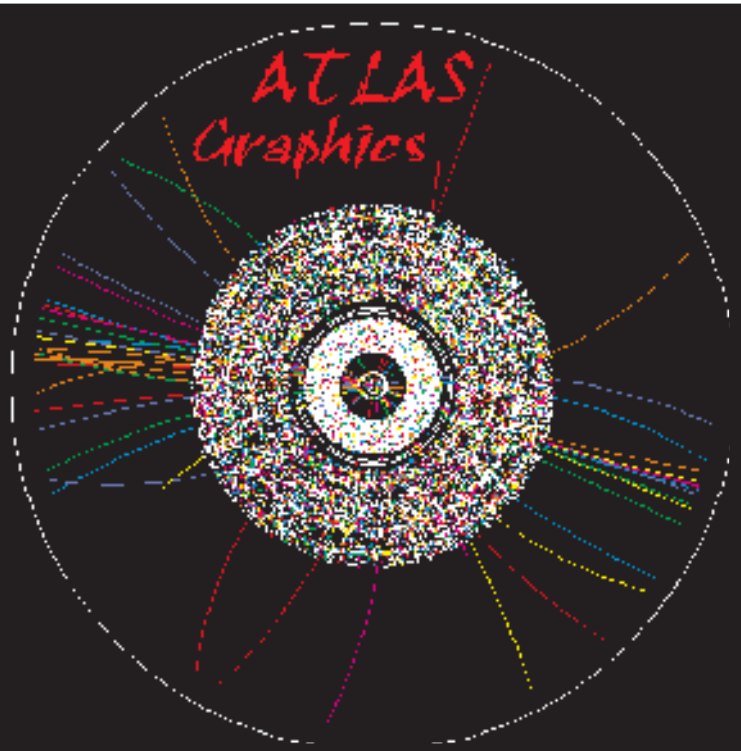
Graphics



- Graphics Evaluation
- Graphics from Athena
- Scenes
- HEP Studio & FreeHEP
- Representations
- JACO
- Java Notes

Atlas SW Week, Nov-Dec'00, CERN

Graphics Evaluation

- several Users have been asked to try various Graphics components and give feedback
- Introductory presentation in Physics Meeting on We
- more details on Graphics Meeting on Th
- report next Software Week (eof'Feb)

Graphics From Athena

- **offline/graphics/GraphicsFromEvent**
- Athena Algorithm
- same functionality as Paso graphics
- easier to use

```
# Setup Athena (see AT Documentation)
# Setup Qt (needed by Aravis)
. #SRT_DIST/current/installed/share/scripts/Qt_env.sh
# Get Graphics Job Options
cp $SRT_DIST/current/installed/share/GraphicsFromEvent/jobOptions.txt .
# Edit jobOptions.txt
# Run Athena (see AT Documentation)
```

Graphics From Athena

jobOptions.txt (number = Level Of Details = 0-9)

