ETC Calculation procedure

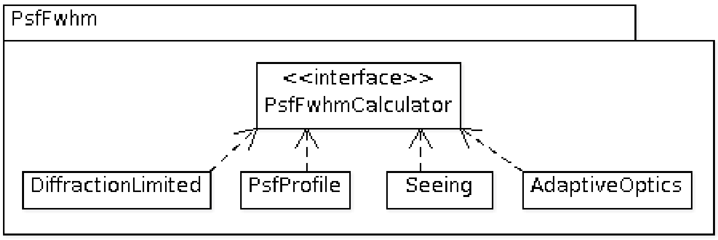
June 15, 2011

Contents

1 Introduction

This document describes in detail the method used by the ETC for calculating the exposure time and the signal to noise ratio. Even though the document describes the software parts responsible for performing the calculation, this document can also be used as a reference by anyone interested in the formula the ETC uses for its calculations in a theoretical level. Each of the following sections describes one of the smaller calculations the SNR calculation has been broken down, together with the different methods to perform the calculation according the given configuration.

2 Psf Fwhm



The **PsfFwhmCalculator** is responsible for calculating the full width half maximum (FWHM) of the point spread function (PSF). The FWHM of the PSF is depended on the wavelength λ, so this calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value, representing the FWHM of the PSF (expressed in *arcsec*).

2.1 DiffractionLimited

This method is used when the user gives the following setup:

- The **PSF type** is set to **Automatic**
- The observation is diffraction limited. This is valid for the following two cases:
  The **Site location** is set to **Space**
  The **Site location** is set to **Ground** AND the option **Diffraction Limited** is selected

This method calculates the theoretical PSF FWHM for the telescope (without AO correction), based on the diameter of the primary mirror and the input wavelength, by using the equation:

$$FWHM_{(\lambda)} = 1.22 * \frac{\lambda * 10^{-8}}{D_1} * 206265 \tag{1}$$

where:
λ: is the wavelength expressed in angstrom (Å)
D_1: is the diameter of the primary mirror expressed in centimeters (cm) (given by the user)
10^{-8}: converts Å to cm
206265: converts $radians$ to $arcsec$

2.2 PsfProfile

This method is used when the user gives as an input a PSF FWHM profile (**PSF Type** is set to **Profile**). For this method, the FWHM of the PSF is calculated based on the profile given by the user, by using interpolation. This has as result the following restrictions:

- The values (x, y) of the profile must be expressed in (Å, *arcsec*)
- The method cannot calculate the FWHM for wavelengths outside of the range defined from the profile

2.3 Seeing

This method is used when the user gives the following setup:

- The **PSF type** is set to **Automatic**
- The observation is seeing limited. This is valid when the **Site location** is set to **Ground** AND the option **Seeing Limited** is selected

For this method, the FWHM of the PSF is calculated by using the following equation:

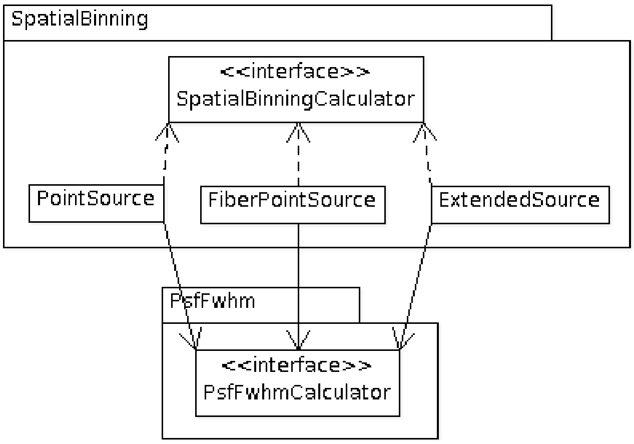
$$FWHM_{(\lambda)} = seeing_{(\lambda)} = seeing_{(Zenith\ in\ V-band)} * AirMass^{0.6} * \left(\frac{\lambda}{5000}\right)^{-0.2} \quad (2)$$

where:
λ: is the wavelength expressed in angstrom (Å)
$seeing_{(ZenithinV-band)}$: is the seeing at zenith, for V-band, expressed in *arcsec* (given by the user)
$AirMass$: is the optical path length through earths atmosphere, relative to that at the zenith (AirMass at zenith is 1). AirMass has no units. (given by the user)

2.4 AdaptiveOptics

This method is used when the **PSF Type** is set to **Adaptive Optics**.
TBD: The method of calculation

3 Spatial Binning



The **SpatialBinningCalculator** is responsible for calculating the number of pixels of the CCD affected by the target, in the spatial direction. Because the spatial binning depends on the FWHM of the PSF, it also depends on the wavelength λ. This means that this calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value, representing the number of pixels in the spatial direction. (*Should it be integer or real???*)

3.1 PointSource

This method is used when the user gives the following setup:

- The **Source** is set to **Point Source**
- The instrument is **not** a **fiber spectrograph**

For this method, the spatial binning is calculated based on the FWHM of the PSF, by using the equation:

$$SpatialBinning_{(\lambda)} = 2 * \frac{FWHM_{(\lambda)}}{p} \tag{3}$$

where:
λ: is the wavelength expressed in angstrom (Å)
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)

3.2 FiberPointSource

This method is used when the user gives the following setup:

- The **Source** is set to **Point Source**
- The instrument is a **fiber spectrograph**

In this case the spatial binning is quantized by the diameter of the fiber, so it is calculated by the following equation:

$$SpatialBinning_{(\lambda)} = 2 * \frac{\left\lceil \frac{FWHM_{(\lambda)}}{\phi_{fiber}} \right\rceil * \phi_{fiber}}{p} \tag{4}$$

where:
λ: is the wavelength expressed in angstrom (Å)
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)
ϕ_{fiber}: is the diameter of the fiber, expressed in *arcsec* (given by the user)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)
$\lceil\ \rceil$: is the ceiling function (rounding up)

3.3 ExtendedSource

For an extended source, because the radius of the source is much bigger of the fiber diameter, the quantization at the case of a fiber spectrograph is negligible, so this method is used at any case of extended source. This method is used when the user gives the following setup:

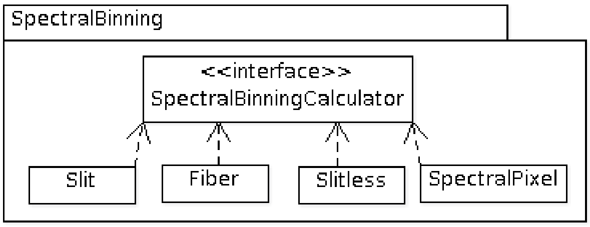
- The **Source** is set to **Extended Source**

For this method the spectral binning is calculated based on the effective radius of the target **and** the FWHM of the PSF. It can be calculated by the following equation:

$$SpatialBinning_{(\lambda)} = 2 * \frac{\sqrt{r_{obj}^2 + FWHM_{(\lambda)}^2}}{p} \tag{5}$$

where:
λ: is the wavelength expressed in angstrom (Å)
r_{obj}: is the effective radius of the target expressed in *arcsec*
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)

4 Spectral Binning



The **SpectralBinningCalculator** is responsible for calculating the number of pixels of the detector affected by light coming from target at a specific wavelength, in the spectral direction. The spectral binning has meaning only for spectroscopy and it should not be used for any other case.

