

Capturing Provenance of Data Curation at BCO-DMO

...

Adam Shepherd
Amber York



1977 Star Wars



TWENTIETH CENTURY-FOX Presents A LUCASFILM LTD. PRODUCTION **STAR WARS**
Starring **MARK HAMILL HARRISON FORD CARRIE FISHER**
PETER CUSHING

and
ALEC GUINNESS

Written and Directed by **GEORGE LUCAS** Produced by **GARY KURTZ** Music by **JOHN WILLIAMS**

Making Films Sound Better
DD DOLBY SYSTEM[®]
Noise Reduction High Fidelity

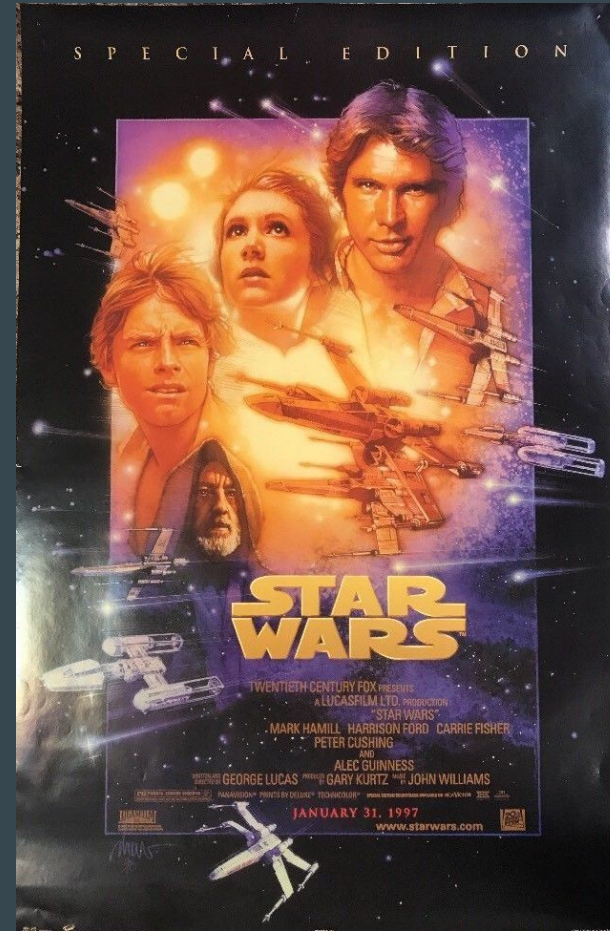
PANAVISION[®] PRINTS BY DE LUXE[®] TECHNICOLOR[®]

Original Motion Picture Soundtrack on 20th Century Records and Tapes

STAR WARS

ONE SHEET 27x41.1" C

1997 Star Wars



STAR WARS

TWENTIETH CENTURY FOX PRESENTS
A LUCASFILM LTD. PRODUCTION

"STAR WARS"

MARK HAMILL HARRISON FORD CARRIE FISHER
PETER CUSHING

AND

ALEC GUINNESS

WRITTEN BY GEORGE LUCAS PRODUCED BY GARY KURTZ MUSIC BY JOHN WILLIAMS

CASTING BY JUDITH ANNE EASTMAN COSTUME DESIGNER JUDITH ANNE EASTMAN EDITOR JUDITH ANNE EASTMAN EXECUTIVE PRODUCERS JUDITH ANNE EASTMAN JUDITH ANNE EASTMAN

JANUARY 31, 1997

www.starwars.com





Did Han Shoot First?



From the Author:

"To me, [the original movie] doesn't really exist anymore. ...

I'm sorry you saw half a completed film and fell in love with it.

But I want it to be the way I want it to be."

