

Online example: ced_{12} [†]

**Seeing Tracks Through
Thick and Thin^{††}**

Christopher Newport University

JLAB

27 May 2011

Heavy Photon Search Meeting

[†]As in cLAS eEVENT dISPLAY (12 GeV)

^{††}As in *thick* and *thin* clients

Outline

1. Introduction
2. Thick & Thin
3. Architecture
4. Features
5. Extendibility
6. Availability

1. Introduction

- Modern version of 6 GeV *ced* (running virtually unmodified for ~15 years)
- It is an *event* display, not a *detector* display
 - Primary role is *not* to visualize the detector.
 - Primary roles: Help *debug* and *diagnose* the detector (online) and to assist in analysis (offline).
 - Unfaithful (to the geometry) displays are often more useful than faithful displays. Especially when there is a lot of “air.” Also, 2D often more useful than 3D.

Outline

1. Introduction
2. Thick & Thin
3. Architecture
4. Features
5. Extendibility
6. Availability

2. Thick and Thin

- Thick: Traditional, full-featured Desktop App
- Thin: **Same** full-featured “Web 2.0 app” aka “Rich Client/Internet Application” delivered in a browser[†]
- We’ll deliver **both**, using (approximately) the same code base

[†] With, perhaps, some minor security related annoyances, such as no access to a local file system—e.g., *ced*₁₂ will not be able to upload your *Quicken* files to the CLAS calibration database. Honest.

Web 1.0 v. Web 2.0

- **Web 1.0**

- Web delivers documents
- Web apps are stateless
- HTML based; browser *renders*

- **Web 2.0[†]**

- Web delivers Rich Internet Applications (RIAs)
- Applications maintain state
- Data centric; browser *contains and delivers*



Rich Internet Apps

1. Browser ***delivers*** virtual machine and ***provides*** real estate.
2. Compiled application runs in vendor VM.
3. VM, not browser, renders.
4. Browser's primary role has changed! It is a VM container.

[†] Web 2.0 is here *now*. Sometimes providing dramatic new interfaces. Sometimes, e.g., *NetFlix*, it takes a decent site and redoes it in a way that produces a much more appealing desktop-like response and experience.

RIA Technologies

- Adobe FLEX (2004.) Uses FLASH player as VM. **~97 percent penetration across all platforms.**¹
- Microsoft Silverlight (2007.) So far, little penetration.⁶⁶⁶
- SUN JavaFX (too late—little chance to succeed.)
- HTML 5 (Interesting—essentially dumps the VM responsibilities onto the browser developers.)³

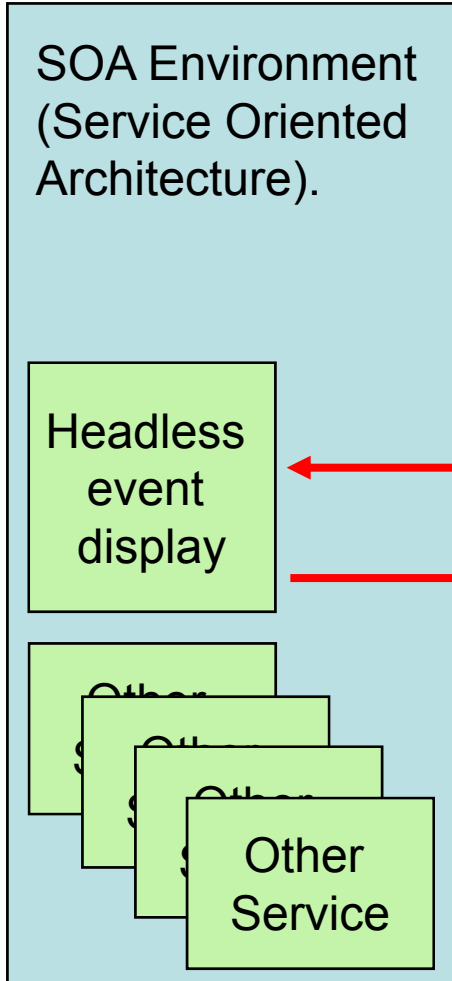
¹ This is the best reason for adopting FLEX; ~nobody will have to download anything.

