AI4EOSC User's Workshop 2023, Bratislava

AI4 | eosc

Model provenance and domain metadata

Mario Santa Cruz López (santacruzm@predictia.es)
Daniel San Martin Segura (daniel@predictia.es)

Co-funded by
the European Union

AI4 | eosc   Co-funded by the European Union

Model provenance and domain metadata

WP5 - Task 5.4

Motivation

Vocabulary & Ontology

PROV

PROV template

Examples

Discussion

Motivation

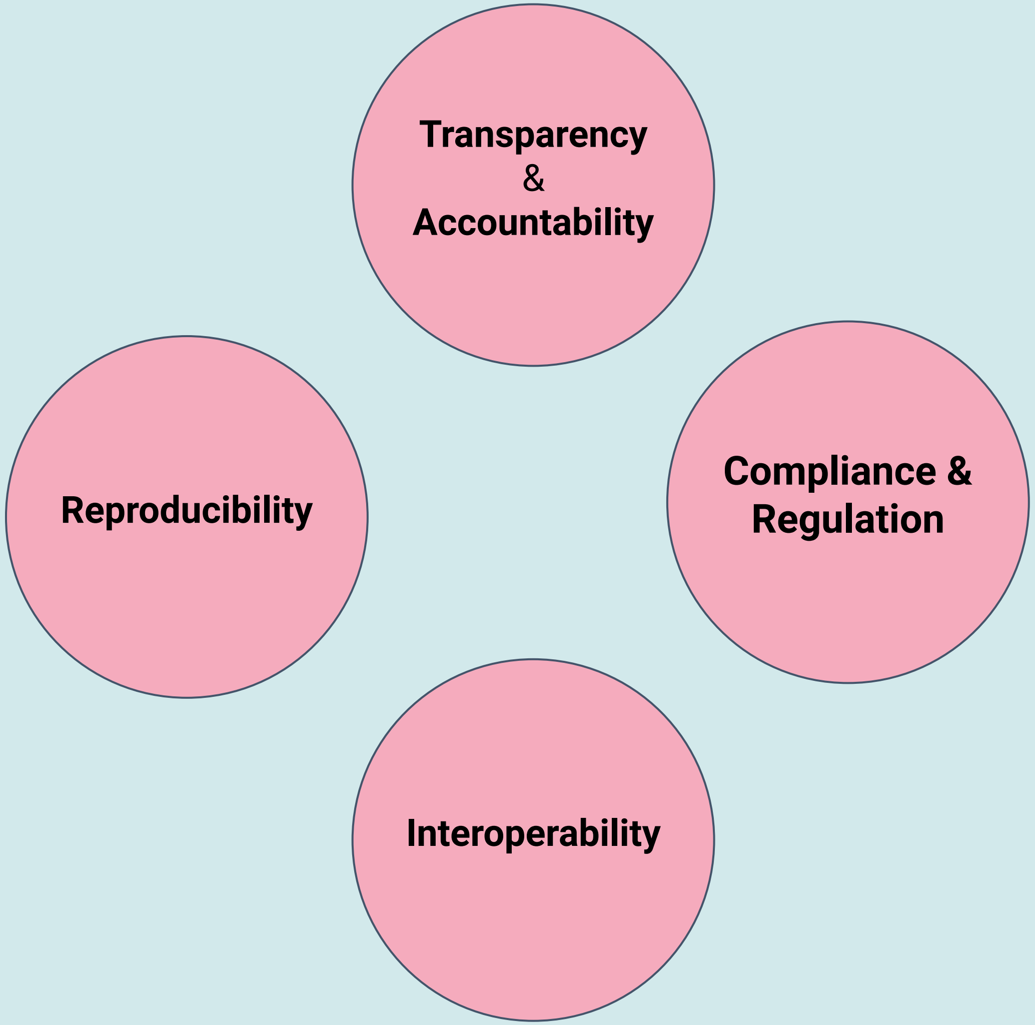
Example: A group of researchers from a research institute have developed ML model to assist in the early detection of diabetes based on factors such as blood glucose levels, medical history or lifestyle factors.

- How are the variables recorded?
- What are the units of measurement?
- What preprocessing steps were applied to the data?
- Were missing values imputed, and if so, how?
- Were outliers or anomalies addressed, and by what criteria?
- How was data quality assessed and ensured?
- What are the usage rights and restrictions?
- How were the datasets splits created?
- ….

Motivation for Semantics in ML Pipelines

Semantics refers to the use of standardized, structured descriptions or metadata to represent and understand the meaning and context of different elements within the pipeline. These elements include data, model parameters, transformations, and the relationships between them.

- Semantics can improve reproducibility in ML research and development.
- Semantics allow for the documentation of model provenance and domain metadata, making it easier to track the origins of models and their data.
- Semantics help in documenting and demonstrating compliance with regulatory requirements, ensuring that models are developed and used in accordance with established guidelines.
- Semantics can enhance interoperability by allowing different systems and tools to understand and process data and metadata consistently.



Transparency & Accountability

Reproducibility

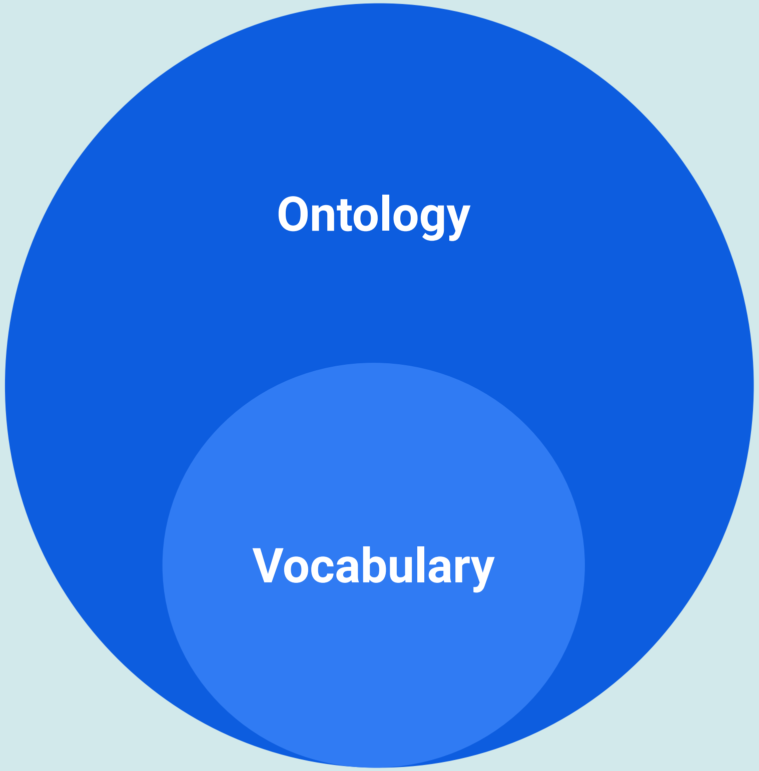
Compliance & Regulation

Interoperability

Vocabulary & Ontology

A **vocabulary** refers to a set of terms or keywords that are used to describe and categorize data, resources, or concepts. These terms are typically defined and organized in a structured way to ensure consistency in the description of metadata. A vocabulary is essentially a list of standardized terms that can be used to tag or annotate metadata, making it more understandable and interoperable.