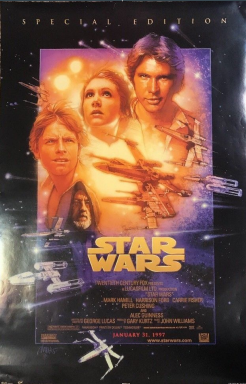
- George Lucas, 2004 interview

1997: A Disturbance in the Force

1977



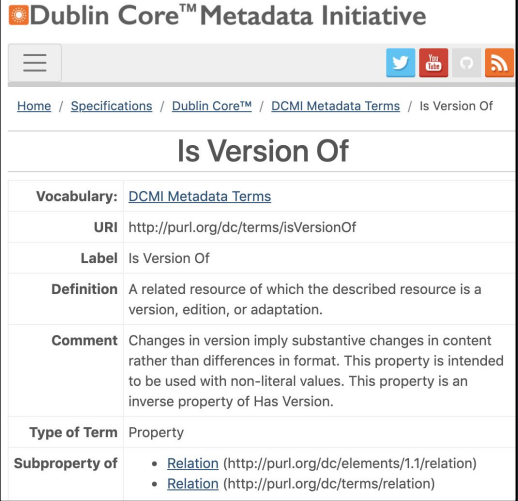
1997



What does this mean for Data?

Philosophically, no issue with *versions* of a creative work.

[dcterms:isVersionOf](#)



The screenshot shows the Dublin Core Metadata Initiative website. The page title is "Is Version Of". The breadcrumb trail is "Home / Specifications / Dublin Core™ / DCMI Metadata Terms / Is Version Of". The page content is a table with the following rows:

Vocabulary:	DCMI Metadata Terms
URI	http://purl.org/dc/terms/isVersionOf
Label	Is Version Of
Definition	A related resource of which the described resource is a version, edition, or adaptation.
Comment	Changes in version imply substantive changes in content rather than differences in format. This property is intended to be used with non-literal values. This property is an inverse property of Has Version.
Type of Term	Property
Subproperty of	<ul style="list-style-type: none">• Relation (http://purl.org/dc/elements/1.1/relation)• Relation (http://purl.org/dc/terms/relation)

How do we explain *what happened* to those who don't know yet?

A Case of 'st1_50m'



An observation made at the location of station '1' at a depth of 50 meters

best_hit_annotation	best_hit_taxon_id	st1_050m	st1_090m	st1_120m	st1_200m	st1_300m	st1_400m	st1_600m	st3_040m	st3_060m	st3_120m	st3_180m
	247490	0	0	0	0	122	121	116	0	0	17	
actase subunit beta (E	330214	0	0	0	1	136	173	153	0	0	18	
.9); K04077 chaperon	167546	80	91	59	35	2	2	1	60	44	24	
CsoS1	167555	155	162	94	38	0	0	0	54	40	39	
substrate binding prot	167546	202	203	169	158	26	30	19	100	86	27	
family	859653	7	7	8	7	51	69	74	3	1	19	
	314261	17	20	19	9	30	35	36	12	15	22	
g protein; family 5	89187	0	0	0	0	50	50	65	0	0	2	
mate--ammonia liga	146891	62	60	54	58	3	4	2	60	53	4	
rate-binding protein	913324	3	2	2	4	33	78	43	0	0	6	
	93058	57	63	76	34	5	4	3	39	40	47	
mic substrate-bindin	375451	0	2	3	0	41	48	44	0	0	6	
spargine ABC trans	488538	2	7	11	2	31	52	44	2	3	5	
	1090946	1	1	7	0	51	47	37	0	0	5	
	859653	88	68	33	29	1	4	3	38	32	10	
oxylase; K01601 ribu	146891	46	57	40	36	1	3	0	37	41	15	
	1073573	20	10	2	1	26	44	29	10	8	2	
alpha (EC:1.2.1.2); KO	639282	0	0	0	0	37	41	36	0	0	3	
rate-binding protein,	644966	0	0	0	0	29	38	31	0	0	1	
family	859653	25	16	15	9	50	55	41	12	14	19	
spargine ABC trans	488538	0	14	18	0	36	45	33	0	0	22	
	1073573	4	5	1	2	35	27	35	0	0	3	

