

Experiment Integration and Validation

David R. Quarrie

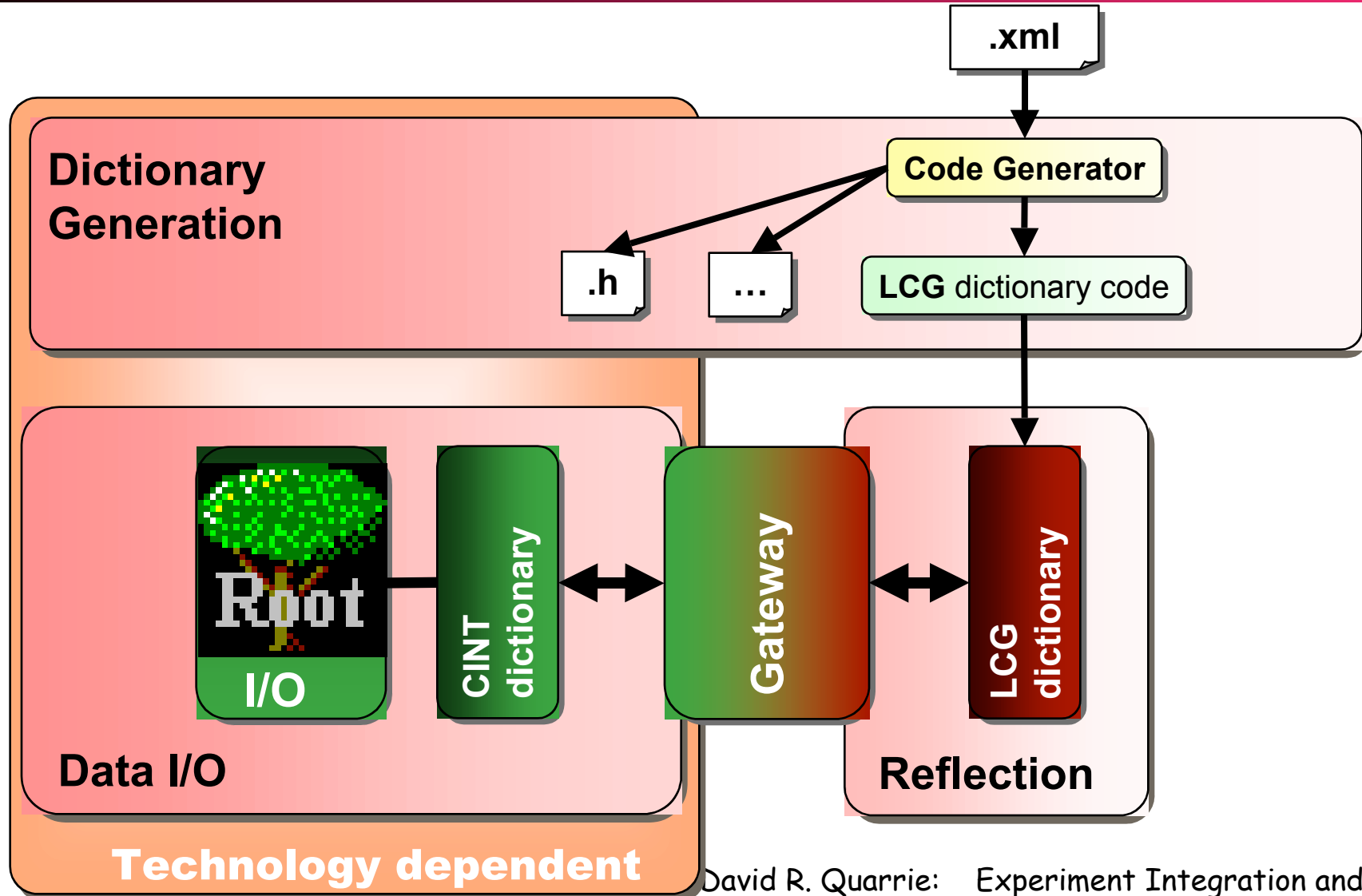
CERN/LBNL

David.Quarrie@cern.ch

Introduction

- Summarize feedback from all experiments on all products
- Main emphasis will be on POOL integration because that's the first one to be deployed by experiments
 - Incorporates significant SEAL components as well
- But I'll try to cover other products/projects
- Brief overview of status of validation
- Some of the lessons learned and responses to those
- Many thanks to people who provided input for this talk

POOL/SEAL Components (LHCb view)



