NOAO Science Archive - Domain Model

Authors: NSA Team, NOAO

Contact: Phillip Warner (pwarner@noao.edu)

This document aims to present a current snapshot of the domain model for the NOAO Science Archive (NSA). The NSA will archive the datasets being produced by NOAO telescopes and pipelines, as well as data coming from partner organizations. The domain model offers a high level, conceptual view of the objects associated with these datasets, thereby providing a valuable resource to fulfill the requirements of the project in terms of an object-based methodology. This model also forms the basis over which the object model (implementation) and data model will be constructed.

This model is constructed using design elements defined by the UML (http://www.uml.org). It is assumed that the reader is familiar with the basic elements, i.e., packages, classes, associations (solid lines, often with an open arrowhead representing navigability), generalizations (solid lines, filled arrowhead), shared aggregations (solid lines, with unfilled diamonds at one end, an open arrowhead at the other end), composite aggregations (like shared aggregations, but with filled diamonds), and dependencies (dashed lines, with an open arrowhead representing navigability). Classes with italicized names are abstract, i.e., classes that are inherited by subclasses. Other elements include multiplicity, roles played by associated elements, and constraints.

Some attempt has been made to incorporate or make this model relatively consistent with the IVOA Quantity and Observation models. Although many similarities exist, there are many classes that do not map directly. Most differences arise due to the requirement for the NSA to store raw data. Although these elements defining raw data acquisition are important for the internal purposes of the NSA, they may be helpful in the context of the VO to ensure that all relevant domain elements are modeled. A model such as this may also help in understanding how VO services may interact with any kind of data product produced by other observatories. An overview of the model is as follows, and will reference the packages and classes found later in this document.

The NSA domain model attempts to provide a model for optical and infrared data acquisition. The model was derived from the classes and concepts contained in "Classes Describing Astronomical Observations", a document written by Frank Valdes (see http://iraf.noao.edu/projects/ccdmosaic/imagedef/classes.html). The data are represented by the class 'DataProduct' where one DataProduct is associated with a single physical manifestation, e.g., a FITS file. DataProducts are decomposed into six different classifications that include raw, calibrated, reduced, and analysis data products, as well as associated calibration data products. This decomposition has been established as a requirements for the NSA system. Examples of such products are included in the packages and diagrams contained in the top-level Data Products package.

The acquisition of a DataProduct is represented by two main components, namely DPGenerator and DPGenerationProcess. The elements that participate in the acquisition are referred to as "configuration elements" and are collected into a subclass of the high-level DPGenerator class. The DPGenerator, i.e., the combination of configuration elements, is used to generate a DataProduct; the information about this generation is incorporated into a class that inherits the DPGenerationProcess class. Associated with the DPGenerationProcess is the ProvenanceResource class, which represents any internal or external resource providing more details about the DPGenerationProcess and associated elements.

DPGenerator examples are contained in the Observing Facilities, Observation, and Processing packages. The Observing Facilities package contains class representations of many physical elements used at the telescope that are involved in creating the DataProduct. Observation is assocated with the ObservationConfiguration class, which provides storage for dynamic configurations of Observing Facilities elements. Processing elements represent external (with respect to the NSA) processing pipelines. Information about these DPGenerators is static, and is used in the definition of the provenance of a DataProduct.

DPGenerationProcess examples are contained in the Observation and Processing packages. Observation (from the Observation package) collects information about, and is associated with a single DataProduct. Processing (from the Processing package) contains various pipeline runs (indicated by the suffix "PipelineRun"; "Pipeline" is the suffix for each type of pipeline), which are associated with one or more DataProducts; that is to say, one or more DataProducts may be used as input, while only one DataProduct, in the role of output, may be associated with a PipelineRun.

DPGenerationProcess is also associated with a top-level State element, which successively contains any number of states (where relevant) associated with the non-null elements of DPGenerator. Other elements associated with a DPGenerationProcess are User, Proposal, and ProvenanceResource. The User association represents the person or entity that initiates the DPGenerationProcess, e.g., the observer. The Proposal assocation provides the necessary information needed to determine which authorized persons or entities have proprietary access to the DataProduct. The ProvenanceResource association provides references to NOAO-internal or external resources that are related to the associated DataProduct, e.g., an abstract, a bibliography, observing logs, source code version repository, etc.

The combination of DPGenerationProcess, DPGenerator, DataProduct, State, User, Proposal, and ProvenanceResource form a node in what is called the Provenance Graph for a DataProduct. This Provenance Graph is used to establish the ancestry of each DataProduct, with the DPGenerationProcess, e.g., the Observation, being the top parent. For more details, see the diagram in the Provenance section.

The NSA Package Diagram contains the top level packages for the domain. Dependencies between packages are shown using the dashed arrow, as defined in the UML standard. Some dependencies are not shown, as they are implied through the dependencies of the other packages.

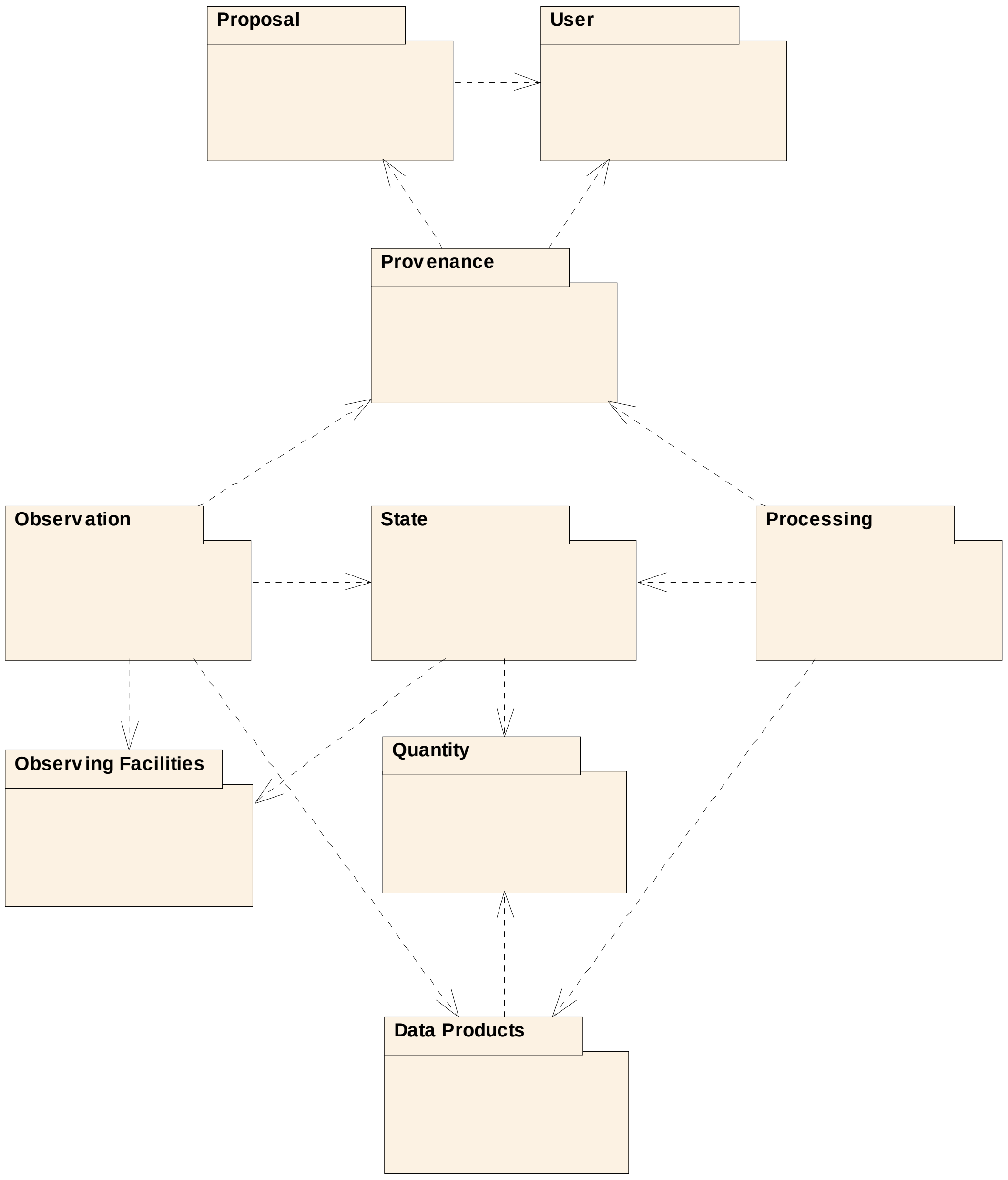


Figure 1 : NSA Package Diagram

Provenance

The provenance for any given DataProduct in the NSA system is represented by the Provenance class diagram. Any process that generates a DataProduct can be represented by a specialization of the DPGenerationProcess. DPGenerationProcess is the top-level class of the class association that forms the Provenance Node. The Provenance Graph, represented by each Node and associations between each Node, describes the ancestry of a given DataProduct. The top-level parent, from which all DataProducts are derived, can be discovered by traversing the Provenance Graph.

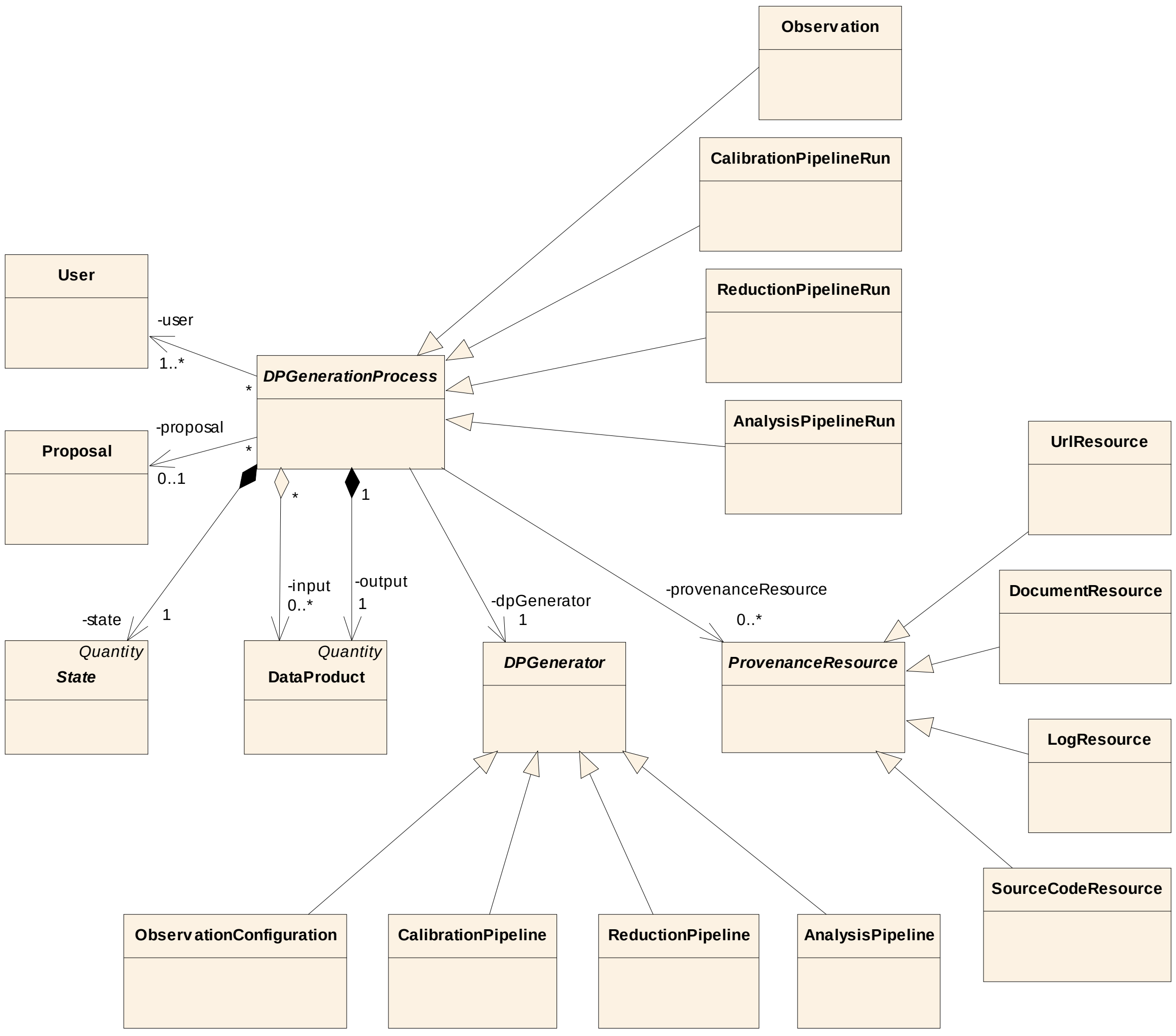


Figure 2 : Provenance Class Diagram

DocumentResource

DocumentResource is a ProvenanceResource that provides a summary and/or a local or external URL to documentation describing the DPGenerationProcess or related information.

