Full sim: status and perspectives

A. Di Simone
INFN Tor Vergata

Outline

- A few things were implemented in view of the next round of production
  - More precise tuning of the bgframes
  - More generator info in metadata
  - Geantinos in Guinea Pig
  - BFieldDumper
  - Radmon (other talk in this session)
  - Optical photons for FTOF (other talk in this session)
- In addition, there are a number of more general, medium-term, longstanding issues we need to start dealing with
  - Event display
  - Event structure
  - Runtime configuration
  - Geometry handling
- There are MANY MORE issues we should be working on, but one must be realistic, and with present manpower they'll have to wait

Tuning of bg frames

- Reminder: bgframes production is “normal” full simulation job, with a few modifications
  - A scoring volume is defined, mostly coinciding with the final_focus
  - At the exit of the scoring volume, all particles BUT neutrons are killed
  - Electrons and photons @ the boundary are saved to file, to be passed to fast-sim
  - Neutrons keep being propagated in the detectors, and their interactions are saved to file to be fed to fast-sim
- Some energy cuts are applied on particles when they are written to file
  - Motivation was that in the fast-sim scale of times, reading from file a large number of entries is a significant overhead
  - Previous cuts were chosen (hard coded) about 2 years ago; now the request was made to better tune them
  - Implemented configurability at runtime of the cuts, with additional macro commands
    - Now committed, under testing
  - This requires a special .mac file to be provided when generating bgframes (**relevant for production system**)
    - BrunoApp -m RadBhabha.Prod.mac -f **FIConfig.mac**

Generator Metadata

- Reminder: metadata are all configuration parameters of **a run**
  - do not change from event to event
- A lot a info is already saved by default
- Alejandro asked to include more details on the RadBhabha generator, like the bunch crossing frequency
- Required some code changes in Brn3BGen and BrnCore
  - Now committed and tested

```
OBJ: TParameter<double>        BbbremBXFreq@1e36       Named templated parameter type
OBJ: TParameter<double>        BbbremCutOff    Named templated parameter type
OBJ: TParameter<double>        BbbremLumiOverFc        Named templated parameter type
OBJ: TParameter<double>        BbbremMinDE     Named templated parameter type
OBJ: TParameter<double>        BbbremNbBhabha  Named templated parameter type
OBJ: TParameter<double>        BbbremPrescale  Named templated parameter type
OBJ: TParameter<double>        BbbremXSec      Named templated parameter type
```

Geantinos in Guinea pig

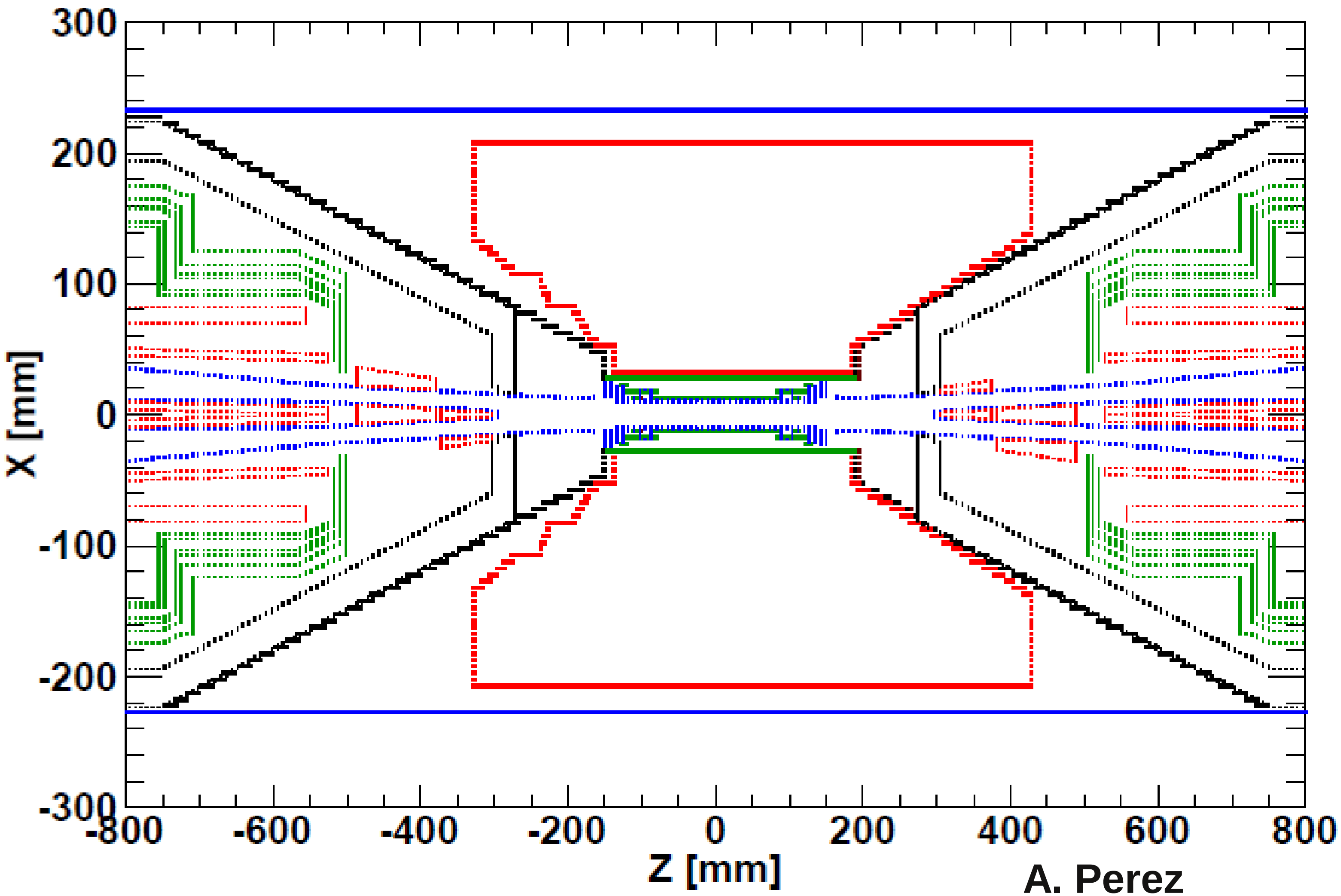
- It was not possible to use geantinos as inputs for guinea pig generator
- Modified the corresponding module (BrnCore/BrnG4HEPEvtInerface.cc) to include geantinos. Code has been committed
- Very useful tool to spot errors on geometries and mother boundaries. A very simple procedure
  - Instrument the volume boundary you want to look at in the usual way

```
/truth/create_boundary final_focus final_focus
/truth/boundary/final_focus/level 0
/truth/boundary/final_focus/trackPDG 0
```

