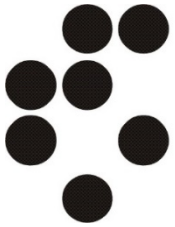


Institut
"Jožef Stefan"
Ljubljana, Slovenija

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

Ontology of Data Mining

Panče Panov

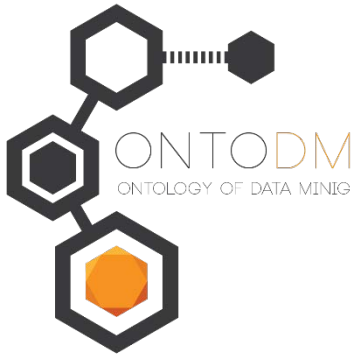
Jožef Stefan Institute, Department of Knowledge Technologies

Ljubljana, Slovenia

Joint work with Larisa Soldatova and Sašo Džeroski

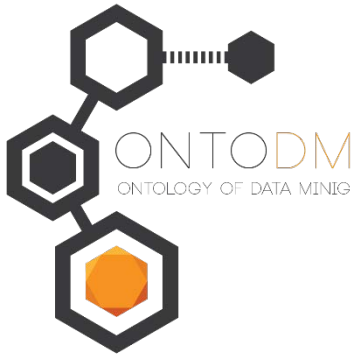
Available @ http://www.ontodm.com

MAESTRA Summer School on Mining Big and Complex Data,
Ohrid, Macedonia, 5 SEP 2016



Outline

- Ontologies in a nutshell
- The domain of data mining
- Ontologies for data mining
  - Ontology of datatypes
  - Ontology of core data mining entities
- Use cases
- Conclusions



What is an ontology?

- Ontology - the “science of being”
- Different meanings in different contexts
  - Philosophical -- categorical analysis
    - ``What are the entities in the world?", ``What are the categories of entities?“
  - Computer science -- creation of engineering models of the reality
    - used by software, directly interpreted and reasoned over by inference engines
    - In AI: “Specification of a conceptualization”
- Our context: Ontologies as data and knowledge models
  - logically defined, flexible and interoperable representations of principal domain entities

Basic ontology vocabulary – The pizza domain

CLASSES

RELATIONS

is-a

Pizza and (hasTopping some CheeseTopping)

Pizza and (hasTopping some MeatTopping)

Pizza and (not (hasTopping some FishTopping)) and (not (hasTopping some MeatTopping))

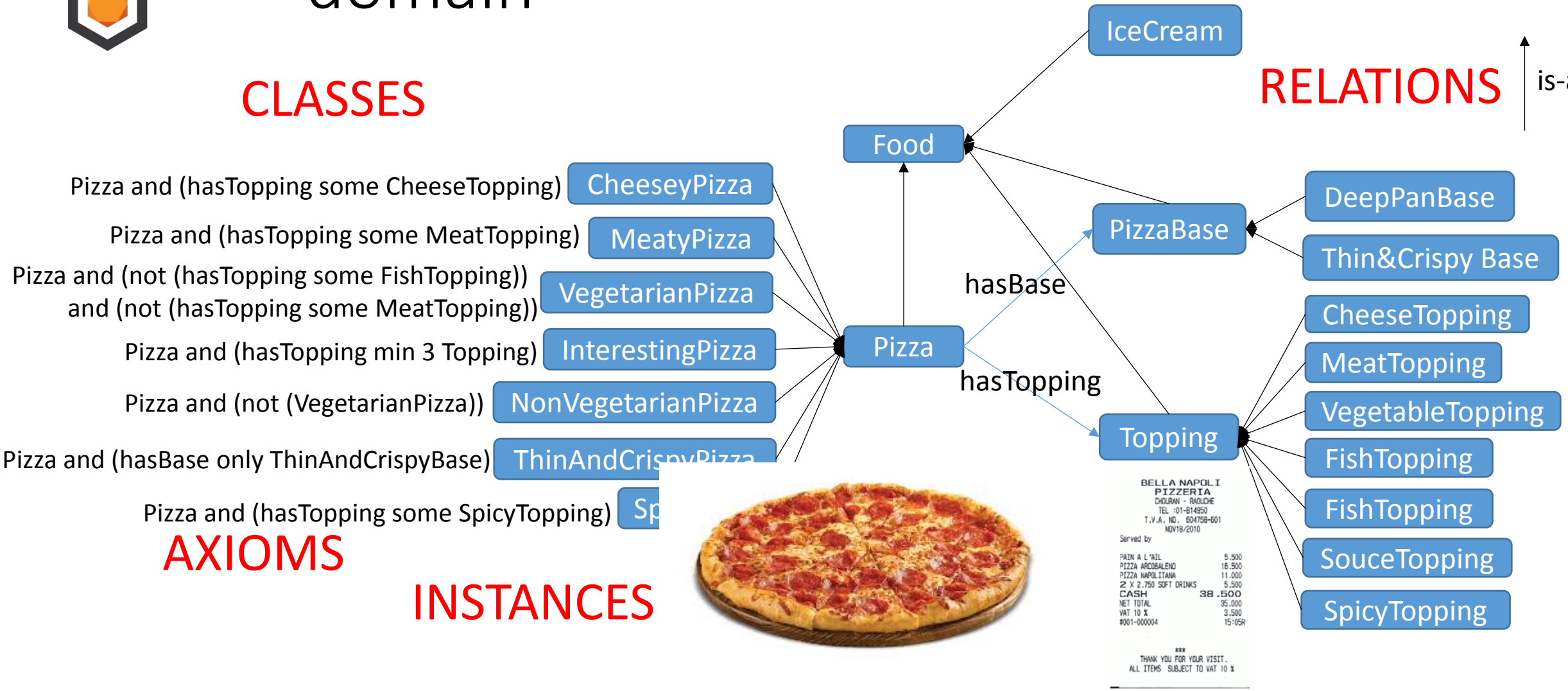
Pizza and (hasTopping min 3 Topping)

Pizza and (not (VegetarianPizza))

Pizza and (hasBase only ThinAndCrispyBase)

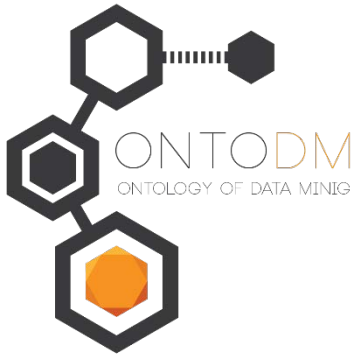
Pizza and (hasTopping some SpicyTopping)

AXIOMS



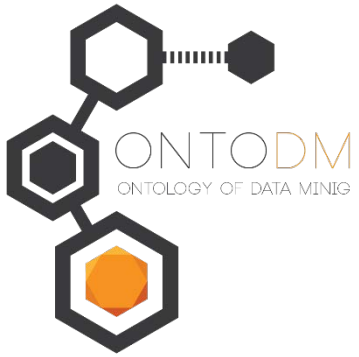
INSTANCES

Ontology source: http://protege.stanford.edu/ontologies/pizza/pizza.owl



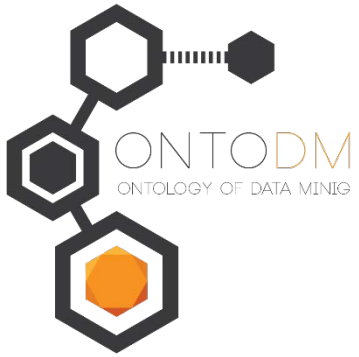
Why are ontologies useful?

- Formalize core domain entities
  - using some logical representation (e.g., Description Logic)
- Specify controlled vocabularies
  - Integration of heterogeneous data
  - Annotation of experimental data
  - Assist to information retrieval
  - Assist to literature mining
- Provide interoperability services
  - Exchange of data among different systems
- ….



How do we build ontologies?

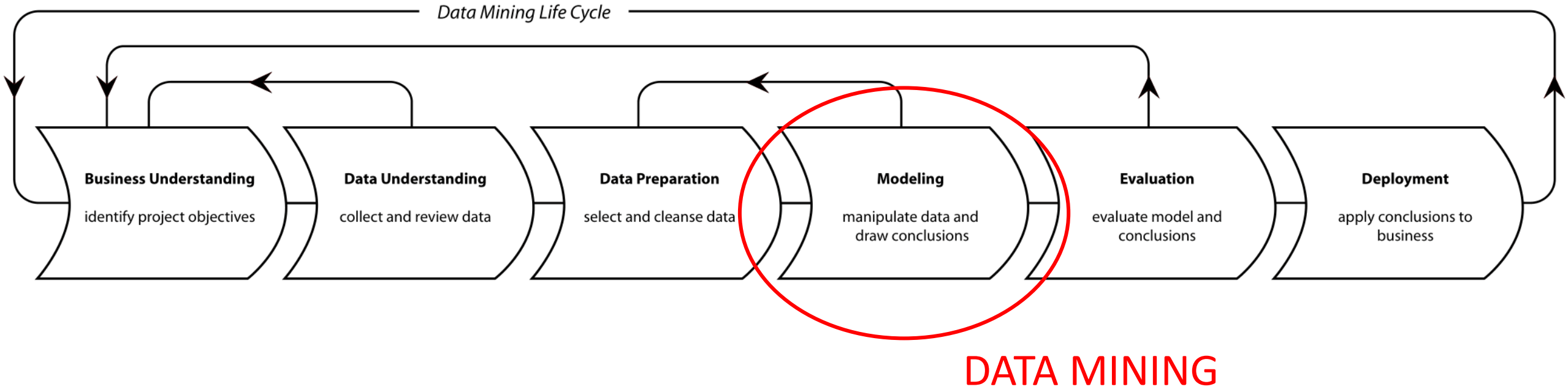
- No universally accepted design principles
- Best practices: Open Biomedical Ontologies (OBO) foundry
  - State-of-the-Art in the biomedical domains
  - 19 recommendations for building ontologies
    - Upper-level ontology as a guidance prototype
    - Standardized relations
    - Avoidance of multiple inheritance
    - Development of orthogonal resources
    - …
- Ontology engineering methodologies:
  - TOVE, METHONTOLOGY, On-To-Knowledge, NeOn ...
- Use of semantic languages, query languages and tools
  - Resource Description Framework (RDF), RDF Schema, Ontology Web Language (OWL)
  - SPARQL and SPARQL-DL query languages
  - Integrated ontology engineering environment - Protégé



Why do we need an ontology of data mining?

