ProvTAP status

F.Bonnarel, CDS
on behalf of M.Servillat, M.Louys, M.Nullmeier,
M.Sanguillon, L.Michel

CENTRE DE DONNÉES
ASTRONOMIQUES DE STRASBOURG

ESCAPE
European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

IVOA

Why a ProvTAP specification ?

- Provenance information can be attached to data in various ways :
  - Embedded in the data « header » itself
  - Linked to the data record via DataLink or URL
  - Retrievable via ProvSAP via data id.
- In addition to that , ProvTAP allows to discover « data » by constraining Provenance features.
  - It's a « reverse » mechanism.

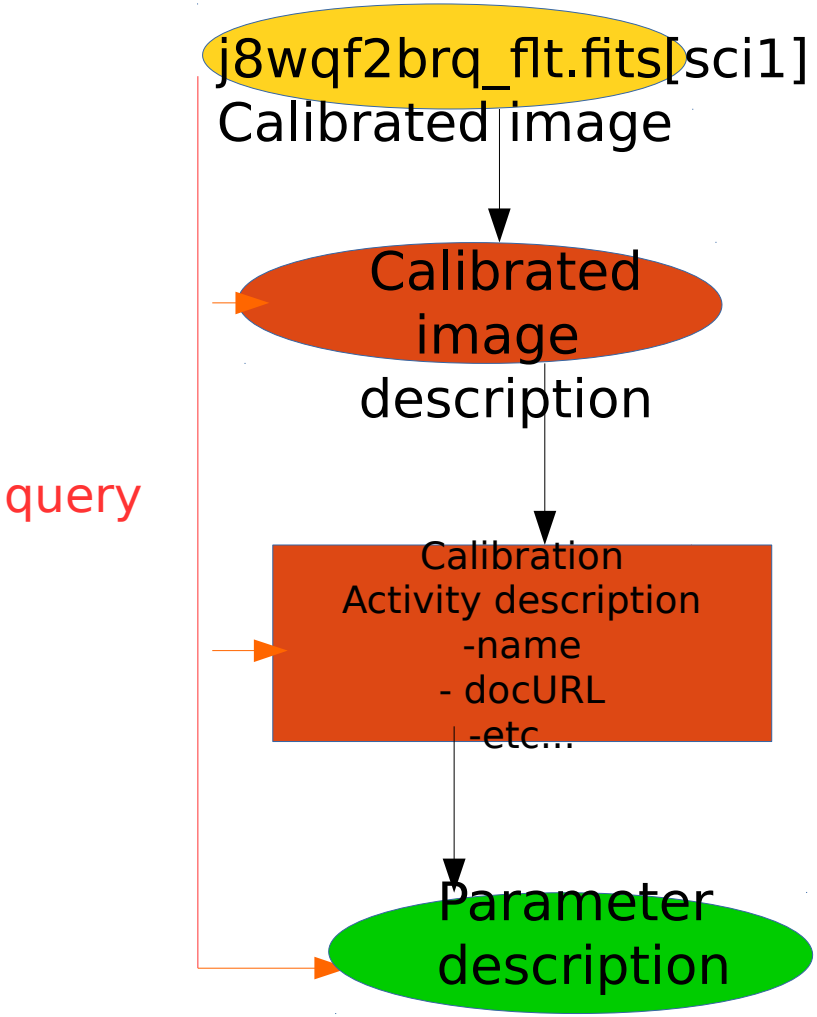
Why a ProvTAP specification ?

Examples :

- Discovering data produced with the same version of a given software.
- Discovering data produced with some specific value of a software configuration parameter :
  - Known by its name
  - Known by its ucd
  - ....
- Discovering a « family » of data : datasets produced by ancestor activities of a given datasets
- Discovering data or activities related to some agent with a given role (operator, editor, author, etc...)

