Jan 26-28
2021 ESIP Winter Meeting

Transparency in Data Management
Capturing Provenance of Data Curation at BCO-DMO

BCO-DMO
Biological & Chemical Oceanography Data Management Office

Adam Shepherd, Woods Hole Oceanographic Institution | **Amber York**, Woods Hole Oceanographic Institution | **Conrad Schloer**, Woods Hole Oceanographic Institution | **Karen Soenen**, Woods Hole Oceanographic Institution

BCO-DMO, a domain-specific repository

The Biological and Chemical Oceanography Data Management Office (BCO-DMO) staff work closely with investigators to serve data and information online from research projects funded by the U.S. National Science Foundation (NSF) Biological and Chemical Oceanography Programs, and the Office of Polar Programs Antarctic Organisms & Ecosystems Program.

The goal of this partnership is to effectively curate marine ecosystem data and accompanying documentation, facilitating efficient data discovery and re-use. Throughout the process, BCO-DMO provides services that support specific phases of the data life cycle.

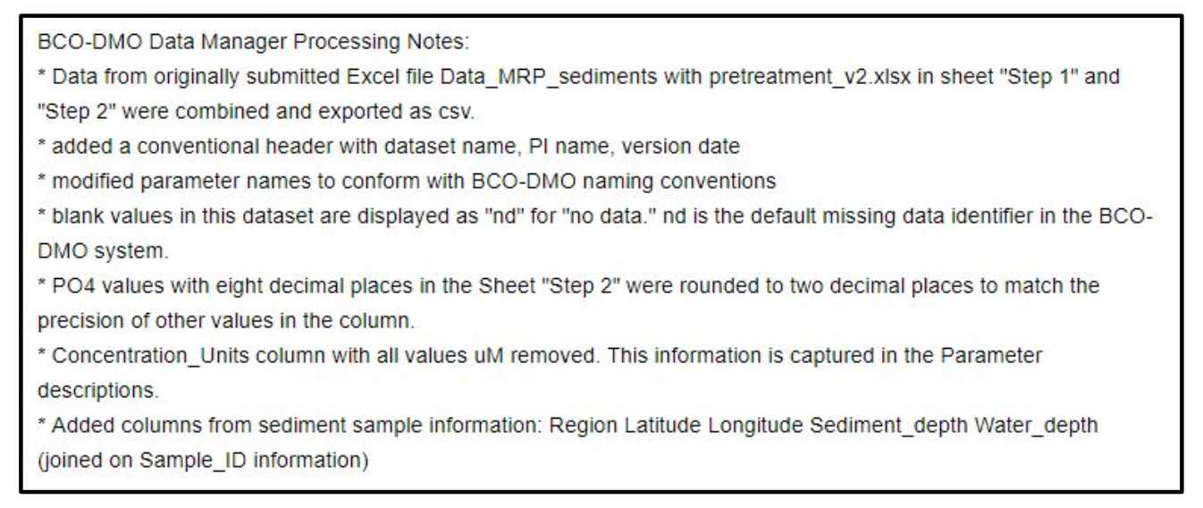
The result is a rich database of research-ready data spanning the full range of marine ecosystem related measurements including in-situ and remotely sensed observations, experimental and model results, and synthesis products. The BCO-DMO system provides access to more than 9000 data sets from more than 900 projects and 2500 researchers.

Why Capturing Provenance?

- An audit-trail explaining what has changed from one version to another
- Evidence of why domain-specific data management are needed for FAIR-*ness*
- Transparency about how archived data differs from originally submitted version

Why ad-hoc changes are insufficient?

1. Hard to Maintain
   - code over time has dependencies
     - software & library updates
   - a single DM is the expert; hard for others to help
   - is documentation as text good enough?