Putting the 'F-I-R' in FAIR

F

Findable

- get all datasets that recorded 'station' or 'depth'

I

Interoperable

- linked to community vocabs

R

Reusable

- *metadata with provenance*

How 'st1_50m' and all 'st(#)_(#m'
became 'station' and 'depth'

	station	depth	spectral_count
6	8	200	0
6	9	40	12
6	9	70	4
6	9	380	0
L	12	40	0
L	12	120	0
L	12	300	0
4	1	50	0
4	1	90	0
4	1	120	0
4	1	200	0
4	1	300	0
4	1	400	0
4	1	600	0
6	3	40	9
6	3	60	0

A Tale of Two Versions

'st(#)_(#)'m'

original
data

original
Data



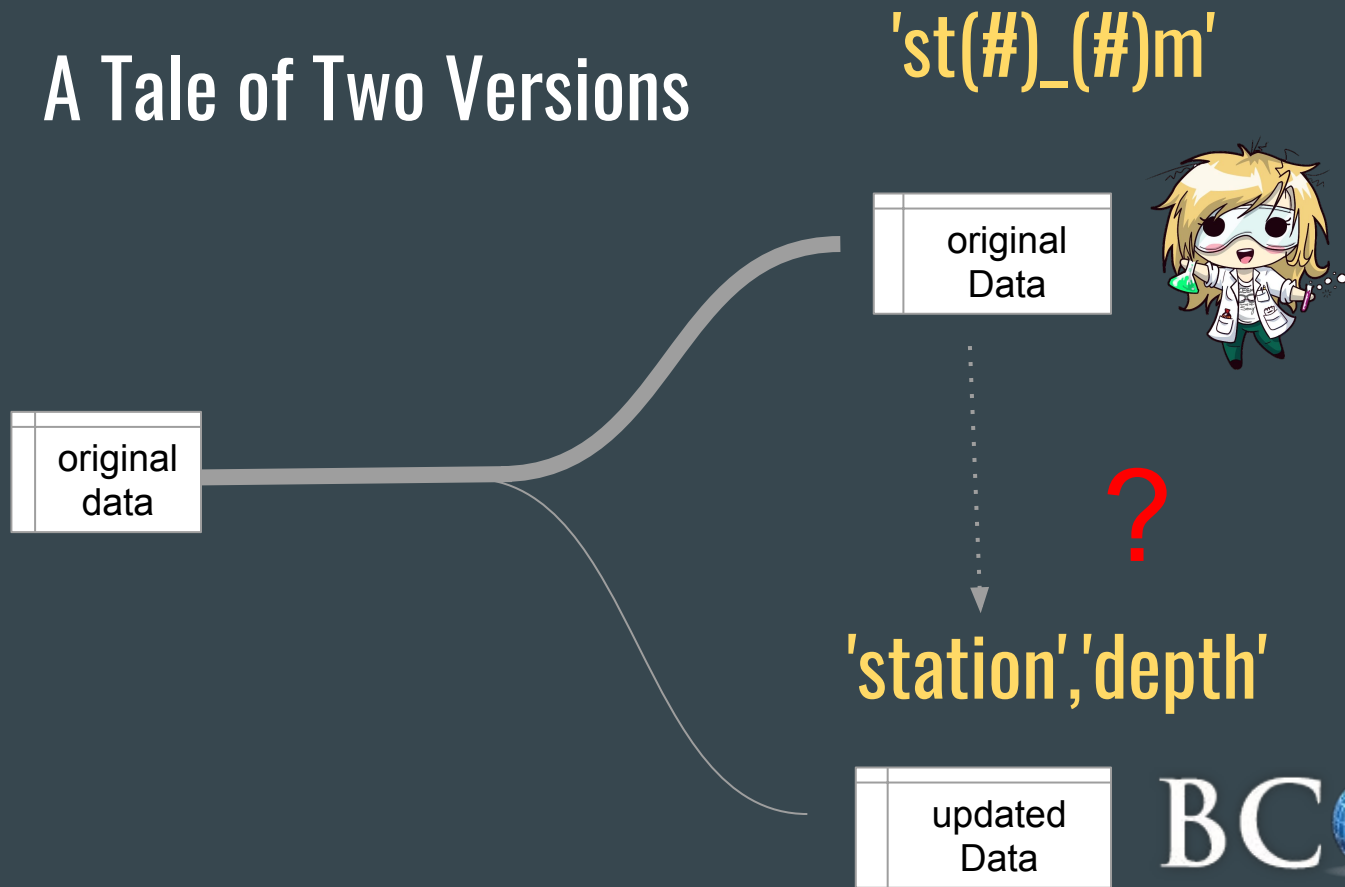
'station','depth'