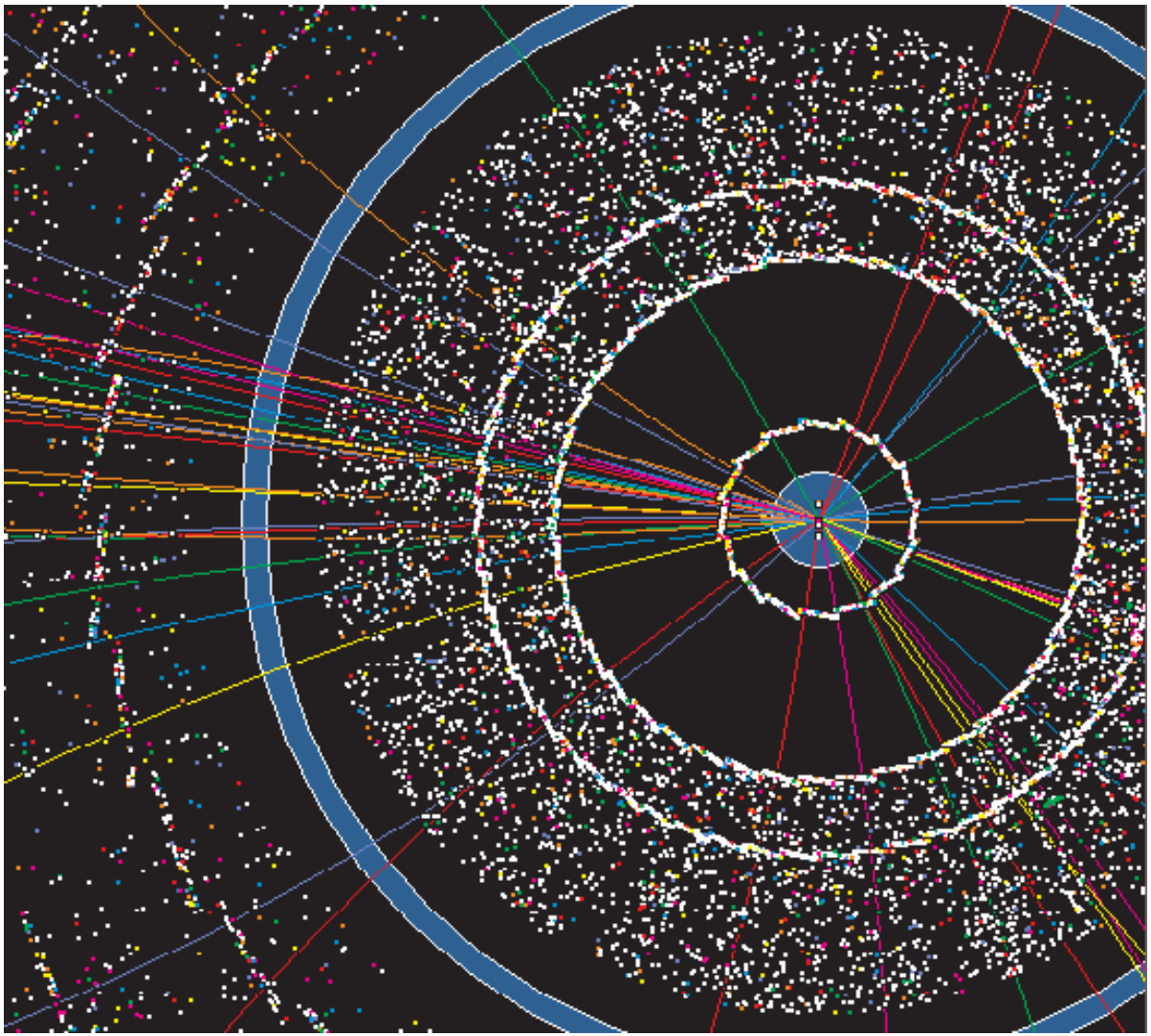
```
Graphics.TXT                = 9; # writes index.txt file
Graphics.XML                = 9; # writes index.xml file
Graphics.Aravis             = 9; # opens Aravis window
Graphics.XMLperEvent        = 9; # writes Event_*.xml files
Graphics.PixelDetectors     = 9;
Graphics.PixelDigits        = 9;
Graphics.SCTDetectors       = 9;
Graphics.SCTDigits          = 9;
Graphics.StripClusters      = 9;
Graphics.PixelSpacePoints   = 9;
Graphics.SCTSpacePoints     = 9;
Graphics.TRTDetectors       = 9;
Graphics.TRTDigits          = 9;
Graphics.MDTDetectors       = 9;
Graphics.MDTDigits          = 9;
Graphics.RPCDetectors       = 9;
Graphics.RPCDigits          = 9;
Graphics.TGCDetectors       = 9;
Graphics.TGCDigits          = 9;
Graphics.CaloRegions        = 9;
Graphics.CaloDigits         = 9;
Graphics.TruthTracks        = 9;
```

