BCO-DMO
Biological & Chemical Oceanography Data Management Office

Towards Capturing Data Curation Provenance using Frictionless Data Package Pipelines

Poster# 2
Oct 10-12
FORCE2018
Montreal, Canada

Adam Shepherd, Woods Hole Oceanographic Institution | **Conrad Schloer**, Woods Hole Oceanographic Institution | **Amber York**, Woods Hole Oceanographic Institution | **Danie Kinkade**, Woods Hole Oceanographic Institution

Abstract

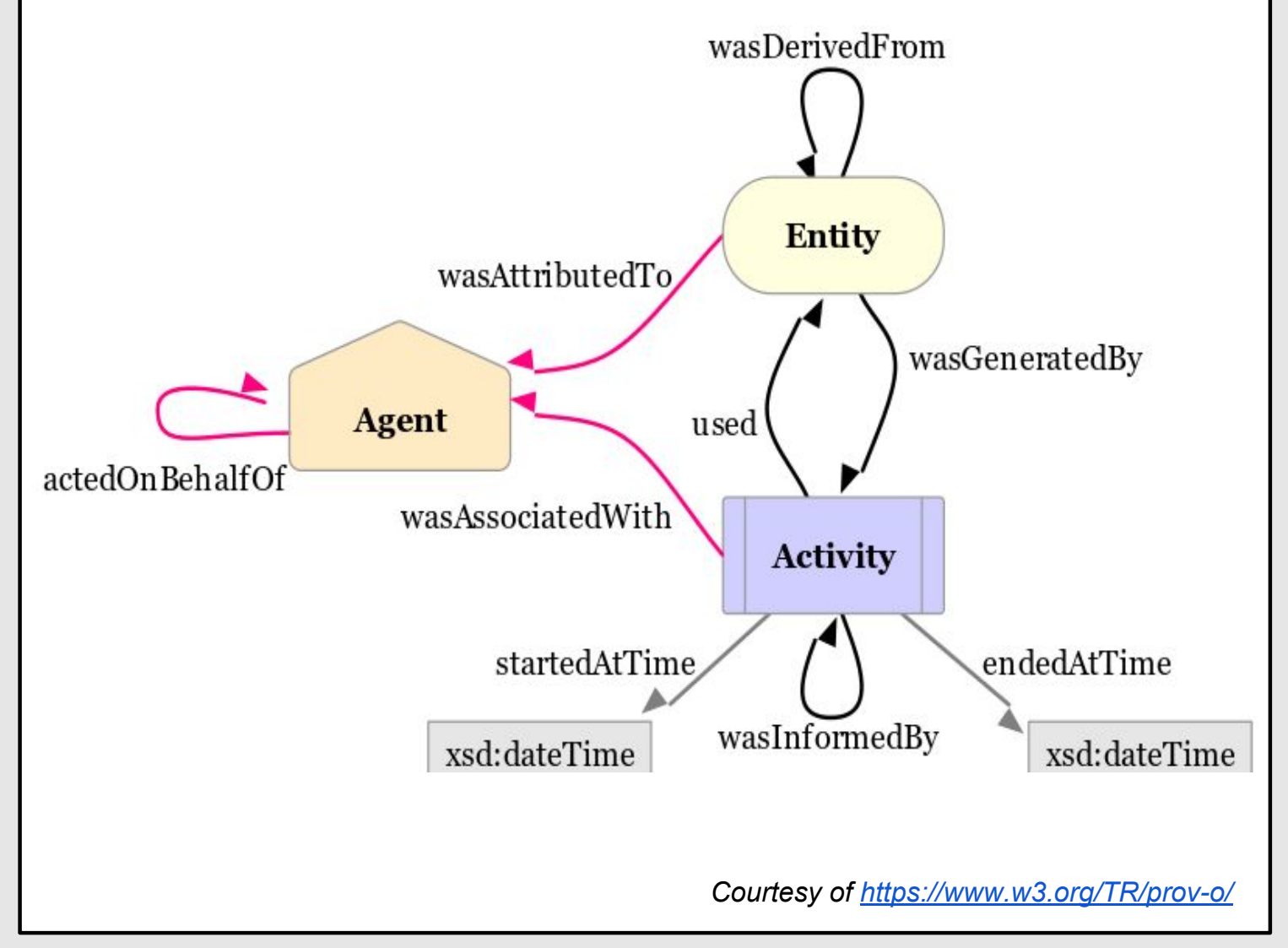
At domain-specific data repositories, curation that strives for FAIR principles often entails transforming data submissions to improve understanding and reuse. The Biological and Chemical Oceanography Data Management Office (BCO-DMO, https://www.bco-dmo.org) has been adopting the data containerization specification of the Frictionless Data project (https://frictionlessdata.io) in an effort to improve its data curation process efficiency. In doing so, BCO-DMO has been using the Frictionless Data Package Pipelines library (https://github.com/frictionlessdata/datapackage-pipelines) to define the processing steps that transform original submissions to final data products. Because these pipelines are defined using a declarative language they can be serialized into formal provenance data structures using the Provenance Ontology (PROV-O, https://www.w3.org/TR/prov-o/). While there may still be some curation steps that cannot be easily automated, this method is a step towards reproducible transforms that bridge the original data submission to its published state in machine-actionable ways that benefit the research community through transparency in the data curation process.

