

High Energy Physics



Fireworks

Physics Analysis Event Display

Dmytro Kovalskyi - **UC, Santa Barbara**

on behalf of the *CMS* offline and computing projects
with active participation of

Matevz Tadel, Alja Mrak-Tadel, Bertrand Bellenot – **CERN**;
Valentin Kuznetsov – **Cornell**; Chris Jones, Lothar Bauerdick - **FNAL**;
Mike Case – **UC, Davis**; Brian Bockelman - **University of Nebraska**;
Johannes Muelmenstaedt, Avi Yagil – **UC, San Diego**



Introduction



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- CMS has two event displays developed and supported by the collaboration, which complement each other:
 - **Iguana** - a general purpose full framework based application with ability to run reconstruction code on the fly. (See poster by Ianna Osborne)
 - **Fireworks** - a light-weight physics analysis oriented event display, showing only what is available in a standard CMS ROOT file.
- Fireworks development started in December 2007 and by Summer 2008 we had a first stable release available to users. By the end of 2008, the active development phase was replaced with maintenance and operation.
- The display is distributed as a stand alone application pre-build for Scientific Linux and Mac OS X.



Staying Focused



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- In order to make the project manageable, we concentrated on a specific use case: **physics analysis**
- Staying focused on the primary objectives allowed us to:
 - **Limit complexity** - ideal geometry, no reconstruction on the fly, simple track propagation, trivial magnetic field model etc
 - Use **schematics and abstraction** where it's adequate and don't try to be absolutely precise unless it's necessary
- These choices allowed us to:
 - Present a prototype in 4 months and release first stable version in 6 months since project started
 - Make display light enough that it can be easily distributed as a single tar-ball with minimal number of external dependencies
 - Port display to Mac OS X (and Windows - prototype only)



Direct Data Access



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- CMS data model is based on **ROOT file format**
- **FWLite** is ROOT plus a bunch of libraries with **Reflex** dictionaries, which allow direct data access from ROOT based applications.
- Our choice to show only what is stored in data allows us to use **direct data access with limited resources**.
- In practical terms it means a user downloads 100-200MB size event display ("player") and copies files of interest to her computer, which is all that she needs to get started.



ROOT - Eve



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- Last **CHEP07** conference Matevz Tadel presented new ROOT package: Eve - a full scale **framework** for **event visualization** using OpenGL in ROOT
- Active participation of ROOT developers in Fireworks design and development allowed to extend Eve and add elements that **can be shared by many experiments** (Alice and CMS currently).
- Using Eve as a visualization framework we separated CMS specific tasks from object rendering, which is done by ROOT.
- This greatly speeds up the development and **allowed physicists to contribute directly** without special knowledge of such advanced technologies as OpenGL



ROOT Reflex



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- ROOT Reflex is used extensively in the event display
- It allows for rich **object and event filtering** capabilities, where methods are not hard-coded, but user defined
- Given large number of methods available for most physics object classes, we use Reflex to help users to **explore available methods** and use "Tab"-completion to improve usability
- Reflex also allows us to explore **inheritance structure** of the data object types. As a result we can use the base class rendering if we do not find a render for the specific class.
- In practice almost all CMS physics object classes inherit from a single base class (**Candidate**), which provide most basic information like 4-momentum of a particle. So any new data can be shown in some simple way without writing new code.



Usability



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- **Typical use case** – looking at events for a physics analysis on a laptop traveling to/from CERN.
- **Make it simple and usable**
 - Given that event displays are not everyday tools, program interface is optimized for intuitive and simple use.
 - **3D accelerator** is recommended, but not required. The event display is routinely **used and tested on non-accelerated computers** to make sure that performance is reasonable.
 - The display is distributed as a stand alone **player** with all necessary components including FWLite in a compact form.
 - No need for remote use over X11, which should be avoided on slow networks anyway.