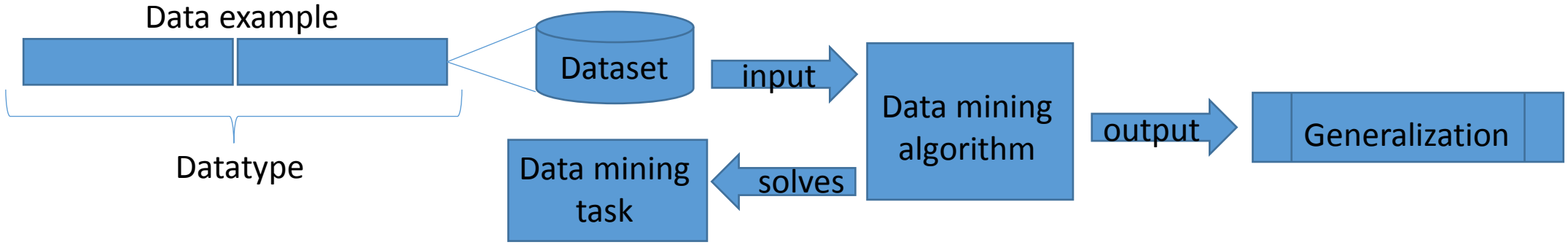
- Help us understand in more depth how things in DM function
- Annotation and querying
  - Machine learning dataset repositories
  - Repositories of data mining algorithms
  - Machine learning experiments
  - Data mining papers
- Automatic workflow construction
- Describe data mining scenarios in domain applications

The Big Picture: The process of knowledge discovery

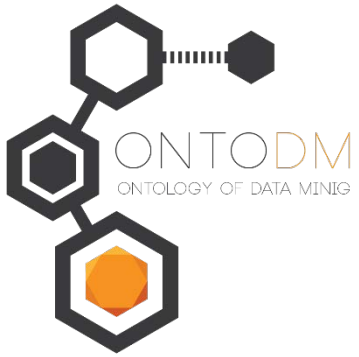


Source: The **CRISP-DM** user guide

The domain of data mining



- Data mining deals with analyzing different types of data
- The data is organized in a form of a dataset
  - Composed of data examples
  - The structure is described by datatype
- The task of data mining is to produce some type of a output from a given dataset
  - We call this output a generalization
  - Predictive model, set of patterns, probability distribution, clustering ...
- A data mining task is solved by using a data mining algorithm
  - Algorithms are implemented as a computer program
  - When executed they take as input a dataset and give as output a generalization

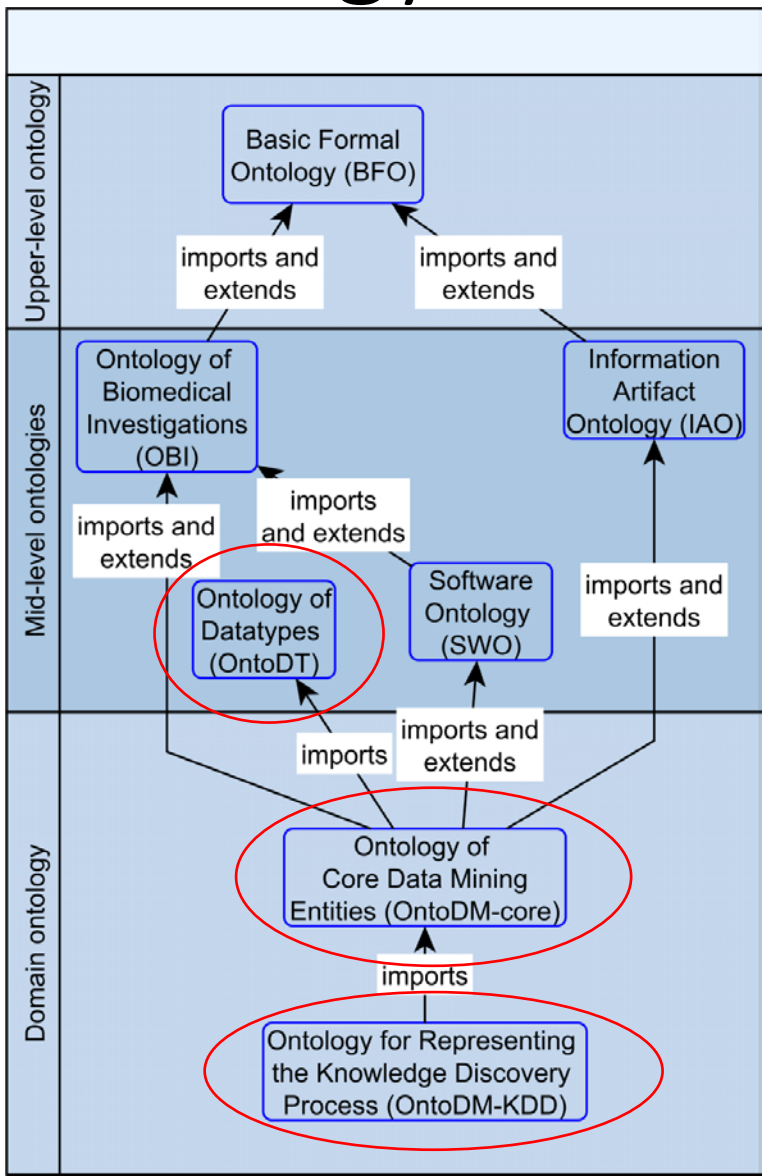


What is our task?

- To formally represent the process of data mining
  - Datatypes, data examples, datasets
  - Data mining tasks
  - Data mining algorithms
  - Models, patterns, sets of clusters, probability distributions (we call these generalizations)
- To formally represent the knowledge discovery (KD) process
  - Phases of the KD process
  - Inputs and outputs
- Express it in a semantic language
  - OWL language based on description logic (DL)

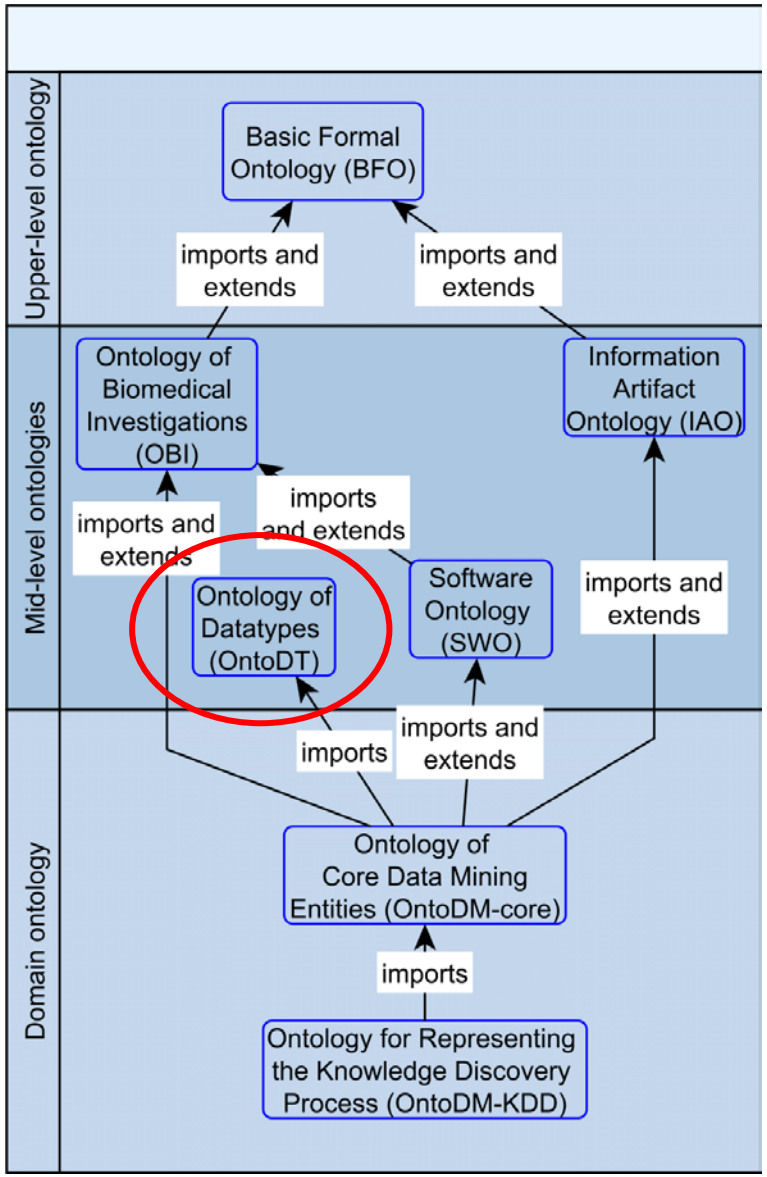
The OntoDM data mining ontology

- Built using best practices from biomedical domains
  - the OBO Foundry principles [Smith et al., 2007]
- Complementary to and integrated with state-of-the-art ontologies for representing scientific knowledge
  - Interoperability with other resources
  - Allows for cross-domain reasoning
- Use of upper level ontology
  - Basic Formal Ontology (BFO) as a template
  - Small set of formally defined relations
- Reuse of classes from other ontologies
  - Ontology of Biomedical Investigations (OBI)
  - Information Artifact Ontology (IAO)
  - Software Ontology (SWO)
- Modular ontology
  - Three ontology modules (OntoDT, OntoDM-core, OntoDM-KDD)
  - Modules can be used together and independently
- Today we focus on OntoDT and OntoDM-core