DPGenerationProcess

DPGeneratorProcess contains information about the processes by which a DPGenerator creates a DataProduct. Information could include, e.g., pipeline logs, observation logs, etc. Classes that inherit from a DPGeneratorProcess contain specific information about the generation of a given DataProduct.

DPGenerator

The DPGenerator abstracts any class that represents an object capable of generating a DataProduct. Examples of such a class are: instrument, calibration pipeline, data reduction pipeline, etc.

LogResource

LogResource is a ProvenanceResource that contains information regarding the logs from a DPGenerationProcess. This may include, e.g., observer logs, pipeline logs, etc.

ProvenanceResource

ProvenanceResource contains information leading to further details of a DPGenerationProcess.

SourceCodeResource

SourceCodeResource is a ProvenanceResource that contains information about source code used, e.g., in a pipeline, including the source itself. Version information is stored in an instance of this class. The resource could be a software and/or configuration file repository.

UrlResource

UrlResource is a type of ProvenanceResource that contains a summary and a URL pointing to a possibly external resource that provides more information regarding the DPGenerationProcess. Examples include a URL to an abstract service or reference documentation, etc.

Proposal

The Proposal package contains the Proposal class, which is dependent upon the User class. The Proposal class contains all information about the proposal that is associated with the Data Product. This information is essential to provide proprietary access to, and to build a provenance for each Data Product.

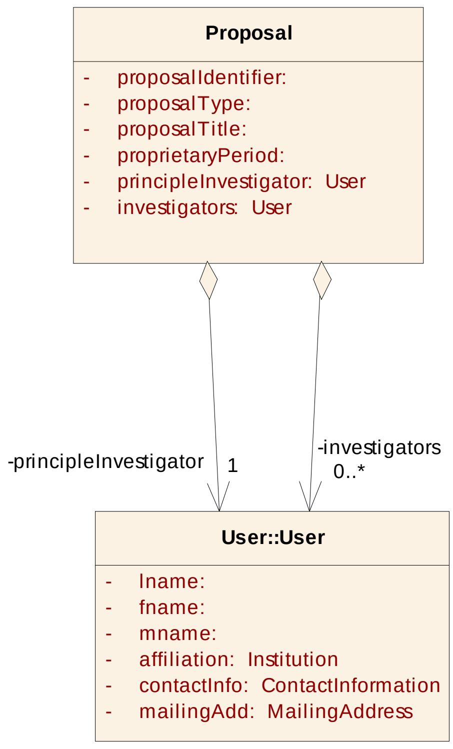


Figure 3 : Proposal Class Diagram

Proposal

The Proposal class incorporates a minimal amount of information from the NOAO proposal database into the NSA system.

User

The User Package contains all classes that store user and user-related information. In the context of the NSA, a User is a registered entity that is given access to all data, including the data for which the User has proprietary access rights. Anonymous users are not considered Users, but still have limited access to data contained within the NSA. Limited access is typically defined as read-only, i.e., retrieval, access to public data. The NSA has plans to provide further access (e.g., to processing services, etc.) to registered users of the NSA.

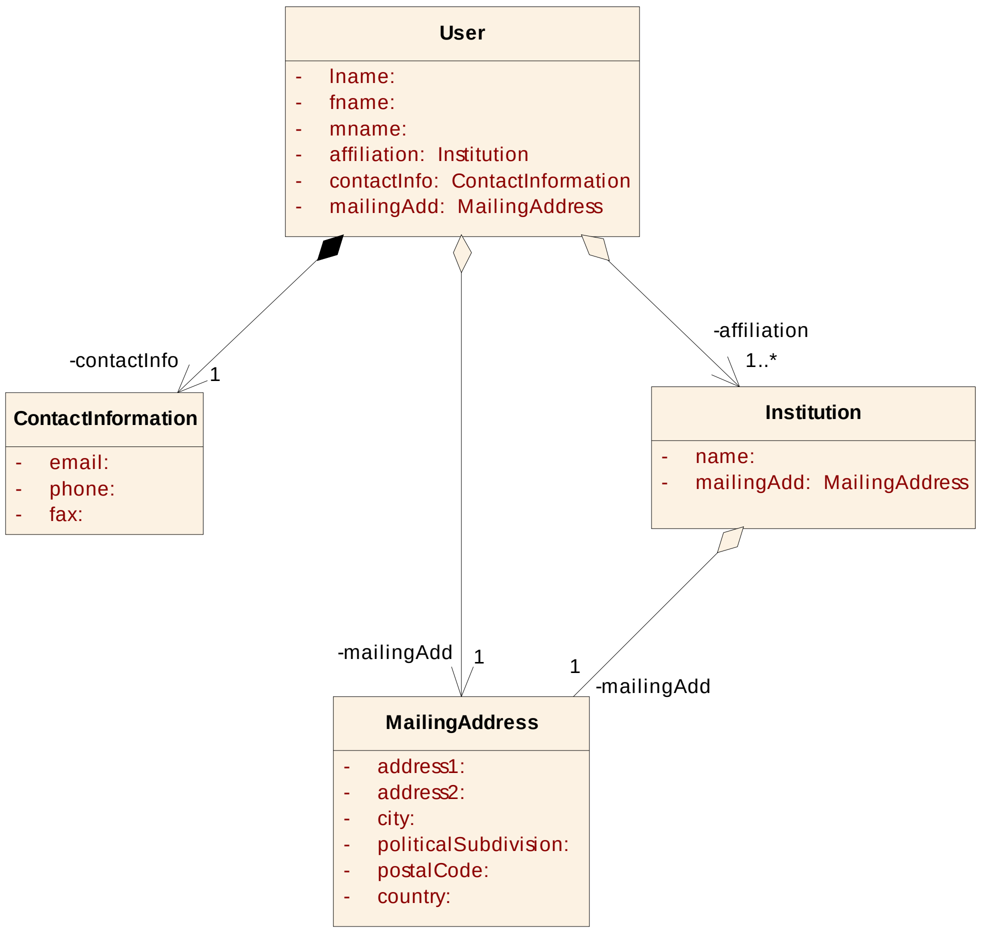


Figure 4 : User Class Diagram

ContactInformation

Contains contact information for a User.

Institution

Contains information about the institution(s) with which a User is associated.

MailingAddress

Contains the mailing address information of a User and/or an Institution. The MailingAddress for an Institution may be identical to that of a User.

User

User contains information about a user of the NSA system. A User that is associated with a Proposal has partial or full access to data taken for that Proposal, depending upon the access rights given by that Proposal's principle investigator. If a User is a principle investigator on the Proposal, then that User has full access rights to all data taken for that Proposal. A User may also be an observer. If a User is not an investigator on the proposal, the User will likely have access to data acquired while he or she was at the telescope. However, access to all other data acquired for the given proposal is authorized only by the proposal's principle investigator.

Observation

Observation contains references to all classes associated with a Data Product, i.e., target, observer, permissions, and provenance information. Provenance information includes all classes that represent the elements of the configuration of the observing facilities and the states thereof (where relevant). An Observation, which represents the act of making a single observation at the telescope, collects this information for a single DataProduct.

Observation groups everything that is associated with a DataProduct. That is to say, any domain object that participates in any way in the generation of a DataProduct is contained by an Observation.

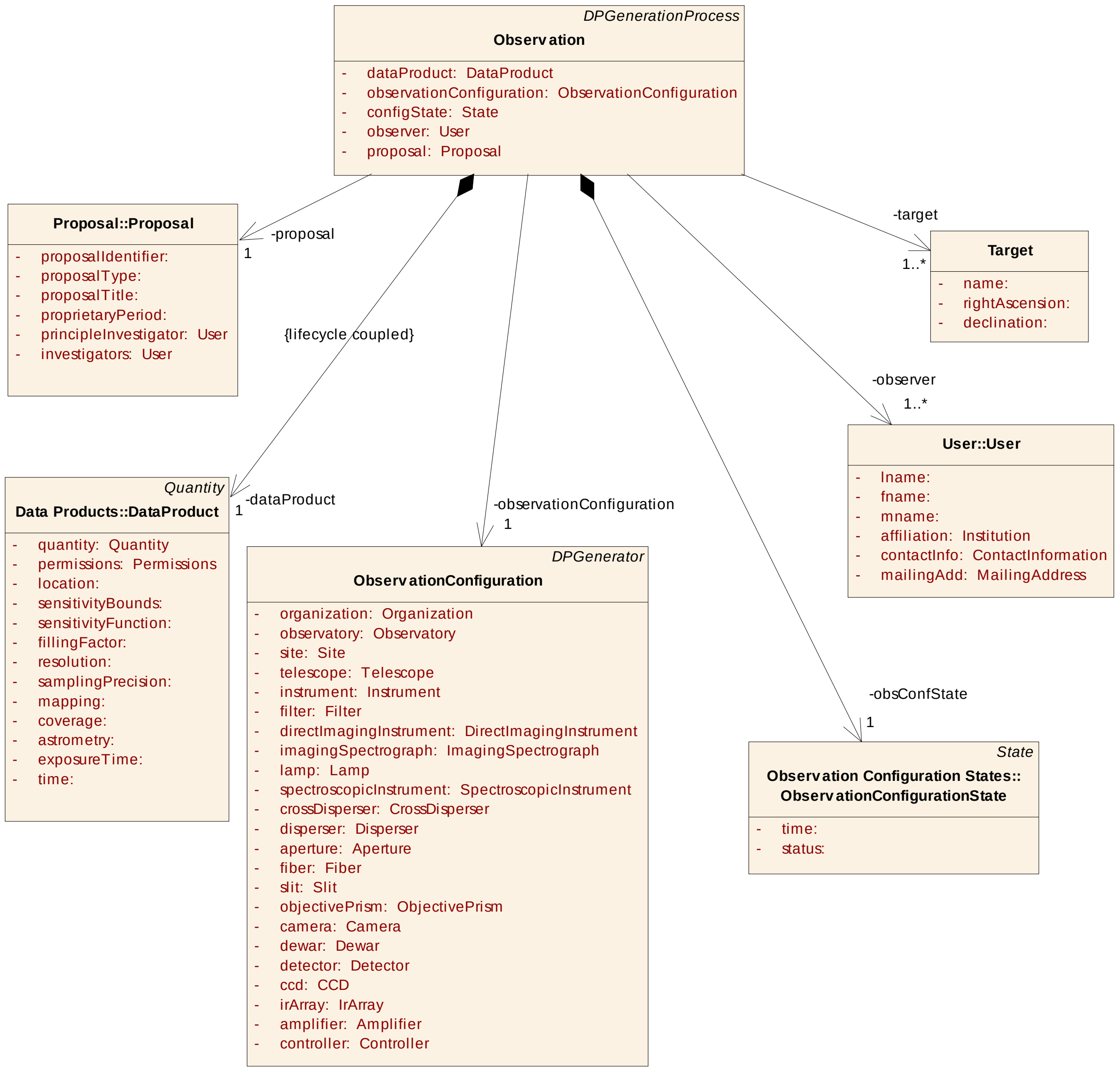


Figure 5 : Observation Class Diagram

Observation

The Observation class is a container for all high-level classes that participate in the acquisition of a raw DataProduct, and includes a reference to the DataProduct. Each high-level class itself may contain classes that are decompositions of the high-level class. Observation is a DPGenerationProcess, or a process by which a raw DataProduct is created.