⁶⁶⁶ But yes, I agree, that is one hard-to-ignore 800lb gorilla.

³ Specification to reach the W3C Candidate Recommendation stage 2012, and W3C Recommendation in the year **2022** or later! However, many parts of the specification are stable and may be implemented early. (Source: *wikipedia*)

Google Maps Paradigm

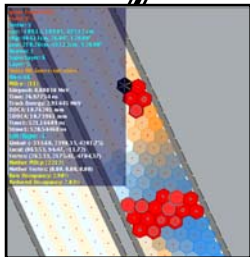
Client tools overlaying server provided image



Toolbar provided by Flex

User zoom to extent

OK here's a new pic



Jefferson Lab - CLAS12 Software Workshop - Mozilla Firefox

Jefferson Lab > Events > CLAS12 Software Workshop

CLAS12 Software Workshop

LINKS

- Circular
- Registration
- Program
- Lodging
- Travel
- Visa
- Participants List

CLAS12 Software Workshop

University of Richmond
Physics Department

May 25-26, 2010

Topics:

- Modern methods for analysis of large data sets
- Status and Interactions for the CLAS12 collision

Done

Outline

1. Introduction
2. Thick & Thin
3. Architecture
4. Features
5. Extendibility
6. Availability

3. Architecture

CLAS 6 (and 12?) Software Motto:
“Standard is Better than Better” †



†If you were to say: “in *practice* it appears that the CLAS 6 motto was: **Complicated is Better than Better,**” I, for one, could not say that you were being uncharitable.

Two Plus One Libraries

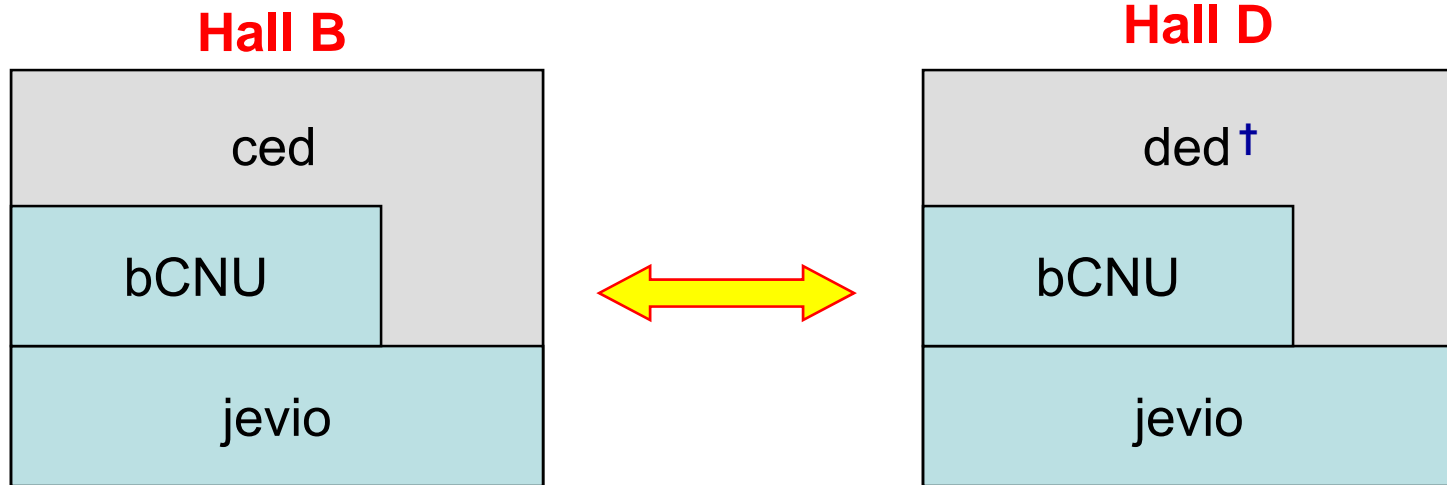
<i>Library</i>	<i>Purpose</i>	<i>Comments</i>
jevio	JAVA I/O for JLab evio format	Originally developed by—adopted and taken-over by JLAB DAQ
bCNU ¹	Multiple Doc Interface (MDI) Framework	JAVA based graphical package. <i>bCNU</i> provides framework and base classes, <i>but knows nothing about any specific detector.</i>
jogl ²	JAVA bindings to OpenGL (3D)	One of two free 3D JAVA solutions. Requires platform specific jars and shared libs.