Design Choices Summary



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- **Fireworks relies on:**
 - CMS side
 - EDM - Event Data Model
 - Code Factorization - Data Formats are kept separately
 - FWLite - ROOT with CMS libraries to access data
 - ROOT side:
 - Eve - visualization framework
 - Reflex - ability to explore object information in C++
- **Key Design Choices:**
 - Show only available information (no reconstruction on demand)
 - Ability to display events off-line without access to databases and event stores (Airplane test)
 - No intermediate data formats - you see what you have in data, the way it is stored and processed. Less room for "operator error".

Display overview

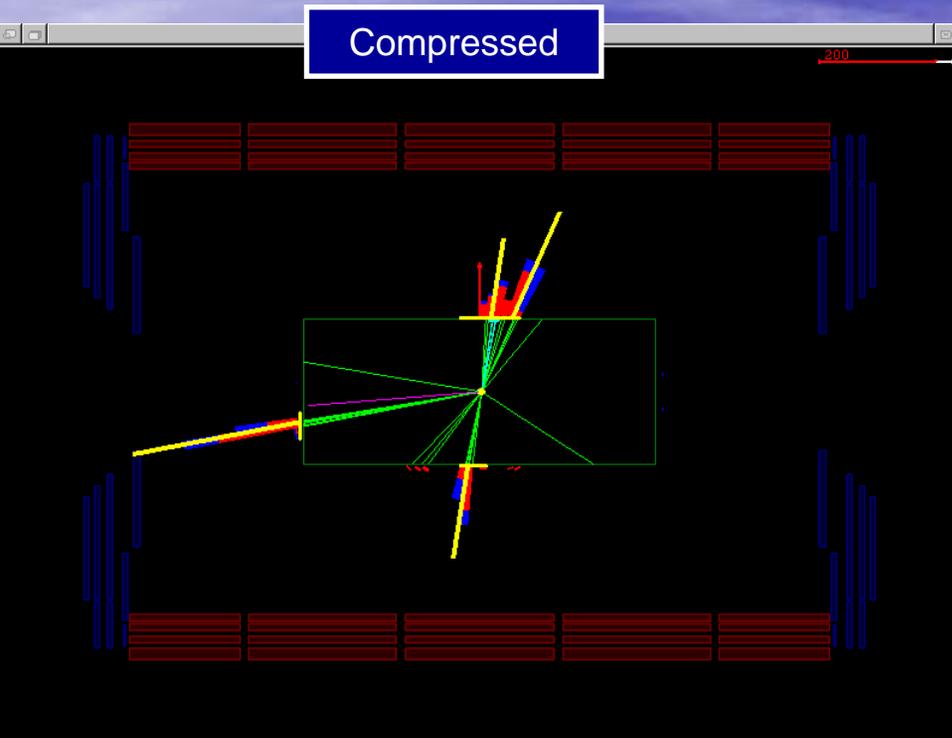
The screenshot displays the FIREWORKS software interface. At the top, there is a menu bar (File, Edit, View, Window, Help) and a control panel with navigation buttons, a delay slider set to 3.0s, and fields for Run (58555), Event (3038740), and Filter. The date and time are Sat Aug 23 12:52:12 2008, and the Lumi block id is 80. The FIREWORKS logo is in the top right corner.

On the left, a 'List View' panel shows a tree of data items with checkboxes. A blue box labeled 'Data Items' points to this panel. The items include ECal, HCal, Jets, Tracks, Muons, Muon 1 (pt: 4.4 GeV), Muon 0 (pt: 3.1 GeV), Electrons, Vertices, MET, DT-segments, CSC-segments, ActiveSiStrips, ActiveSiPixels, and TrackHits.

The main display area is divided into three panels:

- 3D view:** A large central panel showing a 3D reconstruction of the detector with a muon track highlighted in red. A blue box labeled '3D view' points to this panel.
- Rho-Phi:** A panel on the top right showing a 2D projection of the track in the Rho-Phi plane. A blue box labeled 'Rho-Phi' points to this panel.
- 3D Lego:** A panel in the middle right showing a 2D projection of the track in the Rho-Z plane. A blue box labeled 'Lego' points to this panel.
- Rho-Z:** A panel on the bottom right showing a 2D projection of the track in the Rho-Z plane. A blue box labeled 'Rho-Z' points to this panel.