ObservationConfiguration

ObservationConfiguration contains references to all classes that participated (in some fashion) in the creation of a raw DataProduct, i.e., it provides a provenance or historical record of these classes. The configuration of the acquisition elements has no hierarchical structure inside this class, as some attributes may be irrelevant to the given type of observation, as in the case of, e.g., imaging versus spectroscopy. It is sufficient that we capture, in the implementation of the system, each element and the status thereof, rather than the association between the classes, as there is an infinite number of possible configurations.

Target

Target contains information about the intended target of an Observation.

Observing Facilities

The Observing Facilities package includes information about the provenance of the raw Data Product. The structure of class associations is similar to the structure of the associations of the actual physical devices. The relationships are not intended to hold in all cases, but may hold for many. The purpose of the hierarchical structure shown in this package is to provide an overview of the domain, as well as to facilitate the inclusion of all possible classes.

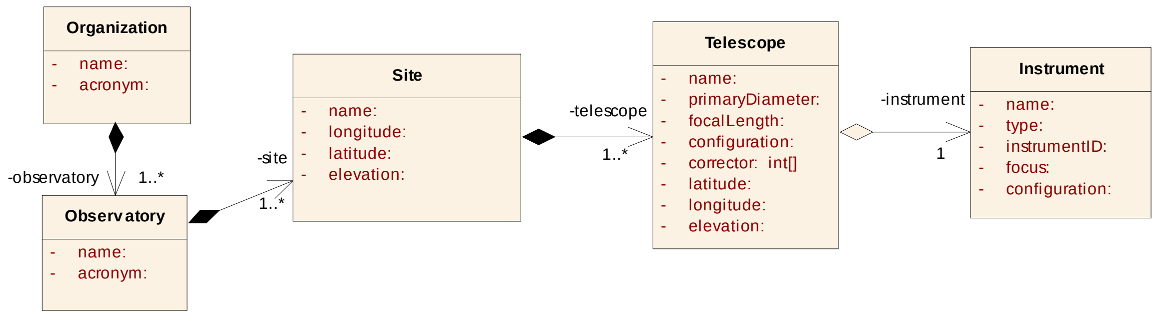


Figure 6 : Observating Facilities Class Diagram

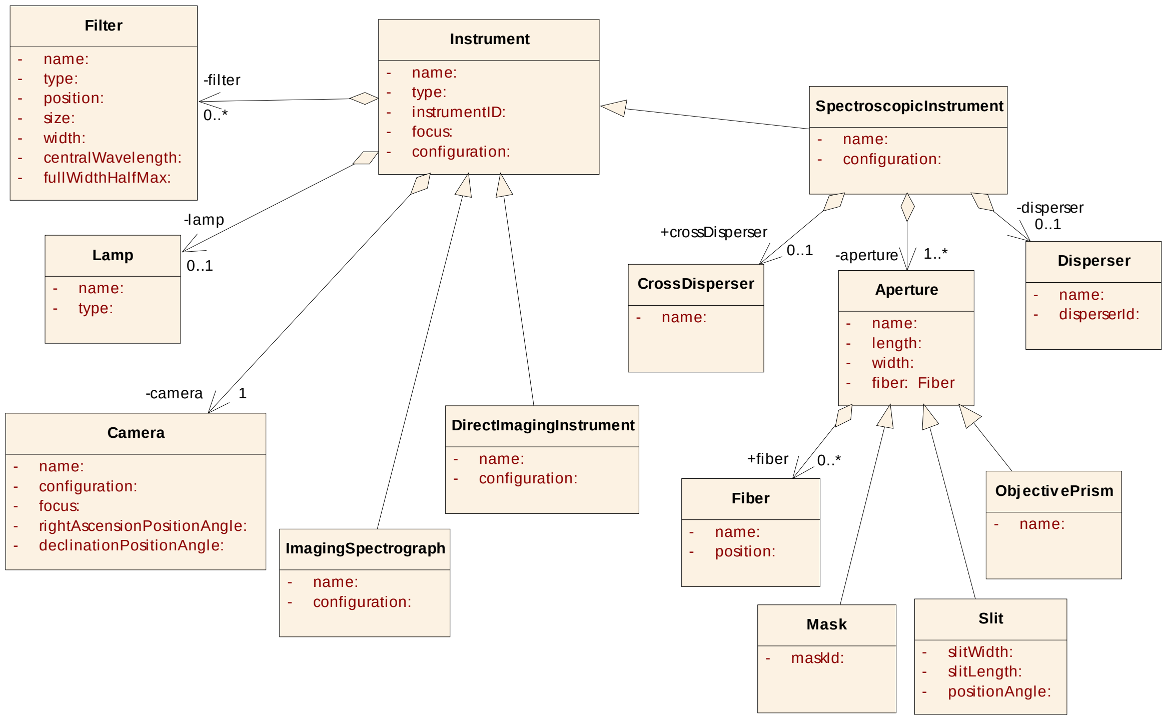


Figure 7 : Instrument Class Diagram

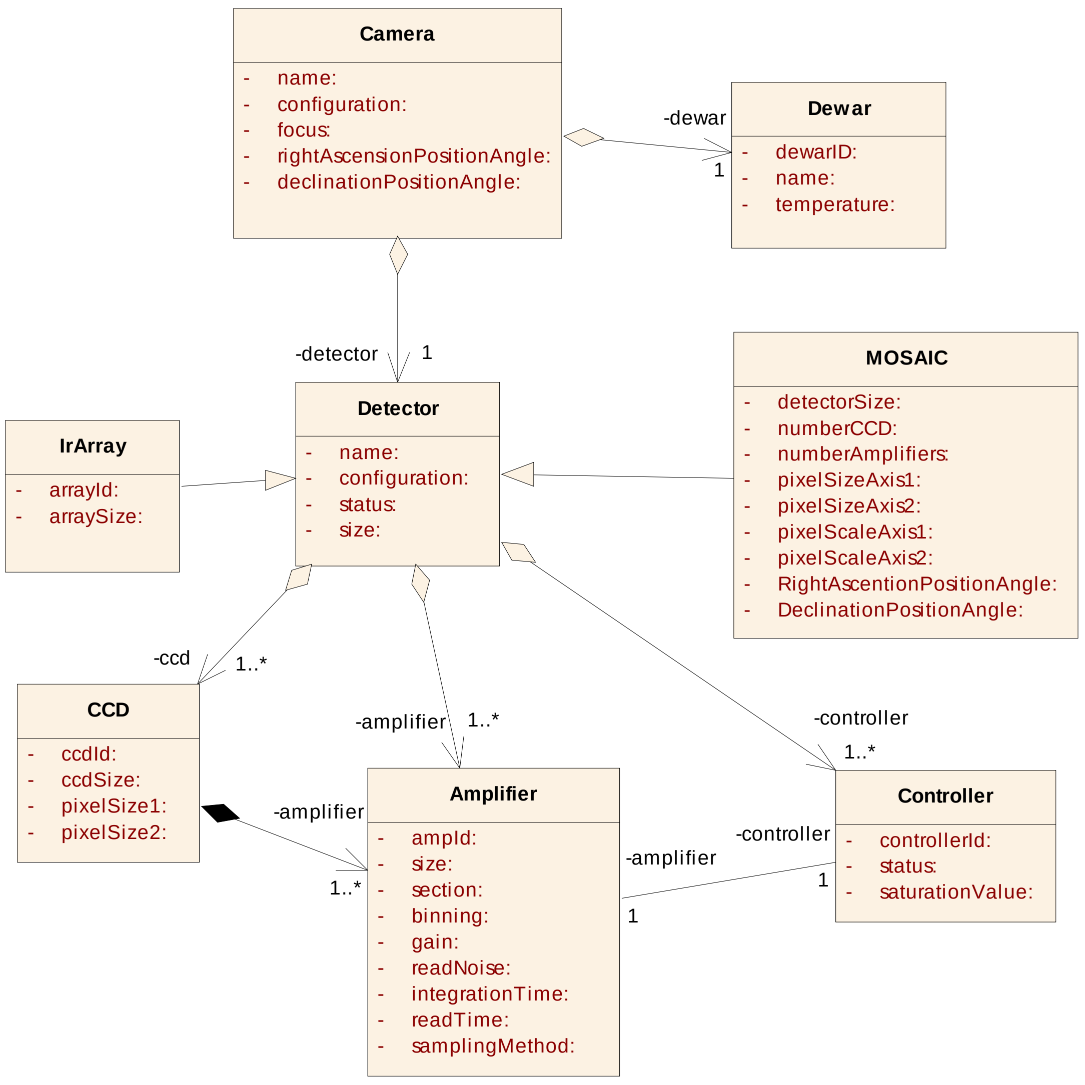


Figure 8 : Camera Class Diagram

Mask

Mask is an Aperture that contains information about the mask used during the acquisition of the DataProduct.

Amplifier

Amplifier contains information about the amplifier used by the Detector during a DataProduct acquisition.

Aperture

Aperture describes the aperture that was used in a DataProduct acquisition.

Camera

Camera contains information about the camera used in a DataProduct acquisition.

CCD

CCD contains information about the CCD used during a DataProduct acquisition.

Controller

Controller contains information about the controller used by the Detector during a DataProduct acquisition.

CrossDisperser

CrossDisperser contains information about the cross disperser used during a DataProduct acquisition.

Detector

Detector contains information about the detector used during a DataProduct acquisition.

Dewar

Dewar contains information about the dewar used during a DataProduct acquisition.

DirectImagingInstrument

An instrument with a combination of camera, filter, detector.

Disperser

Disperser contains information about the disperser used during a DataProduct acquisition.

Fiber

Fiber is an Aperture with specific attributes related to the fiber used during a DataProduct acquisition.

Filter

Filter contains information about the filter used during a DataProduct acquisition.

ImagingSpectrograph

An instrument used to obtain DataCube data products. Examples of such an instrument are: Fabry Perot, IFU, etc.

Instrument

Instrument contains information about the physical instrument.

IrArray

IrArray is a Detector that contains specific information about the IR array used during a DataProduct acquisition.

Lamp

Lamp contains information about the lamp that may have been used during the DataProduct acquisition.

MOSAIC

MOSAIC is a Detector, and contains information specific to the MOSAIC detector used at the 4-meter telescope on KPNO.

ObjectivePrism

ObjectivePrism is an Aperture that contains specific information about the objective prism used during the DataProduct acquisition.

