Lecture 21 – Data Models for Discovery

Metadata and Repositories

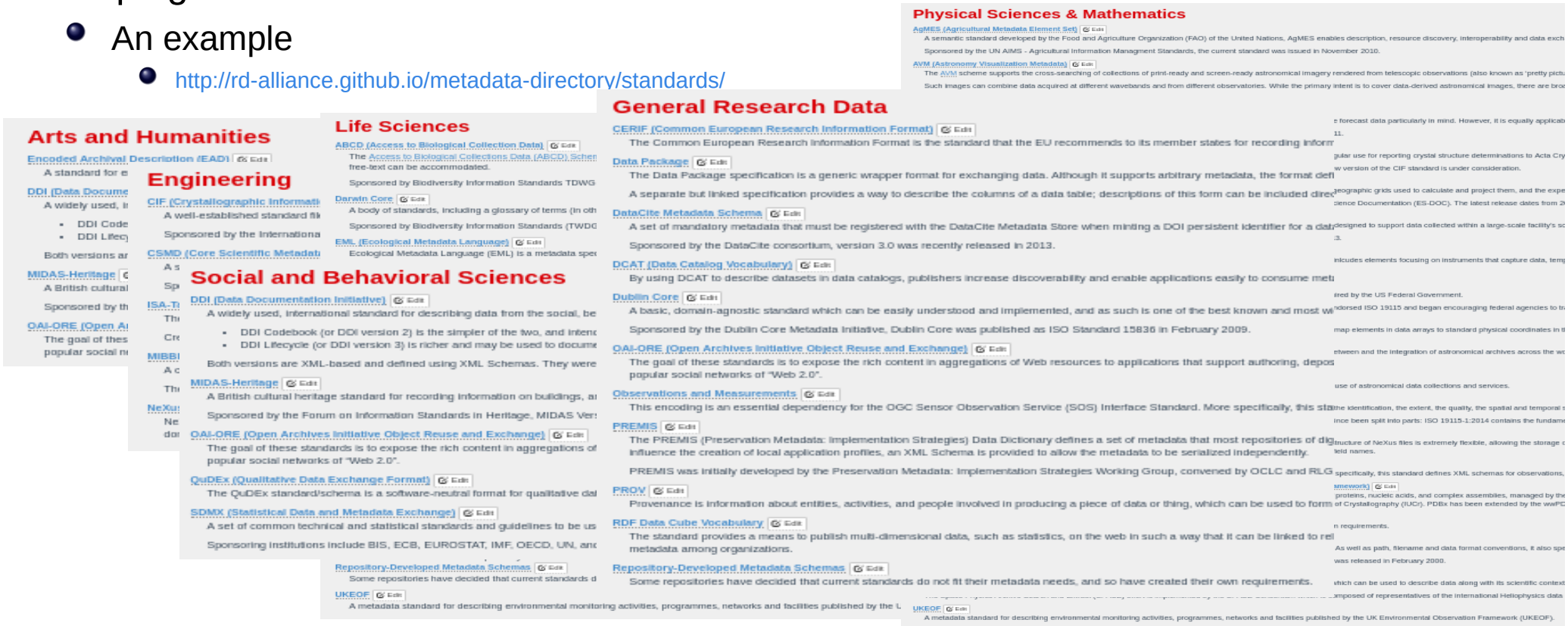
- Data Modelling and metadata modelling are first step in archiving, creating a repository of products
  - Custom ones for specific purposes
  - Common/shared ones to
    - Reach larger communities
    - Interoperate within or outside a research domain
- Models can be standardized exactly as can protocols or other technical specification
  - If not even more
    - Identifiers, vocabularies, formats, …
- Better if standardization is open
  - Communities and organizations exist which have this goal
- (examples and details follow)

Repository Metadata Standards

- Standards for metadata change by domain and granularity
- Keeping track of them is hard work
  - An example
    - ~~http://rd-alliance.github.io/metadata-directory/standards/~~
    - https://rdamsc.bath.ac.uk/

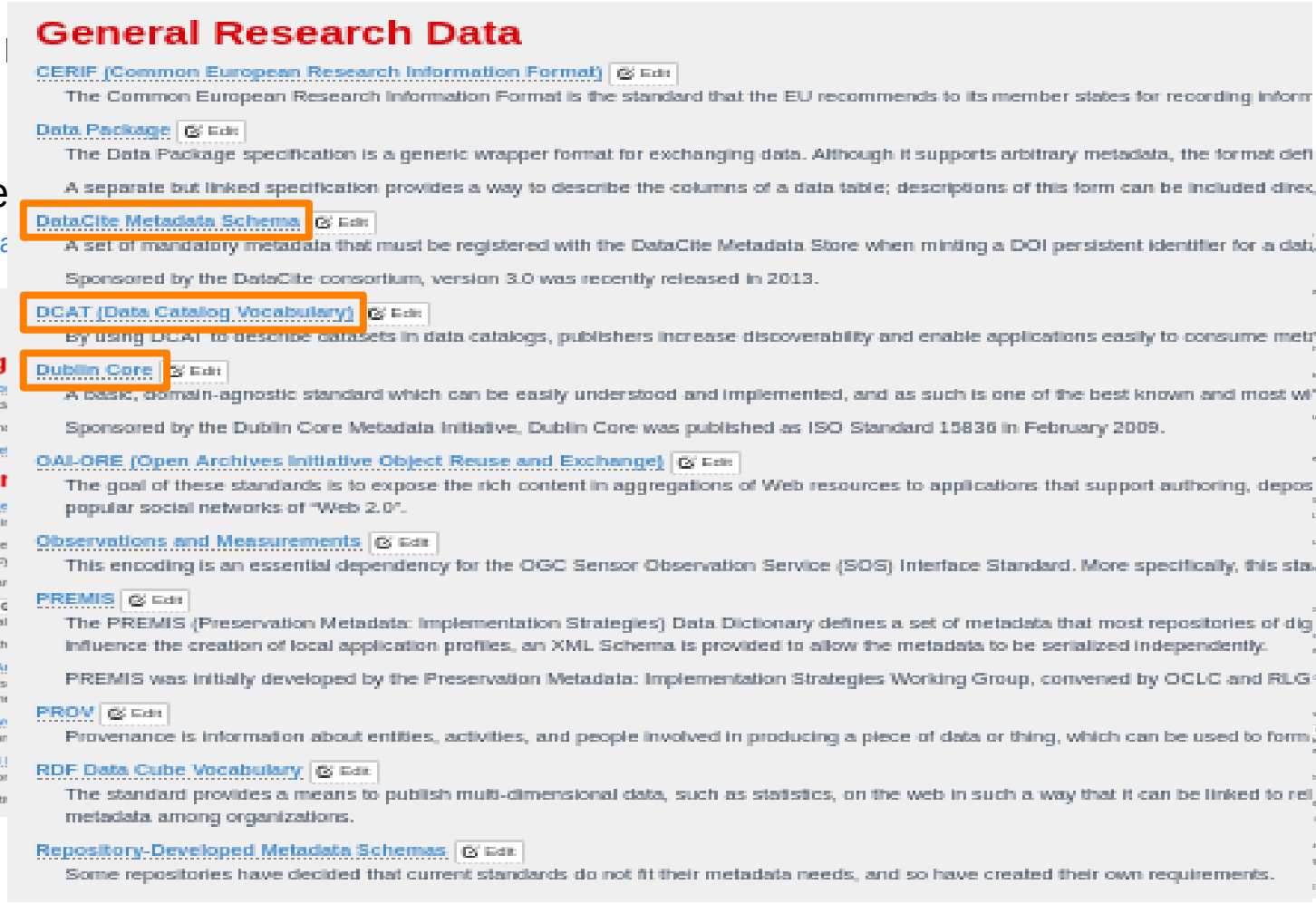
Metadata Standards

- Standards for metadata change by domain and granularity
- Keeping track of them is hard work
  - An example
    - http://rd-alliance.github.io/metadata-directory/standards/