On this platform we are building Hall B and Hall D event displays

¹ bCNU, i.e., “be seein’ you!” (unless that is too cheesy, in which case it stands for Hall **b** and **CNU** collaboration. Your call.)

² This is the “plus one.” It differs from the other two in that a) we didn’t develop it and, more importantly, b) it is quasi-platform dependent.

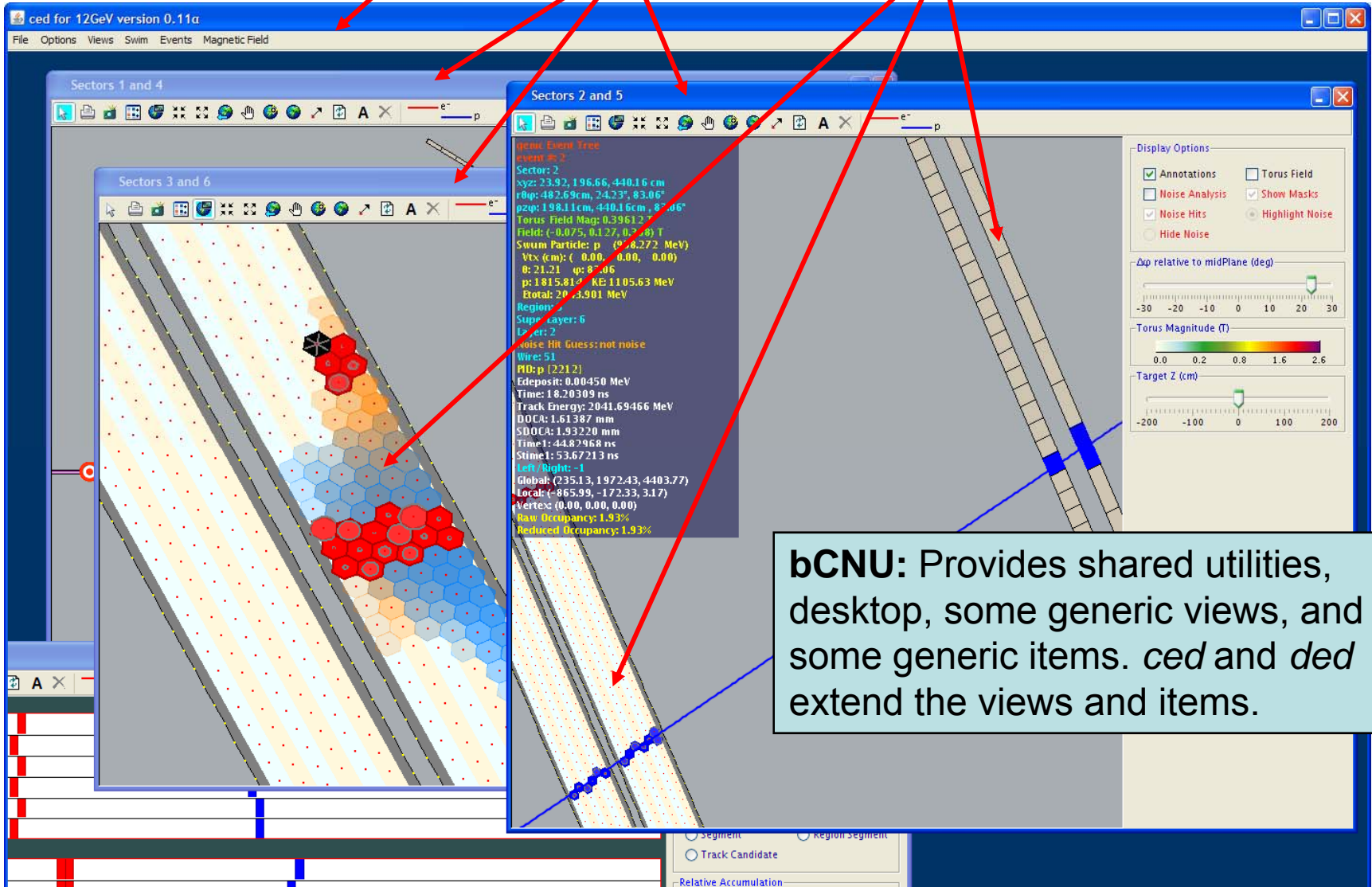
Shared Hall B/D Code Base



Goal: common (or potentially common) features are developed in *bCNU* (or migrated to *bCNU*) with the intent that the *ced* and *ded* code bases are $\sim \frac{1}{4}$ the size of the *bCNU* code base.

† *ded* (pronounced “*dee-e-dee*”) is the Hall D event display. This naming convention is bloody awful, since you would then think *ced* is the Hall C event display, and that *ced* should really be *bed*. But it is what it is.

Multiple Document Interface (MDI): Desktop, Views, & Items



Global Event Tree
 event no: 3
 Sectors: 2
 xyz: 23.92, 196.66, -440.16 cm
 rDip: 482.69cm, 24.23°, 83.06°