Ontology modules

- **OntoDT – Ontology of datatypes**
- Representation of scientific knowledge about datatypes
- Available @ http://www.ontodt.com



ELSEVIER

Information Sciences

Volume 329, 1 February 2016, Pages 900–920

Special issue on Discovery Science

Generic ontology of datatypes

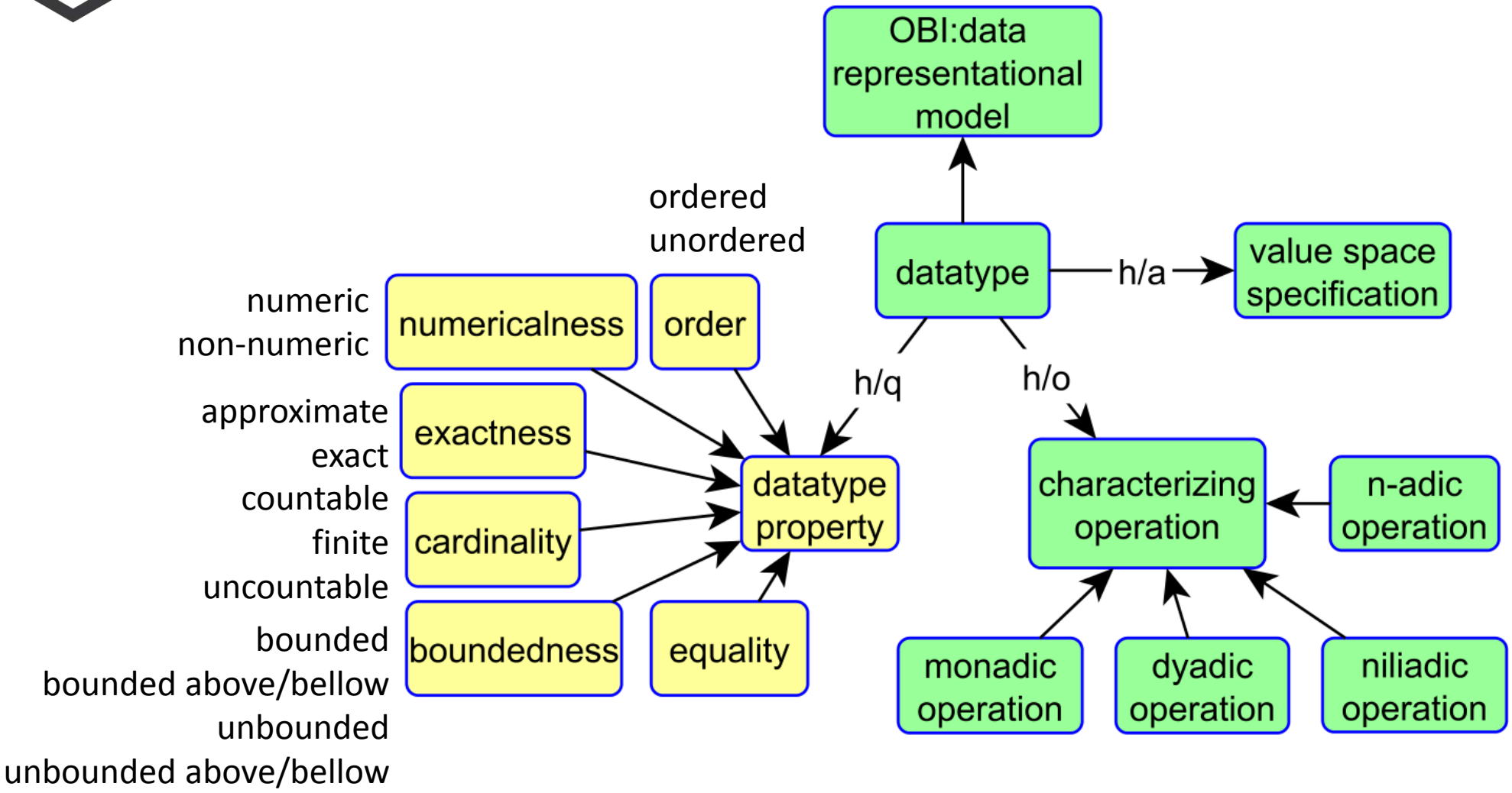
Panče Panov [a], Larisa N. Soldatova [d], Sašo Džeroski [a, b, c]

Ontology of datatypes – OntoDT

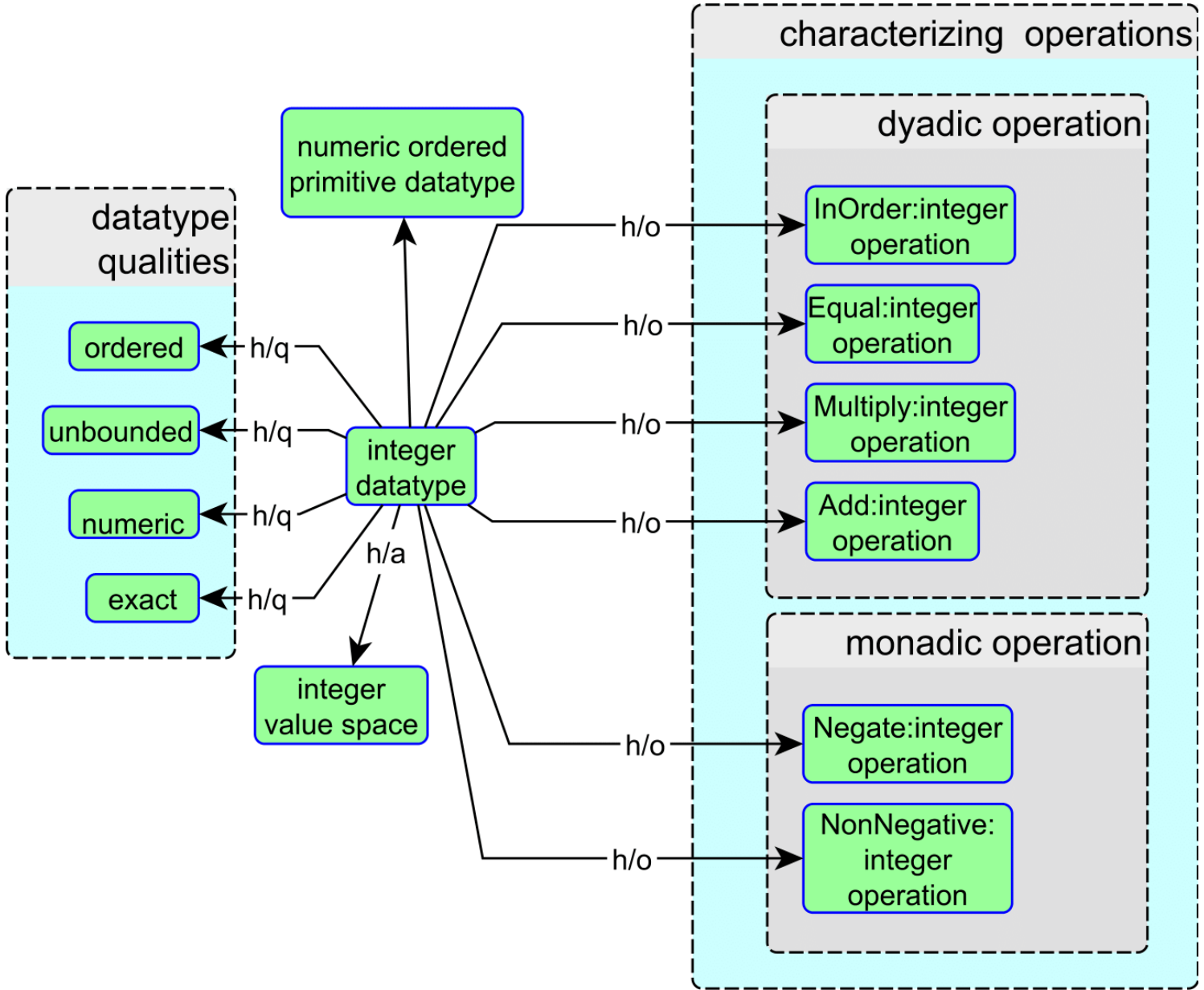
- Mid-level ontology
- Based on International Standard for Datatypes in Computer Systems ISO/IEC 11404
  - Terminology and semantics for a collection of data types
  - Programming languages and software interfaces
  - The datatypes defined in the standard are general in
- The generic nature enables support to a wide range of other applications
- **The notion of a datatype is very important in data mining**
  - Characterize the types of data contained in a dataset
  - Applicability of a data mining task on data from a given datatype
  - Applicability of a data mining algorithm on a dataset