for Atlantis, Wired, GraXML, Persint

Atlantis

- first Java version this year
- MU requested

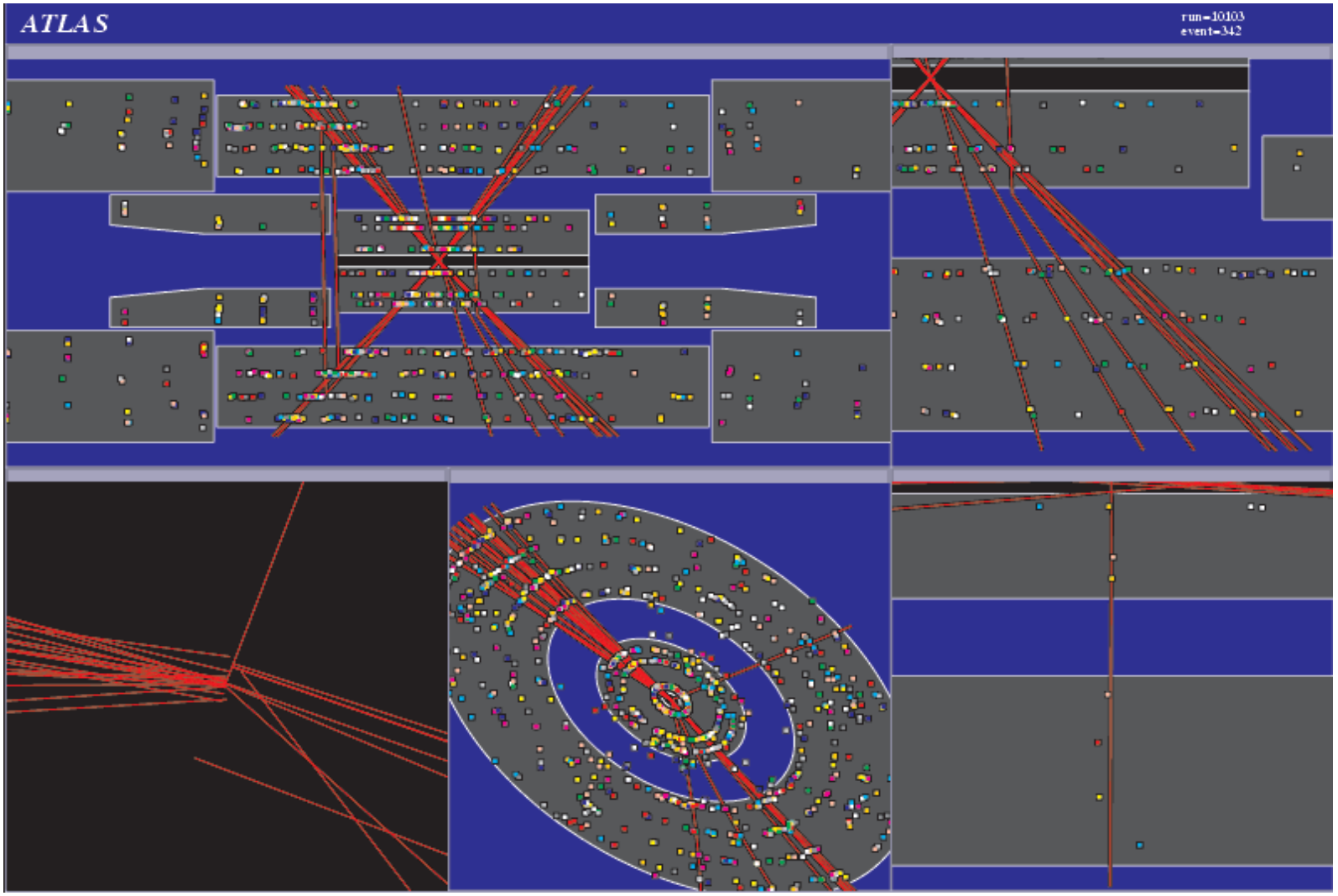
H.Drevermann,
G.Taylor,
P.Klok,