4.1 Slit

This method is used when the user gives the following setup:

- The user has selected to do a calculation for a **Spectral Resolution Element**
- The instrument is a **long slit spectrograph** or an **IFS-slicer spectrograph**

For this method, the spectral binning is calculated based on the width of the slit, by using the equation:

$$SpectralBinning = \frac{w}{p} \tag{6}$$

where:
w: is the width of one slit expressed in *arcsec* (given by the user)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)

4.2 Fiber

This method is used when the user gives the following setup:

- The user has selected to do a calculation for a **Spectral Resolution Element**
- The instrument is a **fiber spectrograph**

For this method, the spectral binning is calculated based on the diameter of the fiber, by using the equation:

$$SpectralBinning = \frac{\phi_{fiber}}{p} \tag{7}$$

where:
ϕ_{fiber}: is the diameter of the slit expressed in *arcsec* (given by the user)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)

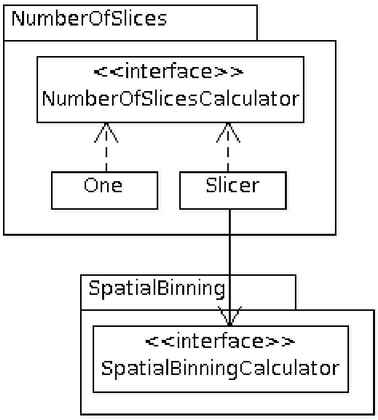
4.3 Slitless

To be defined

4.4 SpectralPixel

This method is used when the user selects to do a calculation for a **Spectral Pixel**. This means that the SNR calculation will be done for only a pixel column of the detector, so this method returns always 1.

5 Number Of Slices



The **NumberOfSlicesCalculator** is responsible for calculating the number of slices which cover the target, when an IFS-slicer is in use.

5.1 One

This method is used in any case that we don't have an IFS-slicer, and it returns always one.

5.2 SlicerPointSource

This method is used when the user gives the following setup:

- The instrument is an **IFS-slicer spectrograph**
- The **Source** is set to **Point Source**

For this method, the number of slices is calculated based on the FWHM of the PSF, by using the equation:

$$N_{Slices(\lambda)} = 2 * \frac{FWHM_{(\lambda)}}{p} \tag{8}$$

where:
λ: is the wavelength expressed in angstrom (Å)
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)

5.3 SlicerExtendedSource

This method is used when the user gives the following setup:

- The instrument is an **IFS-slicer spectrograph**
- The **Source** is set to **Extended Source**

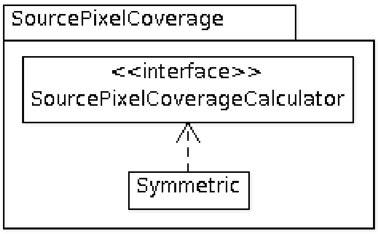
For this method, the number of slices is calculated based on the effective radius of the target **and** the FWHM of the PSF, by using the equation:

$$N_{Slices(\lambda)} = 2 * \frac{\sqrt{r_{obj}^2 + FWHM_{(\lambda)}^2}}{p} \tag{9}$$

where:
λ: is the wavelength expressed in angstrom (Å)

r_{obj}: is the effective radius of the target expressed in *arcsec*
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)

6 Source Pixel Coverage



The **SourcePixelCoverageCalculator** is producing a map of the detector pixels which are affected by the source light, without taking into consideration any slit or fiber losses, or any grism or prism diffraction. The output of the calculator is a two dimensional image for each wavelength λ, which represents with 1 the pixels which are affected by the source light and with 0 the ones that are not.

6.1 Symmetric

This method is used for all the cases the user has **not** given a specific image describing the surface brightness of the source, and the FWHM of the PSF is calculated by the ETC. Firstly, the radius of the source (in *arcsec*) is calculated, by using the equation:

$$R_{(\lambda)} = \sqrt{r_{obj}^2 + FWHM_{(\lambda)}^2} \tag{10}$$

where:
λ: is the wavelength expressed in angstrom (Å)
r_{obj}: is the effective radius of the target expressed in *arcsec* for extended source and 0 for point source
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)

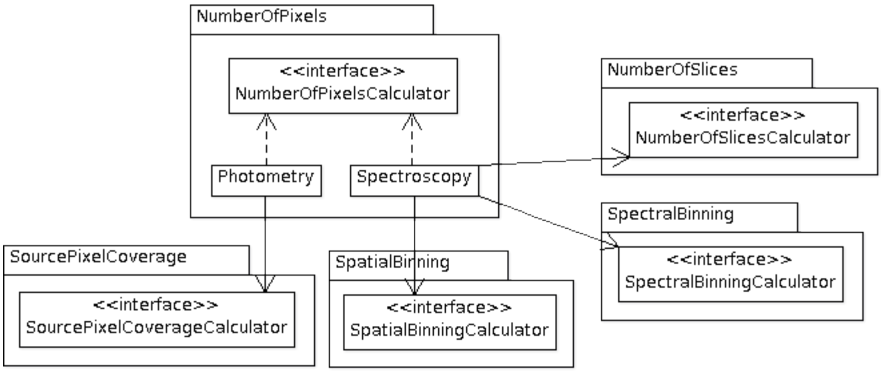
By using this radius the side length of the result image (which is a square) is calculated (in *pixels*), by using the ceiling function:

$$Side_{(\lambda)} = \left\lceil \frac{2 * R_{(\lambda)}}{p} \right\rceil \tag{11}$$

where:
R: is the radius calculated by equation 10 (in *arcsec*)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)
$\lceil\ \rceil$: is the ceiling function (rounding up)

Finally, the result image is constructed with dimensions $Side * Side$. The value of each pixel is set by checking the distance between the center of the image and the pixel corner closer to it. If this distance is bigger than the calculated radius R (equation 10) the pixel gets the value 0. If it is smaller (so there is some light from the source falling on the pixel), it gets the value 1.

7 Number Of Pixels



The **NumberOfPixelsCalculator** is responsible for calculating the total number of pixels of the detector, which are affected by the target. It must be noted that the meaning of pixel here is used as the area of a pixel. At the previous sections **SpectralBinningCalculator** and **SpatialBinningCalculator** the term pixel is used as the length of a pixels side. This calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value, representing the total number of pixels affected by the target.

7.1 Photometry

This method is used in the case of photometry. In this case there is no obstacles on the light path (like slit, fiber, etc) and there is no diffraction medium (like grism, prism, etc), so the result of the **SourcePixelCoverageCalculator** (see above) is a representation of the source on the sensor. What this method does is to just count the non zero pixels of the image result of the **SourcePixelCoverageCalculator**.