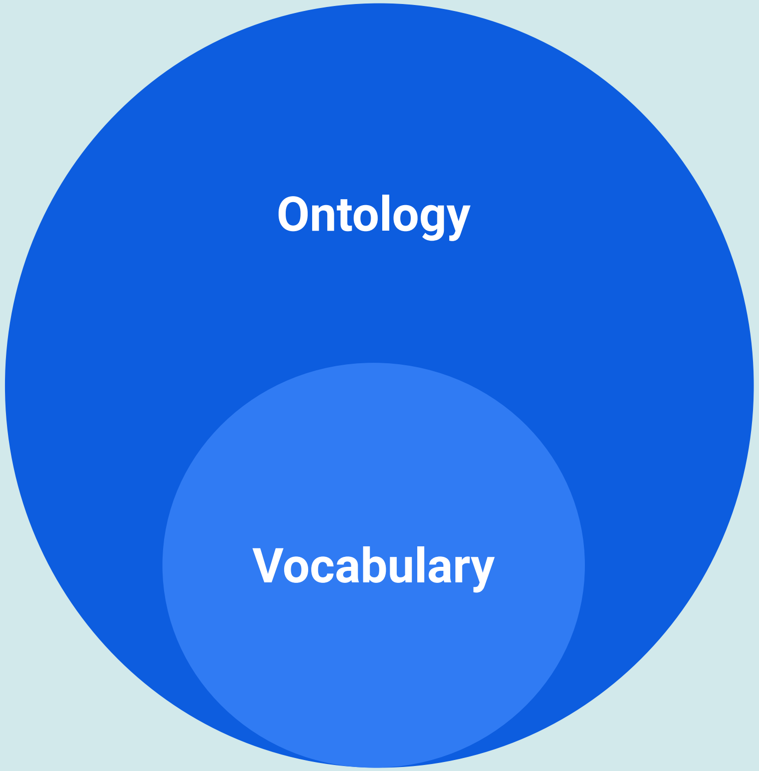
For example, in a vocabulary for a library catalog, terms might include "*author,*" "*title,*" "*subject,*" and "*publication date*". Librarians use these standardized terms to describe each book's metadata consistently.



Vocabulary & Ontology

An **ontology** is a more complex knowledge representation that defines not only terms but also the relationships between those terms and their meaning. Ontologies provide a hierarchical structure for terms and a clear understanding of how they relate to each other. They often include definitions, properties, and constraints that help specify the semantics of terms to represent domain-specific knowledge.

For example, the term "*author*" might be defined as an individual who writes the book.



Music Vocabulary

A list of common music terms that represents the essential elements in the world of music

- Song:
- Artist:
- Genre:
- Album:
- Composer:

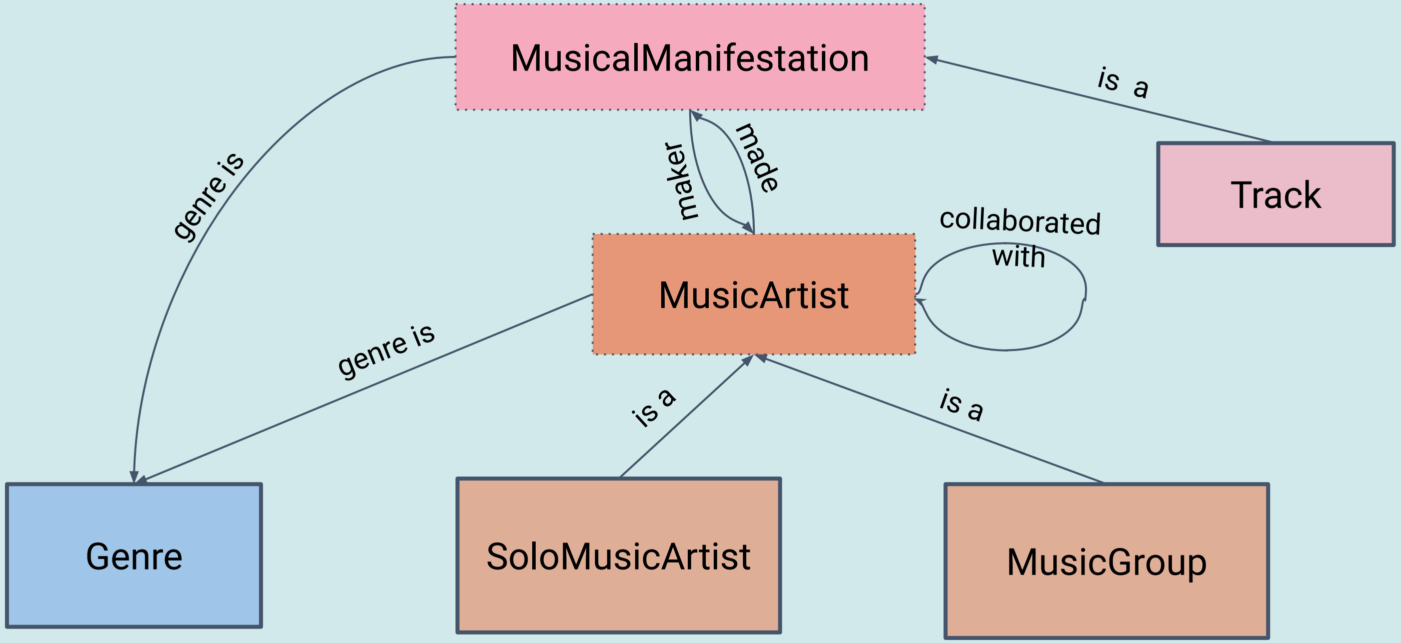
Music Vocabulary

A list of common music terms that represents the essential elements in the world of music

- Song: Bohemian Rhapsody
- Artist: Queen
- Genre: Rock
- Album: A Night at the Opera
- Composer: Freddie Mercury

Music Ontology

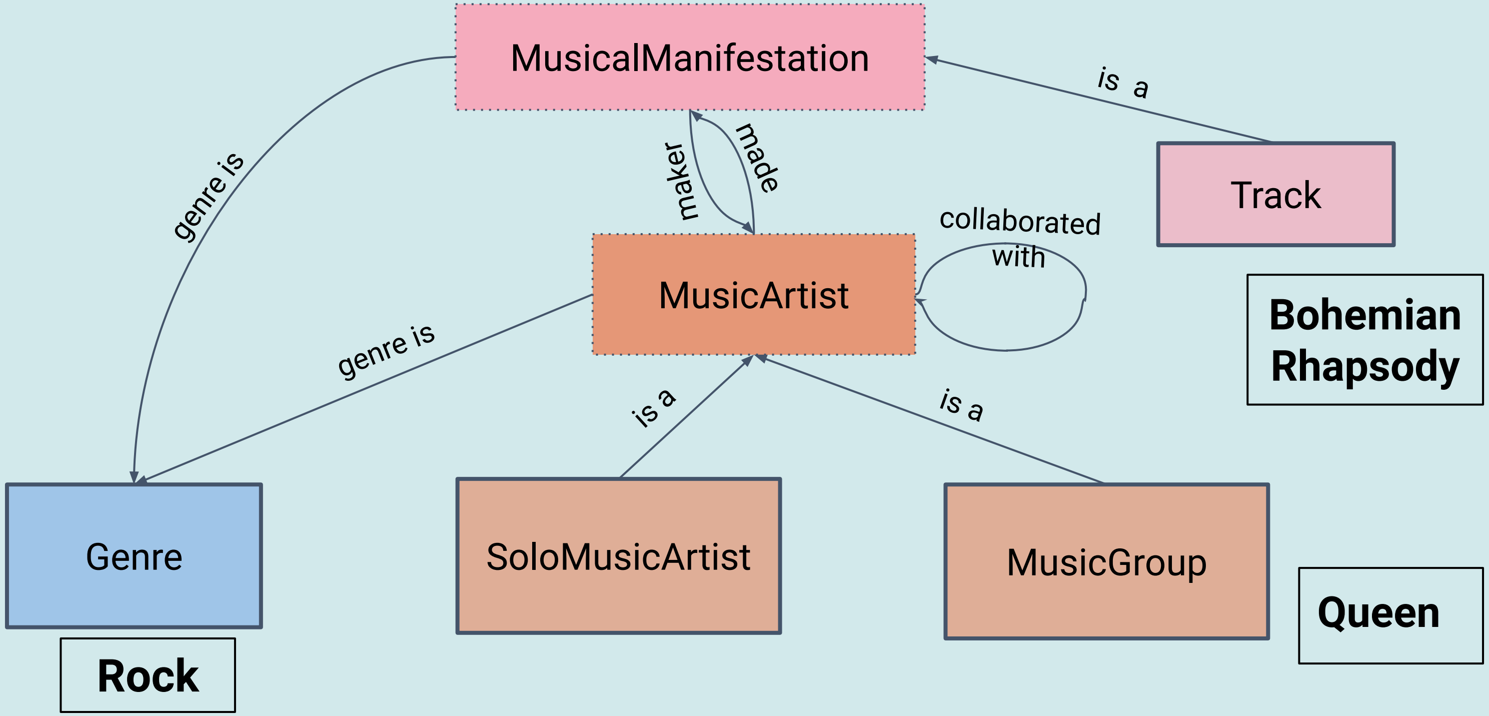
Extends this vocabulary by adding a structured hierarchy and relationships, incorporating concepts and attributes specific to music:



http://musicontology.com/

Music Ontology

Extends this vocabulary by adding a structured hierarchy and relationships, incorporating concepts and attributes specific to music:



http://musicontology.com/

PROV

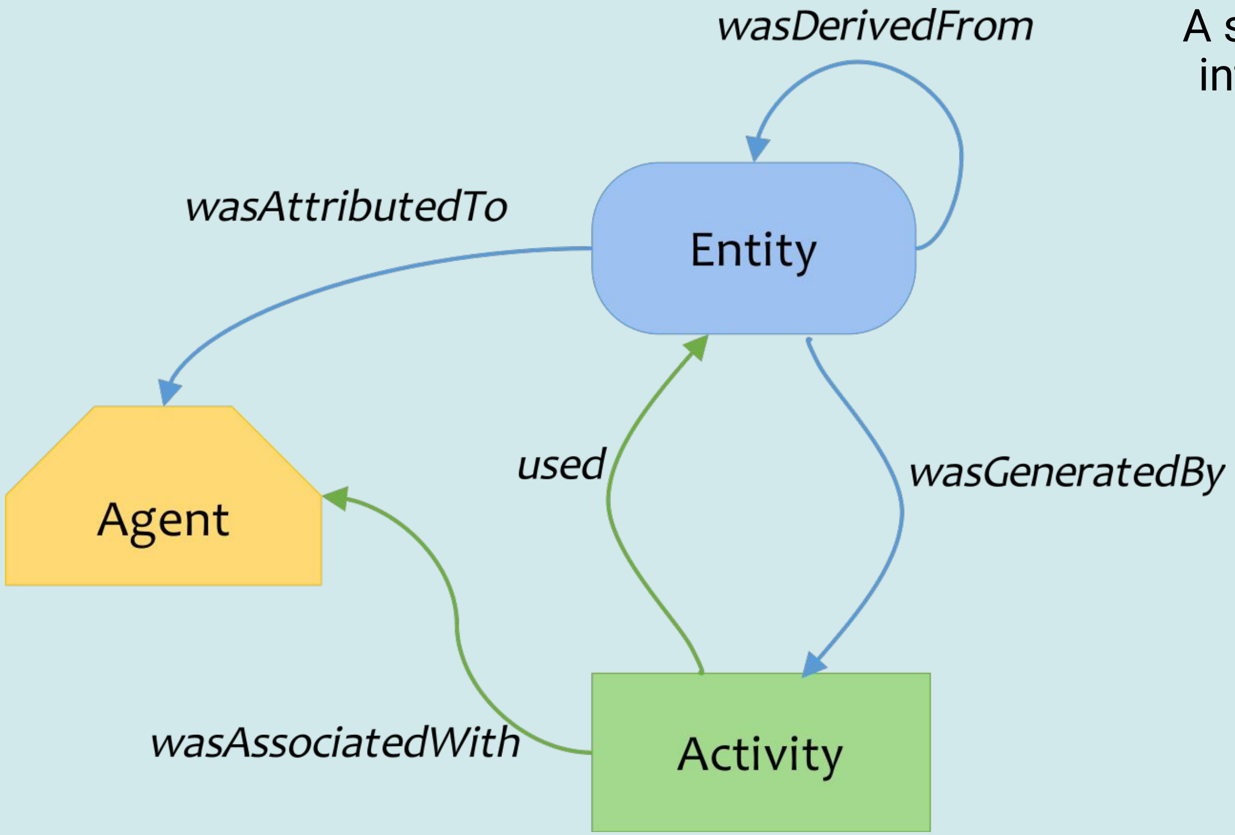


Figure 1: *Overview of the W3C PROV model.*

A standard (data model, serializations, …) to support interchange of provenance information on the Web.

Entity: *Physical, digital, conceptual, or other kind of thing (real or imaginary) with some fixed aspects.*

Activity: *Something that occurs over a period of time and acts upon or with entities; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities.*

Agent: *Something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.*

www.w3.org/TR/prov-overview/

PROV

PROV serves as a **powerful** standard for **representing** this provenance information.

As systems evolve and workflows become more intricate, documenting the provenance of models, data, and processes becomes challenging. Without a standardized approach, representing this information consistently and efficiently can be a cumbersome task.

Moreover, as provenance needs may differ across projects, domains, or even within the same project, adapting to these variations while maintaining a clear and structured representation poses a significant challenge.

PROV template

A **PROV template** is a reusable structure or pattern for representing provenance information.

It's a way to define a template that can be applied to various entities, activities, and agents to capture their provenance in a consistent manner.

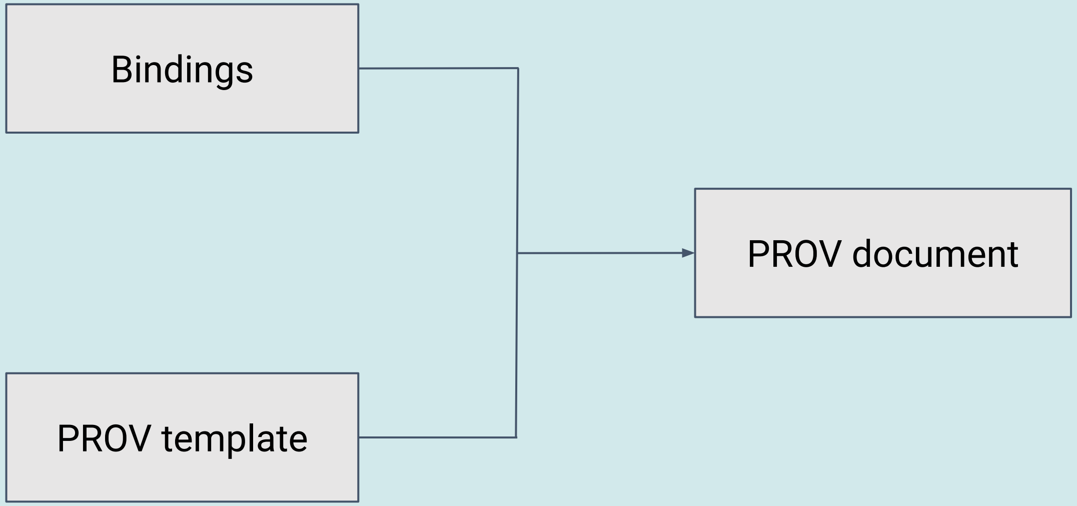
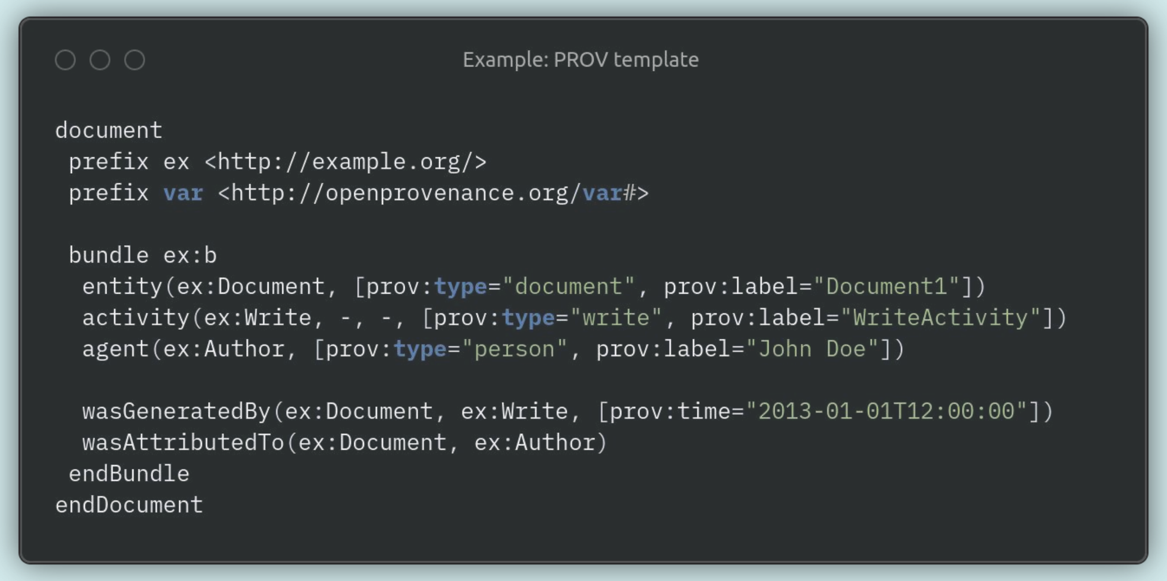