 pDip: 198.11cm, 440.16cm, 87.06°
 Torus Field Mag: 0.396127 T
 Field: (-0.075, 0.127, 0.378) T
 Sigma Particle: p (9.8272 MeV)
 vtx (cm): (0.00, 0.00, 0.00)
 θ: 21.21 φ: 87.06
 p: 1815.819 KE: 1105.63 MeV
 Rotat: 20.3901 MeV
 Region:
 Supr Layer: 6
 Layer: 2
 noise Hit Guess: not noise
 Wire: 51
 PID: p (2212)
 Edeposit: 0.00450 MeV
 Time: 18.20309 ns
 Track Energy: 2041.69466 MeV
 DOCA: 1.61387 mm
 SDOCA: 1.93220 mm
 Time1: 44.87968 ns
 Stime1: 53.67213 ns
 Left/Right: -1
 Global: (235.13, 1972.43, 4403.77)
 Local: (-865.99, -172.33, 3.17)
 Vertex: (0.00, 0.00, 0.00)
 Raw Occupancy: 1.93%
 Reduced Occupancy: 1.93%

Display Options
 Annotations Torus Field
 Noise Analysis Show Masks
 Noise Hits Highlight Noise
 Hide Noise

Δxp relative to midPlane (deg)
 -30 -20 -10 0 10 20 30

Torus Magnitude (T)
 0.0 0.2 0.8 1.6 2.6

Target Z (cm)
 -200 -100 0 100 200

bCNU: Provides shared utilities, desktop, some generic views, and some generic items. *ced* and *ded* extend the views and items.