What is BCO-DMO ?

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 Biological and Chemical Oceanography Sections, the Division of Polar Programs Arctic Sciences and Antarctic Organisms & Ecosystems Program at the U.S. National Science Foundation.

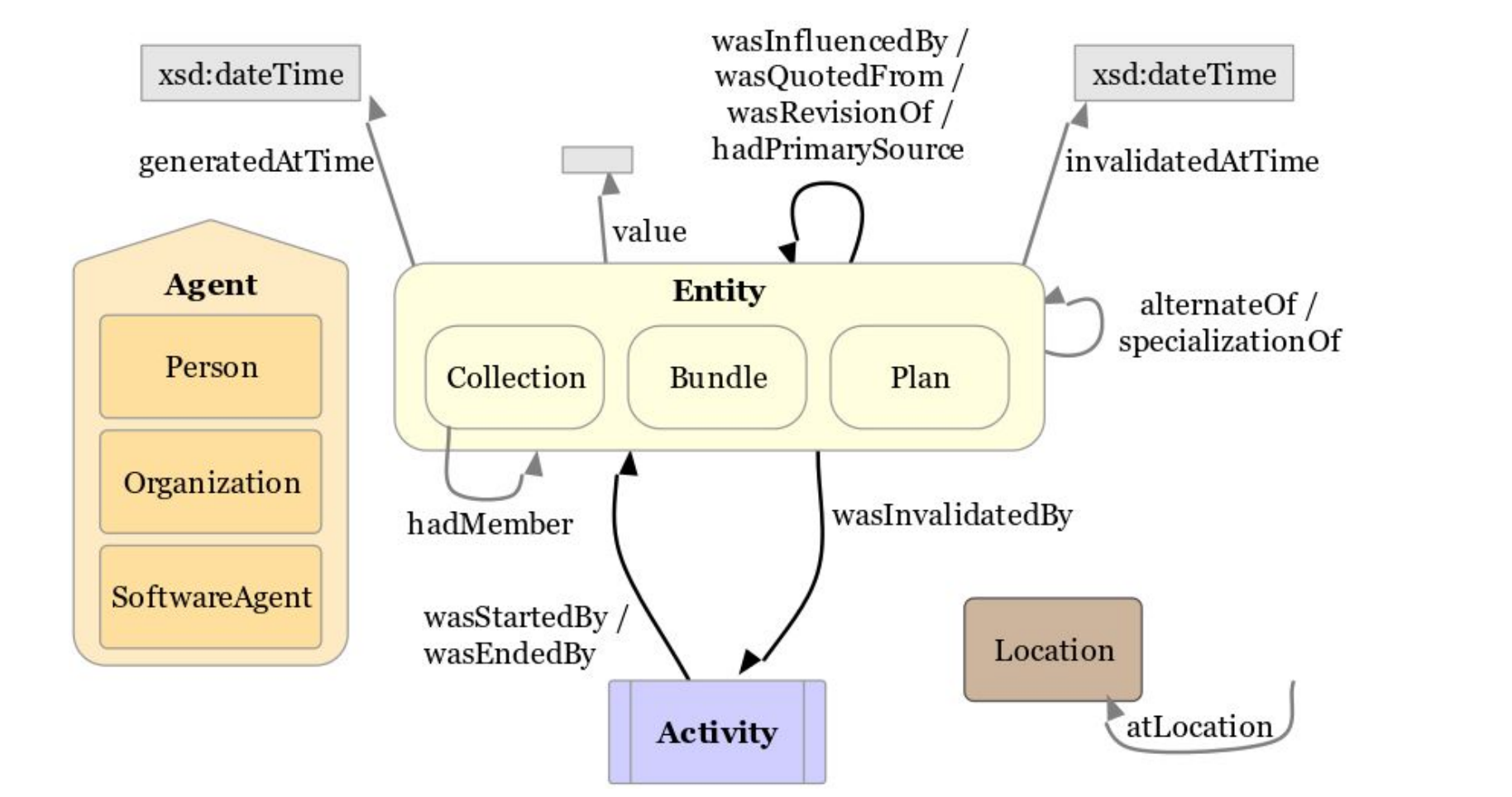
The goal of this partnership is to effectively curate marine ecosystem data and accompanying documentation, facilitating efficient data discovery and re-use. Throughout the process, BCO-DMO provides services that support specific phases of the data life cycle. The result is a rich database of research-ready data spanning the full range of marine ecosystem related measurements including: in situ 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.

