Lecture 22 – Data Models for Discovery (Meta-data models)

Open Data Management & the Cloud

(Data Science & Scientific Computing / UniTS – DMG)

Lecture summary

- Data discovery
- Why data discovery
- How do we use data: some hints
- How do we collect data: some examples and a demo
- Where we collect data (repos, catalgues, archives recap)
- Metadata representasions to make possibleand ease the data collection

What is Data Discovery?

- **Data discovery** involves the collection and evaluation of data from various sources and is often (in businness) used to understand trends and patterns in the data.

  The data discovery process includes

  - connecting multiple data sources,
  - cleansing and preparing the data,
  - sharing the data throughout the organization
  - performing analysis to gain insights into business processes.

What is Data Discovery?

- Huge amounts of data are collected in businness

  on customers, markets, suppliers, production processes, and more.

- Data are collected differently depending from the environment:
  - from online and traditional transactions systems, sensors, social media, mobile devices, and other sources.
  - In Astrophysics from groound- and space-based instrumentsr also from simulations

Why Data Discovery?

Insights are hidden within the data.

- Huge amounts of data are collected in businness

  on customers, markets, suppliers, production processes, and more.

- Need to collect data from different resources
- Decision makers has to extract insight from data
  - Businness Intelligence
  - Visual Analitics
- Researchers also extract knowledge from data

Knowledge, Wisdom, Insight

Knowledge is the accumulation of facts and data that you have learned about or experienced. It's being aware of something, and having information. Knowledge is really about facts and ideas that we acquire through study, research, investigation, observation, or experience.

Wisdom is the ability to discern and judge which aspects of that knowledge are true, right, lasting, and applicable to your life. It's the ability to apply that knowledge to the greater scheme of life. It's also deeper; knowing the meaning or reason; about knowing why something is, and what it means to your life.

Insight is the deepest level of knowing and the most meaningful to your life. Insight is a deeper and clearer perception of life, of knowledge, of wisdom. It's grasping the underlying nature of knowledge, and the essence of wisdom. Insight is a truer understanding of your life and the bigger picture of how things intertwine.

https://www.lifehack.org/articles/communication/what-are-the-differences-between-knowledge-wisdom-and-insight.html

Knowledge, Wisdom, Insight

Christopher Reiss does a great job of summing up the differences on Quora

https://www.quora.com/Wisdom/What-are-the-valuable-differences-between-knowledge-wisdom-and-insight-Beyond-their-basic-definitions-what-benefits-do-they-hold/answer/Christopher-Reiss?share=1&srid=ot

- Knowledge is measuring that a desert path is 12.4 miles long.
- Wisdom is packing enough water for the hike.
- Insight is building a lemonade stand at mile 6.

What is Business Intelligence?

- **Business intelligence (BI)** includes the applications, infrastructure, tools, and best practices that enable
  - access to and
  - analysis of

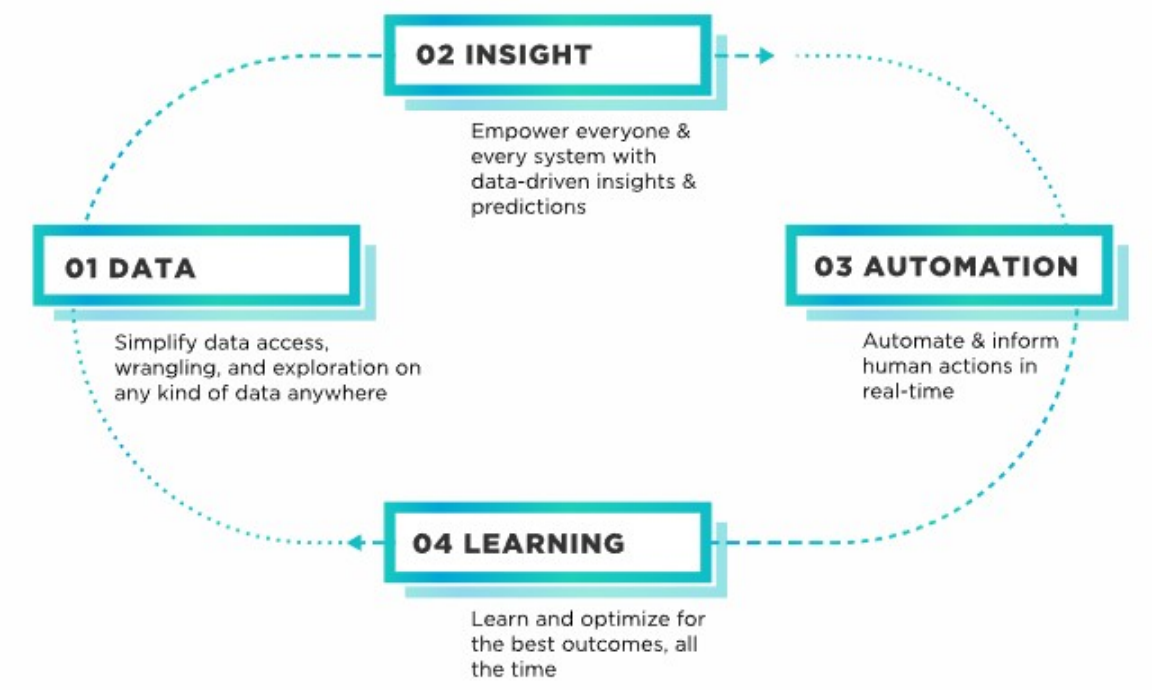
  information to improve and optimize decisions and performance.
- Data are collected from online and traditional transactions systems, sensors, social media, mobile devices, and other sources.

Insights are hidden within that data.

Decision makers need to extract insight from data

- Businness Intelligence
- Visual Analitics

How Business Intelligence woorks



https://www.tibco.com/reference-center/what-is-business-intelligence

Data visualisation

Business intelligence enables data visualizations

Data visualization is the graphical representation of data to help people understand context and significance.

Interactive (clickable) data visualization enables companies to

- drill down to explore details,
- identify patterns and outliers
- change which data is processed and/or excluded.

When data is visualized, it's easier to identify emerging trends, the very first step in deriving insight.

Visual Analytics

Visual analytics is a form of reasoning that uses interactive, visual interfaces.

Visual analytics uses data analytics and interactive visual representations of the data and dashboarding to enable users to interpret large volumes of data.

Visual analytics combine visualization, human factors, and data analysis to gain knowledge from data.

When presented visually, analytics are easier and faster for users to interpret.

Visual analytics make complex issues much easier to understand

The interactive and visual elements are often helpful in communicating what one sees in the data to others and in making better-informed business decisions.

Data scientists can find visual analytics useful to show and clarify businness trends and other concepts to those users that are not data scientists or not know complex statistical algorithms.

Businness Intelligence vs Visual Analytics

- BI concentrates on current and past events recorded in the data
- BA focuses mostly on what is more likely to happen in the future.

Data visualizations answer the “what” questions, but visual analytics help you get to the “why.”

Useful links

https://www.tibco.com/reference-center/what-is-data-discovery

https://www.tibco.com/reference-center/what-is-business-intelligence

https://www.tibco.com/reference-center/guide-to-data-visualization

https://www.tibco.com/reference-center/what-is-visual-analytics

https://www.simplilearn.com/business-intelligence-vs-business-analytics-article

Registry vs Repository

- A **registry** is a list of items with pointers for where to find the items, like the index on a database table or the card catalog for a library.
- A **repository** stores the actual items, like a database table itself or a library's shelves of books.

If you lose a registry, the items still exist; you just may need to *reindex* them.

If you lose a repository, the items are gone.

