The GLAST Object Oriented Data Interface (GOODI)

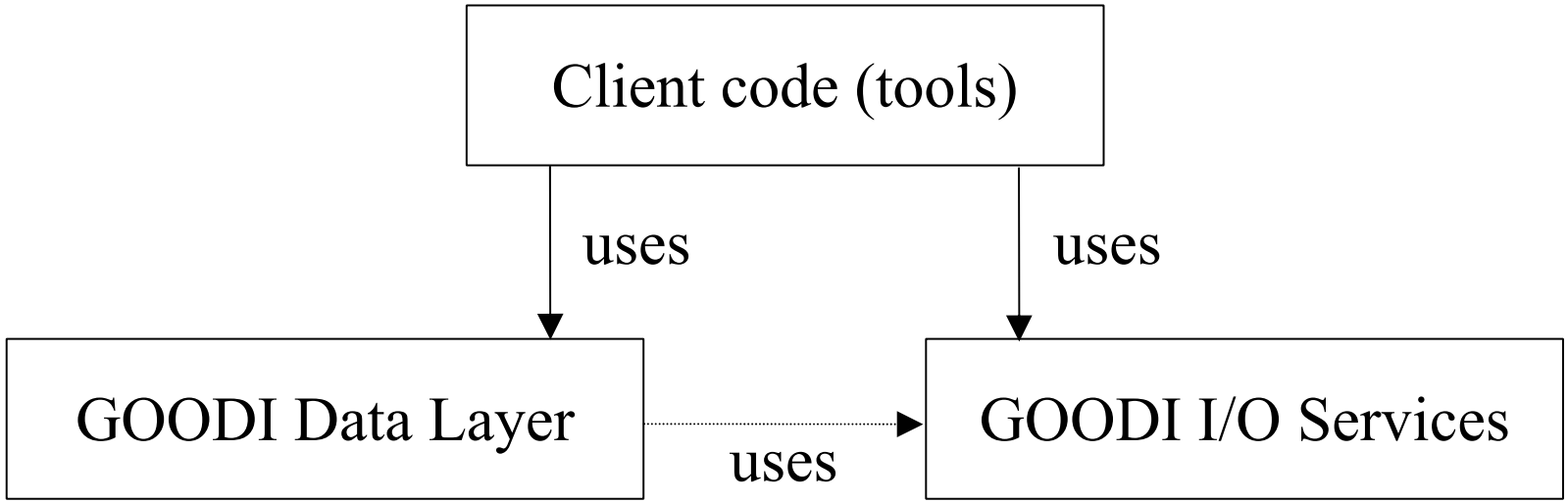
James Peachey, HEASARC

Sandhia Bansal, GSSC

GOODI Overview

- Provides high-level abstractions for data types (event data, binned data, etc.)
- Data type definitions use FITS concepts and language, (e.g. keywords and data columns).
- Provides file I/O for these types while insulating data abstractions from underlying file format.

