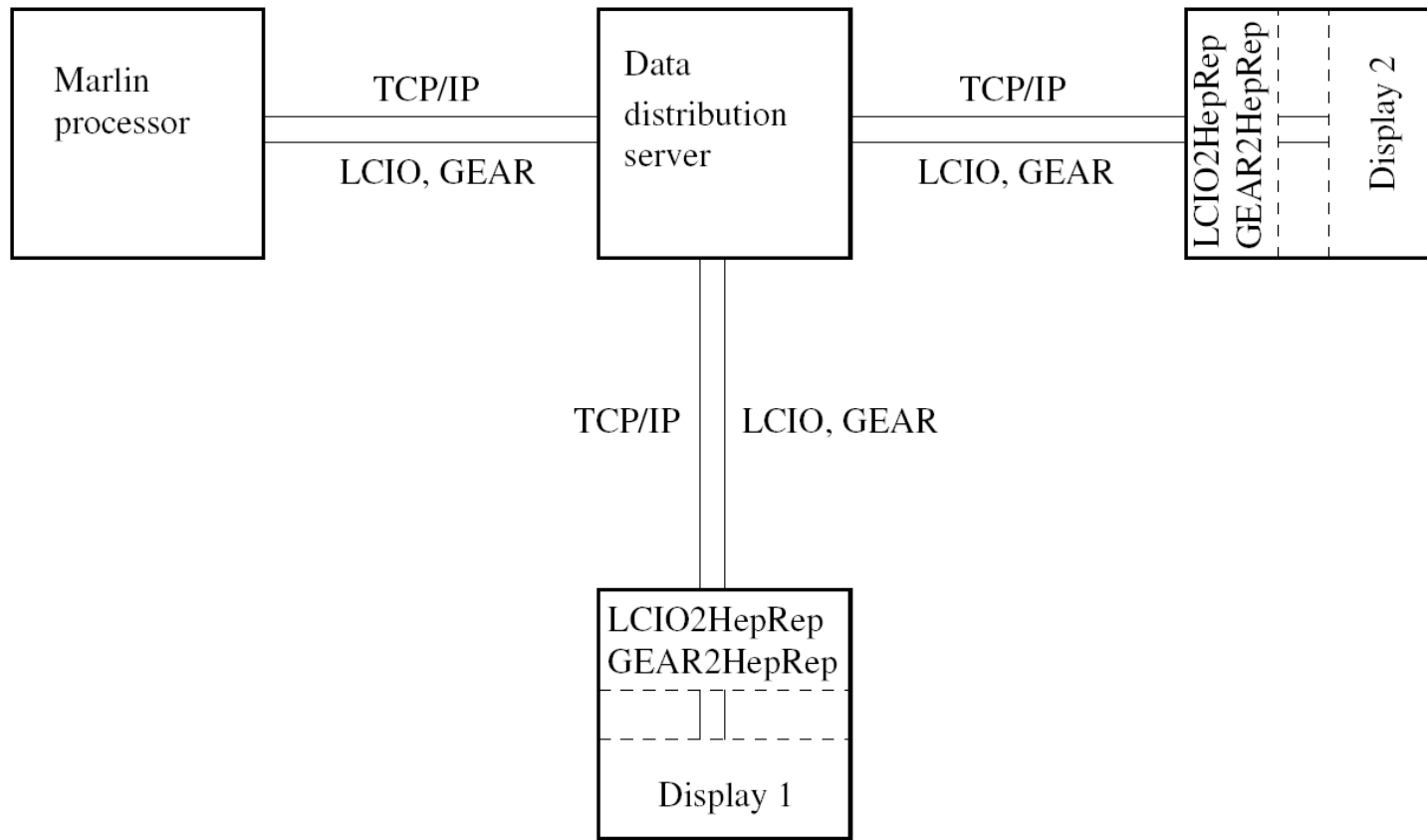


Event Display Discussion

- Discussion started because many 80% solutions are there but no 100%
Many groups seem to be interested
 - CED
 - MarlinED (based on CED) by Jason and TPC display from Jim Hunt
 - CALICE display
 -
- What do we want?
Suggestions:
 - Client/Server approach?
 - Usable for online (testbeam etc.) and offline (analyses) use?
 - 3D representation of full LDC detector and prototypes (OpenGL)
 - Cuts on displayed data