- Sample your geometry: shoot geantinos with a very specified directions and locations to visualized Þ able to sample the geometry with very good precision
- Read your output file, look at the interesting branches and do the plot
- Note: same functionality already implemented using single particles generator. This adds the same to Guinea Pig

Geantinos in Guinea pig

- Useful to understand geometry implementations on a very crowded region



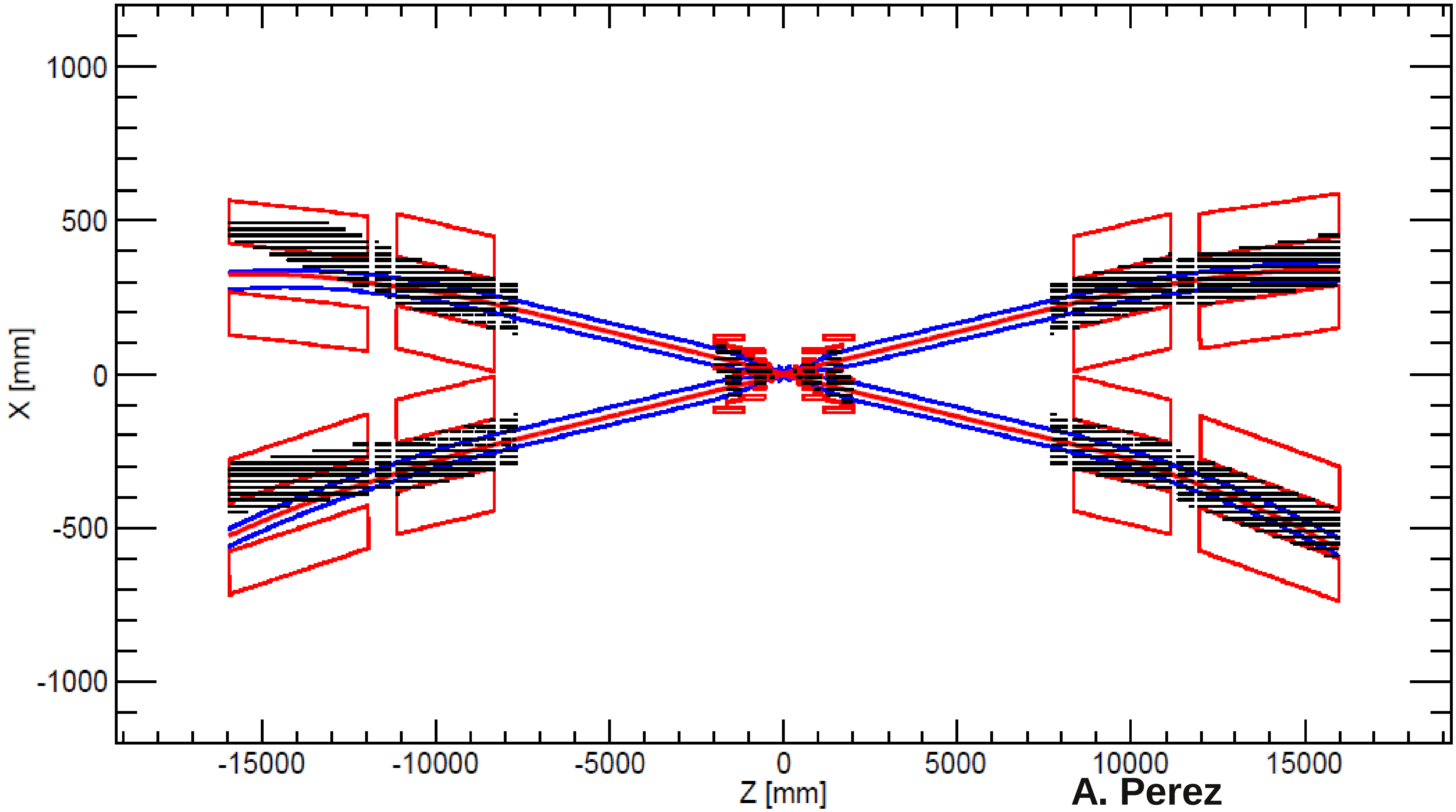
A. Perez

BFieldDumper

- BfieldDumper module previously present in unpackaged version of Bruno
  - Sampling of the magnetic field inside the final focus
