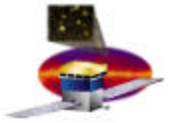
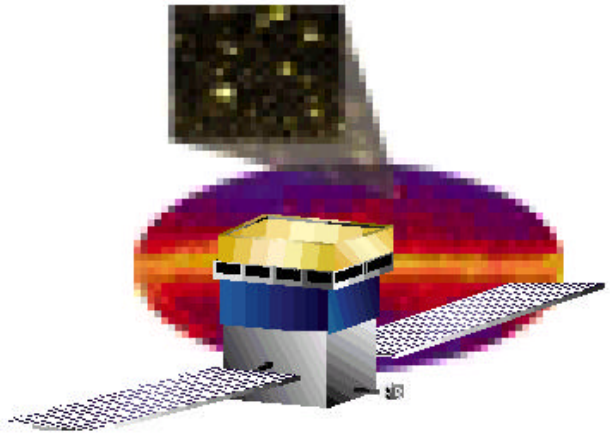


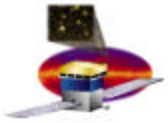
Root, GLAST, and IDL

Heather Kelly
NASA/GSFC
Emergent Corporation

Gamma-ray Large Area Space Telescope

- **Joint NASA/DOE project, with numerous collaborators from many countries.**
- **Pair Conversion Telescope – designed to detect photons in the energy range of 10 keV – 300 GeV.**
- **Scheduled for launch March, 2006.**
- **Modular 4x4 design of identical towers. Each tower consists of a Tracker and CsI Calorimeter. The 4x4 array is covered by an Anti-Coincidence Shield to reject charged cosmic rays.**