 - For prototype display: Control plots should be possible (histograms, graphs etc.)
 - Several different views (representations) should be possible
 - Has to be usable with existing framework (LCIO+GEAR)
 - Should be flexible to meet needs of different groups
 - Should be at least moderately fast (how fast do we need?)
 - Which level of detail is needed, that is possible to display at a usable speed?

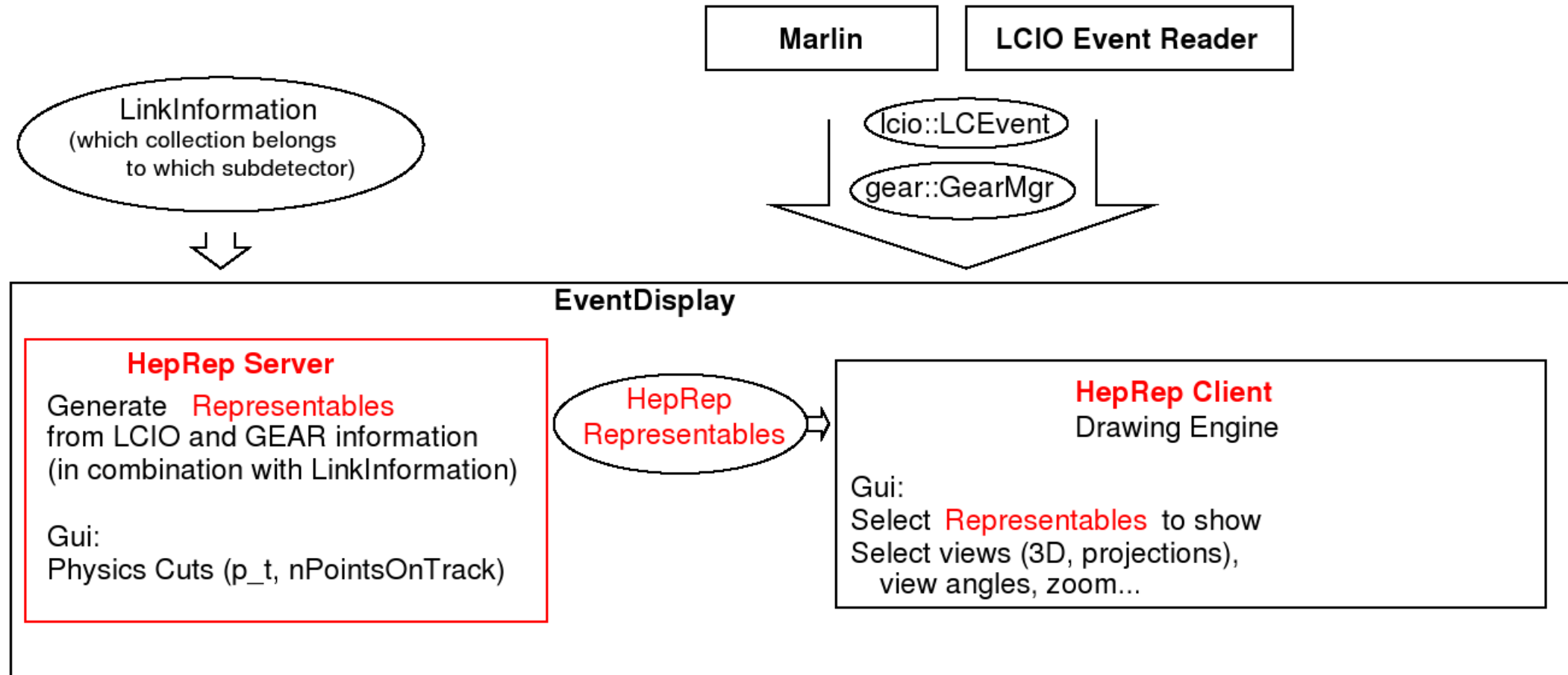
Client/Server?