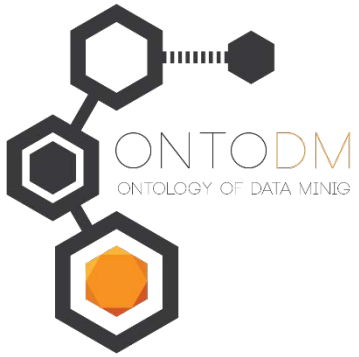
P. Panov, L. N. Soldatova, S. Džeroski (2016) “Generic ontology of datatypes",
Information sciences 329:900-920
doi: 10.1016/j.ins.2015.08.006

The basic structure of OntoDT



Example of a datatype class



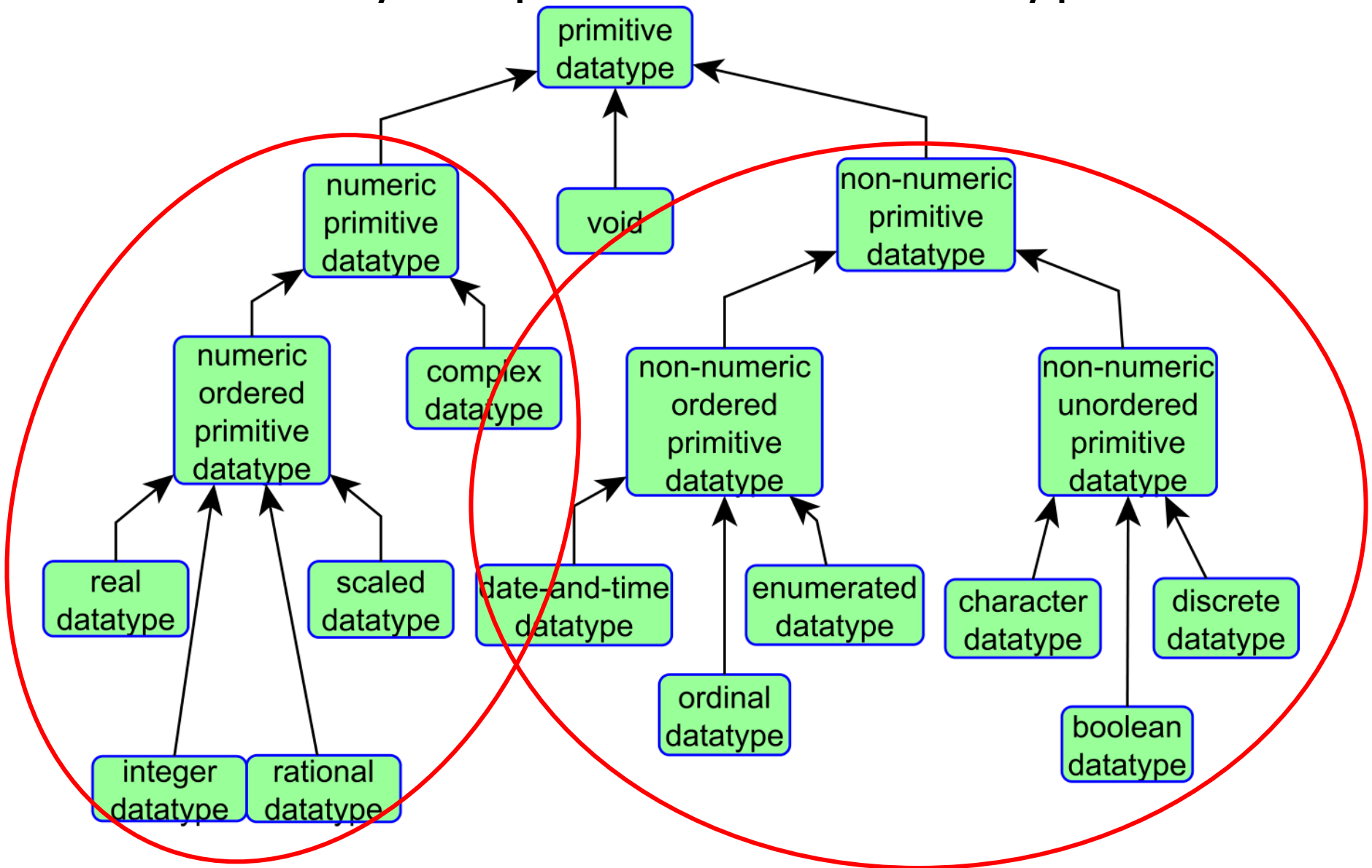


Taxonomy of datatypes

- Primitive datatypes
  - defined by explicit specification and are independent of other datatypes
  - E.g., Real, Integer, Ordinal, Enumerated, Discrete, Boolean
- Generated datatypes
  - syntactically and semantically dependent of other datatypes and are specified implicitly with datatype generators.
  - E.g., bag, set, sequence, array, tuple
- User defined datatypes
  - defined by a datatype declaration
  - allow defining additional identifiers and refinements to both primitive and generated datatypes.
  - E.g., tree, graph