Rho-Z View Options



Rho Z

Save Image ...

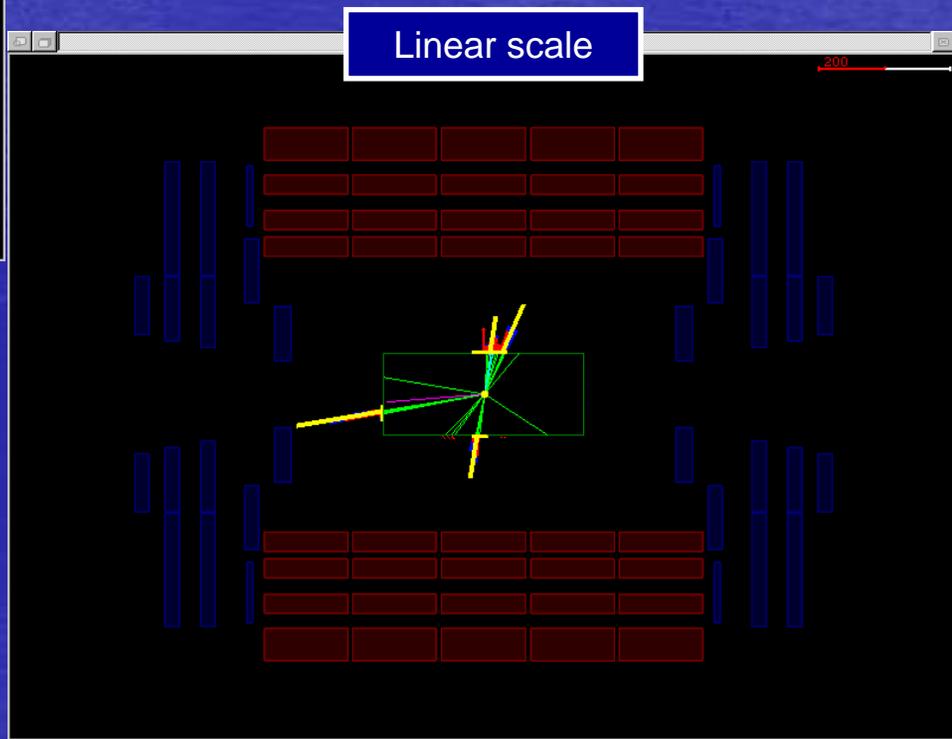
1 Calo compression

0.2 Muon compression

Compress detectors

2 Calo scale

Calo auto scale



Rho Z

Save Image ...