Outline

1. Introduction
2. Thick & Thin
3. Architecture
4. Features
5. Extendibility
6. Availability

4. Features (Current *ced* Views)

<i>View</i>	<i>Comment</i>	generic
All DC	All the drift chambers—approximate geometry	
Sector	Split sectors 1/4, 2/5, 3/6. Faithful geometry. Currently DC and OTOF.	
Monte Carlo	Table of “event generator” records (if any present) showing what tracks were generated	
Event	Drag ‘n drop, expandable tree-view of evio events so that banks can be examined quickly (bCNU)	
Noise	A view with fake data used for testing/explaining the noise detection algorithm.	
Log	Info/Warning/Error messages for debugging (bCNU)	
Socket	Establish and manage evio over a socket (bCNU)	
XML	Drag ‘n drop, tree-view of any XML file (bCNU)	

Some Selected Features

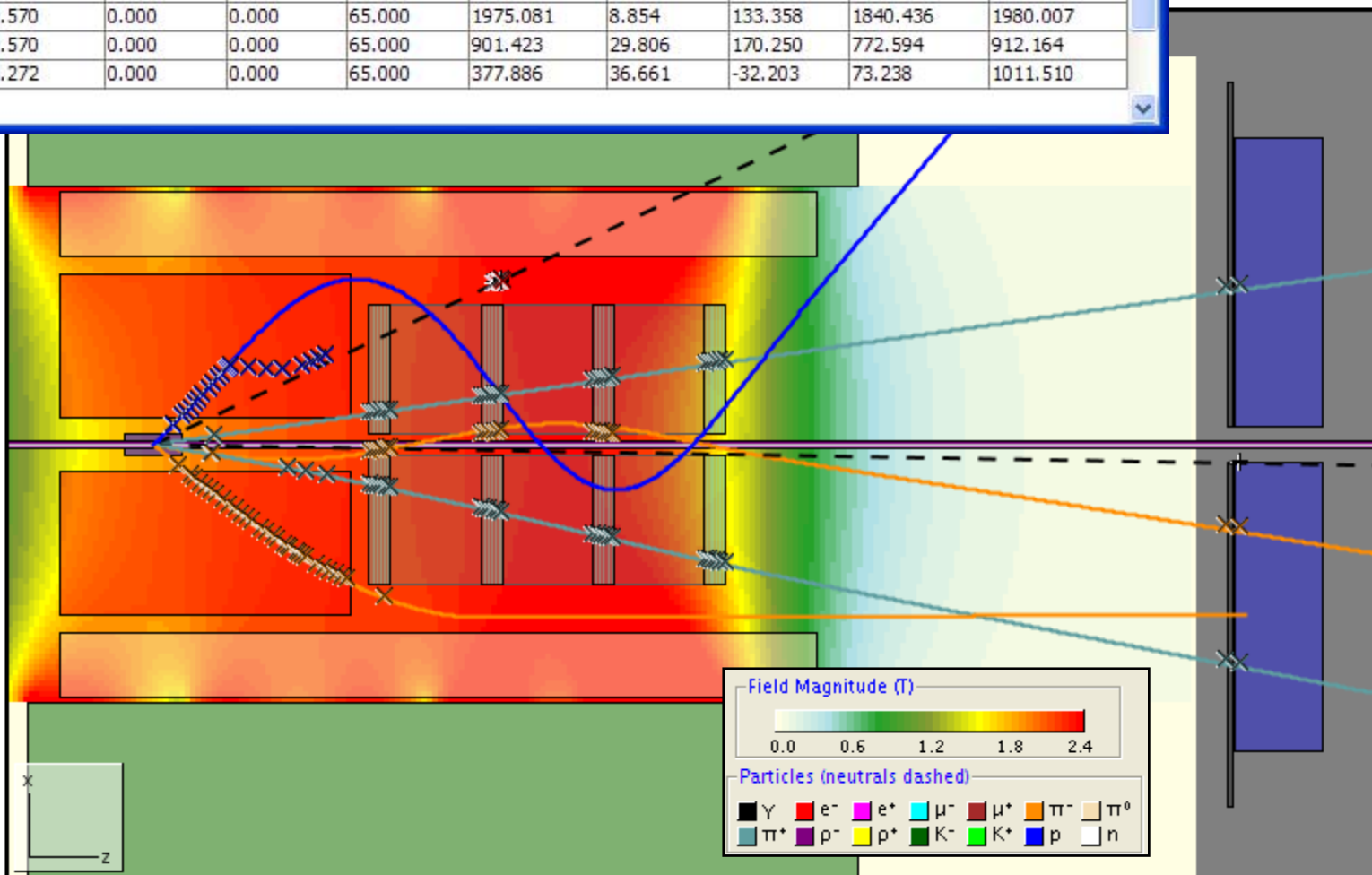
<i>Feature</i>	<i>Comment</i>
Zoom, pan, etc	Expected features for view manipulation (bCNU)
Snap shot	High quality .png image of active area (bCNU)
Heads-up	Mouse-over information displayed on a semi-transparent heads-up display (to preserve real estate) (bCNU)
Moving target	Trivial: target z-location can be changed
Magnetic Field	Uses same field as GEMC
Accumulate	Accumulation mode for looking for hot spots/dead zones
Swim	Runge-Kutta 4 th order for swimming particles (bCNU)
Noise	Improved display of results of noise detection
Clusters/ segments/ candidates	Highlight clusters, segments, track candidates, etc. from the <i>socrat</i> family of track-finders (or any track-finder that stores results in same banks)
Auto rotate	Rotate to initial ϕ of track to see if it lines up with DOCAs

