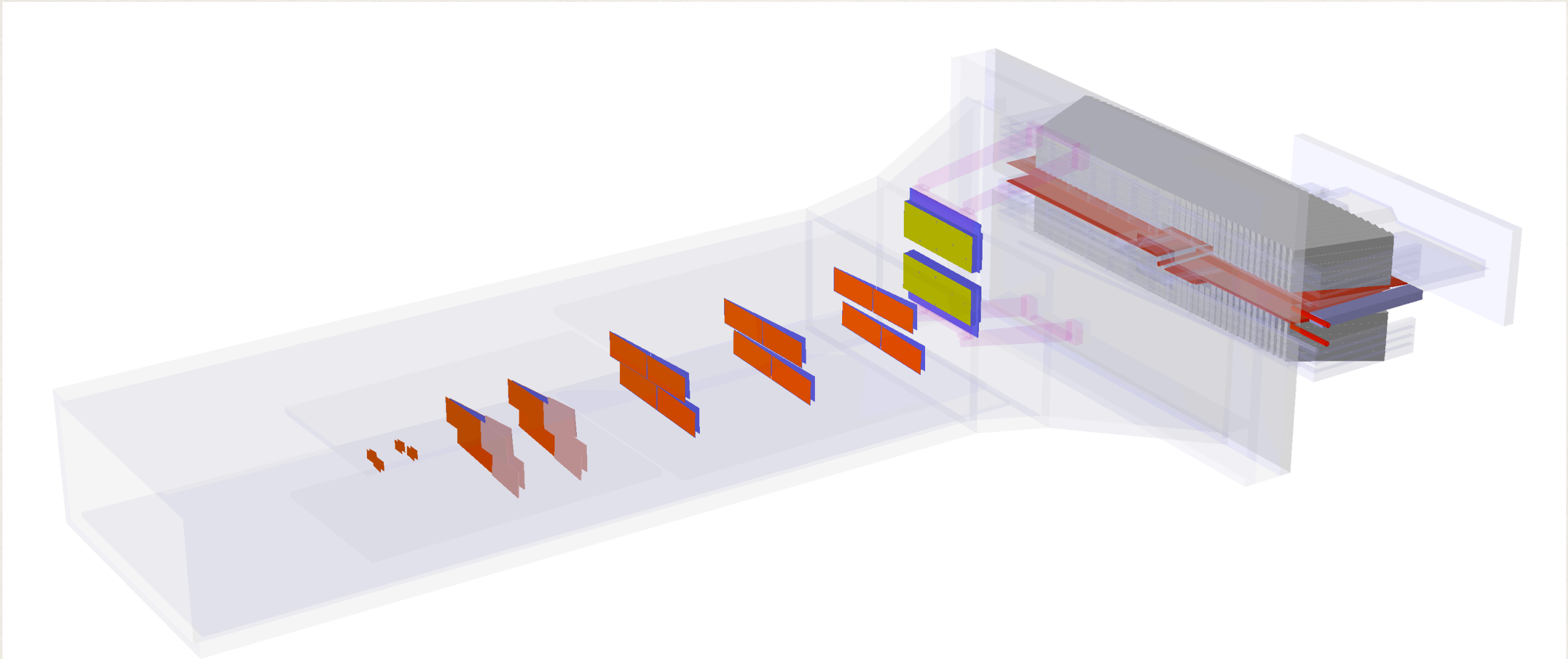


HPS Software

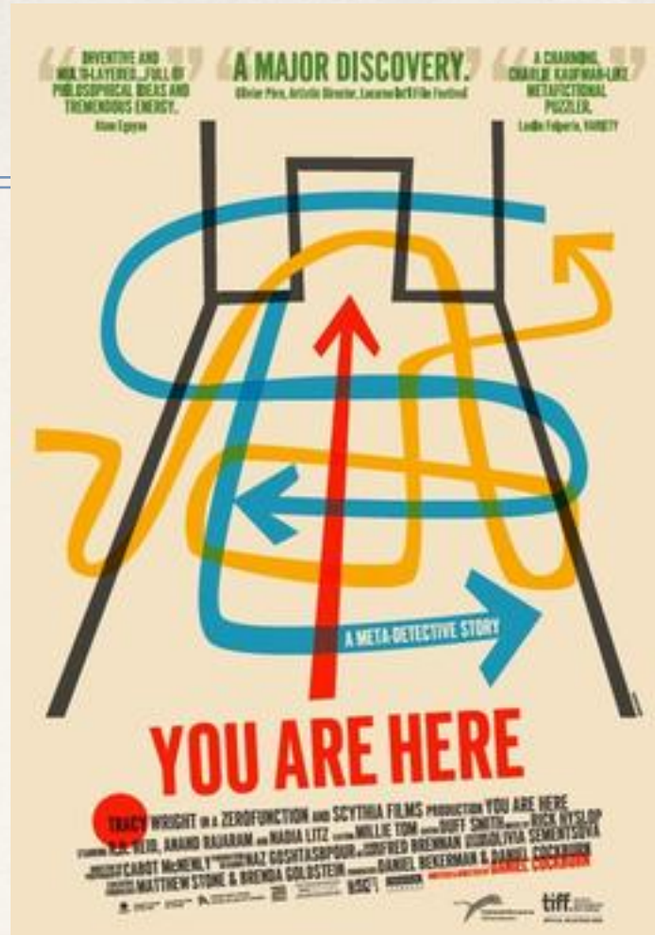
Maurik Holtrop — HPS Collaboration Meeting, JLAb, November 18, 2019