Catalog vs Registry

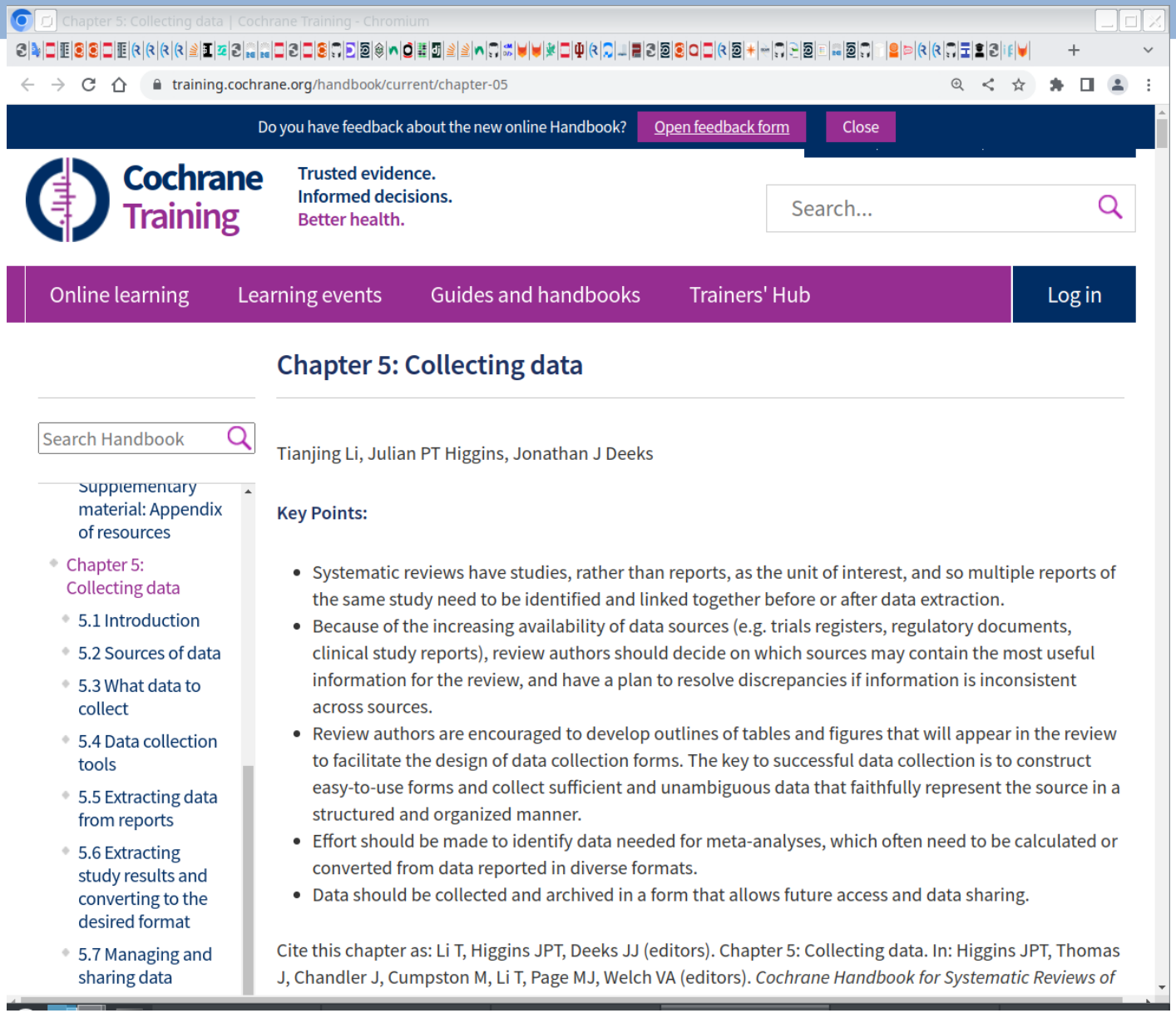
- A **catalog** is a list of something, or a book containing a list. Examples of a catalog:
  - a library's list of all of the books it has available.
  - a booklet showing everything a store has for sale.
- A **registry** is a list of items with pointers for where to find the items. Examples of registry:
  - the index on a database table
  - the card catalog for a library. It is a database containing the network locations of service instances.

The service/data registry is a key part of service/data discovery.

A service registry needs to be highly available and up to date.

Clients can cache network locations obtained from the service/data registry.

How to collect data



https://training.cochrane.org/handbook/current/chapter-05

How to collect data – Web scraping

Web scraping, web harvesting, or web data extraction is a way to extract data from websites.

Web scraping software may directly access the World Wide Web using the HyperText Transfer Protocol or a web browser. Can be done

- manually by a software user,
- automatically by a processes implemented using a bot or web crawler.

Specific collected data is gathered and copied from the web, into a central local database or spreadsheet, for later retrieval or analysis.

https://en.wikipedia.org/wiki/Web_scraping

IVOA-SAMP implementation demo

IVOA: International Virtualbservatory Alliance (see Lecture 4 on FAIR Principles for an introduction)

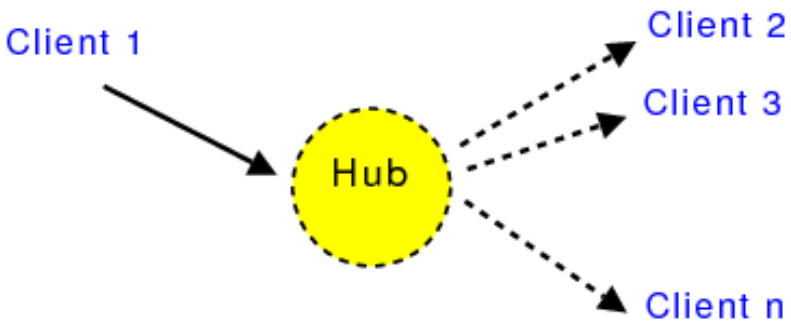
SAMP: Simple Application Messaging Protocol

is a messaging protocol that enables astronomy software tools to interoperate and communicate. It supports communication between applications on the desktop and in web browsers.

https://www.ivoa.net/documents/SAMP/20120411/REC-SAMP-1.3-20120411.html

IVOA-SAMP for data collection

SAMP has a hub-based architecture.



The hub is a single service used to route all messages between clients.

This makes application discovery more straightforward in that each client only needs to locate the hub, and the services provided by the hub are intended to simplify the actions of the client.

A disadvantage of this architecture is that the hub may be a message bottleneck and potential single point of failure.