Some Snapshots

Monte Carlo View

Monte Carlo Events										
Id	name	m (MeV)	x ₀ (cm)	y ₀ (cm)	z ₀ (cm)	p (MeV)	θ	φ	KE (MeV)	Et (MeV)
211	n*	139.570	0.000	0.000	65.000	4760.691	5.539	-15.289	4623.166	4762.736
-211	n ⁻	139.570	0.000	0.000	65.000	299.873	7.415	168.100	191.192	330.762
22	γ	0.000	0.000	0.000	65.000	209.421	20.266	-31.128	209.421	209.421
22	γ	0.000	0.000	0.000	65.000	731.363	2.132	-107.388	731.363	731.363
211	n*	139.570	0.000	0.000	65.000	1975.081	8.854	133.358	1840.436	1980.007
-211	n ⁻	139.570	0.000	0.000	65.000	901.423	29.806	170.250	772.594	912.164
2212	p	938.272	0.000	0.000	65.000	377.886	36.661	-32.203	73.238	1011.510

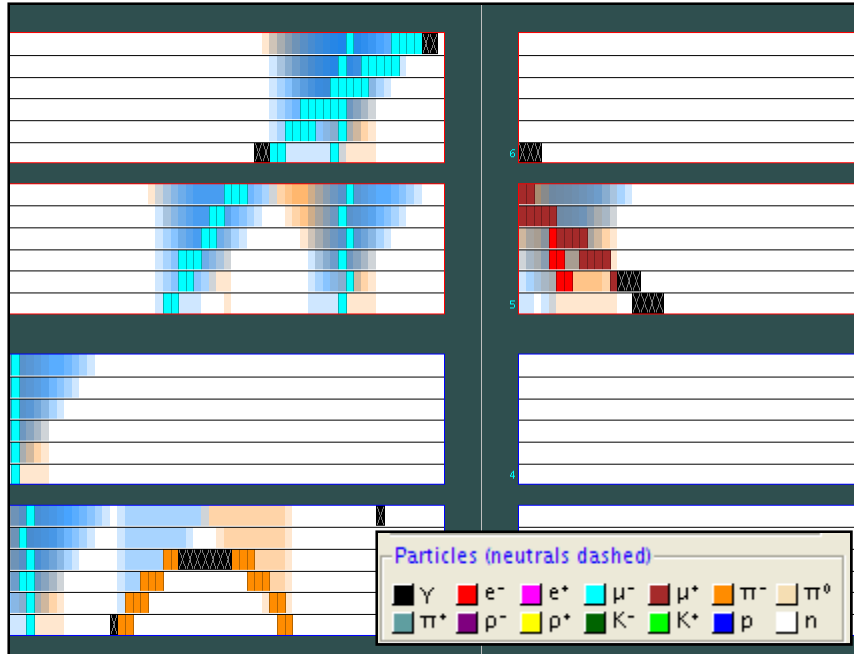
Hall D view
using same
bCNU
framework



More Snapshots

Noise Analysis

Event View



Event View interface showing event details and a list of event banks.