- Output file is a root file with a TTree with the Bx,By,Bz for different coordinates (x,y,z)
- Very useful for spotting errors on the Magnetic model implementation
- Module was not migrated to the new BRN code
- BfieldDumper is now in BRN
  - BrnCore/src/BrnBFieldDumper.cc
  - Can it be moved to the more appropriate BrnGeoMag?
- Modified Bruno invocation to dump the B-field file: -B option, with no inputs. Very similar to Detector survey (-s option)
- Internal parameters hardcoded: sampling region and number of sampling points
  - Must provide runtime configurability

BFieldDumper

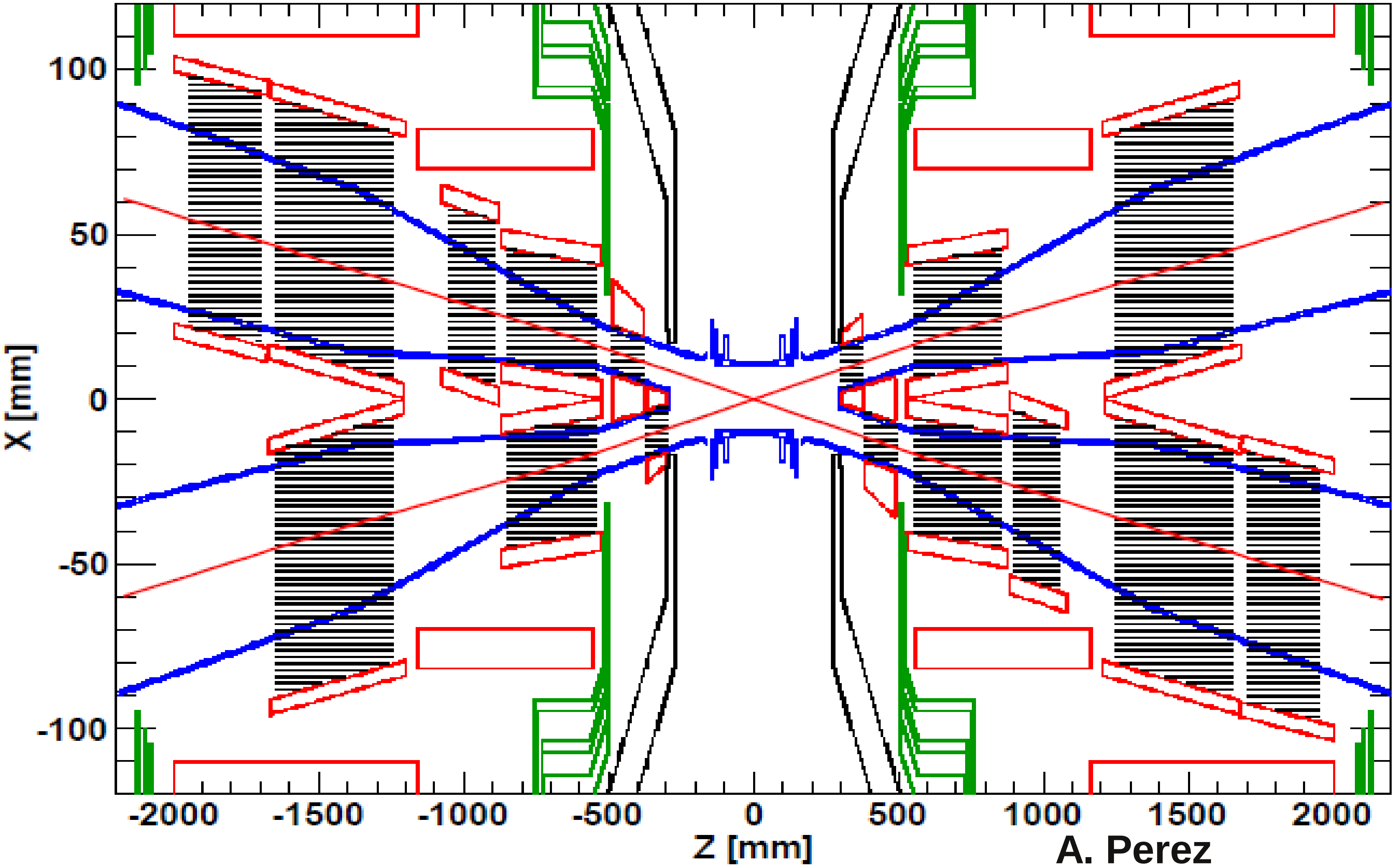
- Some problems on the final focus magnetic model were spotted



