

Notes on workshop

Tunnell @ JAI Oxford

Detector Software from Schedule Review

Tunnell @ JAI Oxford

(not much finished since then...)

Needed *Steps* for *Step 4*

1. Existing in beamline: correct geometry, fields, and materials
2. Basic analysis: get the basic '*bits*' required from the detector (ie. ADC counts, hit location, *good enough* reconstruction, etc.) in a datastructure
3. Analyze data: make sense of the results
 1. Plots in the control room
 2. Basic 'initial paper' analysis and start on cuts/corrections (need analysis group input)
 3. Fancy fun stuff that scales as N_{theses}

Distribution of man- power

- What does *good enough* for physics mean?
- That question may not even apply this year
- We need finish basic stuff: 'where was the hit?', 'why isn't this detector in our geometry?', 'which slab is channel 10 on board 8 hooked up to?'
- We don't have enough people to do things poorly (ie. do things twice)

Resource Loaded Schedule

- Later is a resource loaded schedule for detectors only
- Ignores many man-months for new DAQ unpacker after about 6 man-months already. MAUS will be able to unpack since it's easier, G4MICE not.
- Ignores many man-months for control room plots that don't crash like last year
- No contingency included

	Geometry	Space-points			Tracks	Man-Time w/o contingency
		MC	Data	Both		
Tracker	80% (needs documentation; 0.5 m)	40% Digitization (1-2m)	2m 40% Cabling in CDB 40% VLPC settings in CDB	60% 2m Clustering (ie. 3 planes)	50% (Kalman 6m-1y, Straight track test 2m, helical track test 2m) Code exists but requires KS tests and much cleanup.	1.5y - 2y
TOF	80% 0.5m (needs documentation)	30% Digitization (1.5m)	50% 3m Cabling/Calib. into CDB. Increase calib. bus factor and converge on calib.	80% 2m Clustering (model deadtime)	30% 6m. Working prototype. Many hacks: own beamline geometry, own calibrations. No tests.	13m
EMR	70% (MiceModule limitations)	30% Digitization (in progress, 2m)	0% Calibration (2m) 0% Cabling (1m)	20% Clustering (2m)	0% Form PID (2m) 0% P_z (1m)	10m
KL	20% (not in standard geometry, not checked)	10% No reports from KL group nor usable/testable code. Existing standalone software just for occupancy; unusable for analysis.				3m - 6m?
CKOV	10% (wrong geometry/materials)	0% CKOV nonfunctional; under repaired. Blocks software work. Some routines exist but software very preliminary. Affects TOF resolution.				6m - 1y

Detector Personnel

- Detector Coordinator @ 0.5 FTE: MICE core developer at Oxford
- Tracker @ 2 FTE: 2 FTE at Imperial (pre-doc/post-doc). Others coming (2x undergrad) and going (1x grad). Enough man-power probably if coordinated effectively.
- TOF @ 0.3 FTE: 1 post-doc at UniGe at 0.3 FTE since DAQ expert. 0 FTE on TOF reconstruction.
- EMR @ 0.3 FTE: Focus on EMR production not software. 0.3 FTE student who is graduating thus work possibly 'thesis only'. Two possible new students (Unige/Brunel)
- CKOV @ 0 FTE
- KL @ 0 FTE?: someone possibly exists but trouble contacting

Green Means Publish

- Standalone applications don't cut it; huge overhead for software group
- There are more publishable detectors (ie. ignoring LumiMon, BPM, GVA, etc.) than FTE
- 'mythical man-month': empirical software studies: 'one man can make a baby in nine months but nine men can't make a baby in one month'
- I spend all my time teaching people who speak American; short a US postdoc (new)

Conclusions

- Potential issue, wanted to raise early
- Need some more experience mixed in
- We may make it. We'll try hard. But we may not: 1 month / 0 FTE = infinity.

Dijkstra from 70s

Let me give you just one example illustrating how serious the consequences of the thus engendered confusion may be. One of the planning documents for software research revealed --in a parenthetical remark only-- an unchallenged tacit assumption by referring to **"the tradeoff between cost and quality"**. Now in all sorts of mechanical engineering it may make sense to talk about "the tradeoff between cost and quality", in software development this is absolute nonsense, because **poor quality is the major contributor to the soaring costs of software development**. What can you expect from a planning document that is (implicitly) based on such profound misunderstanding?