NRC · CNRC

From Discovery to Innovation...

The CVO Data Model

Patrick Dowler
Canadian Astronomy Data Centre
Herzberg Institute for Astrophysics

Overview

- Design for science
  - multi-wavelength approach:
    - abstraction
    - generic data types
  - every piece of information has a provenance

- Design for use by software
  - model the content as numbers
  - model the data flow

Common Data Model

- EntryProp: unit of information (property of an entry)
  - propID
  - tupleID
  - value
  - error
  - provenance
- Entry: entryID, EntryProp[]
  - <propID, tupleID> is unique within an Entry
- Catalog: Entry[]
  - <entryID, propID, tupleID> is unique within a Catalog
- EntrySet: Entry[], ConstraintSet, Catalog

Common Data Model

- EntryProp
  - propID: number to use to lookup the name
  - tupleID: multiple values for a property
    - multiple input data
    - multiple processing techniques
  - value: structured object
    - Point2D vs. RA and DEC properties
    - Interval vs. min_x and max_x properties
  - error: overly simplistic characterisation
  - provenance: navigation
    - catalogName
    - entryID, [propID, [tupleID]] [optional]

Observation Model

- observation: a *thing* produced by an archive, *view* of universe

- astronomical *coordinate* system (axes)
  - spatial (*p*)
  - spectral (*e*)
  - temporal (*t*)
  - polarisation?

- an observation is a sample of *p,e,t*
  - *p*: usually a Polygon2D
  - *e*: 1st order approximation: Interval(Number, Number)
  - *t*: 1st order: Interval(Date, Date)

Observation Model: required properties

- data_product : ArchiveLink
- survey : String

- spatial
  - spatial_bounds
  - spatial_sample
  - spatial_fill
  - spatial_resolution
  - spatial_nbins
- spectral
  - spectral_bounds
  - spectral_sample
  - spectral_fill
  - spectral_resolution
  - spectral_nbins
- temporal
  - temporal_bounds
  - temporal_sample
  - temporal_fill
  - temporal_resolution
  - temporal_nbins

- _bounds (region sampled) : Polygon2D (spatial) or Interval
- _sample (size of one sample) : Number
- _fill (fraction of region sampled) : Number
- _resolution (size of resolution element ~PSF) : Number
- _nbins (number of bins on an axis) : Number

Observation Model: optional properties

- content characterisation
  - number_density_pt : Number
  - number_density_ext : Number
  - flux_density_pt : Number
  - flux_density_ext : Number
  - flux_density_lowf : Number
  - number_density_abs : Number
  - number_density_emi : Number
  - flux_SN10 (~depth) : Number

Process Model

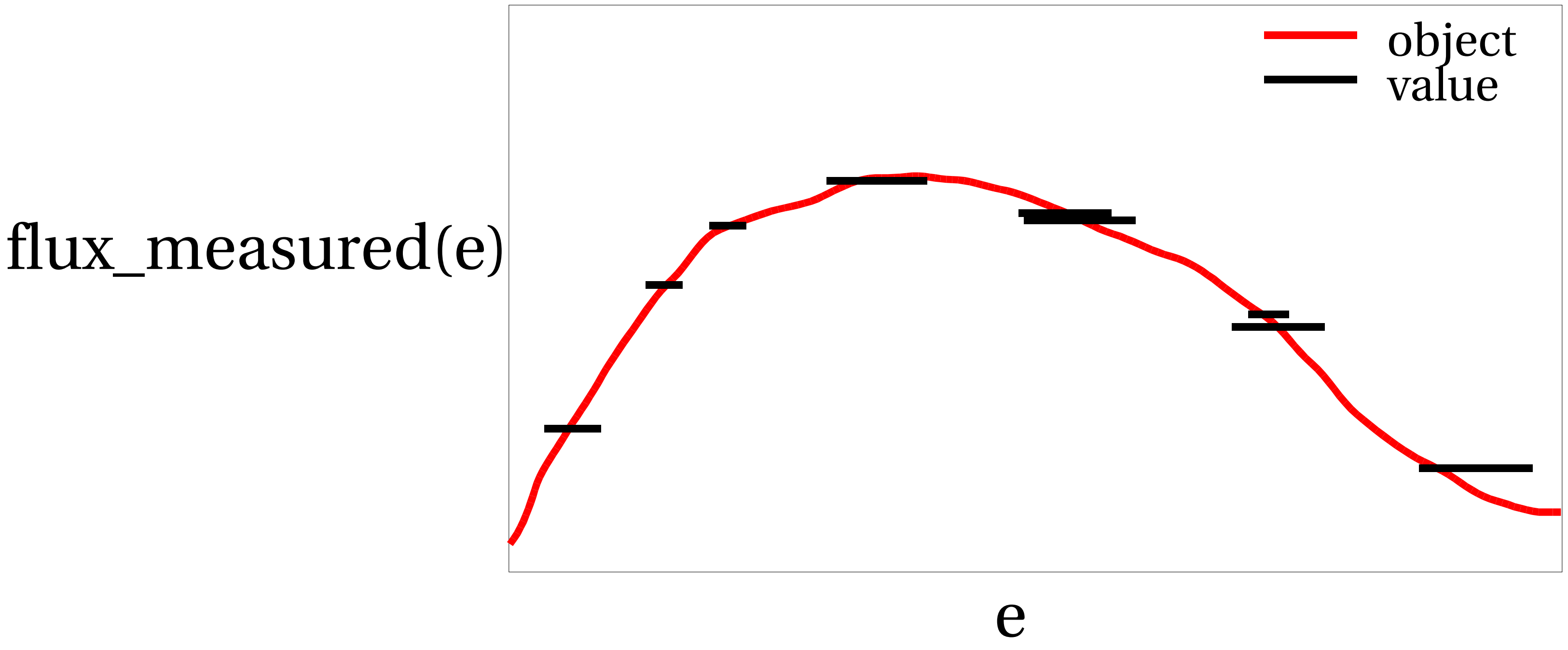
- process: a *thing* that produces some output
  - specifically, a process produces EntryProp(s)
  - they can be for new or existing entries

