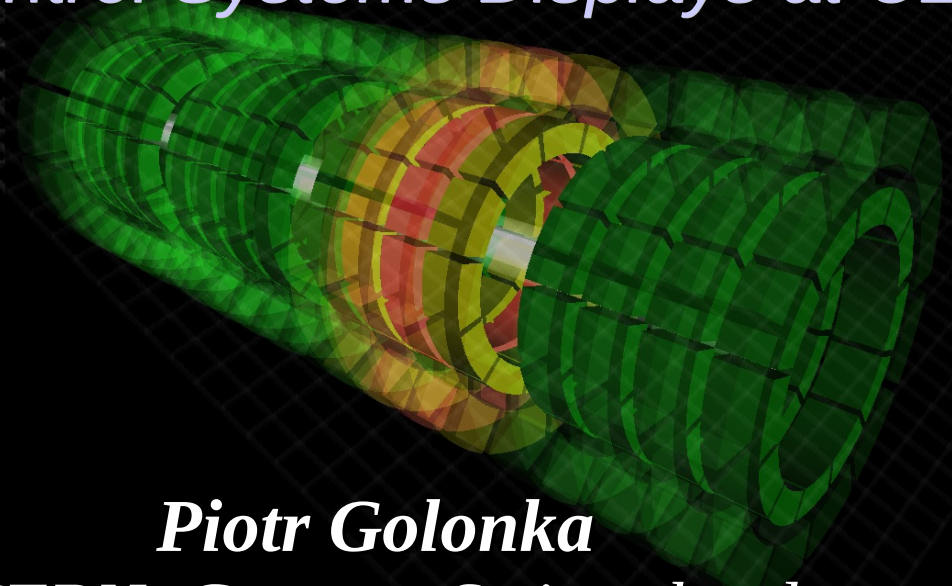
Towards 3D Human Machine Interfaces

Generic 3D Viewer Extension for the Control Systems Displays at CERN



Piotr Golonka
CERN, Geneva, Switzerland

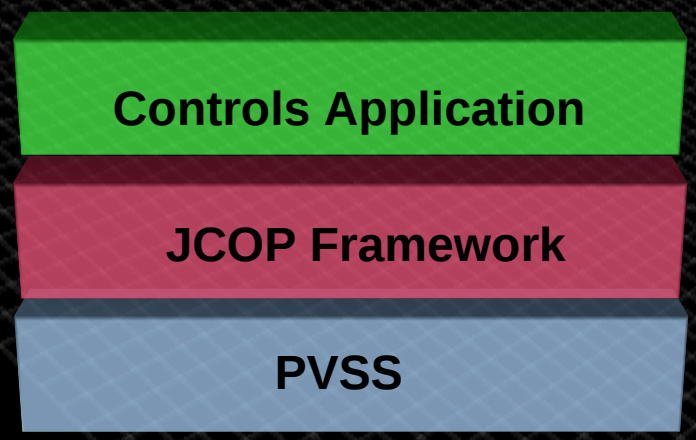
ICALEPCS Conference, Kobe, Japan, October 2009

3D Viewer

- What it is and why (motivation)
  - What makes it different
- Application examples
  - ... and yet more examples
- Showcase for
  - Technology integration
  - Portability
  - Code reuse

Context

- HMI of many Control Systems at CERN

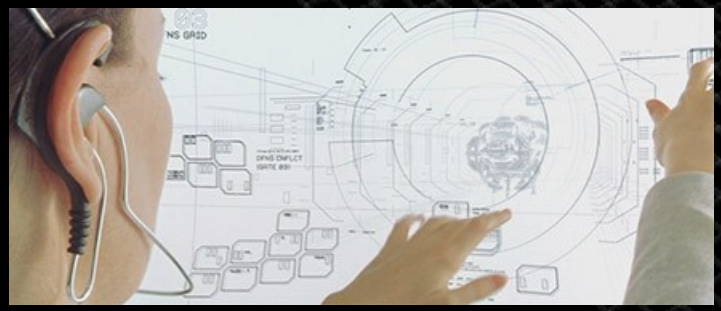


Commonly Used & Developed
eg. Access Control, ConfigurationDB, CAEN Power Supply Integration,...

Commercial SCADA Tool

- User Interface Manager a part of PVSS
  - Graphical design tool for PVSS panels
    - Control widgets: buttons, text boxes, tables, combo boxes
    - Data visualization: trends, charts, histograms
  - Complemented with scripting language (CTRL)
- End result: classical operators' HMIs

- Hollywood concept of a control room
  - Luckily it's not them who decide...
  - But maybe we could learn ...



3D SCADA?

- 3D commercial SCADA already available...
- Serious use case studies (SAP AG)