Outline

- Where we are
- Where we need to go
- Outstanding Tasks and Issues:
  - Critical
  - Important
  - Enhancements
- Conclusions

Where we are

- We have gone through the process of data to publication once, plus several theses.
- All components to process and analyze the data exist.
- However:
  - There are a lot of improvements needed.
  - New issues will crop up that will need to be addressed.
    - We have a lot more data than before, so everything needs to run smarter and faster.
- We also have new people on the team, and new students joining, while previous contributors are moving on.



Where we need to go

- **We need our software to be much easier to use.**
  - Currently there are too many ways to make mistakes.
    - Wrong detector, wrong steering file, incorrect settings, ….
  - Some improvement is documenting better, some of this is restructuring.
    - Creating recipes how to do standard procedures would be a start.
    - Cleaning up, so it becomes more clear what to use.

Where we need to go

- **We need all processes to become more standard.**
  - Calibrations could become routine and standardized.
    - Standard procedures, clear instructions on how to run them.
  - Data processing is fairly close to standardized; recipes would be useful.
    - The MOUSE cuts (standard physics cuts) will need tuning, perhaps improving.
    - Early selection cuts to speed up processing, but these need tuning too.
  - Monte Carlo production is being improved and standardized.
    - Tongtong is working on hps-mc to standardize and improve the MC chain (later today).
  - Standardize analysis:
    - Allows people to work on clear improvements of the existing methods, without reinventing a lot of code.
    - Easier and faster to verify results.
    - Allows people to investigate other physics.

Standardizing code

- **Standardizing code will have big benefits.**
  - More collaborative effort
  - Lower the barrier to get started on an analysis.
  - Easier to get help when you get stuck.
  - Far easier to verify the correctness of an analysis.
  - Less time spend on interfaces and analysis machinery, more time on actual analysis.

Standardizing code

- **Standardizing code is not without cost.**
  - Polder Model: work towards a system build on consensus.
    - Requires a wide level of participation - input from many /
    - You cannot (always) have “your way”.
    - Discuss implementation before doing the work.
    - Respect other people’s preferences.
  - There is extra work to organize the code better.
  - There is extra work to document what is there and how it is used.

Outstanding Tasks and Issues

- **Tracking**
  - Updates for new L0 layer, optimize seed finding.
    - Also, dealing with missing sensors in data.
  - Updates to GBL (PF's talk tomorrow.)
  - Introduction of Kalman Fitter (Robert's talk tomorrow.)
  - Improve seed finders: Pattern recognition (track finding) using 1D strip hits.
  - Alignment (Norman's talk tomorrow.)
  - Efficiency studies, again, now for the 2019 "with holes".
- **Trigger Efficiency studies**
  - Need to complete and verify trigger evio2lcio.
  - Need to implement trigger readout and simulation for MC.
- **Hodoscope.**
  - Finish the hodoscope reconstruction code and integrate with analysis.
  - Add hodoscope to MC trigger simulation.

Outstanding Tasks and Issues

- **Mouse = Standard Physics Cuts.**
  - Update the MOUSE cuts for 2019 data.
  - Initial optimization of the cuts.
  - This will speed up the start of data analysis.
- **Standardize DST and analysis framework**
  - **hpstr** - still needs a lot of work.
    - Here we need more input from a wider group of people.
- **Standardize Bump hunter**
  - Kyle is now using Omar's bump-hunter.
  - Code should become a standard tool.
    - Allow for updates, compare different methods.
  - Need something similar for vertexing search.

Longer term issues

- **Improve MC - data correspondence.**
  - Are we simulating the SVT pileup correctly?
  - Dead/noisy channels - SVT bad channel knockout.
  - Beam current, width, angle, position, on a run by run basis.
  - More accurate timing, with noise.
  - **Get resolutions to agree with data!**

- **Monte Carlo:**
  - New and improved scripts, procedures (Tongtong's talk later)
  - Complete the transition to hps-sim (Cameron's talk later)
  - Complete implementation of WAB biassing.
  - Pulser beam background merging (Me, later)
  - Event mixing in real data.
  - Propagate MC truth through the whole reconstruction chain.

Longer term issues

- **Speed and efficiency:**
  - Improve reconstruction speed
  - Improve and speed up SVT pulse fitting.
  - Improve and speed up ECal pulse fitting.

Finally

- **Lists**
  - Always seems there are lots of to-do lists, issue lists, …
  - The tasks are actually getting done, and then we create new tasks.
  - We *can* use more help resolving them faster!
- **Updates**
  - Though some of the tasks are updates, these can still be very time consuming.
- **People**
  - We have enthusiastic new people on the team contributing to software.
  - Never seems we have enough people, we can use more help, more people looking at the data.