SAMP may not be suitable for extremely high throughput requirements;

may be mitigated by an appropriate strategy for hub restart if failure is likely.

https://www.ivoa.net/documents/SAMP/20120411/REC-SAMP-1.3-20120411.html

IVOA-SAMP implementation example

ESCAPE EU Project https://projectescape.eu/

European Science Cluster of Astronomy \& Particle physics ESFRI research infrastructures

brings together the astronomy, astroparticle, and particle physics communities to address fundamental challenges in data-driven research, inspired by the goals and needs of major European research infrastructures, or ESFRIs

ESAP https://sdc-dev.astron.nl/esap-gui/

https://projectescape.eu/services/esfris-science-analysis-platform-esap

ESFRI Science Analysis Platform

is a platform-service gateway with the capability to access and combine data from multiple collections and stage for subsequent processing and analysis. It allows data discovery and handling of large and distributed data collections. It is a flexible science platform for the analysis of open access data available through EOSC

IVOA-SAMP implementation demo

Open Topcat

java -jar /home/bertocco/work/VO_School_feb2021/topcat-full.jar

Open esap-samp

https://sdc-dev.astron.nl/esap-gui/samp

Hit register

Follow the first example of the tutorial:

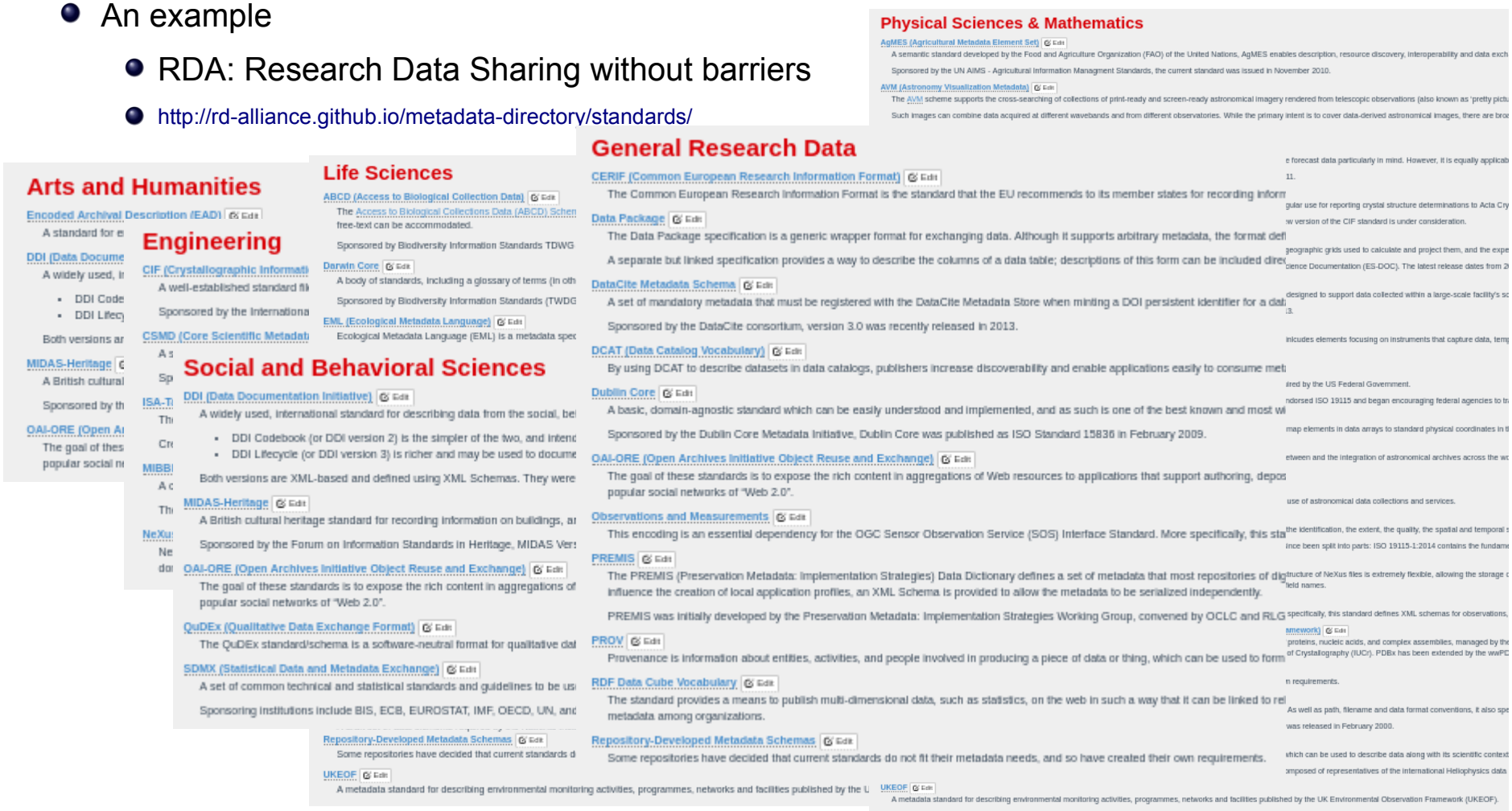
~/work/VO_School_feb2021/slides/topcat$ evince topcat_tutorial.pdf

Metadata and Repositories

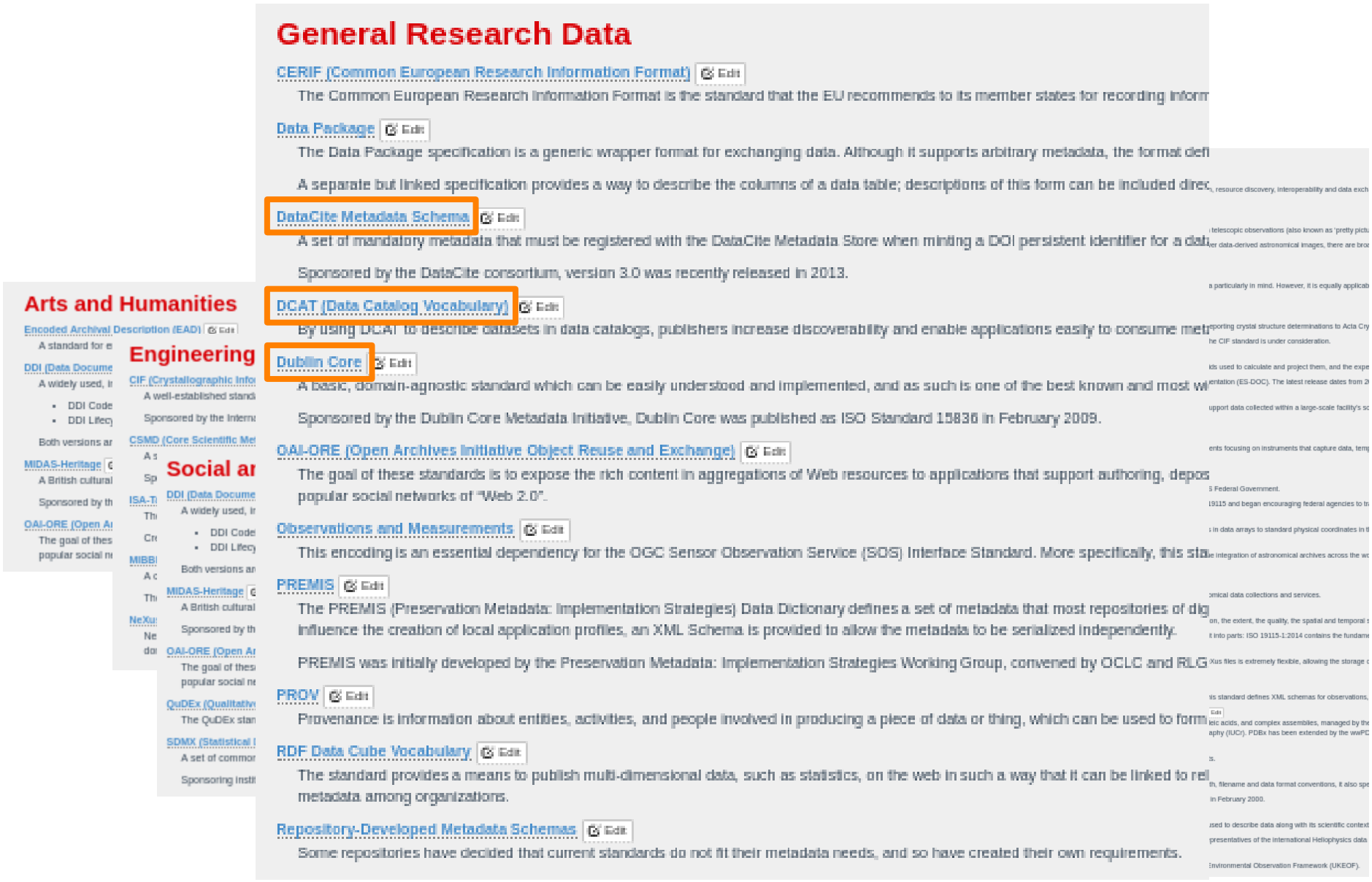
- Data Modelling and metadata modelling are first step in archiving, creating a repository of products
  - Custom ones for specific purposes
  - Common/shared ones to
    - Reach larger communities
    - Interoperate within or outside a research domain
- Models can be standardized exactly as can protocols 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)