ONTODM
ONTOLOGY OF DATA MINIG

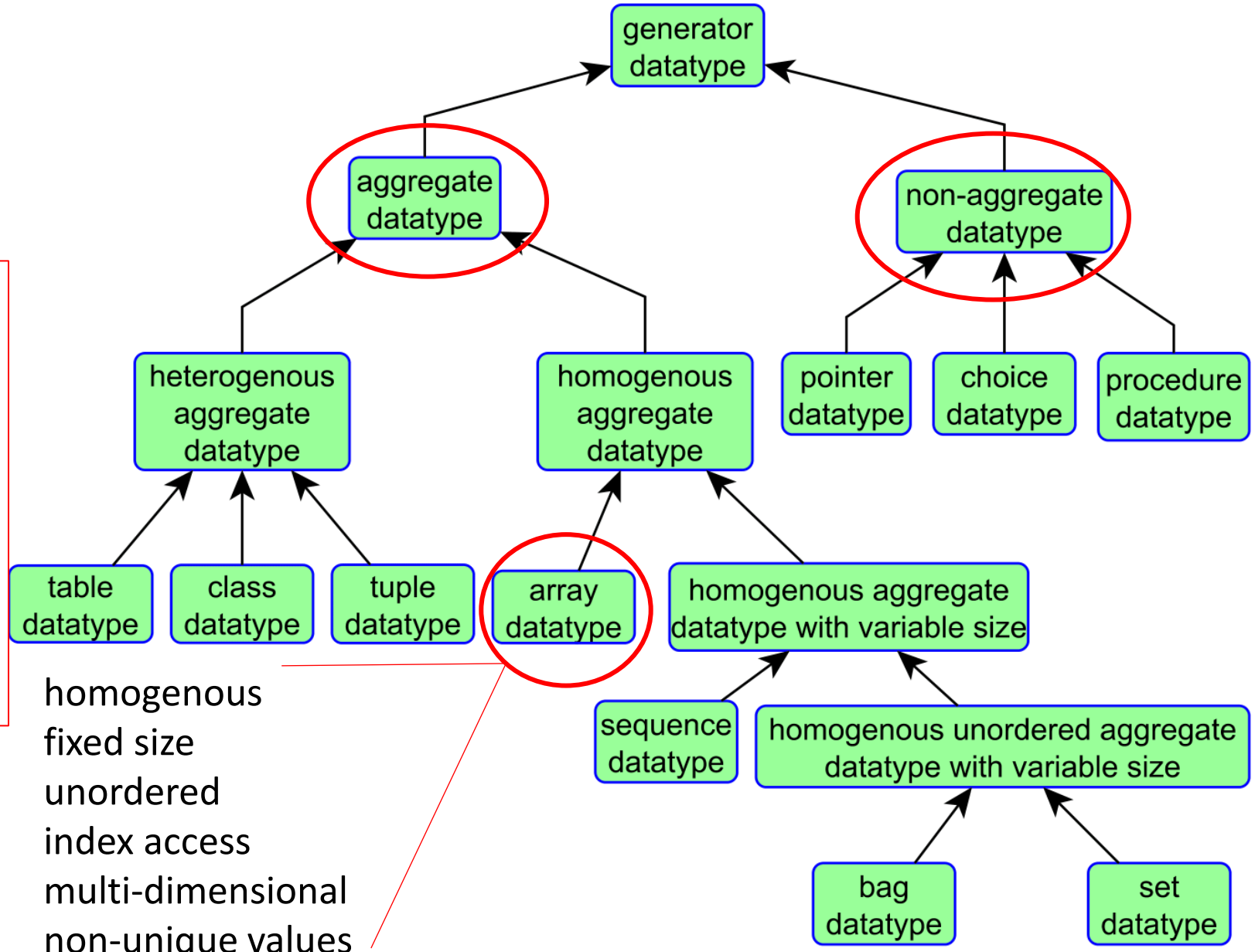
Taxonomy of primitive datatypes



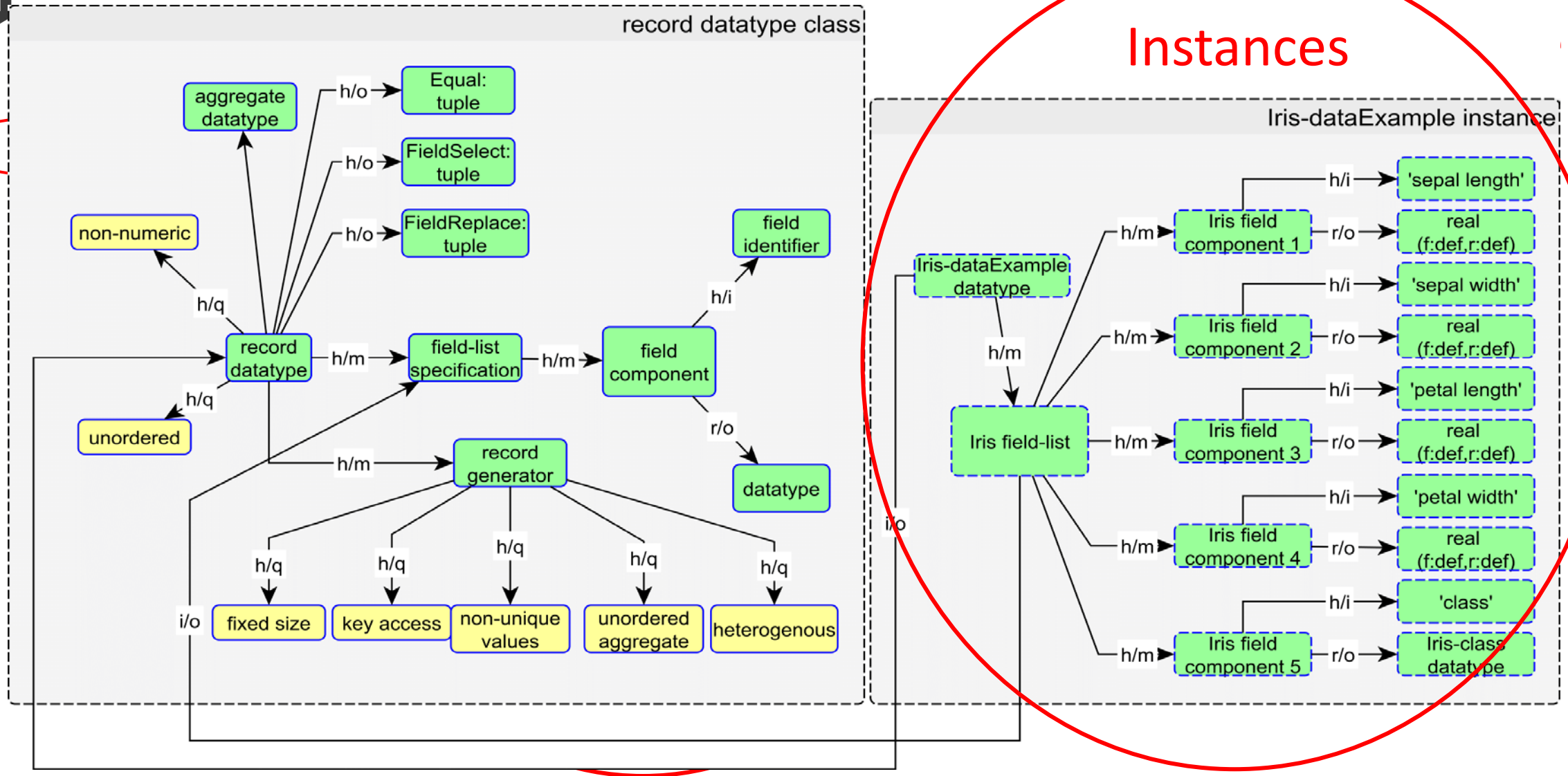
Taxonomy of generated datatypes

Aggregate properties

access type
aggregate-imposed ordering
aggregate-imposed identifier uniqueness
aggregate size
component mandatoriness
homogeneity
recursiveness
uniqueness
structuredness

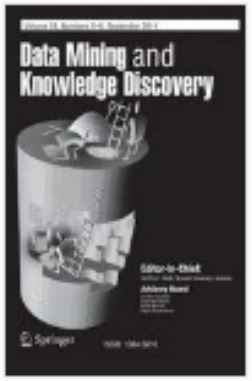


Example: Datatypes describing the Iris dataset



Ontology modules

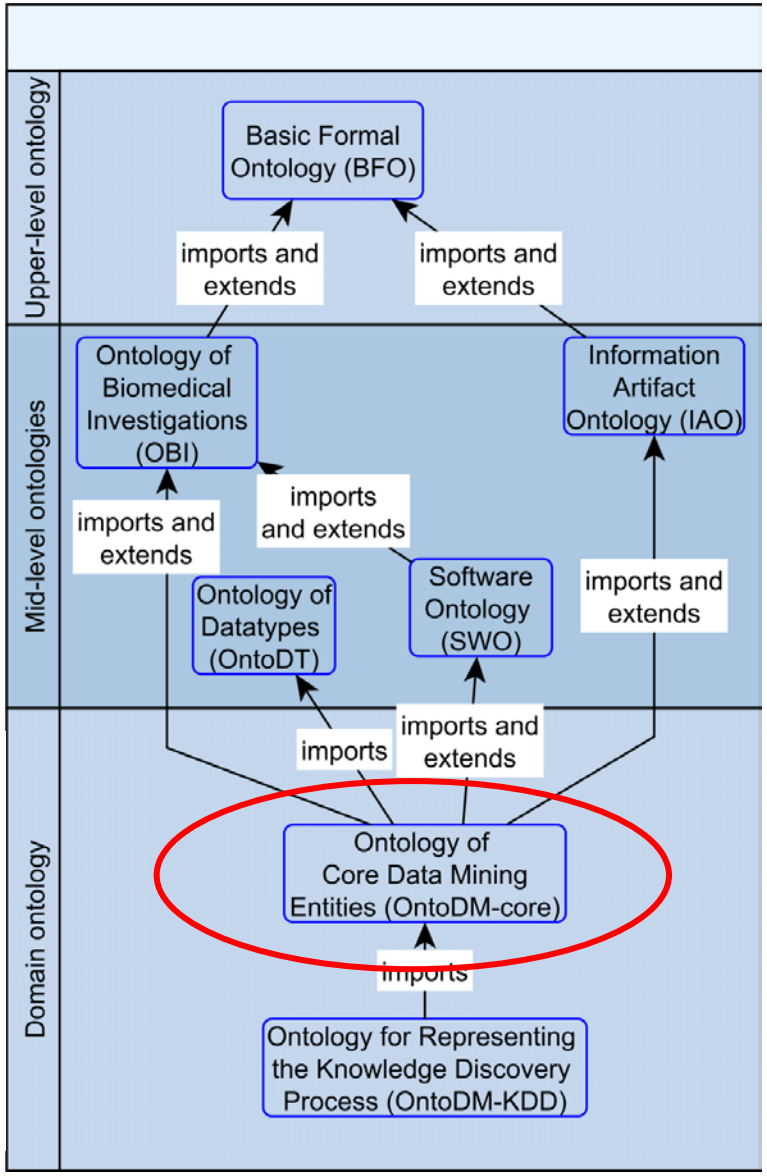
- **OntoDM-core – Ontology of core data mining entities**
- Representation of core data mining entities
- Available @ http://www.ontodm.com

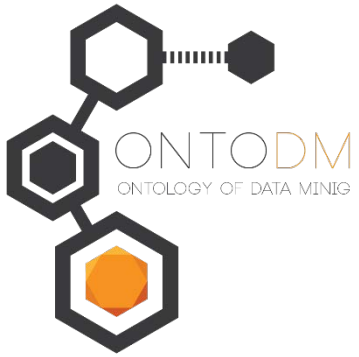


Data Mining and Knowledge Discovery
September 2014, Volume 28, Issue 5, pp 1222–1265