B.S.Nilsson



Wired

M.Donszelmann, C.Loomis

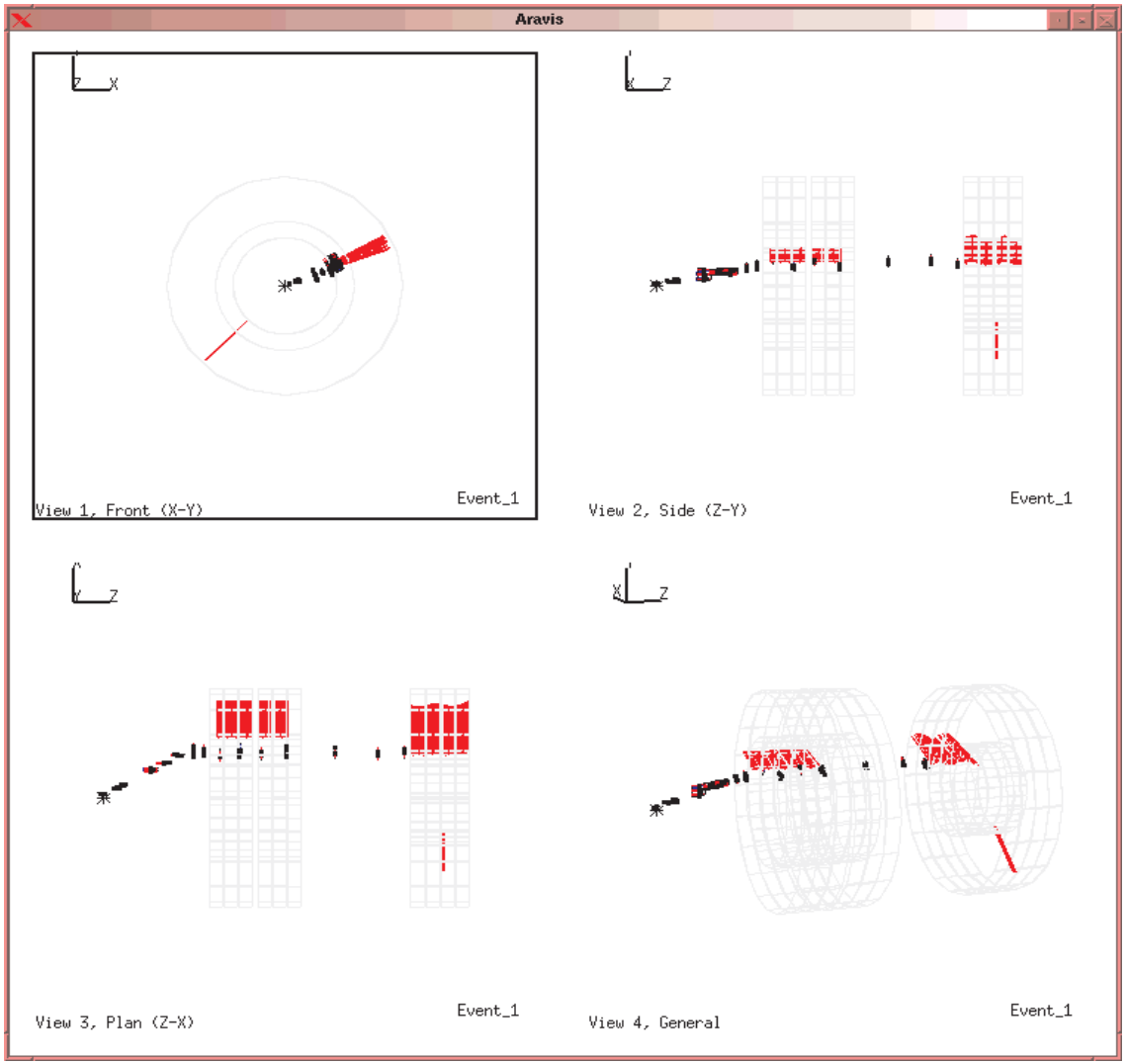
- new, much improved version within 1 month
- part of HEP Studio (see later)