1 Calo compression

0.2 Muon compression

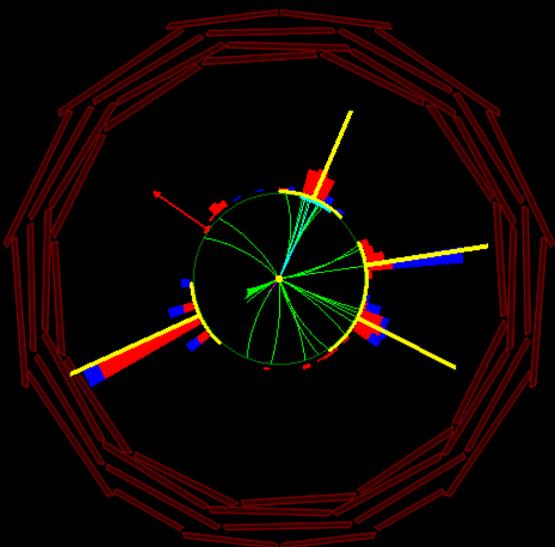
Compress detectors

2 Calo scale

Calo auto scale

Rho-Phi View Options

Compressed

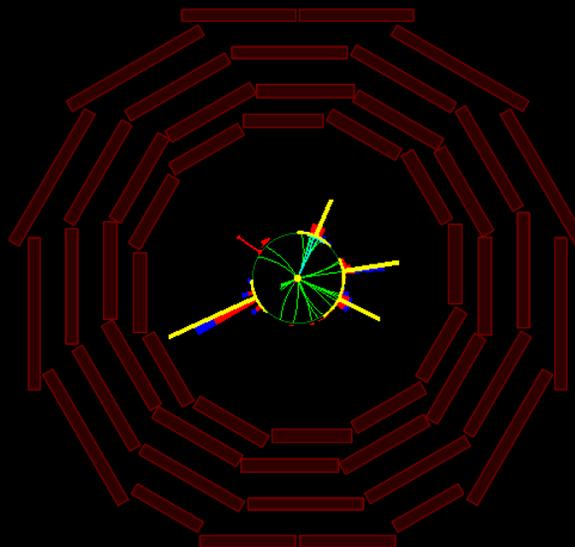


Rho Phi

Save Image ...

- Calo compression
- Muon compression
- Compress detectors
- Calo scale
- Calo auto scale
- Show calo endcaps
- Show HF

Linear scale



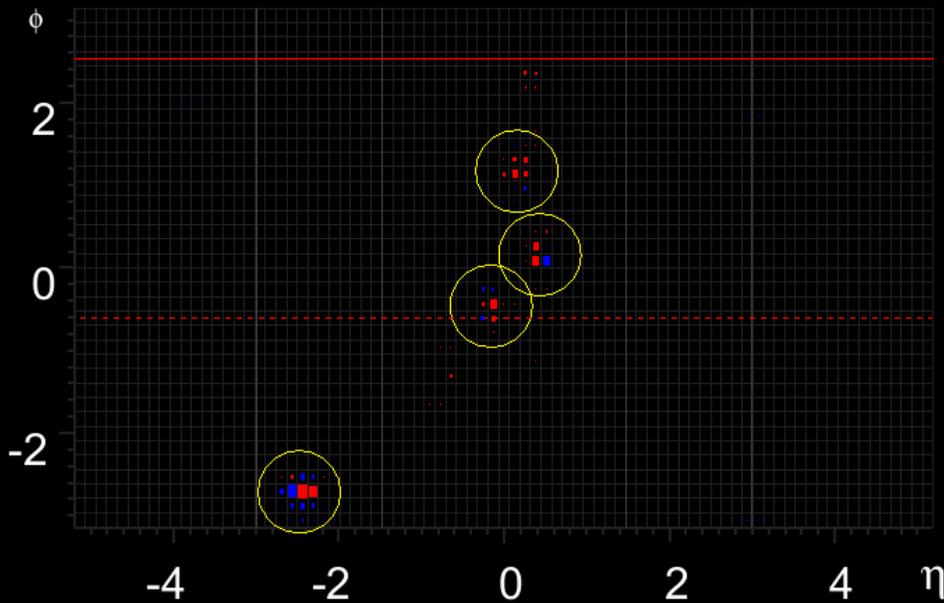
Rho Phi

Save Image ...

- Calo compression
- Muon compression
- Compress detectors
- Calo scale
- Calo auto scale
- Show calo endcaps
- Show HF