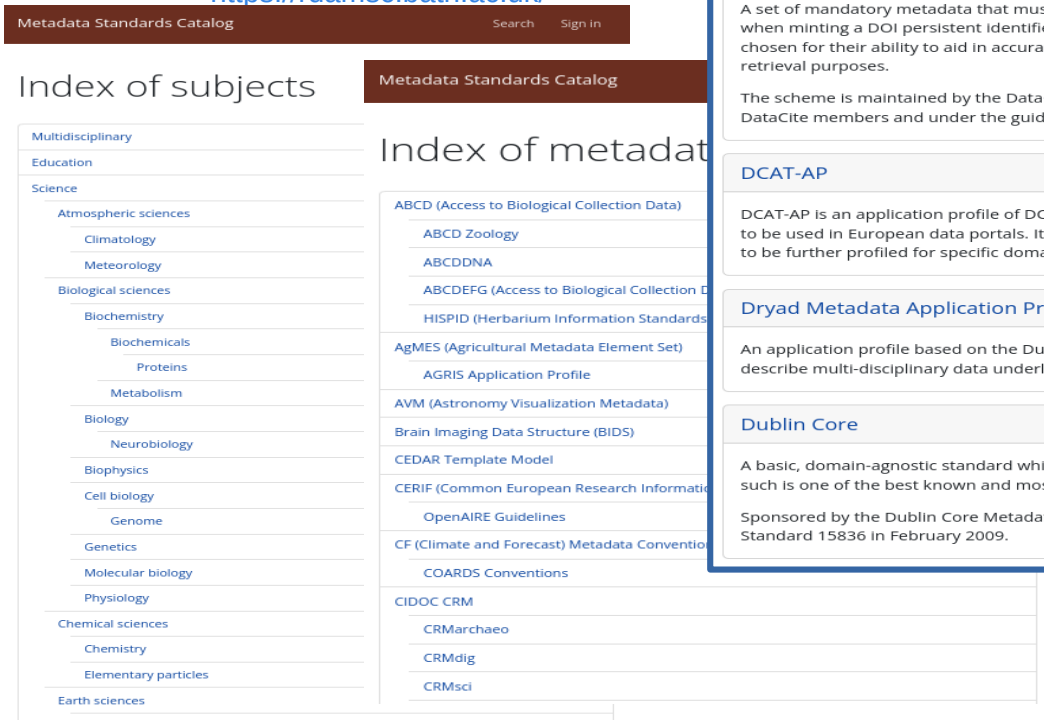
Metadata Standards

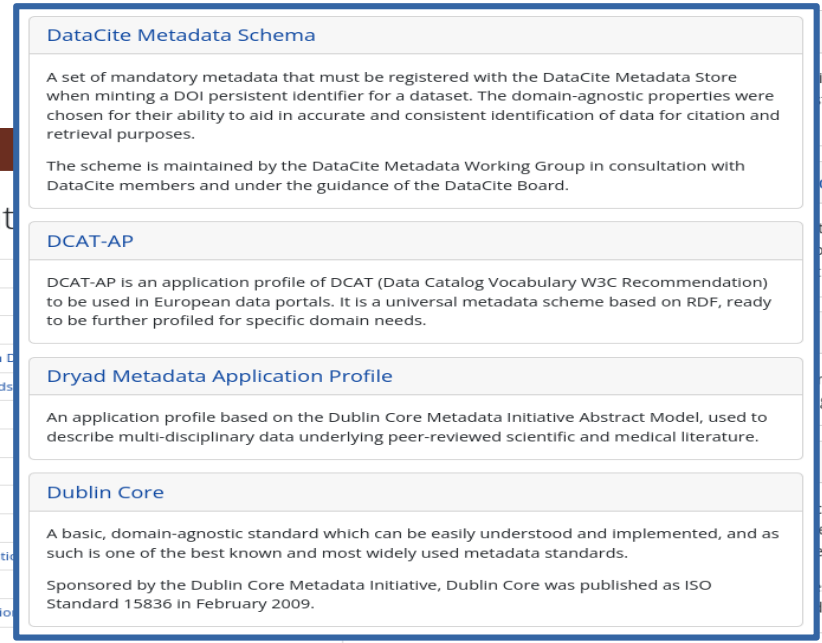
- Standards for
- Keeping track
  - An example
    - http://rd-allia

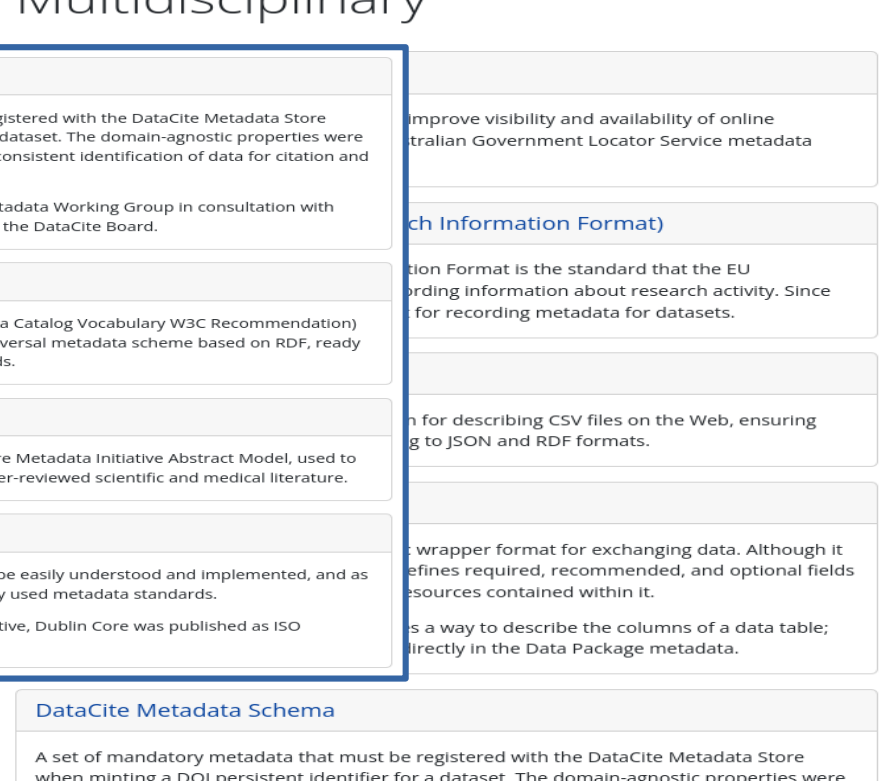


Metadata Standards

- Standards for metadata change by domain and granularity
- Keeping track of them is hard work
  - An example
    - https://rdamsc.bath.ac.uk/







