ATLAS GRAPHICS WG

7 MAY 98

- Overall Strategy
- Design & Core Code
- Current Status
- Arve/Aravis
- Atlantis
- Wired
- Java
- (G)UI
- SubPackages
- Data Sources

OVERALL STRATEGY

- Control Structure
- Data Sources
- SubSystem (Development and Interfacing)
- User Interface

- Ultimate Goal
- Current Status
- Path

CONTROL STRUCTURE

- Ultimate Goal
  - "Unvisible" glue for Data, (G)UI and Graphics
  - Natural component of *Arve*
- Current Status
  - Fundamental functionality available
  - "Special features" (callback,...) missing
  - Only limited add-ons (*PlottableCollection*, *PlottableModelSet*)
  - Not working with Control Domain *Components*
- Path
  - Add new features to the environment without making it complicated
  - Change to work with Control Domain *Components*
  - Change to support "Layred Design" ?

DATA SOURCES

- Ultimate Goal
  - Views can run either standalone or linked (as dynamicaly as possible) with *Arve*
- Current Status
  - Events are just available, so far not build in SRT - so private built only, can read any Zebra tape
  - Also prototype for access to trigger data (*cTrig*) exists
  - Files to feed data to Views (Event executable is about 30MB alone)
  - Simplified files of example objects (*SiDetector*, *SiDigit*) available
  - Basic Tools for making existing classes Plottables (*createPlottableModel* script)
- Path
  - Continue with files
  - Make all required *Arve* objects available to all Views
  - Try to define common file format (*XML* ?)
  - Technicaly prepare (very dynamic) linking

SUBSYSTEMS
DEVELOPMENT AND INTERFACEING

- Ultimate Goal
  - Views can run as standalone as well as integrated into *Arve* (dynamical loading, *Components*)
  - Each Requirement is satisfied by some View
- Current Status
  - Standalone Views
  - Some Views themselves are developing well (*Atlantis*, *Wired*,...)
  - Nothing done for new histograming View candidates (*IE*, *Root*, ...)
- Path
  - Continue with independent development of Graphical Subsystems
  - Prepare hooks for interfacing Graphical Subsystems Directly
  - Look at histograming canddates more closely (Evaluation)

USER INTERFACE

- Ultimate Goal
  - Interchangable UI based on "Layered Design" ?
  - Command line for UI and scripting environment
  - *Object Browser* for GUI
- Current Status
  - Nothing exists
  - Preliminary ideas
- Path
  - Design and build
  - Collaboration with othre Domains (Control, Event)

DESIGN AND CORE CODE

- Current Design Questions
  - Object ownership (create, add/adopt)
  - const / no const
  - Multimethods
  - Tree structure (top-representation choice)
  - Modeler instead of PlottableModel
  - Default PlottableModel
  - Collaboration with Java
- Organisation of Repository
  - Views (+ bridge to External sw, if needed)
  - External sw
  - CrossDomains
  - Requirements to SRT
  - Hyperised code on WWW
  - Full CVS (with history) on WWW

CURRENT STATUS

- "If I have **MyPlottable**"
- edit **MyPlottable**
- createPlottableModel -d<origDomain> -c<crossDomain> MyPlottable
- edit **MyPlottableModel**, **VRMLMyPlottable**,...
- edit **Main**
- run
- see results

CURRENT STATUS

- edit **VRMLMyPlottable**

```
// MMM You shall edit VRMLSiDigit::VRMLSiDigit(SiDigitModel& aModel)
// MMM   to enable VRMLSiDigit to do something at all
VRMLSiDigit::VRMLSiDigit(SiDigitModel& aModel):
  _pModel(&aModel)
{
// EXAMPLE   _pSphere = new Sphere(_pModel->radius(),
// EXAMPLE                         SFVec3f(_pModel->x(),
// EXAMPLE                                 _pModel->y(),
// EXAMPLE                                 _pModel->z()));
// EXAMPLE   add(_pSphere);

  _pColor = new Color(SFColor(1, 0, 0));
  const SiDigit* dig = ((SiDigit*)(_pModel->plottable()));
  _pPoint = new Point(SFVec3f(dig->global_position().x(),
                              dig->global_position().y(),
                              dig->global_position().z()));
  add(_pColor);
  add(_pPoint);
}
```

CURRENT STATUS

- edit **Main**

```
TestEvent::TestEvent()
    :
    m_print_level(0),
    text("a.text"),
    vrml("a.wrl"),
    hbook("a.hbook", 11),
    atlantis("atlantis.txt")

{
    sceneList.add(text);
    sceneList.add(vrml);
    sceneList.add(hbook);
    sceneList.add(atlantis);
}

// ..............

    det_vector_type::const_iterator  first = collector.detectors_begin();
    det_vector_type::const_iterator  last  = collector.detectors_end();

    for ( ; first != last; ++first) {
         sceneList.add(*first);
```