Metadata Standards

- Standards for metadata change by domain and granularity
- Keeping track of them is hard work
  - An example
    - RDA: Research Data Sharing without barriers
    - http://rd-alliance.github.io/metadata-directory/standards/



Metadata Standards



General Research Data

CERIF (Common European Research Information Format) Edit

The Common European Research Information Format is the standard that the EU recommends to its member states for recording inform

Data Package Edit

The Data Package specification is a generic wrapper format for exchanging data. Although it supports arbitrary metadata, the format defi

A separate but linked specification provides a way to describe the columns of a data table; descriptions of this form can be included direc

DataCite Metadata Schema Edit

A set of mandatory metadata that must be registered with the DataCite Metadata Store when minting a DOI persistent identifier for a dat

Sponsored by the DataCite consortium, version 3.0 was recently released in 2013.

DCAT (Data Catalog Vocabulary) Edit

By using DCAT to describe datasets in data catalogs, publishers increase discoverability and enable applications easily to consume met

Dublin Core Edit

A basic, domain-agnostic standard which can be easily understood and implemented, and as such is one of the best known and most wi

Sponsored by the Dublin Core Metadata Initiative, Dublin Core was published as ISO Standard 15836 in February 2009.

OAI-ORE (Open Archives Initiative Object Reuse and Exchange) Edit

The goal of these standards is to expose the rich content in aggregations of Web resources to applications that support authoring, depos

popular social networks of "Web 2.0".

Observations and Measurements Edit

This encoding is an essential dependency for the OGC Sensor Observation Service (SOS) Interface Standard. More specifically, this sta

PREMIS Edit

The PREMIS (Preservation Metadata: Implementation Strategies) Data Dictionary defines a set of metadata that most repositories of dig

influence the creation of local application profiles, an XML Schema is provided to allow the metadata to be serialized independently.

PREMIS was initially developed by the Preservation Metadata: Implementation Strategies Working Group, convened by OCLC and RLG

PROV Edit

Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form

RDF Data Cube Vocabulary Edit

The standard provides a means to publish multi-dimensional data, such as statistics, on the web in such a way that it can be linked to rel

metadata among organizations.

Repository-Developed Metadata Schemas Edit

Some repositories have decided that current standards do not fit their metadata needs, and so have created their own requirements.

Arts and Humanities

Encoded Archival Description (EAD) Edit

A standard for e

DDI (Data Docume

A widely used, i

- DDI Code
- DDI Lifecy

Both versions ar

MIDAS-Heritage

A British cultural

Sponsored by th

OAI-ORE (Open Ar

The goal of thes

popular social n

Engineering

CIF (Crystallographic Info

A well-established standa

Sponsored by the Interna

CSMD (Core Scientific Me

A s

Sp

ISA-T

Th

Cr

MIBBI

A c

Th

NeXu

Ne

do

Social ar

DDI (Data Docume

A widely used, i

- DDI Code
- DDI Lifecy

Both versions ar

MIDAS-Heritage

A British cultural

Sponsored by th

OAI-ORE (Open Ar

The goal of thes

popular social n

QuDEx (Qualitativ

The QuDEx stan

SDMX (Statistical I

A set of commor

Sponsoring Instit

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 - History

- Formally standardized in 1995 in an invitational workshop in Dublin.

"Dublin" refers to Dublin, Ohio, USA where the schema originated during the 1995 invitational OCLC/NCSA Metadata Workshop, hosted by the OCLC (known at that time as Online Computer Library Center), a library consortium based in Dublin, and the National Center for Supercomputing Applications (NCSA).

"Core" refers to the metadata terms as "broad and generic being usable for describing a wide range of resources". T

- The semantics of Dublin Core were established and are maintained by an international, cross-disciplinary group of professionals from librarianship, computer science, text encoding, museums, and other related fields of scholarship and practice.
- In 1999, the first Dublin Core encoding standard was in HTML.
- Starting in 2000, the Dublin Core community focused on "application profiles" – the idea that metadata records would use Dublin Core together with other specialized vocabularies to meet particular implementation requirements. During that time, the World Wide Web Consortium's work on a generic data model for metadata, the Resource Description Framework (RDF), was maturing. As part of an extended set of DCMI metadata terms, Dublin Core became one of the most popular vocabularies for use with RDF, more recently in the context of the linked data movement.
- The Dublin Core Metadata Initiative (DCMI) provides an open forum for the development of interoperable online metadata standards for a broad range of purposes and of business models.
- In 2008, DCMI separated from OCLC and incorporated as an independent entity.
- Currently, any and all changes that are made to the Dublin Core standard, are reviewed by a DCMI Usage Board within the context of a DCMI Namespace Policy (DCMI-NAMESPACE). This policy describes how terms are assigned and also sets limits on the amount of editorial changes allowed to the labels, definitions, and usage comments.

https://en.wikipedia.org/wiki/Dublin_Core

Dublin Core - Definition

- The Dublin Core, also known as the Dublin Core Metadata Element Set, is a set of fifteen "core" elements (properties) for describing resources.
  - "core" because its elements are broad and generic, usable for describing a wide range of resources
  - not anymore only electronic
  - one of the top metadata vocabularies on the web
- Dublin Core Metadata Element Set, Version 1.1

  http://dublincore.org/documents/dces/

Dublin Core – Original Elements

- 15 generic elements for describing resources
  - Contributor – "An entity responsible for making contributions to the resource".
  - Coverage – "The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant".
  - Creator – "An entity primarily responsible for making the resource".
  - Date – "A point or period of time associated with an event in the lifecycle of the resource".
  - Description – "An account of the resource".
  - Format – "The file format, physical medium, or dimensions of the resource".
  - Identifier – "An unambiguous reference to the resource within a given context".
  - Language – "A language of the resource".
  - Publisher – "An entity responsible for making the resource available".
  - Relation – "A related resource".
  - Rights – "Information about rights held in and over the resource".
  - Source – "A related resource from which the described resource is derived".
  - Subject – "The topic of the resource".
  - Title – "A name given to the resource".
  - Type – "The nature or genre of the resource".

https://www.dublincore.org/specifications/dublin-core/dces/

Dublin Core – Evolution

- The core properties are part of a larger set of DCMI Metadata Terms.
- Later formally standardized. “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 - 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
      - new 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 – Evolution

- Dublin Core Metadata Element Set, Version 1.1

https://www.dublincore.org/specifications/dublin-core/dces/

https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

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 specifies the format to beused to reply to a request

    it is used toghethr with the requests methods available in the protocol: ListRecords, ListIdentifiers, and GetRecord to retrieve the records, or the headers of the 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 validation schema