Observatory

Observatory represents the political entity that has the burden of responsibility for management of the different sites. An example an Observatory is "NOAO".

Organization

Organization contains information about the top-level organization, as a political entity, that accepts responsibility for one or more observatories. The Organization is the holder of information, or has the information, about the ownership of a DataProduct.

Site

Site is the geographical location of a given set of telescopes. Examples include "Kitt Peak", or "Cerro Tololo".

Slit

Slit is an Aperture that contains specific information about the slit that was used during the DataProduct acquisition.

SpectroscopicInstrument

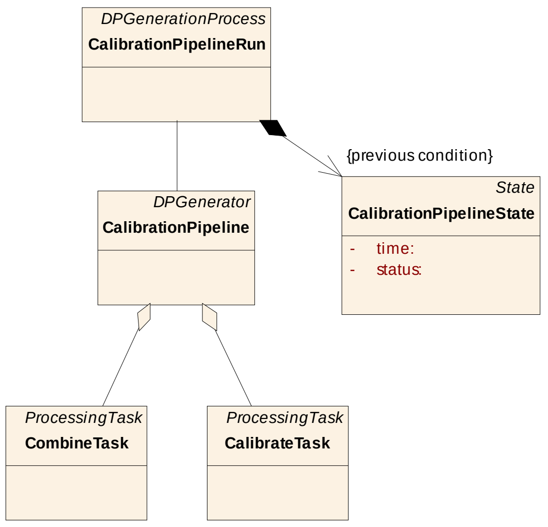
SpectroscopicInstrument inherits all attributes and methods of the Instrument class. This class contains information specific to this type of Instrument. Examples of such an instrument are: Echelle Spec., RC Spec, Hydra, Coudé Feed.

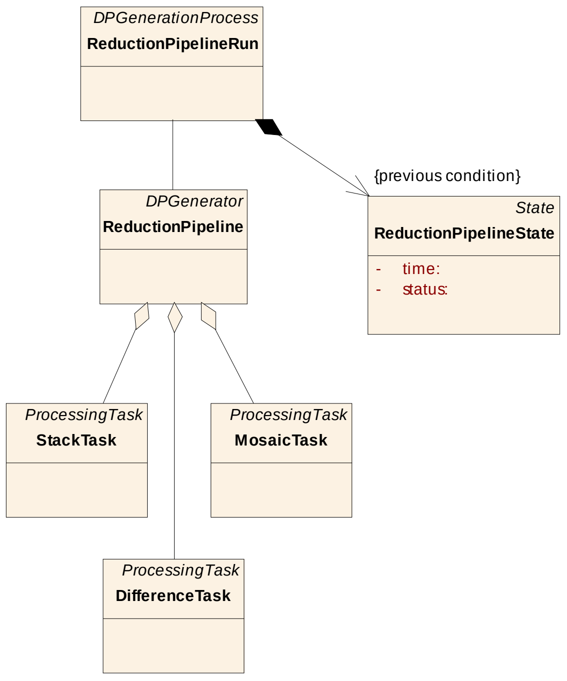
Telescope

Telescope contains information about the physical telescope.

Processing

Processing contains classes associated with various pipeline processes. Each group is modeled after the Provenance assocations. As shown in the Processing diagram, the elements that constitute a Pipeline are the individual tasks of that Pipeline.





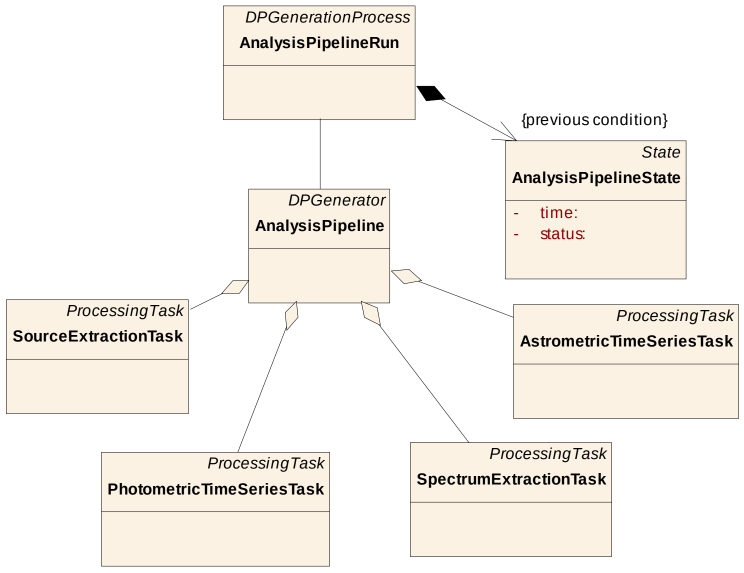


Figure 9 : Processing Class Diagram

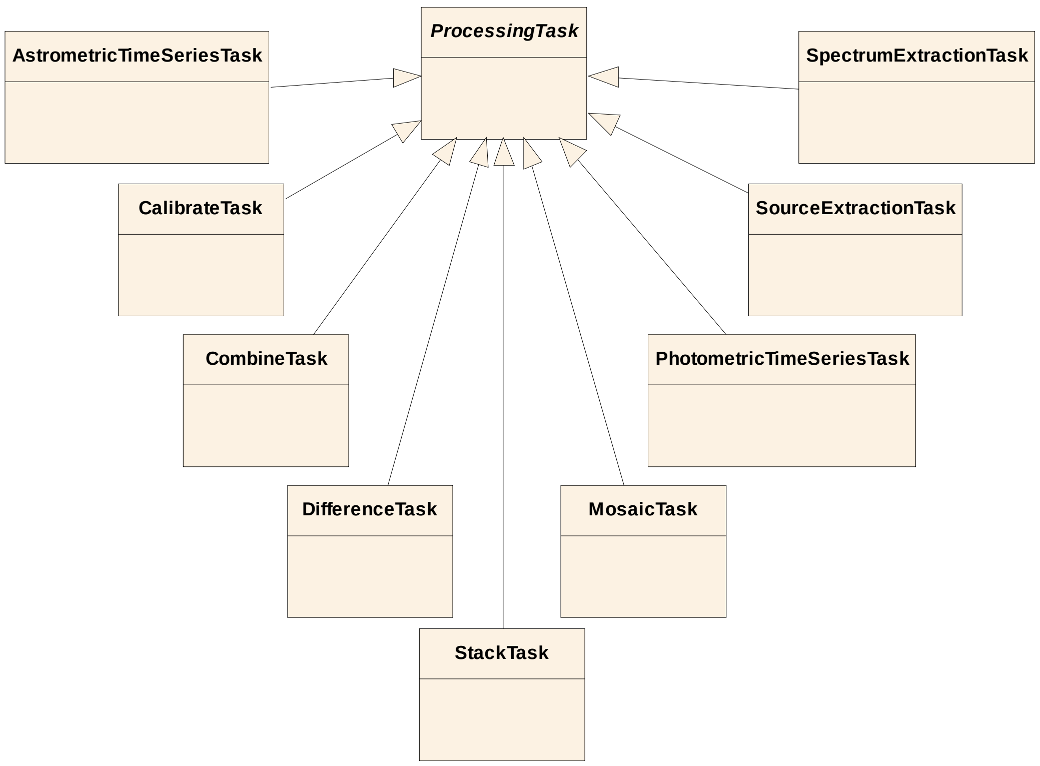


Figure 10 : Processing Task Class Diagram

AnalysisPipeline

AnalysisPipeline contains information about a pipeline that analyzes a DataProduct, e.g., documentation, algorithms used, configuration, etc.

AnalysisPipelineRun

Contains information about a given run of the AnalysisPipeline i.e., a set of tasks provided by the AnalysisPipeline.

AstrometricTimeSeriesTask

AstrometricTimeSeriesTask extracts astrometric information from a list of images and provides a temporal and spatial view of that information.

CalibrateTask

CalibrateTask uses the output from CombineTask to calibrate the raw science DataProduct, thereby creating a Level 2 DataProduct.

CalibrationPipeline

Contains information about a pipeline that calibrates a DataProduct, e.g., documentation, algorithms used, configuration, etc.

CalibrationPipelineRun

Contains information about a given run of the CalibrationPipeline, i.e., a set of tasks provided by the CalibrationPipeline.

CombineTask

CombineTask combines one or more images. Images should typically be, e.g., raw flats, darks, and biases. The output of the CombineTask should be a DataProduct for each CombineTask, e.g., a superflat, superdark, and superbias.

DifferenceTask

DifferenceTask creates a difference image of several images. The resultant DataProduct could be an MEF, or a Single-Image file.

MosaicTask

MosaicTask creates a mosaic of several images. The resultant DataProduct could be an MEF, or a Single-Image file.

PhotometricTimeSeriesTask

PhotometricTimeSeriesTask extracts photometric information from a list of images or spectra and provides a temporal view of that information.

ProcessingTask

The ProcessingTask class is a top-level, generic representation of a task used in pipeline processing. All tasks within the pipeline framework inherit the attributes and methods from this class.

ReductionPipeline

Contains information about a pipeline that reduces a DataProduct, e.g., documentation, algorithms used, configuration, etc.

ReductionPipelineRun

Contains information about a given run of the ReductionPipeline i.e., a set of tasks provided by the ReductionPipeline.

SourceExtractionTask

SourceExtractionTask extracts a list of astronomical sources from level 2 or 3 DataProducts.

SpectrumExtractionTask

SpectrumExtractionTask extracts a 1-D spectrum from calibrated or reduced spectra. The form of the output could be a table.

StackTask

StackTask creates a stack of several images. The resultant DataProduct could be an MEF, or a Single-Image file.

Quantity

The Quantity package contains the Quantity class, as defined by the IVOA. No further decomposition of the Quantity is shown here, as currently the Quantity model only contains interfaces.

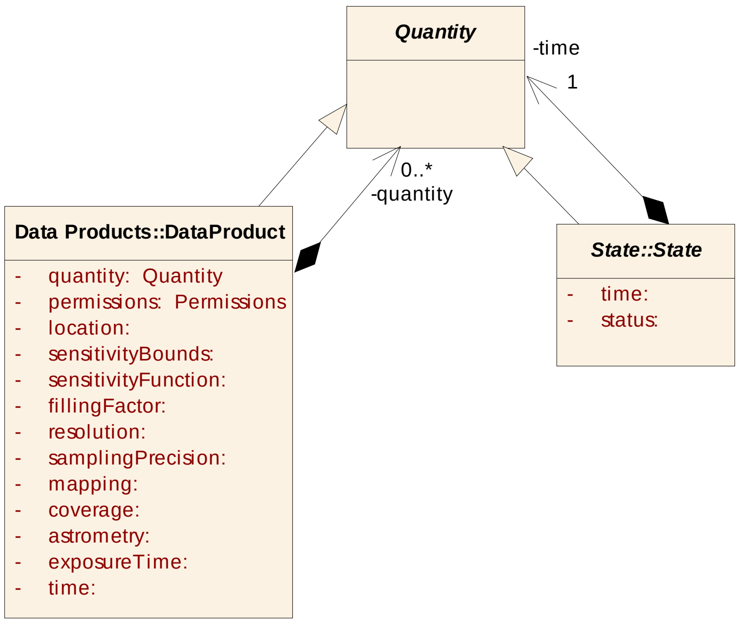


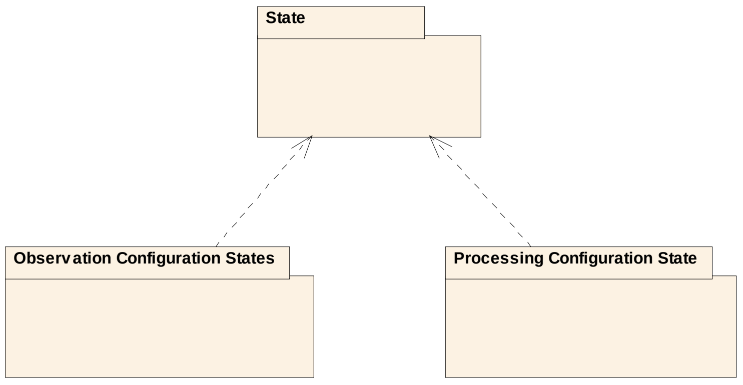
Figure 11 : Quantity Class Diagram

Quantity

IVOA Quantity class; a generic interface that represents any type of Quantity object. Attributes of this class are defined by the Data Model subgroup responsible for the Quantity Model.

