

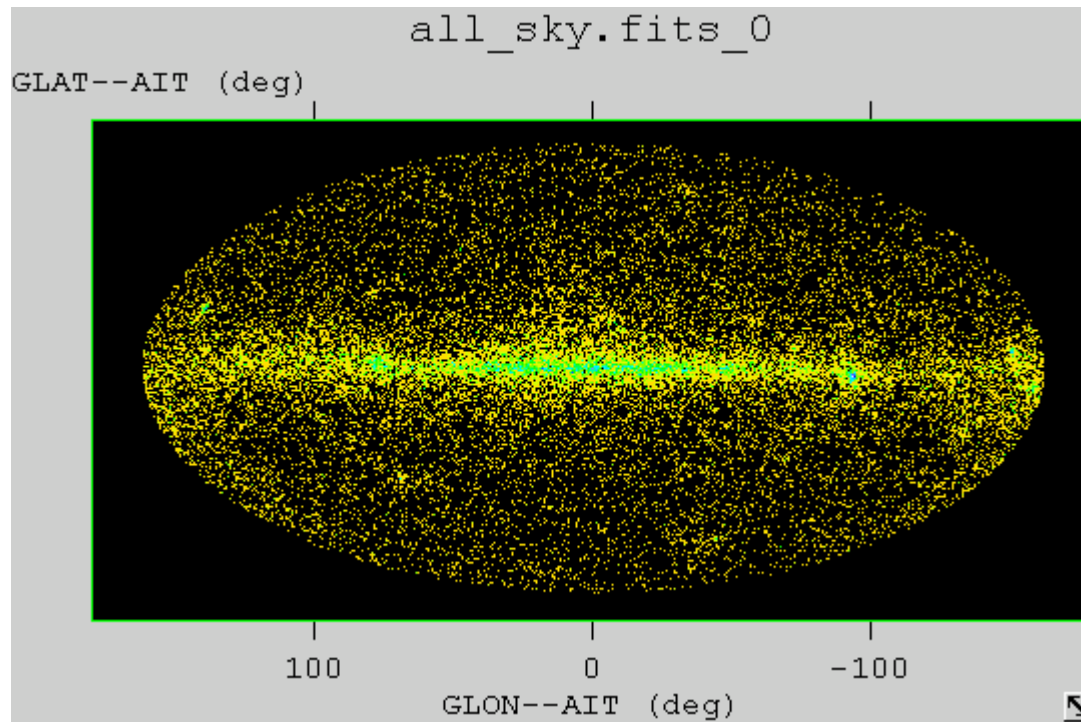
Maps: counts and exposure

Toby Burnett

Count Maps

- I wrote my own `count_map` tool, instead of using `EventBin (CMAP)`, and gained the additional capabilities:
 - supports galactic coordinates!
 - filenames can have absolute paths if start with "/"
 - can specify a filter string
 - read ROOT, FITS, or text files; my interface to FITS is flexible, not restricted to FT1
 - results saved as floats
 - easily customizable—all logic is accessible at the top level
- WHY??? Why not just modify the `GSSC` code, or ask for these features?

A result, as show by the fv tool



The main, for those interested

```
int main(int argc, char * argv[]) {
    try{

        // read in, or prompt for, all necessary parameters, using PIL
        CountMapParameters pars(argc, argv);

        // set up the input data source as a container (from tuple package)
        tuple::RootTable data( pars.eventFile());

        // create the image object
        SkyImage image(pars);

        // define a function object to analyze each row, then apply it to all selected rows
        AddCount counter(pars, data, image);
        std::for_each( data.begin(), data.end(), counter);

        // open output FITS image file (using temporary table package)
        table::FitsService iosrv;
        iosrv.createNewFile(pars.outputFile(), pars.templateFile());

        // write out the result
        image.write(&iosrv);   iosrv.closeFile();

    }catch( const std::exception& e){
        std::cerr << "caught exception: " << e.what() << std::endl;
        return 1;
    }
    return 0;
}
```