- OAI-PMH
  - Open Ar
  - applicat
  - two clas
    - Data
    - Serv
      -
      -
- OAI-PMH s
  from a repo
  - metadat
    request
    format s
  - For purp
    any qua
    - meta
    - XML
    - URL

a harvesting

ata formats

d GetRecord
etadata in the

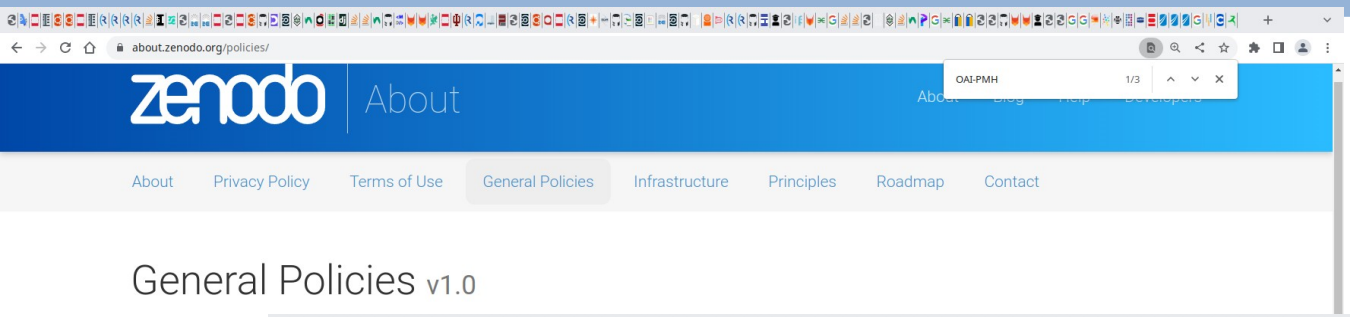
Core, without

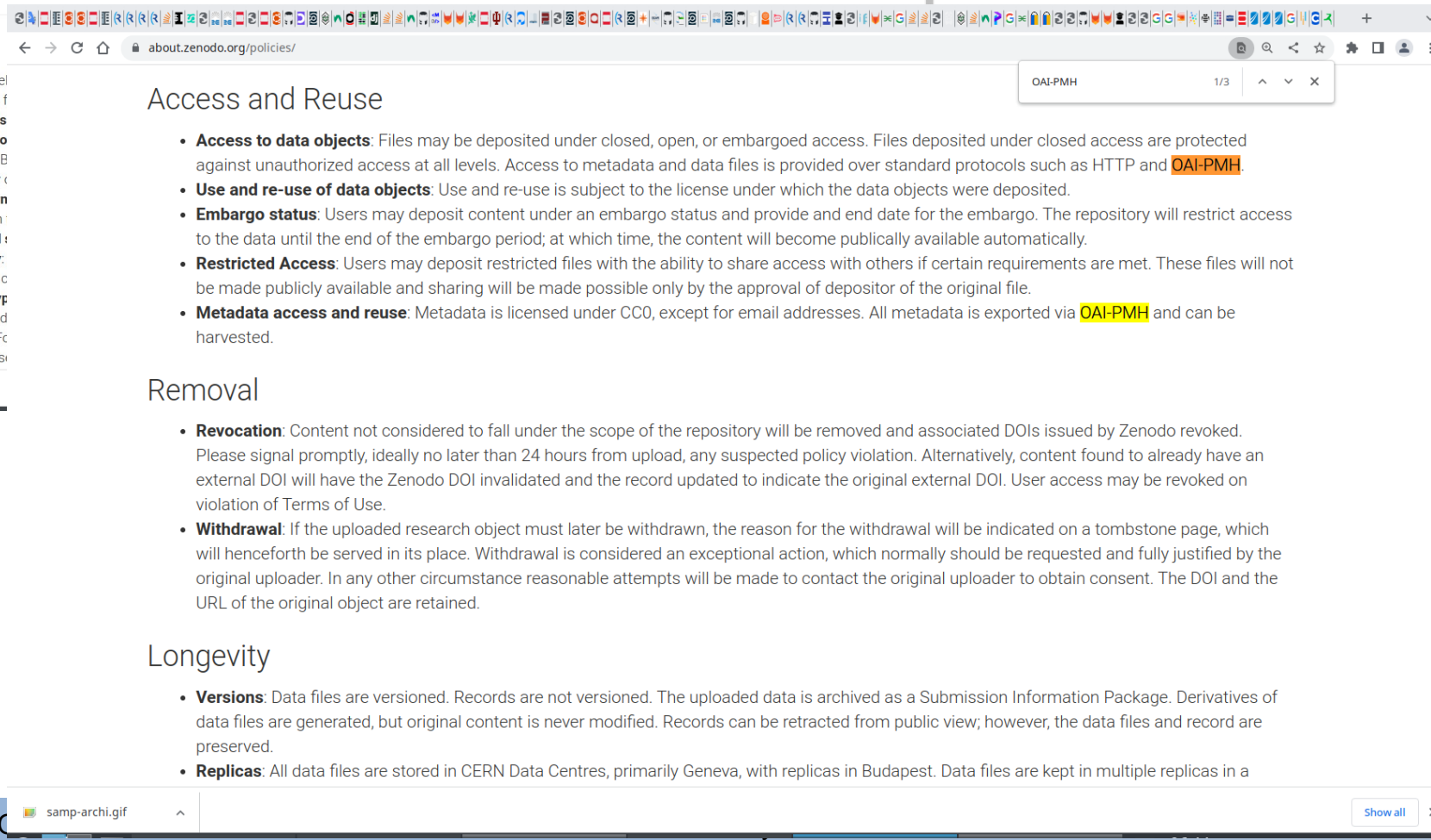
A XML schema for validating Unqualified Dublin Core metadata associated with the reserved `oai_dc metadataPrefix`

```
<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

OAI-PMH & Zenodo





DCAT (W3C technology)

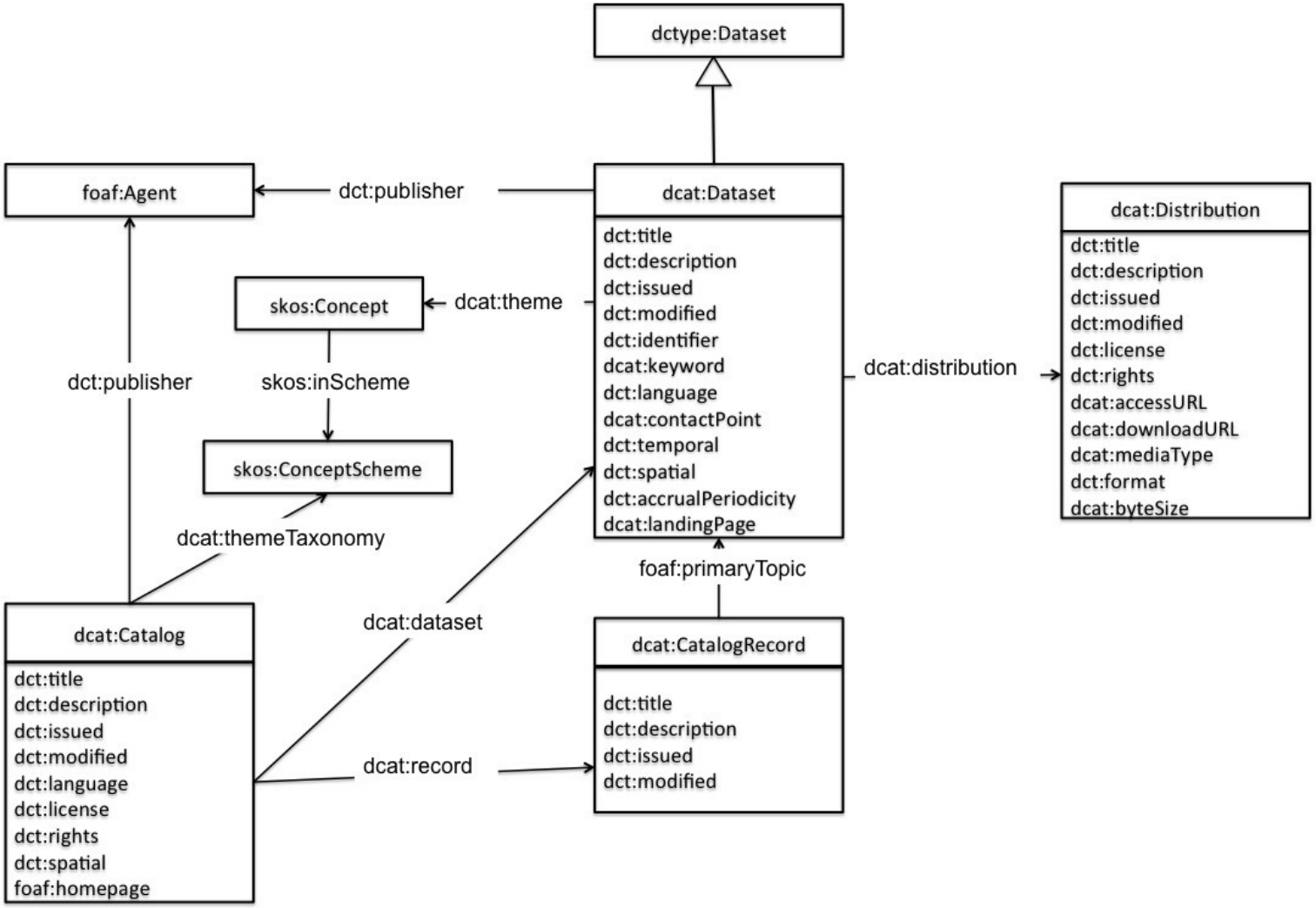
- DCAT is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web
- Using DCAT to describe datasets in data catalogs, publishers increase discoverability and enable applications easily to consume metadata from multiple catalogs
- DCAT does not make any assumptions 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-2/

DCAT (W3C)

- DCAT defines three main classes:
  - dcat:Catalog represents the catalog
  - dcat:Dataset represents a dataset in a catalog.
  - dcat:Distribution represents an accessible form of a dataset
- A dataset does not have to be available as a downloadable file.
  - A dataset that is available via an API can be defined as an instance of dcat:Dataset
  - the API can be defined as an instance of dcat:Distribution
- Class dcat:CatalogRecord describes a dataset entry in the catalog
  - dcat:Dataset represents the dataset itself
  - dcat:CatalogRecord represents the record that describes a dataset in the catalog
    - is optional
    - is used to capture provenance information

DCAT (W3C): Metadata Model

- DCA
  - c
  - c
  - c
- A da
  - A
    c
  - t
- Clas
  - c
  - c



catalog

Prefix dct: namespace to Dublin Core elements 1.1

DataCite Metadata Model

- 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
- http://schema.datacite.org/meta/kernel-4.3/
  - v.4.1: explicit changes to improve software citation
  - v.4.2: alternate identifiers and machine readable licences
  - v.4.3: “funder” identifiers...

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+SubProperties

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

ID	DataCite-Property	Occ	Definition	Allowed values, examples, other constraints
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 and optional
      - 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		2.0 2.0 2.0 2.0 1.12 1.11 1.10 1.10 1.10 1.00
	IVOA Registry Interfaces	1.1		1.1 1.1 1.1 1.1 1.1 1.1 1.0 1.0 1.00 1.02 1.01 1.00
	RM - Resource Metadata for the Virtual Observatory	1.12		1.12 1.12 1.10 1.10 1.01 1.01 1.00 1.00
	StandardsRegExt: a VOResource Schema Extension for Describing IVOA Standards	1.0		1.0 1.0 1.0 1.0 1.0 1.0 1.0
	SimpleDALRegExt - Describing Simple Data Access Services	1.1		1.1 1.1 1.1 1.1 1.0 1.0 1.0 1.0 1.0
	VOResource - an XML Encoding Schema for Resource Metadata	1.1		1.1 1.1 1.1 1.1 1.1 1.03 1.02 1.02 1.01 1.00
	VODataService - A VOResource Schema Extension for Describing Collections and Services	1.1	1.2	1.2 1.1 1.1 1.1 1.1 1.1 1.10
	RegTAP - Registry Relational Schema	1.0	1.1	1.1 1.1 1.0 1.0 1.0 1.0 1.0 1.0 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 (Flexible Image Transport System) 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, …)