Metadata Standards

- Model extensions/integration

Dublin Core

A basic, domain-agnostic standard which can be easily understood and implemented, and as such is c

Sponsored by the Dublin Core Metadata Initiative, Dublin Core was published as ISO Standard 15836

Summary Edit

Standard Website
http://dublincore.org
Specification
http://dublincore.org/specifications/
Related Vocabularies
DCMI Vocabulary Managment Community
Mappings
UK AGMAP (Academic Geospatial Metadata Application Profile)
DataCite Metadata Schema
PROV
DDI (Data Documentation Initiative)
MARC (Machine-Readable Cataloging)
Subjects
General Research Data
Disciplines
Multi-disciplinary

Extensions Add

AGLS Metadata Profile Edit
An application of Dublin Core designed to improve visibility and availability of online resources, orig

AGRIS Application Profile Edit
A metadata standard drawing on Dublin Core and AgMES created specifically to enhance the desc

ANZLIC Metadata Profile Edit
A profile of ISO 19115, also mapping to the AGLS profile of Dublin Core, designed to facilitate effici

Dryad Metadata Application Profile Edit
An application profile based on the Dublin Core Metadata Initiative Abstract Model, used to describ

eBank UK Metadata Application Profile Edit
A Dublin Core Metadata Application Profile created for the eBank UK project, which provides acces

OpenAIRE Guidelines for publication repositories, data archives and CRIS systems Edit
The OpenAIRE Guidelines are a suite of application profiles designed to allow research institutions
the OAI-PMH metadata harvesting protocol:

- The OpenAIRE Guidelines for Literature Repositories are based on Dublin Core;
- The OpenAIRE Guidelines for Data Archives are based on the DataCite Metadata Schema;
- The OpenAIRE Guidelines for CRIS Managers is based on CERIF.

While the focus of each profile is different, they allow for interlinking and the contextualization of res

Resource Metadata for the Virtual Observatory Edit
Defines metadata terms and concepts necessary for discovery and use of astronomical data collec

The extension is based on Dublin Core, but with astronomy-specific extensions. Resource Metadat
and maintained by IVOA Resource Registry Working Group and NVO Metadata Working Group

Dublin Core

- aka the Dublin Core Metadata Element Set
  - invitational workshop in Dublin, Ohio, 1995
  - "core" because its elements are broad and generic, usable for describing a wide range of resources
    - Not anymore only electronic
- 15 generic elements for describing resources
  - Creator, Contributor, Publisher, Title, Date, Language, Format, Subject, Description, Identifier, Relation, Source, Type, Coverage, Rights
- Later formally standardized and today used in countless implementations
  - one of the top metadata vocabularies on the web
  - “Later” because no semantic web (and, e.g., RDF) was available at the time
- Current version
  - Refers to a set of metadata vocabularies and technical specifications
    - Maintained by the Dublin Core Metadata Initiative (DCMI)
  - The full set of vocabularies includes sets of resource classes, vocabulary encoding schemes, and syntax encoding schemes
  - The terms in DCMI vocabularies are intended to be used in combination with terms from other, compatible vocabularies in the context of application profiles and on the basis of the DCMI Abstract Model [DCAM].
- Dublin Core Metadata Element Set, Version 1.1
  - http://dublincore.org/documents/dces/

Dublin Core - evolution

- Semantic web evolution
  - DCMI includes formal domains and ranges in the definitions of its properties
  - not to affect the conformance of existing implementations of "simple Dublin Core"
    - domains and ranges have not been specified for the “initial” fifteen properties
      - namespace dc:
      - http://purl.org/dc/elements/1.1/
    - fifteen new properties with "names" identical to those of the Dublin Core Metadata Element Set Version 1.1 have been created
      - namespace dcterms:
      - http://purl.org/dc/terms/
    - These fifteen new properties have been defined as subproperties of the corresponding properties of DCES Version 1.1 and assigned domains and ranges

Dublin Core – semantic evolution

- Property

Term Name: contributor	
URI:	http://purl.org/dc/elements/1.1/contributor
Label:	Contributor
Definition:	An entity responsible for making contributions to the resource.
Comment:	Examples of a Contributor include a person, an organization, or a service. Typically, the name of a Contributor should be used to indicate the entity.

- Term

Term Name: contributor	
URI:	http://purl.org/dc/terms/contributor
Label:	Contributor
Definition:	An entity responsible for making contributions to the resource.
Comment:	Examples of a Contributor include a person, an organization, or a service.
Type of Term:	Property
Refines:	http://purl.org/dc/elements/1.1/contributor
Has Range:	http://purl.org/dc/terms/Agent
Version:	http://dublincore.org/usage/terms/history/#contributorT-001

Dublin Core – OAI-PMH usage

- OAI-PMH
  - Open Archives Initiative Protocol for Metadata Harvesting
  - application-independent interoperability framework based on metadata harvesting
  - two classes of participants
    - Data Providers support OAI-PMH as a means of exposing metadata
    - Service Providers harvest metadata via the OAI-PMH
      - for building value-added services
      - Harvest: issue OAI-PMH requests
- OAI-PMH supports the dissemination of records in multiple metadata formats from a repository
  - metadataPrefix arguments are used in ListRecords, ListIdentifiers, and GetRecord requests to retrieve records, or the headers of records that include metadata in the format specified by the metadataPrefix
  - For purposes of interoperability, repositories must disseminate Dublin Core, without any qualification
    - metadataPrefix “oai_dc” reserved
    - XML namespace URI → http://www.openarchives.org/OAI/2.0/oai_dc/
    - URL → http://www.openarchives.org/OAI/2.0/oai_dc.xsd.

Dublin Core – OAI-PMH usage

- OAI-PMH
  - Open Arc
  - applicatio
  - two class
    - Data
    - Servi
      -
      -
- OAI-PMH sup
  - metadata
    records, (
  - For purpc
    - meta
    - XML
    - URL

A XML schema for validating Unqualified Dublin Core metadata associated with the reserved `oai_dc metadataPrefix`

```
<schema targetNamespace="http://www.openarchives.org/OAI/2.0/oai_dc/"
        xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified" attributeFormDefault="unqualified">

<annotation>
  <documentation>
      XML Schema 2002-03-18 by Pete Johnston.
      Adjusted for usage in the OAI-PMH.
      Schema imports the Dublin Core elements from the DCMI schema for unqualified Dublin Core.
      2002-12-19 updated to use simpledc20021212.xsd (instead of simpledc20020312.xsd)
  </documentation>
</annotation>

<import namespace="http://purl.org/dc/elements/1.1/"
        schemaLocation="http://dublincore.org/schemas/xmls/simpledc20021212.xsd"/>

<element name="dc" type="oai_dc:oai_dcType"/>

<complexType name="oai_dcType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="dc:title"/>
    <element ref="dc:creator"/>
    <element ref="dc:subject"/>
    <element ref="dc:description"/>
    <element ref="dc:publisher"/>
    <element ref="dc:contributor"/>
    <element ref="dc:date"/>
    <element ref="dc:type"/>
    <element ref="dc:format"/>
    <element ref="dc:identifier"/>
    <element ref="dc:source"/>
    <element ref="dc:language"/>
    <element ref="dc:relation"/>
    <element ref="dc:coverage"/>
    <element ref="dc:rights"/>
  </choice>
</complexType>

</schema>
```