Figure 2: *Example of PROV template.*

PROV template

It might include predefined patterns for entities, activities, and agents, along with their relationships and attributes.

Using templates can help ensure that provenance information is **structured** and represented **uniformly** across different instances.



```
document
 prefix ex <http://example.org/>
 prefix var <http://openprovenance.org/var#>

 bundle ex:b
  entity(ex:Document, [prov:type="document", prov:label="Document1"])
  activity(ex:Write, -, -, [prov:type="write", prov:label="WriteActivity"])
  agent(ex:Author, [prov:type="person", prov:label="John Doe"])

  wasGeneratedBy(ex:Document, ex:Write, [prov:time="2013-01-01T12:00:00"])
  wasAttributedTo(ex:Document, ex:Author)
 endBundle
endDocument
```

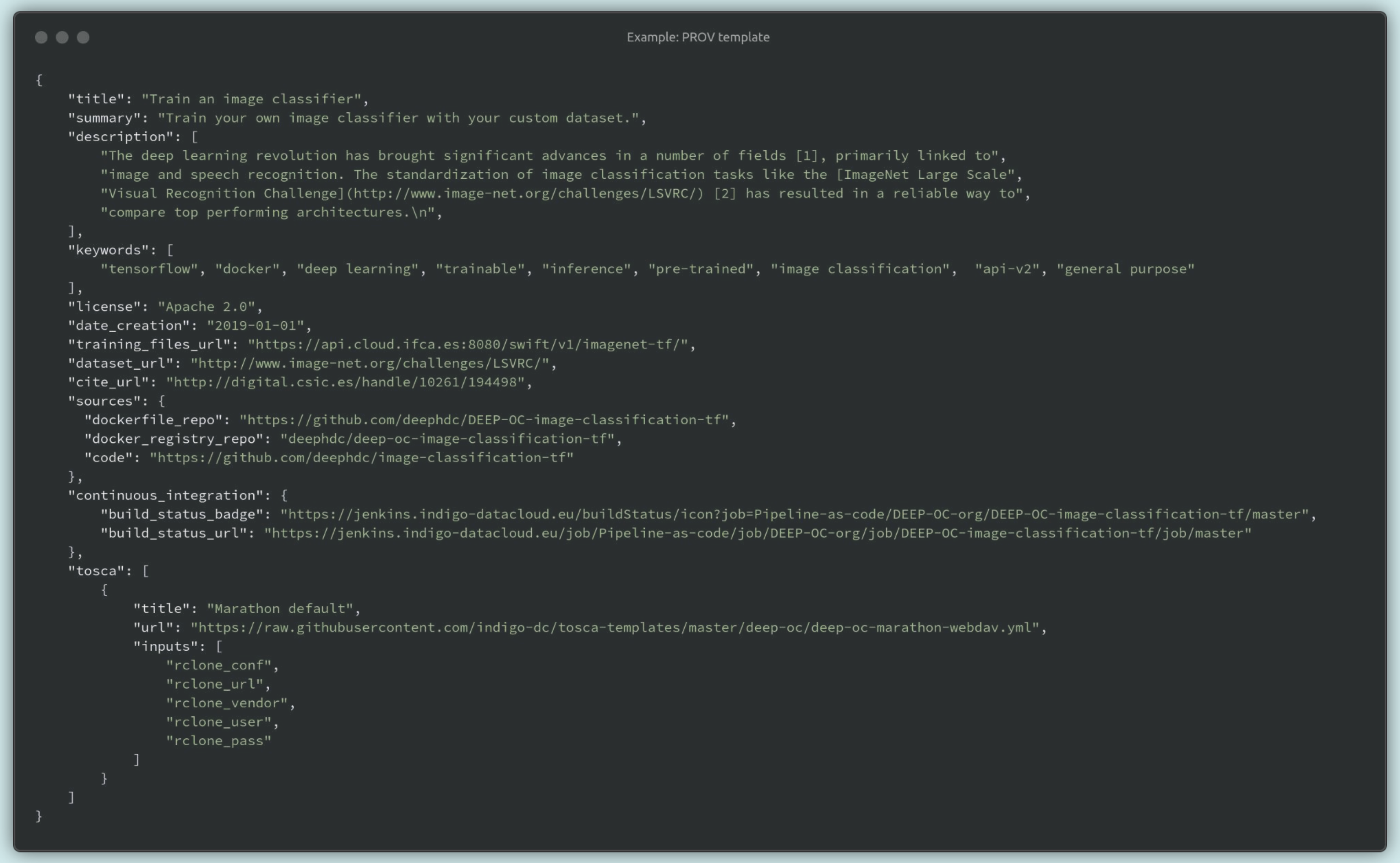
Figure 3: *Example of PROV template.*

AI4 eosc Co-funded by the European Union

Deep HDC - metadata.json

This is the metadata.json from the generic GitHub repository of image classification models containing:

- Title
- License
- Creation date
- Dataset url
- Cite url
- …



```
{
    "title": "Train an image classifier",
    "summary": "Train your own image classifier with your custom dataset.",
    "description": [
        "The deep learning revolution has brought significant advances in a number of fields [1], primarily linked to",
        "image and speech recognition. The standardization of image classification tasks like the [ImageNet Large Scale",
        "Visual Recognition Challenge](http://www.image-net.org/challenges/LSVRC/) [2] has resulted in a reliable way to",
        "compare top performing architectures.\n",
    ],
    "keywords": [
        "tensorflow", "docker", "deep learning", "trainable", "inference", "pre-trained", "image classification",  "api-v2", "general purpose"
    ],
    "license": "Apache 2.0",
    "date_creation": "2019-01-01",
    "training_files_url": "https://api.cloud.ifca.es:8080/swift/v1/imagenet-tf/",
    "dataset_url": "http://www.image-net.org/challenges/LSVRC/",
    "cite_url": "http://digital.csic.es/handle/10261/194498",
    "sources": {
      "dockerfile_repo": "https://github.com/deephdc/DEEP-OC-image-classification-tf",
      "docker_registry_repo": "deephdc/deep-oc-image-classification-tf",
      "code": "https://github.com/deephdc/image-classification-tf"
    },
    "continuous_integration": {
        "build_status_badge": "https://jenkins.indigo-datacloud.eu/buildStatus/icon?job=Pipeline-as-code/DEEP-OC-org/DEEP-OC-image-classification-tf/master",
        "build_status_url": "https://jenkins.indigo-datacloud.eu/job/Pipeline-as-code/job/DEEP-OC-org/job/DEEP-OC-image-classification-tf/job/master"
    },
    "tosca": [
        {
            "title": "Marathon default",
            "url": "https://raw.githubusercontent.com/indigo-dc/tosca-templates/master/deep-oc/deep-oc-marathon-webdav.yml",
            "inputs": [
                "rclone_conf",
                "rclone_url",
                "rclone_vendor",
                "rclone_user",
                "rclone_pass"
            ]
        }
    ]
}
```

Figure 4: *Deep HDC example of metadata.json.*

Deep HDC - bindings.provn

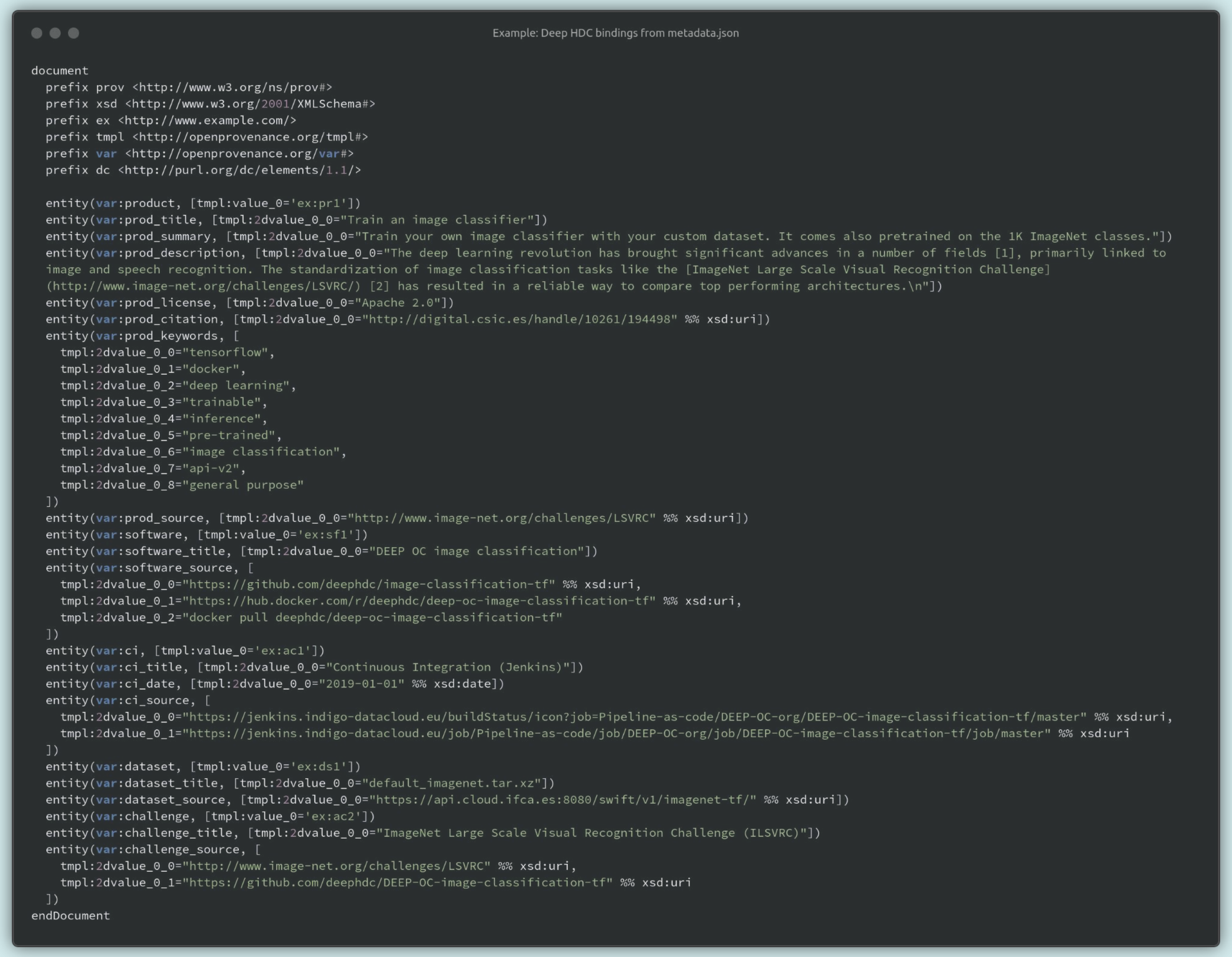
For existing products with metadata.json

Create a script to map JSON fields from the *metadata.json* to variables in a *bindings.provn*.

AI4EOSC products

For each AI4EOSC component, a *bindings.provn* will be generated with the information related to that component such as the execution time, the transformations performed, etc....

https://github.com/danielsms/ai4-prov



Example: Deep HDC bindings from metadata.json

```
document
  prefix prov <http://www.w3.org/ns/prov#>
  prefix xsd <http://www.w3.org/2001/XMLSchema#>
  prefix ex <http://www.example.com/>
  prefix tmpl <http://openprovenance.org/tmpl#>
  prefix var <http://openprovenance.org/var#>
  prefix dc <http://purl.org/dc/elements/1.1/>

  entity(var:product, [tmpl:value_0='ex:pr1'])
  entity(var:prod_title, [tmpl:2dvalue_0_0="Train an image classifier"])
  entity(var:prod_summary, [tmpl:2dvalue_0_0="Train your own image classifier with your custom dataset. It comes also pretrained on the 1K ImageNet classes."])
  entity(var:prod_description, [tmpl:2dvalue_0_0="The deep learning revolution has brought significant advances in a number of fields [1], primarily linked to
  image and speech recognition. The standardization of image classification tasks like the [ImageNet Large Scale Visual Recognition Challenge]
  (http://www.image-net.org/challenges/LSVRC/) [2] has resulted in a reliable way to compare top performing architectures.\n"])
  entity(var:prod_license, [tmpl:2dvalue_0_0="Apache 2.0"])
  entity(var:prod_citation, [tmpl:2dvalue_0_0="http://digital.csic.es/handle/10261/194498" %% xsd:uri])
  entity(var:prod_keywords, [
    tmpl:2dvalue_0_0="tensorflow",
    tmpl:2dvalue_0_1="docker",
    tmpl:2dvalue_0_2="deep learning",
    tmpl:2dvalue_0_3="trainable",
    tmpl:2dvalue_0_4="inference",
    tmpl:2dvalue_0_5="pre-trained",
    tmpl:2dvalue_0_6="image classification",
    tmpl:2dvalue_0_7="api-v2",
    tmpl:2dvalue_0_8="general purpose"
  ])
  entity(var:prod_source, [tmpl:2dvalue_0_0="http://www.image-net.org/challenges/LSVRC" %% xsd:uri])
  entity(var:software, [tmpl:value_0='ex:sf1'])
  entity(var:software_title, [tmpl:2dvalue_0_0="DEEP OC image classification"])
  entity(var:software_source, [
    tmpl:2dvalue_0_0="https://github.com/deephdc/image-classification-tf" %% xsd:uri,
    tmpl:2dvalue_0_1="https://hub.docker.com/r/deephdc/deep-oc-image-classification-tf" %% xsd:uri,
    tmpl:2dvalue_0_2="docker pull deephdc/deep-oc-image-classification-tf"
  ])
  entity(var:ci, [tmpl:value_0='ex:ac1'])
  entity(var:ci_title, [tmpl:2dvalue_0_0="Continuous Integration (Jenkins)"])
  entity(var:ci_date, [tmpl:2dvalue_0_0="2019-01-01" %% xsd:date])
  entity(var:ci_source, [
    tmpl:2dvalue_0_0="https://jenkins.indigo-datacloud.eu/buildStatus/icon?job=Pipeline-as-code/DEEP-OC-org/DEEP-OC-image-classification-tf/master" %% xsd:uri,
    tmpl:2dvalue_0_1="https://jenkins.indigo-datacloud.eu/job/Pipeline-as-code/job/DEEP-OC-org/job/DEEP-OC-image-classification-tf/job/master" %% xsd:uri
  ])
  entity(var:dataset, [tmpl:value_0='ex:ds1'])
  entity(var:dataset_title, [tmpl:2dvalue_0_0="default_imagenet.tar.xz"])
  entity(var:dataset_source, [tmpl:2dvalue_0_0="https://api.cloud.ifca.es:8080/swift/v1/imagenet-tf/" %% xsd:uri])
  entity(var:challenge, [tmpl:value_0='ex:ac2'])
  entity(var:challenge_title, [tmpl:2dvalue_0_0="ImageNet Large Scale Visual Recognition Challenge (ILSVRC)"])
  entity(var:challenge_source, [
    tmpl:2dvalue_0_0="http://www.image-net.org/challenges/LSVRC" %% xsd:uri,
    tmpl:2dvalue_0_1="https://github.com/deephdc/DEEP-OC-image-classification-tf" %% xsd:uri
  ])
endDocument
```