BCO-DMO Data Manager Processing Notes:
* Data from originally submitted Excel file Data_MRP_sediments with pretreatment_v2.xlsx in sheet "Step 1" and "Step 2" were combined and exported as csv.
* added a conventional header with dataset name, PI name, version date
* modified parameter names to conform with BCO-DMO naming conventions
* blank values in this dataset are displayed as "nd" for "no data." nd is the default missing data identifier in the BCO-DMO system.
* PO4 values with eight decimal places in the Sheet "Step 2" were rounded to two decimal places to match the precision of other values in the column.
* Concentration_Units column with all values uM removed. This information is captured in the Parameter descriptions.
* Added columns from sediment sample information: Region Latitude Longitude Sediment_depth Water_depth (joined on Sample_ID information)

2. Duplication of effort
   - same need written multiple times across multiple technologies
   - different DMs write same code for different projects
   - difficult to foster/enforce reuse of code bits
   - are all implementations executing the same procedures? same order?

Declarative Workflows (DWs) to Capture Provenance

* Declarative workflows focus on '**what**' to do.
* Software focuses on '**how**' best to do it.

Declarative workflows as a tool for **Standardised Collaboration**:
- *Consistently* express DMs intents
- *Approachable* for non-coders: focus on 'what' should be done over 'how' to do it
- A *shared language*: DMs *understand* each others pipelines

Declarative workflows serve as **Workable Data**:
- Configuration data: code interpreting DWs can be changed/swapped without impacting DM intent
- .yaml files because declarative workflows are stored in a data format they can be automatically converted into a provenance record/ machine actionable provenance record.

Frictionless Data - Data Package Pipelines

FRICTIONLESS DATA

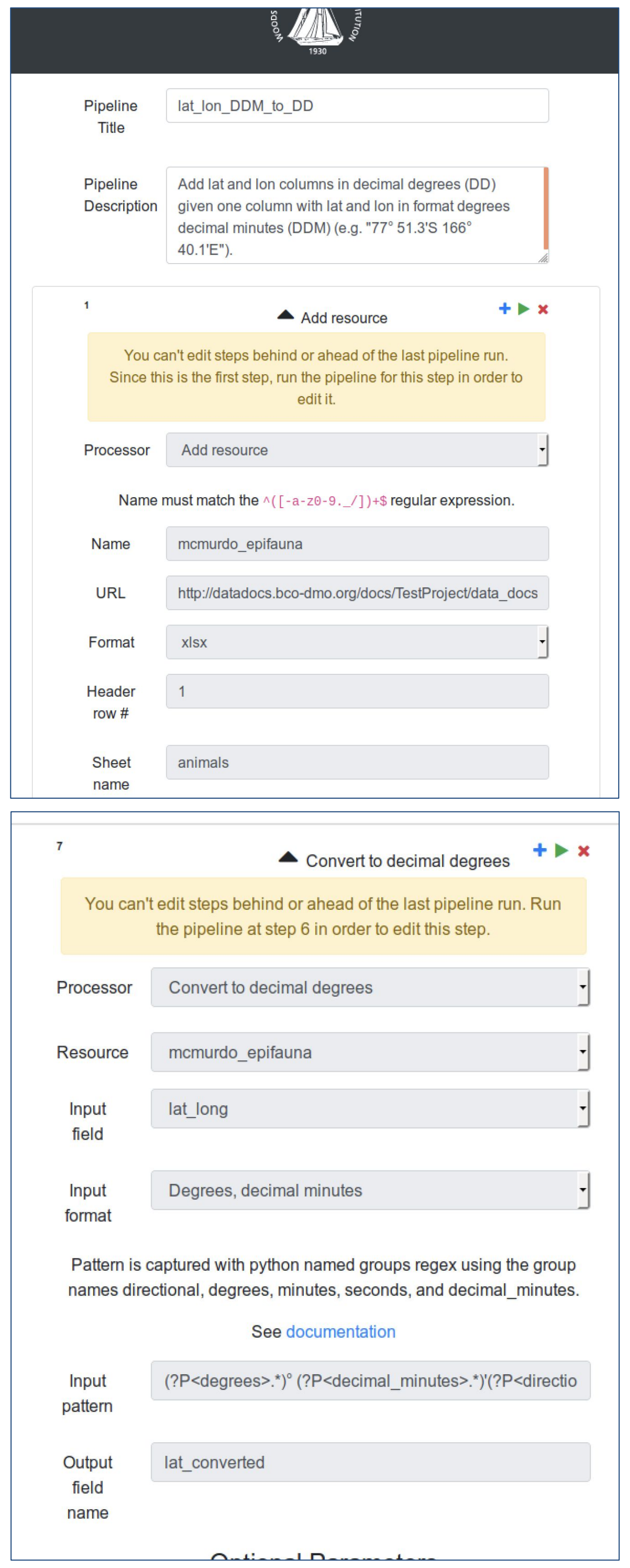
Framework to build stream-processing tabular data workflows.