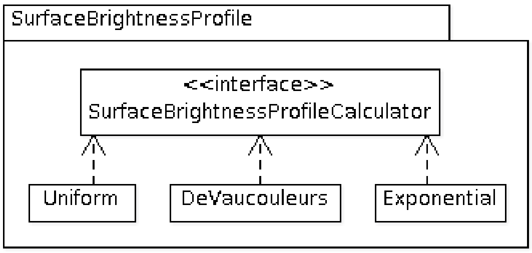
7.2 Spectroscopy

This method is used in the case of spectroscopy. In this case the number of pixels can be calculated with the equation:

$$N_{pix(\lambda)} = N_{slices(\lambda)} * \lceil SpatialBinning_{(\lambda)} \rceil * \lceil SpectralBinning \rceil \tag{12}$$

where:
$N_{slices(\lambda)}$: is the number of slices in the case of an IFS-slicer, or 1 otherwise (see **NumberOfSlicesCalculator** above)
$SpatialBinning_{(\lambda)}$: is the number of pixels affected in the spatial direction (see **SpatialBinningCalculator** above)
$SpectralBinning$: is the number of pixels affected in the spectral direction (see **SpectralBinningCalculator** above)
$\lceil\ \rceil$: is the ceiling function (rounding up)

8 Surface Brightness Profile



The **SurfaceBrightnessProfileCalculator** is responsible for providing the surface brightness profiles $I_{(r)}$. These profiles, in the polar coordinate system, are depended only by the radial coordinate, so a function *value(r)* is provided by the calculator. Because the double integration on cartesian coordinate system is also necessary (for slit spectroscopy), a method *value(x,y)* is also provided.

8.1 Uniform

For this method the uniform spatial profile is used. This profile follows the following distribution:

$$I_{(r)} = \begin{cases} 1 & , r \leq r_{obj} \\ 0 & , r > r_{obj} \end{cases} \tag{13}$$

8.2 De Vaucouleurs

For this method the De Vaucouleurs spatial profile is used. This profile follows the following distribution:

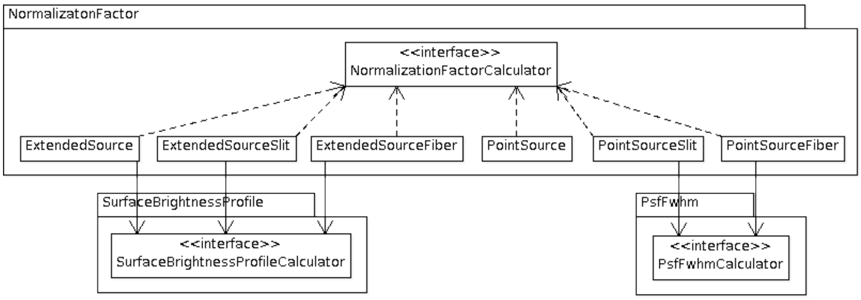
$$I_{(r)} = e^{-7.67 * \left(\frac{r}{r_{obj}}\right)^{1/4}} \tag{14}$$

8.3 Exponential

For this method the exponential spatial profile is used. This profile follows the following distribution:

$$I_{(r)} = e^{-1.68 * \left(\frac{r}{r_{obj}}\right)} \tag{15}$$

9 Normalization Factor



The **NormalizationFactorCalculator** is responsible for calculating the normalization factor $C_{(\lambda)}$. The normalization factor has different meaning for point sources and for extended sources.

For point sources, the magnitude AB given by the user corresponds to the total flux of the target. In this case the normalization factor is a number in the range [0,1], which shows what percentage of the targets flux is passing through an existing slit or fiber (in any other case it is one).

For extended sources, the magnitude AB given by the user corresponds to the maximum flux of the profile. In this case the normalization factor is a number with which the flux needs to be multiplied to calculate the total flux, taking in consideration any loses because of a slit or fiber. In this case the flux can be bigger than one. Note that in this case $C_{(\lambda)} \neq 1$ for the cases we don't have a slit or fiber.

9.1 PointSource

This method is used when the user gives the following setup:

- The instrument is NOT a long slit or fiber spectrograph
- The **Source** is set to **Point Source**

For this method the normalization factor is always one.

9.2 PointSourceSlit

This method is used when the user gives the following setup:

- The instrument is a **slit spectrograph**
- The **Source** is set to **Point Source**

In this case the normalization factor takes in consideration the losses because of the slit. It is the double integral of a two dimensional Gaussian distribution with standard deviation for both axis:

$$\sigma_{PSF} = \frac{FWHM_{(\lambda)}}{2.37} \tag{16}$$

in the ranges $[-FWHM, FWHM]$, $[-\frac{w}{2}, \frac{w}{2}]$
where:
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)
w: is the slit width expressed in *arcsec*
(*Final equation to be defined*)

9.3 PointSourceFiber

This method is used when the user gives the following setup:

- The instrument is a **fiber spectrograph**
- The **Source** is set to **Point Source**

In this case the normalization factor takes in consideration the losses because of the fiber. It is the integral of a Gaussian distribution with standard deviation:

$$\sigma_{PSF} = \frac{FWHM_{(\lambda)}}{2.37} \tag{17}$$

in the range $[-\frac{\phi_{fiber}}{2}, \frac{\phi_{fiber}}{2}]$, so the normalization factor can be calculated as

$$C_{(\lambda)} = ERF\left[\frac{\phi_{fiber}/2}{\sqrt{2} * \sigma_{PSF}}\right] \tag{18}$$

where:
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)
ϕ_{fiber}: is the diameter of the fiber expressed in *arcsec*
ERF: is the error function

9.4 ExtendedSource

This method is used when the user gives the following setup:

- The instrument is NOT a long slit or fiber spectrograph
- The **Source** is set to **Extended Source**

For this method the normalization factor is calculated by the double integral of the surface brightness profile $I_{(r)}$ over a disk with radius r_{obj}. Because the profile is affected only by the radius, it is easier to calculate it in the polar coordinates:

$$C_{(\lambda)} = \int_{0}^{2\pi}\int_{0}^{r_{obj}} I_{(r)} r dr d\theta = 2\pi * \int_{0}^{r_{obj}} I_{(r)} r dr \tag{19}$$

where:
r_{obj}: is the effective radius of the source expressed in *arcsec* (given by the user)
$I_{(r)}$: is the surface brightness profile (see **SurfaceBrightnessProfileCalculator** above)

9.5 ExtendedSourceSlit

This method is used when the user gives the following setup:

- The instrument is a **slit spectrograph**
- The **Source** is set to **Extended Source**

For this method the normalization factor is calculated by the double integral of the surface brightness profile $I_{(r)}$ over a rectangular defined by the r_{obj} and the slit width.

$$C_{(\lambda)} = \int\limits_{-r_{obj}}^{r_{obj}} \int\limits_{-w/2}^{w/2} I_{(x,y)} dxdy \tag{20}$$

where:
r_{obj}: is the effective radius of the source expressed in *arcsec* (given by the user)
w: is the slit width expressed in *arcsec* (given by the user)
$I_{(x,y)}$: is the surface brightness profile (see **SurfaceBrightnessProfileCalculator** above)

9.6 ExtendedSourceFiber

This method is used when the user gives the following setup:

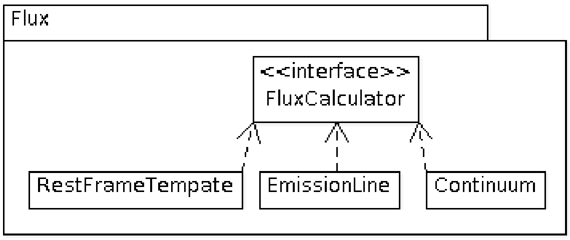
- The instrument is a **fiber spectrograph**
- The **Source** is set to **Extended Source**

For this method the normalization factor is calculated by the double integral of the surface brightness profile $I_{(r)}$ over a disk with radius ϕ_{fiber}. Because the profile is affected only by the radius, it is easier to calculate it in the polar coordinates:

$$C_{(\lambda)} = \int\limits_{0}^{2\pi} \int\limits_{0}^{\phi_{fiber}/2} I_{(r)} r dr d\theta = 2\pi * \int\limits_{0}^{\phi_{fiber}/2} I_{(r)} r dr \tag{21}$$

where:
ϕ_{fiber}: is the fiber diameter expressed in *arcsec* (given by the user)
$I_{(r)}$: is the surface brightness profile (see **SurfaceBrightnessProfileCalculator** above)

10 Flux



The **FluxCalculator** is responsible for calculating the flux of the target, $F_{(\lambda)}$. The flux is depended on the wavelength λ, so this calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value, representing the flux (expressed in $erg/s/cm^2$/Å).