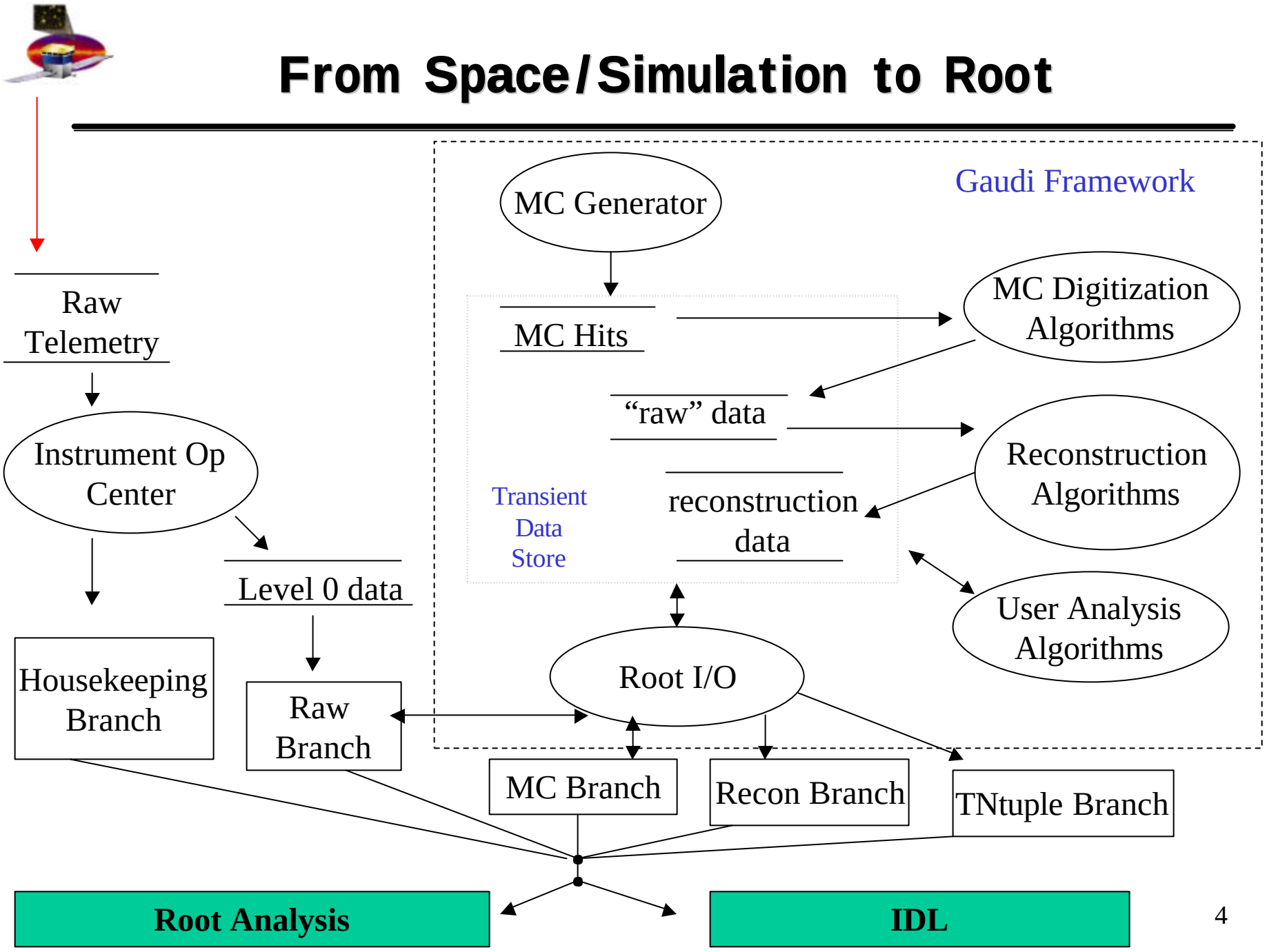




GLAST's Use of Root

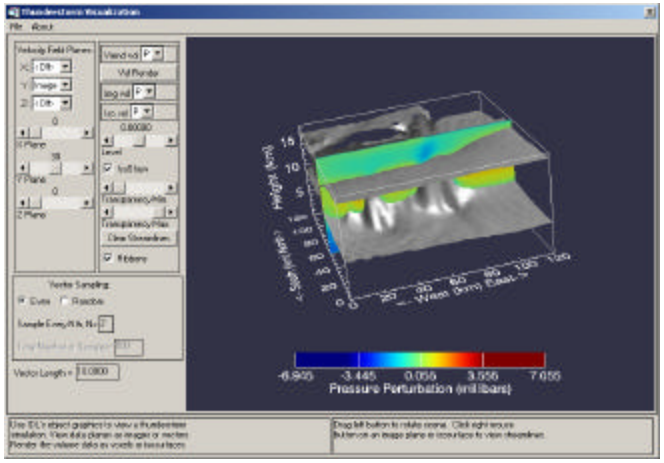
- First use of Root – 1999 SLAC beamtest.
  - For both data storage and analysis
- Created our first Root classes to store raw detector data.
- Mid-2000 created Root classes to store our reconstructed data.
- Since then, we have migrated our simulation and reconstruction algorithms to the Gaudi framework.
- Currently preparing for a balloon flight this summer.
  - Now we will store HSK data in Root as well.
- Also gearing up for bulk Monte Carlo production.
  - Plan to store MC data in Root.
  - We are now prepared to handle Root I/O to store output data and to re-ingest that data for re-processing.

From Space/Simulation to Root



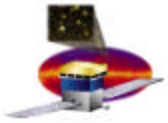
Interative Data Language

- A commercial product available from Research Systems Inc.
- It is an analysis tool kit, familiar to many astrophysicists and earth scientists.
- Very polished professional software, supported on a variety of platforms: Unix, Windows, VMS.
- Strong visualization tools.
- Accepts data in a variety of formats including FITs and HDF.
- Drawbacks: Expensive, commercial software
- There are some GLAST collaborators who are faithful IDL users!



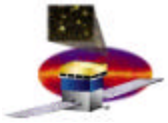
How will GLAST support IDL users?

- **Do nothing.**
  - **We have limited resources, we could choose to just support Root as our analysis toolkit and provide C++ code to access the data.**
- **Convert Root data into some other format that IDL will ingest.**
  - **Do we want 2 persistent copies of the same data?**
- **Provide a mechanism for IDL users to access Root files directly from within IDL.**
  - **Root is an I/O library and an analysis toolkit.**
  - **Allow users to choose what analysis toolkit they wish to use.**
  - **Good for the users and good for Root.**
    - **In the end…it is about accessing the data.**
    - **Many users have extensive libraries of IDL routines.**
    - **Some are not interested in learning C++.**
    - **Others are very excited to use Root as an analysis toolkit.**
    - **We provide data access and a set of common routines written in C++ that will be accessible to both Root and IDL users.**