- Processing data packages using pipelines/workflows
- Modular components: BCO-DMO built domain-specific processors
- A pipeline is a list of processing steps for a data package.
- Processing steps are defined in a declarative way using YAML.
- 20 processors in the standard library
- BCO-DMO extended with 18 additional processors

github.com/frictionlessdata/datapackage-pipelines   github.com/BCODMO/bcodmo_frictionless

BCO-DMO UI for Frictionlessdata Datapackage Pipelines

Watch Demo !!!
(YouTube)



Pipeline Title: lat_lon_DDM_to_DD

Pipeline Description: Add lat and lon columns in decimal degrees (DD) given one column with lat and lon in format degrees decimal minutes (DDM) (e.g. "77° 51.3'S 166° 40.1'E").

1 ▲ Add resource

You can't edit steps behind or ahead of the last pipeline run. Since this is the first step, run the pipeline for this step in order to edit it.

Processor: Add resource

Name must match the ^([-a-z0-9._/])+$ regular expression.

Name: mcmurdo_epifauna

URL: http://datadocs.bco-dmo.org/docs/TestProject/data_docs

Format: xlsx

Header row #: 1

Sheet name: animals

7 ▲ Convert to decimal degrees

You can't edit steps behind or ahead of the last pipeline run. Run the pipeline at step 6 in order to edit this step.

Processor: Convert to decimal degrees

Resource: mcmurdo_epifauna

Input field: lat_long

Input format: Degrees, decimal minutes

Pattern is captured with python named groups regex using the group names directional, degrees, minutes, seconds, and decimal_minutes.