PROV Data Model

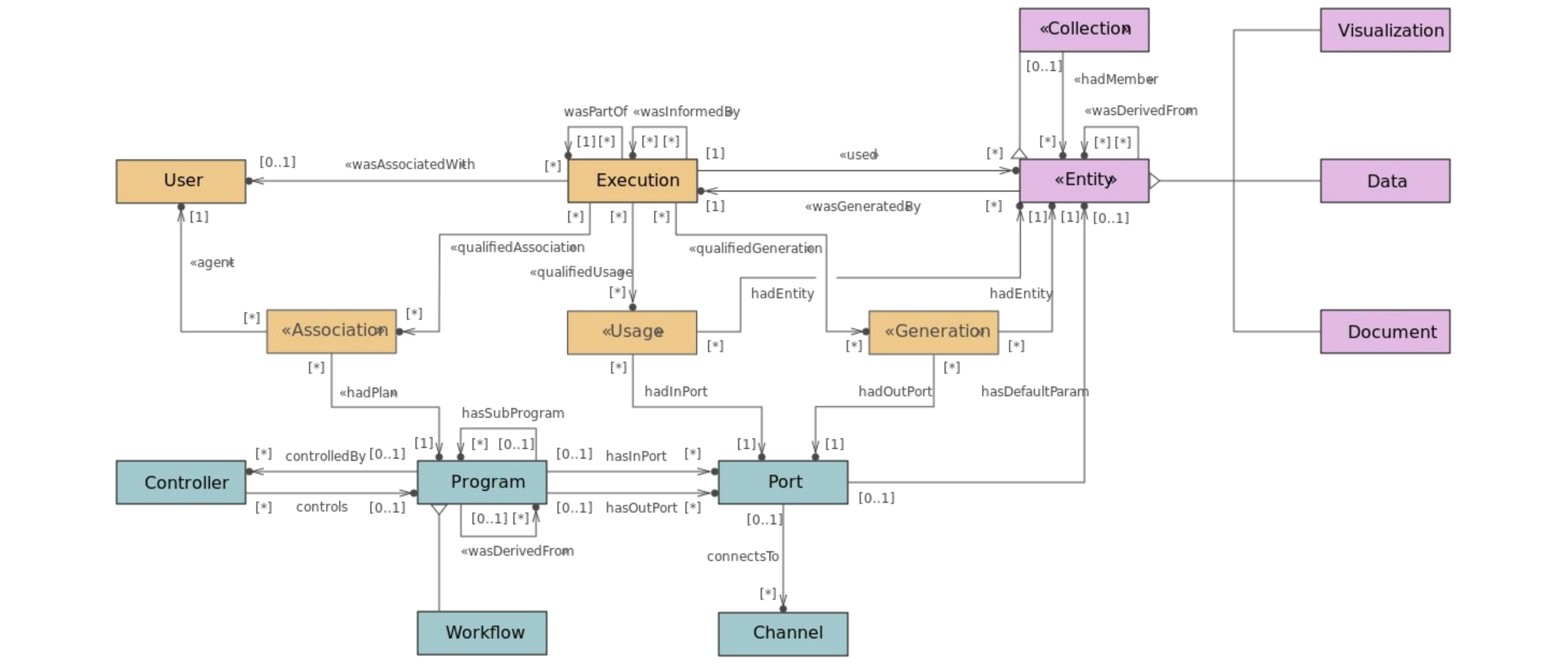


Courtesy of https://www.w3.org/TR/prov-o/

PROV-O Extended Data Model



PROVONE Data Model



Courtesy of https://purl.dataone.org/provone-v1-dev

funding provided by
National Science Foundation
Award# 1435578

Why Provenance?