State

State contains all classes associated with State classes.



State
Observation Configuration States
Processing Configuration State

Figure 12 : State Package Diagram

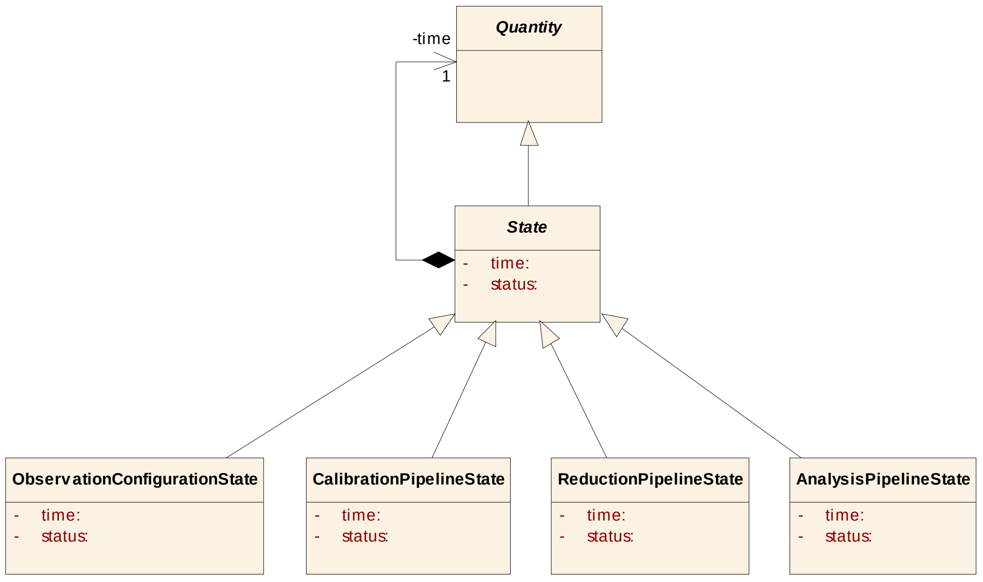


Figure 13 : State Class Diagram

State

The State class contains the status of each element associated with a DPGenerator, at the time of acquisition of the DataProduct.

Observation Configuration States

Observation Configuration States contains all State classes associated with each Observation Configuration element, if relevant.

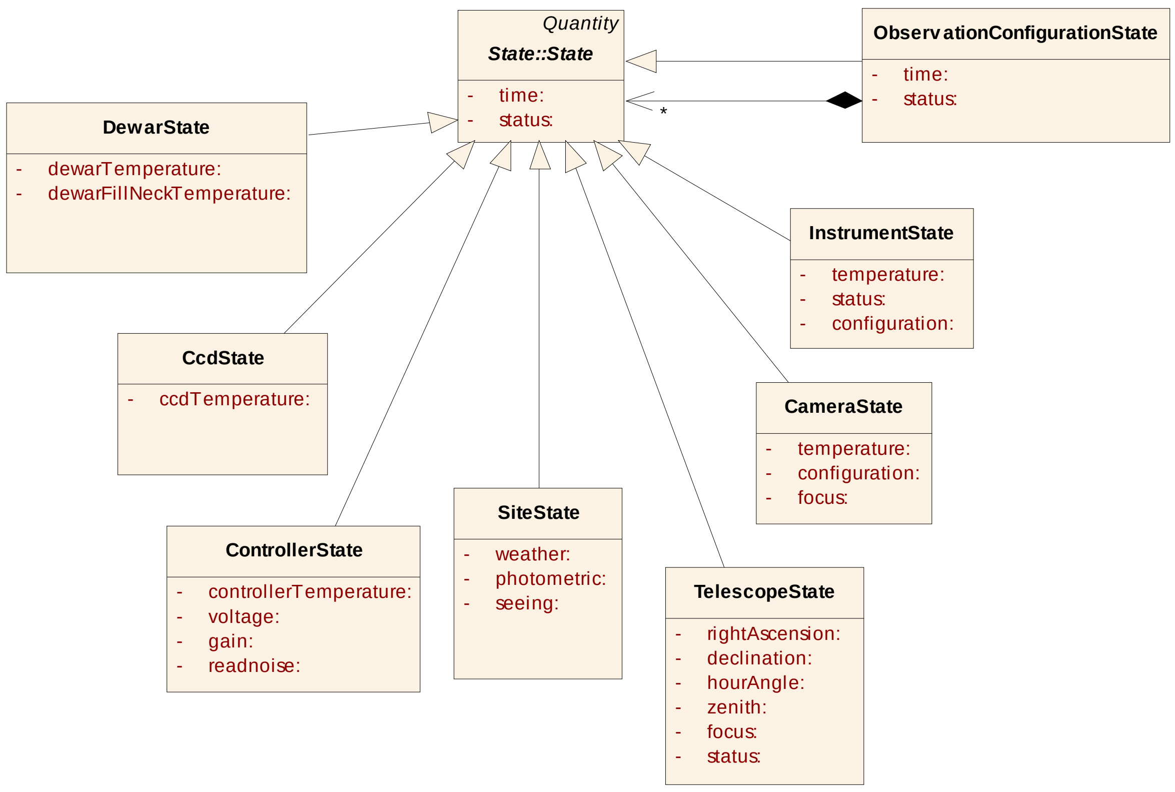


Figure 14 : Observation Configuration States Class Diagram

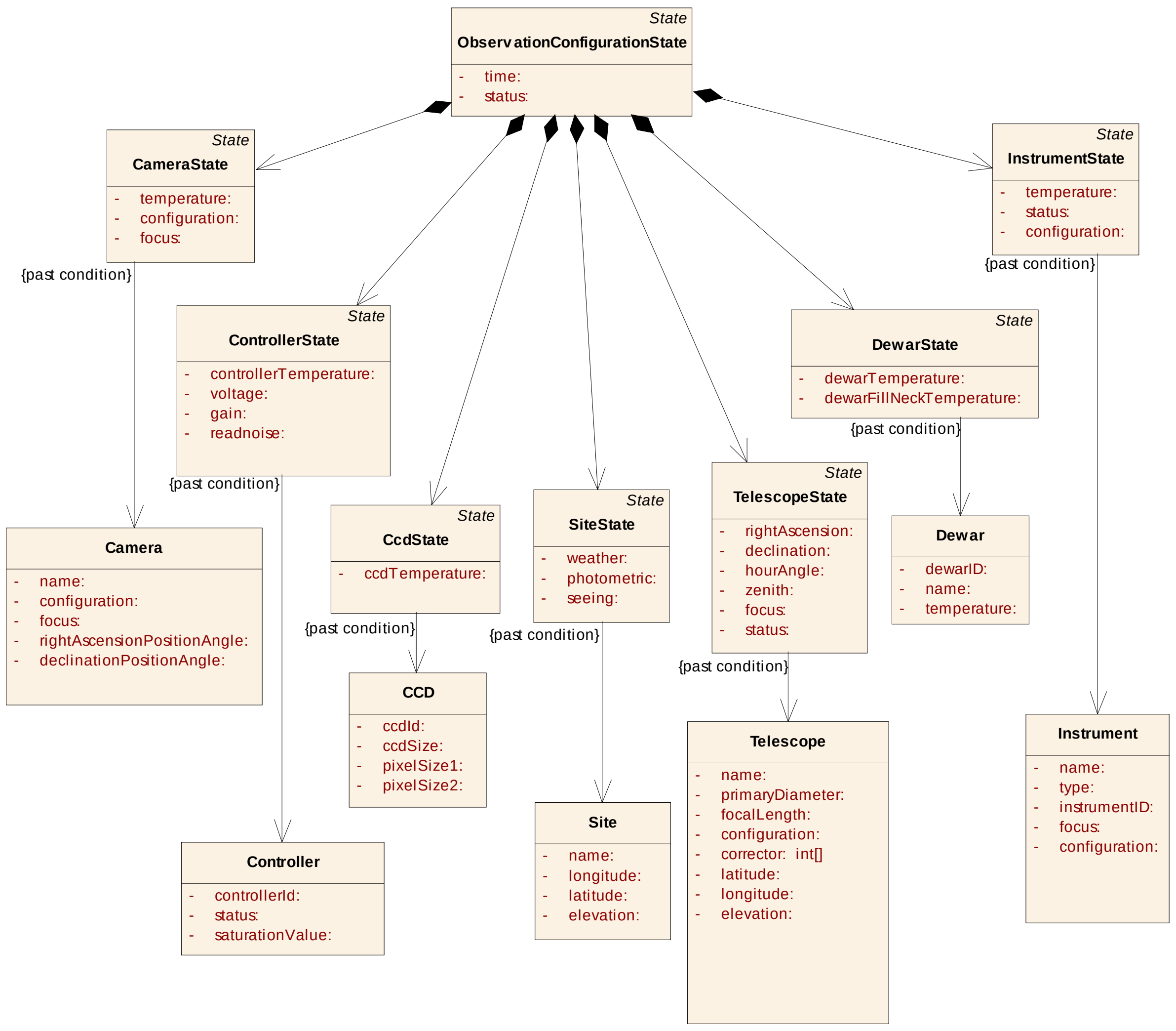


Figure 15 : Observation Configuration Class Diagram

CameraState

Contains the state information of the referenced Camera at the time of the DataProduct acquisition.

CcdState

Contains the state information of the referenced CCD at the time of the DataProduct acquisition.

ControllerState

Contains the state information of the referenced Controller at the time of the DataProduct acquisition.

DewarState

Contains the state information of the referenced Dewar at the time of the DataProduct acquisition.

InstrumentState

Contains the state information of the referenced Instrument at the time of the DataProduct acquisition.

ObservationConfigurationState

ObservationConfigurationState is a State that contains the overall status of a given observing configuration. This class aggregates other classes that represent the status of each element associated with the ObservationConfiguration. Each state is recorded at the moment in time a DataProduct was acquired. This may not include all elements, as not all elements actually emit a status.

SiteState

Contains the state information of the referenced Site at the time of the DataProduct acquisition.

TelescopeState

Contains the state information of the referenced Telescope at the time of the DataProduct acquisition.

Processing Configuration State

Processing State contains all classes associated with the state of any pipeline process.

