

JAS/Wired + Geant 4

Tony Johnson

July 2001

Contents

- What is JAS?
- What is WIRED?
 - Future Directions
- JAS+AIDA+GAG+Wired + Geant 4= ?
- Making it easy to use
- Future Directions
- Demo?

Introduction to JAS

- Pure Java Analysis Environment
 - Data Format Independent
 - Modular/Extensible via Plugins/Data Interface Modules
 - Rich Easy to use GUI
 - Built in editor/compiler for writing analysis code
 - Local and Client-Server Operation
 - Originally targeted at offline analysis – but also used extensively for online monitoring
 - Written entirely in Java

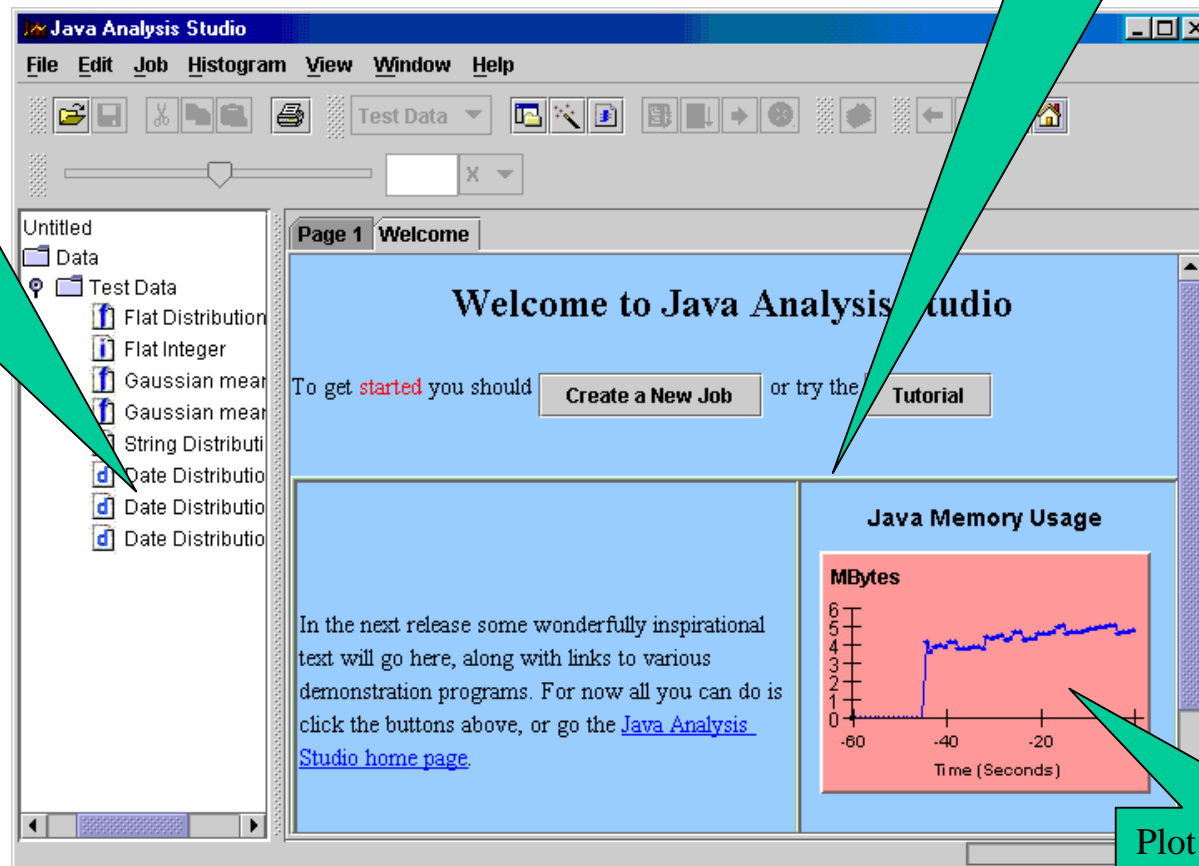
JAS GUI

Tree provides access to analysis objects:

- Histograms
- Plots
- Data Sets
- Analysis Routines

In principle any object. Each item has popup menus and double click action.

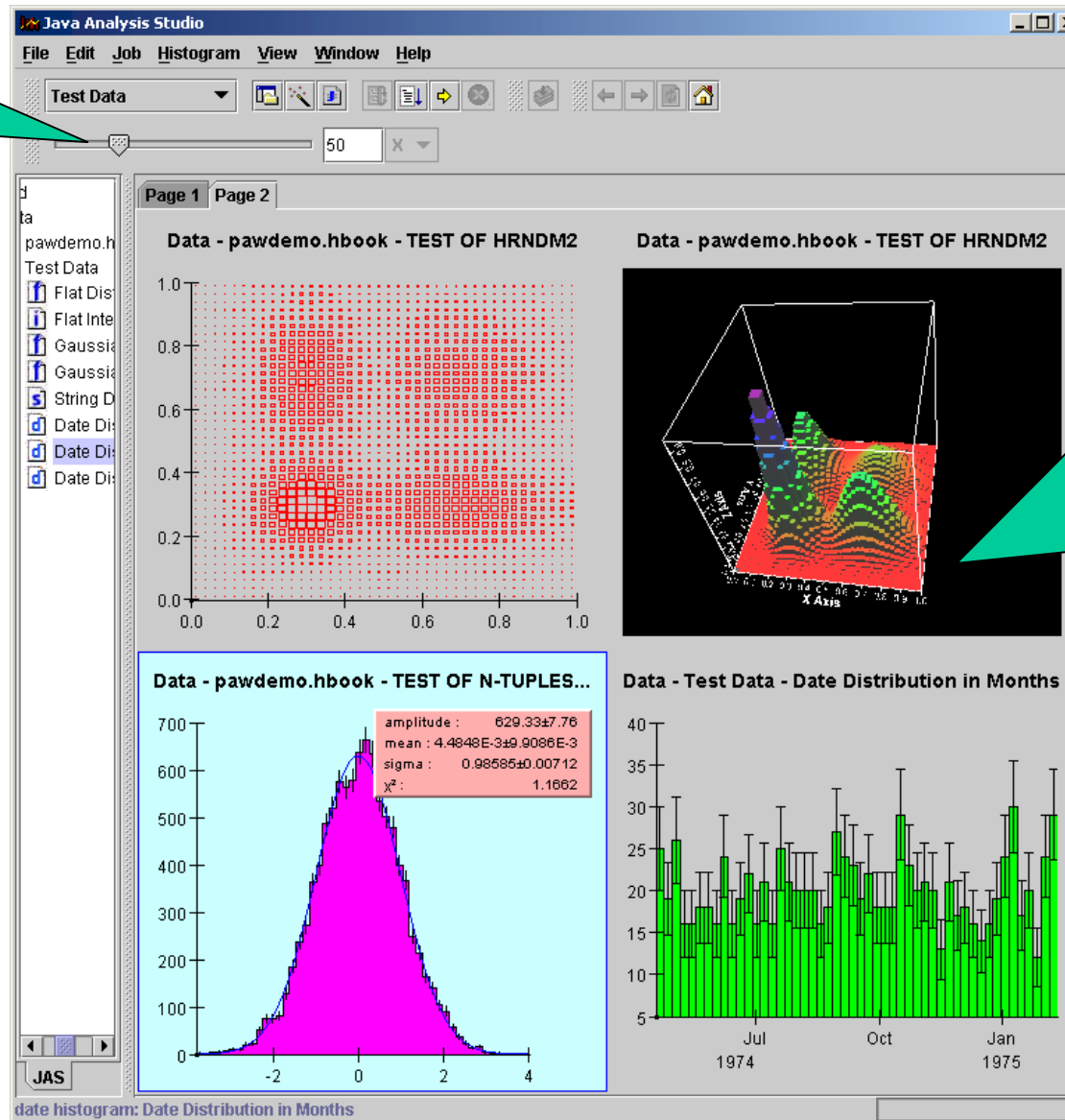
Built in HTML viewer with embeddable “objects” (buttons, plots, etc).