- Transparency about how archived data differs from originally submitted version
- Empirical evidence of why domain-specific data management are needed for FAIR-*ness*

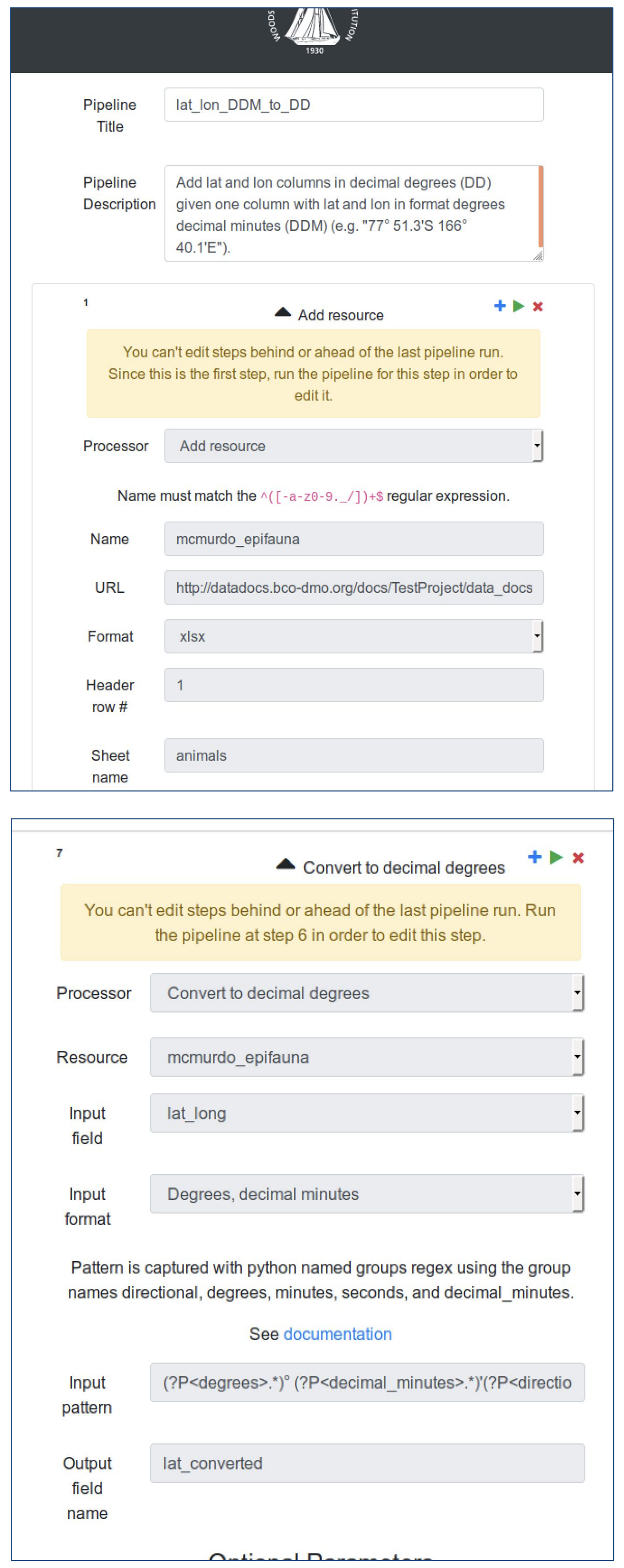
Frictionlessdata Datapackage Pipelines

Framework for building workflows for processing data.