Aravis

T.Burnet,
R.Candlin,
D.Candlin

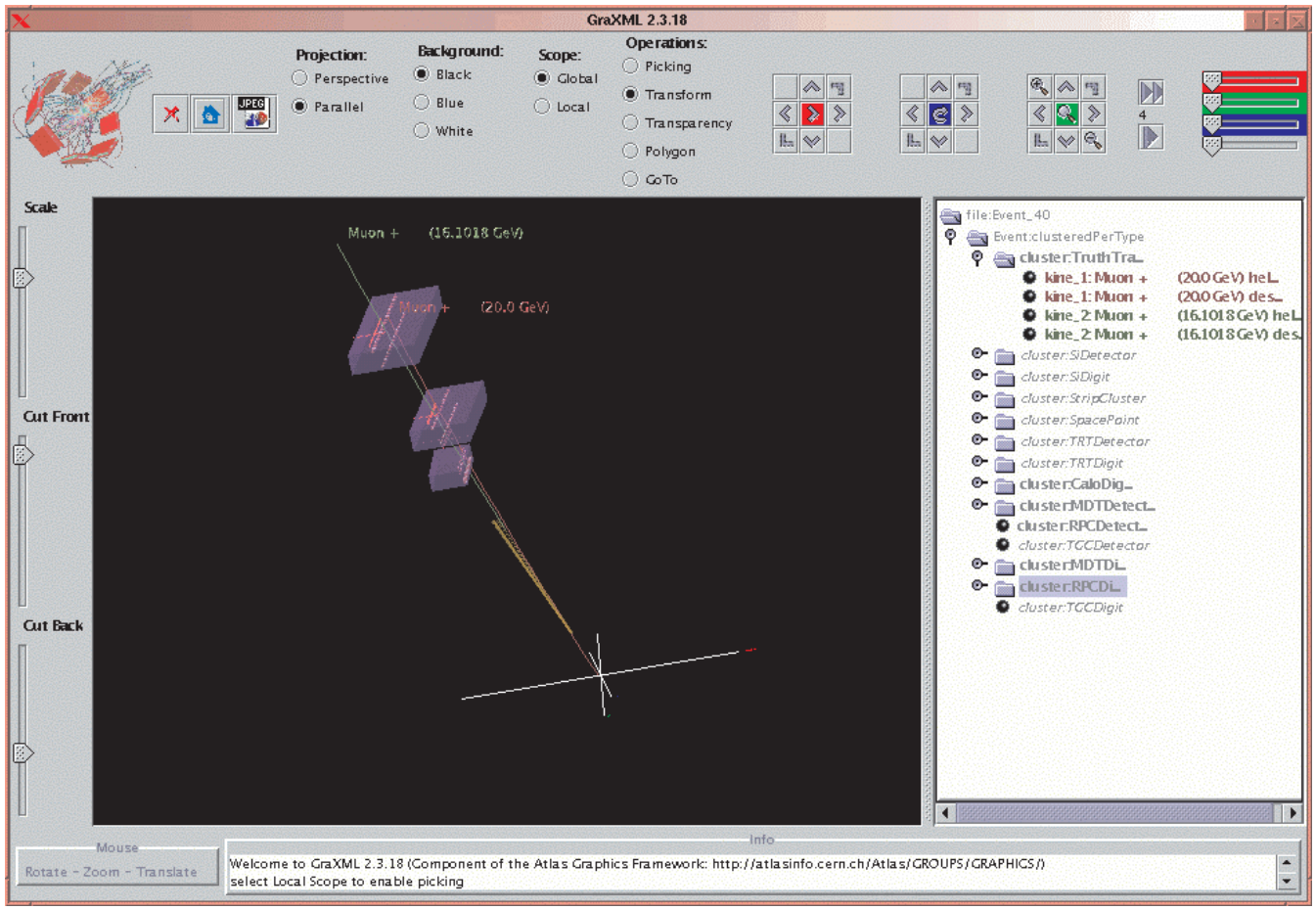
- based on Qt
- other Reps added



GraXML

J Hrivnac

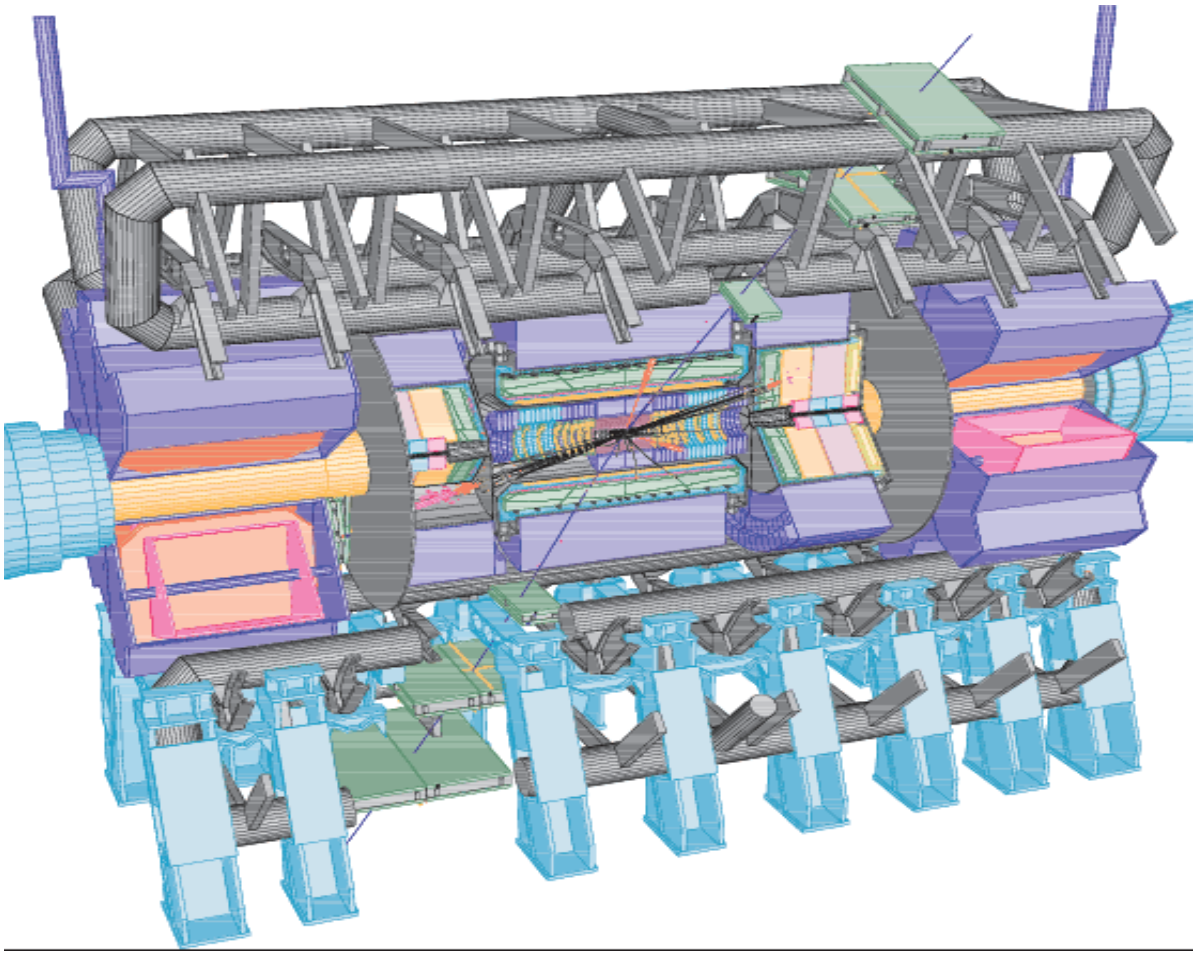
- reads:
  - AGDD: Atlas Detector Description
  - AliDD: Extended Atlas Detector Description
  - atlasevent: Atlas Event
- Shape3D objects for GeantH solids moved to FreeHEP (=> HEP Studio), other components will follow
- Design improved