ProvHiPS ADQL query examples :

Find out parameter descriptions of parameter used to generate a calibrated file



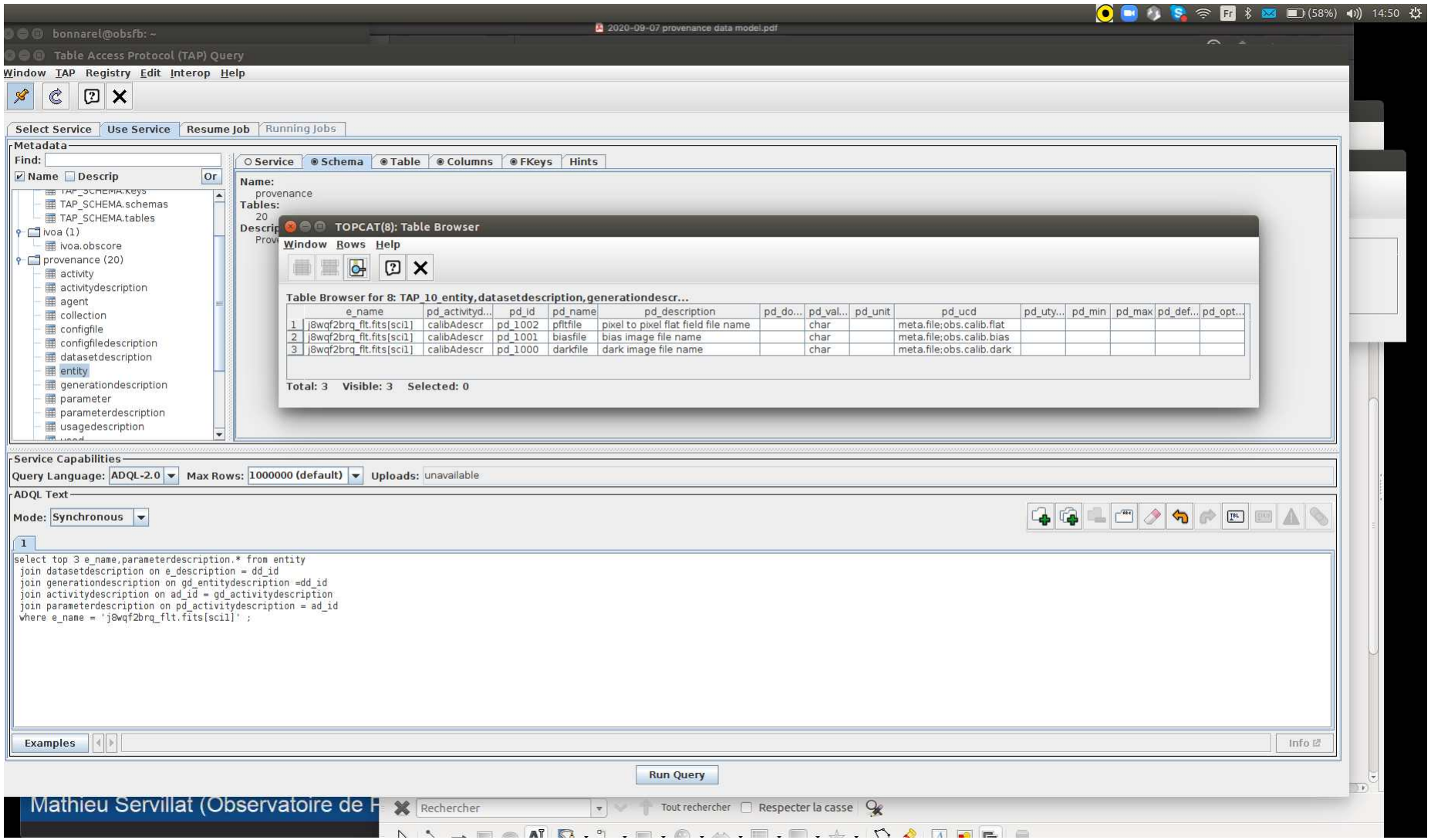
ProvHiPS ADQL query examples :

Find out parameter descriptions of parameter used to generate a calibrated file

```
select top 3 * from entity
join datasetdescription on e_description = dd_id
join generationdescription on gd_entitydescription =dd_id
join activitydescription on ad_id = gd_activitydescription
join parameterdescription on pd_activitydescription = ad_id
where e_name = 'j8wqf2brq_flt.fits[sci1]' ;
```