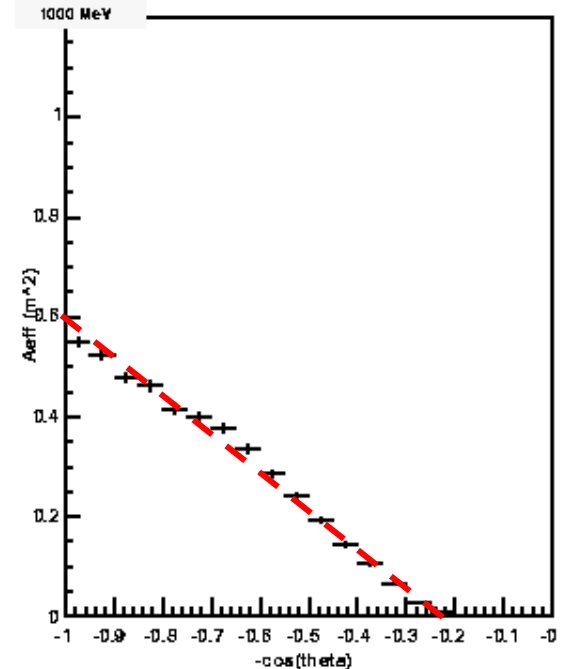
application is a
collaboration
among 5
objects

And the pil parameter file

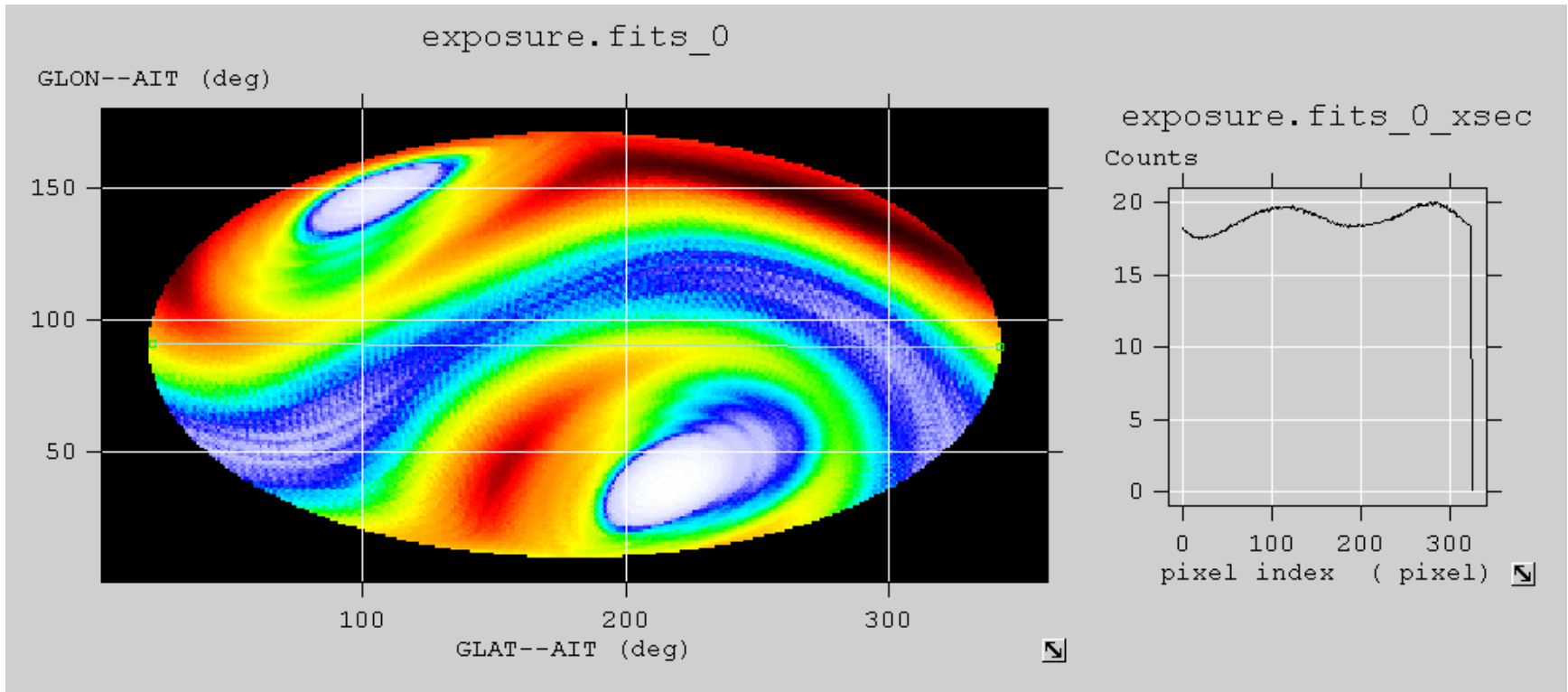
```
clobber, b, a, "YES", , , "Overwrite existing output file?: "  
chatter, i, a, 10, , , , "Chatter: "  
mode, s, a, "h", , , ""  
# uncomment the following for interactive/replacement mode  
#mode, s, a, "ql", , , ""  
#-----  
#  
eventfile,s, a, "/glast/DC1/root_files/DC1_goodGamma.root", , , "Name of the Event Data File:"  
filter, s, a, "Mc_src_Id/1000==4", , , "filter expression:"  
outfile, s, a, "blazars.fits", , , "Name of the output file:"  
#  
ra_name, s, a, "FT1Ra", , , "name of the RA field"  
dec_name, s, a, "FT1Dec", , , "name of the DEC field"  
#-----  
#Parameter for Spatial binning  
#  
npix, i, a, 1024, , , "Total pixel size of the image in each axis: "  
npixy, i, a, 512, , , "Total pixel size of the image in y axis: "  
imgsize, i, a, 360, , , "Total physical size of the image(x): "  
imgsizey, i, a, 180, , , "Total physical size of the image (y): "  
xref, r, a, 0, , , "RA at the image center: "  
yref, r, a, 0, , , "DEC at the image center: "  
rot, r, a, 0, , , "Rotation angle of Y-axis relative to the physical coordinate: "  
projtype, s, a, "AIT", , , "Projection Method (CAR, SIN, TAN, ARC, NCP, GLS, MER, AIT, STG): "  
use1b, b, a, "YES", , , "use l,b instead of ra,dec :"
```