Persint

Samusog

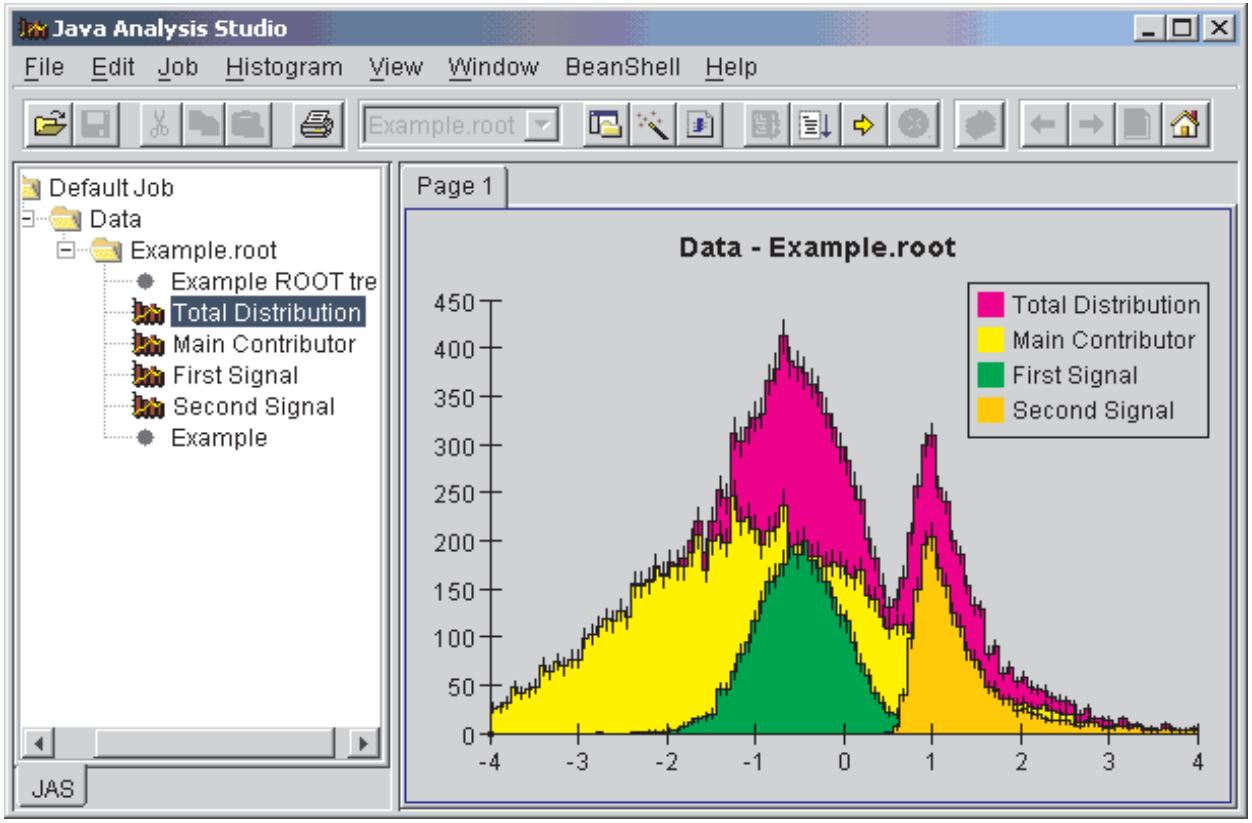
- will become available as External package
- does (will) support standard XML files
- unclear plans



