingMarkupLanguage
BPMN	Business Process Modeling Notation
CRISP-DM	Cross-industry Standard Process for Data Mining
ERP	Enterprise Resource Planning
DMOP	Data Mining OPtimization
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
KOS	Knowledge organization system
OWL	Web Ontolodgy Language
PFA	Portable Format for Analytics
PMML	Predictive Model Markup Language
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
UML	Unified Modeling Language
URI	Unified Resource Idenifier
W3C	World Wide Web Consortium
XML	Extensible Markup Language

List of figures

List of tables

References

[1] http://www.bpmn.org

[2] L.S. Gould, B2MML Explained., Automotive Design & Production. 119 (2007)

[3] C.M. Keet, A. Ławrynowicz, C. D'Amato, A. Kalousis, P. Nguyen, R. Palma, R. Stevens, M. Hilario, The Data Mining OPtimization Ontology, Journal of Web Semantics. 32 (2015) 43–53. doi:10.1016/j.websem.2015.01.001

[4] P. Panov, L. Soldatova, S. Džeroski, OntoDM-KDD: Ontology for representing the knowledge discovery process, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013: pp. 126–140. doi:10.1007/978-3-642-40897-7_9.

[5] J. Vanschoren, L.N. Soldatova, Exposé: An ontology for data mining experiments, Proc. of the 3rd Int. Workshop on Third Generation Data Mining: Towards Service-Oriented Knowledge Discovery (SoKD). (2010) 31–46.

[6] MONSOON Deliverable D4.1 – Initial Semantic Frameworkfor dynamic multi-scale industry modelling..