File path: t\ced\data\test.ev

event#: 12

num events: 10000

Array Data

[526]	-4.5641612
[527]	-4.7771458
[528]	-5.0160836
[529]	-4.9629501
[530]	-3.8373860
[531]	-4.0010420
[532]	-4.2391999
[533]	-4.2691126
[534]	-4.5887432
[535]	-4.7229978
[536]	-4.6443698
[537]	-4.8246501
[538]	-5.0483304
[539]	-5.2783288
[540]	-5.4474770
[541]	-5.7613397
[542]	-6.0842036
[543]	4.48237902

Event List:

- BANK of INT32s len (ints): 544 tag: 500 num: 1 datalen (bytes): 2172 [<#children: 0>
- BANK of INT32s len (ints): 544 tag: 500 num: 2 datalen (bytes): 2172 [DC_ digitize
- BANK of INT32s len (ints): 544 tag: 500 num: 23 datalen (bytes): 2172 [sector] <
- BANK of INT32s len (ints): 544 tag: 500 num: 24 datalen (bytes): 2172 [SuperLay
- BANK of INT32s len (ints): 544 tag: 500 num: 25 datalen (bytes): 2172 [Layer] <
- BANK of INT32s len (ints): 544 tag: 500 num: 26 datalen (bytes): 2172 [Wire] <#
- BANK of BANKs len (ints): 23937 tag: 500 num: 200 datalen (bytes): 95744 [DC_Regi
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 1 datalen (bytes): 4344 [Ede
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 2 datalen (bytes): 4344 [glob:
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 3 datalen (bytes): 4344 [glob:
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 4 datalen (bytes): 4344 [glob:
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 5 datalen (bytes): 4344 [local
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 6 datalen (bytes): 4344 [local**
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 7 datalen (bytes): 4344 [local
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 8 datalen (bytes): 4344 [time]
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 9 datalen (bytes): 4344 [parti
- BANK of DOUBLE64s len (ints): 1087 tag: 500 num: 10 datalen (bytes): 4344 [trac