Ontology of core data mining entities

Panče Panov, Larisa Soldatova, Sašo Džeroski



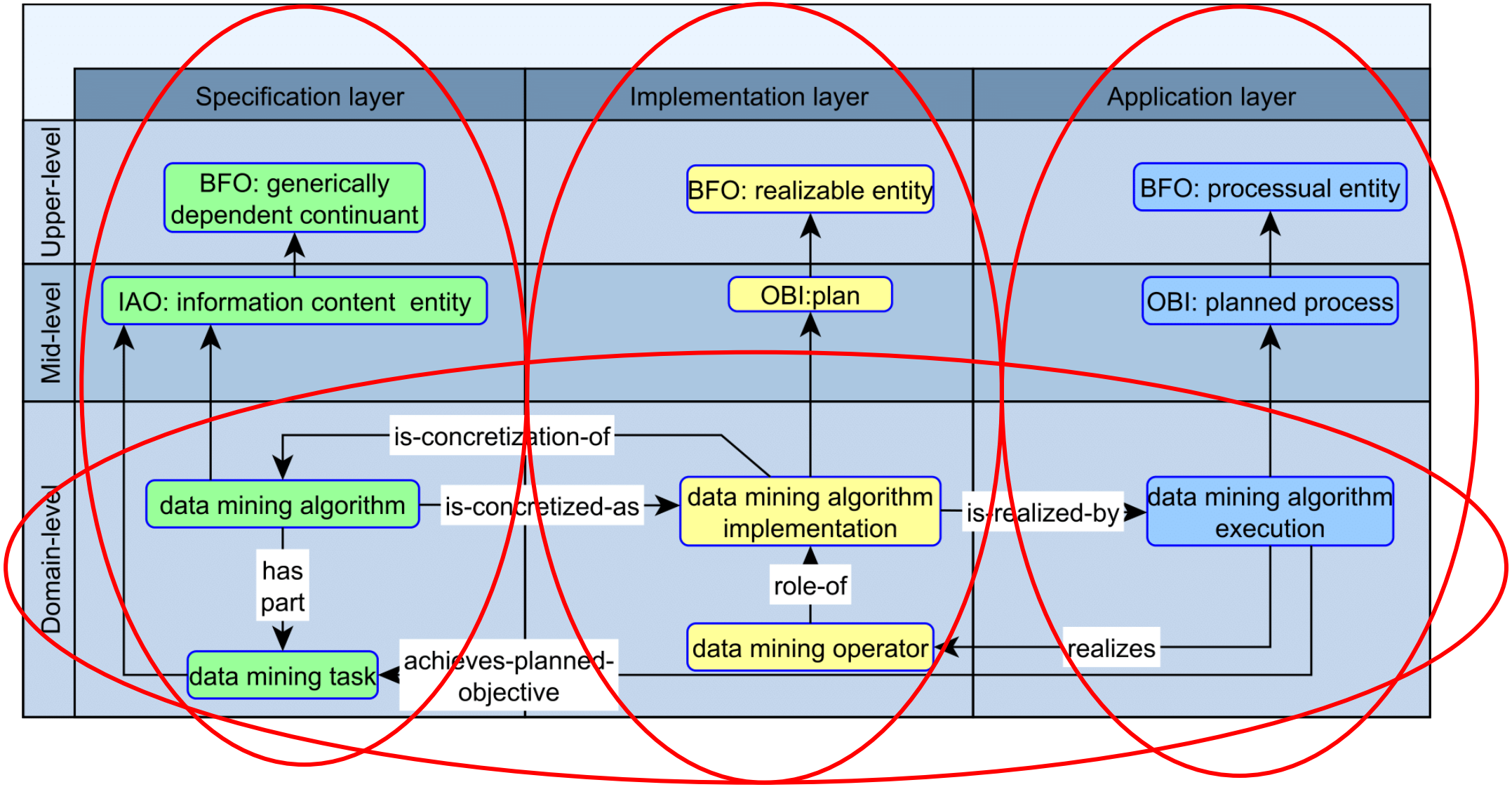


Ontology of core data mining entities - OntoDM-core

- Based on a proposal for a general framework for data mining (Džeroski, 2007)
- OntoDM-core describes the most essential data mining entities
  - Data specification
  - Dataset
  - Data mining task
  - Data mining algorithm
  - Generalizations (patterns, models)
- Taxonomies of datasets, data mining tasks, generalizations, data mining algorithms based on the type of data.
- Representational framework for description of mining of structured data

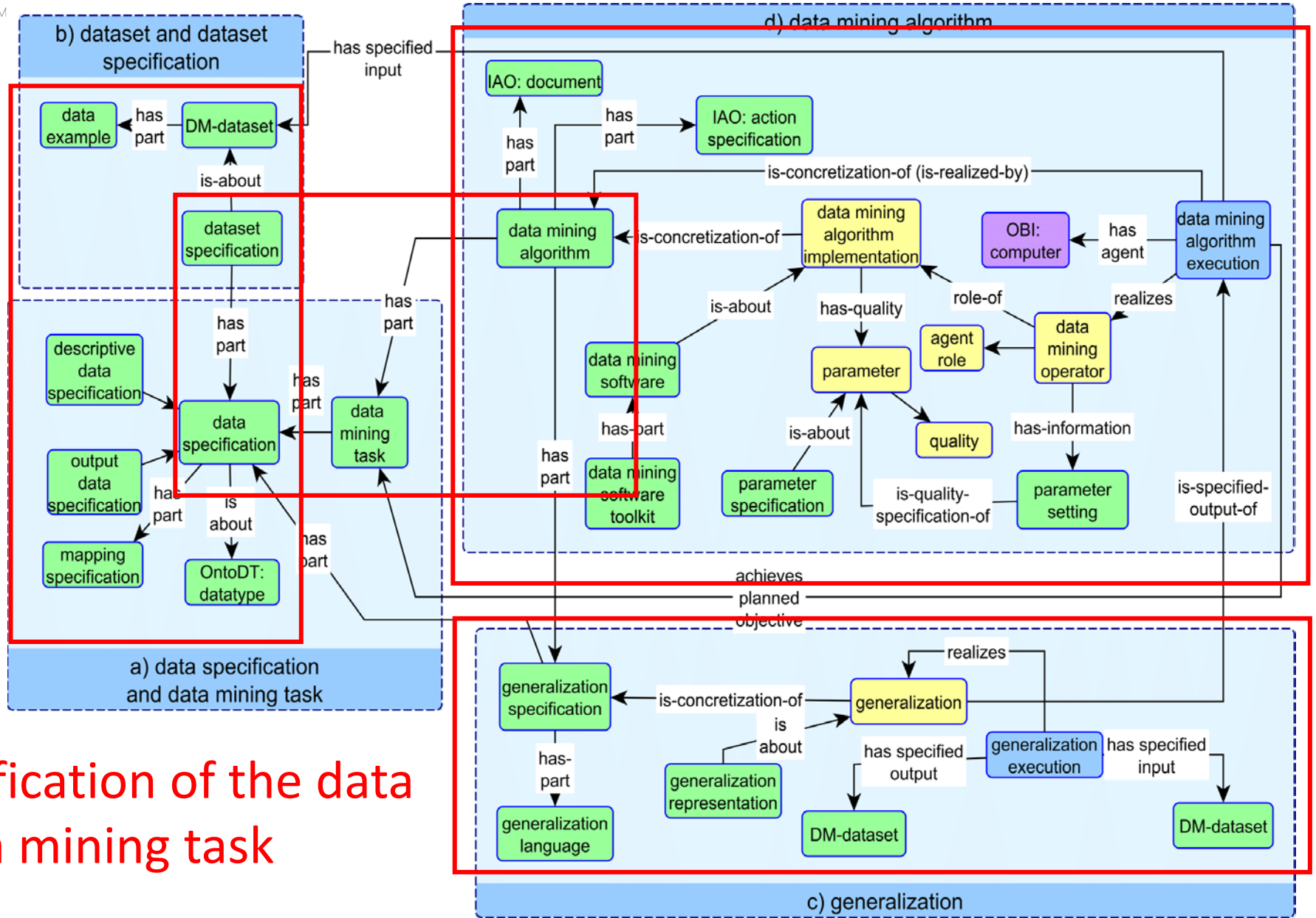
P. Panov, L. N. Soldatova, S. Džeroski (2014) "Ontology of core data mining entities", Data mining and knowledge discovery 28(5-6):1222-1265

Design structure



Core data mining entities

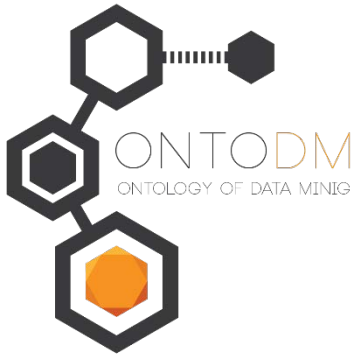
Data mining algorithm



Outputs

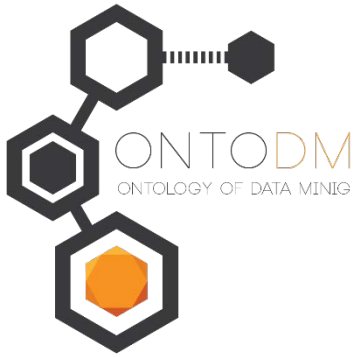
Specification of the data

Data mining task



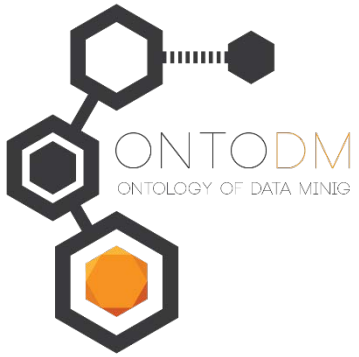
Data specification

- One of the most important representational aspects
  - Data specification describes the datatype of the data examples
  - Datatypes are further specified with the OntoDT ontology
- Defines other entities
  - dataset specification, data mining task, generalization specification
- Two types of specifications
  - Descriptive data specification
    - data used for descriptive purposes (e.g., attributes or features)
  - Output data specification
    - data used for predictive purposes (e.g., classes/targets)