A. Perez

BFieldDumper

- The magnetic model problems has now been fixed

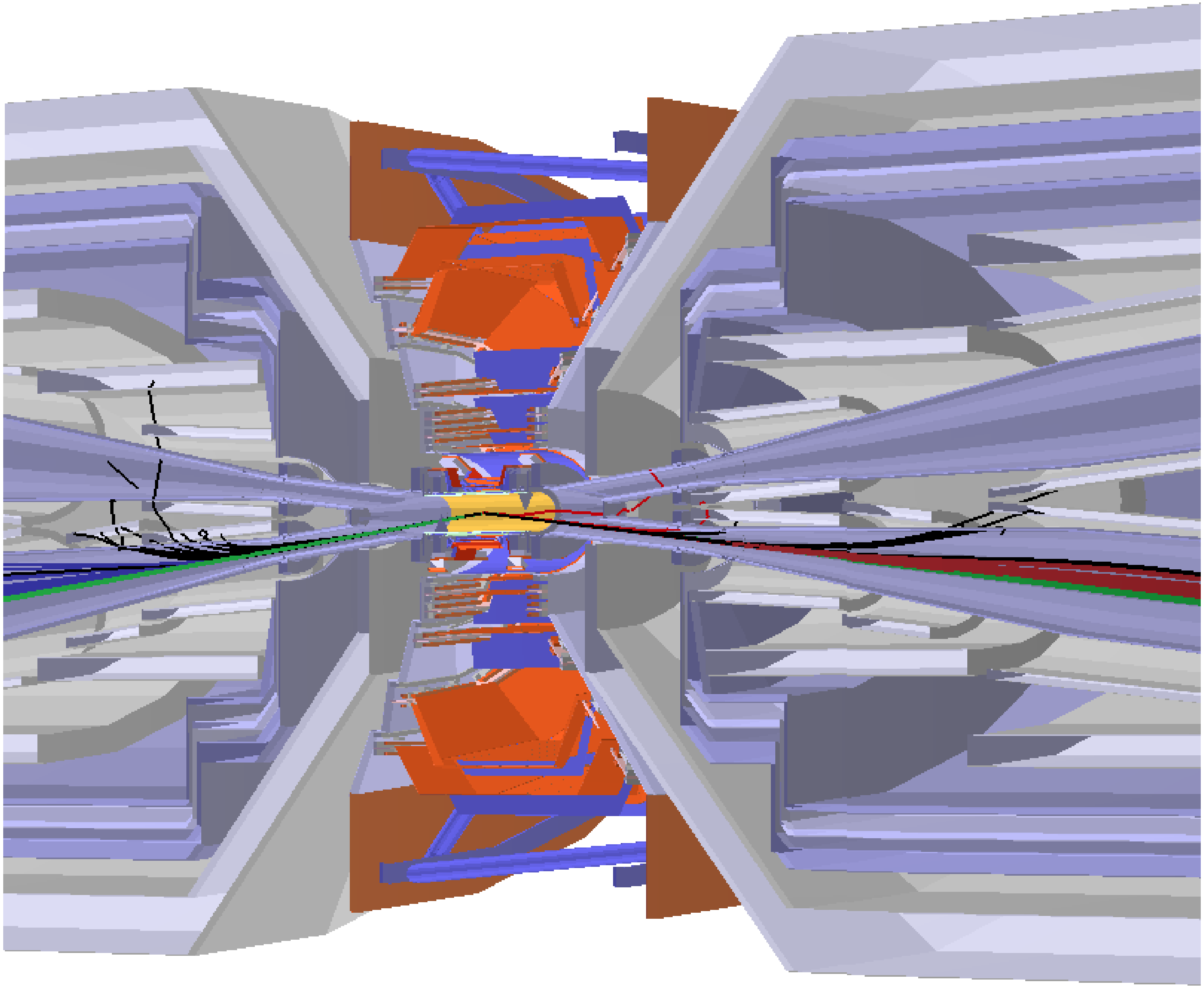


A. Perez

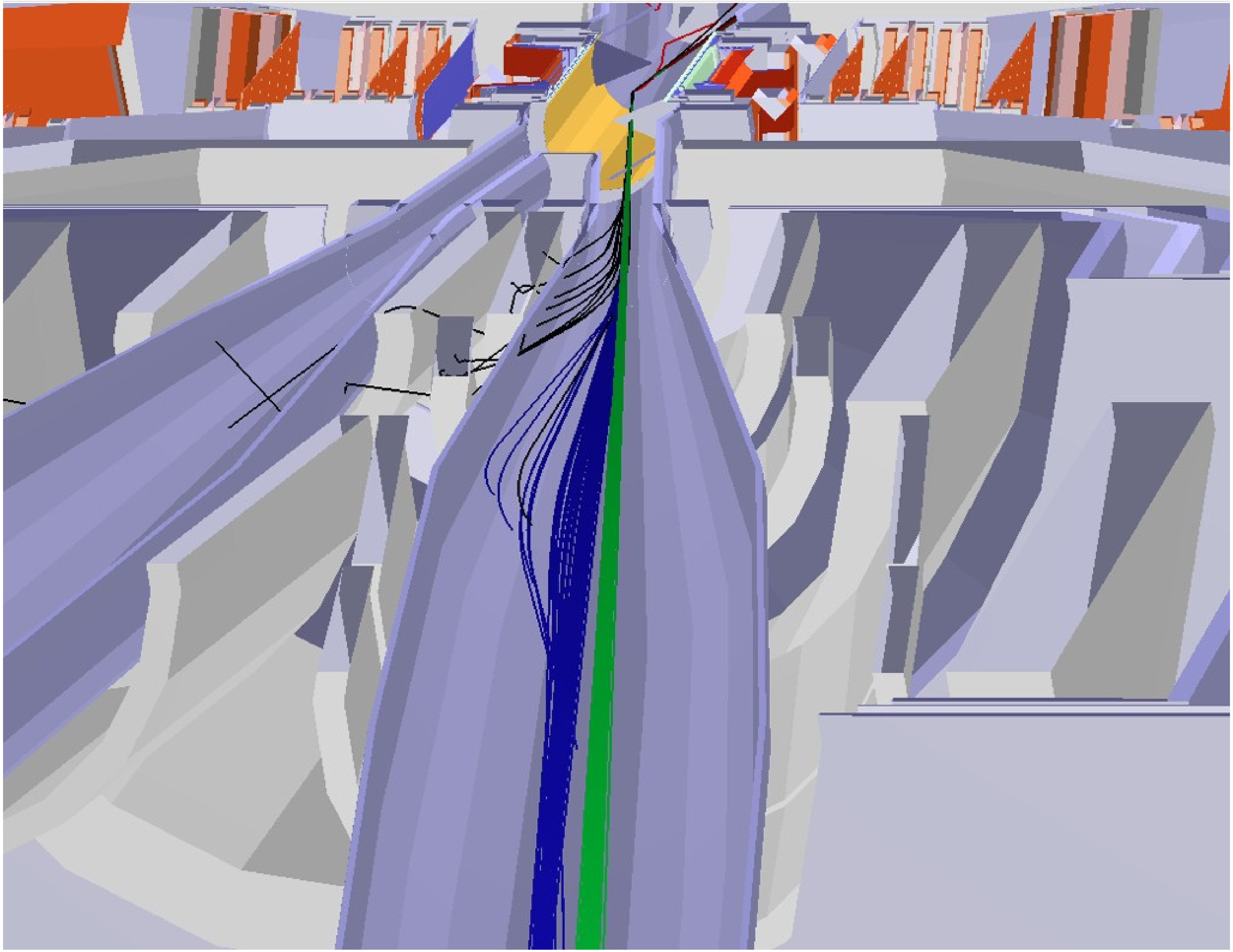
Event display

- All the truth information (including trajectories) was persistified using native ROOT classes
  - One of the reasons was to allow easy drawing within root
  - Never really exploited until the Vienna meeting, where Eugenio produced some astounding ROOT displays
    - See next slides