Figure 5: Bindings for *previous metadata.json example.*

AI4 | eosc
Co-funded by
the European Union

Deep HDC - template.provn

The example PROV template defines 2 entities, 2 activities, 1 agent and 5 relations between them.

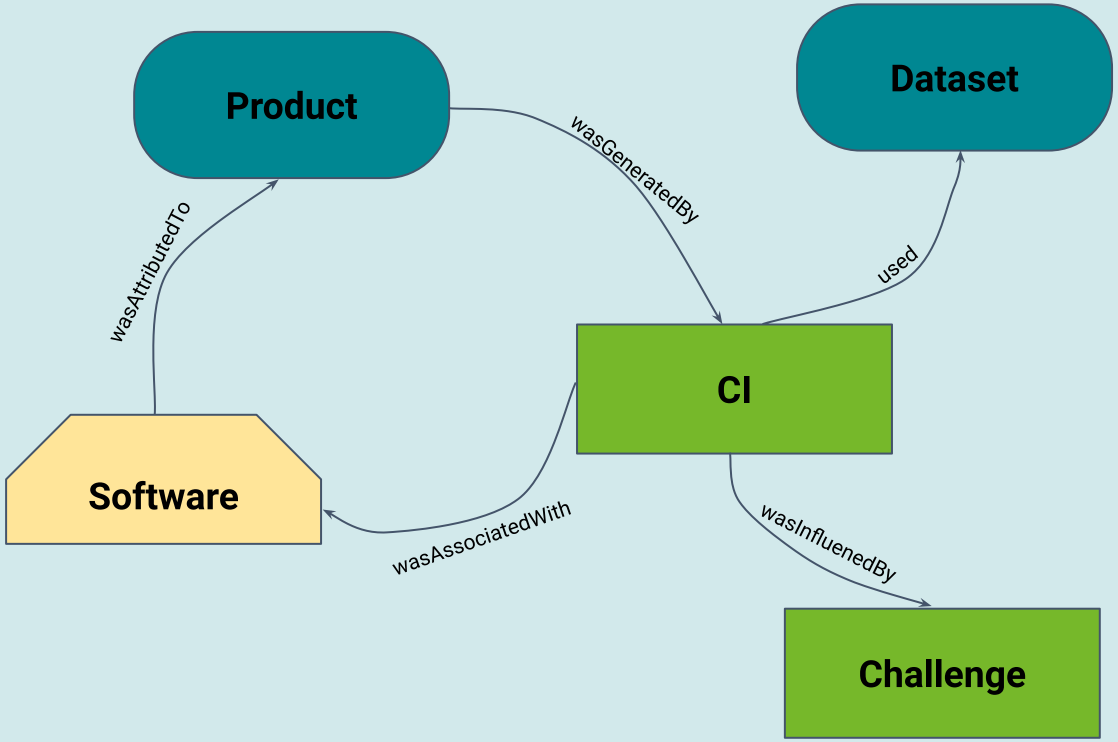
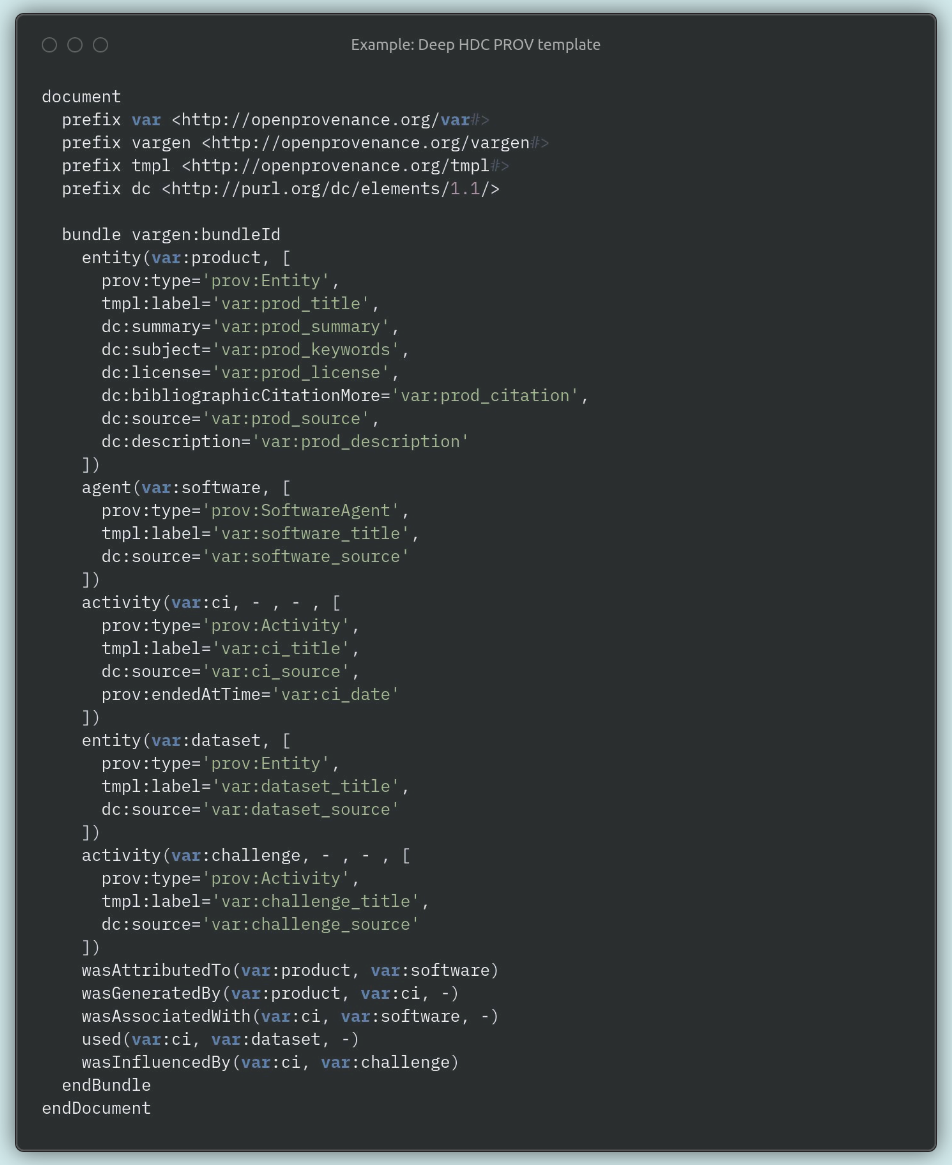