Here it must be noted that the meaning of the flux differs for point and extended sources. For point sources the method returns the total flux of the target, when for extended sources it returns the maximum flux value of the surface brightness profile. The magnitude AB (m_{AB}) given by the user changes meaning accordingly.

10.1 RestFrameTemplate

This method is used when the user gives a **template** for the **spectral distribution** of the source flux. In this case the spectral distribution is redshifted according the redshift given by the user and normalized by the flux calculated by the magnitude AB. This is done with the following steps:

- Convert the magnitude AB to flux by using the equation:

$$F_x = 10^{-\frac{m_{AB}+48.6}{2.5}} * \frac{c}{\lambda_x^2} \tag{22}$$

  where:
  m_{AB}: is the AB magnitude (given by the user)
  c: is the speed of light expressed in Å$/s$
  λ_x: is the central wavelength of the filter selected by the user, expressed in Å

- Redshift the template given by the user. This has the restriction that the x values of the template must be expressed in Å. The y values can be expressed in anything, as they are going to be normalized by the calculated flux. The equation to calculate the wavelengths of the redshifted template is:

$$\lambda_{(z)} = \lambda_{(0)} * (1+z) \tag{23}$$

  where:
  $\lambda_{(z)}$: is the redshifted wavelength (expressed in Å)
  $\lambda_{(0)}$: is the restframe wavelength (expressed in Å)
  z: is the redshift (given by the user)

- Normalize the redshifted template by the flux calculated earlier. This is done in two steps:
  - Calculate the value of the redshifted template for the central wavelength of the filter selected by the user (F_{0x}). To calculate this value spline interpolation is used.
  - Calculate the normalized values. For this the following equation can be used:

$$F_z = F_0 * \frac{F_x}{F_{0x}} \tag{24}$$

    where:
    F_z: is the normalized value
    F_0: is the value of the redshifted template before the normalization
    F_x: is the Flux at the central wavelength of the filter as calculated earlier
    F_{0x}: is the value calculated on the previous step

- Calculate the Flux value for the given wavelength λ. This is done by using spline interpolation on the redshifted, normalized template.

10.2 EmissionLine

This method is used when the user selects the **Emission Line** option for the **spectral distribution** of the sources flux. In this case the flux is calculated as:

$$F_{(\lambda)} = \begin{cases} 10^{-\frac{m_{AB}+48.6}{2.5}} * \frac{c}{\lambda_{em}^2} & , \lambda = \lambda_{em} \\ 0 & , \text{otherwise} \end{cases} \tag{25}$$

where:
m_{AB}: is the AB magnitude (given by the user)
λ: is the wavelength the flux is calculated for, expressed in Å
λ_{em}: is the wavelength of the emission line, calculated as $\lambda_{em} = \lambda_{ref} * (1+z)$
λ_{ref}: is the wavelength of the emission line in restframe, expressed in Å (given by the user)
z: is the redshift (given by the user)
c: is the speed of light expressed in Å$/s$

NOTE: Special care must be taken when the calculation is done for an array of wavelengths which does not contain the $\lambda_{(em)}$ exactly.

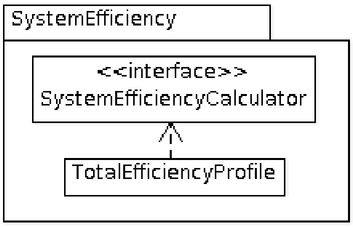
10.3 Continuum

This method is used when the user selects the **Continuum** option for the **spectral distribution** of the sources flux. In this case the flux is constant for any wavelength and is calculated as:

$$F_{(\lambda)} = 10^{-\frac{m_{AB}+48.6}{2.5}} * \frac{c}{\lambda_x^2} \tag{26}$$

where:
m_{AB}: is the AB magnitude (given by the user)
c: is the speed of light expressed in $\text{Å}/s$
λ_x: is the central wavelength of the filter selected by the user, expressed in Å

11 System Efficiency



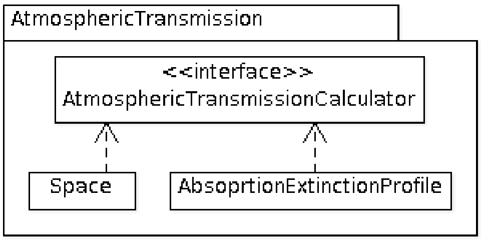
The **SystemEfficiencyCalculator** is responsible for calculating the total efficiency of the system, $\varepsilon_{(\lambda)}$. The efficiency is a number in the range [0, 1], which shows how efficiently the instrument (excluded the filter) in use transfers the light. The efficiency is depended on the wavelength λ, so this calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value between 0 and 1, representing the efficiency.

11.1 TotalEfficiencyProfile

This method is uses the profile of the total efficiency of the system, given by the user. In this case the user is responsible to take in consideration all the different parts of the system. This method calculates the total efficiency for the requested wavelength from the given profile, by using interpolation. This has as result the following restrictions:

- The x values must be expressed in Å
- The y values of the profile must be in the range [0, 1]
- The method cannot calculate the efficiency for wavelengths outside the range defined from the profile

12 Atmospheric Transmission


AtmosphericTransmission
<<interface>>
AtmosphericTransmissionCalculator
Space
AbsoprtionExtinctionProfile

The **AtmosphericTransmissionCalculator** is responsible for calculating the total atmospheric transmission, $\xi_{(\lambda)}$. The atmospheric transmission is a number in the range [0, 1], which shows how much of the light is transmitted through the atmosphere. It depends on the wavelength λ, so this calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value between 0 and 1.

12.1 Space

This method is used when the user selects the **Space** option for the **Site location**. In this case there is no atmosphere, so the atmospheric transmission is always one.