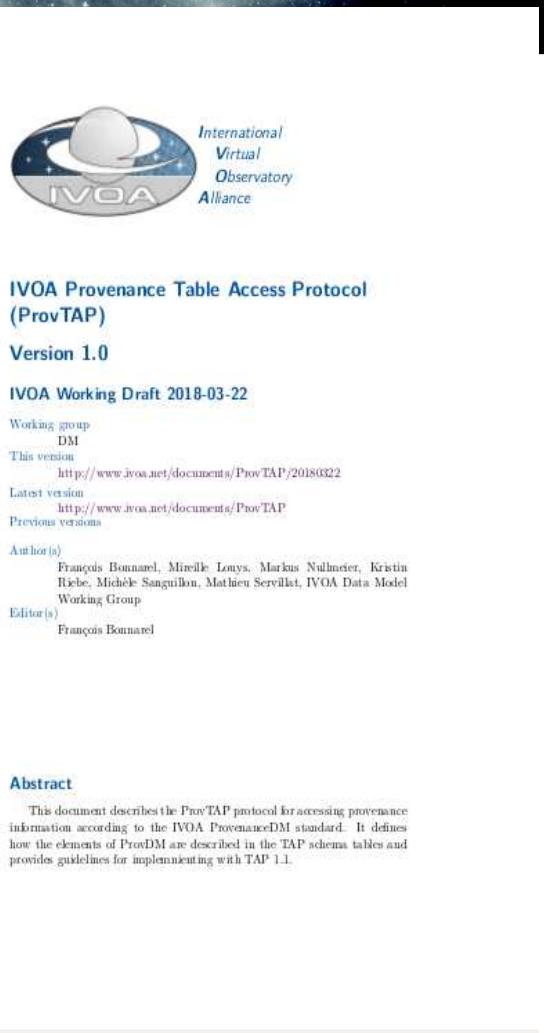
ProvHiPS ADQL query examples :

Find out parameter descriptions of parameter used to generate a calibrated file



ProvTAP = where are we ?

- There is an internal draft on the IVOA DAL pages
- Was not a WD by lack of discussion
- TAP schema mapping classes as tables
- ProvHiPS (provenance of HiPS and HiPS tiles) is an implementation prototype



International Virtual Observatory Alliance

IVOA Provenance Table Access Protocol (ProvTAP)

Version 1.0

IVOA Working Draft 2018-03-22

Working group
DM
This version
http://www.ivoa.net/documents/ProvTAP/20180322
Latest version
http://www.ivoa.net/documents/ProvTAP
Previous versions

Author(s)
François Bonnarel, Mireille Louys, Markus Nullmeier, Kristin Riebe, Michèle Sanguillon, Mathieu Servillat, IVOA Data Model Working Group
Editor(s)
François Bonnarel

Abstract

This document describes the ProvTAP protocol for accessing provenance information according to the IVOA ProvenanceDM standard. It defines how the elements of ProvDM are described in the TAP schema tables and provides guidelines for implemnienting with TAP 1.1.

ProvTAP = where are we ?
Discussion among authors

- Column fixed names : several have the same attribute name. Do we need prefixed names ?
- Utypes for pointers to other classes
- VOTable database dump : usefulness , Foreign Keys ?
- Query building and response visualisation
  - Graphical interface for query building ?
  - Definition of CTE ?
  - VODML Annotation of the response table ?
  - Visualisation of the result (voprov) ?

ProvTAP = where are we ?
External point of view

- DAL chairs
  - 1 to 1 class/table mapping too ambitious. Need for simplification/ denormalization
- ESFRI projects (within ESCAPE)
  - Looking for simplified/partial views
    - Tracing the first step ?
    - Concept of « ProvCore » : minimal model attributes ?
    - Depends from project requirements and uses cases