- Sketch from Peter (MarlinTPC meeting):



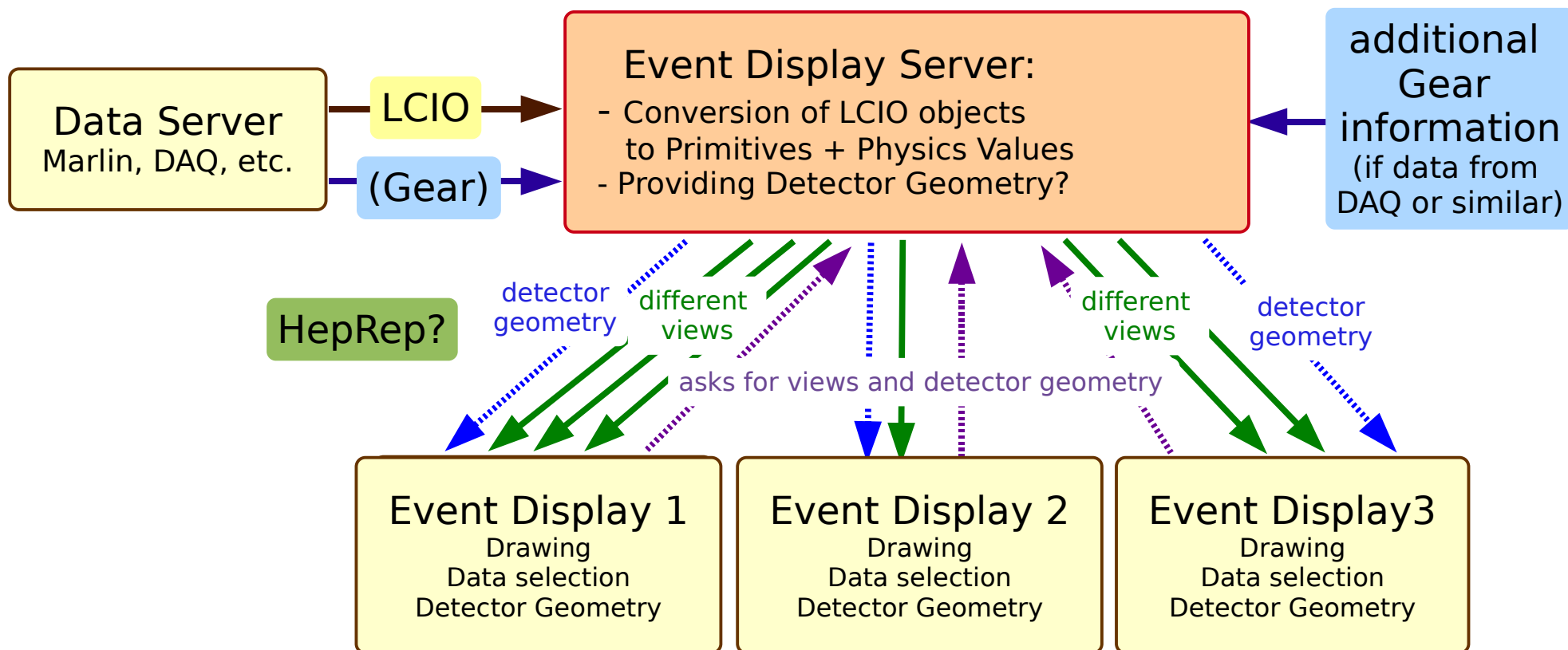
Client/Server?

- Outline proposal from Martin (MarlinTPC meeting)



Client/Server?

- **Main Question:** Where should the logic be?
- **In the server:**
 - sends ready to display data to a display client which only draws the event (and allows picking and changes to draw options)
 - Possible transfer format: HepRep
 - Several views by choosing different sources on server? (large data transfer?)
- **In the client:**
 - receives only data (and geometry) and creates representables itself → full flexibility for the user
 - Add LCIO drawing functions?