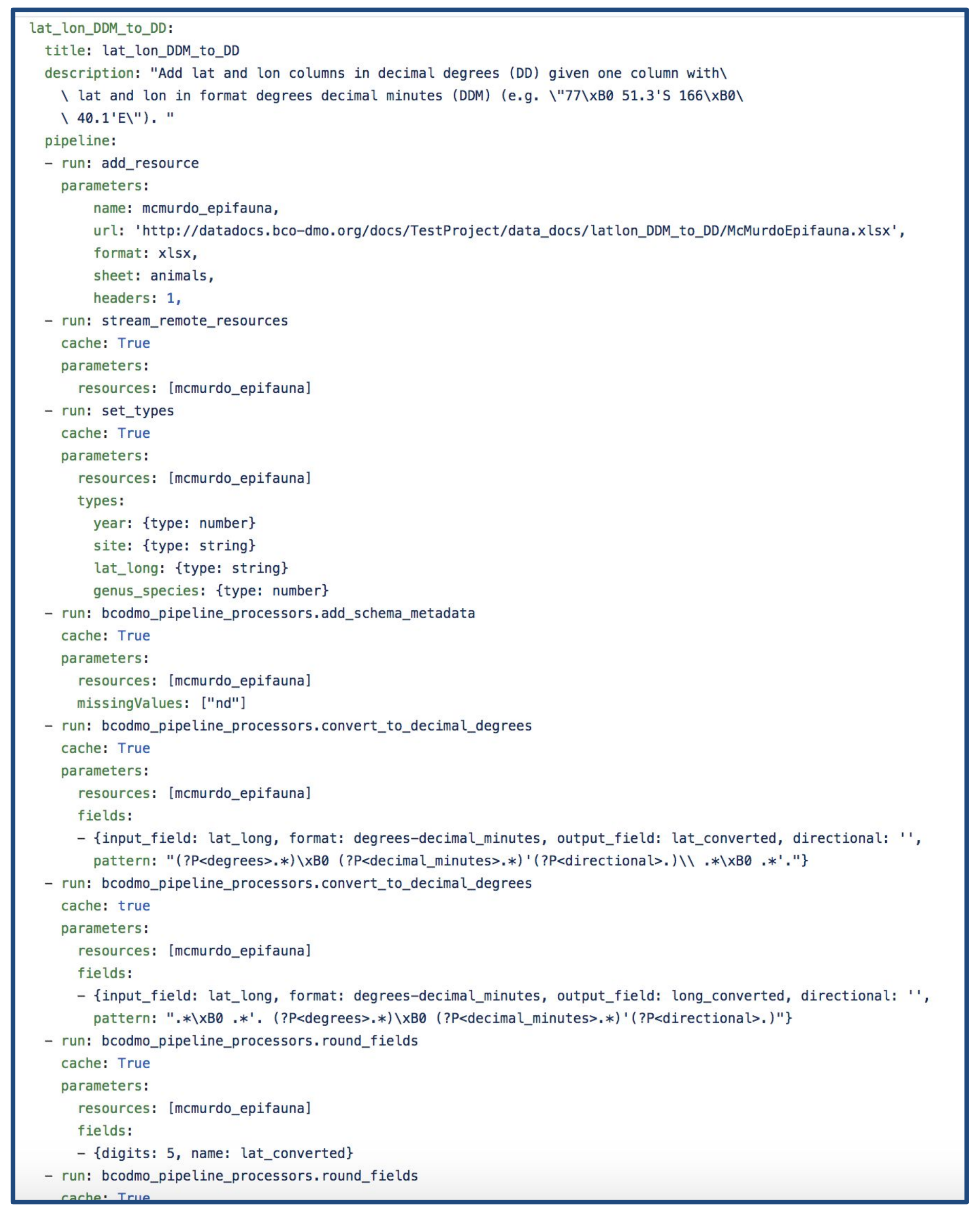
See documentation

Input pattern: (?P<degrees>.*)° (?P<decimal_minutes>.*)'(?P<directio

Output field name: lat_converted

Optional Parameters

- **Encourages consistency**
  across commonly occurring processing tasks
- **Ensures proper data validation** occurs
  before data made publicly available
- *Example* pipeline-spec.yaml
- *Example* datapackage.json



```
lat_lon_DDM_to_DD:
  title: lat_lon_DDM_to_DD
  description: "Add lat and lon columns in decimal degrees (DD) given one column with\
    \ lat and lon in format degrees decimal minutes (DDM) (e.g. \"77\xB0 51.3'S 166\xB0\
    \ 40.1'E\"). "
  pipeline:
  - run: add_resource
    parameters:
        name: mcmurdo_epifauna,
        url: 'http://datadocs.bco-dmo.org/docs/TestProject/data_docs/latlon_DDM_to_DD/McMurdoEpifauna.xlsx',
        format: xlsx,
        sheet: animals,
        headers: 1,
  - run: stream_remote_resources
    cache: True
    parameters:
      resources: [mcmurdo_epifauna]
  - run: set_types
    cache: True
    parameters:
      resources: [mcmurdo_epifauna]
      types:
        year: {type: number}
        site: {type: string}
        lat_long: {type: string}
        genus_species: {type: number}
  - run: bcodmo_pipeline_processors.add_schema_metadata
    cache: True
    parameters:
      resources: [mcmurdo_epifauna]
      missingValues: ["nd"]
  - run: bcodmo_pipeline_processors.convert_to_decimal_degrees
    cache: True
    parameters:
      resources: [mcmurdo_epifauna]
      fields:
      - {input_field: lat_long, format: degrees-decimal_minutes, output_field: lat_converted, directional: '',
        pattern: "(?P<degrees>.*)\xB0 (?P<decimal_minutes>.*)'(?P<directional>.)\\ .*\xB0 .*'."}
  - run: bcodmo_pipeline_processors.convert_to_decimal_degrees
    cache: true
    parameters:
      resources: [mcmurdo_epifauna]
      fields:
      - {input_field: lat_long, format: degrees-decimal_minutes, output_field: long_converted, directional: '',
        pattern: ".*\xB0 .*'. (?P<degrees>.*)\xB0 (?P<decimal_minutes>.*)'(?P<directional>.)"}
  - run: bcodmo_pipeline_processors.round_fields
    cache: True
    parameters:
      resources: [mcmurdo_epifauna]
      fields:
      - {digits: 5, name: lat_converted}
  - run: bcodmo_pipeline_processors.round_fields
    cache: True
```