This Schema is available at http://www.openarchives.org/OAI/2.0/oai_dc.xsd

repository

equests to retrieve
ne metadataPrefix

any qualification

DCAT (W3C)

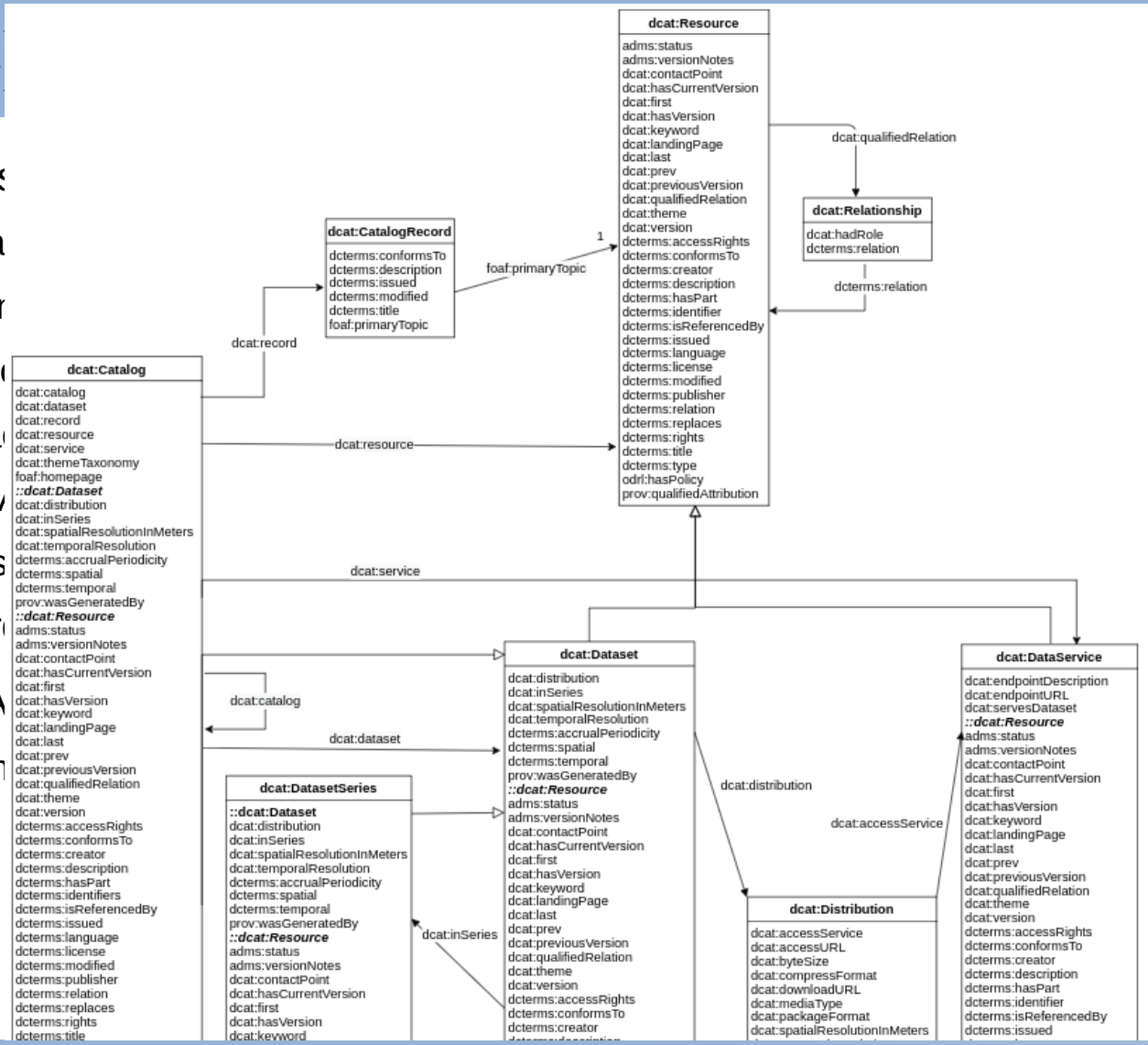
- DCAT is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web
- Using DCAT to describe datasets in data catalogs, publishers increase discoverability and enable applications easily to consume metadata from multiple catalogs
- DCAT does not make any assumptions about the format of the datasets described in a catalog
  - Other, complementary vocabularies may be used together with DCAT to provide more detailed format-specific information
- https://www.w3.org/TR/vocab-dcat/
  - https://www.w3.org/TR/vocab-dcat-3/

DCAT (W3C)

- DCAT v.3 defines seven main classes:
  - Catalog: metadata collection for resources
  - Resource: general parent class for specific typed resources
  - Dataset: collection of data, in the broadest sense
  - Distribution: accessible form of a dataset
  - DataService: API like for dataset(s) access or processing functions
  - DatasetSeries: collection of DataSet(s) (grouping purpose)
  - CatalogRecord: metadata record in a Catalog, for primary metadata information
- Evolution of DCAT enriched its schema
  - e.g. went from 3 to 7 main classes from v1 to v3

DCAT (W3C

- DCAT v.3 defines
  - Catalog: meta
  - Resource: ger
  - Dataset: colle
  - Distribution: a
  - DataService: A
  - DatasetSeries
  - CatalogRecor
- Evolution of DCA
  - e.g. went from



DataCite Metadata

- The DataCite Metadata Schema
  - list of core metadata properties chosen for an accurate and consistent identification of a resource for citation and retrieval purposes
  - The resource that is being identified can be of any kind, but it is typically a dataset
  - term ‘dataset’: its broadest sense
- Collaborate with the Dublin Core Metadata Initiative (DCMI) to maintain a Dublin Core Application Profile for the schema
- Presents 3 different levels of obligation for the metadata properties
  - Mandatory (M) properties must be provided
  - Recommended (R) properties are optional, but strongly recommended for interoperability
  - Optional (O) properties are optional and provide richer description
- https://schema.datacite.org/meta/kernel-4.5/
  - the schema evolves over time… v.4.7 released a few months ago

DataCite Metadata – Properties

Table 1: DataCite Mandatory Properties

ID	Property	Obligation
1	Identifier (with mandatory type sub-property)	M
2	Creator (with optional given name, family name, name identifier and affiliation sub-properties)	M
3	Title (with optional type sub-properties)	M
4	Publisher	M
5	PublicationYear	M
10	ResourceType (with mandatory general type description sub-property)	M

Table 2: DataCite Recommended and Optional Properties

ID	Property	Obligation
6	Subject (with scheme sub-property)	R
7	Contributor (with optional given name, family name, name identifier and affiliation sub-properties)	R
8	Date (with type sub-property)	R
9	Language	O
11	AlternateIdentifier (with type sub-property)	O
12	RelatedIdentifier (with type and relation type sub-properties)	R
13	Size	O
14	Format	O
15	Version	O
16	Rights	O
17	Description (with type sub-property)	R
18	GeoLocation (with point, box and polygon sub-properties)	R
19	FundingReference (with name, identifier, and award related sub-properties)	O