12.2 AbsorptionExtinctionProfile

This method is used when the user selects the **Ground** option for the **Site location**. This method uses the profiles for the atmospheric absorption and for the atmospheric extinction, given by the user. The atmospheric transmission is calculated with the equation:

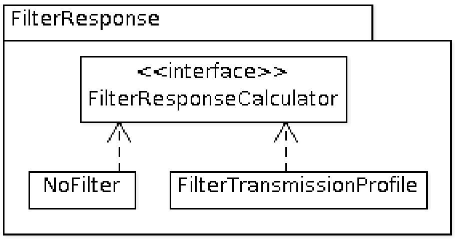
$$\xi_{(\lambda)} = \xi_{abs(\lambda)} * 10^{-0.4 * X * k_{(\lambda)}} \tag{27}$$

where:
λ: is the wavelength the atmospheric transmission is calculated for, expressed in Å
$\xi_{(\lambda)}$: is the atmospheric transmission
$\xi_{abs(\lambda)}$: is the atmospheric absorption calculated with interpolation (given by the user)
X: is the air mass (given by the user)
$k_{(\lambda)}$: is the atmospheric extinction calculated with interpolation (given by the user)

The usage of the two profiles has a result the following restrictions:

- For both templates the x values must be expressed in Å
- The method cannot make calculations outside the range defined from the profile

13 Filter Response



The **FilterResponseCalculator** is responsible for calculating the efficiency of the filter in use, $Filter_{(\lambda)}$. This efficiency is a number in the range [0, 1], which shows how efficiently the filter in use transfers the light. The efficiency is depended on the wavelength λ, so this calculator should get as input a value representing the wavelength (expressed in Å) and it should return a single value between 0 and 1, representing the efficiency.

13.1 NoFilter

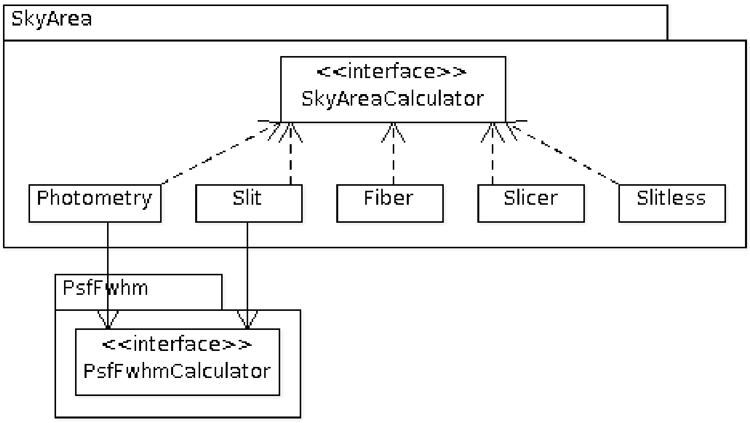
This method is used when there is no filter used, so the filter response is always one.

13.2 FilterTransmissionProfile

This method is used when the user gives a profile for the filter transmission. In this case the values are calculated using interpolation. This has a result the following restrictions:

- The x values of the template must be expressed in Å
- The method cannot calculate the filter transmission for wavelengths outside the range defined from the profile

14 Sky Area



The **SkyAreaCalculator** is responsible for calculating the area (in $arcsec^2$) of the sky that contributes to the sky noise. Because the sky area depends on the FWHM of the PSF it also depends on the wavelength λ, so this calculator should get as input a value representing the wavelength (expressed in Å).

14.1 Photometry

This method is used in the case of photometry. In this case the area of the sky contributing to the background noise is depended only on the PSF and is calculated with the following equation:

$$A_{sky} = \pi * FWHM_{(\lambda)}^2 \tag{28}$$

where:
A_{sky}: is the calculated area of the sky
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)

14.2 Slit

This method is used when the instrument is a slit spectrograph. In this case the area of the sky is calculated with the following equation:

$$A_{sky} = 2 * FWHM_{(\lambda)} * w \tag{29}$$

where:
A_{sky}: is the calculated area of the sky
$FWHM$: is the FWHM of the PSF expressed in *arcsec* (see **PsfFwhmCalculator** above)
w: is the slit width in *arcsec* (given by the user)

14.3 Fiber

This method is used when the instrument is a fiber spectrograph. In this case the area of the sky is depended only on the diameter of the fiber and is calculated with the following equation:

$$A_{sky} = \pi * \left(\frac{\phi_{fiber}}{2}\right)^2 \tag{30}$$

where:
A_{sky}: is the calculated area of the sky
ϕ_{fiber}: is the fiber diameter expressed in *arcsec* (given by the user)

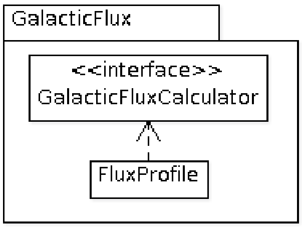
14.4 Slicer

To be defined

14.5 Slitless

To be defined

15 Galactic Flux



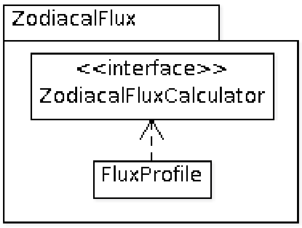
The **GalacticFluxCalculator** is responsible for calculating the galactic flux. The result is expressed in $erg/s/cm^2/\text{Å}/arcsec^2$.

15.1 FluxProfile

This method uses a profile for the galactic flux. In this case the values are calculated using interpolation. This has a result the following restrictions:

- The x values of the template must be expressed in Å
- The y values of the template must be expressed in $erg/s/cm^2/\text{Å}/arcsec^2$
- The method cannot calculate the flux for wavelengths outside the range defined from the profile

16 Zodiacal Flux



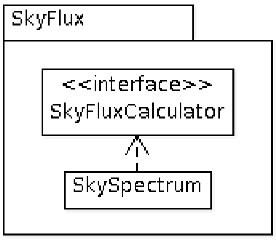
The **ZodiacalFluxCalculator** is responsible for calculating the zodiacalzodiacal flux. The result is expressed in $erg/s/cm^2/\text{Å}/arcsec^2$.

16.1 FluxProfile

This method uses a profile for the zodiacal flux. In this case the values are calculated using interpolation. This has a result the following restrictions:

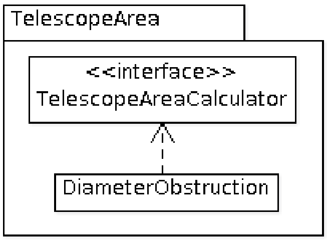
- The x values of the template must be expressed in Å
- The y values of the template must be expressed in $erg/s/cm^2/\text{Å}/arcsec^2$
- The method cannot calculate the flux for wavelengths outside the range defined from the profile

17 Sky Flux



To be defined

18 Telescope Area



The **TelescopeAreaCalculator** is responsible for calculating the total area of the telescope mirror which reflects light, expressed in cm^2.

18.1 DiameterObstruction

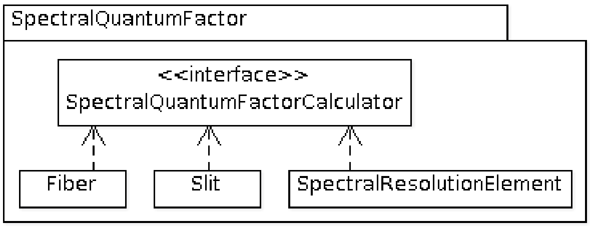
This method is used in the case of a circular mirror. In this case the telescope area is calculated based on the diameter of the mirror, accounting also any obstruction of the light, by using the equation:

$$A_{tel} = \pi * \left(\frac{D_1}{2}\right)^2 * (1 - ob) \tag{31}$$

where
D_1: is the diameter of the primary mirror expressed in centimeters (cm) (given by the user)
ob: is the total light obstruction (a number in the range [0,1])

NOTE: The obstruction is stored internally from the system as a number in the range [0,1], but a user of the GUI should use % values (in the range [0,100]).