POOL Integration Approaches

- CMS and ATLAS share model parsing C++ .h files using gcc_xml and generating dictionary filling code
 - XML configuration file allows overrides (e.g. transient data members)
- LHCb use XML files as primary description and generate both C++ .h file and dictionary filling code from there
- In both approaches the SEAL dictionary plays a central role
- Integration tested two major areas
 - The filling of the SEAL dictionary (and the dictionary itself)
 - Coupling to ROOT I/O through gateway

CMS & POOL

- Fruitful collaboration with POOL team since inception
- 2.6 FTE direct participation
- Efficient communication
 - Savannah Portal
 - Direct mail and phone exchange among developers
 - In person meetings when required
- Continuous and prompt feedback
 - CMS typically provides feedback on any new pre-release in few hours
 - POOL typically responds to bug reports in 24/48 hours
 - ▶ Only a few took more than a week to be fixed in a new pre-release

CMS Integration Experience

- First POOL production release on 30 June 2003
 - In reality just an honest prototype with many bugs, missing features and major performance problems
 - Demonstrated that internal unit and integration testing had poor coverage and inadequate complexity
- COBRA release base on POOL 1.2.0 available in early August
 - Still not production ready (unexplained errors and crashes)
- CMS put many weeks of effort into debugging, in close collaboration with POOL team

CMS Current Status

- First public release of CMS products (COBRA etc.) based on POOL 1.3.x available on 30 Sep 2003
- Used in production, deployed to physicists
 - e.g. 2 Million events produced with OSCAR (G4 simulation) in a weekend
 - New tutorials (just started) based on software released against POOL 1.3.3
- Essentially same functionality as Objectivity-base code, but:
 - No concurrent update of databases
 - ▶ No direct connection to central database while running
 - Remote access limited to RFIO or dCache
 - No schema evolution
- Still a few bugs, missing features and performance problems
 - Affect more complex use-cases
 - Impact the deployment to a large developer/user community