- Among Recommended
  - Description (17) is considered the most important
  - Especially in connected usage with the Recommended sub-property
    - descriptionType = ”Abstract”

DataCite Metadata – Mandatory

ID	DataCite-Property	Occ	Definition	Allowed values, examples, other constraints
1	Identifier	1	The Identifier is a unique string that identifies a resource. For software, determine whether the identifier is for a specific version of a piece of software, (per the Force11 Software Citation Principles[13]), or for all versions.	DOI (Digital Object Identifier) registered by a DataCite member. Format should be "10.1234/foo"
1.1	identifierType	1	The type of Identifier.	*Controlled List Value:* DOI
2	Creator	1-n	The main researchers involved in producing the data, or the authors of the publication, in priority order. To supply multiple creators, repeat this property.	May be a corporate/institutional or personal name. Note: DataCite infrastructure supports up to 8000-10000 names. For name lists above that size, consider attribution via linking to the related metadata.
2.1	creatorName	1	The full name of the creator.	Examples: Charpy, Antoine; Foo Data Center Note: The personal name, format should be: family, given. Non-roman names may be transliterated according to the ALA-LC schemas[14].
2.1.1	nameType	0-1	The type of name	*Controlled List Values:* Organizational Personal
2.2	givenName	0-1	The personal or first name of the creator.	Examples based on the 2.1 names: Antoine; Mae
2.3	familyName	0-1	The surname or last name of the creator.	Examples based on the 2.1 names: Charpy; Jemison
2.4	nameIdentifier	0-n	Uniquely identifies an individual or legal entity, according to various schemas.	The format is dependent upon schema.
2.4.1	nameIdentifierScheme	1	The name of the name identifier schema.	If nameIdentifier is used, nameIdentifierScheme is mandatory. Examples: ORCID[15], ISNI[16]
2.4.2	schemeURI	0-1	The URI of the name identifier schema.	Examples: http://www.isni.org http://orcid.org
2.5	affiliation	0-n	The organizational or institutional affiliation of the creator.	Free text.
3	Title	1-n	A name or title by which a resource is known. May be the title of a dataset or the name of a piece of software.	Free text.
3.1	titleType	0-1	The type of Title.	*Controlled List Values:* AlternativeTitle Subtitle TranslatedTitle Other

- Properties 4,5 have occurrence 1 (being mandatory) without mandatory sub-properties
- Property 10 has mandatory resourceTypeGeneral sub-property, with values in a controlled list:
  - Audiovisual, Collection, DataPaper, Dataset, Event, Image, InteractiveResource, Model, PhysicalObject, Service, Software, Sound, Text, Workflow, Other

DataCite Metadata – Rec. & Opt.

- ...some details
- Most Recommended/Optional properties and sub-properties
  - Have values within controlled list vocabularies
    - 7 Contributor [0-n]: Free text
      - 7.1 contributorType [1]: controlled list
        - ContactPerson, DataCollector, DataCurator, DataManager, Distributor, Editor, HostingInstitution, Producer, ProjectLeader, ProjectManager, ProjectMember, RegistrationAgency, RegistrationAuthority, RelatedPerson, Researcher, ResearchGroup, RightsHolder, Sponsor, Supervisor, WorkPackageLeader, Other
  - Specify free text values through (optional) schema & value URI identifiers
    - 6 Subject [0-n]: Free text
      - 6.1 subjectScheme [0-1] The name of the subject scheme: Free text
      - 6.2 schemeURI [0-1] The URI of the subject identifier scheme
      - 6.3 valueURI [0-1] The URI of the subject term
  - Point to external standard formats, models, schemas, …
    - 9 Language [0-1]: allowed values from IETF BCP 47, ISO 639-1 language codes
      - Examples: en, de, fr

VOResource

- Metadata expressed through XSD documents (and associated Recommendation documents)
  - “Resource Metadata” describes the basic concepts
  - VOResource brings it to XSD and provides a technical entry point
    - Multiple extensions follow: standards, simple access protocols, collections and services, ...
  - Connected interfaces and identifiers specifications

ReR	IVOA Identifiers	2.0	
	IVOA Registry Interfaces	1.1	
	RM - Resource Metadata for the Virtual Observatory	1.12	
	StandardsRegExt: a VOResource Schema Extension for Describing IVOA Standards	1.1	
	SimpleDALRegExt - Describing Simple Data Access Services	1.2	
	VOResource - an XML Encoding Schema for Resource Metadata	1.2	
	VODataService - A VOResource Schema Extension for Describing Collections and Services	1.2	1.3
	RegTAP - Registry Relational Schema	1.2	
	DocRegExt - Educational Resources in the VO		1.0

- http://ivoa.net/documents/ (ReR section in the table)
  - But TAPRegExt in the DAL part…
  - (future) maybe protocol dedicated extensions will end up in the protocol document itself

Resource Metadata for the VO

- Starts out from FITS usage scenario
- General concepts are or map directly Dublin Core
  - The harvesting interface to the Registry is OAI-PMH
- Hierarchical system for metadata management
  - Lower levels provide more extensive and complex metadata
    - description of query syntax, access protocols, and usage policies
- Basic concepts
  - Resource is a general term
    - Described in terms of who curates or maintains it
    - Can be given a name and a unique identifier
  - Organisation is specific type of resource that brings people together to pursue participation in VO applications
    - Can be hierarchical and range greatly in size and scope
      - University, observatory, or government agency, …, scientific project, space mission, or individual researcher
      - A provider is an organisation that makes data and/or services available to users over the network
  - Service is any VO resource that can be invoked by the user to perform some action on their behalf
    - Query service supports a query/response protocol
    - Non-query services: copy or delete files on remote files systems, mail information, kill existing jobs, authorize actions, …
    - Registry is a query service for which the response is a structured description of resources
- Resource metadata include
  - Identity metadata (name, identifier, …)
  - Curation metadata (who supports the resource, its availability, …)
  - Content metadata (types of data, sky coverage, spectral coverage, …)

Resource Metadata - Structure

- Identity
  - Title, Shortname, Identifier (IVOID)
- Curation
  - Publisher (with PublisherID), Creator, Contributor
  - Date, Version
  - Contact
- General Content
  - Subject (controlled IAU vocabulary), Description, Source (Bibliographic reference), ReferenceURL, Type (controlled vocabulary), ContentLevel (target user), Relationship
- Collection & Service
  - Facility, Instrument
  - Coverage: spatial, spectral, bounds, resolution
  - UCD, format, rights
  - Quality flags, validation, uncertainties
- Interface & Capabilities
  - Interface: BaseURL and other URLS
  - Capability: identified by a StandardID (IVOID)

VOResource

- Specifies through XSD hierarchical structure of Resource Metadata
  - What's a timestamp?
    - vr:UTCTimestamp

```
<xs:simpleType name="UTCTimestamp" >
  <xs:restriction base="xs:dateTime" >
    <xs:pattern
                value="\d{4}-\d\d-\d\dT\d\d:\d\d:\d\d(\.\d+)?Z?" />
  </xs:restriction>
</xs:simpleType>
```

  - Relation among Interface and Capability elements

```
<capability xsi:type="ex:ExampleCapType"
  standardID="ivo://example.com/std/exampleAccess"
  xmlns:ex="http://ivoa.net/std/example-1.xsd">
  ...
</capability>
```

```
<capability>
  <interface xsi:type="vr:WebBrowser">
    <accessURL use="full"
      >http://example.org/browser-service</accessURL>
  </interface>
</capability>
```