HepRep Basics

- Advantages:
 - Standard format for HEP event displays
 - Many solutions already available (Java and C++ implementations, display, servers etc.)
- Disadvantages:
 - Large files (binary HepRep?)
 - Another intermediate layer, an additional file format
- Overview over possible primitives:

Primitive values and their meaning.

DrawAs Value	# of Points	Behaviour
Point (default)	1*	Draws a marker at every point.
Line	2*	Draws a line segment between all the points given.
(Helix)	1*	Draws one or more of helices.
Polygon	3*	Draws a (closed) polyline connecting the last point to the first point.
Circle	1*	Draws one or more circles centered around the given points.
Curve	4+3n	Draws a cubic-bezier curve for the given points. Order of Points is as normally used for bezier curves. First section of the curve needs 4 points: P0, CP0b, CP1a, P1 where P is Point and CP is Control Point. Subsequent sections need sets of 3 points: CP1b, CP2a, P2 through CPn-1b, CPna, Pn.
Ellipse	1*	Draws one or more ellipses centered around the given points.
Ellipsoid	1*	Draws one or more ellipsoids centered around the given points.
Prism	2n	Draws a prism. First half of the Points define inner face, second half define outer face. Faces are connected from point 0 to point n/2, point 1 to point n/2+1, etc.
Cylinder	2*	Draws one or more cylinders or cones. Every end can have its own radius and angle.
EllipsePrism	2*	Draws one or more cylinders with elliptical ends. Every end can have its own ellipse parameters.
Text	0	Draws text.

Binary HepRep

Uses WBxml, see <http://www.w3.org/TR/wbxml/>, which is

- a web standard to send XML to wireless devices
- each tag/attribute has a binary code
- structure is the same as XML
- numbers can be written as doubles and integers

Advantages

- smaller than XML
- compact
- can be still compressed by zlib compression
- faster to parse
- easy lookup of tags and attributes
- no conversion of ASCII numbers to doubles and integers

Disadvantages

- not human readable/editable
- HepRep interfaces need to be stable

Implementations

- freehep-jheprep contains both a Binary HepRep reader and
- freehep-cheprep contains only a Binary HepRep writer
- Added as supported format in Geant 4.7.1
- Supported as reader in WIRED 4.0

HepRep Displays: Personal(!) Experiences

- WIRED:
 - Version 4 included as plugin in JAS3
 - Displays HepRep input
 - Gets slow with detailed geometry / event
 - Good functionality (picking, rotation etc.) and easy to use
- FRED
 - Based on Ruby
 - I couldn't get newer versions to run on DESY SL4 machine and Ubuntu 7.04
 - Older version did not run stable
- HepRApp:
 - Java Application in single JAR file
 - Difficulties to open some examples (seems quite picky)
 - Also slow with detailed events
 - Cumbersome controls (a bit confusing)
 - **[Tip]** if you want to try it, start display with:

```
java -client -Xmx384m -Dsun.java2d.opengl=true -jar HepRApp.jar
```

HEPREP Experiences

- Small and simple test program which writes 100 events from LCIO file (TPC prototype data) to HepRep XML file
- LCIO file size (1000 events) \approx 28MB (Event: 98304 ADC values in 192 TrackerRawData objects)
 - used 100 events and 256 of 512 timebins:
 - LCIO data effective size \sim 1.5MB** \leftrightarrow \sim 5M ADC values (4,915,200)
- **Added information:**
 - 192 x channel sum (to display pulses on pad plane)
 - geometry (xyz + simple detector)
 - drawing options (primitive, color)
- 1st test with simple C++ driver utility for HepRep 1.0 XML files from Joseph Perl

- Results:

Type	Duration	Size Zipped	HepRepXML
all values	369 sec	32MB	1.9GB
noise threshold	9 sec	1.1MB	39MB

+ zip: 54 sec = 423 sec

- 2nd test: cheprep HepRep 2.0 C++ driver for XML or Binary HepRep files (incl. zip while writing)