- A pipeline is a list of processing steps for a datapackage.
- Processing steps are defined in a declarative way using YAML.

https://github.com/frictionlessdata/datapackage-pipelines

BCO-DMO UI for Frictionlessdata Datapackage Pipelines



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.

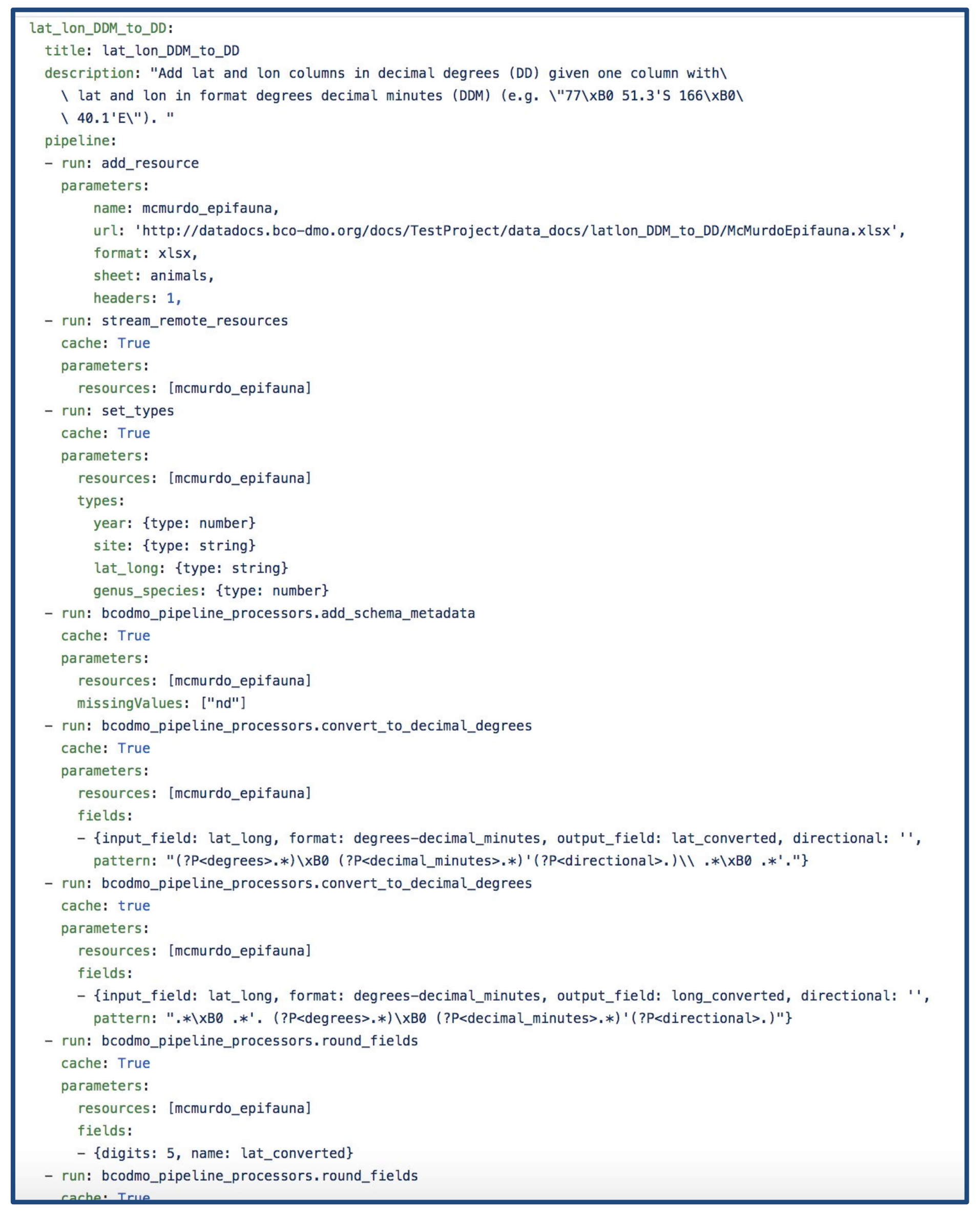
See documentation

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
- **Manages where data is stored** across the system
- **Generates a workflow:** ***pipeline-spec.yaml***