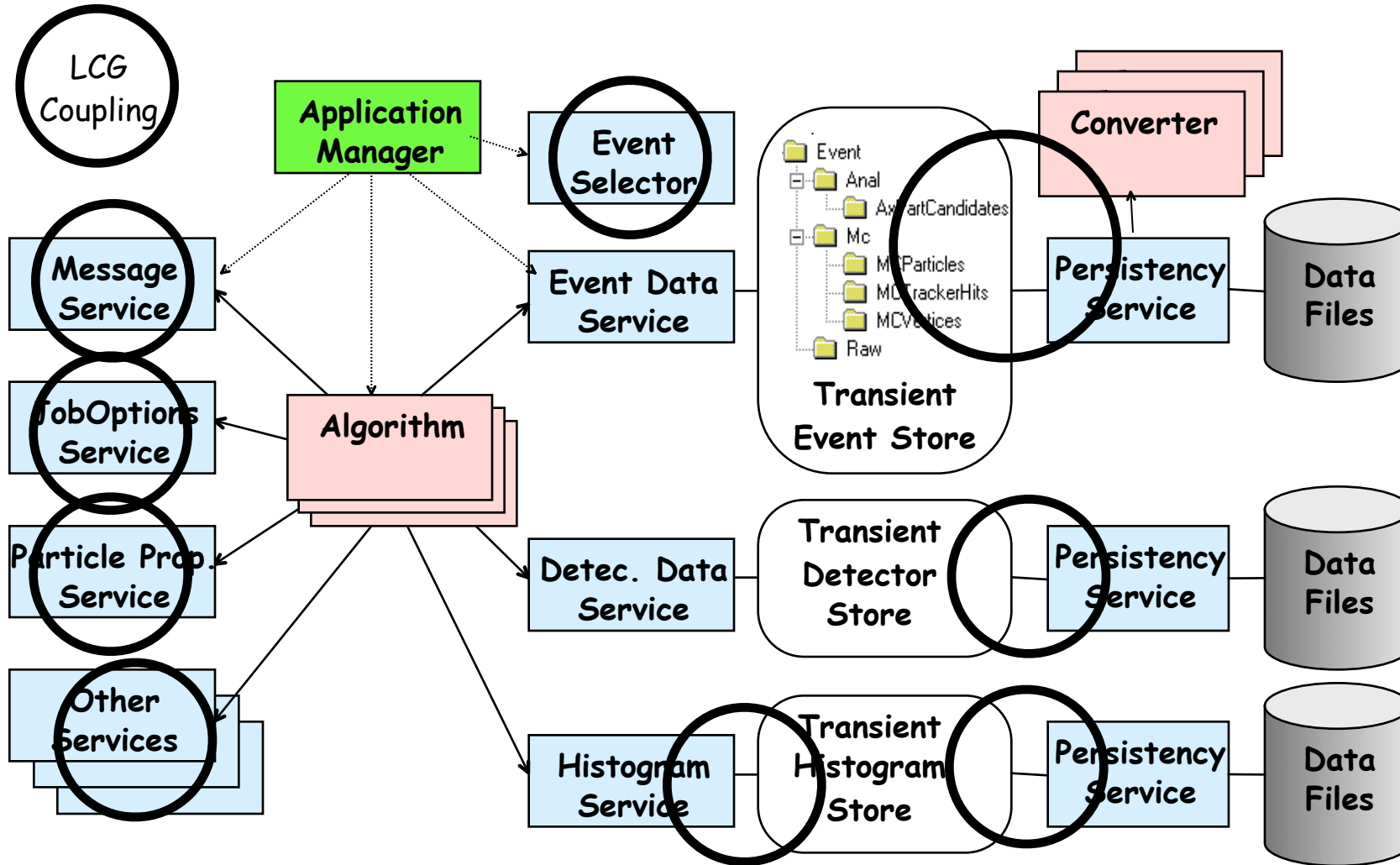
CMS Comments on SEAL

- SEAL looks like a collection of quite heterogeneous and independent products
 - SPI distribution hides the complexity of the project well
 - Difficult for individuals willing to test or integrate a single component
 - Dependencies on ROOT & CLHEP makes integration difficult
- Apart from what was ported from Iguana, CMS does not make any direct use of SEAL at the moment
 - No plan to use the dictionary outside pool
 - No plan to use high level framework infrastructure (e.g. whiteboard)
 - Minuit investigations begun
- Concerns about impact of different experiment priorities on the project

CMS Plans

- POOL schema frozen for next 18 months
- Follow a minimalist approach to avoid further confrontations with bugs, missing features, performance problems
- New projects (such as Conditions DB) will build upon LCG/AAA software and make larger use of it
- Bottom line is that POOL is no longer on critical path towards CMS Data Challenge in 2004

Gaudi/Athena & LCG Components



LHCb Strategy

- Adiabatic adaptation of Gaudi to SEAL/POOL
 - Slow integration according to available manpower (1 year for full migration)
 - Take advantage of face-lifting of “bad” interfaces and implementations
- Minimal changes to interfaces visible to physicists
- Integration of SEAL in steps
 - Dictionary integration and plugin manager
 - Use of SEAL services later
- Integration of POOL earlier than SEAL
 - Working prototype by the end of the year
- Necessity to read “old” ROOT data
- Keep the LHCb event model unchanged

LHCb Status

- Mechanics for persistency of event data implemented
 - New Gaudi plugin created
 - New Gaudi conversion service
 - ▶ One converter class for all object types
 - New EventSelector service to access implicit POOL collections
- SEAL dictionary generator recently completed
- MC Truth relationships recently completed

ATLAS POOL/SEAL Strategy

- Phased integration of POOL/SEAL
 - POOL first
 - Then client access to SEAL dictionary
 - Then other SEAL components
- Design of POOL integration started before end of 2002
- Actual integration started in spring 2003

ATLAS POOL/SEAL Integration Feedback

- Integration took much longer than originally expected
- The fact that ATLAS developer working on this had not been a part of the core POOL team (and was remote) contributed
- Lots of nuisance technical obstacles
 - Conflicts in how cmt/scram/ATLAS/SPI handle build environments, compiler/linker settings, external packages & versions...
 - Conflicts between Gaudi/Athena dynamic loading infrastructure and SEAL plugin management
 - Conflicts in lifetime management with Athena/POOL transient caches
 - Figuring out "gotchas" like classes with private constructors
 - Obscure error messages
 - etc.

ATLAS SEAL/POOL Status

- Required functionality is essentially in place
 - Main outstanding problem is that of CLHEP matrix class
 - ATLAS has not yet tested EDG RLS-based file catalog
 - POOL explicit and implicit collections are available via Athena, but fuller integration will require extensions from POOL (and refactoring on ATLAS side)
- We're moving from an expert to general developer environment
- The event data model is being filled out
- Goal is to have it essentially complete by end of the year
 - Although parts of transient model are still being redesigned
- Python scripting support is being incorporated now
 - PyROOT, PyLCGDict, GaudiPython, etc.