Structure: BANK

Tag: 500

Length: 4348 bytes

Data type: DOUBLE64

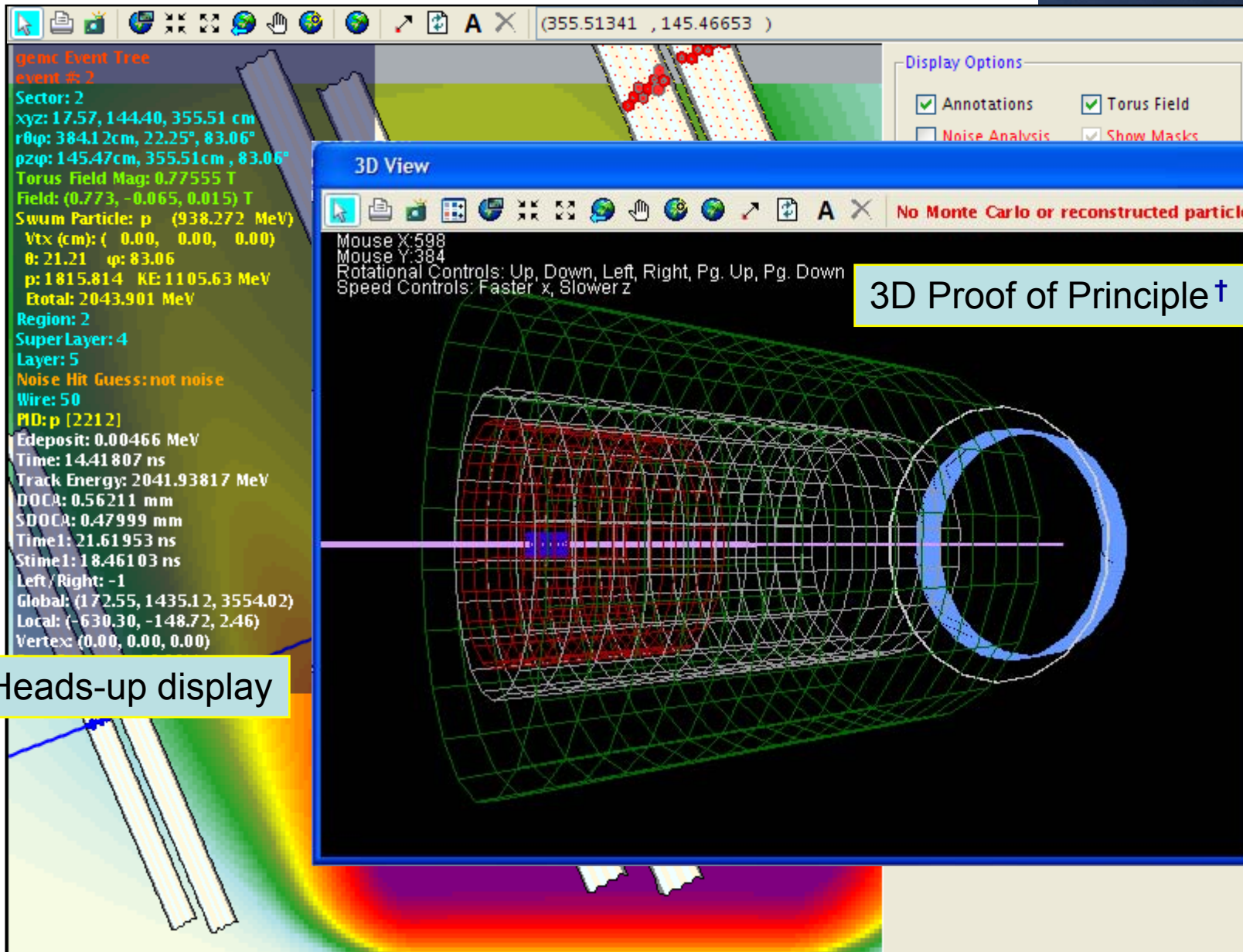
Number: 6

Description: local_y

Progress: progress

File Explorer showing the directory structure:

- cedDevelopment
 - bcNU
 - ced
 - .svn
 - bin
 - config
 - data
 - .svn
 - a_zedrofield.ev
 - b.ev
 - clas12_torus_fieldn
 - clas12_torus_fieldn
 - sector_1_wires.dat
 - sptorus_map.dat
 - sptorus_map_binar
 - test.ev
 - zerob.ev
 - docs



† For those who know about such things, the “lightweight v. heavyweight” issue is not a problem.

Outline

1. Introduction
2. Thick & Thin
3. Architecture
4. Features
5. Extendibility
6. Availability

5. Extendibility

Two steps to adapt another experiment:

- 1) Geometry → bCNU graphical primitives
- 2) Events → evio

These follow step 0, which is the hardest:
design what you want to see.

6. Availability: Obtaining *ced* †

`svn scheckout [URL]` Where [URL] is:

<https://clas12svn.jlab.org/repos/trunk/clas12/cedExport>

→ *cedExport*, with *ced.sh* for launching on linux, unix or Mac OS X. And *ced.bat* for launching on the other 95% of all computers. **There is no build procedure--**such is the beauty of Java.

From then on, use `svn update` → the latest.

On linux, launch the script via: `bash ced.sh`

† You need a JLab CUE account.

<rant> Forgot your password? Look on the little piece of paper in your desk! Since we all have multiple accounts, all with out-of-phase über-unbreakable mandatory password shelf lives, we (well, not me) have resorted to writing them down. Everything is *much* more secure! **</rant>**