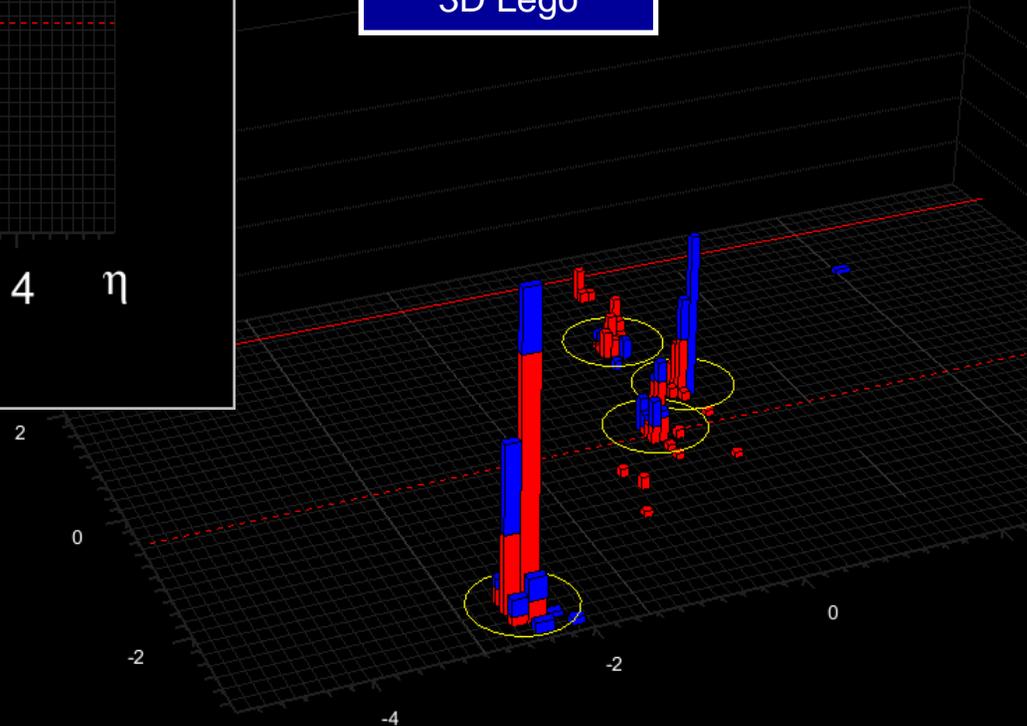
Lego View Options

Top Box View



- Top view shows color of the dominant energy contribution in each bin using log scale to map total energy to box size
- Bin size is automatically adjusted depending on the size of the view area

3D Lego



- At high resolution (large view area) binning is done according to the trigger tower size
- Currently the smallest bin is 2×2 tower, but this will be changed as soon as we test new code to support smaller bins

Text View Options

```

Dump to file event_dump.txt  append to file  Dump to terminal  Dump to printer  enscript -r -f Courier7

Run 10      Event 1
MET 37.5 GeV  MET phi 2.535      Sum ET 340.8 GeV      MET significance 2.03 (GeV)^1/2
    
```

Jets						
Et	eta	phi	ECAL	HCAL	emf	
130.7	-2.477	-2.713	461.9	321.7	0.589	0
78.8	0.432	0.156	41.0	45.2	0.476	0
63.7	-0.157	-0.465	39.8	24.7	0.618	0
55.0	0.156	1.166	47.9	7.8	0.860	0
12.5	0.341	2.293	13.2	0.0	1.000	0
5.3	-0.741	-1.278	6.8	0.0	1.000	0
1.9	0.331	-1.264	2.0	0.0	1.000	0

```

Dump to file event_dump.txt  append to file  Dump to terminal  Dump to printer  enscript -r -f Courier7

Run 10      Event 1
MET 37.5 GeV  MET phi 2.535      Sum ET 340.8 GeV      MET significance 2.03 (GeV)^1/2
    
```

L1 muons			L1 EM objects			L1 jets		
Et	eta	phi	Et	eta	phi	Et	eta	phi
-----	-----	-----	-----	-----	-----	-----	-----	-----

Et	eta	phi	E/p	H/E	fbrem	Electr	dei
18.1	0.197	1.200	1.485	0.000	0.011	-0.002	
14.0	0.148	1.082	2.881	0.000	0.925	-0.002	
14.0	0.153	1.186	1.723	0.000	-0.074	-0.005	
78.0	-2.479	-2.734	5.439	0.058	0.806	-0.011	
77.4	-2.486	-2.684	13.042	0.058	0.144	-0.001	
77.4	-2.486	-2.745	3.405	0.058	0.051	-0.006	
75.5	-2.512	-2.781	26.679	0.058	0.416	0.009	
71.7	-2.528	-2.779	23.161	0.058	1.311	0.045	

```

Dump to file event_dump.txt  append to file  Dump to terminal  Dump to printer  enscript -r -f Courier7

Run 10      Event 1
MET 37.5 GeV  MET phi 2.535      Sum ET 340.8 GeV      MET significance 2.03 (GeV)^1/2
    
```