updated
Data



Biological & Chemical Oceanography Data Management Office

A Tale of Two Versions



How do we explain what happened to those who don't know?

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)

How do we explain what happened
to those who don't know?

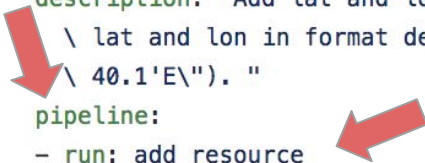
Declarative Workflows *Over Ad-Hoc Scripting*

Declarative Workflows - A set of steps to execute

```
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,
```

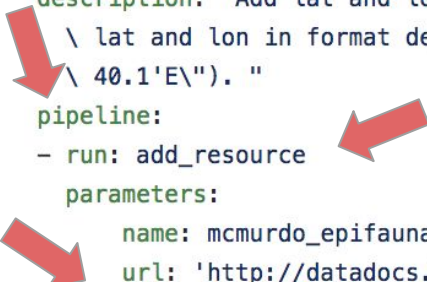
Declarative Workflows - Each step is "named"

```
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,
```



Declarative Workflows - Each step has inputs

```
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,
```



Declarative Workflows - More steps

```
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
```

```
    paramet
```

```
      nam
```

```
      url
```

```
      for
```

```
      she
```

```
      hea
```

```
  - 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>.)"}
```

Declarative Workflows - Names identify code to execute

```
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: ad
```

```
    paramet
```

```
      nam
```

```
      url
```

```
      for
```

```
      she
```

```
      hea
```

```
  - 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]
```

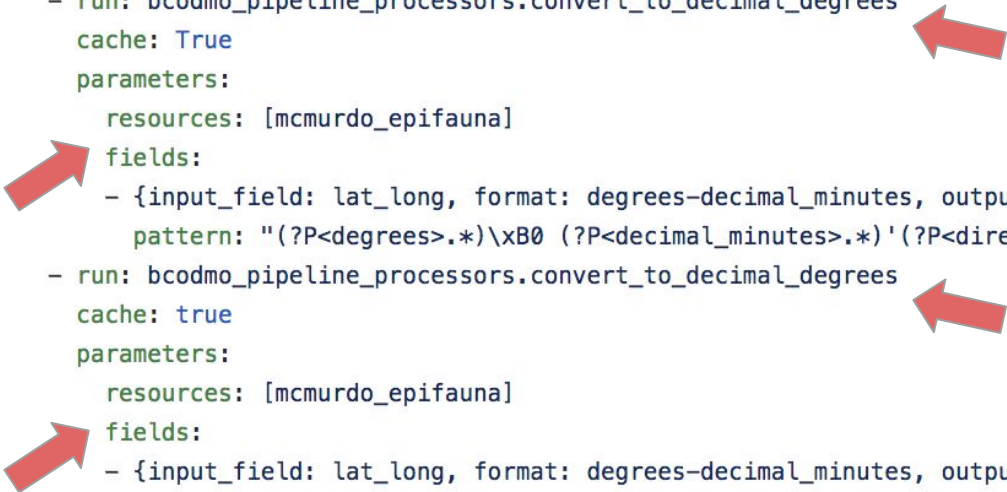
```
      fields:
```

```
      - {input_field: lat_long, format: degrees-decimal_minutes, output_field: long_converted, directional: '',  
        pattern: ".*\xB0 .*'. (?P<degrees>.*)\xB0 (?P<decimal_minutes>.*)'(?P<directional>.)"}
```