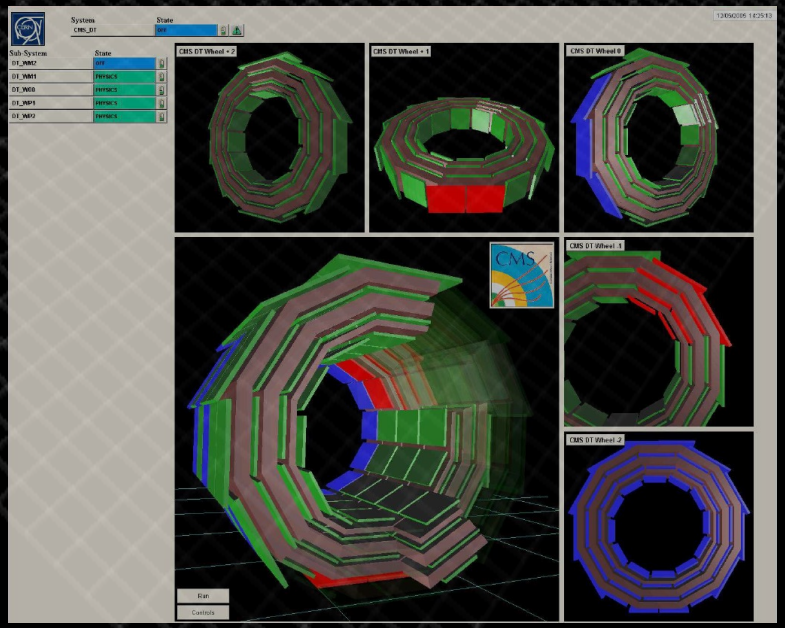
  ”In manufacturing, 3D visualisation has not been used to its full potential.(...) While 3D visualisation holds a lot of promises it is not always a cost-effective or ideal solution. (...) Also, while certain use cases are frequently discussed as a potential uses for 3D they may not be appropriate for the task at hand. (...)”

  [“Use Cases and Concepts for 3D Visualisation in Manufacturing”, Bernhard Wolf, Gerald Mofor, Jochen Rode, SAP AG]

  - Communication and demonstration
  - Plant and process engineering
  - Training
  - Localisation: locate hot spots for intervention
  - **Monitoring and control**
  - Maintenance and repair (visualize instructions)
- Become interesting where large scale come to play

3D Detector View

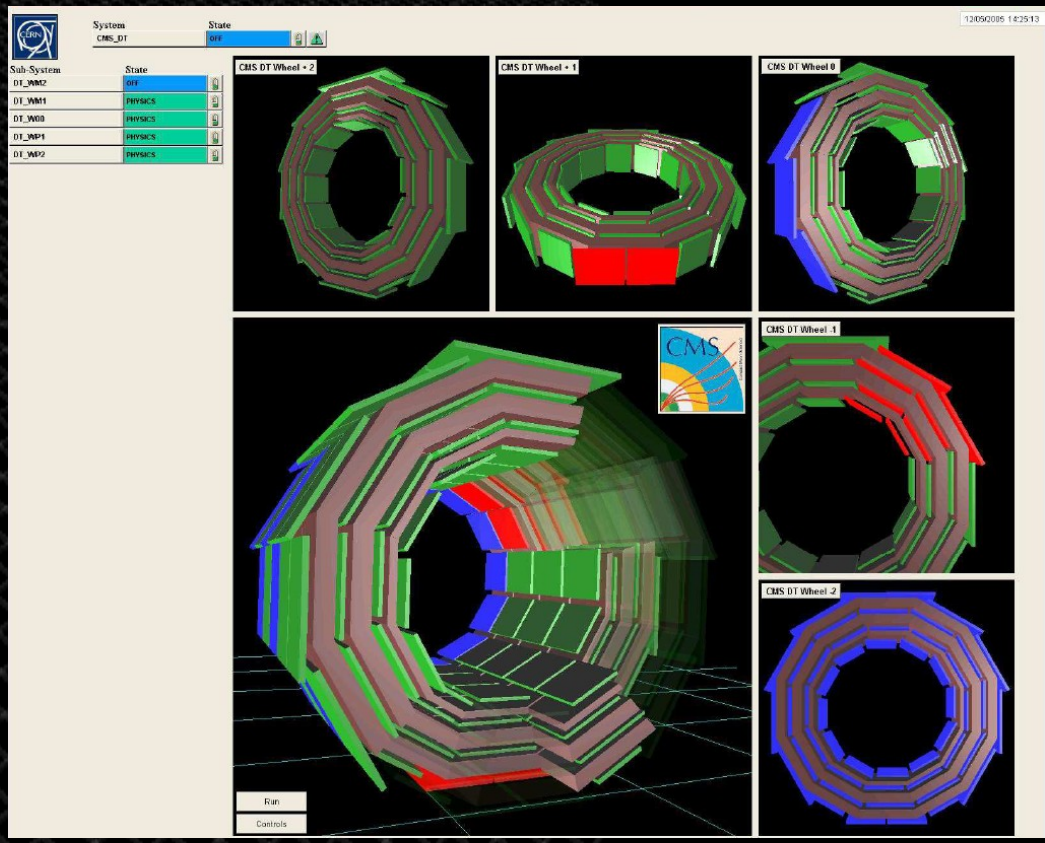
- Pioneer work:
  - 3D visualization for the Detector Control System of the CMS Detector at CERN
  - Robert Gomez-Reino, CMS Central DCS Team



  - Motivation:
    - *Geometrical alerts*:
      display of spatial correlation for parts of the detector deserving attention, easy and intuitive navigation

3D Detector View

- Replacement for classical synoptic view
  - Geometrical view of detector subsystems
    - Familiar to shifter or expert
  - Colors used to display the states
  - Spatial correlation of events/alarms/states
  - Observe details, while keeping general context, orientation, geography
  - Intuitively navigate to desired part using mouse, click on the object for which you want more details



3D Detector View

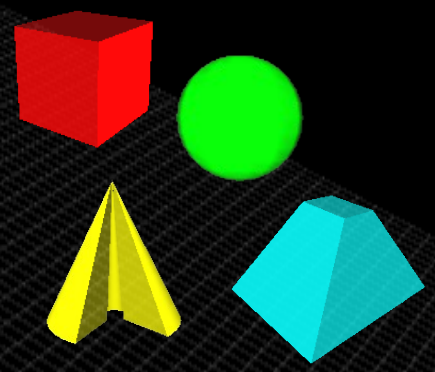
- Prototype implementation
  - Java3D, within an ActiveX allows to embed it within PVSS
  - Windows platform only...