Plot Widget, shows data in real time, optimized for fast refresh performance

JAS Plotter

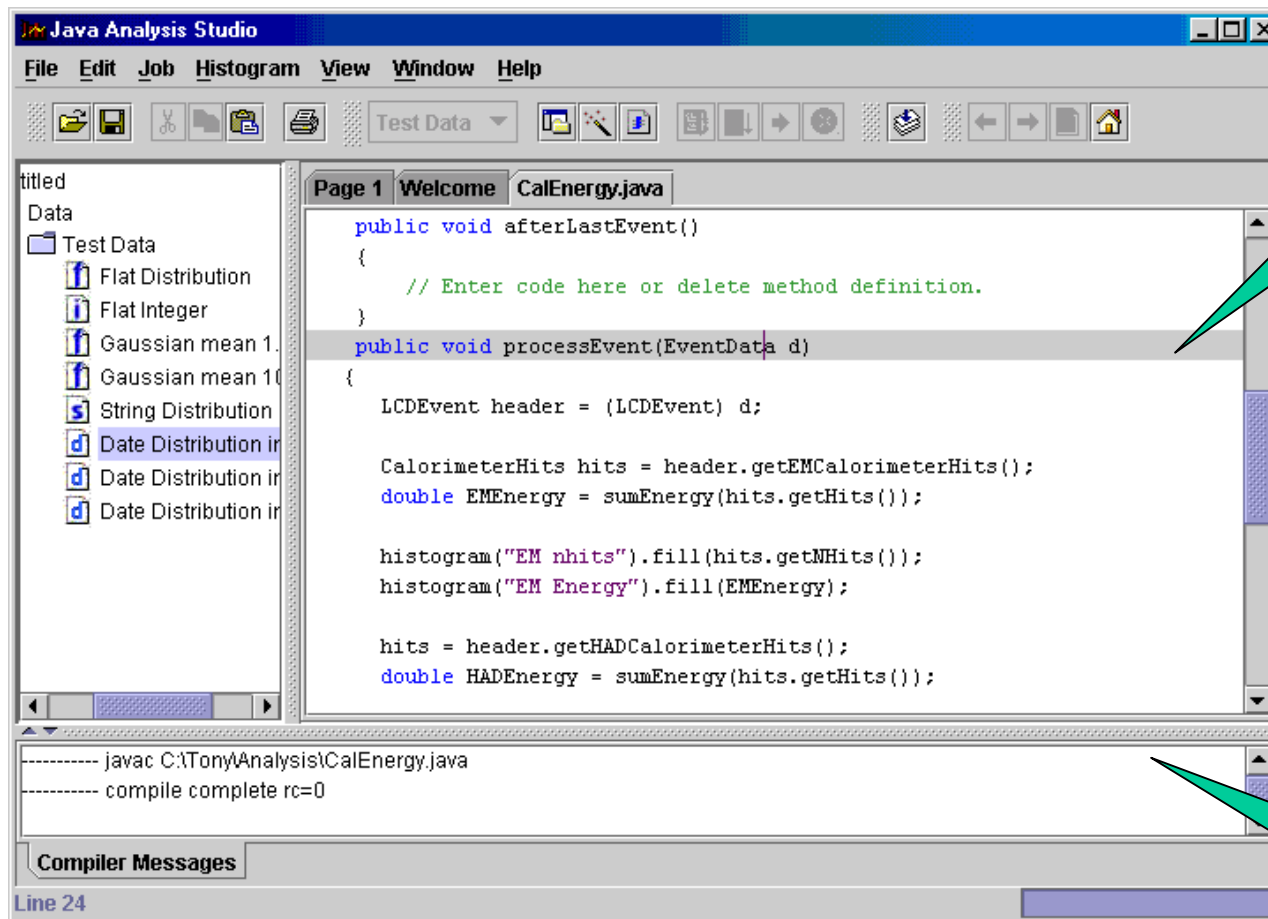
Rebin slider can be used to dynamically change # of bins.



Pages can display histograms. User can control layout, add remove plots, etc.

Plots are highly interactive, can be manipulated by the user by dragging on the axis, or bounding box. Labels (title, legend, axis labels) can be updated by clicking and typing.

JAS Editor/Compiler



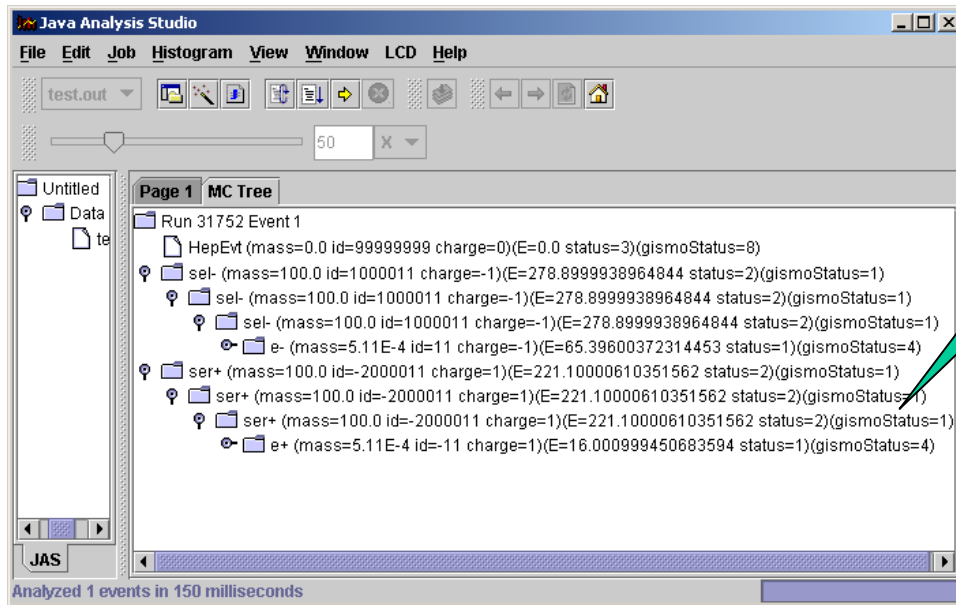
Built in code editor
with syntax
highlighting (based
on open-source
JEdit)

Built-in Java compiler.
Can dynamically load
(and unload) analysis
code.

Extensible via Plugins

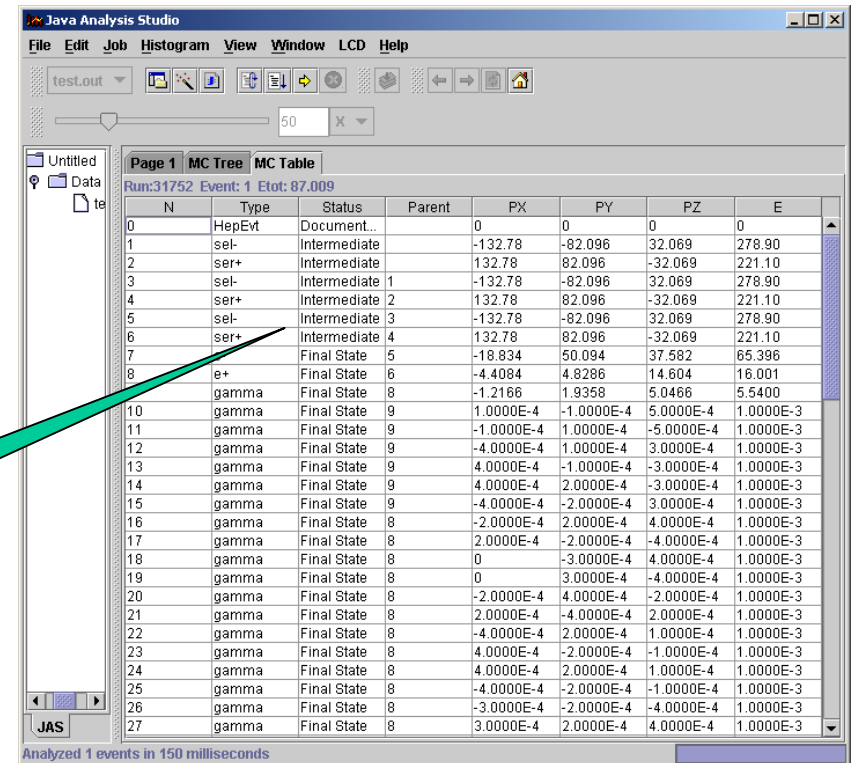
- Plugins can:
 - Define experiment specific utilities (event display, analysis utilities, specialized tables).
 - Define data interfaces to handle new types of data.
 - Define new plotting routines (e.g. to display special display).
 - Add menus, create control areas, consoles, and output pages.
 - Plugins will be more flexible in JAS 3.0 (see discussion of FreeHEP application framework, later).