Maps: what is the exposure?

- No tool yet to make a simple exposure map, so use the same framework
- Ingredients:
 - Measured angular dependence of the effective area: \sim linear in $\cos\theta$ to zero at 0.25, independent of energy
 - History file (I use the ascii file that we used to generate the pointing)
- Processing
 - create a 3-d histogram of time in bins of ra_z , dec_z , $\cos\theta$ (J. McEnery suggestion) (An Exposure object)
 - create a SkyImage object with given pixels and transformation, all from standard pil file
 - Have the SkyImage object request values for each of its pixels from the Exposure object, via a functor Requester.



Exposure: the 1-day map



Units are percent of total exposure.

scales wrong:
standard AIT
projection

profile along the
galactic equator

What is a functor?

- Example: the effective area used for the exposure map

```
/** @class Aeff
  @ brief function class implements effective area
  */
class Aeff : public Exposure::Aeff{
public:
  Aeff(double cutoff=0.25):m_cutoff(cutoff){}

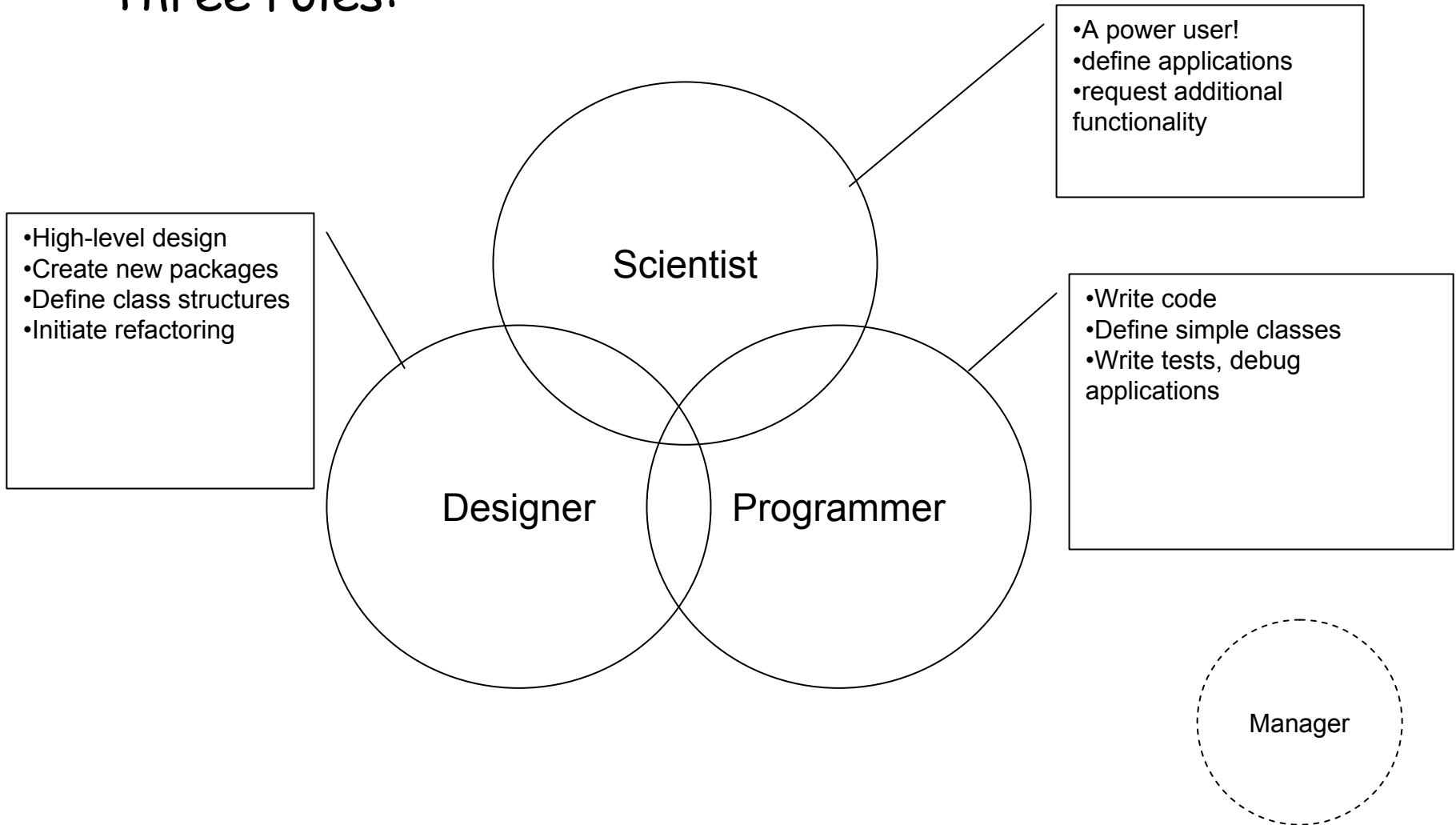
  double operator()(double costh) const
  {
    if(m_cutoff==2.) return 1.0;
    return costh<m_cutoff? 0 : (costh-m_cutoff)/(1.-m_cutoff);
  }

  double m_cutoff;
};
```

the heart: an
operator that
does the work

Thoughts on Scientific Software Design

- Three roles:



Observations

- Best case: one person fulfills all 3 roles (sort of like yours truly)
- OK, but not as good: three people, but good communication
- Worst case: no designer, scientist and programmer are different people, infrequent communication