19 Spectral Quantum Factor



The calculation of the flux for the source signal and the background noise is done by calculating the photons which pass through any obstacles (like slit, fiber, etc) and reach the detector. For spectroscopy though, the SNR calculation can be done in two modes, for a **Spectral Resolution Element** and for a **Spectral Pixel** (a pixel column in the spatial direction). At the second case not all the flux detected by the detector should be included in the calculation. The **SpectralQuantumFactorCalculator** is responsible for calculating the factor (SQF) with which the flux needs to be multiplied to be corrected, according the above.

19.1 Slit

This method is used when the user gives the following setup:

- The user has selected to do a calculation for a **Spectral Pixel**
- The instrument is a **long slit spectrograph** or an **IFS-slicer spectrograph**

For this method, the spectral quantum factor is calculated based on the width of the slit, by using the equation:

$$SQF = \frac{p}{w} \tag{32}$$

where:
w: is the width of one slit expressed in *arcsec* (given by the user)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)

19.2 Fiber

This method is used when the user gives the following setup:

- The user has selected to do a calculation for a **Spectral Pixel**
- The instrument is a **fiber spectrograph**

For this method, the spectral quantum factor is calculated based on the diameter of the fiber, by using the equation:

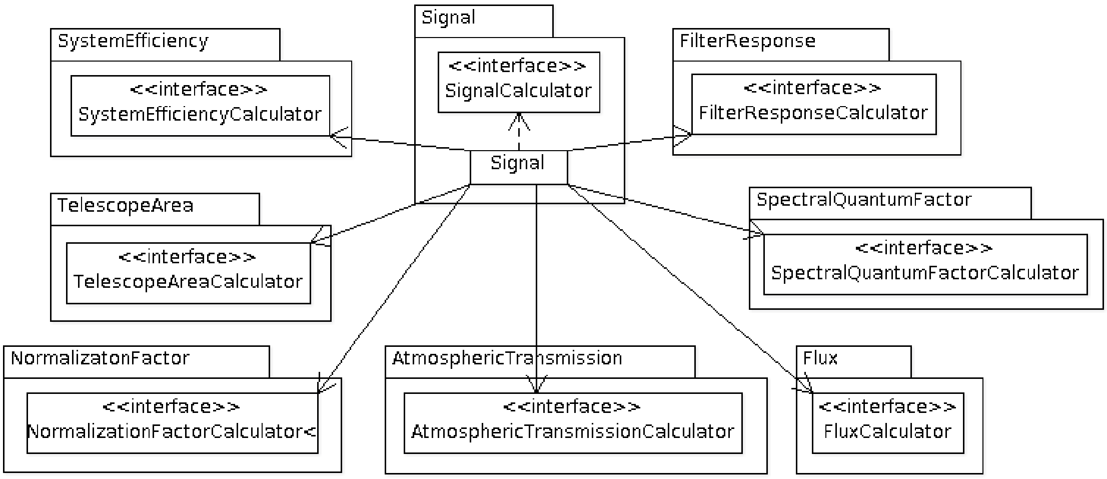
$$SQF = \frac{p}{\phi_{fiber}} \tag{33}$$

where:
ϕ_{fiber}: is the diameter of the slit expressed in *arcsec* (given by the user)
p: is the pixel scale of the instrument in the spatial direction, expressed in *arcsec/pixel* (given by the user)

19.3 SpectralResolutionElement

This method is used when the user selects to do a calculation for a **Spectral Resolution Element** or for the case of **imaging**. In this case all the flux calculated should be included in the SNR calculation, so this method returns always 1.

20 Signal



The **SignalCalculator** is responsible for calculating the total number of photons from the target ($S_{(\lambda)}$) detected by the instrument per second, for a specific wavelength, expressed in $\gamma/s/$Å.

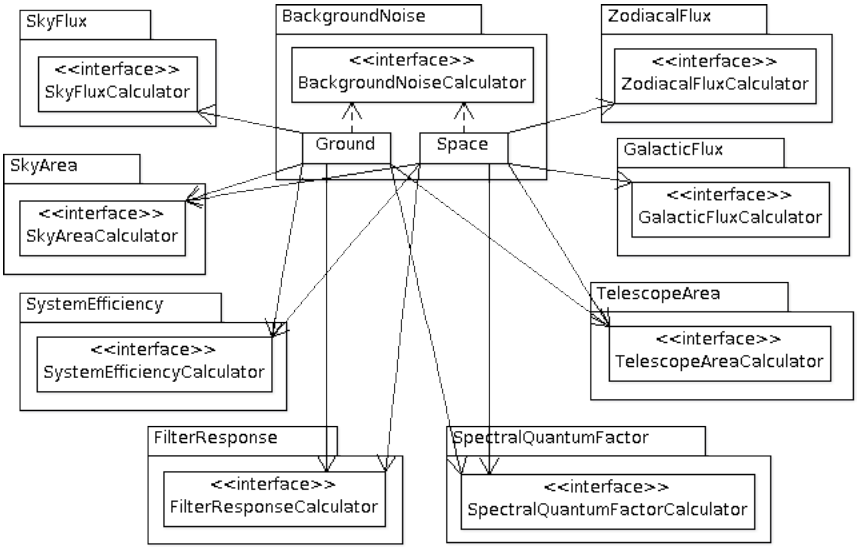
20.1 Signal

This method uses the flux calculated by the **FluxCalculator** and it takes in consideration the size of the primary mirror and any losses before the light reach the CCD. The equation used is:

$$S_{(\lambda)} = F_{(\lambda)} * C_{(\lambda)} * \xi_{(\lambda)} * \varepsilon_{(\lambda)} * Filter_{(\lambda)} * A_{tel} * \frac{\lambda}{h * c} * SQF \tag{34}$$

where
F: is the flux expressed in $erg/s/cm^2/$Å(see **FluxCalculator**)
C: is the normalization factor (see **NormalizationFactorCalculator**)
ξ: is the atmospheric transmission (see **AtmosphericTransmissionCalculator**)
ε: is the total system efficiency, excluding the filter (see **SystemEfficiencycalculator**)
$Filter$: is the filter response (see **FilterResponseCalculator**)
A_{tel}: is the total effective primary mirror area (see **TelescopeAreaCalculator**)
λ: is the wavelength the signal is calculated for, expressed in Å
h: is the Planck constant expressed in $erg * s$
c: is the speed of light expressed in Å$/sec$
SQF: is the spectral quantum factor (see **SpectralQuantumFactorCalculator**)

21 Background Noise



The **BackgroundNoiseCalculator** is responsible for calculating the total number of photons because of background noise ($BN_{(\lambda)}$) which are detected by the instrument per second and for a specific wavelength, expressed in $\gamma/s/\text{Å}$. It must be noted that only the background noise which affects pixels related with the target signal is included.