Tracks													
pt	eta	phi	d0	d0 err	z0	z0 err	vtx x	vtx y	vtx z	pix layers	strip layers	outermost layer	
34.1	0.425	0.171	0.010	0.001	3.665	0.002	0.033	-0.005	3.679	2	13	125 61	
21.9	-2.518	-2.724	-0.047	0.002	3.419	0.011	0.046	-0.031	3.601	3	3	125 13	
21.4	-2.486	-2.745	-0.012	0.002	3.486	0.009	0.032	0.000	3.663	3	9	125 20	
20.0	-0.180	-0.445	-0.015	0.001	3.688	0.003	0.033	0.001	3.683	3	16	125 31	
18.8	-0.205	-0.489	-0.015	0.001	3.683	0.004	0.032	0.000	3.677	3	14	125 25	
18.2	-0.170	-0.419	-0.012	0.001	3.689	0.003	0.032	-0.001	3.684	3	13	125 17	
15.6	-2.514	-2.749	-0.009	0.002	3.429	0.010	0.031	0.003	3.611	3	7	125 11	
12.2	0.197	1.200	0.040	0.002	3.668	0.001	0.042	-0.004	3.670	4	15	125 36	
9.3	0.090	1.212	0.052	0.002	3.687	0.006	0.053	-0.008	3.688	3	2	125 9	
8.4	0.187	1.161	0.035	0.002	3.681	0.002	0.037	-0.002	3.683	3	15	125 31	
8.0	-0.217	-0.456	-0.016	0.002	3.691	0.002	0.033	0.001	3.685	4	15	125 25	
8.0	-2.519	-2.713	-0.027	0.013	3.494	0.057	0.038	-0.013	3.675	1	7	125 9	
6.0	0.472	0.227	0.008	0.002	3.666	0.002	0.032	-0.000	3.681	3	14	125 31	
5.7	0.144	1.200	0.038	0.007	3.677	0.005	0.039	-0.003	3.679	3	4	125 21	

pt	global	tk	SA	calo	iso(3)	iso(5)	tr	pt	eta	phi
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Vertices									
index	vx	vx err	vy	vy err	vz	vz err	tracks	chi2	ndof
0	0.035	0.001	-0.001	0.001	3.683	0.001	57 66.920	105.480	

- All columns are sort-able
- Objects can be selected and selection propagates to all other views



Plug-in System



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- **Plug-in's** allow users to write their own code to visualize what they need.
- **Input** is a vector of objects, called a collection, which represents a typical data unit in CMS.
- **Output** is a ROOT Eve container of graphical primitives that ROOT knows how to render.
- **User contribution** is the code that translates the CMS data object to a set of graphical primitives. Ideal geometry is available for most detectors.
- The plug-in's can be compiled without touching the main code with no exposure to internals of the display.



Remote Data Access



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- Data access is not a problem specific to event displays, but nevertheless ignoring it may render even the best display useless.
- CMS uses distributed storage model with data distributed across the world.
- We are trying to make sure that users will get access to events that they need.
- **"Pick Event"** - service to find and extract a few events with known identifiers like run and event number.
 - Typical use case: found an interesting event in ntuples and want to look at it in details.
- **"Random Access"** - access to any event within reasonable amount of time anywhere in the world. The main reason why it can be done is that in principle we don't need that much information and requests are rare.



Conclusion



List View



Views

Physics objects

Triggers

Tracking

3D



Introduction

Design Choices

- Staying focused
- Direct Data Access
- ROOT – Eve
- ROOT – Reflex
- Usability
- Summary

Display Overview

- Main View
- Rho-Z View
- Rho-Phi View
- Lego View
- Text Views

User Support

- Plug-in system
- Remote Data Access

Conclusion

- CMS developed a new light-weight event display - Fireworks
- Direct participation of ROOT developers in the project allowed for quick implementation and integration of new ideas.
- Fireworks became a working tool for many users at CMS. We get lots of positive feedback and it becomes normal to see Fireworks pictures in presentations.
- Many things still can be done and many features implemented, but the main goal of having a useful tool for a physicist to analyze data is achieved.