Source Model

- source: a *thing* in an observation
  - provenance: an analysis process
- pedantic: *f(p,e,t), g(p,e,t), ...*
- convention: factor out *p*
  - *position, f(e,t), g(e,t), ...*
- simplicity: factor out *t*
  - *position, temporal_bounds, f(e), g(e), ...*
- new required data type: unary function
  - value: Number
  - domain: Interval

Source Model

- required source properties:
  - position : Point2D
  - time_bounds : Interval
  - flux_measured(e) : Function



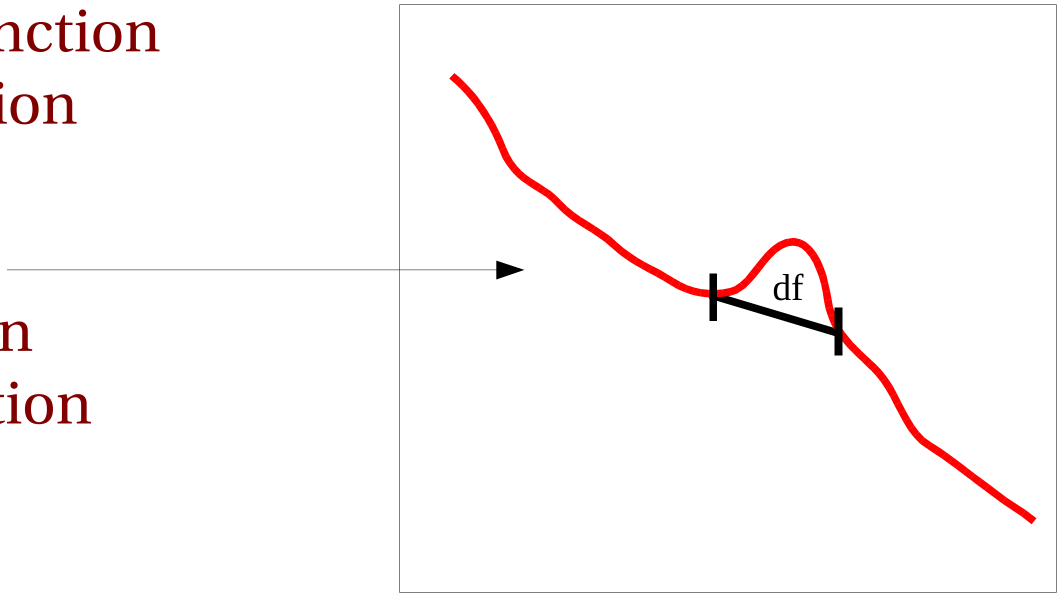
Source Model

- optional source properties
  - flux_interpolated(e) : Function
  - spectral_index(e): Function

  - delta_flux(e) : Function
  - equiv_width(e) : Function
  - velocity_width(e) : Function

  - fwhm(e) : Function
  - Kron_radius(e) : Function
  - ellipticity(e) : Function
  - pos_angle(e) : Function

  - redshift : Number



Object model

- object: a *thing* in the universe
  - provenance: the cross-matching process
    - maybe the source EntryProp that was matched

- can't really factor out *t* :-(
  - object properties derived from many observations

- can't really factor out p :-(
  - object has different size and shape at different *e*
  - object may have different position at different *e*
  - moving objects have different position at different *t*

Object Model

- new required data type: ternary function!!!
  - most object properties are *f(p,e,t)*
  - a few may be *z(p,t)* e.g. redshift

- Or, ignore those pesky details and use the source model as is

- extra properties: if you have redshift, you could shift the domain (*e*)
  - e.g. rest_flux_measured(*e*')

Object model

- required source properties:
  - flux_measured(p,e,t) : Function

Object model

- optional object properties:
  - flux_interpolated(p,e,t) : Function
  - spectral_index(p,e,t): Function

  - delta_flux(p,e,t) : Function
  - equiv_width(p,e,t) : Function
  - velocity_width(p,e,t) : Function

  - fwhm(p,e,t) : Function
  - Kron_radius(p,e,t) : Function
  - ellipticity(p,e,t) : Function
  - pos_angle(p,e,t) : Function

  - redshift(p,t)?? : Function

Summary

- same common DM for observation, process, source, and object
  - Entry: entryID, EntryProp[]
  - EntryProp: propID, tupleID, value, error, provenance

- minimal but very useful list of observation properties
  - spatial/spectral/temporal sampling: required
  - content characterisation: optional

- very minimal list of source properties
  - position, temporal_bounds, flux_measured: required
  - other properties: optional
  - spectral_bounds is the domain of a Function data type