  - Provide guidelines to extend the VOResource schema

Standards Extensions

- Extend VOResource to add 3 resource types
  - vstd:Standard
  - vstd:ServiceStandard
  - vstd:StandardKeyEnumeration

vstd:Standard Type Schema Definition

```
<xs:complexType name="Standard" >
  <xs:complexContent >
    <xs:extension base="vr:Resource" >
      <xs:sequence >
        <xs:element name="endorsedVersion" type="vstd:EndorsedVersion"
                    maxOccurs="unbounded" />
        <xs:element name="schema" type="vstd:Schema" minOccurs="0"
                    maxOccurs="unbounded" >
        <xs:element name="deprecated" type="xs:token" minOccurs="0" />
        <xs:element name="key" type="vstd:StandardKey" minOccurs="0"
                    maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

An example of a Standard resource that summarizes this specification

```
<?xml version="1.0" encoding="UTF-8"?>
<ri:Resource xsi:type="vstd:Standard" status="active"
             created="2012-02-17T11:15:00" updated="2012-02-17T11:15:00"
             xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
             xmlns:vstd="http://www.ivoa.net/xml/StandardsRegExt/v1.0"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <title> StandardsRegExt: a VOResource Schema Extension for Describing IVOA Standards </title>
    <shortName> StandardsRegExt </shortName>
    <identifier>   ivo://ivoa.net/std/StandardsRegExt   </identifier>
    <curation>
          ......
    </curation>
    <content>
          ......
    </content>
    <endorsedVersion status="pr"> 1.0 </endorsedVersion>
    <schema namespace="http://www.ivoa.net/xml/StandardsRegExt/v1.0">
      <location>http://www.ivoa.net/xml/StandardsRegExt/v1.0</location>
      <description>
        the VOResource extension XML Schema for registering standards
      </description>
      <example>http://rofr.ivoa.net/examples/StandardsRegExt.xml</example>
      <example>http://rofr.ivoa.net/examples/SIA.xml</example>
      <example>http://rofr.ivoa.net/examples/VOSpace.xml</example>
    </schema>
</ri:Resource>
```

Observational Core Metadata

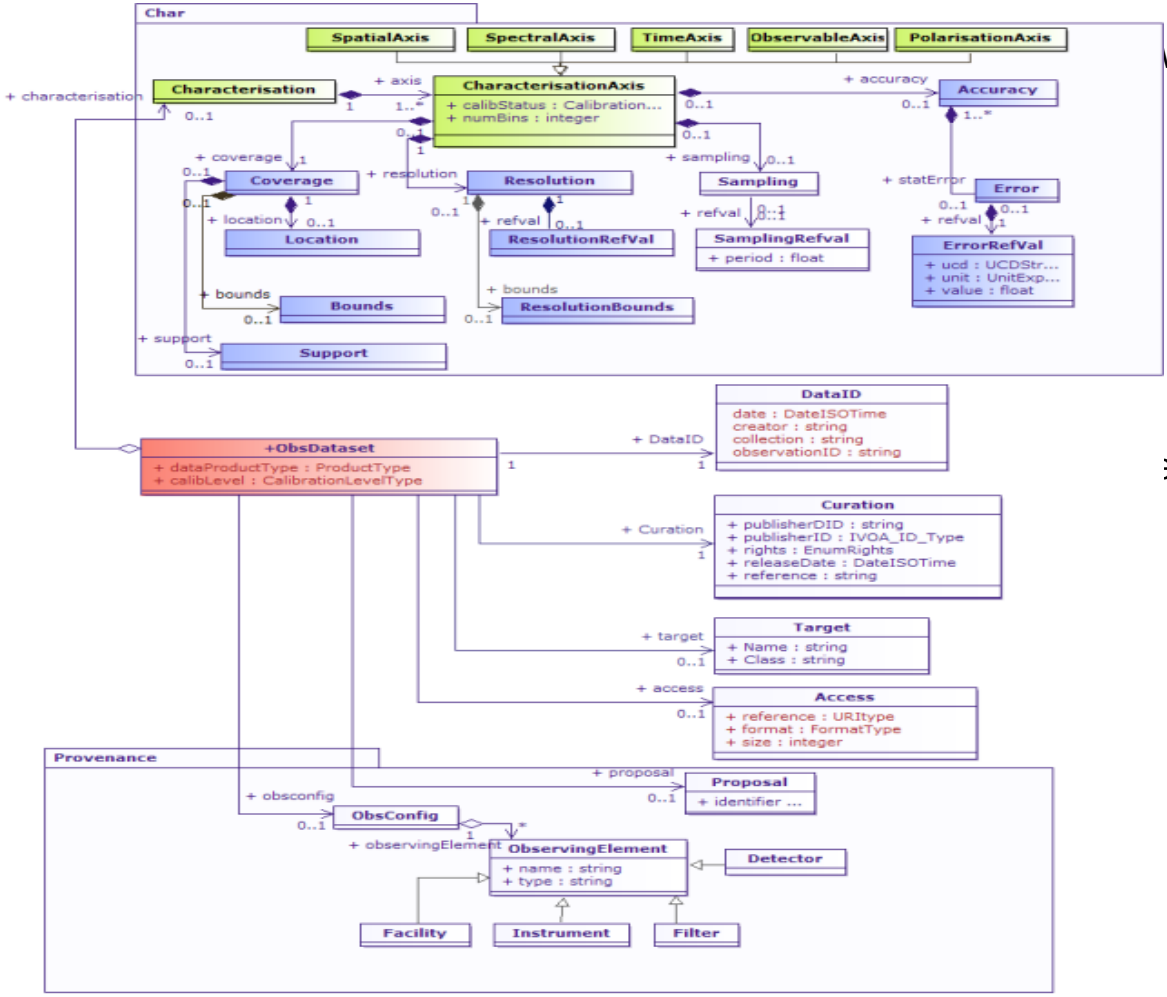
- Core components of the Observation data model necessary to perform data discovery when querying data centers for astronomical observations of interest
- Focus is on data discovery
  - A number of use-cases have been defined
  - Aimed at finding observational data products
  - Broadcasting the same query to multiple archives
    - global data discoverability and accessibility
- Need to give data providers a set of metadata attributes that they can easily map to their database system in order to support queries
- http://www.ivoa.net/documents/ObsCore/20170509/REC-ObsCore-v1.1-20170509.pdf

Observational Core Metadata

- Core component
  centers for astro
- Focus is on data
  - A number of
  - Aimed at find
  - Broadcasting
    - global d
- Need to give dat
  in order to suppo
- http://www.ivoa.ne

when querying data

eir database system



ObsCore – Flat View

- Flat table approach
- Mandatory Structure but NULL-able values
  - Exceptions
    - calib_level, obs_collection, obs_id, obs_publisher_did
- Mandatory
  - Units
  - Data domain
  - Coordinate frames
- Comprehensive usage of
  - Vocabularies
  - Identifiers
- Limited number of mandatory elements
  - Optional standardized ones