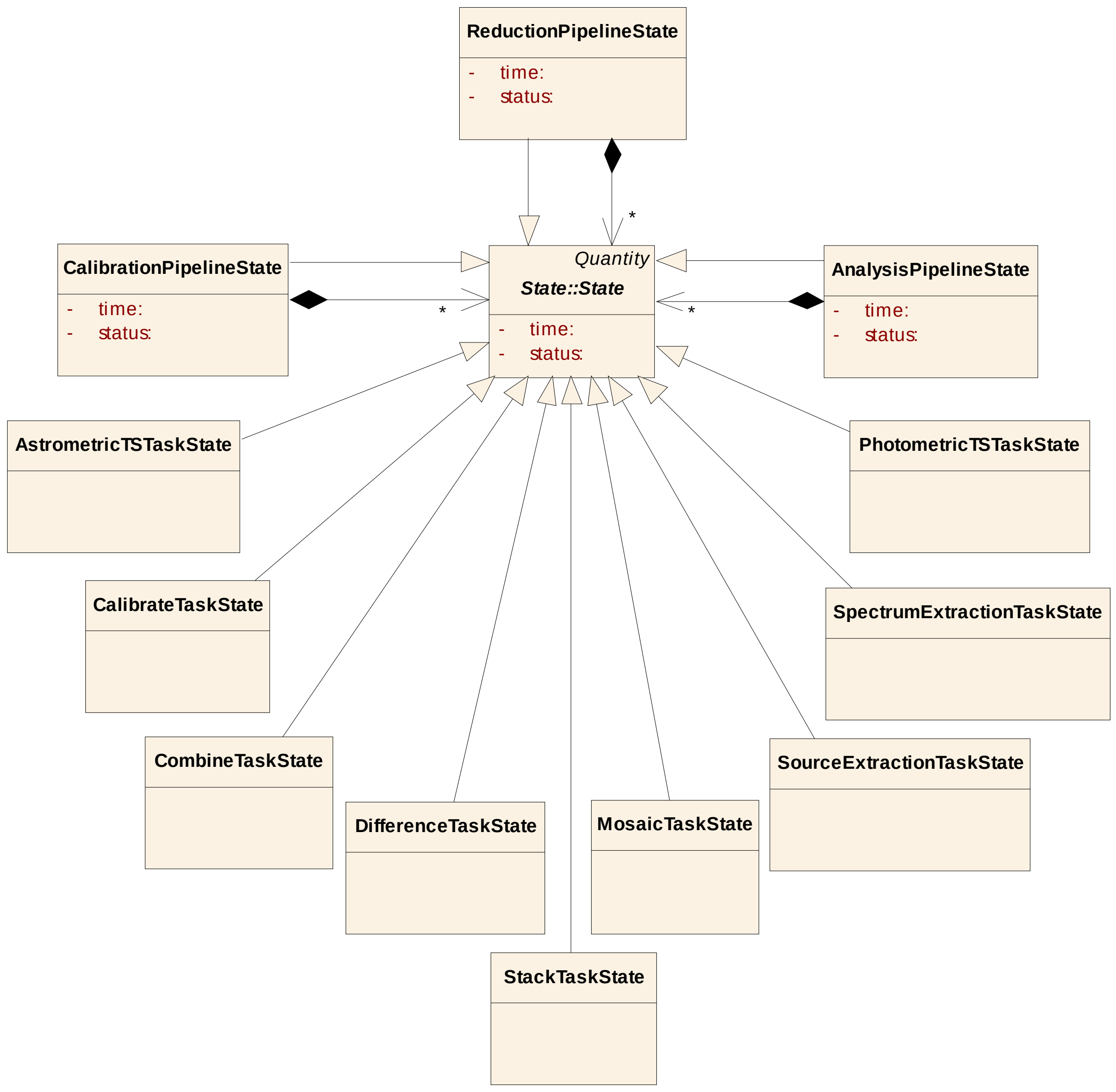


Figure 16 : Processing Configuration State Class Diagram

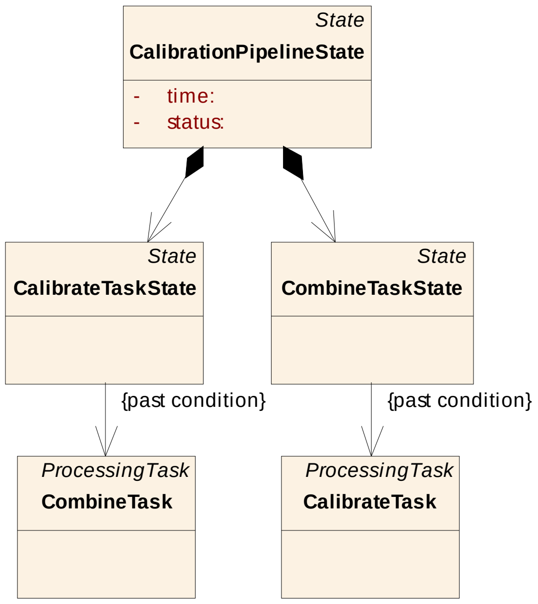


Figure 17 : Calibration Pipeline Configuration Class Diagram

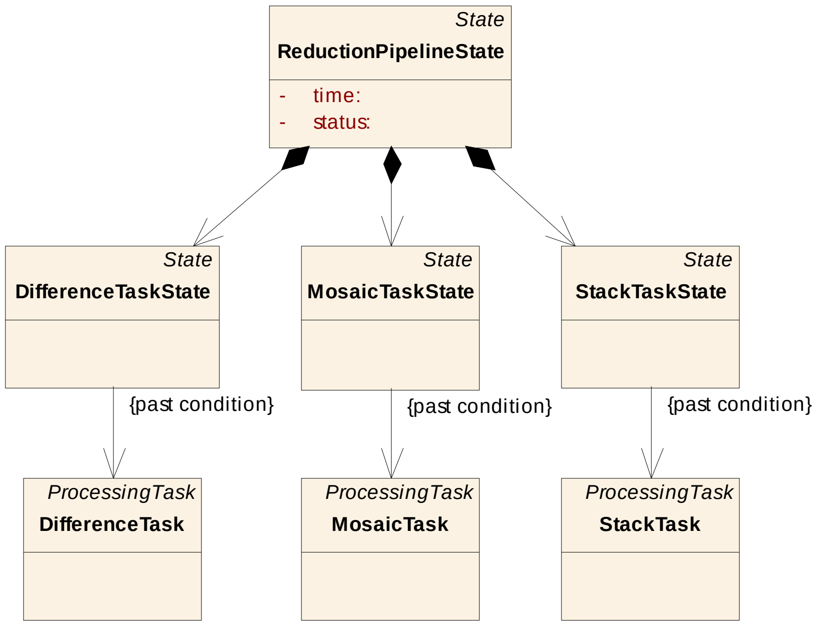


Figure 18 : Reduction Pipeline Configuration Class Diagram

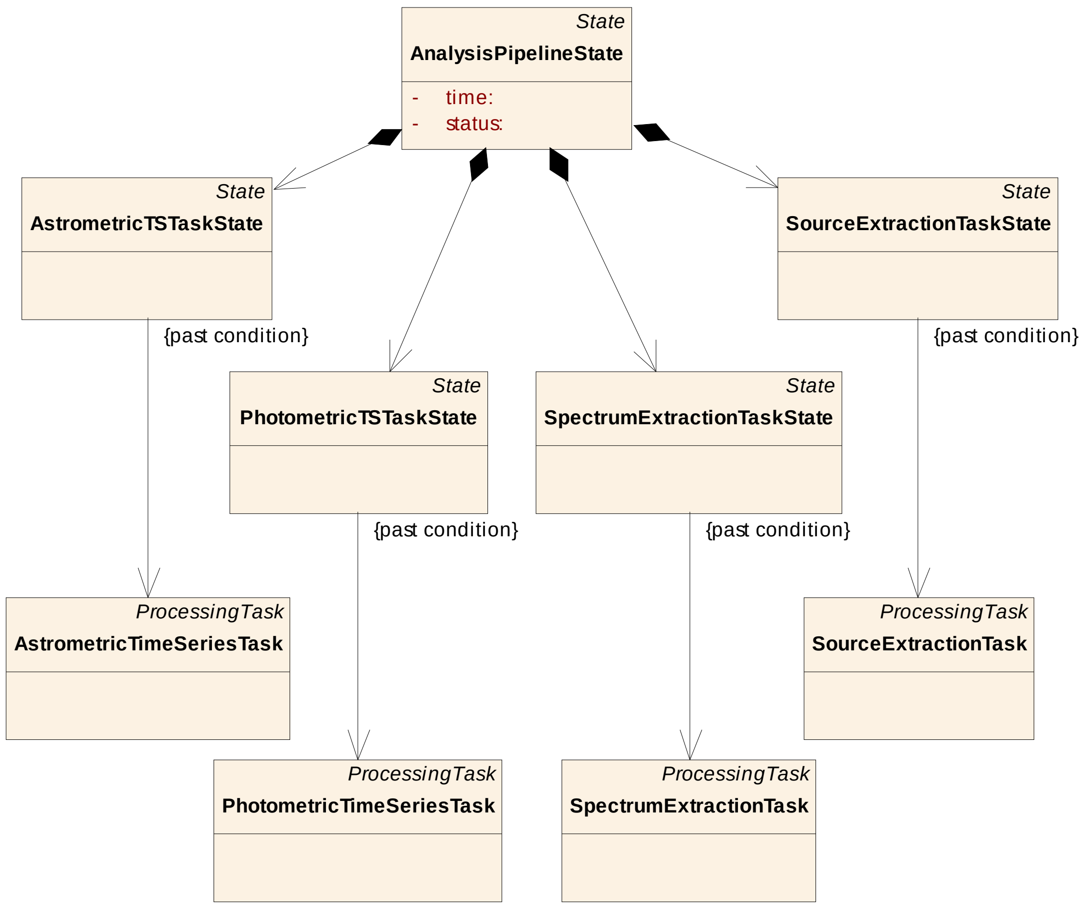


Figure 19 : Analysis Pipeline Configuration Class Diagram

AnalysisPipelineState

AnalysisPipelineState contains state information about an instance of AnalysisPipelineRun.

AstrometricTSTaskState

AstrometricTSTaskState contains the information regarding the state of a given instance of AstrometricTimeSeriesTask.

CalibrateTaskState

CalibrateTaskState contains the information regarding the state of a given instance of CalibrateTask.

CalibrationPipelineState

CalibrationPipelineState contains state information about an instance of CalibrationPipelineRun.

CombineTaskState

CombineTaskState contains the information regarding the state of a given instance of CombineTask.

DifferenceTaskState

DifferenceTaskState contains the information regarding the state of a given instance of DifferenceTask.

MosaicTaskState

MosaicTaskState contains the information regarding the state of a given instance of MosaicTask.

PhotometricTSTaskState

PhotometricTSTaskState contains the information regarding the state of a given instance of PhotometricTSTask.

ReductionPipelineState

ReductionPipelineState contains state information about an instance of ReductionPipelineRun.

SourceExtractionTaskState

SourceExtractionTaskState contains the information regarding the state of a given instance of SourceExtractionTask.

SpectrumExtractionTaskState

SpectrumExtractionTaskState contains the information regarding the state of a given instance of SpectrumExtractionTask.

StackTaskState

StackTaskState contains the information regarding the state of a given instance of StackTask.

Data Products

DataProducts contains the high-level classes associated with any and all Data Products, namely DataProduct and DataProductCollection. The high-level DataProduct class is separated into 6 different classifications, shown here as packages. Each of these classifications is described in the appropriate section below.