21.1 Ground

This method is used for the case of a telescope located on earth. In this case the only background noise in consideration is the sky noise. The equation used is:

$$BN_{(\lambda)} = SkyFlux_{(\lambda)} * \varepsilon_{(\lambda)} * Filter_{(\lambda)} * A_{tel} * A_{sky(\lambda)} * \frac{\lambda}{h * c} * SQF \tag{35}$$

where
$SkyFlux$: is the flux from the sky expressed in $erg/s/cm^2/\text{Å}/arcsec^2$ (see **SkyFluxCalculator**)
ε: is the total system efficiency, excluding the filter (see **SystemEfficiencycalculator**)
$Filter$: is the filter response (see **FilterResponseCalculator**)
A_{tel}: is the total effective primary mirror area (see **TelescopeAreaCalculator**)
A_{sky}: is the total area of sky contributing at the noise (see **SkyAreaCalculator**)
λ: is the wavelength the signal is calculated for, expressed in Å
h: is the Planck constant expressed in $erg * s$
c: is the speed of light expressed in $\text{Å}/sec$
SQF: is the spectral quantum factor (see **SpectralQuantumFactorCalculator**)

21.2 Space

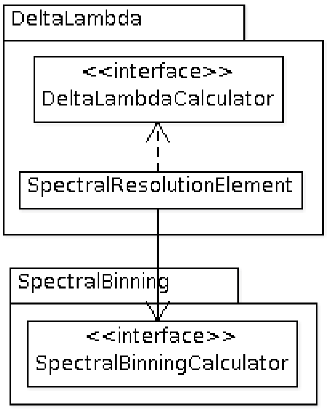
This method is used for the case of a satellite telescope. In this case there is no sky noise (as the telescope is in space), so the only noises contributing in the background noise are the zodiacal and galactic noises. The equation used is:

$$BN_{(\lambda)} = (ZodiacalFlux_{(\lambda)} + GalacticFlux_{(\lambda)}) * \varepsilon_{(\lambda)} * Filter_{(\lambda)} * A_{tel} * A_{sky(\lambda)} * \frac{\lambda}{h * c} * SQF \tag{36}$$

where
$ZodiacalFlux$: is the zodiacal flux expressed in $erg/s/cm^2/\text{Å}/arcsec^2$ (see **ZodiacalFluxCalculator**)
$GalacticFlux$: is the galactic flux expressed in $erg/s/cm^2/\text{Å}/arcsec^2$ (see **GalacticFluxCalculator**)
ε: is the total system efficiency, excluding the filter (see **SystemEfficiencycalculator**)
$Filter$: is the filter response (see **FilterResponseCalculator**)

A_{tel}: is the total effective primary mirror area (see **TelescopeAreaCalculator**)
A_{sky}: is the total area of sky contributing at the noise (see **SkyAreaCalculator**)
λ: is the wavelength the signal is calculated for, expressed in Å
h: is the Planck constant expressed in $erg * s$
c: is the speed of light expressed in Å$/sec$
SQF: is the spectral quantum factor (see **SpectralQuantumFactorCalculator**)

22 Delta Lambda



The **DeltaLambdaCalculator** calculates the range of wavelengths (in Å) around a specific wavelength which overlap with it. The delta lambda ($\Delta\lambda$) has meaning only for spectroscopy and it should not be used for any other case.

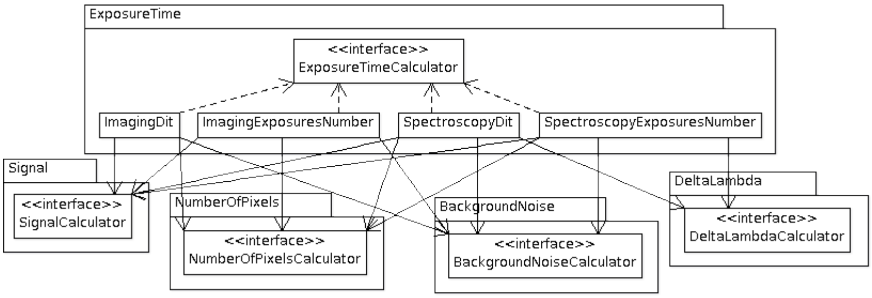
22.1 SpectralResolutionElement

This method calculates ranges in spectral resolution elements. It uses the equation:

$$\Delta\lambda_{(\lambda)} = SpectralBinning * \frac{\lambda}{\Delta\lambda_{/pixel(\lambda)}} \tag{37}$$

where
$SpectralBinning$: is the spectral binning (see **SpectralBinningCalculator**
λ: is the wavelength the $\Delta\lambda$ is calculated for, expressed in Å
$\Delta\lambda_{/pixel(\lambda)}$: is the spectral resolution per pixel of the instrument, expressed in Å(given by the user)

23 Exposure Time



The **ExposureTimeCalculator** is responsible for calculating the exposure time (expressed in s) for a fixed Signal to Noise Ratio. The different calculation methods are all based on the same equation:

$$SNR_{ref} = \frac{S_{(\lambda_{ref})} * T}{\sqrt{S_{(\lambda_{ref})} * T + BN_{(\lambda_{ref})} * T + N_{pix} * (D * T + n_{exp} * R^2)}} \tag{38}$$

where
T: is the exposure time to be calculated
SNR_{ref}: is the fixed signal to noise ratio (given by the user)
λ_{ref}: is the wavelength for which the SNR is given, expressed in Å(given by the user)
S: is the signal (see **SignalCalculator**)
BN: is the background noise (see **BackgroundNoiseCalculator**)
N_{pix}: is the number of CCD pixels affected by the target light (see **NumberOfPixelsCalculator**)
D: is the dark current expressed in $e^{-}/s/pixel$ (given by the user)
n_{exp}: is the number of exposures (given by the user)
R: is the readout noise expressed in $e^{-}/pixel$ (given by the user)

23.1 DIT - Number of exposures

The user can select one of two options, give the number of exposures or give the detector integration time (DIT). In the case the user gives the number of exposures, the equation 38 is solved for T as is. In the case the user gives the DIT, the number of exposures is replaced with:

$$n_{exp} = \frac{T}{DIT} \tag{39}$$

and the resulting equation is solved for T.

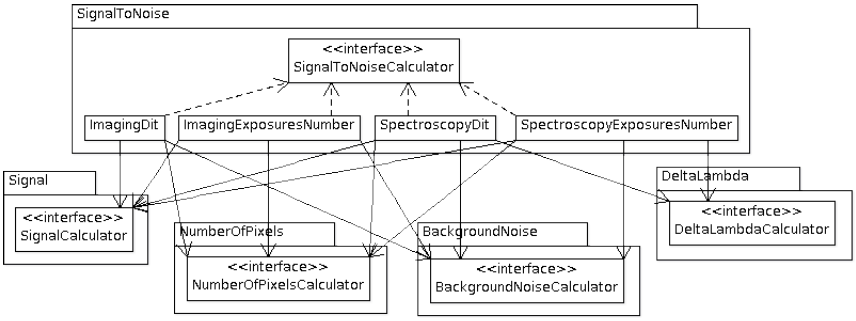
23.2 Imaging

In the case of imaging, the SNR_{ref} given by the user is the signal to noise ratio for the total filter wavelength range. In this case, the $S_{(\lambda_{ref})}$ and $BN_{(\lambda_{ref})}$ of the equation 38 are calculated by integrating the signal $S_{(\lambda)}$ and the background noise $BN_{(\lambda)}$ on the filter wavelength range (by using the values returned by the **SignalCalculator** and the **BackgroundNoiseCalculator**). These values are then replaced in the equation 38 and it is solved for T.