Declarative Workflows - Each step has its own inputs

```
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: ad
      parameters:
        name:
        url:
        for:
        she
        hea
    - 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>.)"}

```



Example: Convert latitude format

```
- run: bcdmo_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 .*'."}
```

```
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
  \ 48.1'E\")."
  pipeline:
  - run: add_resource
  parameters:
    name: mcmurdo_epifauna,
```

```
cache: True
parameters:
  resources: [mcmurdo_epifauna]
  missingValues: ["nd"]
- run: bcdmo_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: bcdmo_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: bcdmo_pipeline_processors.round_fields
cache: True
parameters:
  resources: [mcmurdo_epifauna]
  fields:
  - {digits: 5, name: lat_converted}
- run: bcdmo_pipeline_processors.round_fields
cache: True
```


Example: Convert latitude format

```
- run: bcdmo_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 .*'."}
```

lat_long
77° 51.1'S 166° 40'E
77° 51'S 166° 39.7'E
77° 51'S 166° 39.7'E
77° 51'S 166° 39.7'E
77° 51'S 166° 39.6'E
77° 51'S 166° 39.6'E
77° 51'S 166° 39.6'E
77° 50.9'S 166° 39.4'E

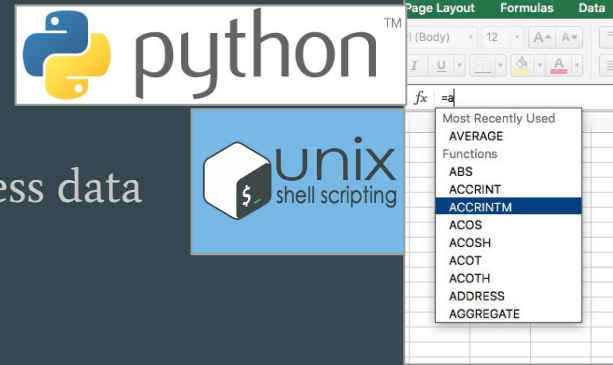


lat_dd	lon_dd
-77.855	166.6683
-77.855	166.6683
-77.855	166.6683
-77.8517	166.6667
-77.8517	166.6667
-77.8517	166.6667
-77.85	166.6617
-77.85	166.6617

```
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°x80 51.3'S 166°x80
  \ 40.1'E\"). "
pipeline:
- run: add_resource
parameters:
  name: mcmurdo_epifauna,
  mcmurdo_epifauna.xlsx',
```

```
cache: True
parameters:
  resources: [mcmurdo_epifauna]
  missingValues: ["nd"]
- run: bcdmo_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: bcdmo_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: bcdmo_pipeline_processors.round_fields
cache: True
parameters:
  resources: [mcmurdo_epifauna]
  fields:
  - {digits: 5, name: lat_converted}
- run: bcdmo_pipeline_processors.round_fields
cache: True
```

Why not use ad-hoc Scripting?



At BCO-DMO, Data Managers (DM) were writing code to process data

Challenges:

Maintenance: code over time has dependencies

- software & library updates; what breaks with software updates
- individual DM is the expert; is documentation good enough?

Redundancy: 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?

Quality Control: Ex: "What datasets that needed their coordinates converted to decimal degrees?"

Why Declarative Workflows?

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

The 'what' gives DMs a shared language for expressing their intents, *consistently*.