Figure 6: Diagram of *PROV template example, template.provn.*

https://github.com/danielsms/ai4-prov



```
document
  prefix var <http://openprovenance.org/var#>
  prefix vargen <http://openprovenance.org/vargen#>
  prefix tmpl <http://openprovenance.org/tmpl#>
  prefix dc <http://purl.org/dc/elements/1.1/>

  bundle vargen:bundleId
    entity(var:product, [
      prov:type='prov:Entity',
      tmpl:label='var:prod_title',
      dc:summary='var:prod_summary',
      dc:subject='var:prod_keywords',
      dc:license='var:prod_license',
      dc:bibliographicCitationMore='var:prod_citation',
      dc:source='var:prod_source',
      dc:description='var:prod_description'
    ])
    agent(var:software, [
      prov:type='prov:SoftwareAgent',
      tmpl:label='var:software_title',
      dc:source='var:software_source'
    ])
    activity(var:ci, - , - , [
      prov:type='prov:Activity',
      tmpl:label='var:ci_title',
      dc:source='var:ci_source',
      prov:endedAtTime='var:ci_date'
    ])
    entity(var:dataset, [
      prov:type='prov:Entity',
      tmpl:label='var:dataset_title',
      dc:source='var:dataset_source'
    ])
    activity(var:challenge, - , - , [
      prov:type='prov:Activity',
      tmpl:label='var:challenge_title',
      dc:source='var:challenge_source'
    ])
    wasAttributedTo(var:product, var:software)
    wasGeneratedBy(var:product, var:ci, -)
    wasAssociatedWith(var:ci, var:software, -)
    used(var:ci, var:dataset, -)
    wasInfluencedBy(var:ci, var:challenge)
  endBundle
endDocument
```

Figure 7: *Example of PROV template, template.provn, for a Deep Hybrid DataCloud product.*

Deep HDC - output.provn

This is a **PROV document** generated following the general structure specified in the *template.provn* file.

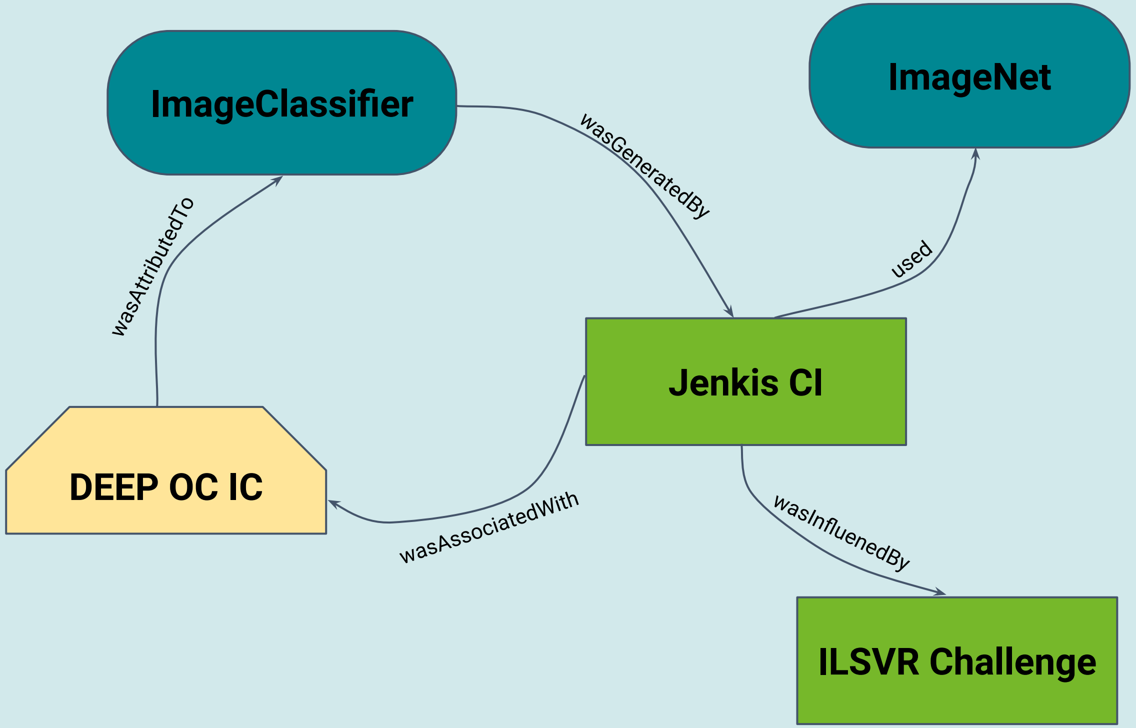
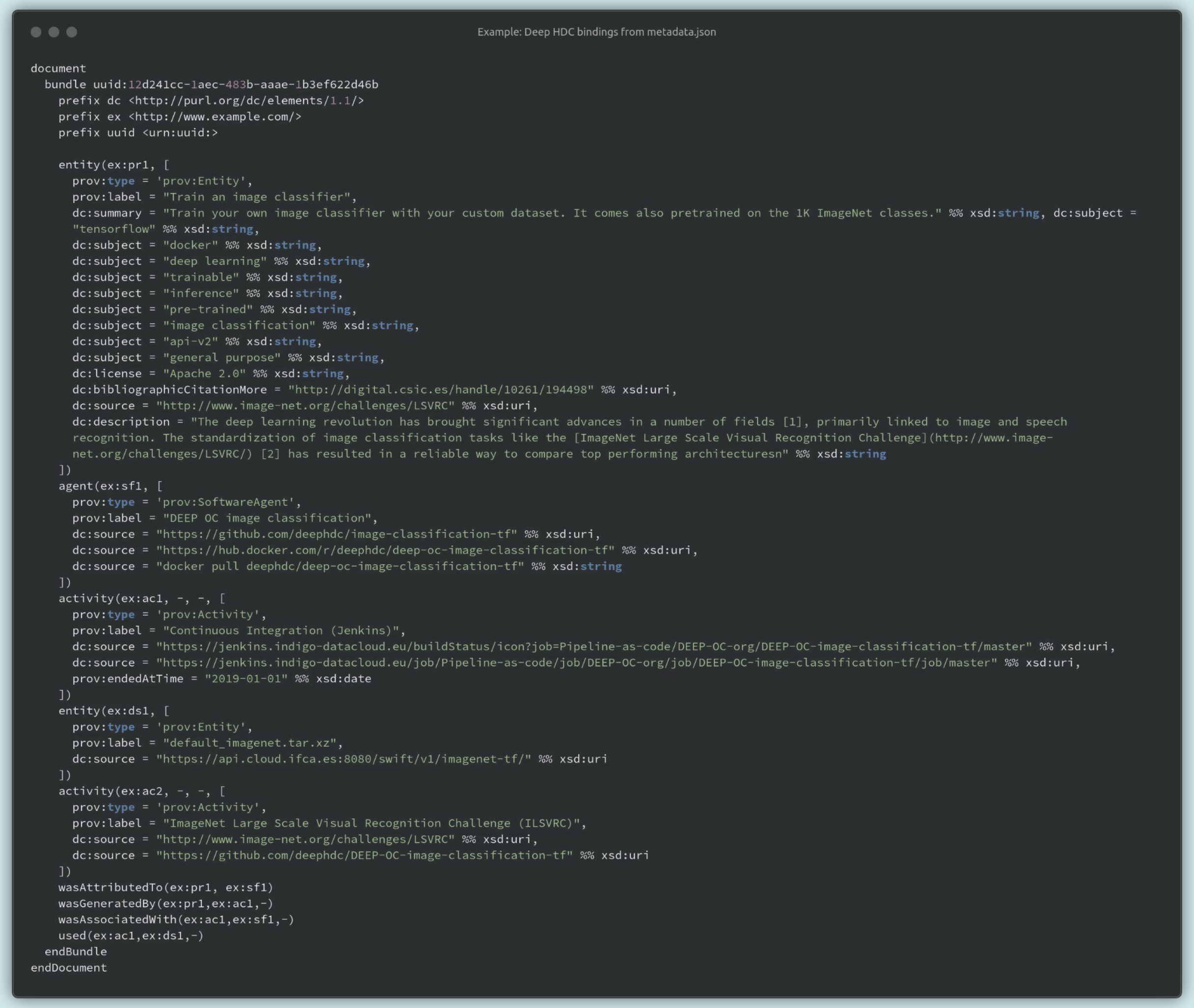