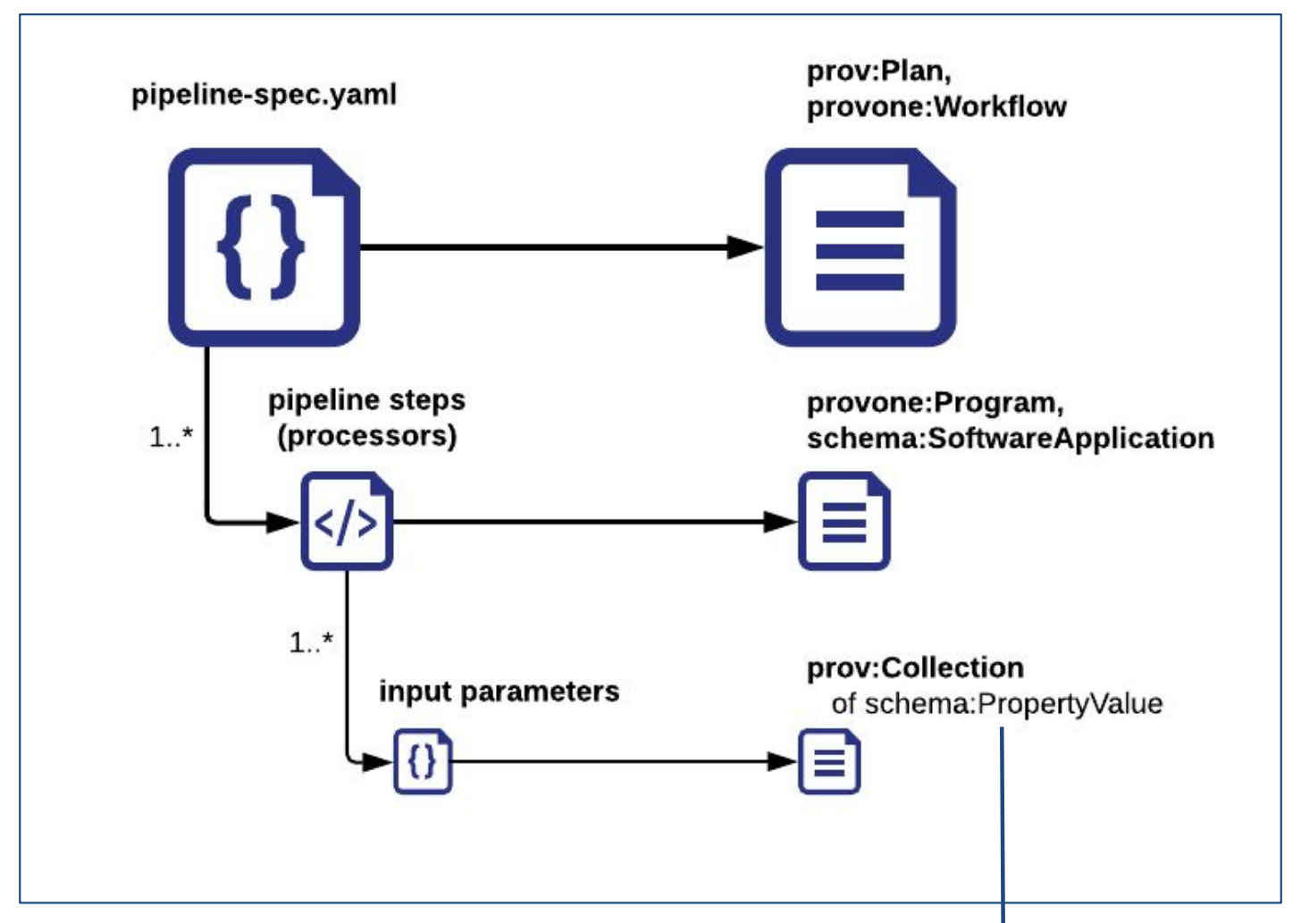


```
lat_lon_DDM_to_DD:
  title: lat_lon_DDM_to_DD
  description: "Add lat and lon columns in decimal degrees (DD) given one column with\
    \ lat and lon in format degrees decimal minutes (DDM) (e.g. \"77\xB0 51.3'S 166\xB0\
    \ 40.1'E\"). "
  pipeline:
  - run: add_resource
    parameters:
        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
```

for specs: **https://frictionlessdata.io/**
for tooling: **https://frictionlessdata.io/software/**