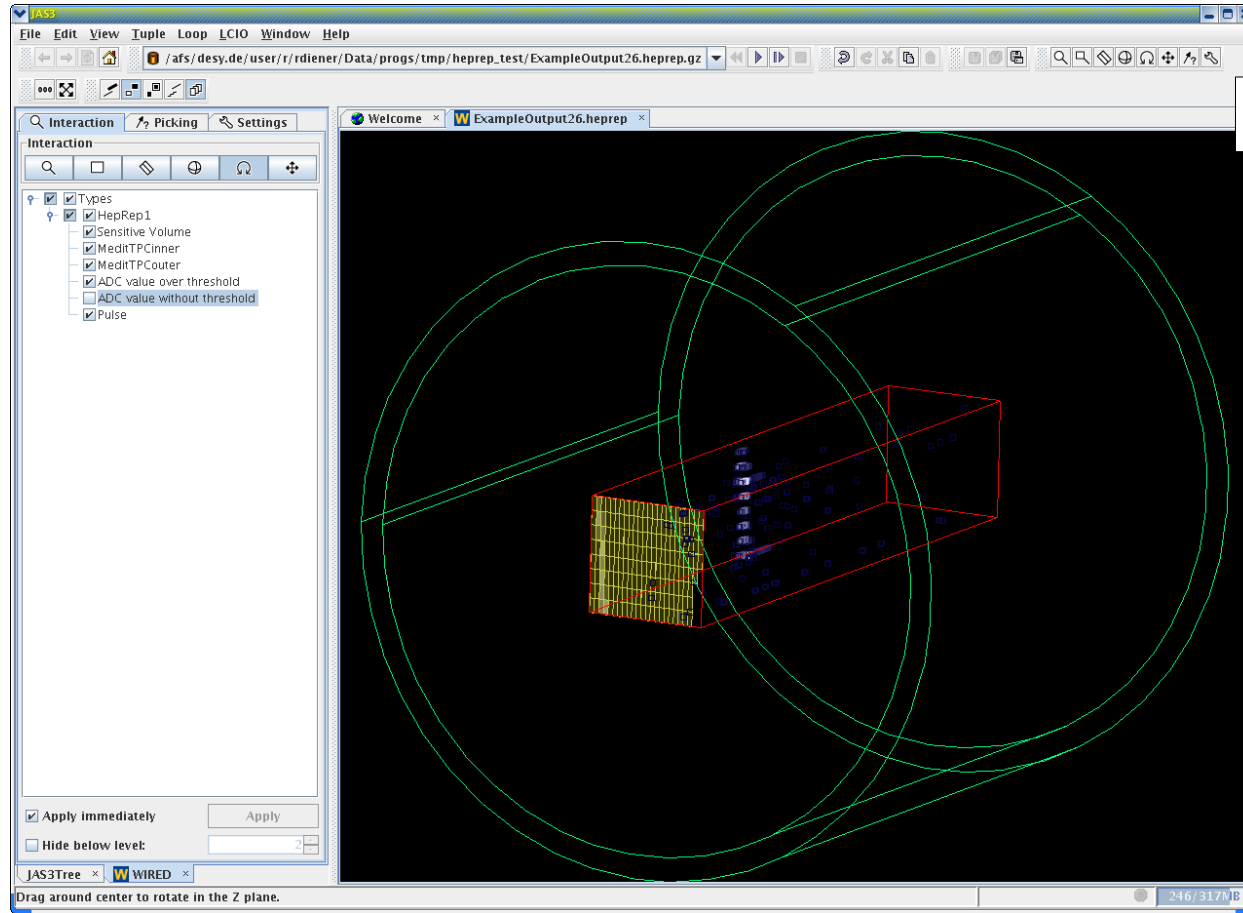
- Results:

Format	Type	Duration	Size Zipped	Deflated
HepRep XML	all values	828 sec	31MB	1.5GB
	noise threshold	16 sec	1.1MB	25MB
Binary HepRep	all values	583 sec	31MB	261MB
	noise threshold	10 sec	789KB	5.0MB

could probably be optimized

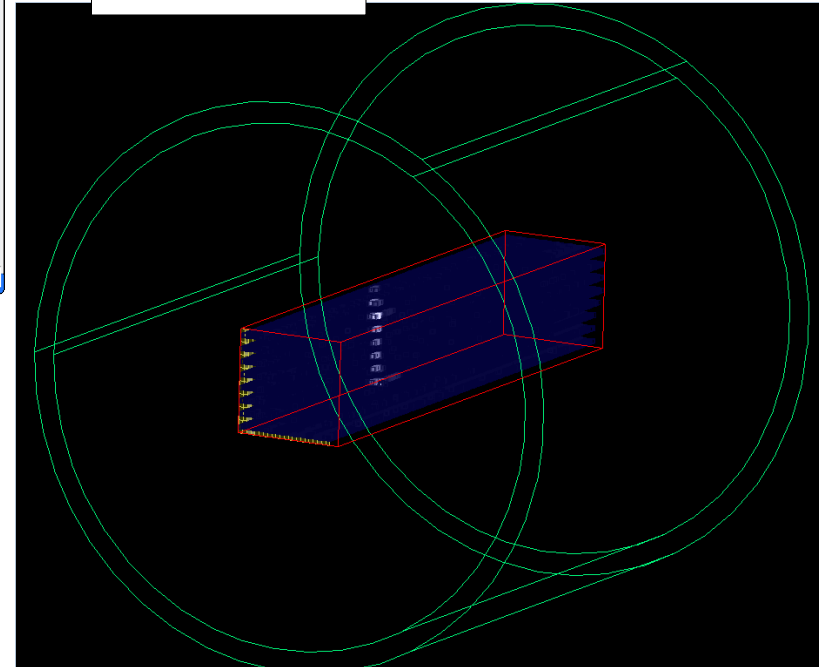
HEPREP Experiences

- Plots of the example geometry and Event



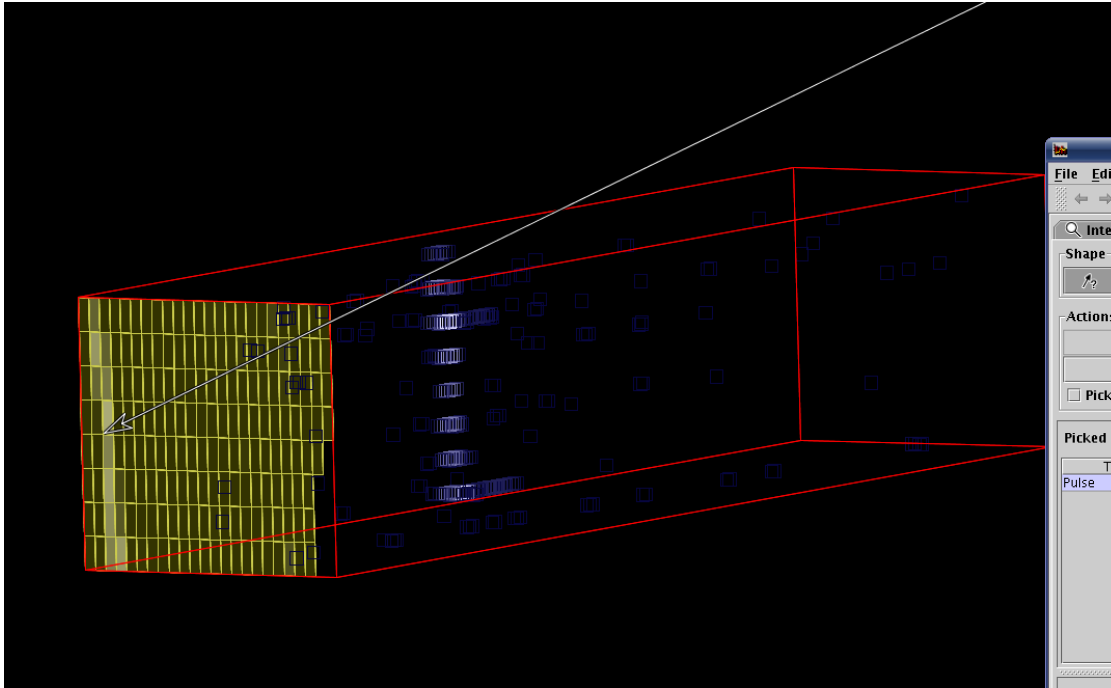
noise threshold

all values



HEPREP Experiences

- Picking in Wired4 (JAS3):
 - Shows information written in HepRep file



The screenshot shows the JAS3 software interface with the 'Picking' panel active. The panel displays the following information:

Interaction | **Picking** | **Settings** | **Cuts**

Shape

Actions / Settings

- Zoom into Region
- Translate to Picked Object
- Pick while Moving/Dragging

Picked objects (1): Options...

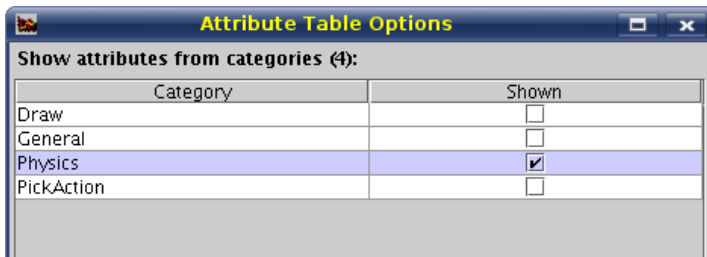
Type	Points	Children
Pulse	8	0

Attributes of picked object (12): Options...

Name	Value	Unit	Node
Channel Number	1645.0		Instance
Phi1	0.0000		Type
Phi2	0.0000		Type
Pulse Charge Sum	7.0000		Instance
RadiusX	5.0000		Type
RadiusY	5.0000		Type
RadiusZ	5.0000		Type
Sum	5.0000		Type
Sum1	5.0000		Type
Sum2	5.0000		Type
Theta1	0.0000		Type
Theta2	0.0000		Type

