https://atlas-vp1.web.cern.ch/

"VP1" (ATLAS)

Riccardo Maria BIANCHI (Pittsburgh)
HSF CWP Workshop - San Diego - 25 Jan 2017

VP1 is the ATLAS 3D event display; it is a general-purpose tool used in physics analysis, detector development, reconstruction and simulation checking and debugging, outreach

Current status:

Technologies:

- graphics engine: **Coin3D** (OpenGL) + **SoQt**
- GUI: **Qt**

Pros:

- *Architecture*:
  - Being integrated in the experiment framework, VP1 can access all ATLAS data
- *Graphics libraries:*
  - Coin3D is "***scene-graph***" based: easy to match tree-like GeoModel objects to Coin primitives

Cons:

- *Architecture*:
  - VP1 **needs the whole experiment framework** to work. Thus, it only runs on the **SLC6 platform**.
  - Very **slow** when **run remotely** through SSH because of 3D data being sent through the network.
  - It is **widely used by experts**, but **less by physics analysis end users**, because most of them they do not know the framework and they only run the analysis on ROOT ntuples on their laptops
- *Graphics libraries:*
  - Coin3D and SoQt are **not supported and maintained anymore**. Also, they are very old and sometimes they show issues with modern compilers
  - Coin3D has **known issues** with transparency and it does not exploit modern "tricks"

Event data picking:

- "Event picking" is now a burden for people who want to make displays for a specific event: it is a slow and many-steps process, requiring **framework knowledge**

What we would like:

Technologies:

- An event display tool should be **easy to install** and **use** by (analysis) end-users
- It should be the **same tool** for experiment data (through the framework) and analysis objects (official ntuples)
- It should be built on top of **modern** and, most of all, **maintained libraries**
- The libraries should be easily integrated in existing code
- If "game engines", they should provide tools to **use external libraries** as well: for example, ROOT is needed to open the data files

Event data picking:

- An **easy way to get events** from the grid, in **different formats**, ready to be read in the visualization tool (an "event service" is under development, in ATLAS). Event picking should be easy for users

Geometry:

Geometry is **very similar** in all HEP experiments. Basically just a collection of primitives and coordinates, stored in a smart way to save memory.
Why not having a **common package and a common format?** Sharing the development and the tools among the experiments.