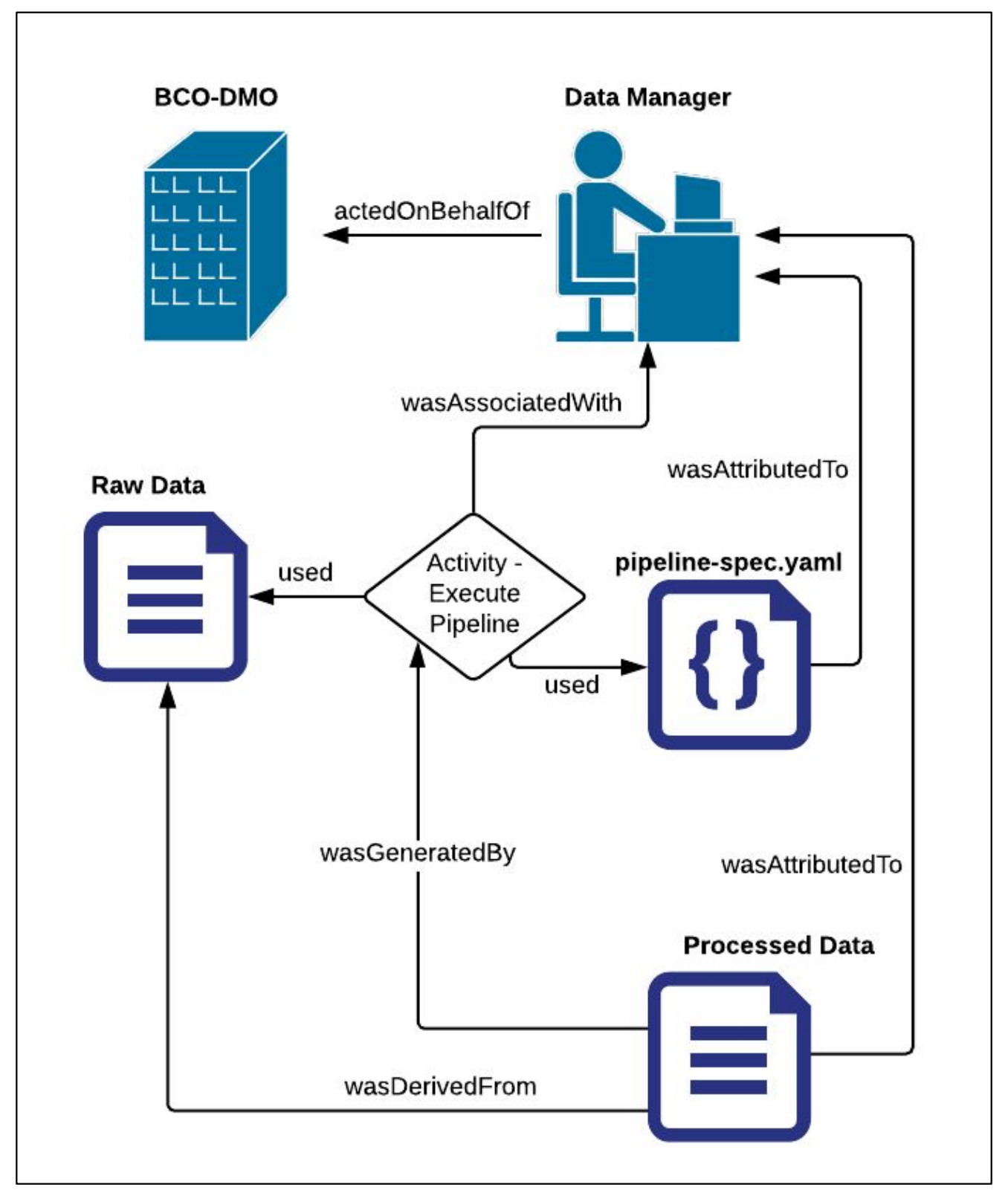
Read more: blog.bco-dmo.org/2020/02/09/frictionless-data-pipelines-for-ocean-science