  - Showstopper for its wider adoption and use in the other experiments' Control Rooms
  - Maintenance?

JCOP Framework 3D Viewer

- New implementation of the concept
  - Functionality of the existing prototype: replacement
  - Portable
  - Longevity and maintenance
    - Hosted as one of the JCOP Framework components
    - Use of standard technologies
  - Integrated with native PVSS UI technologies
    - EWO: Enhanced Widget Object
    - Controllable, scriptable
  - Generic, High level

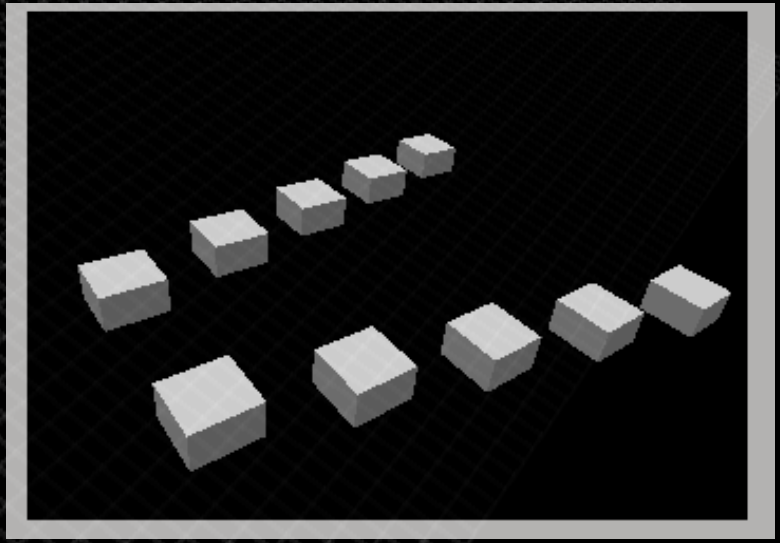
Fw3DViewer: generic approach



- Programmer constructs a “scene”
  - A set of typically used shape types
  - Instances: shapes
    - added/removed dynamically
    - referred by a unique name
  - Shape have properties
    - Geometry, position, rotation
    - Color and transparency
  - 3D Viewer widget displays the scene,
    - Navigate, interact with shapes (by clicking them)
  - Integration with native PVSS mechanism
    - GEDI, scripting, properties, shape-selection events

Ad hoc geometry scripting

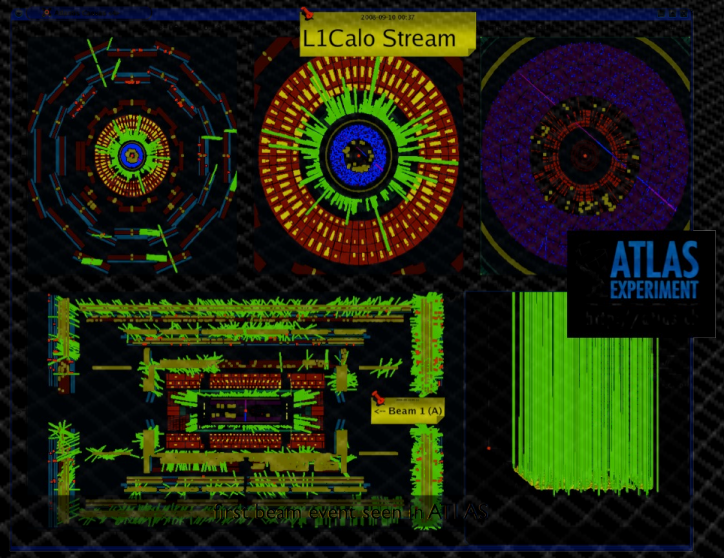
```
main()
{
  for (int stationA=1;stationA<=5;stationA++) {
    this.addShapeInGroup(
        "SubDetectorA","BOX", "SubDetA_station"+stationA,
        makeDynString("x",(2*stationA),
                      "y",(1.0+0.5*stationA),
                      "z",0.0,
                      "dx",0.5,
                      "dy",0.5,
                      "dz",0.3));

  }

  for (int stationB=1;stationB<=5;stationB++) {
    this.addShapeInGroup(
        "SubDetectorA","BOX", "SubDetB_station"+stationB,
        makeDynString("x",(2*stationB),
                      "y",(-1.0-0.5*stationB),
                      "z",0.0,
                      "dx",0.5,
                      "dy",0.5,
                      "dz",0.3));

  }
}
```

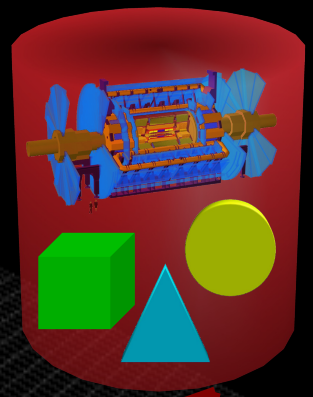


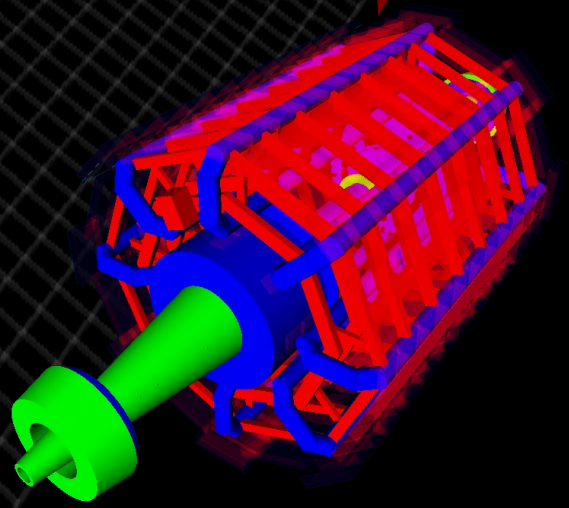
- Object names under strict control
- No unnecessary detail
- Variant:
  - extract data from a XML file, reprocess in the script
  - ... or from a database...