OPEN KNOWLEDGE INTERNATIONAL With support from Alfred P. Sloan Foundation Google.org ODI open data institute

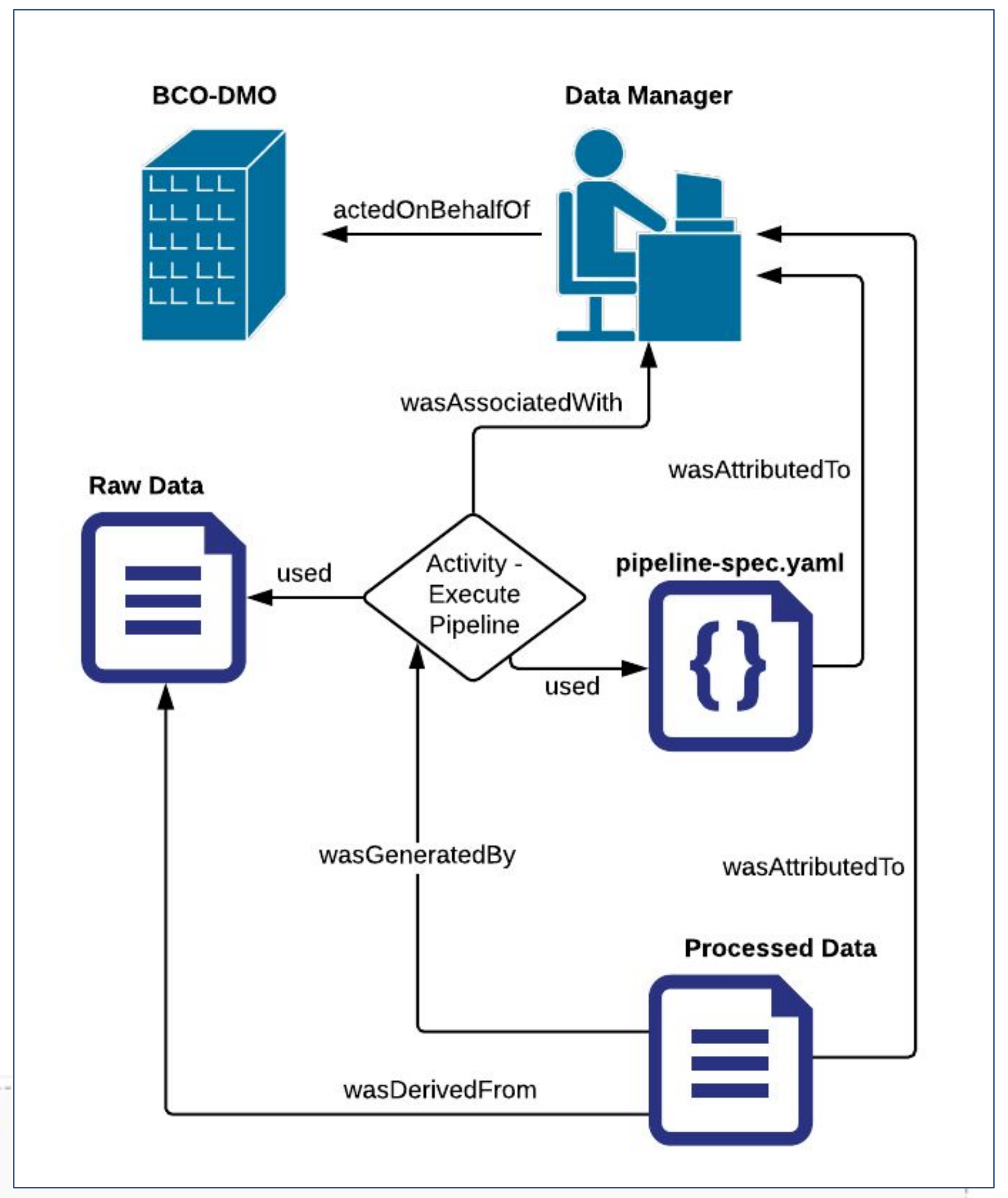
From Pipeline to PROV


pipeline-spec.yaml
prov:Plan, provone:Workflow
1..*
pipeline steps (processors)
provone:Program, schema:SoftwareApplication
1..*
input parameters
prov:Collection of schema:PropertyValue

```
Sub propertyValueForParameter(parameter)
  prop_value = new PropertyValue(parameter.name)

  If parameter.value is an Object
    prop_value = propertyValueForParameter(parameter.value)
  Else
    prop_value = parameter.value

  Return prop_value
End Sub
```



```
@prefix : <http://data.example.org/id/dataset/1234/v1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix provone: <http://purl.dataone.org/provone/2015/01/15/ontology#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

: a prov:Bundle,
        prov:Entity ;
    prov:generatedAtTime "2018-09-21T13:38:10+00:00"^^xsd:dateTime ;
    prov:wasAttributedTo :alice .

:frictionless-data-pkg a schema:DigitalDocument,
        prov:Data,
        prov:Entity ;
    schema:encodingFormat "application.vnd.datapackage+json"^^xsd:string ;
    schema:url "https://example.org/dataset/1234/v1/datapackage.json"^^xsd:anyURI ;
    prov:qualifiedGeneration [ a prov:Generation ;
            prov:activity :executed-pipeline ;
            prov:endTime "2018-09-21T13:38:10+00:00"^^xsd:dateTime ;
            prov:startTime "2018-09-21T13:37:53+00:00"^^xsd:dateTime ] ;
    prov:wasGeneratedBy :executed-pipeline .

:processed-data a schema:Dataset,
        prov:Entity ;
    schema:distribution [ a schema:DataDownload ;
            schema:contentUrl "https://example.org/dataset/1234/v1/McMurdoEpifauna.csv"^^xsd:anyURI ;
            schema:encodingFormat "text/csv"^^xsd:string ] ;
    prov:hadPrimarySource :raw-data ;
    prov:qualifiedGeneration [ a prov:Generation ;