Open Knowledge Foundation

for specs: **frictionlessdata.io/**
for tooling: **github.com/frictionlessdata/**

With support from Alfred P. Sloan Foundation Google.org open data institute

Elevating pipelines: From Pipeline to PROV



In the BCO-DMO Knowledge Graph, this provenance enables:

1) transparency across data curation process

2) speed, consistency across DM staff

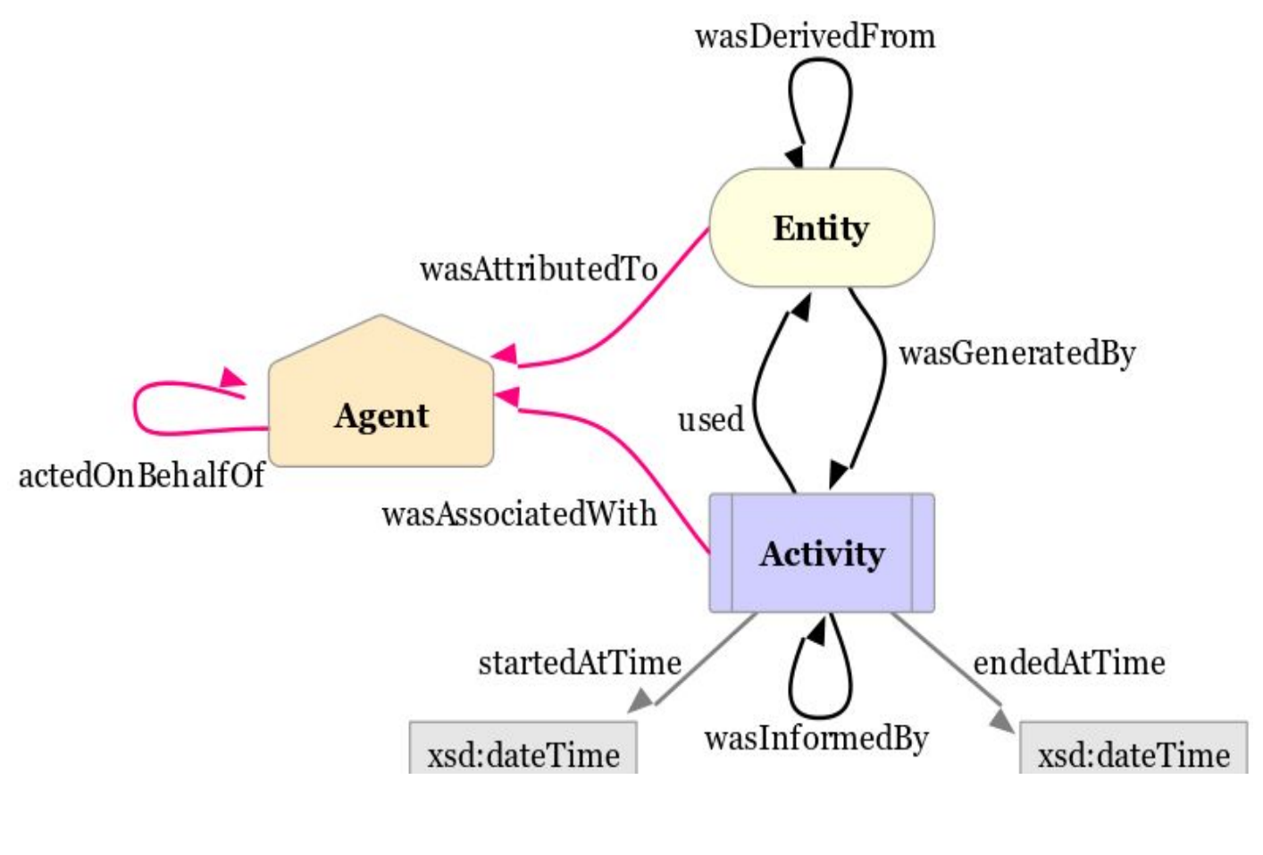
3) query & quality control
*ex: **List all datasets that used the "convert_to_decimal_degrees" processor***

See a provenance record (RDF-Turtle)