Geometry data

- Geometry Database
  - used by off-line data analysis (reconstruction) software and event displays

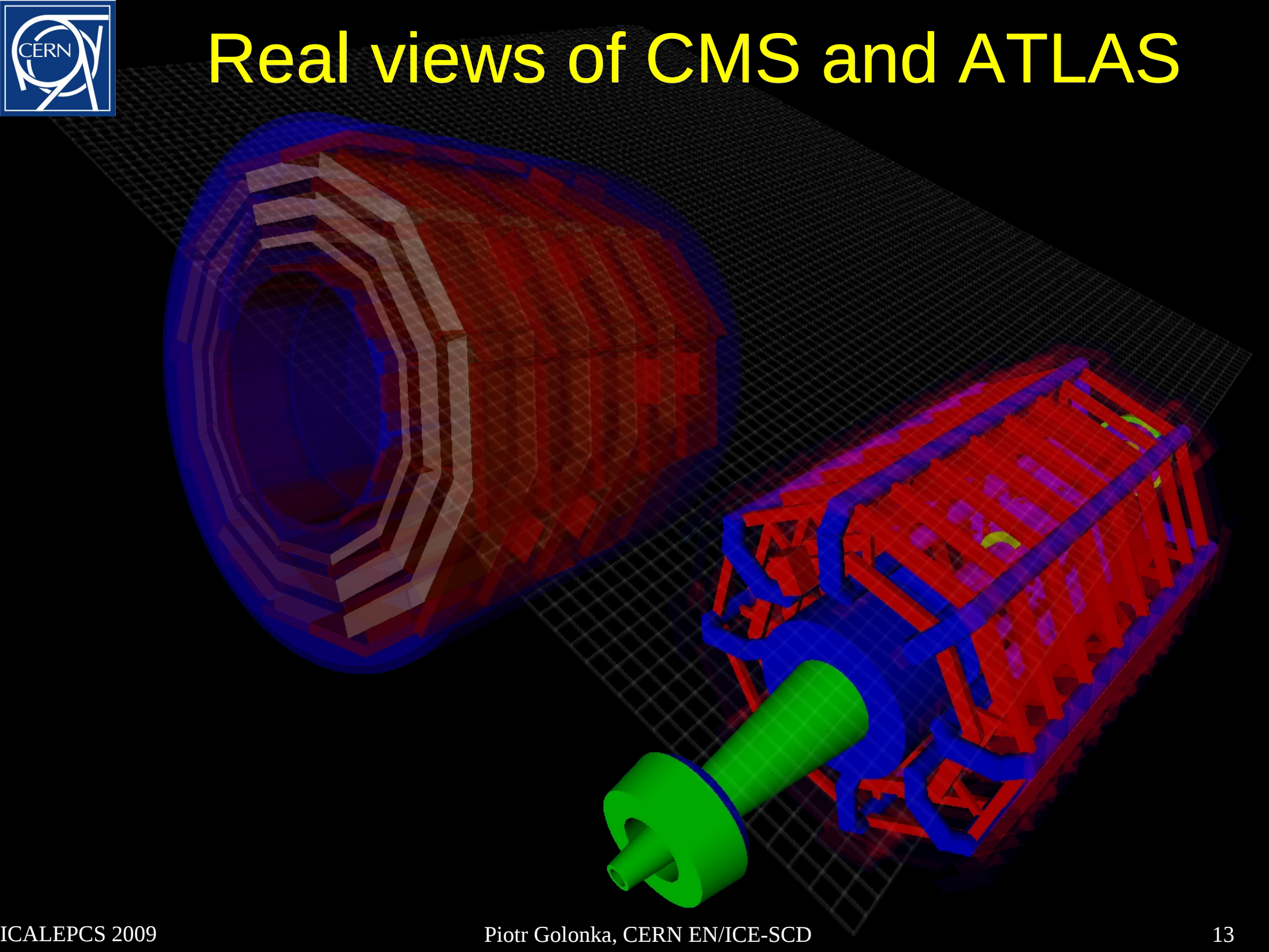