en.wikipedia.org/wiki/FITS

Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages

Get citation

FITS

From Wikipedia, the free encyclopedia

For other uses, see FITS (disambiguation).

Flexible Image Transport System (**FITS**) is an open standard defining a digital file format useful for storage, transmission and processing of data: formatted as multi-dimensional arrays (for example a 2D image), or tables.[3] FITS is the most commonly used digital file format in astronomy. The FITS standard was designed specifically for astronomical data, and includes provisions such as describing photometric and spatial calibration information, together with image origin metadata.

The FITS format was first standardized in 1981;[4] it has evolved gradually since then, and the most recent version (4.0) was standardized in 2016. FITS was designed with an eye towards long-term archival storage, and the maxim *once FITS, always FITS* represents the requirement that developments to the format must be backward compatible.

FITS

Filename extension	`.fits` , `.fit` , `.fts`
Internet media type	`image/fits` `application/fits`[1]
Developed by	IAU FITS Working Group[2]
Initial release	1981; 41 years ago
Latest release	4.0 July 2016; 6 years ago
Type of format	image format, structured data
Website	fits.gsfc.nasa.gov

Image metadata is stored in a human-readable ASCII header. The information in this header is designed to calculate the byte offset of some information in the subsequent data unit to support direct access to the data cells. Each FITS file consists of one or more headers containing ASCII card images[a] that carry keyword/value pairs, interleaved between data blocks. The keyword/value pairs provide information such as size, origin, coordinates, binary data format, free-form comments, history of the data, and anything else the creator desires: while many keywords are reserved for FITS use, the standard allows arbitrary use of the rest of the name-space.

FITS is also often used to store non-image data, such as spectra, photon lists, data cubes, or structured data such as multi-table databases. A FITS file may contain several extensions, and each of these may contain a data object. For example, it is possible to store x-ray and infrared exposures in the same file.

See: https://fits.gsfc.nasa.gov/
https://www.vaticanlibrary.va/it/il-patrimonio/il-progetto-di-digitalizzazione.html
https://www.vaticanlibrary.va/it/il-patrimonio/fits-files.html

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 (Unified Content Descriptors), 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, when needed

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

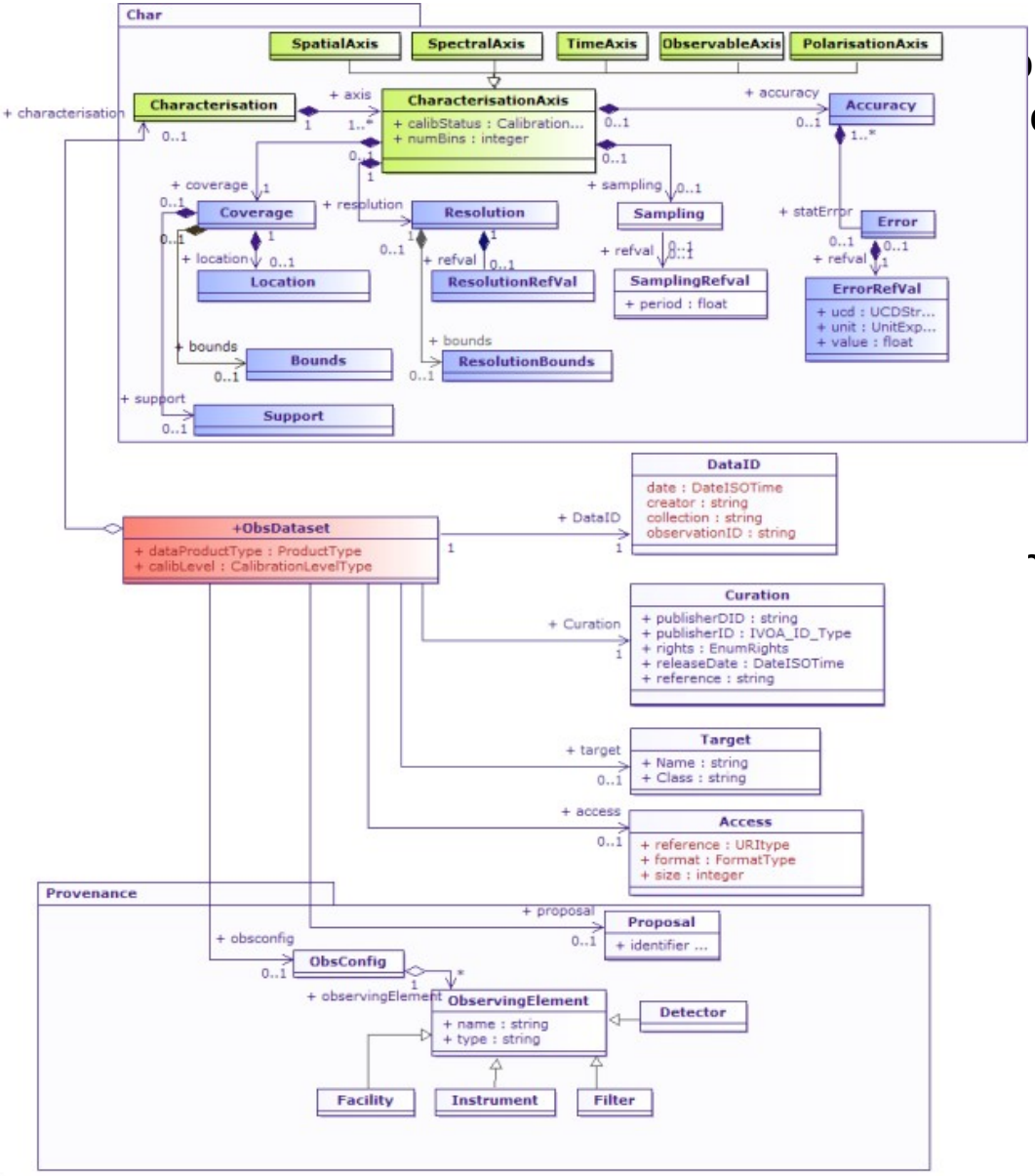
```
<?xml version="1.0" encoding="UTF-8"?>
<ri:Resource xsi:type="vstd:Standard" status="active"
             created="2012-02-17T11:15:00" updated="2012-02-17T11:15:00"
             xmlns:ri="http://www.ivoa.net/xml/RegistryInterface/v1.0"
             xmlns:vstd="http://www.ivoa.net/xml/StandardsRegExt/v1.0"
             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 Model