23.3 Spectroscopy

In the case of spectroscopy, the SNR_{ref} given by the user is the signal to noise ratio for the spectral element around the λ_{ref}. In this case, the $S_{(\lambda_{ref})}$ and $BN_{(\lambda_{ref})}$ of the equation 38 are calculated by using the **SignalCalculator** and the **BackgroundNoiseCalculator** and multiplying the result by $\Delta\lambda$, calculated by using the **DeltaLambdaCalculator**. These values are then replaced in the equation 38 and it is solved for T.

24 Signal To Noise



The **SignalToNoiseCalculator** is responsible for calculating the SNR for a fixed exposure time or for one calculated by the **ExposureTimeCalculator**. The different calculation methods are all based on the same equation:

$$SNR_{(\lambda)} = \frac{S_{(\lambda)} * T}{\sqrt{S_{(\lambda)} * T + BN_{(\lambda)} * T + N_{pix} * (D * T + n_{exp} * R^2)}} \tag{40}$$

where
$SNR_{(\lambda)}$: is the signal to noise ratio to be calculated
T: is the exposure time expressed in s (given by the user or calculated by the **ExposureTimeCalculator**)
λ: is the wavelength for which the SNR is being calculated for, expressed in Å
S: is the signal (see **SignalCalculator**)
BN: is the background noise (see **BackgroundNoiseCalculator**)
N_{pix}: is the number of CCD pixels affected by the target light (see **NumberOfPixelsCalculator**)
D: is the dark current expressed in $e^{-}/s/pixel$ (given by the user)
n_{exp}: is the number of exposures (given by the user)
R: is the readout noise expressed in $e^{-}/pixel$ (given by the user)

24.1 DIT - Number of exposures

Like for the exposure time calculation, if the user gives the detector integration time (DIT), then the n_{exp} on equation 40 is replaced with:

$$n_{exp} = \frac{T}{DIT} \tag{41}$$

24.2 Imaging

In the case of imaging, the ETC gives as result the SNR for the total filter wavelength range. In this case, the $S_{(\lambda)}$ and $BN_{(\lambda)}$ of the equation 40 are calculated by integrating the signal and the background noise on the filter wavelength range (by using the values returned by the **SignalCalculator** and the **BackgroundNoiseCalculator**). These values are then replaced in the equation 40 and is calculated the SNR for all the filter range.

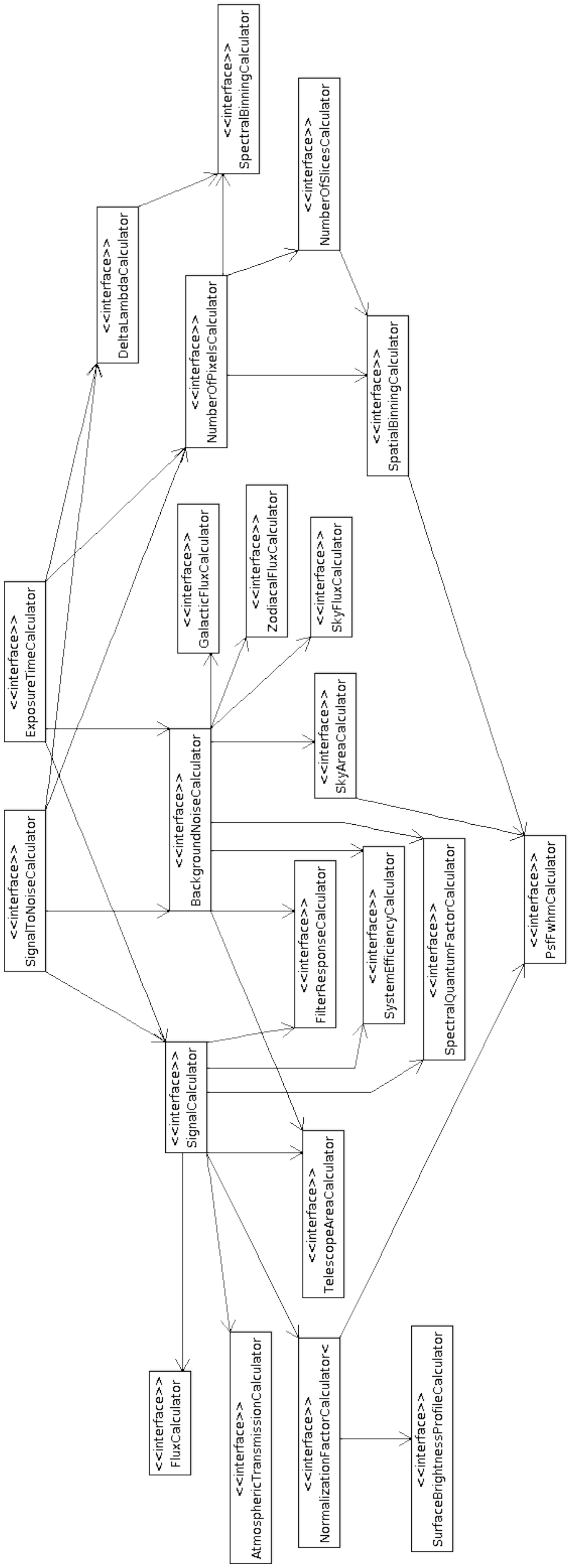
24.3 Spectroscopy

In the case of spectroscopy, the ETC gives as result an array of SNR values, one for each spectral resolution element, for the range of wavelengths given by the user. The procedure for this calculation is the following:

- By using the **DeltaLambdaCalculator** and the wavelength range given by the user, is calculated an array with the wavelengths representing the center of the spectral resolution elements.
- The signal and the background noise values for these wavelengths are calculated by using the **SignalCalcualtor** and **BackgroundNoiseCalculator**.
- The above values are multiplied with the $\Delta\lambda$, to calculate the total signal and the total background noise for each spectral element unit.
- Finally, these values are used to calculate the array of SNR values, by using the equation 40.

25 Overall diagram

The following diagram shows the overall dependencies over the different calculators.



<<interface>>
SignalToNoiseCalculator
<<interface>>
ExposureTimeCalculator
<<interface>>
DeltaLambdaCalculator
<<interface>>
SpectralBinningCalculator
<<interface>>
NumberOfPixelsCalculator
<<interface>>
NumberOfSlicesCalculator
<<interface>>
SpatialBinningCalculator
<<interface>>
BackgroundNoiseCalculator
<<interface>>
GalacticFluxCalculator
<<interface>>
ZodiacalFluxCalculator
<<interface>>
SkyFluxCalculator
<<interface>>
SkyAreaCalculator
<<interface>>
FilterResponseCalculator
<<interface>>
SystemEfficiencyCalculator
<<interface>>
SpectralQuantumFactorCalculator
<<interface>>
PsfFwhmCalculator
<<interface>>
SignalCalculator
<<interface>>
TelescopeAreaCalculator
<<interface>>
FluxCalculator
<<interface>>
AtmosphericTransmissionCalculator
<<interface>>
NormalizationFactorCalculator<
<<interface>>
SurfaceBrightnessProfileCalculator