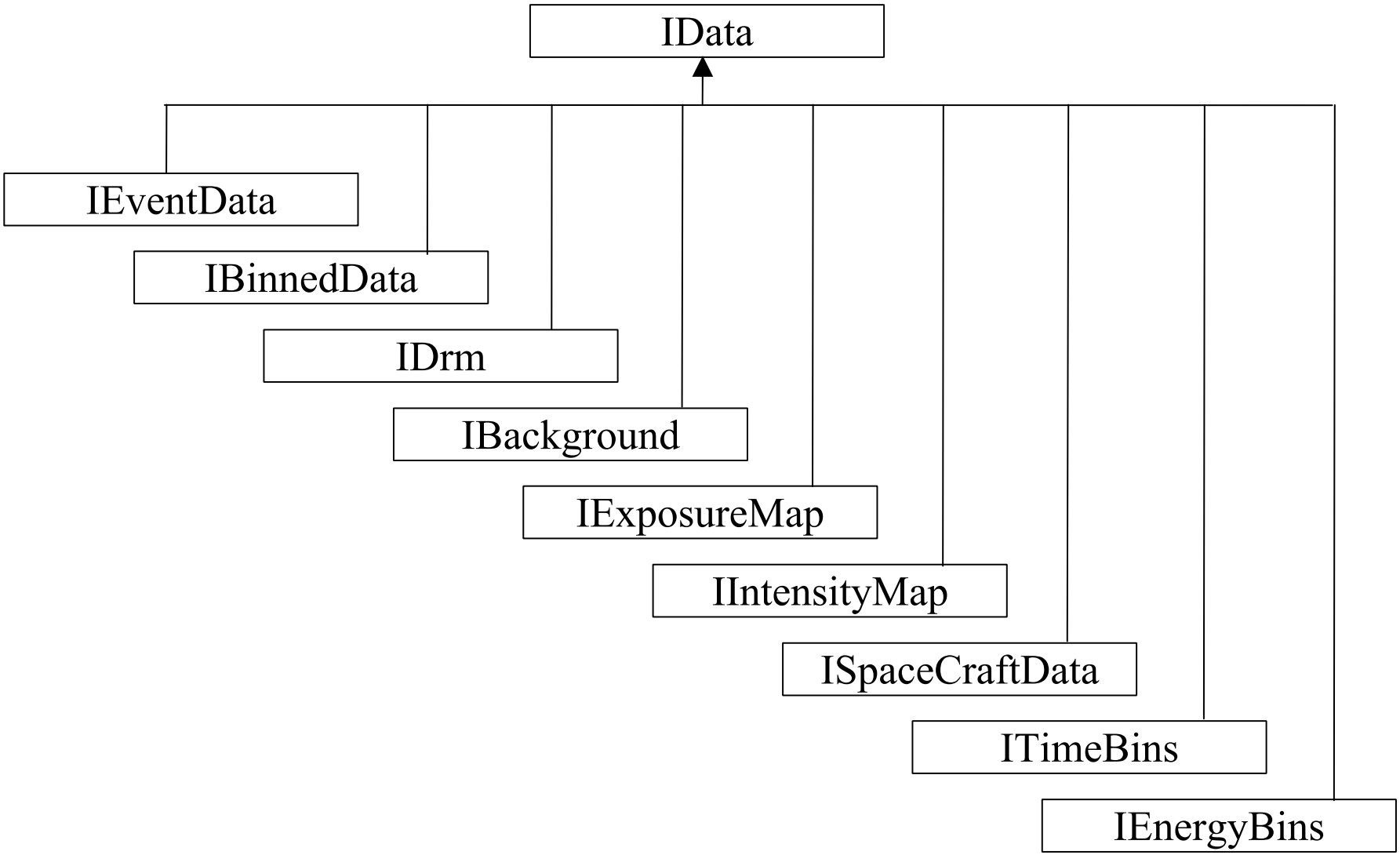
How GOODI is structured



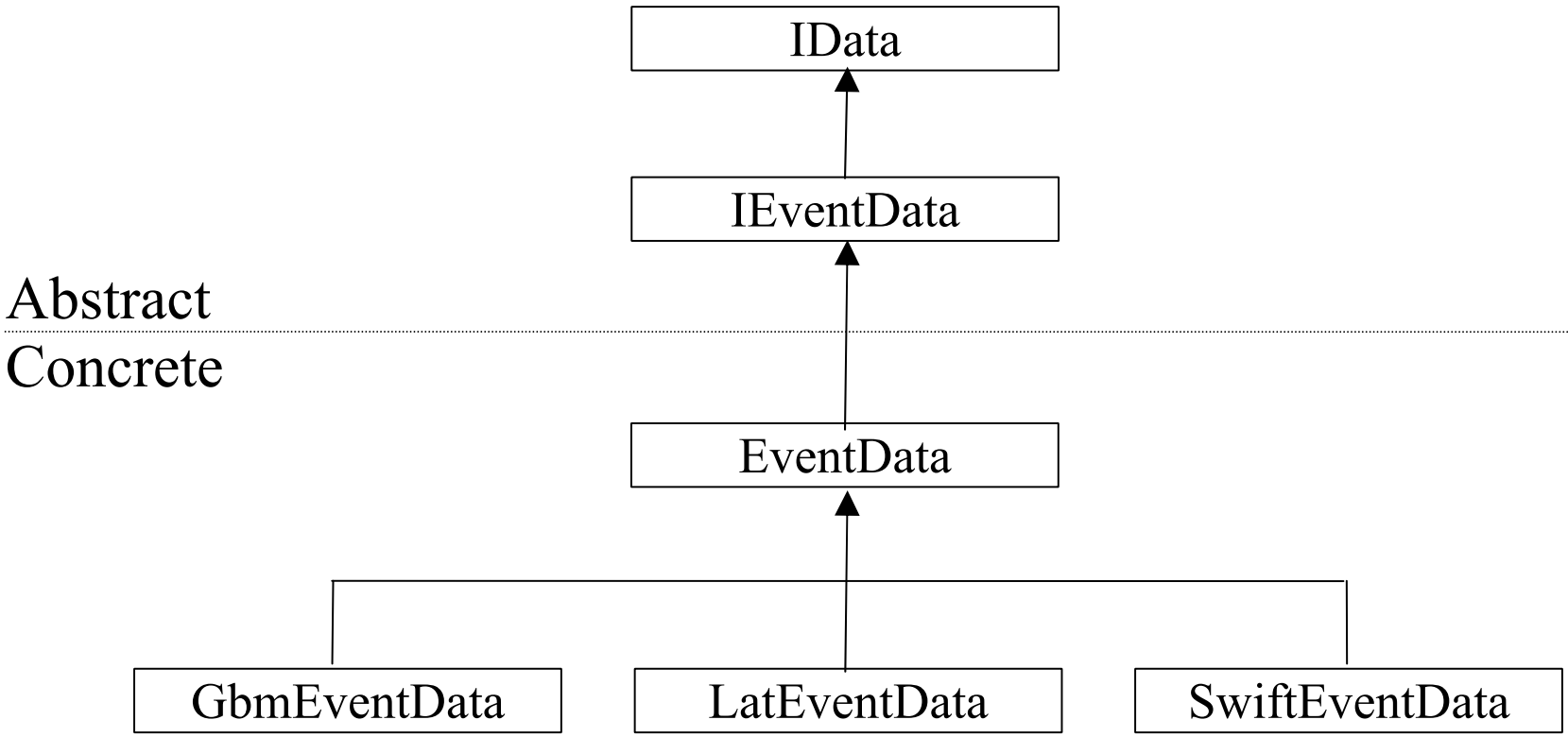
- Data objects use FITS concepts, but are not tied to the FITS format.
- Methods provided to convert data to C++ standard library (STL) containers.

- Abstract interface with concrete implementation for FITS format using Cfitsio.
- Root implementation possible/planned.

GOODI Data Layer Inheritance Tree



Event Data Classes



Client Pseudocode Sample 1: Creating a Data Object from a File

```
// Use a factory object to create an I/O service object.
IDataIOService * file = FileFactory.create("filename.fits");

// From this I/O service object, create a data object.
IEventData * eventData = EventDataFactory.create(file);

// (Check to make sure this succeeded...)

// Read the desired data from the I/O service object.
eventData->readAllKeys(file);
eventData->readTime(file);

// Obtain a keyword from the data object.
std::string telescope;
eventData->getKey("telescop", telescope);

// Obtain a container of data from the data objet.
std::vector<double> &time = eventData->time();
```

Client Pseudocode Sample 2: Creating a File from a Data Object

```
// Use a factory object to create an I/O service object.
// The underlying file is also created with the help of a
// FITS template file.
IDataIOService * file = FileFactory.create("filename.fits", "FITS-
template.tpl");

// (Check to make sure this succeeded...)

// Assume the events data already exists. Write its
// data to the I/O service object.
eventData->write(file);
```

Future Work

- Provide Root implementation of the I/O service, possibly coordinating the work with Reiner Rohlfs, the author of AstroRoot.
- Finish fleshing out the rest of the IData inheritance tree.

Appendix: EventData Inheritance Tree