- 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: UML

- Core compo … perform data discovery wl … ons of interest
- Focus is on
  - A number
  - Aimed at
  - Broadcas
    - global
- Need to giv … ney can easily map to their
- http://www.ivoa.…



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

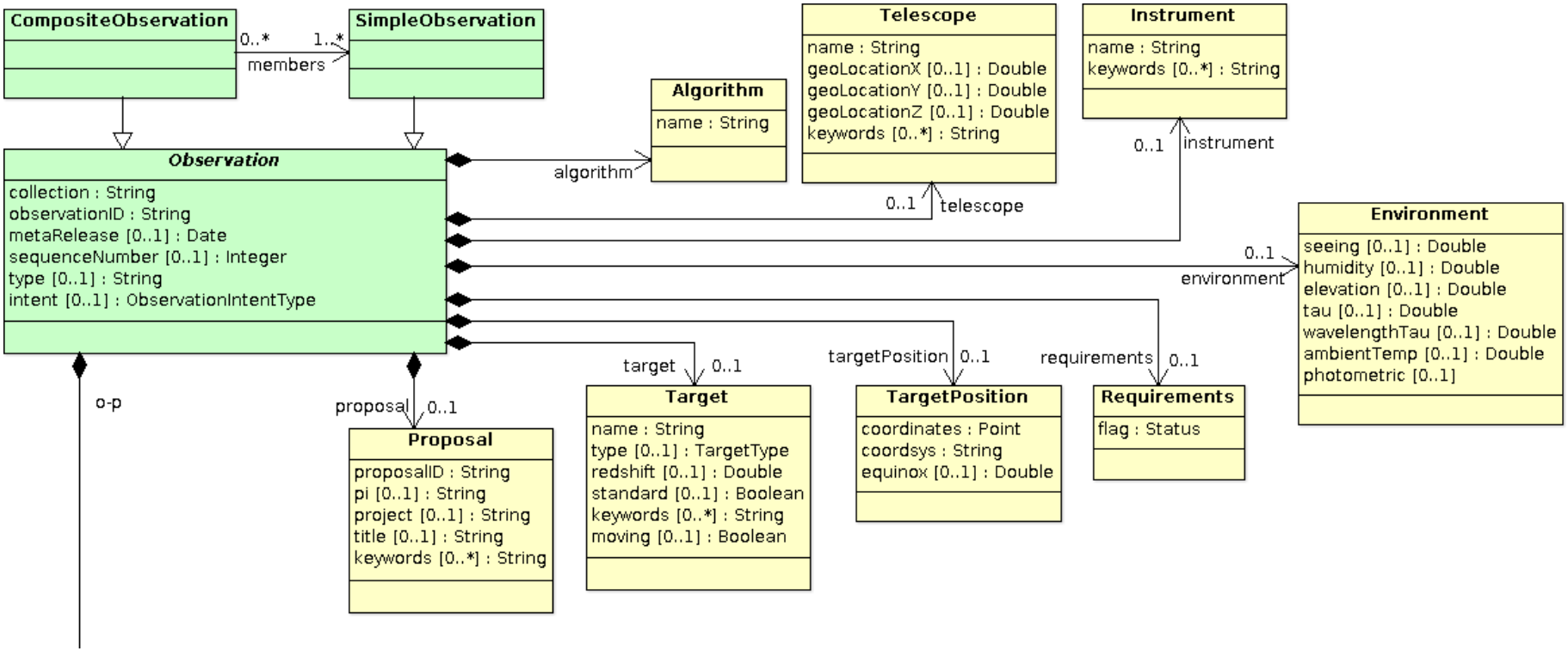
Column Name	*Unit*	*Type*	*Description*
dataproduct_type	unitless	String	Logical data product type (image etc.)
calib_level	unitless	enum integer	Calibration level {0, 1, 2, 3, 4}
obs_collection	unitless	String	Name of the data collection
obs_id	unitless	String	Observation ID
obs_publisher_did	unitless	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