  - Custom additions allowed

Column Name	*Unit*	*Type*	*Description*
dataproduct_type	unitless	String	Logical data product type (image etc.)
calib_level	unitless	enum integer	Calibration level {0, 1, 2, 3, 4}
obs_collection	unitless	String	Name of the data collection
obs_id	unitless	String	Observation ID
obs_publisher_did	unitless	String	Dataset identifier given by the publisher
access_url	unitless	String	URL used to access (download) dataset
access_format	unitless	String	File content format (see in App. BB.5.2)
access_estsize	kbyte	integer	Estimated size of dataset in kilo bytes
target_name	unitless	String	Astronomical object observed, if any
s_ra	deg	double	Central right ascension, ICRS
s_dec	deg	double	Central declination, ICRS
s_fov	deg	double	Diameter (bounds) of the covered region
s_region	unitless	String	Sky region covered by the data product (expressed in ICRS frame)
s_xel1	unitless	integer	Number of elements along the first spatial axis
s_xel2	unitless	integer	Number of elements along the second spatial axis
s_resolution	arcsec	double	Spatial resolution of data as FWHM
t_min	d	double	Start time in MJD
t_max	d	double	Stop time in MJD
t_exptime	s	double	Total exposure time
t_resolution	s	double	Temporal resolution FWHM
t_xel	unitless	integer	Number of elements along the time axis
em_min	m	double	Start in spectral coordinates
em_max	m	double	Stop in spectral coordinates
em_res_power	unitless	double	Spectral resolving power
em_xel	unitless	integer	Number of elements along the spectral axis
o_ucd	unitless	String	UCD of observable (e.g. phot.flux.density, phot.count, etc.)
pol_states	unitless	String	List of polarization states or NULL if not applicable
pol_xel	unitless	integer	Number of polarization samples
facility_name	unitless	String	Name of the facility used for this observation
instrument_name	unitless	String	Name of the instrument used for this observation

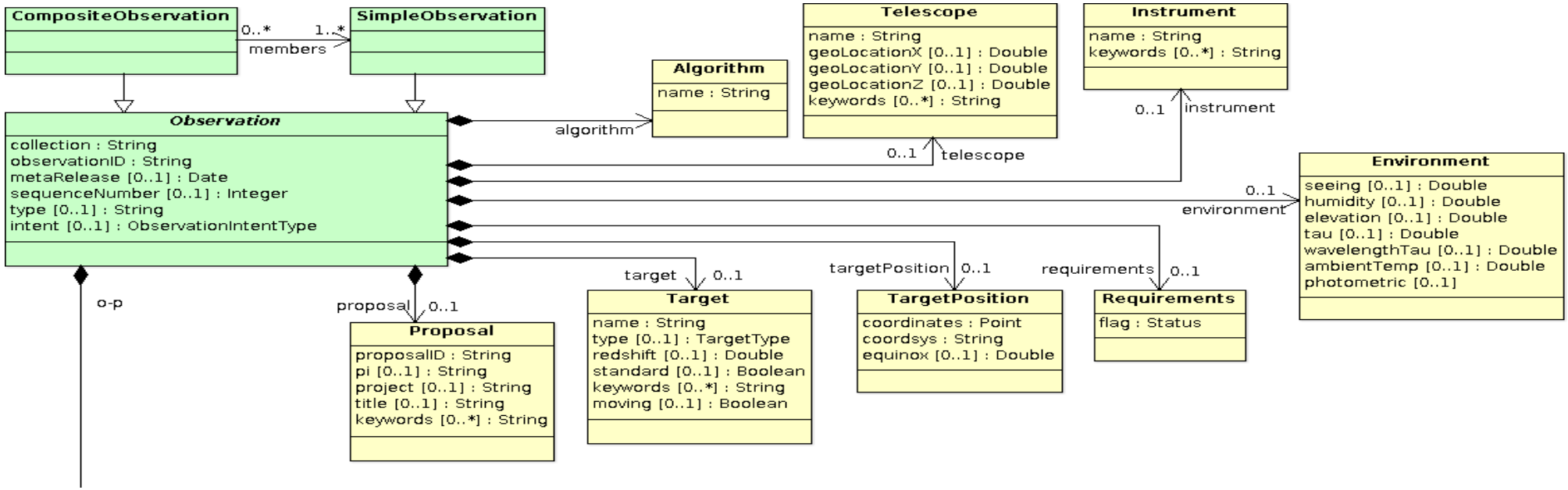
CAOM

- Common Archive Observation Model, enables
  - Storage of observational metadata from the complete set of telescopic data
  - Searching through that metadata using a single interface
- The generalized capability of CAOM comes at the expense of some model complexity and the requirement of adopting a language that is unfamiliar to users
- To decrease the learning curve for users
  - expose CAOM via a simplified search web page interface
  - expose via a Table Access Protocol (TAP) web service
    - for users requiring access to more details of the observations and greater flexibility in query construction
- Model structure
  - Observation: overall container for all associated datasets (top level of the model)
  - Plane: to store each dataset associated to an Observation
  - Artifact: the actual data files containing the observational data (e.g. FITS)
  - Part: each describable part within an Artifact that has a complex data structure
    - Description and discovery of the Part(s) rely on the Artifact's internal metadata content
  - Chunk: further fine-grain level if also Part is a complex data structure (rare)
    - Usually not clearly separated in term of Artifact metadata
- http://www.opencadc.org/caom2/

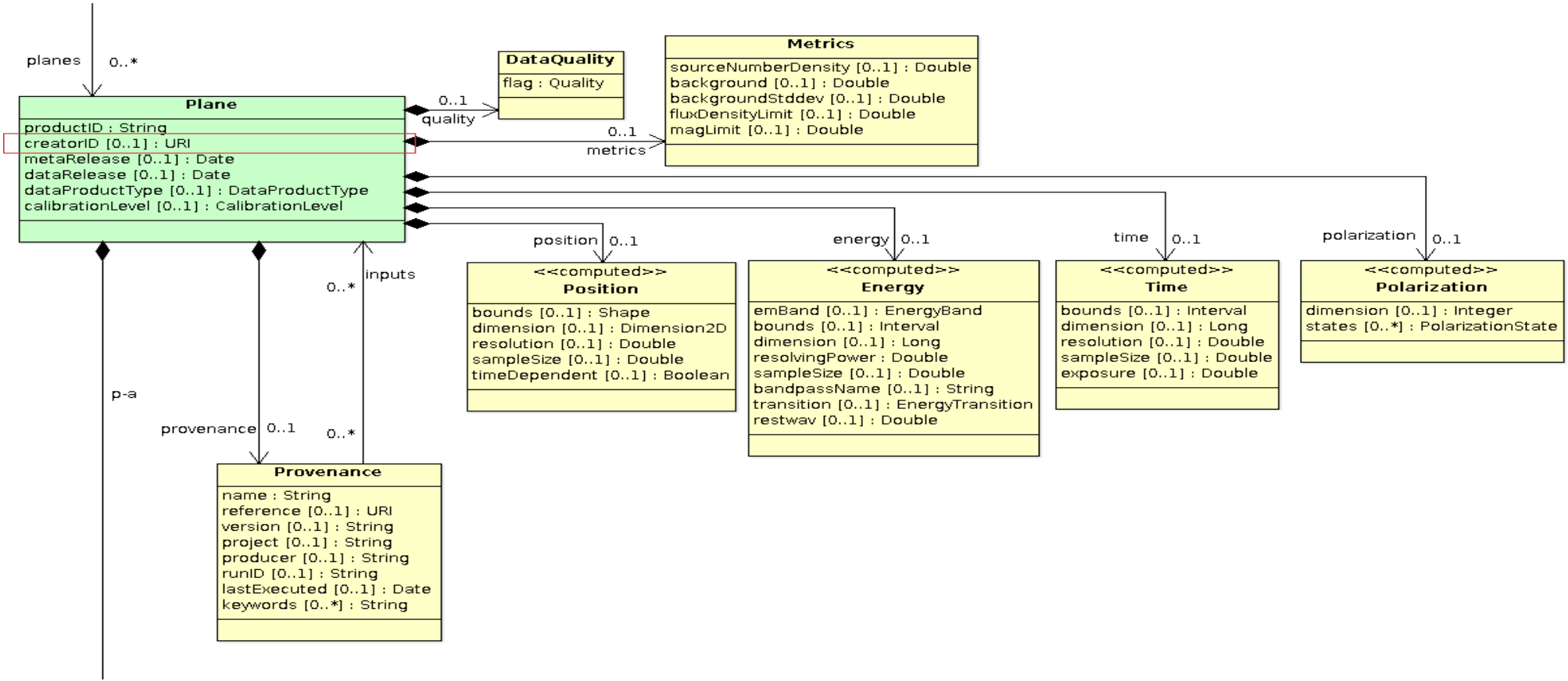
```
Observation
  -> Plane
    -> Artifact
        -> Part
            -> Chunk
        -> Part
            -> Chunk
        -> Part
            ...