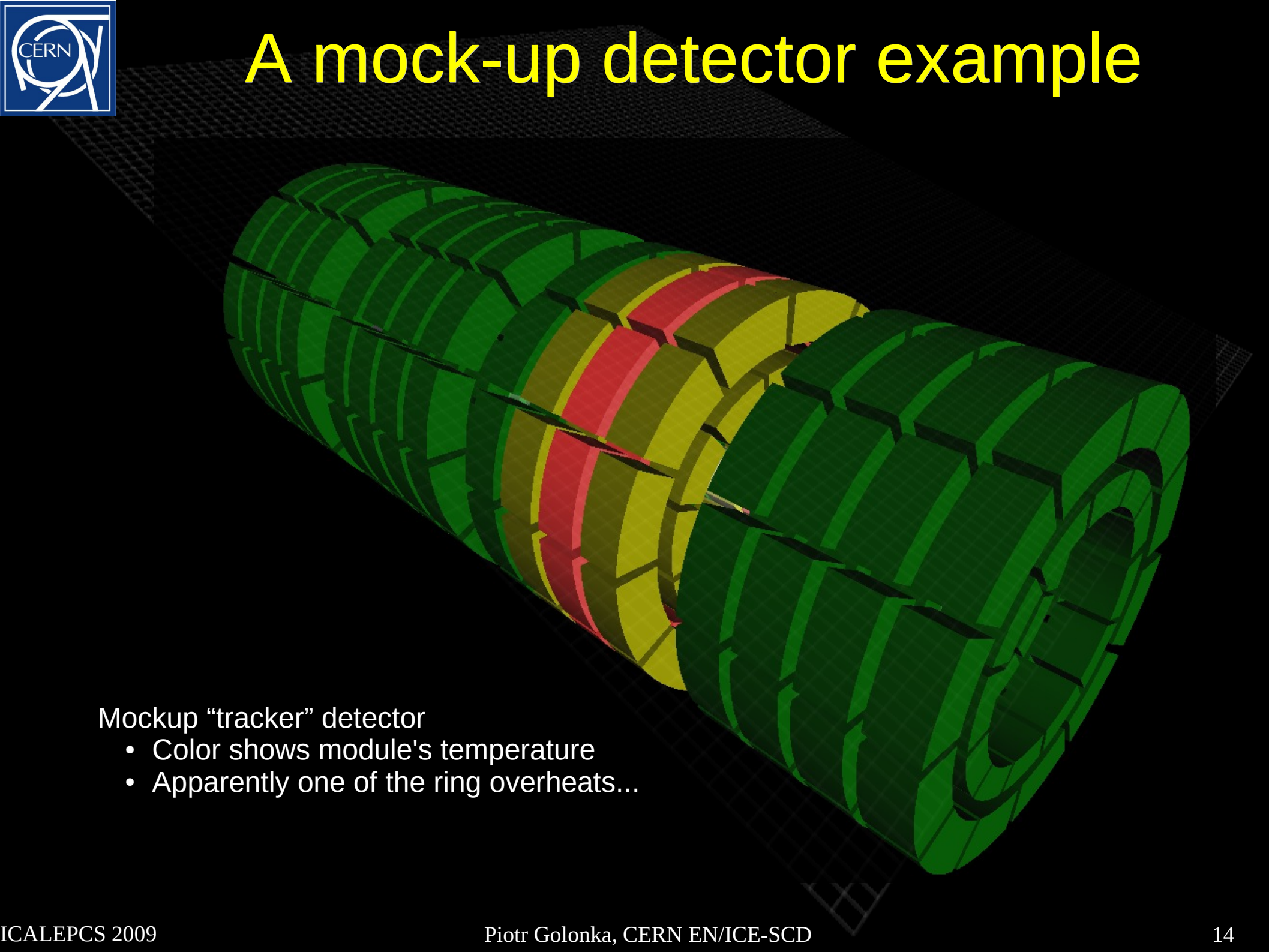


  - how about we make use of it?
    - Need binding with controls-related entities

Real views of CMS and ATLAS



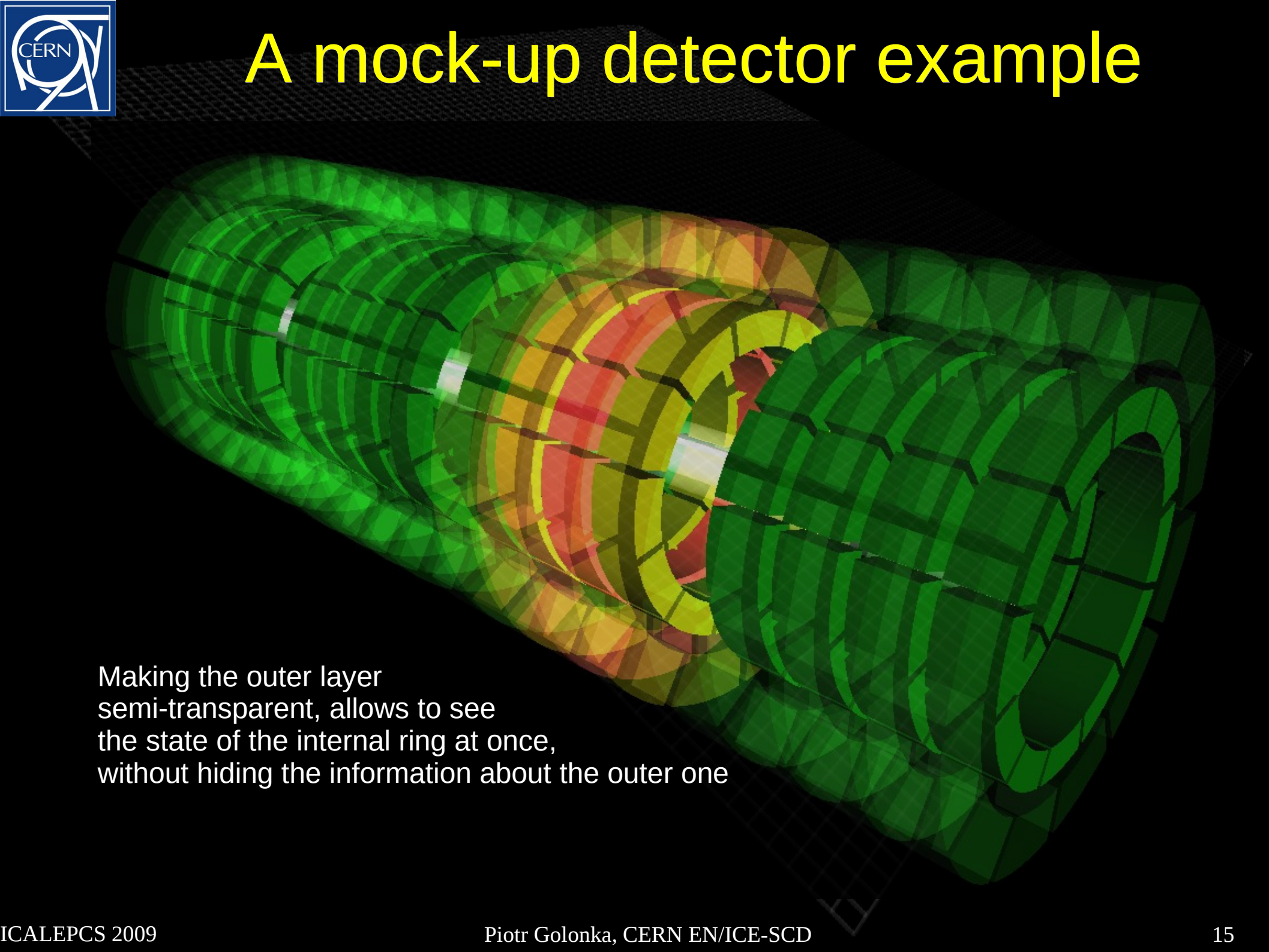