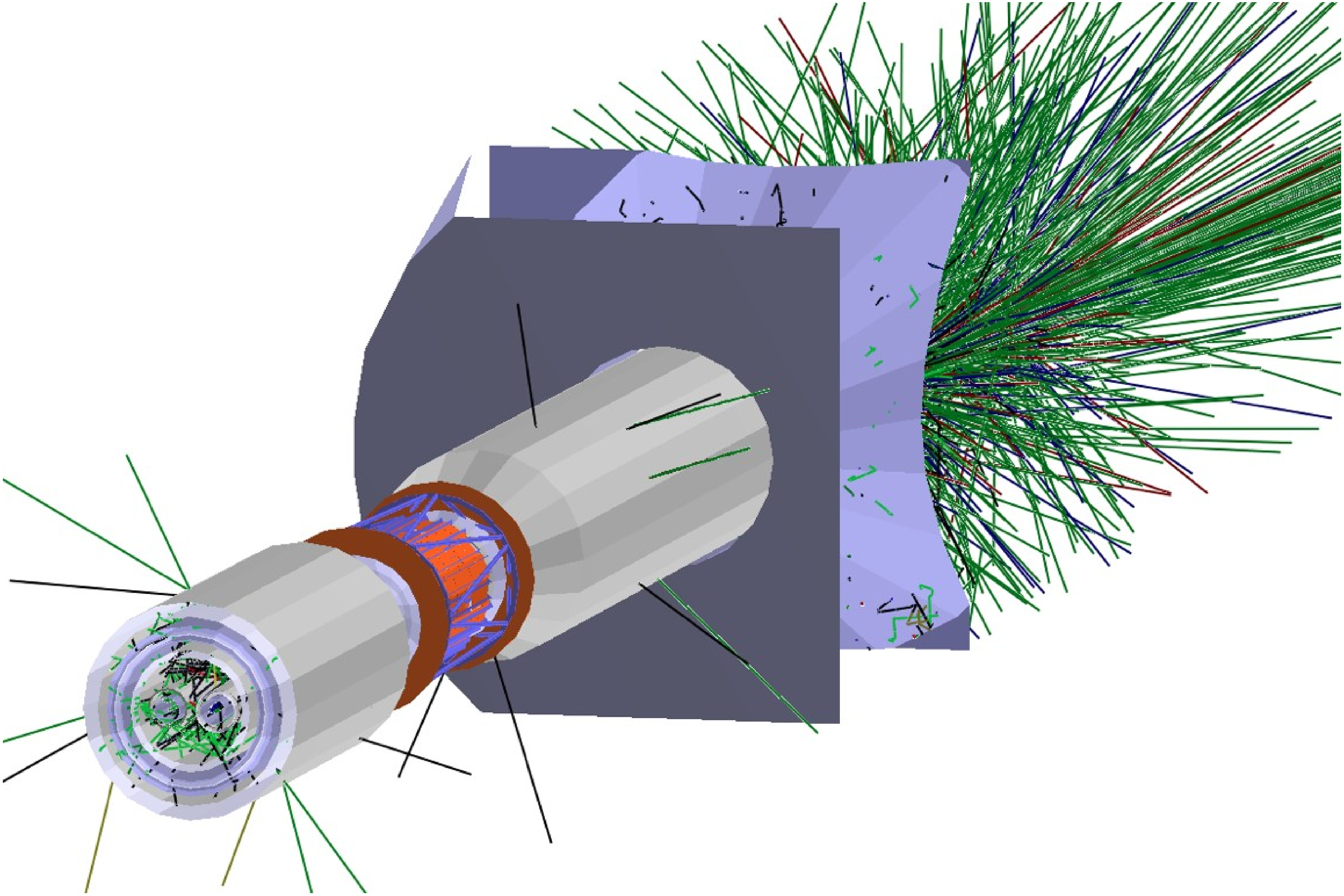
Event display



Event display



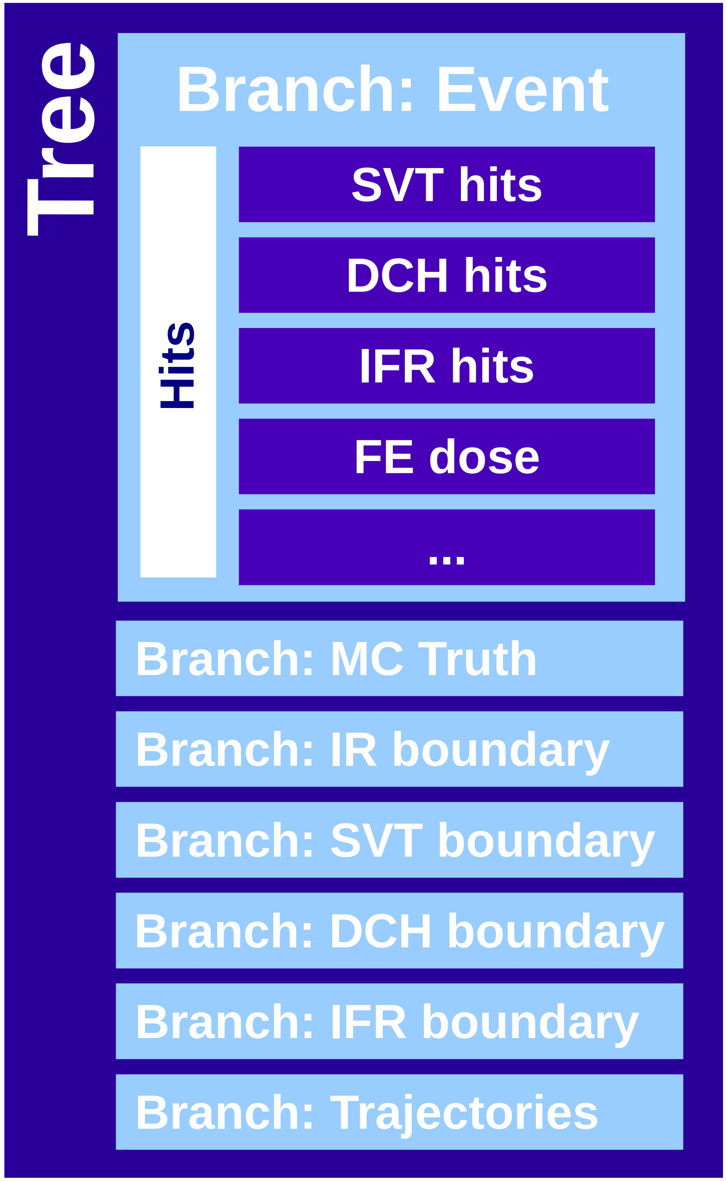
Event display



Event display

- Given the very nice performance of ROOT as event display, I propose we concentrate our very limited manpower in supporting it as the only "official" display
- Newcomers should be pushed towards it
  - Example macro already available in BrnRunTime
- A few nice side effects of this display:
  - It allows accelerated graphics, without adding GL dependencies on top of Bruno/G4
    - Safe also from the point of view of building/deploying
  - It does not require a special intermediate file to be produced by G4
    - All it needs is the "normal" hit file with truth information and the input gdml for the geometry
- Eventually, we could drop other visualization drivers altogether
  - Raytracer could probably be kept, as it is the only one to give you reliable visualization of all "exotic" volumes

Event structure



Tree

Branch: Event

Hits

SVT hits

DCH hits

IFR hits

FE dose

...