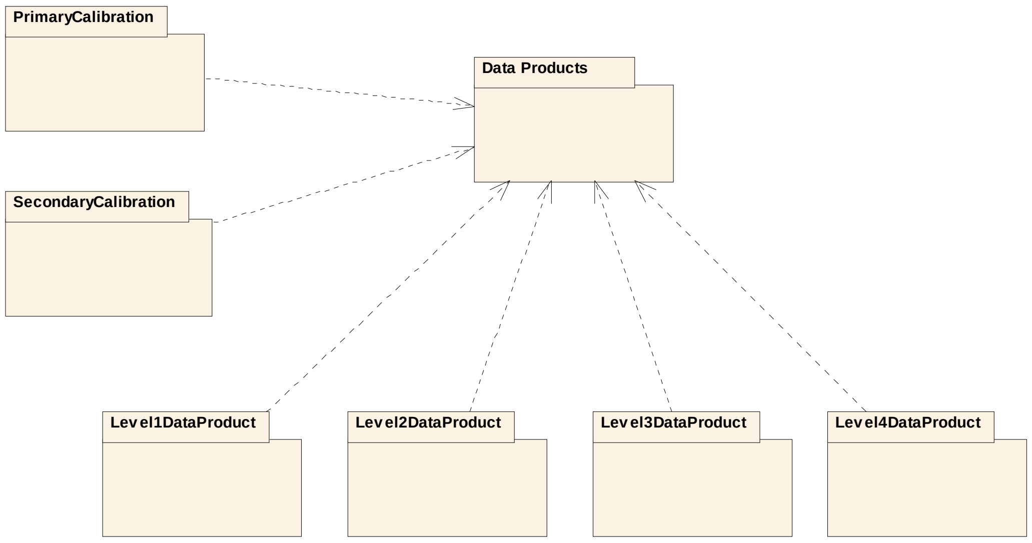


Figure 20 : Data Products Package Diagram

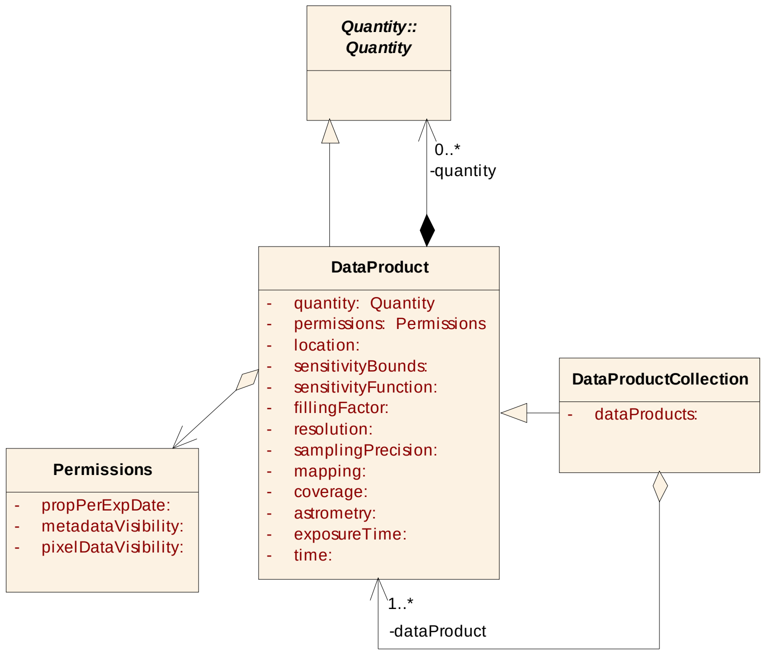


Figure 21 : Data Products Class Diagram

DataProduct

DataProduct is a high-level class that provides basic structure for all DataProducts. Information contained in a DataProduct could be, e.g., references to scientific papers produced using data extracted from the DataProduct.

DataProductCollection

DataProductCollection contains information about, and references to, a collection of DataProducts. This class is used to group DataProducts in various manners.

Permissions

Permissions contains information about the rights by which a User can access a DataProduct. Most, if not all metadata associated with a given Data Product is immediately and publicly available. However, the pixels of the DataProduct are NOT publicly available for an established period of time, which is the Proprietary Period. The established Proprietary Period among AURA institutions is 18 months.

A Permissions object not only restricts access to the pixel data to the primary investigator and collaborators, but also allows the Primary Investigator to provide access to any other User, regardless of whether that User is included in the author list of the Proposal or is an observer. The operation of adding access rights to a given User will be performed through the future workspace class: "Project".

Using a Permissions object, the primary investigator is allowed to reduce the Proprietary Period for any given DataProduct. The class does not, however, provide any method to increase the period, as any extension must be requested through NOAO Directors.

PrimaryCalibration

Primary Calibration contains all Data Products used by a prompt pipeline. A primary calibration product can be generated in advance of receipt or examination of raw science observations. In many cases these will be generated from independent calibration observations. Examples of Primary Calibrations are shown in the diagram below.

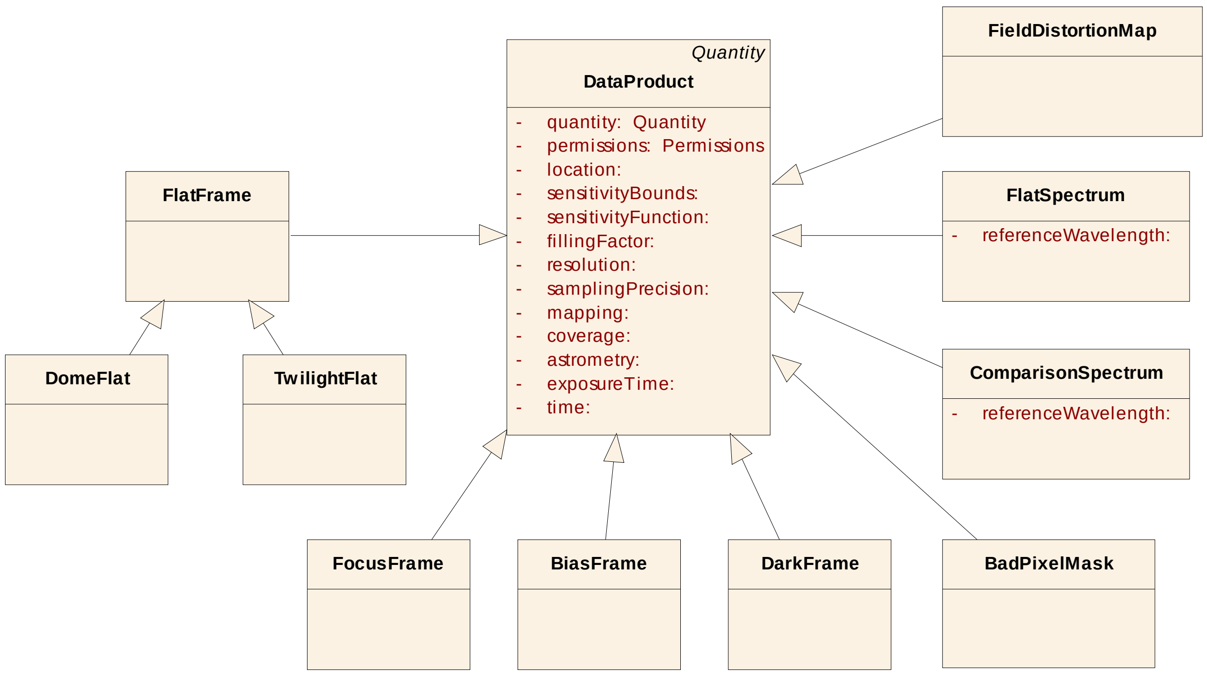


Figure 22 : Primary Calibration Class Diagram

BadPixelMask

BadPixelMask contains information about the bad pixel mask calibration product. These types of pixel masks represent static bad pixels.

BiasFrame

BiasFrame contains information about the bias-level raw calibration product.

ComparisonSpectrum

ComparisonSpectrum contains information about the comparison-level raw calibration spectrum.

DarkFrame

DarkFrame contains information about the dark-level raw calibration image.

DomeFlat

A type of FlatFrame taken using a Dome screen.

FieldDistortionMap

FieldDistortionMap contains information about the field distortion of the Data Products.

FlatFrame

Flat contains information about the flat-field raw calibration product.

FlatSpectrum

FlatSpectrum contains information about the flat-level raw calibration spectrum.

FocusFrame

FocusFrame contains information about the focus-frame raw calibration image.

TwilightFlat

TwilightFlat is a FlatFrame taken using sky during twilight.

SecondaryCalibration

Secondary Calibration Data Products are those used by a delayed pipeline. These products can only be generated after the initial reduction, examination, and evaluation of a set of raw observations. Examples of Secondary Calibrations are contained in the diagram below.

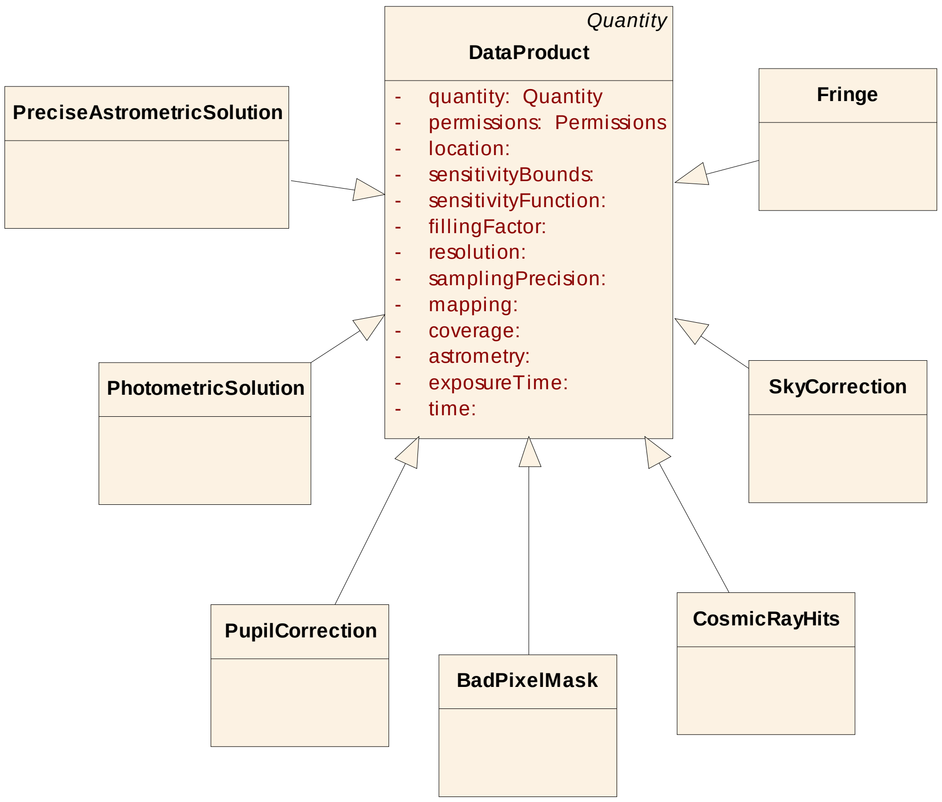


Figure 23 : Secondary Calibration Class Diagram

BadPixelMask

SecondaryBadPixelMask contains information about the bad pixel mask calibration product. This SecondaryCalibration product represents variable bad pixels, or those pixels that were not represented by the PrimaryCalibration products. Examples include cosmic ray hits, very hot pixels, etc.

CosmicRayHits

CosmicRayHits contain information about the cosmic ray hits that occurred during the DataProduct acquisition.

Fringe

Fringe contains information about the fringe correction for a given set of DataProducts.

PhotometricSolution

PhotometricSolution contains information about the photometric solution for a given set of DataProducts.

PreciseAstrometricSolution

PreciseAstrometricSolution contains information about the astrometric solution for a given set of DataProducts.

PupilCorrection

PupilCorrection contains information about the pupil correction for a given set of DataProducts.

SkyCorrection

SkyCorrection contains information about the sky correction for a given set of DataProducts. This calibration product represents a sky flat which has been derived from the data of the corresponding acquisition.