A mock-up detector example



Mockup “tracker” detector

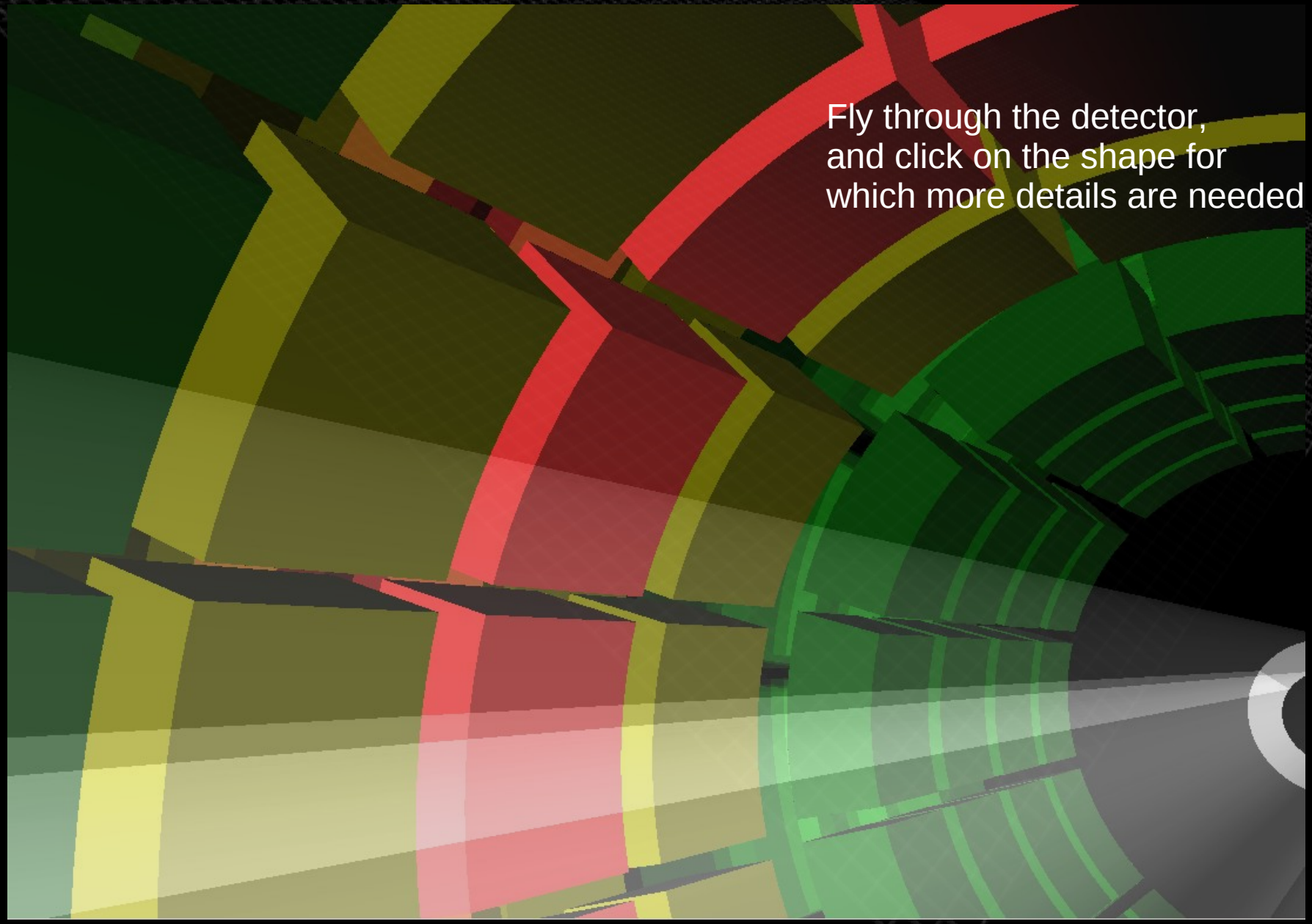
- Color shows module's temperature
- Apparently one of the ring overheats...