  -> Plane
    -> Artifact
        ...
  -> Plane
    ...
```

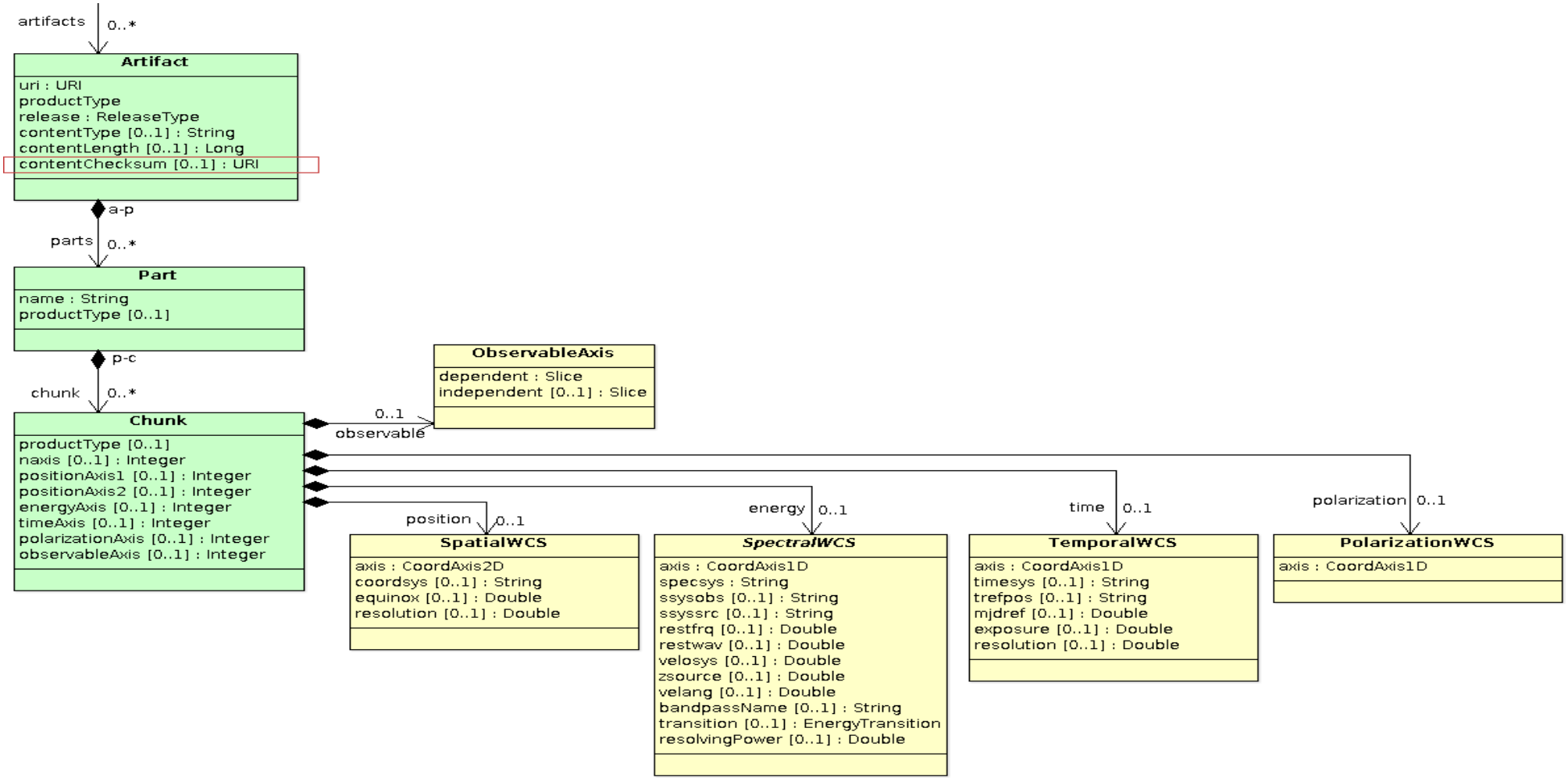
CAOM – Observation



CAOM – Plane


planes
0..*
Plane
productID : String
creatorID [0..1] : URI
metaRelease [0..1] : Date
dataRelease [0..1] : Date
dataProductType [0..1] : DataProductType
calibrationLevel [0..1] : CalibrationLevel
0..1
quality
DataQuality
flag : Quality
0..1
metrics
Metrics
sourceNumberDensity [0..1] : Double
background [0..1] : Double
backgroundStddev [0..1] : Double
fluxDensityLimit [0..1] : Double
magLimit [0..1] : Double
position
0..1
<<computed>>
Position
bounds [0..1] : Shape
dimension [0..1] : Dimension2D
resolution [0..1] : Double
sampleSize [0..1] : Double
timeDependent [0..1] : Boolean
energy
0..1
<<computed>>
Energy
emBand [0..1] : EnergyBand
bounds [0..1] : Interval
dimension [0..1] : Long
resolvingPower : Double
sampleSize [0..1] : Double
bandpassName [0..1] : String
transition [0..1] : EnergyTransition
restwav [0..1] : Double
time
0..1
<<computed>>
Time
bounds [0..1] : Interval
dimension [0..1] : Long
resolution [0..1] : Double
sampleSize [0..1] : Double
exposure [0..1] : Double
polarization
0..1
<<computed>>
Polarization
dimension [0..1] : Integer
states [0..*] : PolarizationState
0..*
inputs
p-a
provenance
0..1
0..*
Provenance
name : String
reference [0..1] : URI
version [0..1] : String
project [0..1] : String
producer [0..1] : String
runID [0..1] : String
lastExecuted [0..1] : Date
keywords [0..*] : String

CAOM – Artifact, Part, Chunk



CAOM – Access to Instances