Examples of Plugins



The screenshot shows the 'MC Tree' view in Java Analysis Studio. It displays a hierarchical tree structure for 'Run 31752 Event 1'. The tree starts with 'HepEvt' and branches into 'sel-' and 'ser+' particles, which further branch into 'e-' and 'e+' particles. Each node contains detailed parameters like mass, id, charge, energy, and status.

Particle decay tree



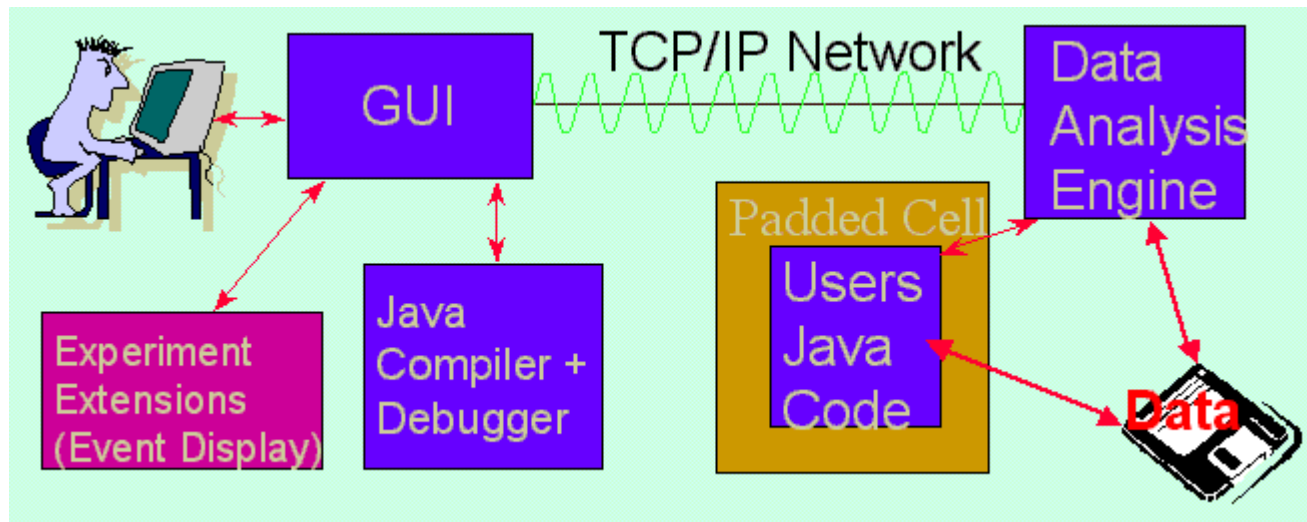
The screenshot shows the 'MC Table' view in Java Analysis Studio. It displays a table with columns for N, Type, Status, Parent, PX, PY, PZ, and E. The table lists 27 rows of particle data, including 'HepEvt', 'sel-', 'ser+', 'e-', 'e+', and 'gamma' particles, along with their respective parameters.

N	Type	Status	Parent	PX	PY	PZ	E
0	HepEvt	Document...	0	0	0	0	0
1	sel-	Intermediate		-132.78	-82.096	32.069	278.90
2	ser+	Intermediate		132.78	82.096	-32.069	221.10
3	sel-	Intermediate	1	-132.78	-82.096	32.069	278.90
4	ser+	Intermediate	2	132.78	82.096	-32.069	221.10
5	sel-	Intermediate	3	-132.78	-82.096	32.069	278.90
6	ser+	Intermediate	4	132.78	82.096	-32.069	221.10
7		Final State	5	-18.834	50.094	37.582	65.396
8	e+	Final State	6	-4.4084	4.8286	14.604	16.001
9	gamma	Final State	8	-1.2166	1.9358	5.0466	5.5400
10	gamma	Final State	9	1.0000E-4	-1.0000E-4	5.0000E-4	1.0000E-3
11	gamma	Final State	9	-1.0000E-4	1.0000E-4	-5.0000E-4	1.0000E-3
12	gamma	Final State	9	-4.0000E-4	1.0000E-4	3.0000E-4	1.0000E-3
13	gamma	Final State	9	4.0000E-4	-1.0000E-4	-3.0000E-4	1.0000E-3
14	gamma	Final State	9	4.0000E-4	2.0000E-4	-3.0000E-4	1.0000E-3
15	gamma	Final State	9	-4.0000E-4	-2.0000E-4	3.0000E-4	1.0000E-3
16	gamma	Final State	8	-2.0000E-4	2.0000E-4	4.0000E-4	1.0000E-3
17	gamma	Final State	8	2.0000E-4	-2.0000E-4	-4.0000E-4	1.0000E-3
18	gamma	Final State	8	0	-3.0000E-4	4.0000E-4	1.0000E-3
19	gamma	Final State	8	0	3.0000E-4	-4.0000E-4	1.0000E-3
20	gamma	Final State	8	-2.0000E-4	4.0000E-4	-2.0000E-4	1.0000E-3
21	gamma	Final State	8	2.0000E-4	-4.0000E-4	2.0000E-4	1.0000E-3
22	gamma	Final State	8	-4.0000E-4	2.0000E-4	1.0000E-4	1.0000E-3
23	gamma	Final State	8	4.0000E-4	-2.0000E-4	-1.0000E-4	1.0000E-3
24	gamma	Final State	8	4.0000E-4	2.0000E-4	1.0000E-4	1.0000E-3
25	gamma	Final State	8	-4.0000E-4	-2.0000E-4	-1.0000E-4	1.0000E-3
26	gamma	Final State	8	-3.0000E-4	-2.0000E-4	-4.0000E-4	1.0000E-3
27	gamma	Final State	8	3.0000E-4	2.0000E-4	4.0000E-4	1.0000E-3

Particle tabulation

Remote Data Access

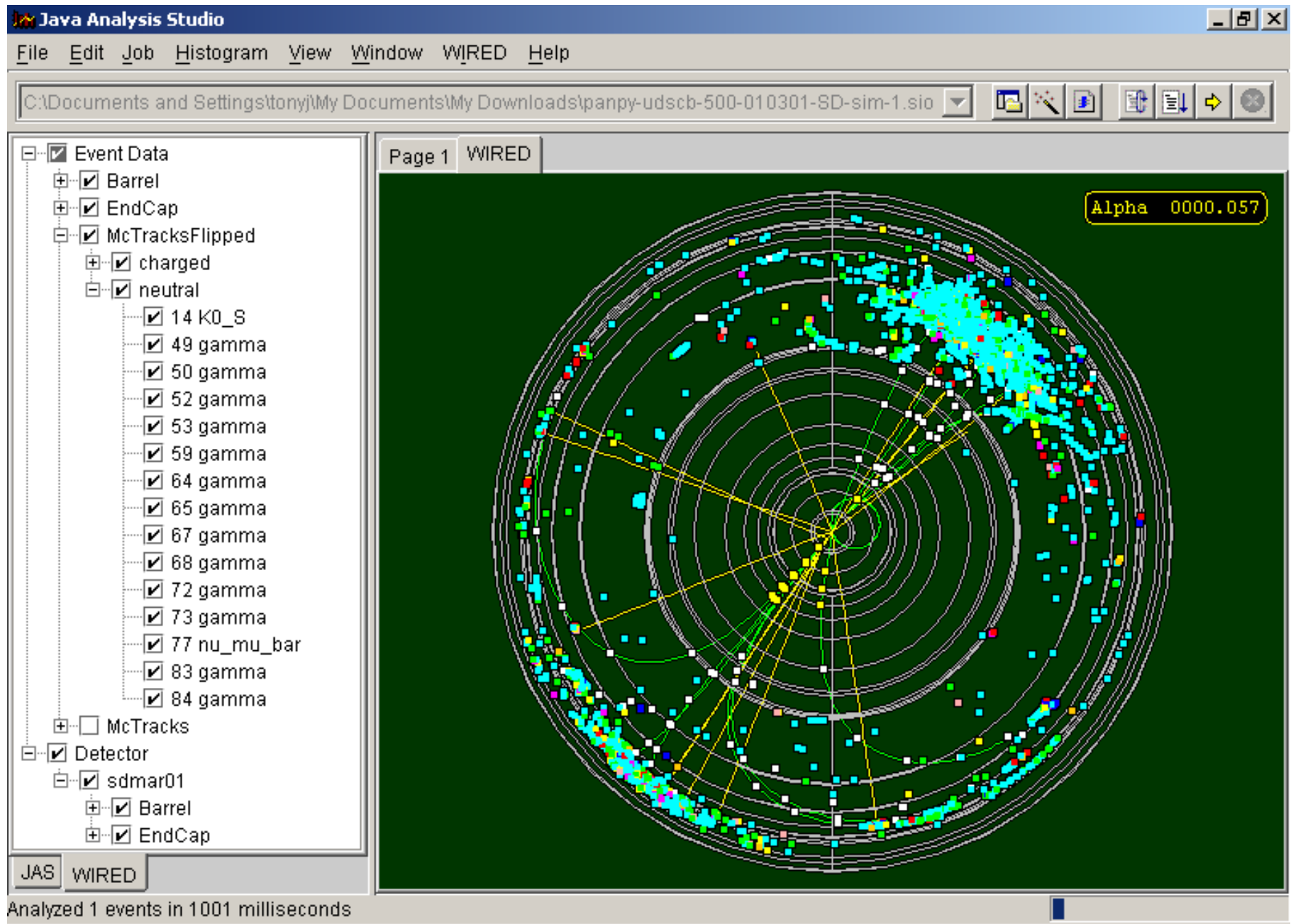
- Rather than transporting peta-bytes of data to the physicist
 - Transport the physics analysis code to the data
 - Transparently - so that it feels just like local data access
 - Just ship histogram contents back to the physicists desktop (on demand)
- Allows remote analysis with modest network bandwidth
- Allows user to “feel” as if using local machine even when accessing remote data.