            prov:activity :executed-pipeline ;
            prov:endTime "2018-09-21T13:38:09+00:00"^^xsd:dateTime ;
            prov:startTime "2018-09-21T13:37:54+00:00"^^xsd:dateTime ] ;
    prov:wasDerivedFrom :pipeline-spec ;
    prov:wasGeneratedBy :executed-pipeline .

:step-1-add-resource a provone:Program,
        prov:Entity ;
    schema:supportingData :step-1-add-resource-inputs .

:step-1-add-resource-inputs a schema:DataFeed ;
    schema:dataFeedElement [ a prov:Collection ;
            rdfs:comment "A single step in pipeline."@en-US ;
            prov:hadMember [ a provone:Data,
                        schema:PropertyValue,
                        prov:Entity ;
                    schema:name "run"^^xsd:string ;
                    schema:value "add_resource"^^xsd:string ],
                [ a provone:Data,
                        schema:PropertyValue,
                        prov:Entity ;
                    schema:name "parameters"^^xsd:string ;
                    schema:value [ a schema:PropertyValue ;
                            schema:name "headers"^^xsd:string ;
                            schema:value 1 ],
                        [ a schema:PropertyValue ;
                            schema:name "name"^^xsd:string ;
                            schema:value "mcmurdo_epifauna"^^xsd:string ],
                        [ a schema:PropertyValue ;
                            schema:name "url"^^xsd:string ;
                            schema:value "https://example.org/dataset/1234/original/20180921T123456Z/McMurdoEpifauna.xlsx"^^:
                        [ a schema:PropertyValue ;
                            schema:name "format"^^xsd:string ;
                            schema:value "xlsx"^^xsd:string ],
                        [ a schema:PropertyValue ;
                            schema:name "sheet"^^xsd:string ;
                            schema:value "animals"^^xsd:string ] ] ] .
```