Figure 8: Diagram of *PROV document example.*

https://github.com/danielsms/ai4-prov



```
document
  bundle uuid:12d241cc-1aec-483b-aaae-1b3ef622d46b
    prefix dc <http://purl.org/dc/elements/1.1/>
    prefix ex <http://www.example.com/>
    prefix uuid <urn:uuid:>

    entity(ex:pr1, [
      prov:type = 'prov:Entity',
      prov:label = "Train an image classifier",
      dc:summary = "Train your own image classifier with your custom dataset. It comes also pretrained on the 1K ImageNet classes." %% xsd:string, dc:subject =
      "tensorflow" %% xsd:string,
      dc:subject = "docker" %% xsd:string,
      dc:subject = "deep learning" %% xsd:string,
      dc:subject = "trainable" %% xsd:string,
      dc:subject = "inference" %% xsd:string,
      dc:subject = "pre-trained" %% xsd:string,
      dc:subject = "image classification" %% xsd:string,
      dc:subject = "api-v2" %% xsd:string,
      dc:subject = "general purpose" %% xsd:string,
      dc:license = "Apache 2.0" %% xsd:string,
      dc:bibliographicCitationMore = "http://digital.csic.es/handle/10261/194498" %% xsd:uri,
      dc:source = "http://www.image-net.org/challenges/LSVRC" %% xsd:uri,
      dc:description = "The deep learning revolution has brought significant advances in a number of fields [1], primarily linked to image and speech
      recognition. The standardization of image classification tasks like the [ImageNet Large Scale Visual Recognition Challenge](http://www.image-
      net.org/challenges/LSVRC/) [2] has resulted in a reliable way to compare top performing architecturesn" %% xsd:string
    ])
    agent(ex:sf1, [
      prov:type = 'prov:SoftwareAgent',
      prov:label = "DEEP OC image classification",
      dc:source = "https://github.com/deephdc/image-classification-tf" %% xsd:uri,
      dc:source = "https://hub.docker.com/r/deephdc/deep-oc-image-classification-tf" %% xsd:uri,
      dc:source = "docker pull deephdc/deep-oc-image-classification-tf" %% xsd:string
    ])
    activity(ex:ac1, -, -, [
      prov:type = 'prov:Activity',
      prov:label = "Continuous Integration (Jenkins)",
      dc:source = "https://jenkins.indigo-datacloud.eu/buildStatus/icon?job=Pipeline-as-code/DEEP-OC-org/DEEP-OC-image-classification-tf/master" %% xsd:uri,
      dc:source = "https://jenkins.indigo-datacloud.eu/job/Pipeline-as-code/job/DEEP-OC-org/job/DEEP-OC-image-classification-tf/job/master" %% xsd:uri,
      prov:endedAtTime = "2019-01-01" %% xsd:date
    ])
    entity(ex:ds1, [
      prov:type = 'prov:Entity',
      prov:label = "default_imagenet.tar.xz",
      dc:source = "https://api.cloud.ifca.es:8080/swift/v1/imagenet-tf/" %% xsd:uri
    ])
    activity(ex:ac2, -, -, [
      prov:type = 'prov:Activity',
      prov:label = "ImageNet Large Scale Visual Recognition Challenge (ILSVRC)",
      dc:source = "http://www.image-net.org/challenges/LSVRC" %% xsd:uri,
      dc:source = "https://github.com/deephdc/DEEP-OC-image-classification-tf" %% xsd:uri
    ])
    wasAttributedTo(ex:pr1, ex:sf1)
    wasGeneratedBy(ex:pr1,ex:ac1,-)
    wasAssociatedWith(ex:ac1,ex:sf1,-)
    used(ex:ac1,ex:ds1,-)
  endBundle
endDocument
```

Figure 9: *Output PROV document of the previous examples.*

But...

For example, the *ImageClassifier* or *ImageNet* are entities not defined. Both can be seen as subclasses of entity, but *ImageNet* refers to a dataset and *ImageClassifier* to a model.

Solutions:

- Define them ourselves in our own ontology.
- Extend existing ontologies as far as possible.
  - **ProvONE**: A extension Data Model for Scientific Workflow Provenance
  - **Prov-ML**: A data representation for Machine Learning.

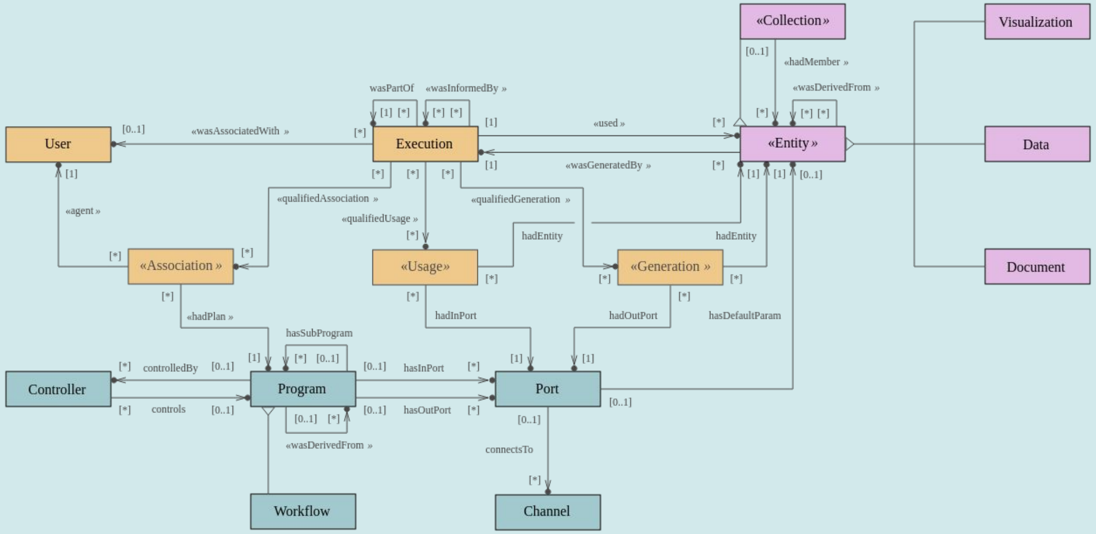


Figure 10: *ProvONE* conceptual Model UML diagram.

AI4 | eosc

Co-funded by
the European Union

AI4EOSC

ai4eosc-po@listas.csic.es

ai4eosc.eu

Reach us!

Thank you for your attention
Project Coordinator: Álvaro López García - aloga@ifca.unican.es