WIRED

- Experiment Independent Event Display
 - Typically requires very little customization for new detectors/experiments
 - Client can be completely experiment independent
 - Large amount of functionality built in
 - Greatly reduces time needed to create event display
 - Works in client-server mode
 - Transfers HepReps from client to server
 - Via XML, RMI, Corba, ...
 - Written entirely in Java

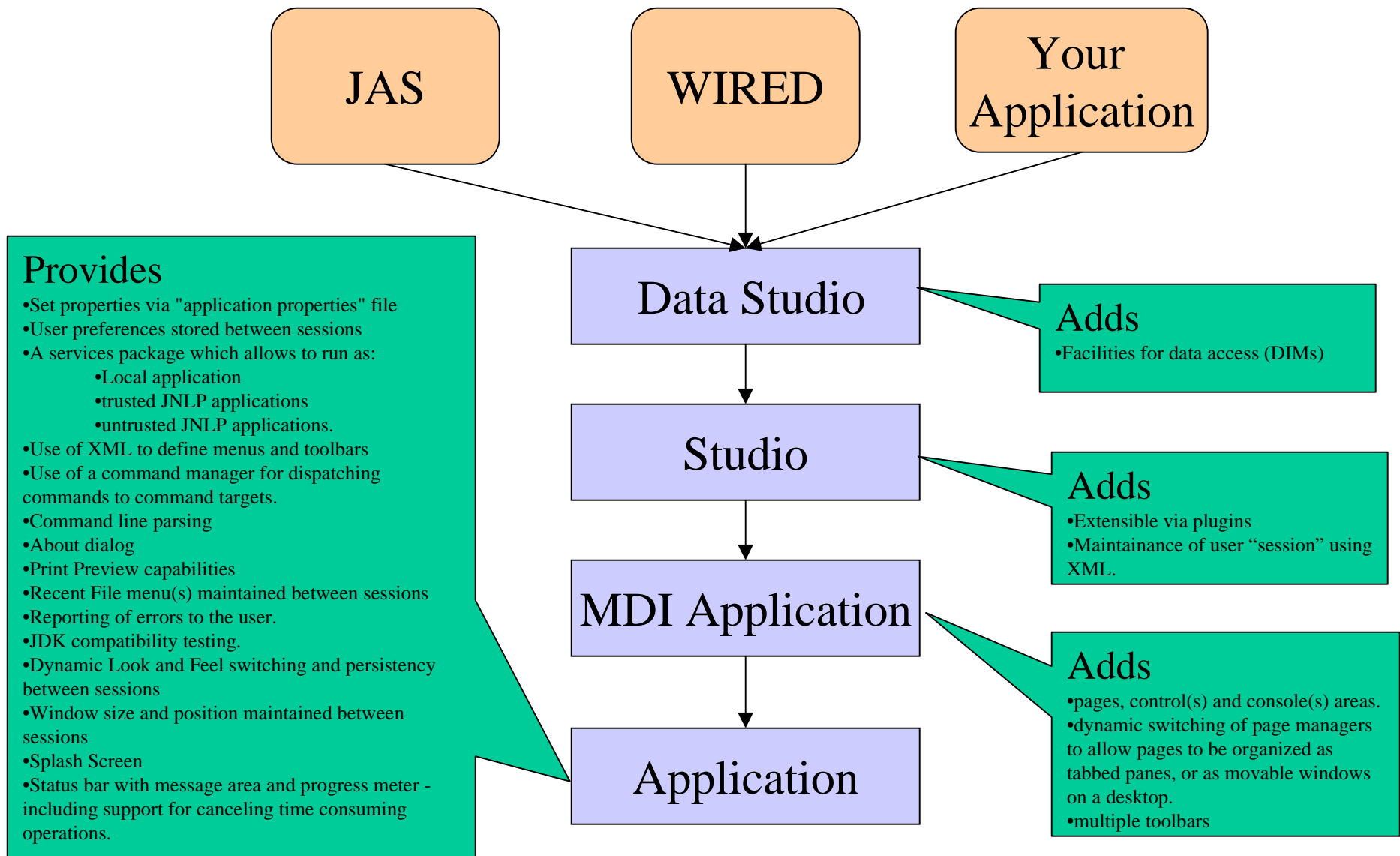
JAS+Wired



Future of JAS+Wired

- Java FreeHEP library repository for common utilities and common framework
 - Open source library includes:
 - Graphics utilities
 - Plotter, EPS utilities
 - Physics specific classes
 - Four vectors, jet finders, particle properties, ...
 - IO utilities for Root, StdHEP...
 - Encourage extensive (Java) code reuse
 - Six code developers + many contributions from students etc.