→ Flexibility with staffing

- ◆ easier to teach non-coders **'what'** should be done over **'how'** to do it
- ◆ any DM can step in/out of a project

github.com/BCODMO/bcodmo_frictionless#bcodmo_pipeline_processorsconvert_to_decimal_degrees

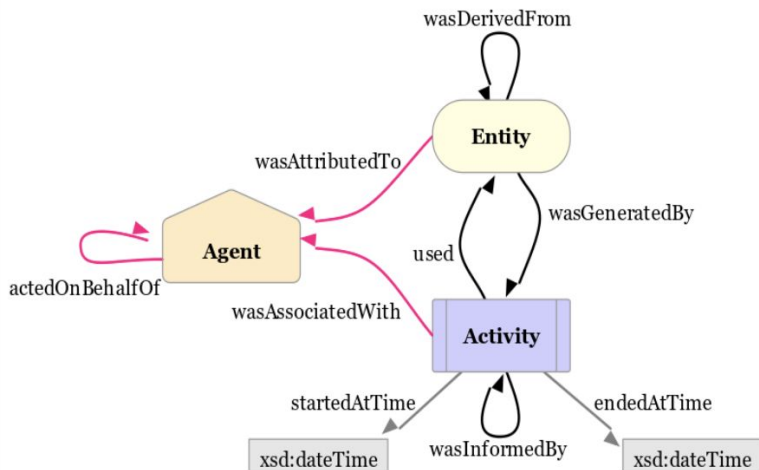
→ As configuration data,

- ◆ code interpreting DWs can be changed/swapped without impacting DM intent
- ◆ Q: All datasets that used: *'convert_to_decimal_degrees'*

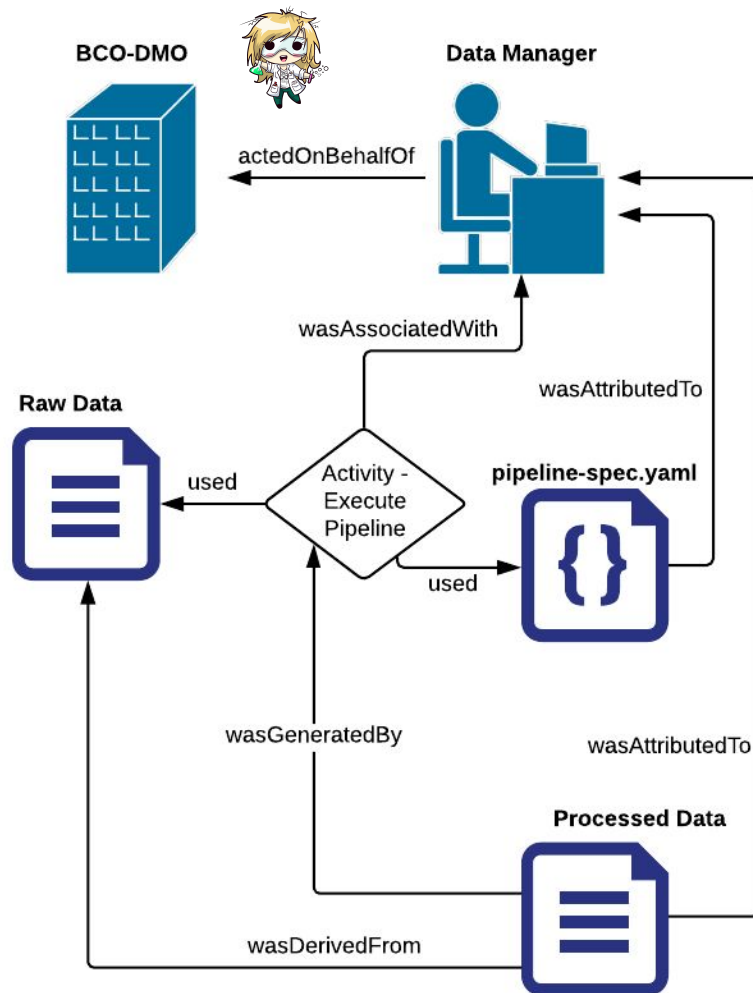
Declarative workflows can serve as **provenance**.