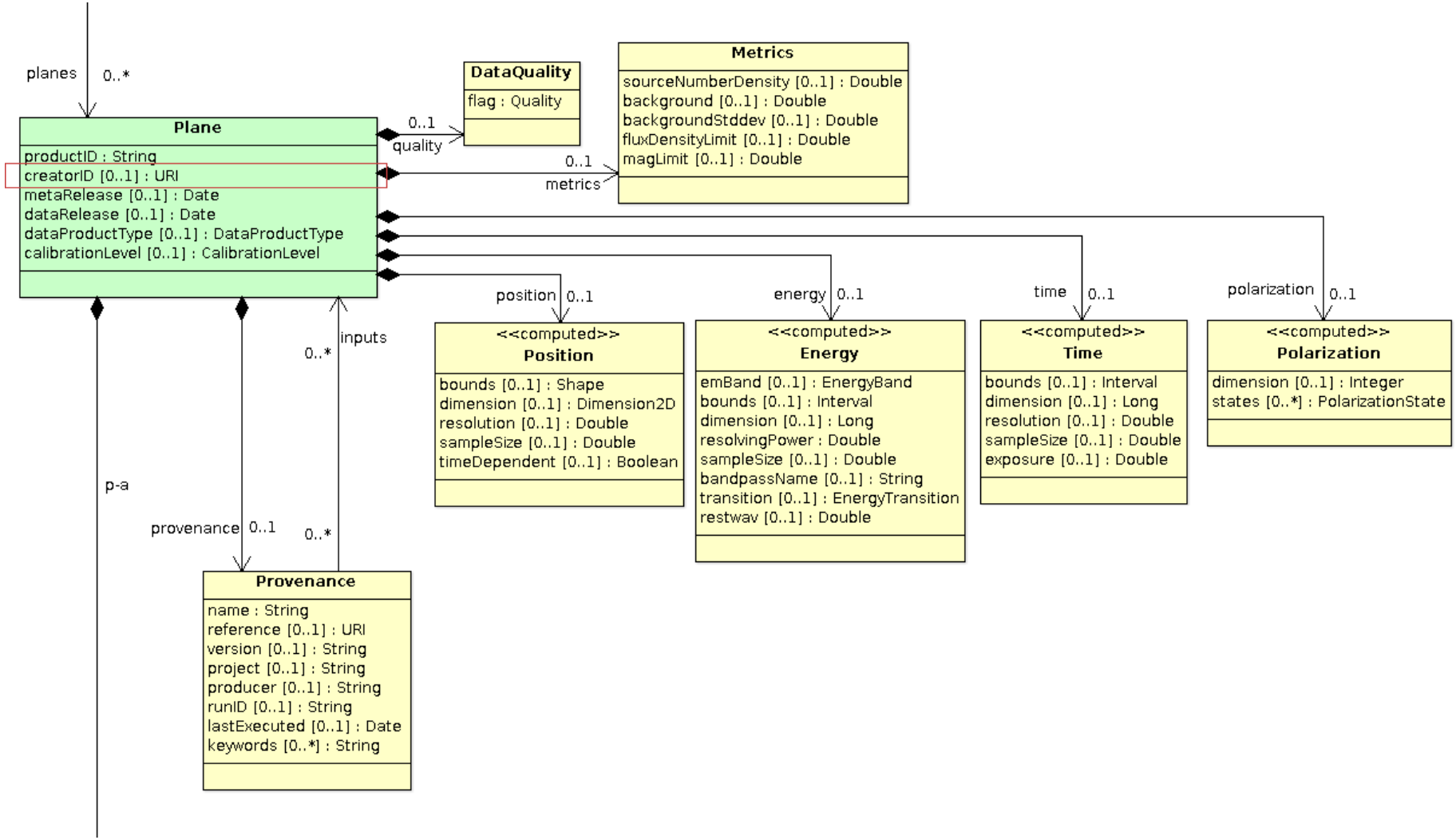
- Common Archive Observation Model, enables
  - Storage of observational metadata from the complete set of telescopic data
  - Searching through that metadata using a single interface
- The generalized capability of CAOM comes at the expense of some model complexity and the requirement of adopting a 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 (e.g. FITS header data unit)
    - 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
    ...
```

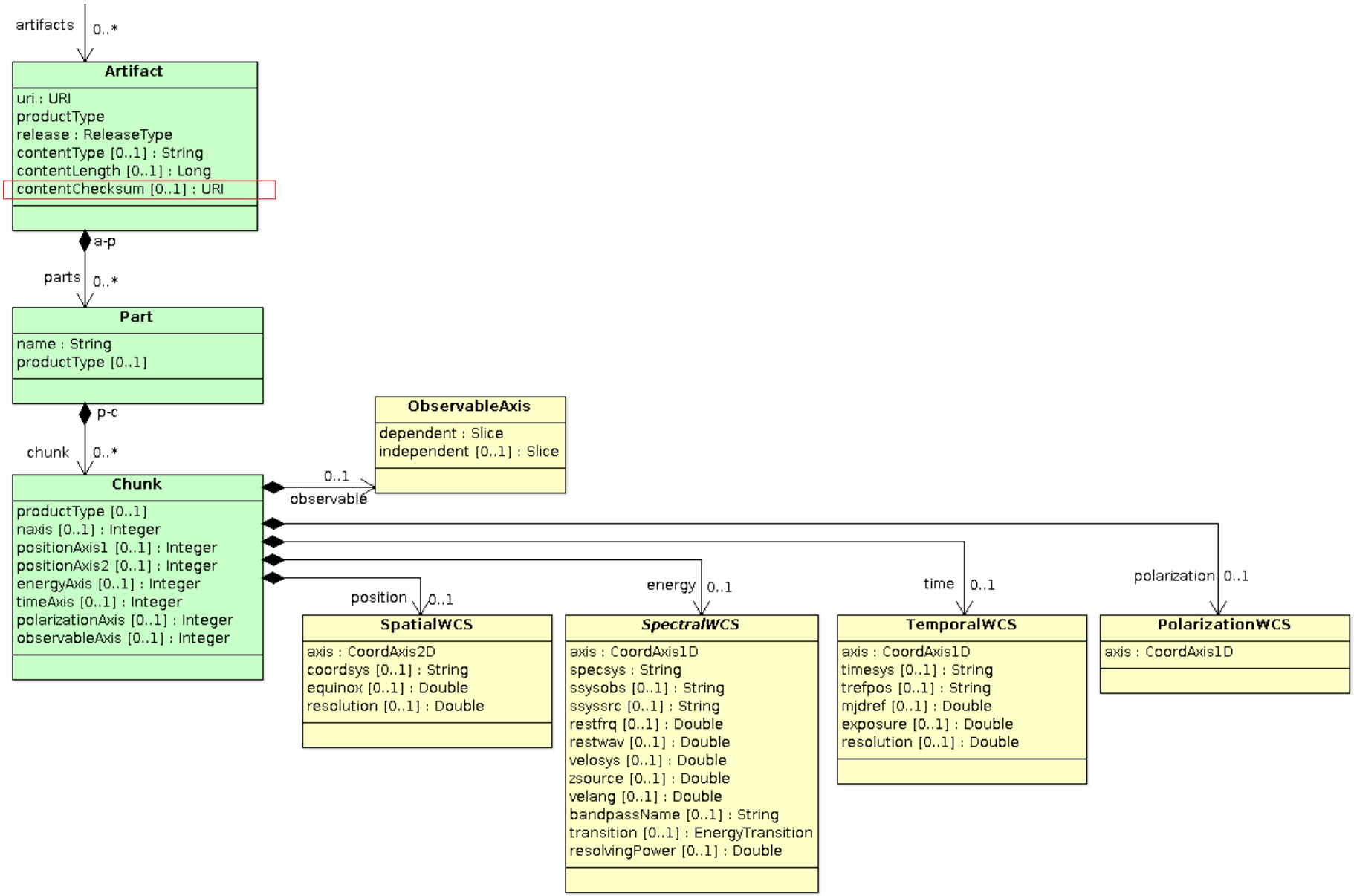
CAOM – Observation



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