Root2IDL: Proof of Concept

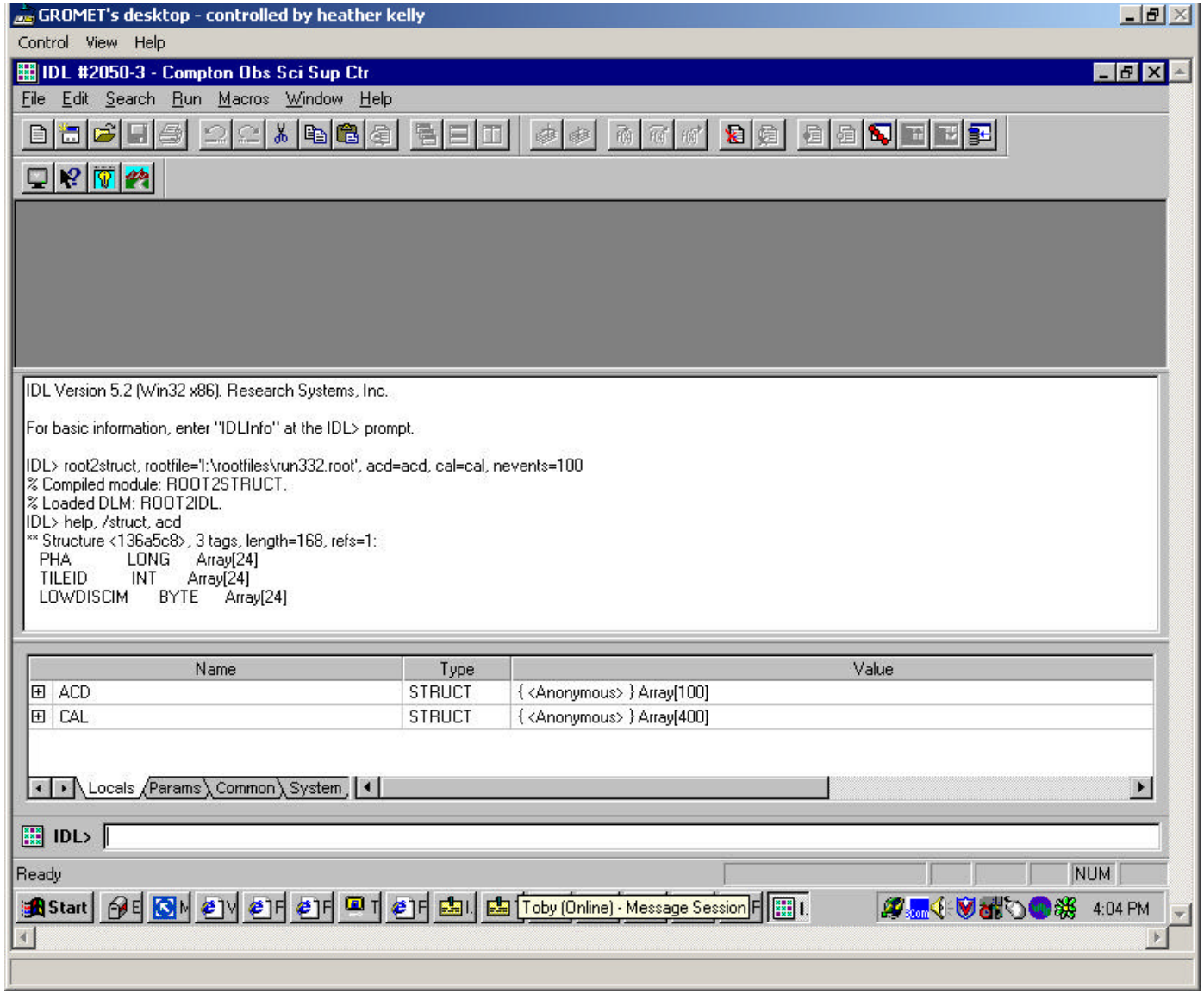
- For our 1999 beamtest – stored our data in Root trees.
- IDL is written in ANSI C – on all platforms.
- IDL provides a mechanism to call external routines through Dynamic Load Modules (shareable library).
- A DLM provides a common interface to IDL, that allows IDL to call the routines within a user-defined shared library.
  - The fact that the routine is external is transparent to the user.
    - Behaves as an IDL system routine.
- For the beamtest, a DLM, Root2IDL, was created to handle the *specific* TTree structure we were using.
- The DLM reads in the Root tree data and converts it into an IDL structure (equivalent to a C structure).
- The data is then available within IDL.

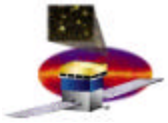


Example Interface to IDL

```
#include "export.h“   // Include file available with IDL distribution
static IDL_VPTR testfun(int argc, IDL_VPTR *argv)
{
   return IDL_StrToSTRING("TESTFUN");
}
int IDL_Load(void)
{
    static IDL_SYSFUN_DEF2 function_addr[ ] = {
       { testfun, "TESTFUN", 0, IDL_MAXPARAMS, 0, 0},
     };
    /*   Register our routine.*/
    return  IDL_SysRtnAdd(function_addr, TRUE, ARRLEN(function_addr));
}
```

Root2IDL in Action





Problems / To Do List

- IDL's DLM interface has migrated in versions 5.2…5.4.
- Need a generic interface for all Root files.
  - Our Root classes will certainly evolve – plus a general utility may be useful to the community at large.
  - This is possible due to TStreamerInfo!
- Optimization is necessary.

- Plan
  - Upgrade the Root2IDL DLM to handle our specific Root class structure for our upcoming balloon flight and MC simulations.
  - Once things settle down, we can then proceed to provide a generic utility for any Root file.

Suggestions for Root

- **Windows support**
  - **Many GLAST collaborators are Windows users and will perform some analysis using Root interactively on Windows. There are quirks in the GUI.**
- **Root and Gaudi**
  - **Projects also using Gaudi as their framework would love to see a persistency mechanism that handles "real" Root.**
- **It is difficult to "keep up" with the upgrades in Root from version to version. We may not always be able to upgrade in a timely manner.**
- **Some Great New Things about Root**
  - **Dedicated users**
  - **Tree Friendship**
  - **Apache server**