JAS

T.Johnson

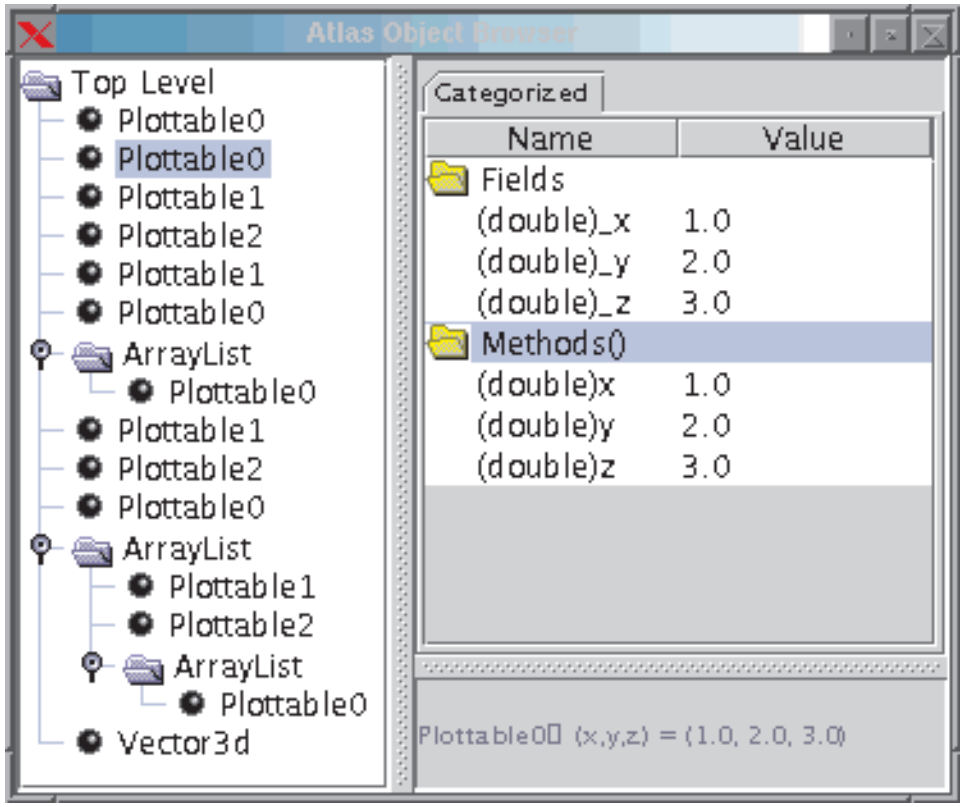
- reads directly (among others):
  - HBook/Paw files
  - HTL from Objectivity
  - Root files
- connects to C++ using AIDA + JACO
- part of HEP Studio (see later)



Object Browser

L.Cunha,
L.Tuura,
J.Hrivnac

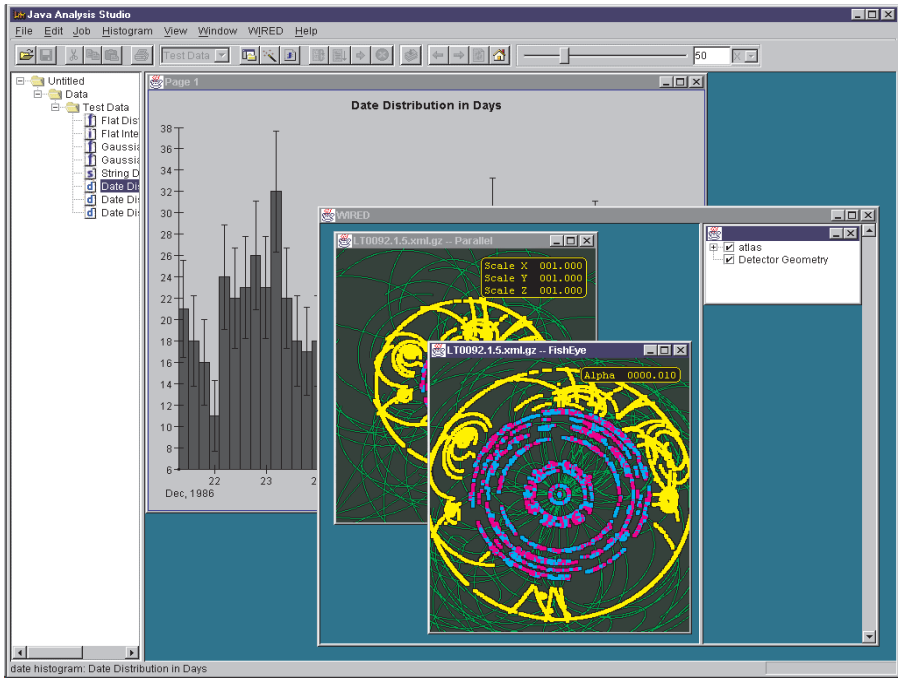
- moved to FreeHEP (=> HEP Studio)
- improved:
  - can access public, package, protected, private members
  - can access fields and methods
  - can access inherited members
- future extensions:
  - Context Sensitive Actions
  - Drag&Drop into HEP Studio
  - JACO interface to C++



