

GammaLib - Feature #611

Modify IRF interface

12/04/2012 12:24 PM - Knödlseider Jürgen

Status:	Closed	Start date:	12/04/2012
Priority:	High	Due date:	
Assigned To:	Knödlseider Jürgen	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	00-07-00		

Description

The actual IRF interface has a lot of arguments for all observed (GInstDir, GEnergy, and GTime) and true (GSkyDir, GEnergy, and GTime) parameters. A more generic interface, allowing also for a more efficient implementation, is provided by regrouping the parameters into classes.

For the observed parameters, the GEvent class may be used.

For the true parameters, the GPhoton class may be used. This allows also easy extension to handle for example polarization.

The GResponse method should then have the following interfaces:

```
virtual double irf(const GEvent& event,
                  const GPhoton& photon,
                  const GObservation& obs) const = 0;
virtual double npred(const GPhoton& photon,
                   const GObservation& obs) const = 0;
```

Additional, the following interfaces can be used for the GResponse virtual methods:

```
virtual double irf(const GEvent& event,
                  const GModelSky& model,
                  const GEnergy& srcEng,
                  const GTime& srcTime,
                  const GObservation& obs) const;
virtual double irf_ptsrc(const GEvent& event,
                       const GModelPointSource& model,
                       const GEnergy& srcEng,
                       const GTime& srcTime,
                       const GObservation& obs) const;
virtual double irf_extended(const GEvent& event,
                          const GModelExtendedSource& model,
                          const GEnergy& srcEng,
                          const GTime& srcTime,
                          const GObservation& obs) const;
virtual double irf_diffuse(const GEvent& event,
                         const GModelDiffuseSource& model,
                         const GEnergy& srcEng,
                         const GTime& srcTime,
                         const GObservation& obs) const;
virtual double npred(const GModelSky& model,
                   const GEnergy& srcEng,
                   const GTime& srcTime,
                   const GObservation& obs) const;
virtual double npred_ptsrc(const GModelPointSource& model,
                         const GEnergy& srcEng,
                         const GTime& srcTime,
                         const GObservation& obs) const;
virtual double npred_extended(const GModelExtendedSource& model,
                             const GEnergy& srcEng,
                             const GTime& srcTime,
```

```

        const GObservation&    obs) const;
virtual double npred_diffuse(const GModelDiffuseSource& model,
        const GEnergy&        srcEng,
        const GTime&         srcTime,
        const GObservation&    obs) const;

```

It remains to be seen whether the remaining pairs of GEnergy and GTime should be combined in a GPhoton object where the spatial direction is ignored. Alternative, a new class GSource could be introduced that has (GModelSky, GEnergy and GTime) as members. GSource could be even a base class that implements GPointSource, GDiffuseSource, and GExtendedSource

History

#1 - 12/04/2012 11:04 PM - Knödseder Jürgen

- Target version set to 00-07-00

#2 - 12/04/2012 11:38 PM - Knödseder Jürgen

- Status changed from New to In Progress

- % Done changed from 0 to 10

The work has been started on the irf() and npred() interfaces. So far, the COMPTEL interface has been adapted.

#3 - 12/05/2012 12:36 AM - Knödseder Jürgen

- % Done changed from 10 to 30

All observed parameters have now been replaced by a GEvent argument.

The actual interface is:

```

virtual double irf(const GEvent&    event,
        const GPhoton&    photon,
        const GObservation& obs) const = 0;
virtual double npred(const GPhoton&    photon,
        const GObservation& obs) const = 0;

virtual double irf(const GEvent&    event,
        const GModelSky&    model,
        const GEnergy&    srcEng,
        const GTime&    srcTime,
        const GObservation& obs) const;
virtual double irf_ptsrc(const GEvent&    event,
        const GModelPointSource& model,
        const GEnergy&    srcEng,
        const GTime&    srcTime,
        const GObservation& obs) const;
virtual double irf_extended(const GEvent&    event,
        const GModelExtendedSource& model,
        const GEnergy&    srcEng,
        const GTime&    srcTime,
        const GObservation& obs) const;
virtual double irf_diffuse(const GEvent&    event,
        const GModelDiffuseSource& model,
        const GEnergy&    srcEng,
        const GTime&    srcTime,
        const GObservation& obs) const;
virtual double npred(const GModelSky&    model,
        const GEnergy&    srcEng,
        const GTime&    srcTime,
        const GObservation& obs) const;
virtual double npred_ptsrc(const GModelPointSource& model,
        const GEnergy&    srcEng,
        const GTime&    srcTime,
        const GObservation& obs) const;
virtual double npred_extended(const GModelExtendedSource& model,
        const GEnergy&    srcEng,
        const GTime&    srcTime,

```

```
        const GObservation&    obs) const;  
virtual double npred_diffuse(const GModelDiffuseSource& model,  
        const GEnergy&        srcEng,  
        const GTime&         srcTime,  
        const GObservation&    obs) const;
```

Now I have to see whether the source parameters should all be combined in a new GSource class.

#4 - 12/06/2012 02:12 AM - Knödseder Jürgen

- % Done changed from 30 to 100

All interfaces have been changed. The result has been merged into the devel branch.

#5 - 12/06/2012 02:13 AM - Knödseder Jürgen

- Status changed from In Progress to Closed