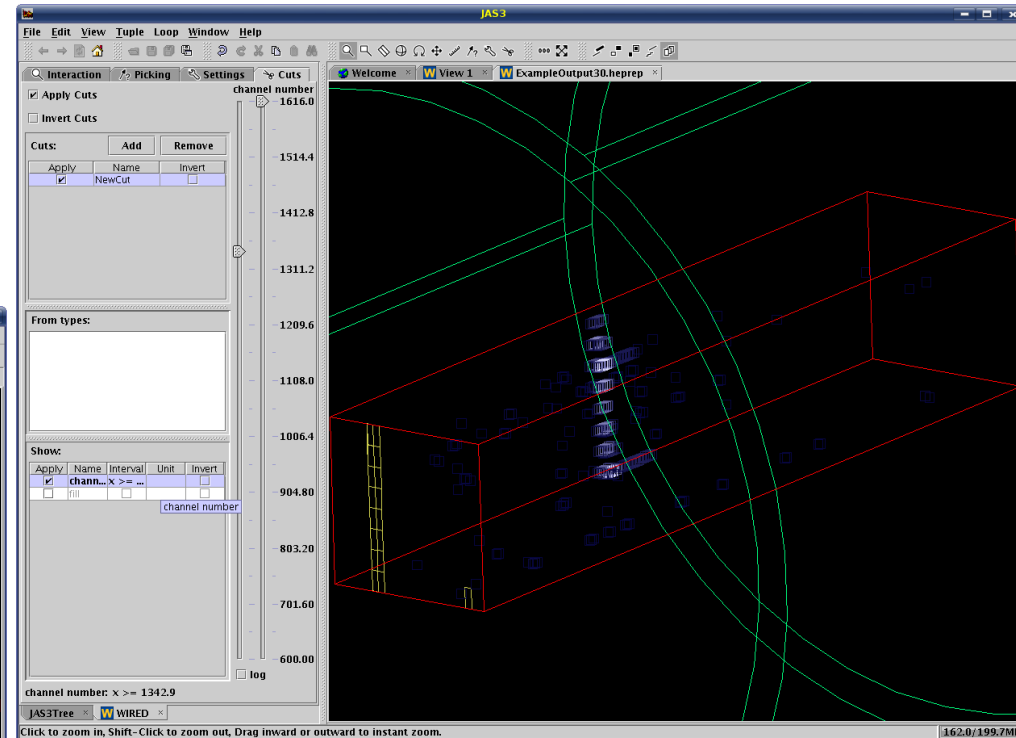
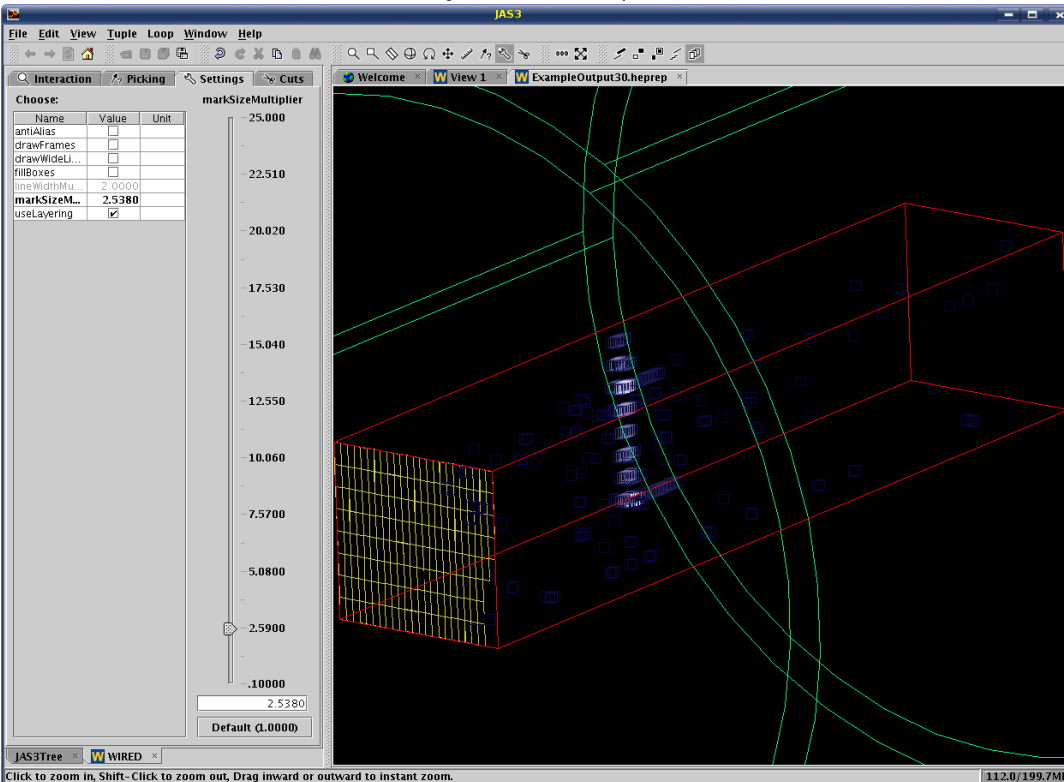
At the bottom of the interface, there is a status bar that reads: "Hover over or near an object to show information."

- Different categories of displayed information possible



HEPREP Experiences

- Options in Wired4 (JAS3):
- Cuts on properties possible (here channel #) →
- Value not only as color but also as size possible ↓



- General experiences while working on this example:
- Not very well documented (lookup in *.h files and examples)
- Needs quite some RAM

Misc

- Can we use an existing solution and expand it?
- 3D drawing speed could probably be optimized → Jan's talk?
- LCIO/Gear transfer over TPC/IP: use implementation for DAQ solutions?
- Another possibility: ROOT (used for example by ALICE: AliEVE)
- Manpower could be a problem → separate tasks and distribute effort (different groups)?

- Links:
 - [Discussion thread in ILC forum](#) (contains many links, too)
 - [HepRep / HepRep2](#)
 - [Agenda of MarlinTPC discussion](#) (and [minutes](#) of this meeting)