



THE UNIVERSITY
OF ARIZONA



The Software Review — what did we learn so far?

Graeme Stewart & Walter Lampl

Reminder: Proposal

- Arising from discussions in the Software Quality session at S&C week 1 year ago
- We agreed to undertake a high level review of subsystem *algorithmic* code
 - This should concentrate on the *design*, not on the specifics of lines of code
- Why do it now?
 - We know that Run 3 is some time away
 - However, the scale of the challenge to get our code ready for multi-threading is very large (3M lines C++, 1M lines python, 2300 packages)
 - LS1 software updates were planned and prepared too late
 - So this time we want to be ready well in advance
 - See Walter's slide from yesterday, re. Run 2 and AthenaMT release planning

Inputs

- People!
 - This is a community review, so we are the experts
 - A few people from each area charged with helping review other areas and providing input for their domain
- Documentation
 - A higher level description of the code design
 - Pointers to documentation, if it exists
 - A workflow, showing how algorithms and tools interact with the data (multiple slides)
 - Which (parts) of the code should be rewritten
 - Comments/information on framework issues
 - Comments/information on how the code is tested and validated

Framework Issues

- Incidents
 - Reviewing the use of these, view to not using them
- Thread hostile issues
 - Hidden caches, which circumvent StoreGate (hidden dependency, plus serialisation)
 - Global variables (if non-const)
 - `const_cast`
- Public Tools
 - Should be migrated to *private tools* or *services*
- Non-Thread safe resource access
- Python in the event loop
- Can algorithms become const
- Internal parallelisation opportunities



How did it go...?

- Long term goals with soft deadline are very vulnerable to delays
 - It took until February to get started (initial proposal had been September 2015)
 - Mainly because of 20.7 release for 2016 data taking
- Engagement of communities was pretty much proportional to their software efforts and commitment
 - Stronger areas found it easier to contribute reviews and prepare material (simulation, tracking)
 - One key person made all the difference in a number of areas (SCT, egamma, tau)
 - Some communities really struggled and also had to confess that *no one* actually knows the software very well now (TRT)
- This is positive because it's uncovering areas of weakness and addressing them during the review material preparation

Date	Topic	Indico link	Nominated Reviewers	Note Taker	Jira Epic
Wed 10th Feb	egamma	https://indico.cern.ch/event/491668/	Everyone!	Graeme	ATLASRECTS-2886
Thursday 25th Feb	Core Simulation	https://indico.cern.ch/event/503127/	Graeme, Ed, Jochen	Ed	ATLASSIM-2624
Wed 9th Mar	<i>Cancelled</i>		Zach		
Wed 16th Mar	<i>Cancelled</i>		Jovan		
Wed 23th Mar	<i>Cancelled</i>		Mark H, Zach, TJ		
Wed 30th Mar	Tracking	https://indico.cern.ch/event/514075/	Ed, John, Jovan, Frank W	Graeme	ATLASRECTS-3035
Wed 6th Apr	Tau	https://indico.cern.ch/event/515386/	Steve, Elmar, Andrea	Walter	ATLASRECTS-3051
Wed 13th Apr	<i>No meeting</i>		Ed, Will L, Ben W		
Wed 20th Apr	<i>No meeting</i>		Jovan, Iain Bertram, Andrea, Marjorie		
Wed 27th Apr	<i>No meeting</i>		Steve, Frank W, Iain Bertram		
Wed 4th May	SCT	https://indico.cern.ch/event/524842/	Frank F, TJ, Jovan	Walter	ATLASRECTS-3176
Wed 11th May	Muon	https://indico.cern.ch/event/528132/	Steve, John, Andrea		
Wed 18th May	Flavour Tagging	https://indico.cern.ch/event/532201/	Mark H, Zach	Walter	
Wed 25th May	<i>No meeting</i>		Ben W, Marjorie, Elmar		Fast simulation - delayed
Wed Jun 1st	<i>No meeting</i>				Clashes with Simulation Workshop
Wed Jun 8th					Clashes with Software TIM in Glasgow
Wed Jun 15th	Pixel		Shaun, John, Elmar		
Tue Jun 21	TRT		Will L. , Shaun, Mark H, John		
Wed Jun 22nd	Trigger		TJ, Frank F		
Tue Jun 28th	Jet/PFlow/MET				
Wed Jun 29th	Calo		Jovan, Frank W		
Tue Jul 5th	Digitization		Ben W, Alex M, Shaun		
Wed Jul 13					
Wed Jul 20	Generators		Alex M		

- Walter and I have to admit to being tardy with the conclusions on occasion too

The Review Material

- Generally impressive material has been shown
 - Timely preparation of material is critical
 - Google docs format has proved to be absolutely ideal for having a dialogue in advance
 - Sometimes misses key points (especially on multi-threading questions)
 - Sometimes too long (Muons presented a staggering 99 slides!)
 - Clearly good if this becomes some internal documentation, but hard to fit into the allotted time

The Reviewers

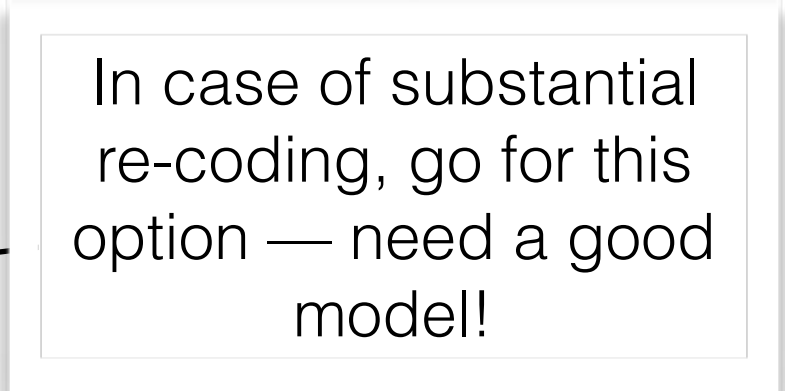
- Impressive!
 - Reviews have contributed a great deal
 - Hard work done to look at the review material in advance
 - Perspicacious comments on the structure of the code
 - Even some really detailed digging into code itself, which we did not think was feasible
 - Especially some good comments on threading issues and some key design problems therein

Observations I

- Code debt is quite high
 - Code has been adapted and extended and lost original design clarity
- There is considerable code duplication in some areas
 - Cut and paste seems to have been a common design pattern
 - Looking for this with a tool makes sense
 - Shows the importance of really good strong examples
- ‘Opportunities for parallelism’ seemed to be taken too much to heart
 - A few discussions about parallelising things whose total CPU consumption is actually quite small
 - Restructuring ‘super algorithms’ to break them apart where possible is a much better strategy for most pieces of reconstruction code

Observations II

- No real show stoppers observed re. data flow
 - A lot of data held in tools is not even a problem if used within the same algorithm
 - Private tools!
 - However, better to move to StoreGate and be explicit
 - 100% needed when public tools are used to cache data between algorithms
- `const_cast` is definitely an issue
 - Some of this solved at core level
- Ownership of objects can be confusing and unclear
- Let's get the basic items done first and build on that
 1. Non-thread hostile code
 2. Algorithms and tools that are clonable
 3. Re-entrant algorithms and tools



In case of substantial re-coding, go for this option — need a good model!

Looking Forward

- Finish the reviews by the summer
- Start planning how we make the code MT ready in detail
- Generally start work on this from Autumn
- A lot of coding, testing and hard work to come
 - Good examples will be critical — we already saw a lot of inappropriate copy and paste
- The review is only the start
 - But it's a good start!