HEP Studio

T.Johnson,
M.Donszelmann

- Graphics & Analysis Java Framework build from independent Modules
- first version within a month
- based on FreeHEP (Java "cernlib")
- contains:
  - JAS
  - Wired
- will contain:
  - GraXML
  - Object Browser
  - Atlantis
- uses HEP standards:
  - AIDA for Histogramming (=> interoperability with Lizard, Open Scientist,...)
  - HEPRep for Event/Detector Display



Representations

- Now:
  - ID Digits & Detectors
  - MU Digits & Detectos
  - TruthTracks
- Soon:
  - ErrorPoints from Moore
  - Tracks from Moore
- Jan'01:
  - Hits
  - Tracks from Hits
- Later:
  - Calorimeters
  - Reconstructed Objects

Java Access to C++ Objects

T.Johnson, M.Donszelmann

- parses all C++ headers
- work going on on interface to Event (i.e. generation of interface code)
- can be generalised later (in collaboration with AT)
- gives many features "for free"
  - Introspection
  - Data Dictionary
  - Java Scripting
- part of FreeHEP

Java Notes

The Java programming language
has had a huge impact in the world outside HEP
and is now making itself noticed within HEP.

Here we discuss why we should consider it,
and what the impact might be.

- Java in Atlas - coding standards
  (ATL-SOFT-2000-002)
- On the importance of Java to Atlas
  (ATL-COM-SOFT-2000-004)

Coding Standards

1. Standards
    - 1.1 Existing standards
    - 1.2 Naming Conventions and the Package Structure
    - 1.3 Overriding methods of Object
    - 1.4 Import statements and qualification of names
    - 1.5 Clarity of code
    - 1.6 Resources
    - 1.7 Avoid surprises
2. Programming advice
3. Build Environment
    - 3.1 Handling by ART
    - 3.2 Building and testing
    - 3.3 JAR files
    - 3.4 Documentation
    - 3.5 Tools
4. Guidelines for Java - C++ co-existence
    - 4.1 Interfacing
    - 4.2 Impact on C++

QC Group ?

Java Importance

1. Introduction
2. Benefits of Java
    - 2.1 Automatic memory management
    - 2.2 No "*", "->" or "$" operators to worry about just "."
    - 2.3 Clean definition of interfaces
    - 2.4 No ".h" files
    - 2.5 World-wide naming scheme
    - 2.6 No preprocessor
    - 2.7 Byte code can execute anywhere with a JVM
    - 2.8 Fast development
    - 2.9 Fast execution
    - 2.10 Thread management is part of the language
    - 2.11 Exact reproducibility of results
    - 2.12 "Reflection" facilities
    - 2.13 "Serialization"
    - 2.14 Dynamic optimisation
    - 2.15 Dynamic loading
    - 2.16 Javadoc generated documentation
    - 2.17 Excellent tool support
    - 2.18 Easy to configure for compilation
3. Non-problems
    - 3.1 Performance for numerical operations
    - 3.2 3D graphics performance
    - 3.3 Numerics
    - 3.4 Memory Management

Java Importance

4. Possible problems
- 4.1 Handling of too many objects
- 4.2 Running in the Event Filter and Level 2 Trigger

5. Unresolved problems
- 5.1 Access to databases
- 5.2 Java <-> C++ collaboration
- 5.3 Non-existence of immutable objects

6. Other features

7. Place for Java in HEP
- 7.1 Framework
- 7.2 Reconstruction
- 7.3 Simulation
- 7.4 Database
- 7.5 Analysis and Graphics
- 7.6 User Interface
- 7.7 Grid

8. Recommendations
- 8.1 Architecture
- 8.2 ATLAS Support
- 8.3 CERN Support

9. Acknowledgements

Java Importance

In general, we can say that
there are no serious obstacles to introduce Java
as the main implementation language for the HEP software development.
All frequently claimed Java problems
are either not problems at all, or they are already solved.
At the same time many of the Java advantages
will have clear positive impact on the functionality, performance and maintainability
of our software.
Also the number of existing standard and mature Java applications
and class libraries will make the future software development for HEP
much faster, safer and easier.

In a mixed Java-C++ environment,
one can't fully exploit all the Java advantages.
So it would be preferable to replace C++ by Java to benefit from OO
without fighting the implementation language.
Common experience shows, that once programmers start using Java,
they won't go back.
It appears that the time is right to start the move towards Java and
we should not delay further.

Java Importance

Java is emerging as
the best general purpose computing language
available at the moment
and as such
we expect it to become
the 'default' language for most tasks in HEP.