Click the AWS lambda to see Python code converts Pipeline to PROV at:
github.com/BCODMO/dpp2prov

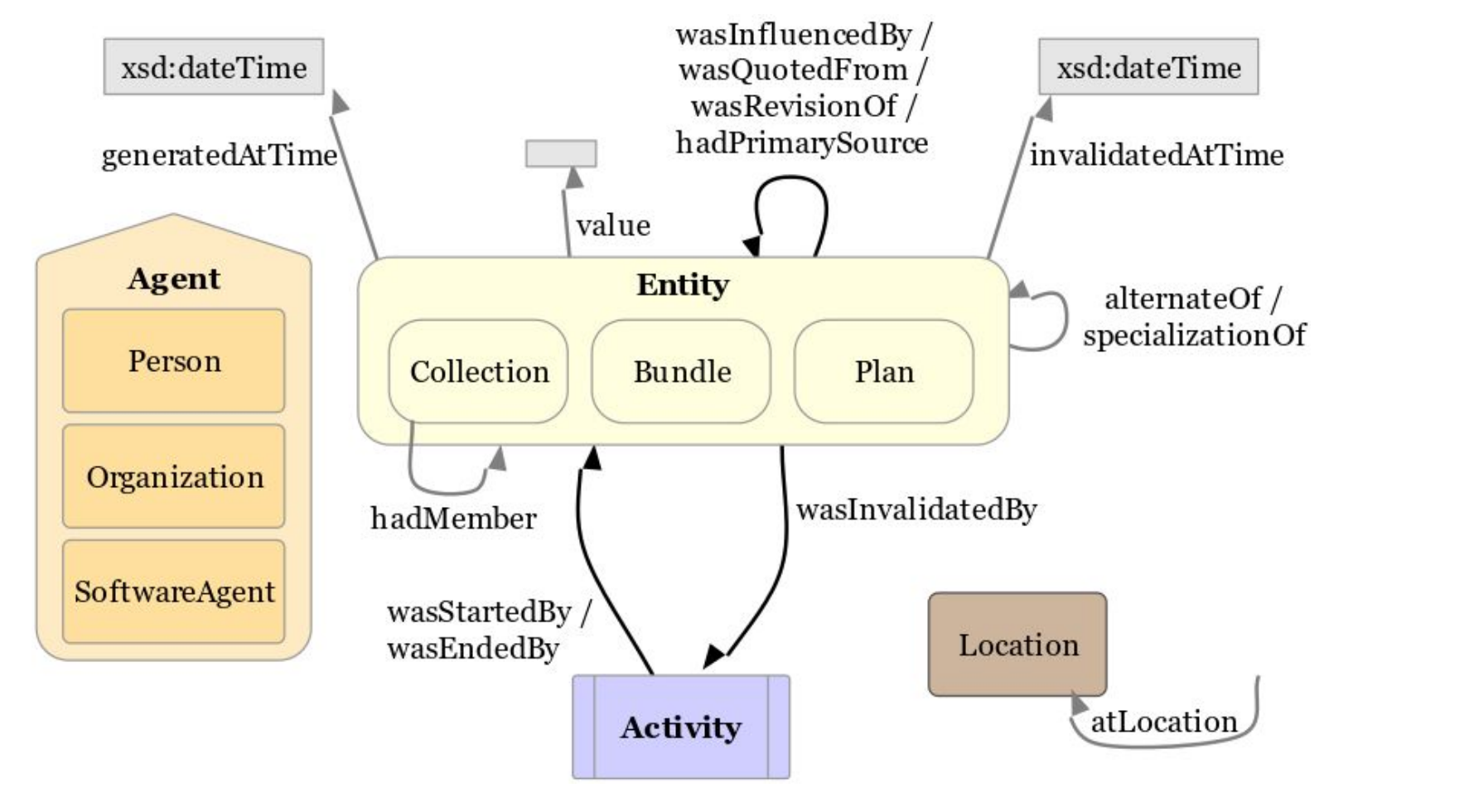
Machine readable provenance

PROV Data Model



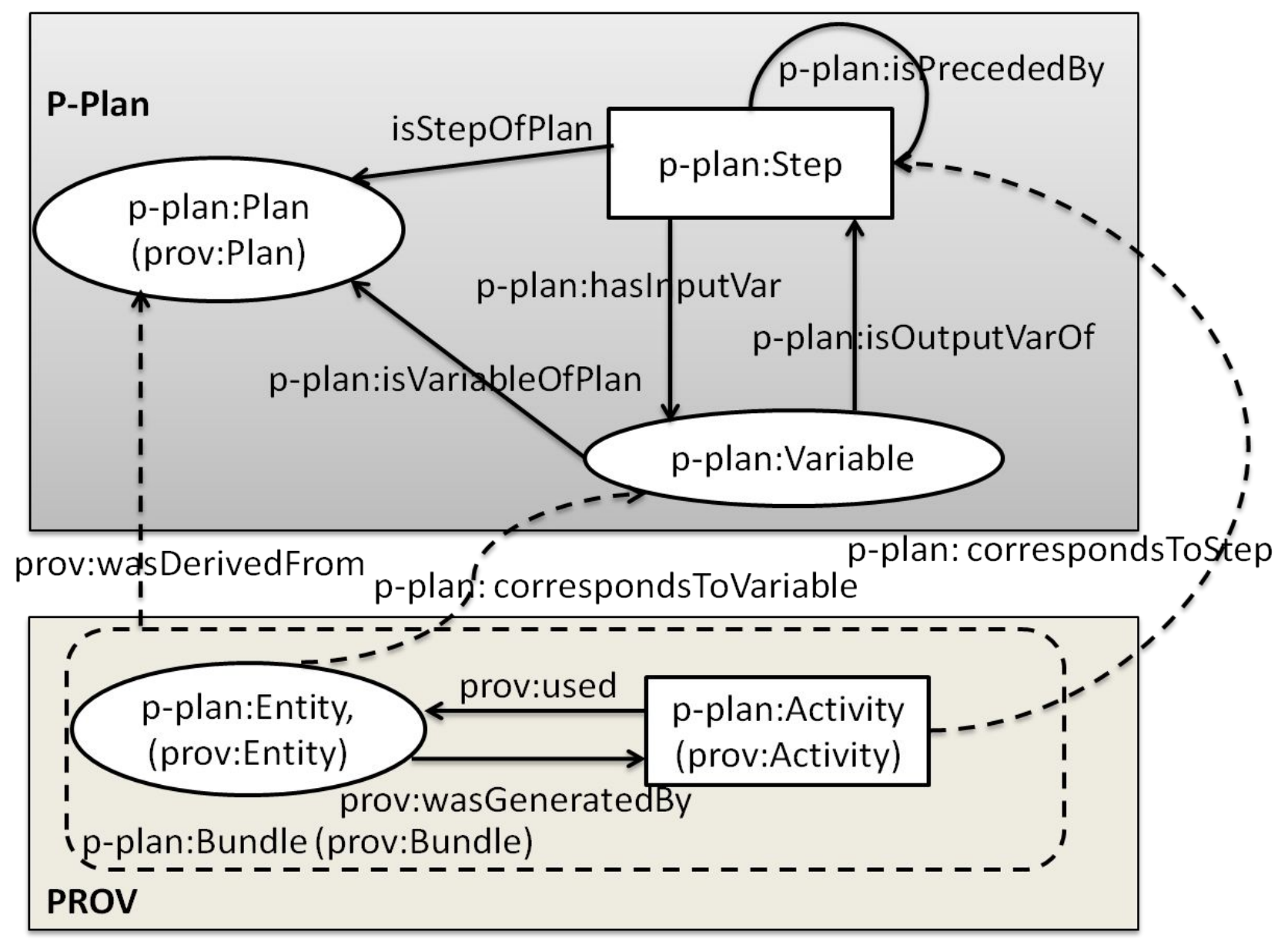
Courtesy of https://www.w3.org/TR/prov-o/

PROV-O Extended Data Model



Courtesy of https://www.w3.org/TR/prov-o/

Ontology for Provenance & Plans (P-Plan)



Courtesy of http://purl.org/net/p-plan#

WOODS HOLE OCEANOGRAPHIC INSTITUTION

funding provided by
National Science Foundation
Award# 1924618