A mock-up detector example

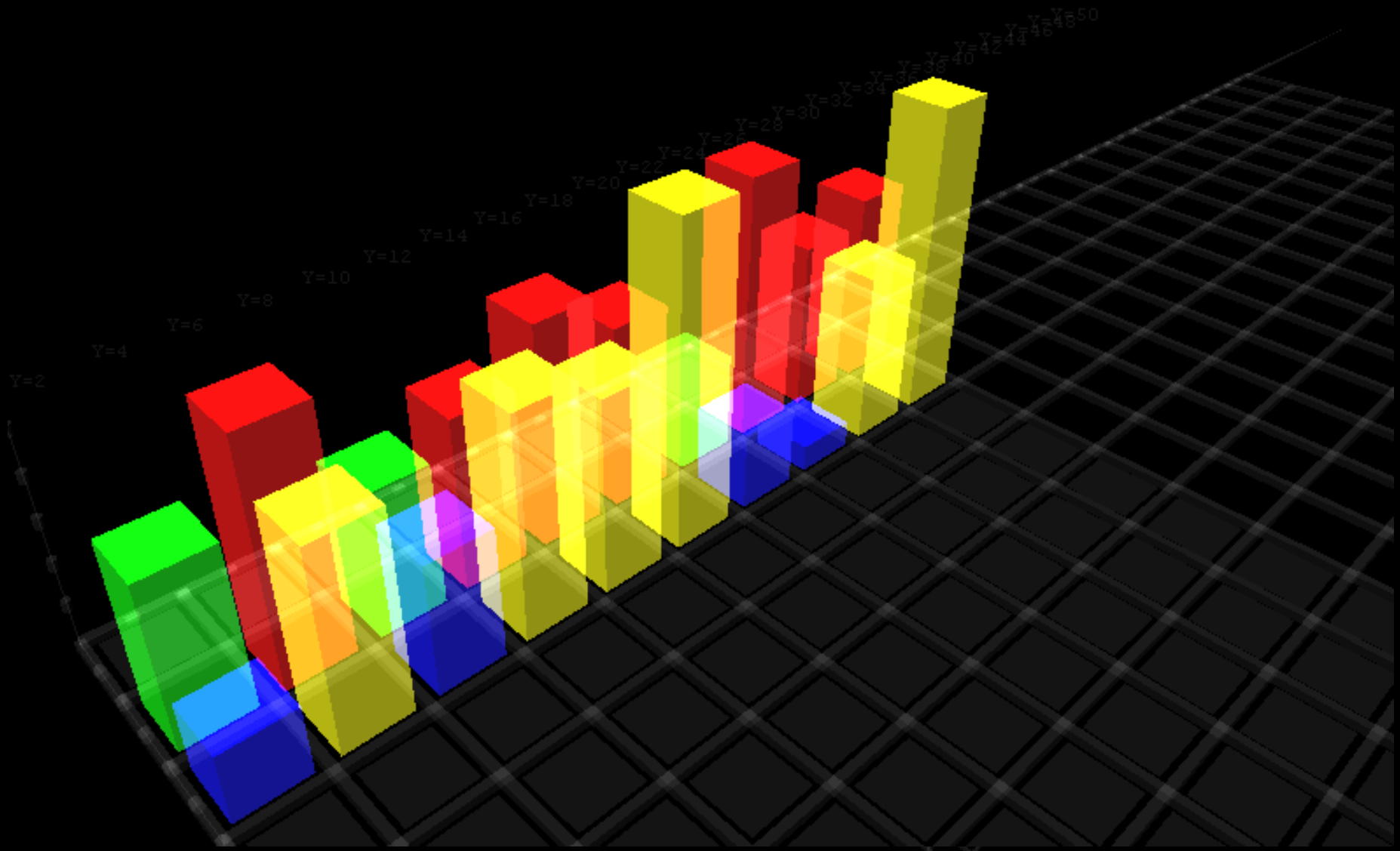


Making the outer layer
semi-transparent, allows to see
the state of the internal ring at once,
without hiding the information about the outer one