Workflow as PROV

PROV Data Model



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



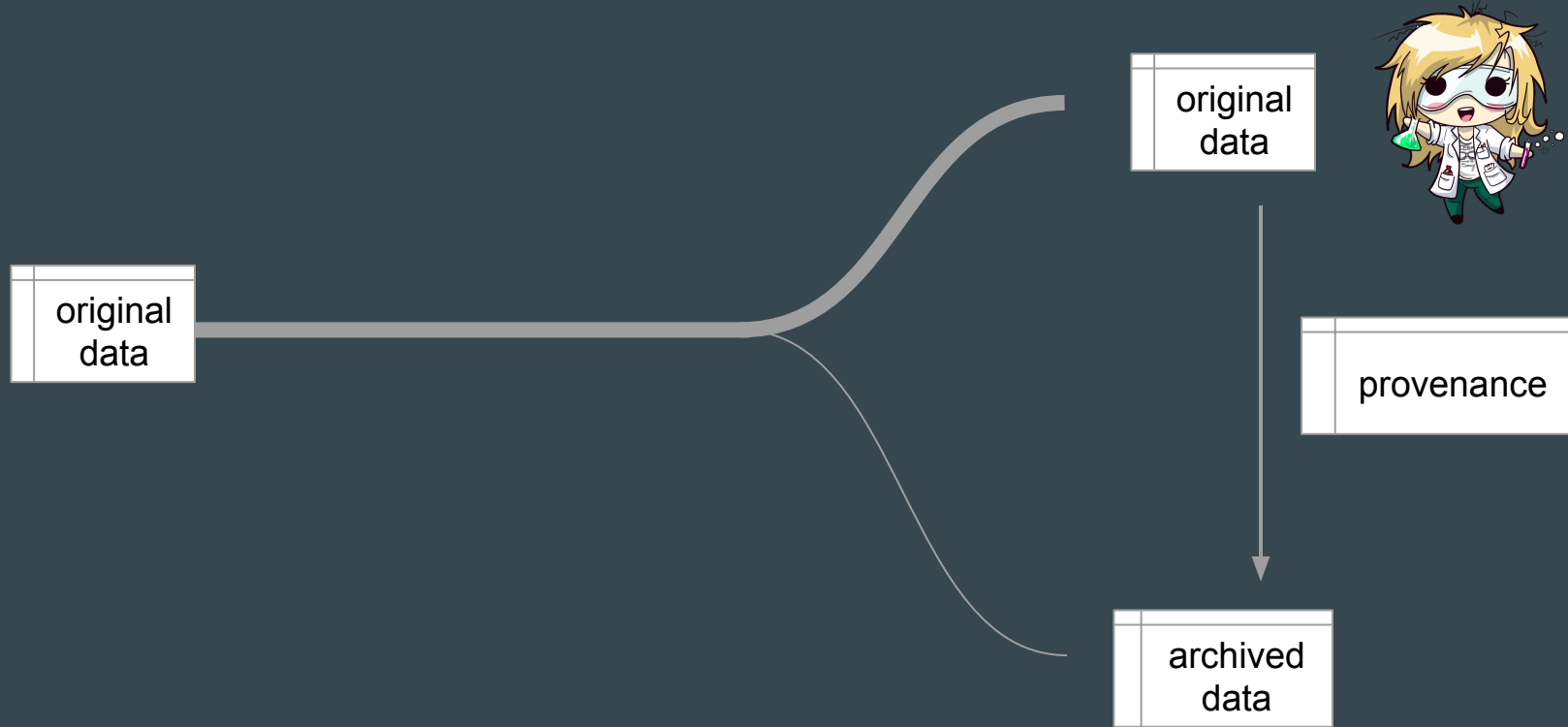
What use is this PROV?

The provenance is data describing how the archived version changed the original



What use is this PROV?

The provenance is data describing how the archived version changed the original



What use is this PROV?

The provenance is data describing how the archived version changed the original



Workflow Tools at BCO-DMO



frictionlessdata.io

pypi.org/project/dataflows

github.com/BCODMO/bcodmo_frictionless

bit.ly/bcodmo-curation-prov