SkyFlat

A type of Flat taken using the sky.

Level1DataProduct

Level1DataProduct contains information about all raw images, spectra, etc., i.e., direct raw or unprocessed data as delivered directly by the instrument. Examples of Level 1 Data Products are contained in the diagram below.

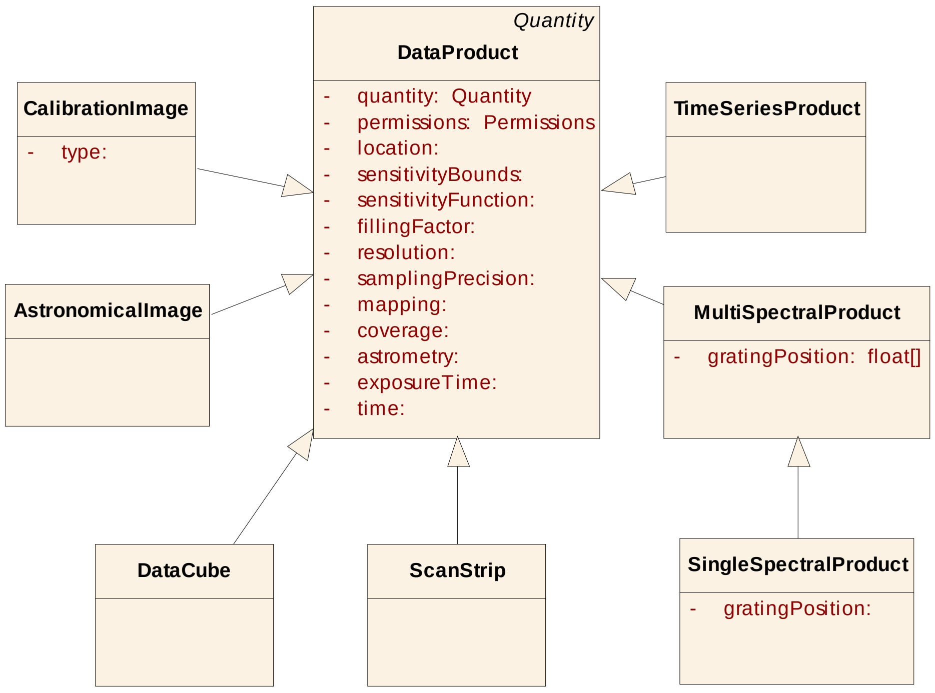


Figure 24 : Level 1 Data Product Class Diagram

AstronomicalImage

Level1AstronomicalImage contains information about a raw snapshot of the sky.

CalibrationImage

CalibrationImage contains information about a raw calibration image.

DataCube

DataCube is the resultant DataProduct from an IFU instrument.

MultiSpectralProduct

MultiSpectralProduct contains information about a DataProduct containing multiple spectra.

ScanStrip

ScanStrip, as a Level 1 Data Product, represents a raw scan strip.

SingleSpectralProduct

SingleSpectralProduct is a MultiSpectralProduct containing a single spectrum.

TimeSeriesProduct

TimeSeriesProduct represents a raw DataProduct that was acquired in time series acquisition.

Level2DataProduct

Level2DataProducts are reduced single observations as might be delivered by a prompt pipeline reduction. Calibration operations have been applied to the data. The level and complexity of the reduction may vary, but the data are always traceable to a single observation. Examples of Level 2 Data Products are contained in the diagram below.

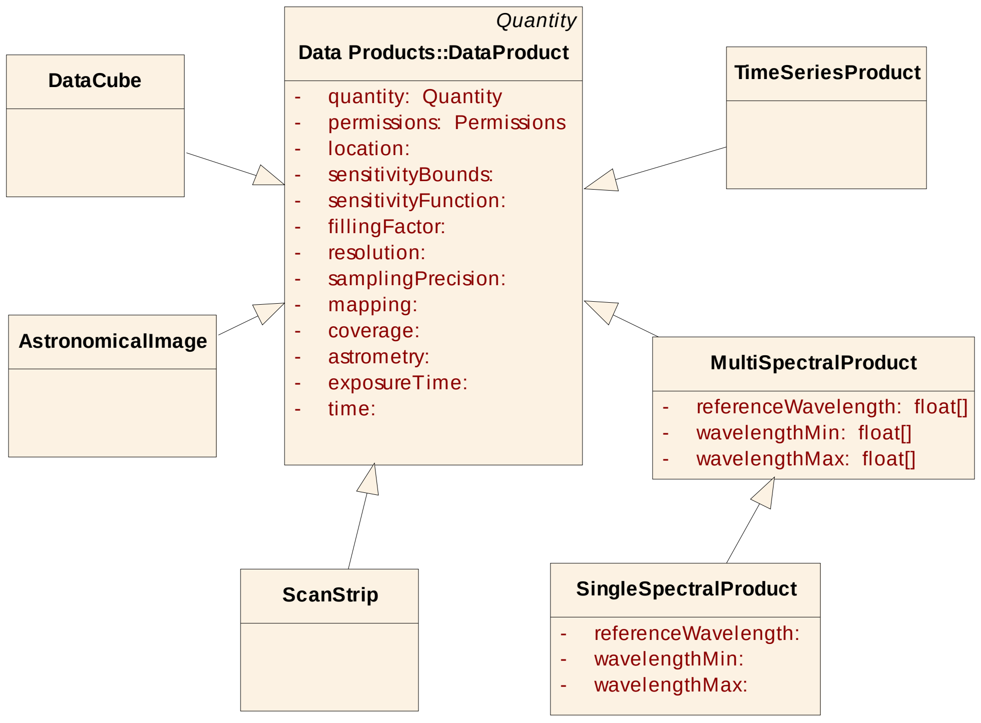


Figure 25 : Level 2 Data Product Class Diagram

AstronomicalImage

AstronomicalImage as a Level 2 DataProduct represents a calibrated snapshot of the sky.

DataCube

DataCube is a product that is derived from an IFU.

MultiSpectralProduct

MultiSpectralProduct as a Level 2 Data Product represents a calibrated DataProduct containing multiple spectra.

ScanStrip

ScanStrip, as a Level 2 Data Product represents a calibrated scan strip.

SingleSpectralProduct

SingleSpectralProduct as a Level 2 Data Product represents a calibrated DataProduct containing a single spectrum.

TimeSeriesProduct

TimeSeriesProduct as a Level 2 Data Product represents a calibrated DataProduct used in time series analysis.

Level3DataProduct

Level3DataProduct contains datasets produced by the combination of multiple observations. In the context of images, these may be stacked of difference images, for example. The format of the data remains similar to that of its components. Examples of Level 3 Data Products are contained in the diagram below.

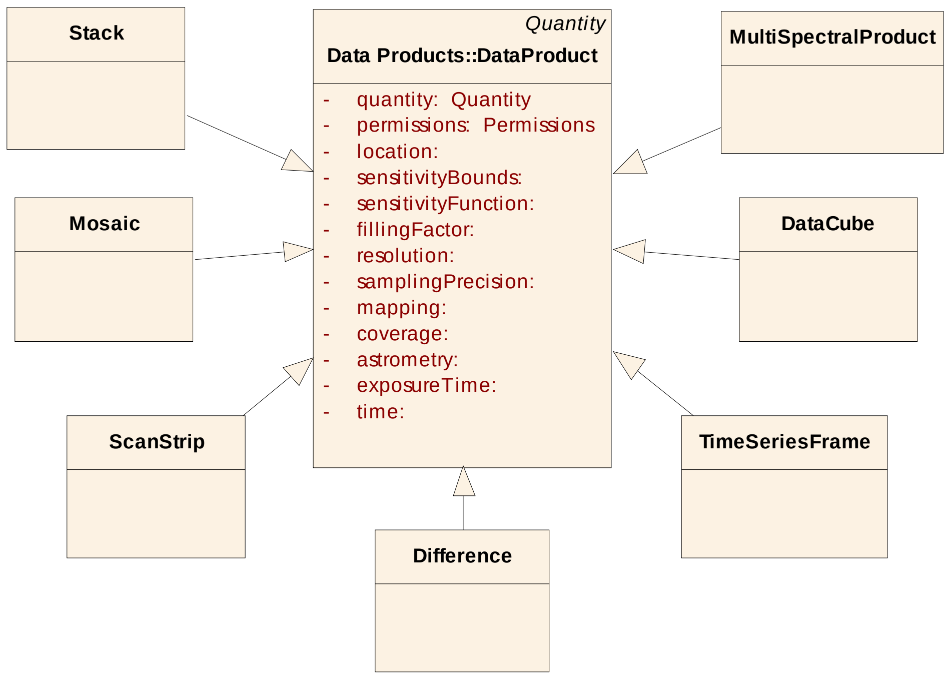


Figure 26 : Level 3 Data Product Class Diagram

DataCube

DataCube as a Level 3 Data Product is a more finely processed product that is derived from an IFU.

Difference

Difference represents the resultant difference between two DataProducts.

Mosaic

Mosaic represents several images mosaicked
into one image.

MultiSpectralProduct

MultiSpectralProduct as a Level 2 Data Product represents a calibrated DataProduct containing multiple spectra.

ScanStrip

ScanStrip, as a Level 3 Data Product represents a reduced scan strip product.

Stack

Stack represents an image stack of one or more DataProducts.

TimeSeriesFrame

TimeSeriesFrame, as a Level 3 Data Product represents a more finely reduced DataProduct used in time series analysis.

Level4DataProduct

Level4DataProduct contains reduced or extracted data. This includes catalogs, one dimensional spectra, photometric time series, and so on. The form of level 4 data is different and in general more compact than their source. Examples of Level 4 Data Products are contained in the diagram below.

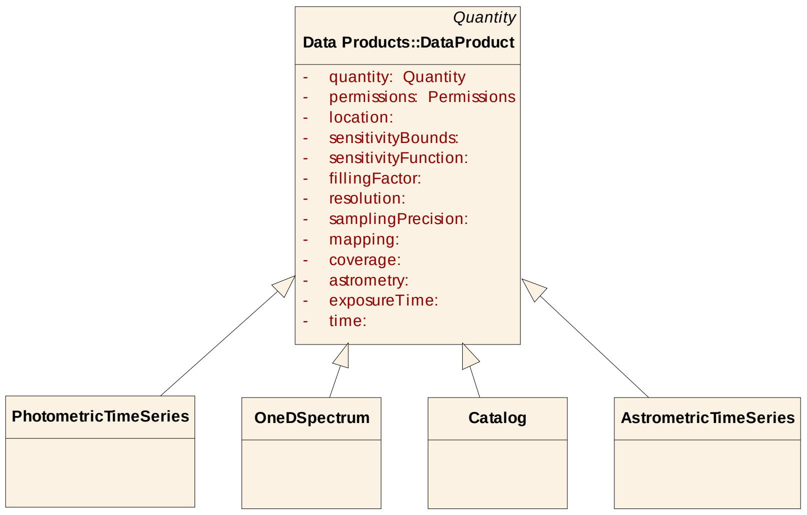


Figure 27 : Level 4 Data Product Class Diagram

AstrometricTimeSeries

AstrometricTimeSeries contains the attributes and other information regarding an astrometric time series, derived from a given dataset. The series will probably exist in tabular form.

Catalog

Catalog contains attributes and other information about a catalog derived from a given dataset.

OneDSpectrum

OneDSpectrum contains attributes of a one-dimensional spectrum derived from a given dataset. This spectrum could possibly exist in tabular form.

PhotometricTimeSeries

PhotometricTimeSeries contains the attributes of a photometric time series derived from a given dataset. This most likely exists in tabular form.