CURRENT STATUS

- see *AsciiText*

```
SiDigit: pos(x,y,z),local(x,y,z),ch(rz,rphi),ph,(thres,noise,dead):
         -12.1  -0.2884   -47.08  0.1477       0  -1.375  36 125  2445    0    0
0
SiDigit: pos(x,y,z),local(x,y,z),ch(rz,rphi),ph,(thres,noise,dead):
         -12.46  -0.3185   -47.08  0.1427       0  -1.015  48 124  2433    0    1
0
SiDigit: pos(x,y,z),local(x,y,z),ch(rz,rphi),ph,(thres,noise,dead):
         -12.5  -0.2458   -47.08  0.2177       0  -0.985  49 139 15855    0    0
0
```

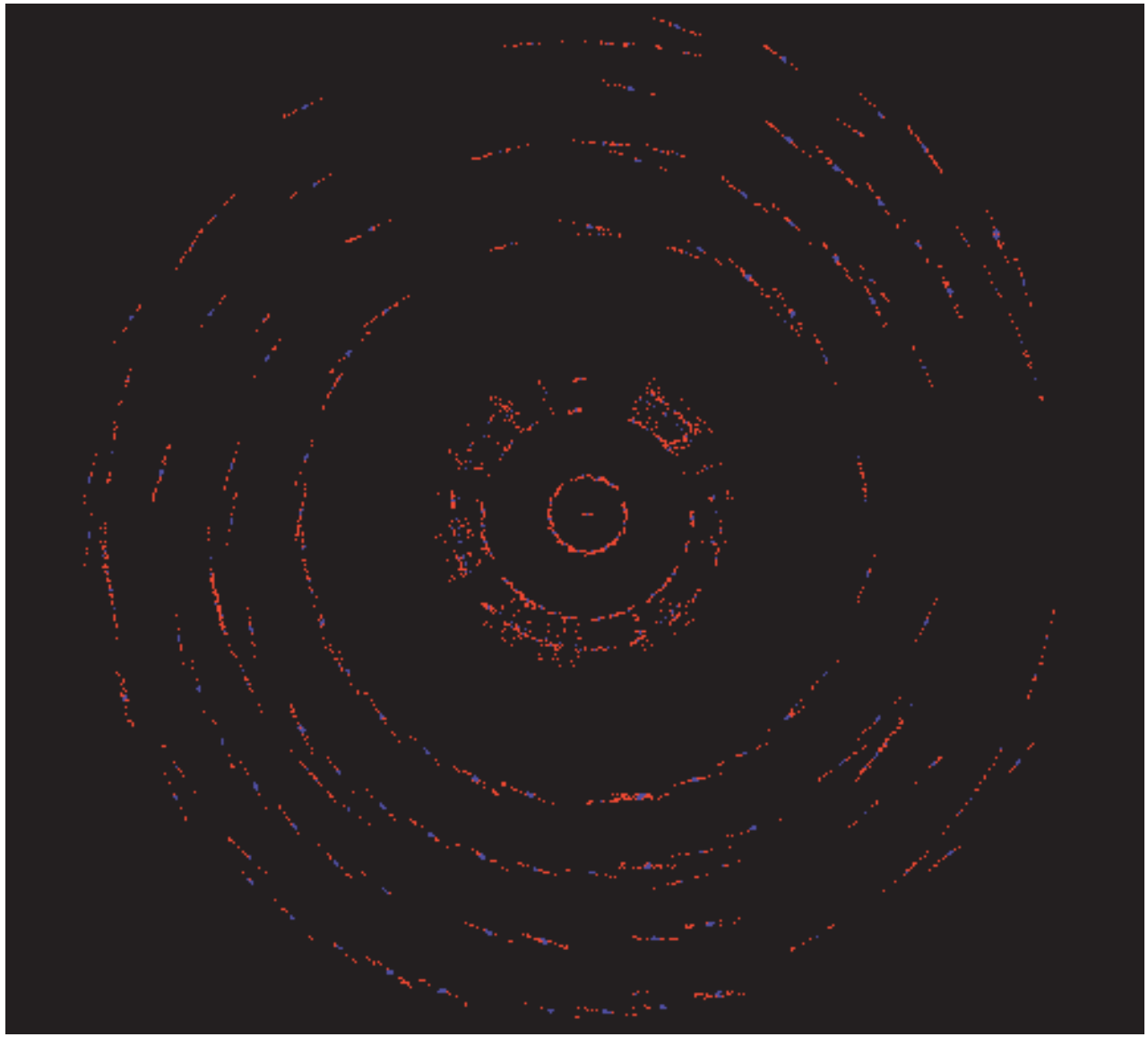
CURRENT STATUS

- see *AtlantisText*

```
E       Event Number =  1
*       IDRUN        =  1      NEVENT  =      1
*
*
S       164     det     region  lay     12.7147 -0.916635       -47.0825
S       164     det     region  lay     12.715  -0.917028       -47.0825
S       0       det     region  lay     13.4159 -0.872702       -47.0825
S       0       det     region  lay     13.5361 -0.873705       -47.0825
S       0       det     region  lay     14.616  -0.87242        -47.0825
```