1 table per class

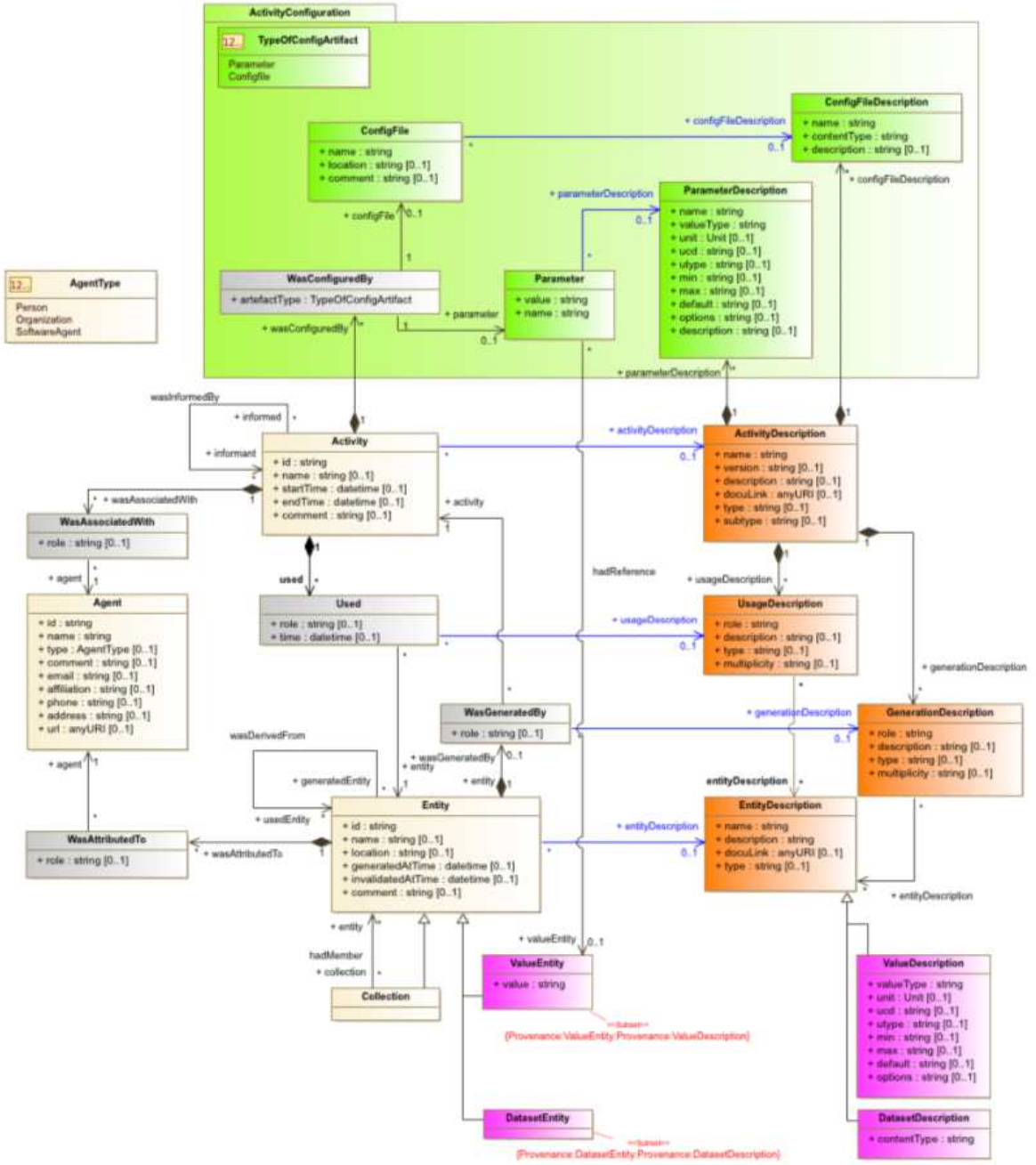


Figure 8: Full class diagram of the IVOA Provenance Data Model.

Solutions

Denormalization : adding « description » classes attributes to the « execution » classes.

- A lot of redundancy.

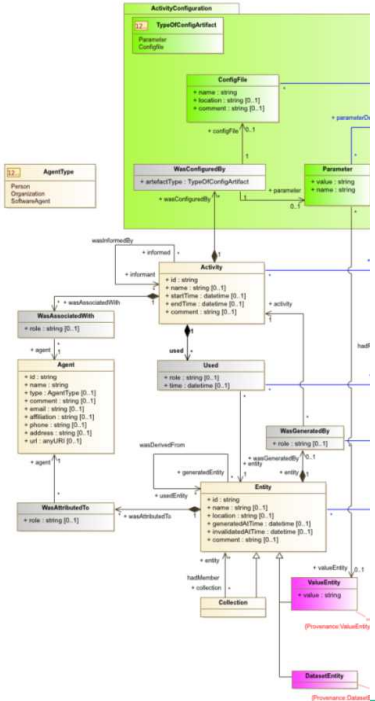


Figure 8: Full class diagram of the IVO

Solutions

- Single step = single table (= join)
- Columns : generation_role, activity_name, activity_type, associated_agent_role, associated_agent_name, used_entity_role, used_entity_pointer, etc.
- → Redundancy (several lines for a single entity/activity).
- → Recursivity ?

Solutions

- Simplification of « descriptions » « linkage »
  - suppress UsageDescription and GenerationDescription ?
- May introduce other difficulties ?

How to go on ?

- Keep the full ProvTAP schema and propose views ?
- Work out which views to actually create
- Can these views become simplified TAP schemata part of the standard ?
- Open widely the discussion or add participants to the project ?