Branch: MC Truth

Branch: IR boundary

Branch: SVT boundary

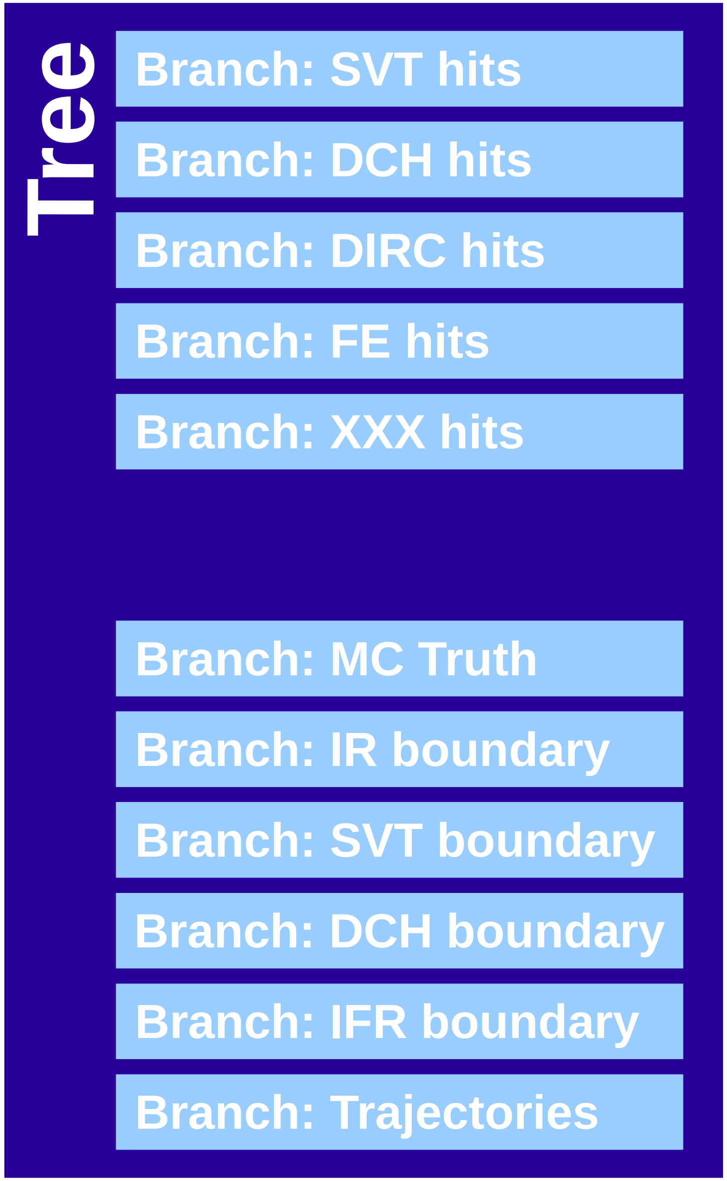
Branch: DCH boundary

Branch: IFR boundary

Branch: Trajectories

- Present event structure based on a monolithic event containing all the hits
- Truth information was split in separate branches some time ago
- Disadvantage of the monolithic event is mainly lack of flexibility
  - The Event class is where one loses modularity, since all detectors must be declaring their outputs to it
  - Adding a new detector requires (trivial, hence error prone) code changes in several places across three packages
    - BrnCore, BrnApp, BrnXXX

Event structure



Tree

Branch: SVT hits

Branch: DCH hits

Branch: DIRC hits

Branch: FE hits

Branch: XXX hits

Branch: MC Truth

Branch: IR boundary

Branch: SVT boundary

Branch: DCH boundary

Branch: IFR boundary

Branch: Trajectories

- My proposal is to push modularity to its full extent, and remove the global event altogether
  - NOT for this production
- Each detector will manage its hits and SensitiveDetectors with no couplings to other detectors
  - Drawback: this will break **all** existing analyses
    - Fixes should be trivial, though

Runtime configuration

- Really, we are reaching the limit of what one can do with .mac files
  - I know I have been claiming this since three years, but things are now **really** bad
  - We are not far away from the point where a key parameter may have to be hard coded because we just don't know how to access it at runtime
- I would like to resume experimenting with python-based configuration
  - My previous pyBruno tests were using pyBoost for dictionary generation
    - I propose now to benchmark how SWIG behaves
  - This requires, most likely, that we drop static linking in favor of shared libraries
    - I would need some help for the building/configuration
  - I also propose to delay all the static/dynamic linking debate until we have a working prototype and we can measure its performances
  - Discussion as CSG seemed to suggest that python is not favoured by the release-building team
    - On the other hand, no alternative was offered
    - No decision taken yet

Geometry handling

- Complaint by users about copy/replication of gdml files across geometry variants
- Reminder: present schema is that a geometry variant is a folder within BrnRunTime
  - Each folder must be self-contained (because of a gdml limit), hence one really has to copy files around
  - Each modification to the geometry (or a new geometry altogether) requires a new release of the whole Bruno code
- The duplication problem could be mitigated, as suggested by Eugenio, by using svn links
- Still, one can't use a given Bruno release to run on older versions of a geometry variant
  - Nor with newer geometry variants/versions
- We propose to decouple geometry releases from Bruno releases
  - Geometry becomes an external dependency, like many others
    - Just easier, because it has no code inside, only plain gdml files
  - From the Bruno side, the choice of a geometry release must be done using an env variable
    - We could also provide some scripts to link/copy a geometry release locally for manual modifications
  - In this context, each geometry variant_version becomes a new geometry release
    - A sensible numbering/naming scheme can be agreed upon
- Interaction with the release-building team lead to an agreement for a first simplified implementation, where svn tags are used as releases, and the production system takes care of assembling the correct runtime before submitting the jobs

Conclusions

- Many interesting developments done recently
- More to come in the near future
- Even more would come if we had more manpower...