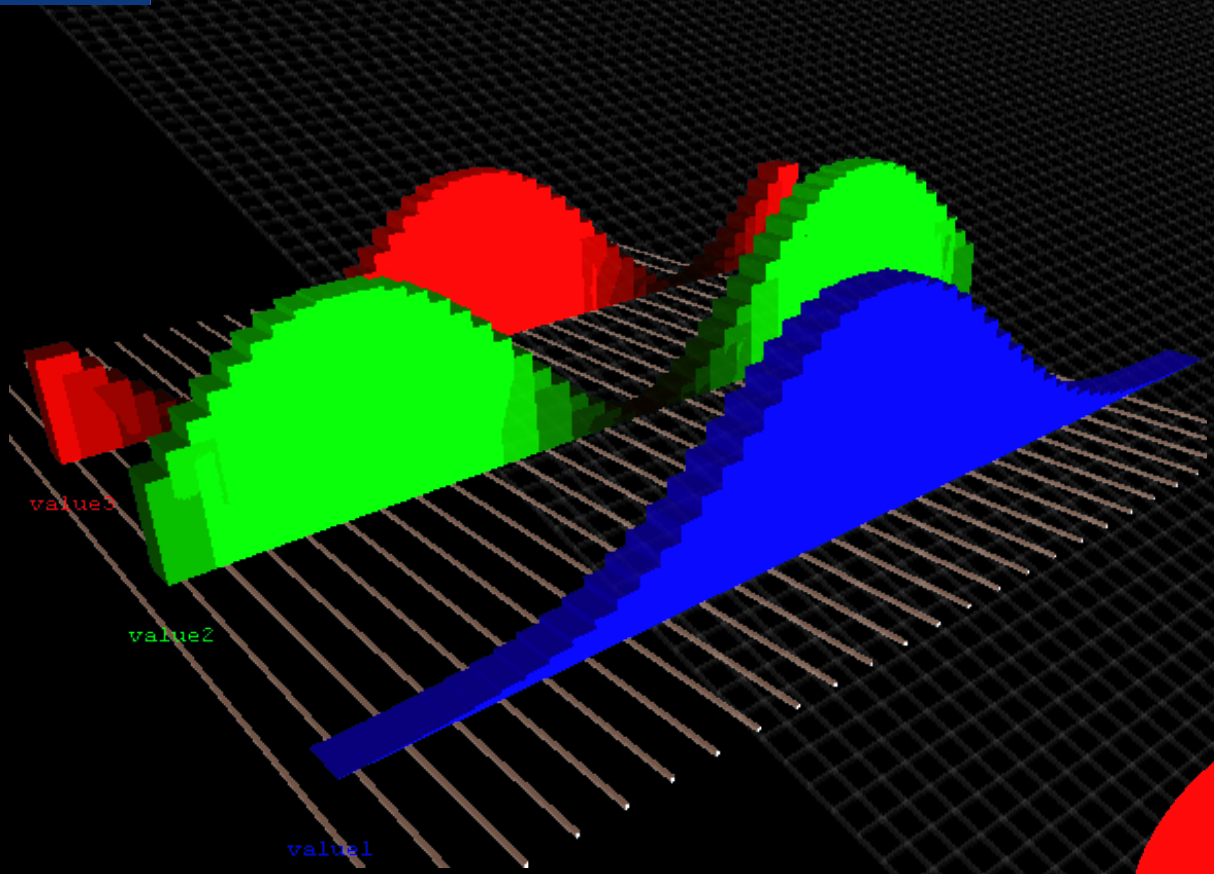
A mock-up detector example

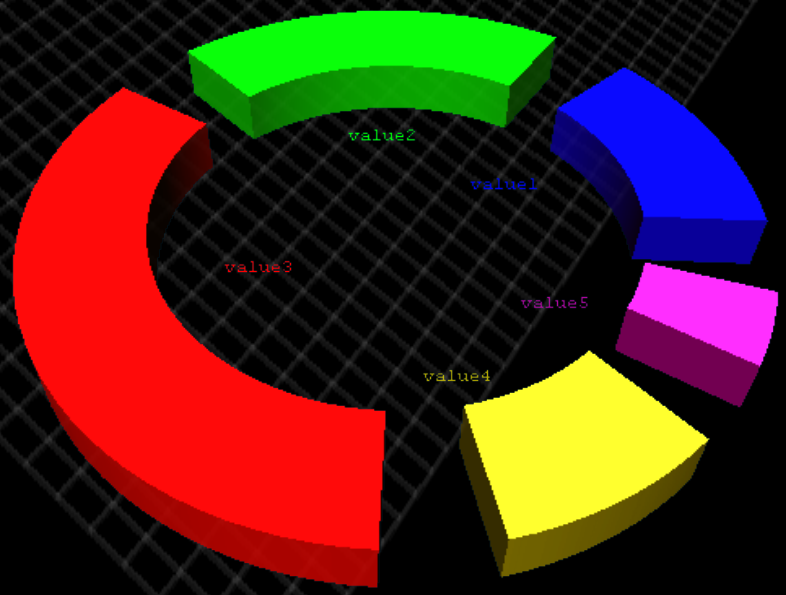


Other applications: a 2D live histogram...



Trend plots, charts





Technologies

Technologies: Qt

- C++ toolkit for GUI & application development
  - Multi-platform (Linux, Unix, Windows, MacOS, embedded platforms)
  - Single-source portability made real
  - Developed by NOKIA (ex: Trolltech)
    - Your next Nokia smart-phone will likely feature it...
  - Open Source and commercial licenses
  - Huge open source community (KDE)
  - Picked up by PVSS since version 3.5
    - Enhanced Widget Object (EWO)
      native UI plugin mechanism

Technologies: Open Inventor

- Open Inventor: high level, 3D programming API
  - Designed and implemented by sgi in 90's
  - Aim: make graphics programming more efficient
    - A higher-level C++ layer on top of OpenGL
    - Retained mode:
      - the client interacts with a model built of geometrical shapes;
      - Open Inventor engine renders the scene with OpenGL, applying optimizations such as hidden surface removals
    - Simplifies code, esp. for non-OpenGL experts
    - Delivers a library of objects
      - Geometrical shapes: spheres, boxes, etc
      - Cameras, lights

Technologies: Open Inventor

- Multi-platform
  - Unix/Linux, Windows, MacOS, ...
- Interfaces
  - Native platform bindings
  - Qt
- Extendability: HEPVis
  - Shape types used in HEP software, such as GEANT, to describe geometry:
    - polyhedras, trapezoids,...

Technologies: Open Inventor

- Availability
  - Open Source (since 2000)
  - Visualizations Sciences Group (since 2009)
    - Commercial edition, support
    - Further development of the standard
    - Numerous Extensions
    - De facto standard for many industrial and scientific visualization
  - Coin3D: independent implementation of OpenInventor 2.1 API
    - Open Source and Commercial license

Technologies: Integrate

JCOP Framework
3D Viewer

Qt

EWO | **Open Inventor**

PVSS | **HEPVis**

fw3DViewer: ~ 40kB of C++ code that integrates and interfaces huge codes of external libraries, packed with functionality

Summary

- 3D Viewer: extension for PVSS-based HMI
  - Generic, fully programmable
  - Based on strong standard technology
  - Implementations of synoptic views
    - Subdetectors of CMS and ATLAS
    - Others may follow
    - LHC (?)
  - The range of possible use cases is open
    - Data viewers
    - Custom, generic widgets