Examples of different types of data

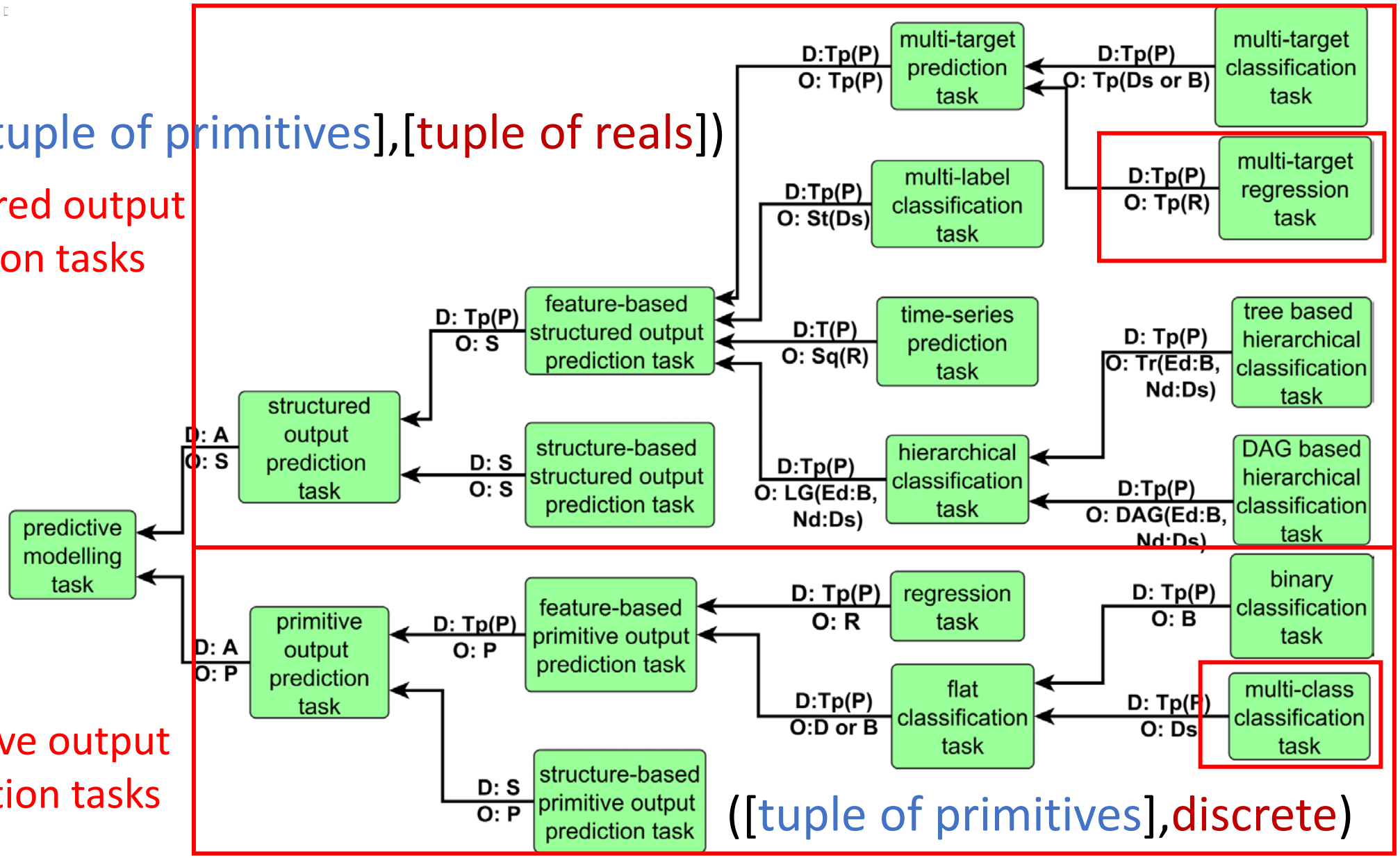
- Unlabeled data - only descriptive part
  - Feature-based data example ([tuple of primitives])
  - Transactional data example ({set of discrete})
- Labeled data - both descriptive and output parts
  - Feature-based data example with primitive output
    - ([tuple of primitives],real)
    - ([tuple of primitives],boolean)
    - ([tuple of primitives],discrete)
  - Feature-based data example with structured output
    - ([tuple of primitives],[tuple of reals])
    - ([tuple of primitives],[tuple of discrete])
    - ([tuple of primitives],{set of discrete})
    - ([tuple of primitives],(sequence of real))
    - ([tuple of primitives],tree with boolean edges and discrete nodes)
    - ([tuple of primitives],DAG with boolean edges and discrete nodes)

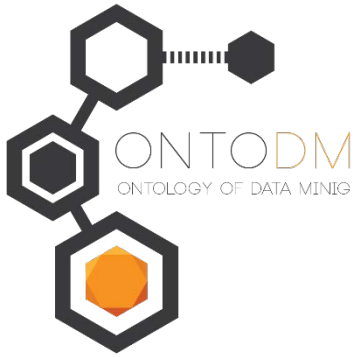


Data mining task

- The task of data mining is to produce a generealization from given data
  - Patterns, models, clusterings
- Data mining task depends directly on the data specification
  - Taxonomy of data mining tasks based on the data specifications
- Four top level data mining tasks
  - Clustering - defined for unlabeled data
  - Pattern discovery - defined for unlabeled data
  - Probability distribution estimation - defined for unlabeled data
  - Predictive modeling - defined for labeled data
- Predictive modeling is represented in more detail

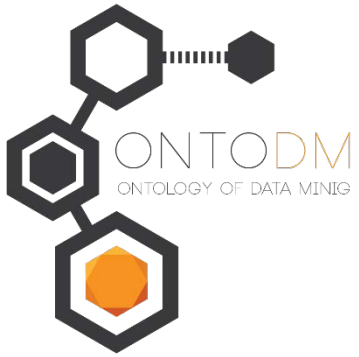
Predictive modeling task taxonomy





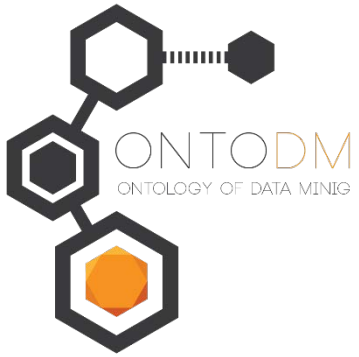
Use case: Annotating and querying data mining algorithm repositories

- We annotated a software system that contains algorithms for structured output prediction using OntoDM-core and OntoDT
  - The Clus software – Tree and rule learning system based on predictive clustering (https://dtai.cs.kuleuven.be/clus/)
  - Gives as output generalizations predictive clustering trees (PCTs) and rules (PCRs)
  - Single and ensemble algorithms for primitive and structured output prediction tasks
    - Classification, regression, multi-target prediction, hierarchical classification, multi-label classification, time-series prediction
- We populated the ontology with Clus instances:
  - 29 DM task instances, 22 generalization specification instances, 48 DM algorithm instances, 40 classes of datasets and 79 instances of datasets, 2 language specification instances



Types of queries and reasoning

- Query types
  1. Queries that concern only classes (TBox queries)
     Find all subclasses of predictive modeling single generalization algorithm that has as part structured output prediction task.
  2. Queries that concern only instances(ABox queries)
     Find all data mining algorithms that can be applied on Yeast dataset having as a result a generalizion that is expressed in the language of PCRs.
  3. Mixed queries (ABox/TBox queries)
     Find all datasets to which the bagging of multi-target classification PCTs algorithm can be applied.
- Reasoning
  - HermiT 1.3.8 reasoner
  - asserted+inferred ontology was queried
  - SPARQL and SPARQL-DL query languages



Example

Find all algorithms that solve a structured output prediction task, produce a generalization expressed in the language of PCTs as output, and are applicable to the EDM dataset

```
SELECT DISTINCT ?dataMiningAlgorithm
WHERE { ?dataMiningAlgorithm  RO:has_part  ?dataMiningTask .
  ?datasetSpecification RO:has_part  ?outputDataSpecification .
  ?generalizationSpecification RO:has_part  OntoDM-clus:OntoDM_clus_00106 .
  ?datasetSpecification RO:has_part  ?descriptiveDataSpecification .
  ?dataMiningTask RO:has_part  ?outputDataSpecification .
  ?generalizationSpecification RO:has_part  ?descriptiveDataSpecification .
  ?dataMiningAlgorithm RO:has_part  ?generalizationSpecification .
  ?generalizationSpecification RO:has_part  ?outputDataSpecification .
  ?dataMiningTask  RO:has_part  ?descriptiveDataSpecification .
  ?datasetSpecification OBO:IAO_0000136  ?datasetInstance .
  ?datasetInstance  rdf:type  OntoDM-clus:OntoDM_clus_00266 .
  ?outputDataSpecification rdf:type  OntoDM-core:OntoDM_000027 .
  ?datasetSpecification rdf:type  OntoDM-core:OntoDM_000031 .
  ?descriptiveDataSpecification rdf:type  OntoDM-core:OntoDM_000247 .
  ?dataMiningTask rdf:type  OntoDM-core:OntoDM_600958 .
  ?dataMiningAlgorithm rdf:type  OntoDM-core:OntoDM_000038 .
}
```