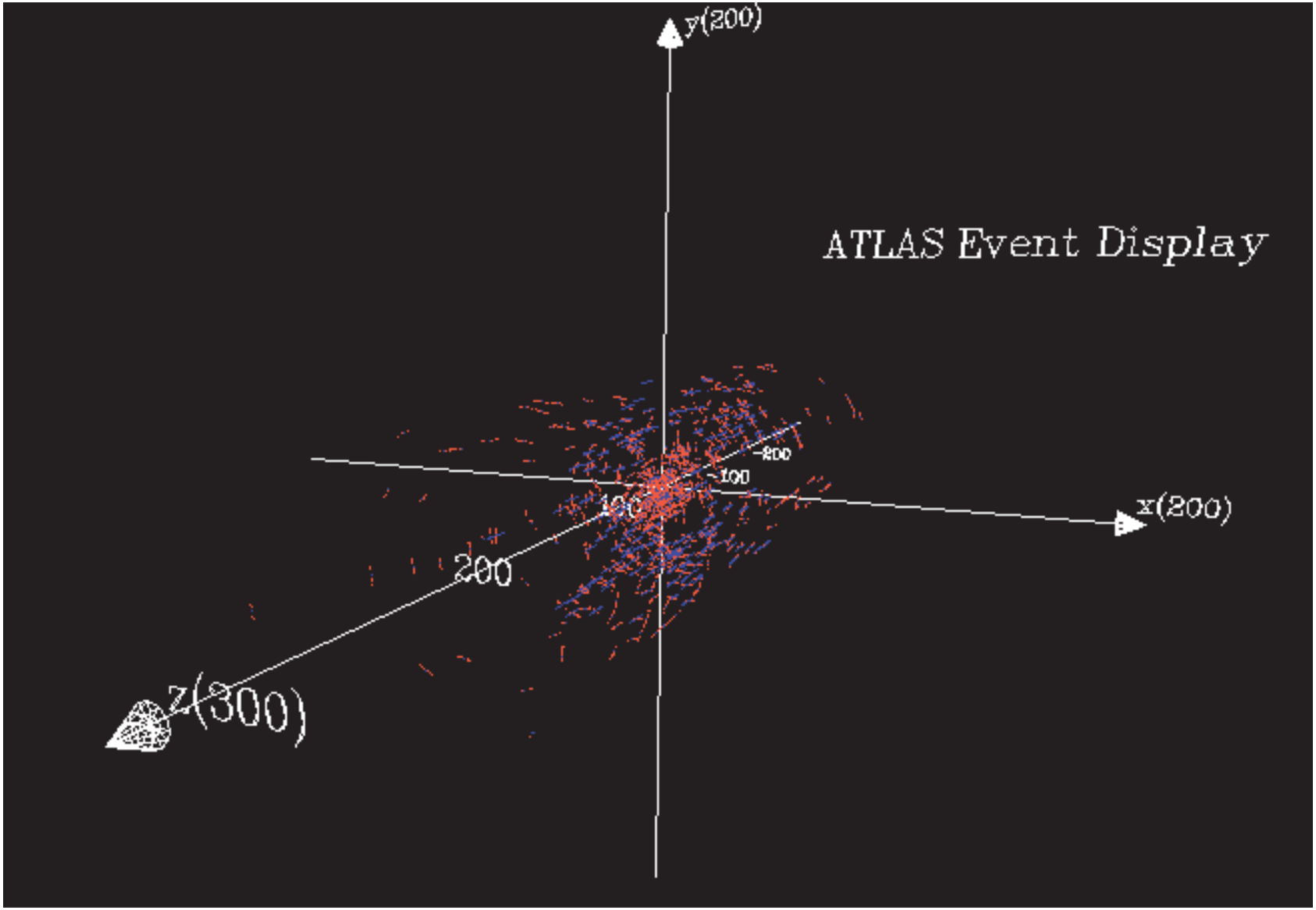
CURRENT STATUS

- see *VRML*



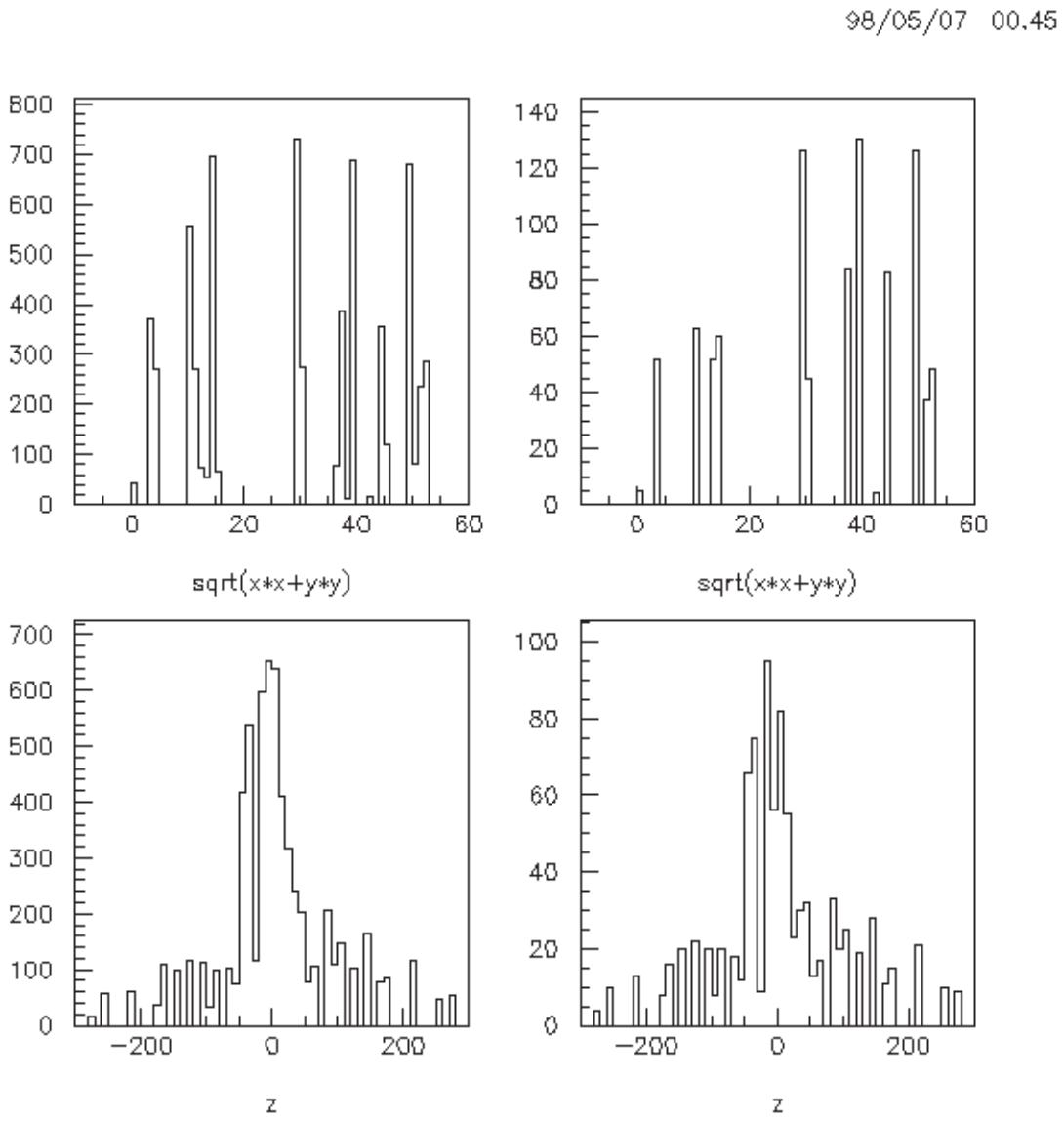
CURRENT STATUS

- see *VRML*



CURRENT STATUS

- see *HBookTuple*



ARVE ARAVIS

- Arve->Aravis
- Aravis->Arve
- Any volunteer ?

ATLANTIS

- Atlantis itself
  - GUI
  - Manual
  - Tutorial (spoken, written)
  - More Unix-like
- Integration
  - File format
  - Bridge

WIRED

- Integration
- XML
- ...

JAVA

- Graphics control is now ported
- Histograms are interfaced
- How to proceede

GUI

- Object Browser

SUBPACKAGES

- ASP Evaluation Procedure
- Known Hot Candidates (maintainer should be defined)
  - Atlantis
  - Wired
  - IE
  - Root
  - VRWave
  - JAS
  - Aravis
  - Geant4
- New Candidates
  - Make exploatory study

DATA SOURCES

- From Trigger Domain
- From Event Domain
  - Concentrate on this
  - Collaboration with other Domains