FreeHEP Application Framework

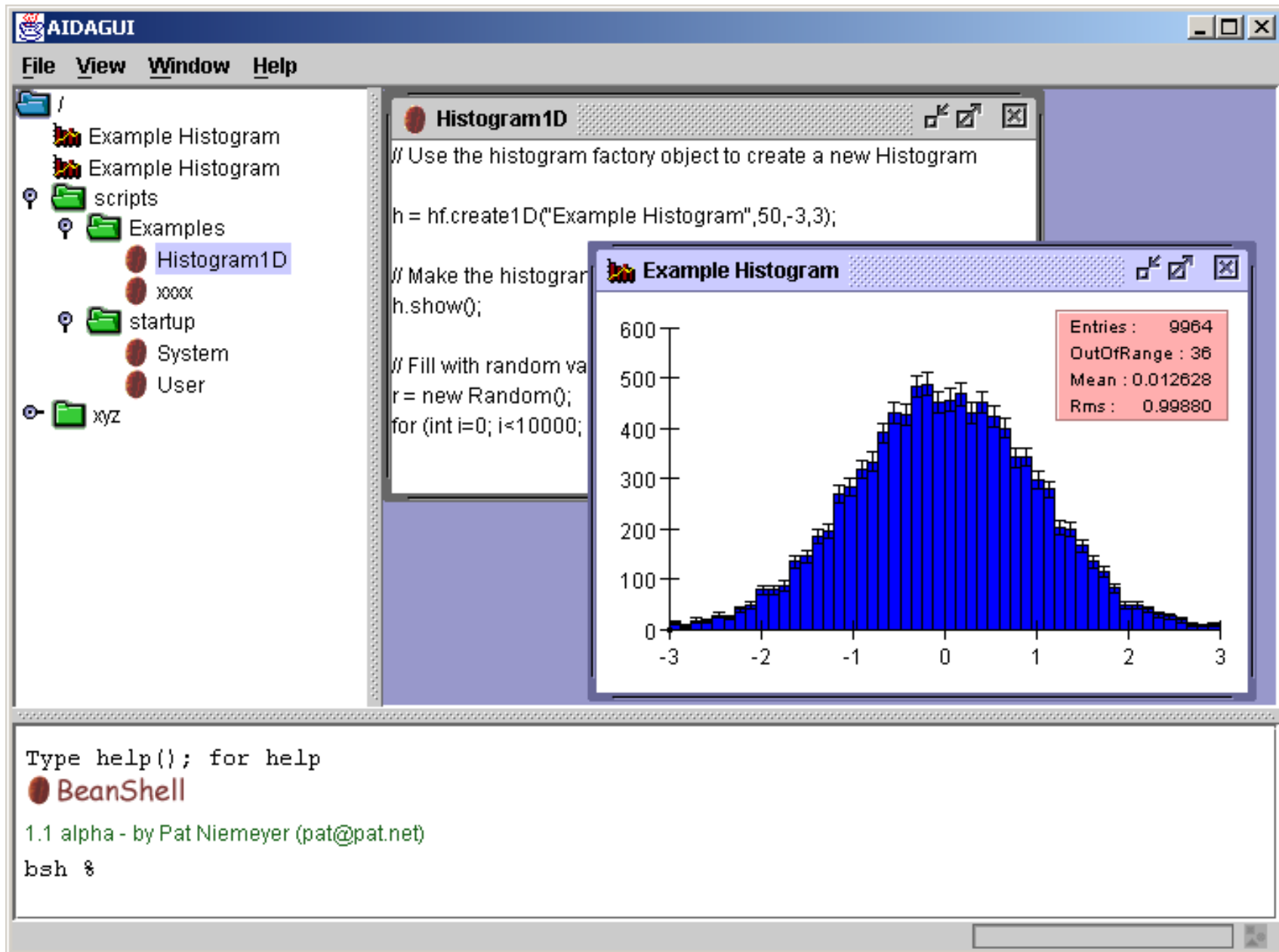


JAS 3

Studio will extend plugin concept so application becomes bare framework, with:

- Data Access Plugin (c.f. JAS DIM's)
 - Histogramming (**AIDA**) Plugin
 - Plotting Plugin
 - Editing Plugin
 - Compiler Plugin
 - **Scripting** Plugin
 - WIRED (Event Display) Plugin
 - Etc. etc. etc.
- Plugins can communicate either explicitly, or (better) though
 - “service interfaces”
 - “service registries”