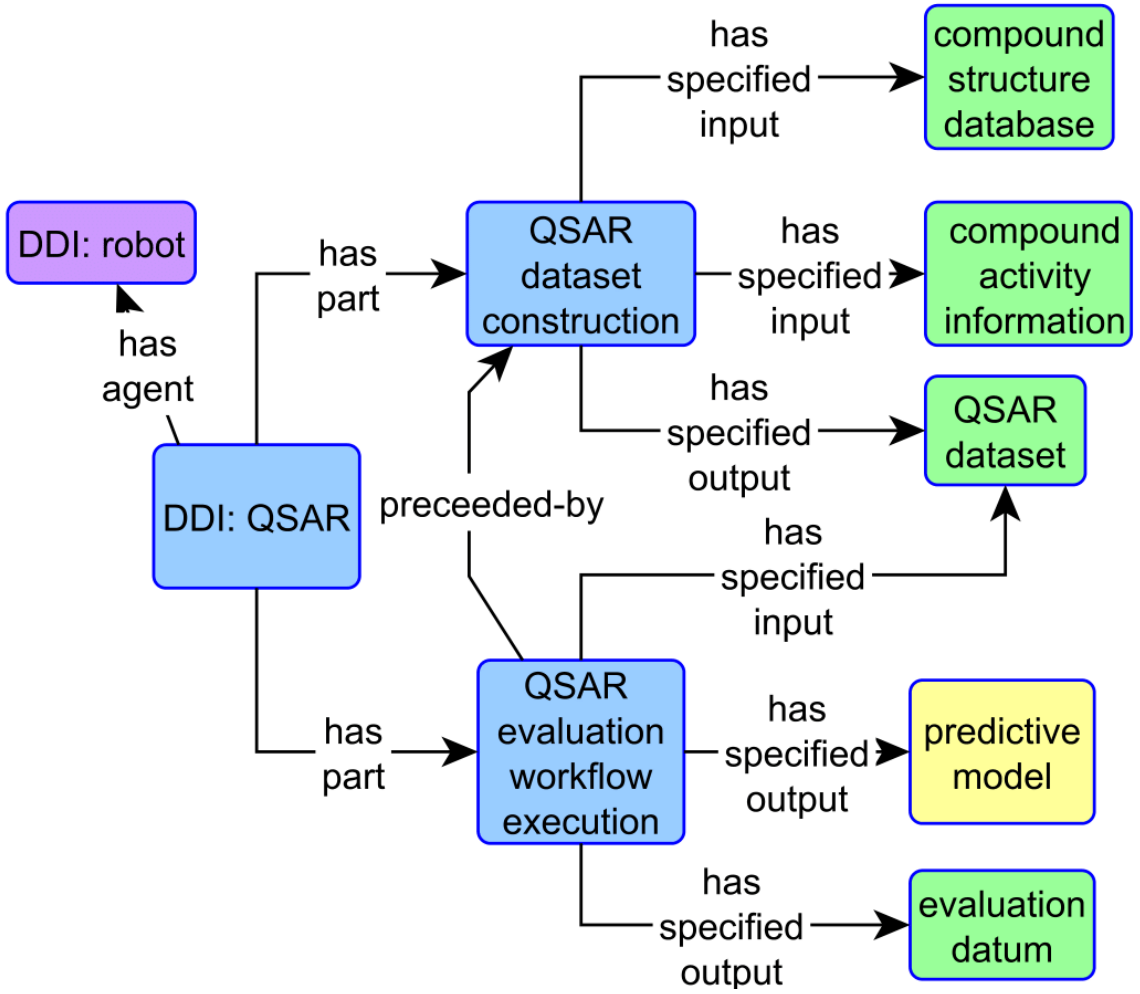
SPARQL QUERY

QUERY RESULTS

dataMiningAlgorithm
algorithm_s:clus-Bagging-MTC-PCTs
algorithm_s:clus-RandomSubspace-MTC-PCTs
algorithm_s:clus-SubBag-MTC-PCTs
algorithm_s:clus-RandomForest-MTC-PCTs
algorithm_s:clus-MTC-PCTs
algorithm_s:clus-MTR-PCTs
algorithm_s:clus-Bagging-MTR-PCTs
algorithm_s:clus-RandomSubspace-MTR-PCTs
algorithm_s:clus-RandomForest-MTR-PCTs
algorithm_s:clus-SubBag-MTR-PCTs

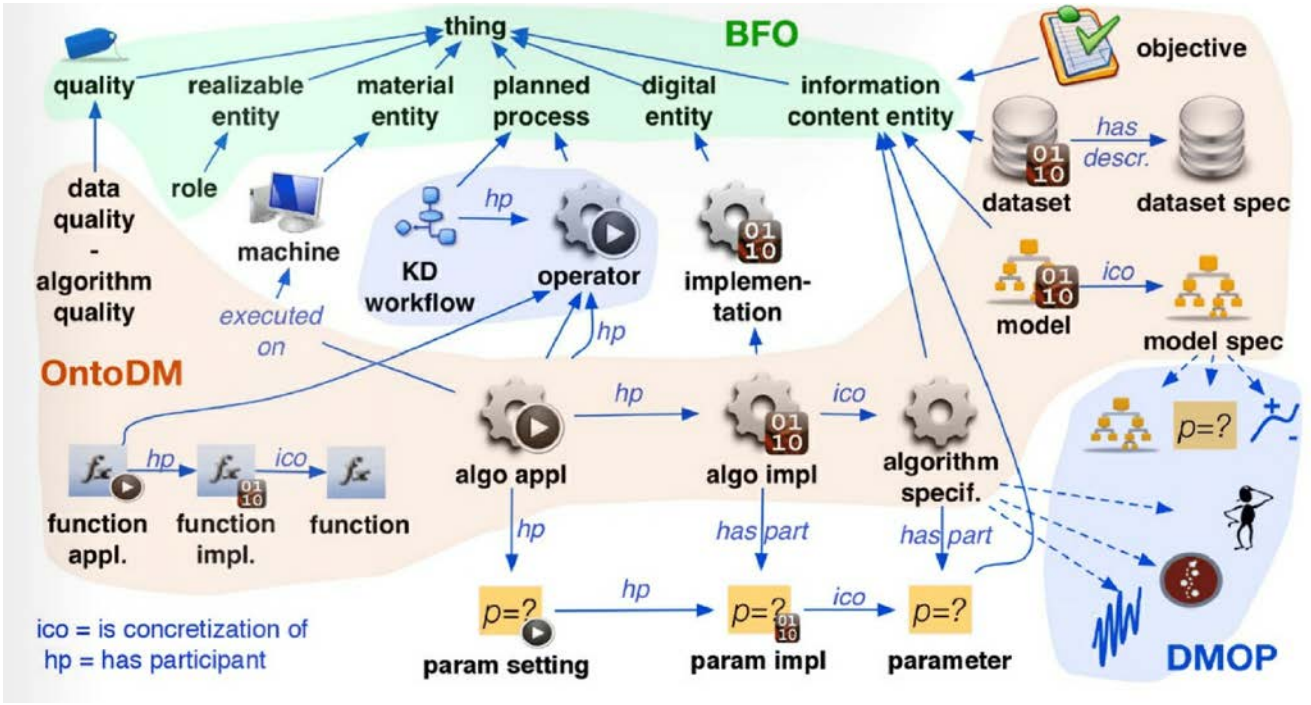
Use case: Annotation and modelling of QSAR studies

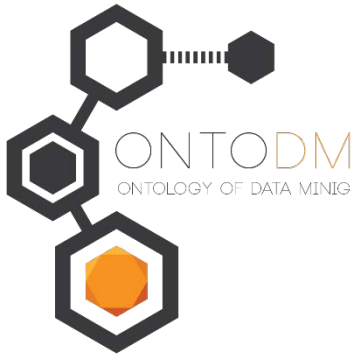
- Quantitative Structure-Activity Relationship (QSAR) modeling
  - Key components of the drug discovery pipeline
  - Used for rapid prediction and virtual pre-screening of compound activity
- QSAR algorithm is usually a DM algorithm
  - Input is a description of a set of compounds with associated pharmacological activities
  - Output is a predictive model of activity
- What did we do?
  - Used OntoDM-core in a combination with the Drug Discovery Ontology (DDI) to represent the QSAR modeling process
  - Proposed an annotation schema for annotating QSAR studies



Use case: Representing machine learning experiments

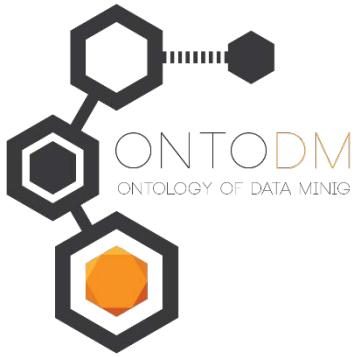
- Storing information about machine learning (ML) experiments
  - Machine learning experiment database proposed and implemented by Vanschoren et al. (2012)
  - This effort recently evolved to the OpenML platform available at http://openml.org/
- The database design and overall framework is based on the ontology named Exposé
  - Built using the same design principles as OntoDM-core
  - Uses OntoDM-core as a mid-level ontology and reuses and extends its classes





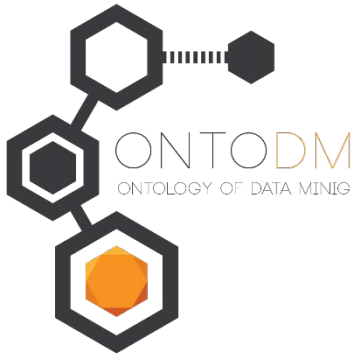
Conclusion

- OntoDM is designed and implemented by following ontology best practices and design principles
  - 3 modular ontolologies
  - Used together or independently depending on the use case
  - Fully interoperable with many application domain resources
- Generic representation used to describe the mining of structured data
  - It can be easily extended to cover new tasks and algorithm that operate on data of arbitrarily complex datatypes
- Provides support for a variety of applications:
  - Annotation and representation of datasets, data mining algorithms
  - Data mining scenarios, and knowledge discovery scenarios
  - Annotation and comparison of QSAR studies
  - Annotation of articles containing data mining terms
  - It can be used as a mid-level ontology by other ontologies



Future work

- Several directions for future work
- Align and map OntoDM-core to other upper-level ontologies (e.g., YAMATO)
- Allign to and reuse other related domain ontologies
- Extend OntoDM-core with components of learning algorithms (e.g., distance functions)
- Extend the ontology in the dimensions covered by the MAESTRA project (work in progress)
  - data stream mining
  - semi-supervised mining
  - network context
- Populate the ontology with more instances



Questions and feedback are welcome!