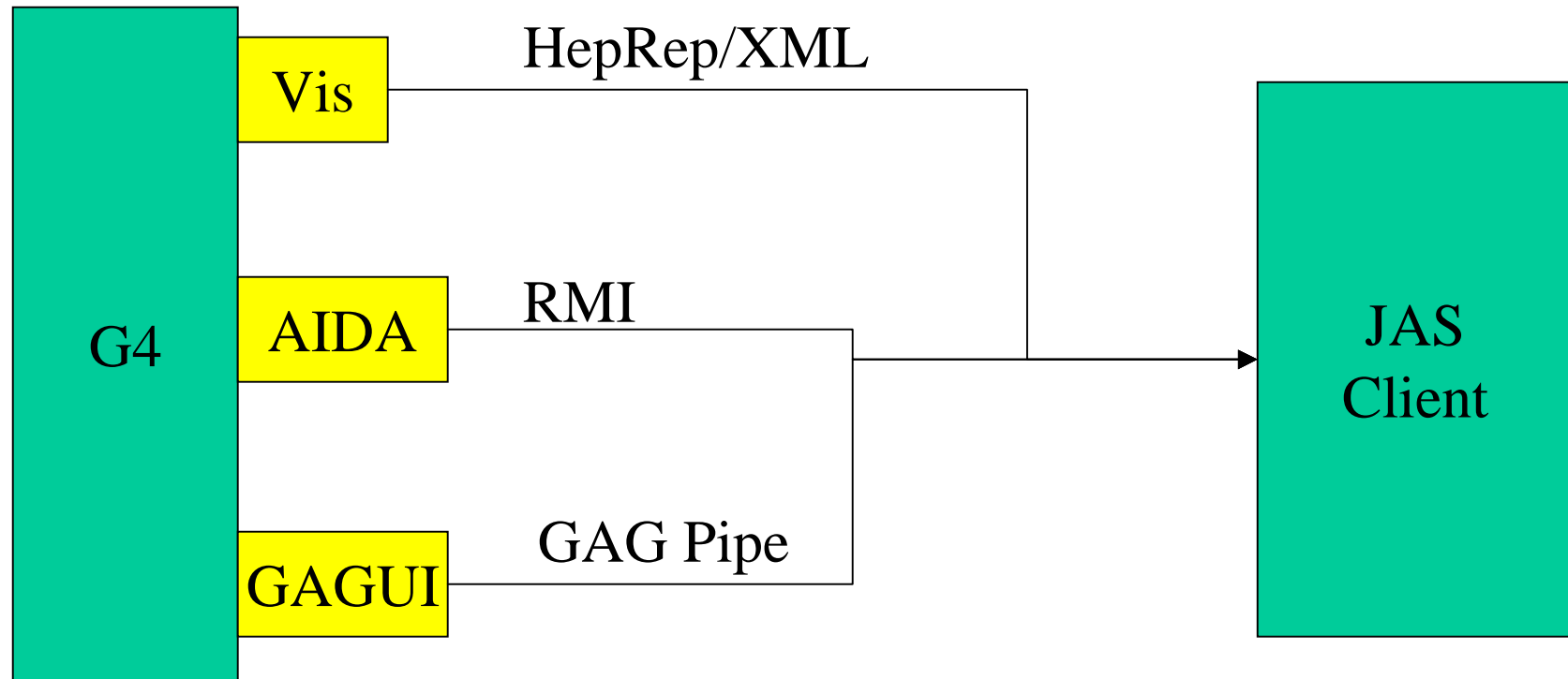
JAS 3 Prototype



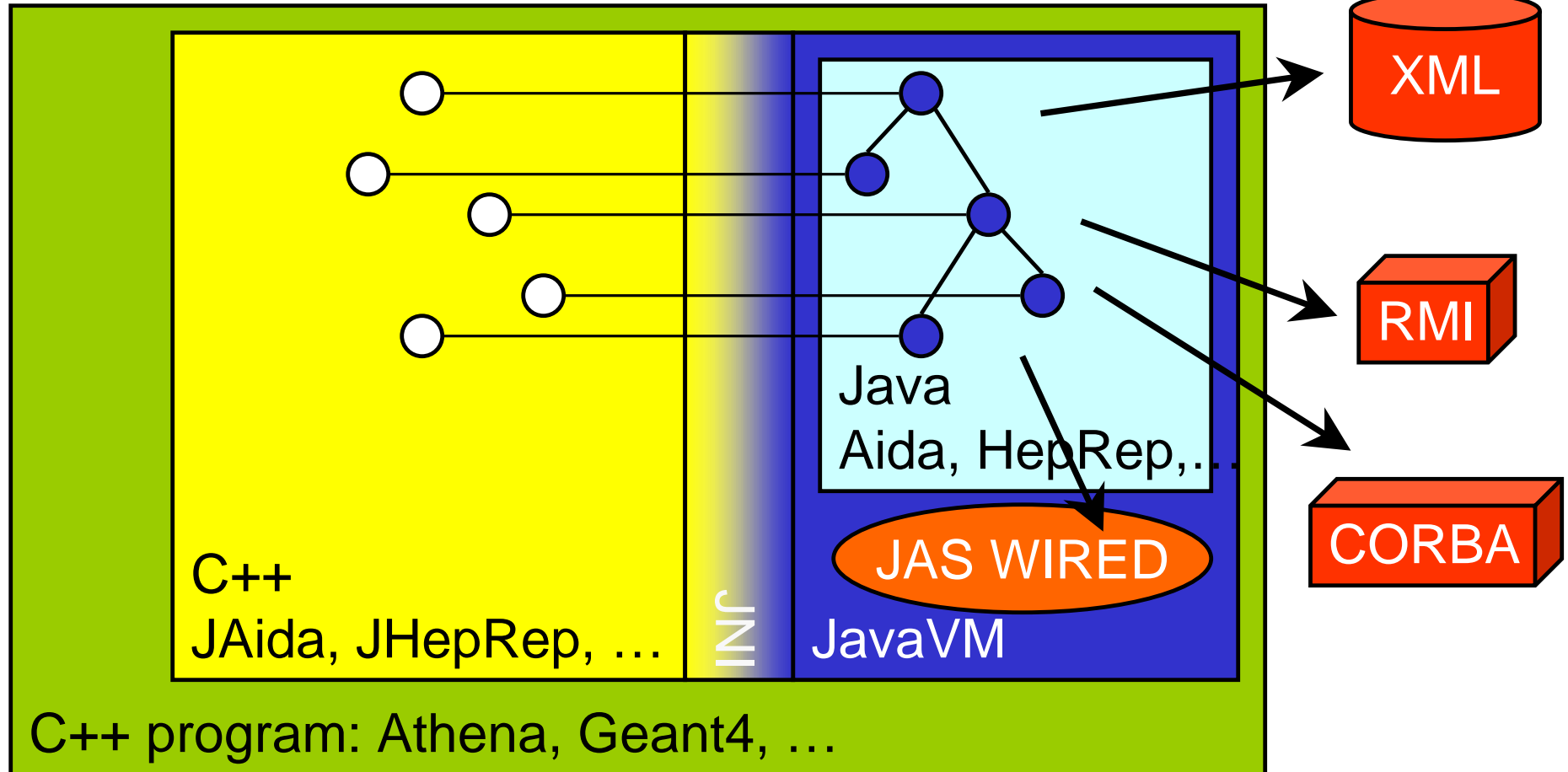
JAS+Wired+Geant4

- For Geant 4 we have combined
 - JAS
 - Wired + HepRep
 - AIDA – Abstract Interface for Data Analysis
 - GAG – Geant4 Adaptive GUI
 - Geant4
 - JAWGGH?

JAS+WIRED+Geant4

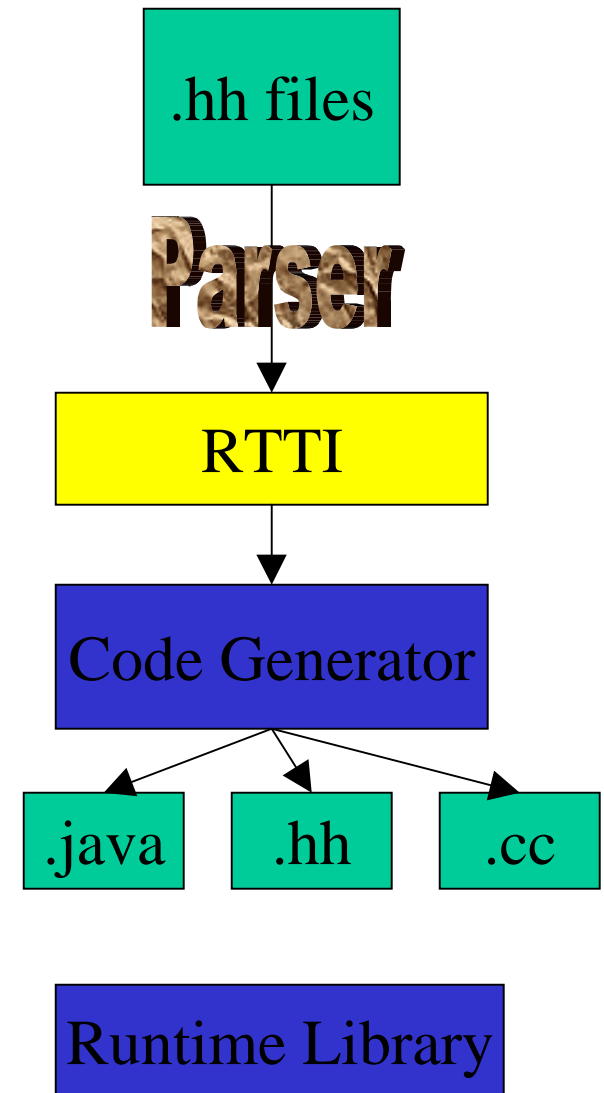


C++ -> Java

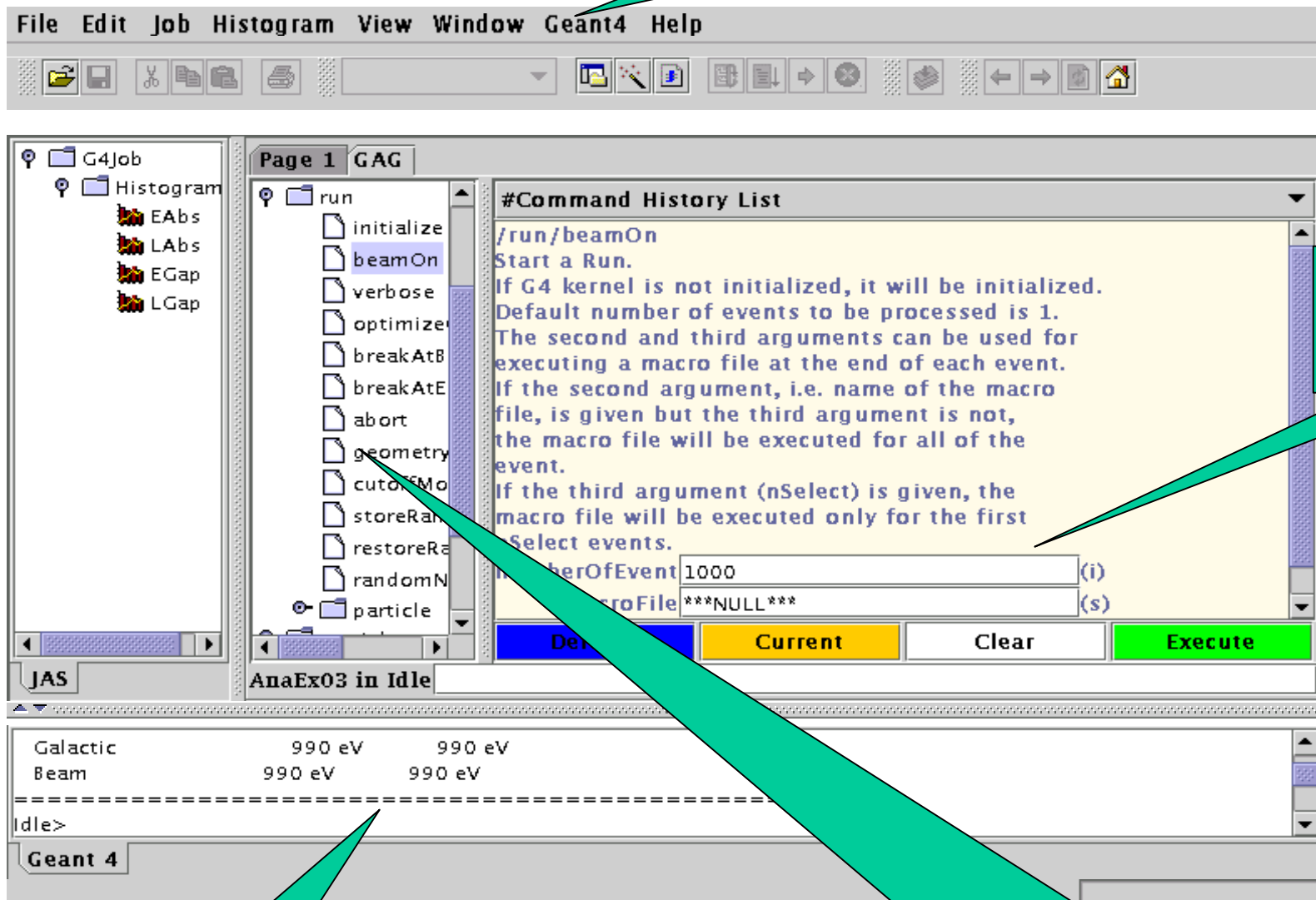


JACO (Java Access to C++ Objects)

- Interoperability of Java and C++ is certainly possible
 - But tedious
 - Code must be written for each method
 - Need to deal with starting JavaVM
 - Need to deal with object lifetimes
- JACO – Aims to solve this
 - Automatic generation of glue code
 - Small library to deal with details
- Aimed at Event Display and Event Analysis



Geant 4 menu. Used to start G4 executable



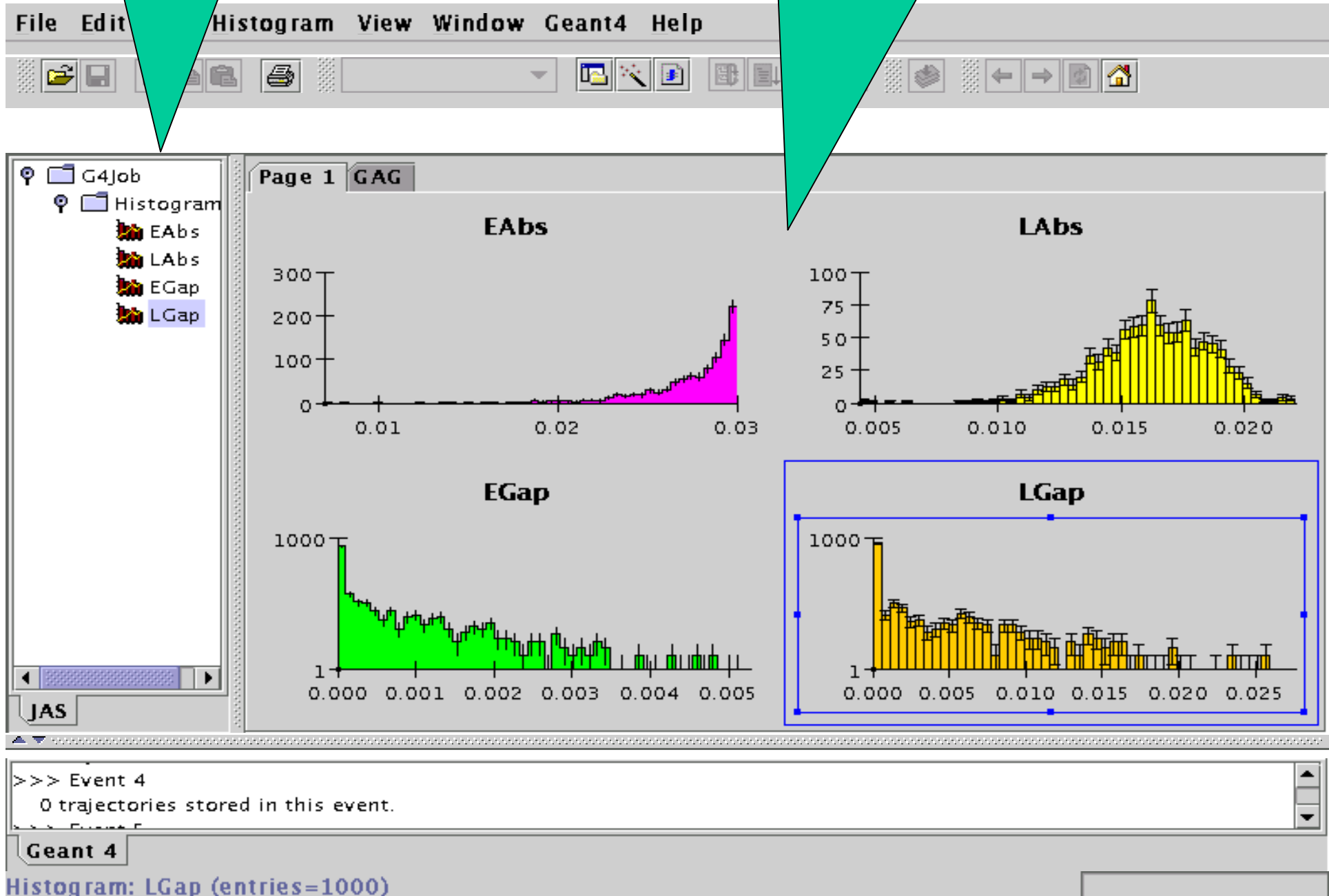
Command Arguments

G4 Output

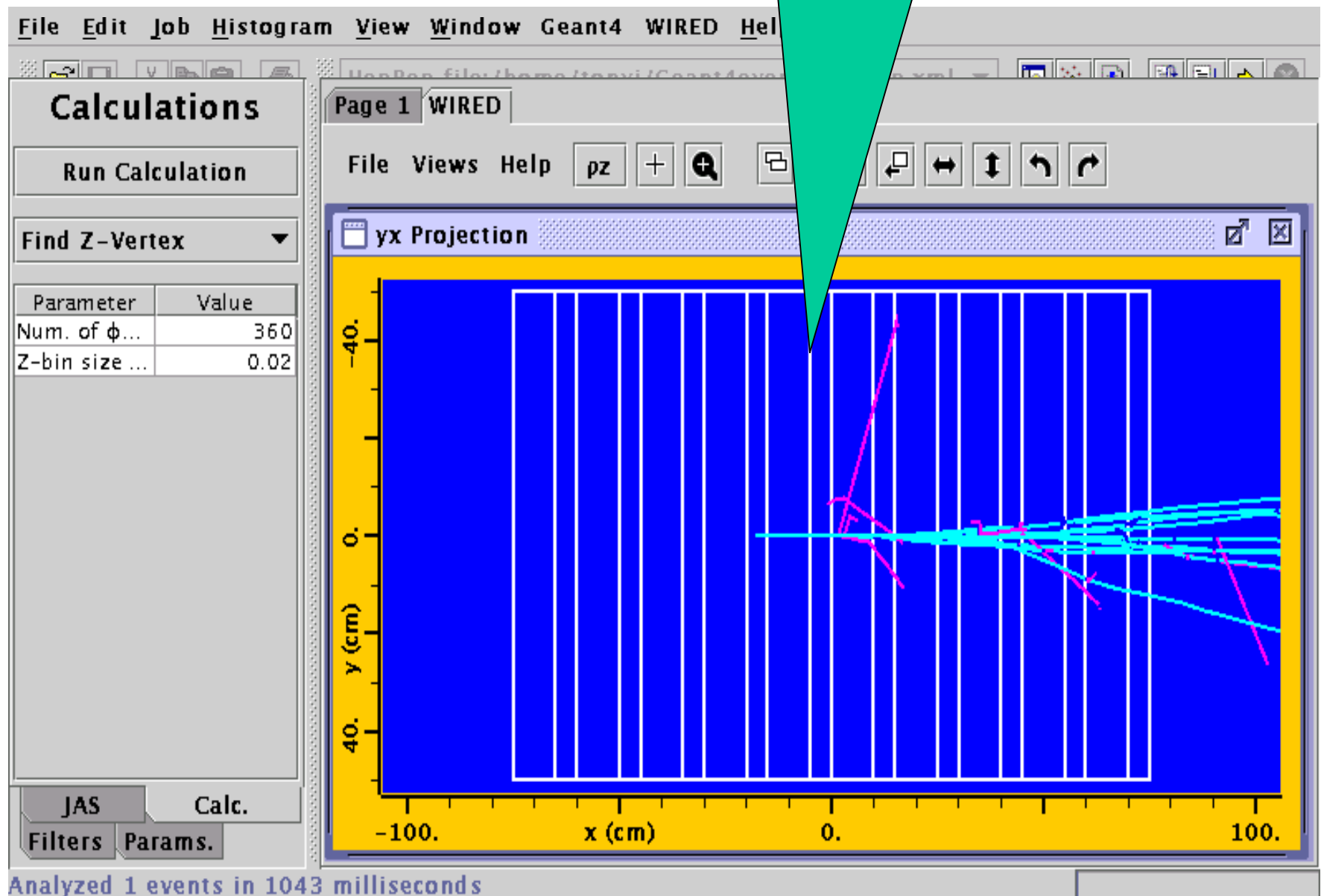
Currently Available Commands

Available Histograms

Histos update in real time



Wired Event Display
shows current event



Problems with Current System

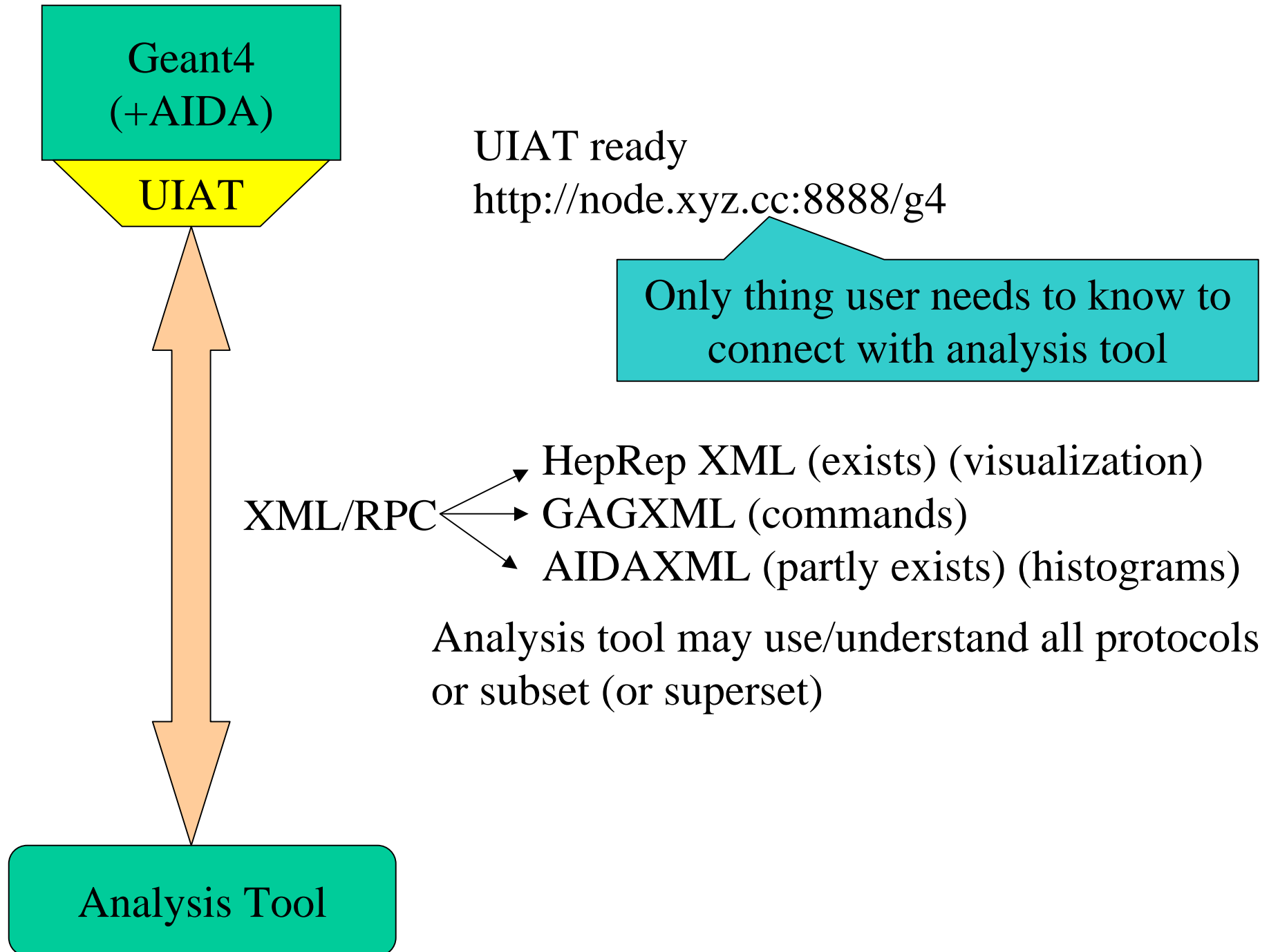
- **Too hard for users to configure**
 - Must build Geant 4 with correct environment variables set (to select for example GAG, JAS)
 - User should not need to recompile/relink to use different analysis tool.
 - Need to link with JavaVM for JAS histogram access
- **Still uses AIDA 1.0**
 - Need to move ASAP to AIDA 2.x
- **Protocol underneath is a mess**
 - GAG uses pipe and many special @@ commands
 - JAS uses Remote Method Invocation (RMI)
 - Wired uses HepRep + XML (currently written to file)
- **Need to support distributed analysis**

Future...???

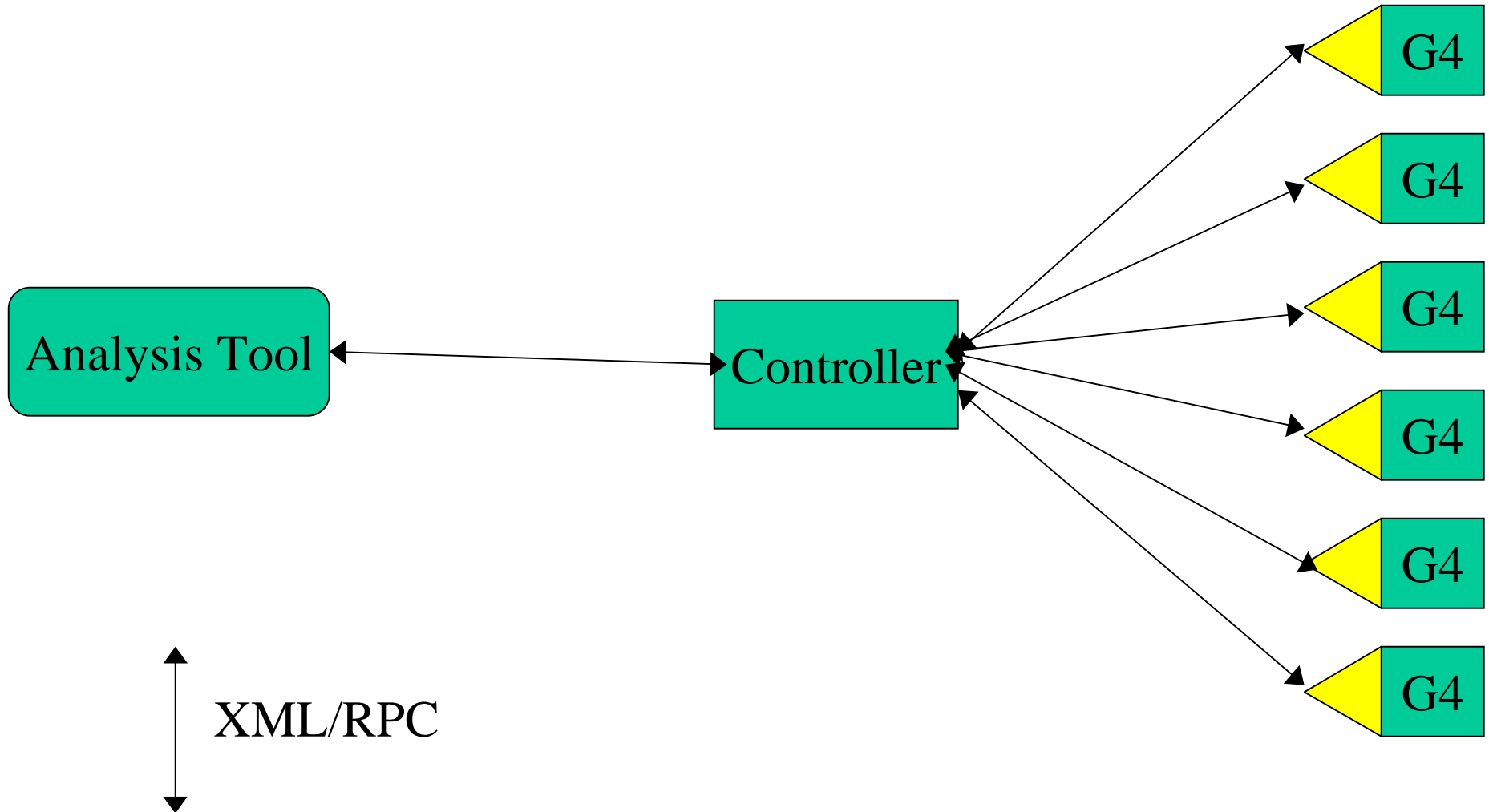
Warning – half baked ideas ahead

Unified Analysis Tool Interface

- Global UIAT
 - Reads commands from standard input (or macro)
 - Can be used in batch
 - AND listens for commands on socket connection
 - UIAT must be multi-threaded, but not G4 itself
- Should use XML/RPC for communication
 - Simple, light-weight, language neutral, open source
 - Uses http and XML for communication



Naturally extends for GRIDification



Conclusion

- JAS+WIREG+GAG+AIDA
 - Gives single GUI combining
 - Visualization
 - Histogram Filling/Viewing/Saving
 - GUI/Command interface
- Needs more work on ease to setup
 - Perhaps can make communication independent of analysis tool?

Acknowledgments + Links

- Thanks to:
 - Wired Team, GAG Team, AIDA Team, FreeHEP developers
- Links
 - <http://jas.freehep.org>
 - <http://aida.freehep.org>
 - <http://wired.cern.ch>
 - <http://heprep.freehep.org>
 - <http://java.freehep.org>