PI Feedback

- Re-implementing of Gaudi/Athena histogram service based on AIDA/ROOT implementation (needed by both LHCb & ATLAS) is complete
 - Available in next Gaudi release
- ATLAS does not presently intend to take advantage of enlarged AIDA API
 - Wait until more experience gained from physics analysis
- CMS has ported to PI all code that was previously based on ANAPHE
 - A few missing features were identified but have been rectified by PI team

Simulation Feedback; Geant4

- LHCb, CMS and ATLAS have long history of active involvement with Geant4
- Extensive validation studies
- Physics and memory/cpu performance have reached point where it's deemed to be ready for production
- All are now actively deploying (CMS) or in the final preparations for deployment (LHCb, ATLAS) for their Data Challenges in 2004
- Also ATLAS use for combined testbeam Apr-Oct 2004

Simulation Feedback: Generators

- CMS has already produced generator-level events for their DC -04 Data Challenge
- CMS is in process of integration tests of GENSER
 - First simulation is a candidate as first user
- LHCb believe that the generators should be treated as external (cf Boost, Xerces) and should not be copied into a CVS repository as GENSER which couples them together
- ATLAS is in the process of validating the GENSER generator distribution
 - Migration to use some generators from distribution underway now

SPI Feedback

- Primary focus is the LCG projects themselves
 - But very useful interactions support for the experiments
- Savannah portal is widely used
 - Primarily for bug tracking
 - Response for upgrades has been tempered by problems with OpenSource environment, but these have apparently been addressed now
- ATLAS, CMS and LHCb are migrating to the SPI External installations
- Useful scripts and procedures are available from QA/testing
 - Although "some assembly required" because of different package structures
- Generally very good interactions with SPI

Build Tool Feedback

- The build/configuration tool issue has been painful
 - `scram/cmt/appwork`
- I suspect that no-one is really happy with the present situation
 - This might be an inevitable conclusion given the disparate systems already in use by the experiments
- LHCb and ATLAS are very supportive of the decision to add support for CMT to the products

LCG and ALICE

- By the time LCG started ALICE had already a full system in place, including a distributed computing grid solution
 - Of course still far from the final system!
- ALICE is not depending on any of the LCG AA projects
 - POOL functionality is provided by a combination of AliEn file catalogue and native ROOT
 - SEAL and PI functionalities is provided directly by ROOT
- ALICE is collaborating intimately with the ROOT and FLUKA team
- ALICE is very worried by existing unnecessary duplications and supports strongly the pledge to reconsider the user-provider relation with ROOT and “converge” expressed by the internal review

LCG and ALICE

- ALICE develops generic technologies of interest to LCG
 - The Virtual MonteCarlo has been declared of interest by the Simulation project
 - Unfortunately no manpower has been found to be assigned to it and ALICE is continuing its development alone
 - The geometrical modeller as a montecarlo-independent complement to the virtual montecarlo
 - The PROOF system, developed together with the ROOT team for interactive parallel and distributed analysis
 - Demonstrated together with AliEn at Supercomputing 2003
 - To be used in production for the ALICE Data Challenge 1H04
 - The AliEn is a complete but open and extensible Grid solution based on Web Services model and standard protocols

Validations

- Most major validation activity is still to come
 - However, 6 of the 14 milestones have already been met

Description	Date	Status
CMS POOL integration: POOL persistency of CMS event	2003/7/31	Done v=0
CMS POOL acceptance for PCP	2003/7/31	Done v=0
CMS SEAL integration supporting POOL usage	2003/7/31	Done v=0
ATLAS POOL integration: POOL persistency in Release 7	2003/9/10	Done v=1
ATLAS SEAL integration supporting POOL usage	2003/9/10	Done v=1
CMS POOL validation with PCP data	2003/10/31	Done v=-10
ATLAS int: ROOT implementation of AIDA histograms in Athena	2003/11/30	
LHCb POOL integration: Gaudi persistency replaced by POOL	2003/12/19	
LHCb integration: SEAL plugin manager integrated in Gaudi	2003/12/19	
ATLAS integration: SEAL integration into Athena	2003/12/31	
ATLAS POOL validation with DCI data	2004/1/19	
ATLAS validation of POOL metadata/event collections	2004/1/31	
ATLAS POOL validation with complete Event Data Model	2004/3/31	
ATLAS int: Generic simulation framework validated by ATLAS	2004/4/30	

Lessons Learned and Responses (1)

- Effort needed for integration generally under estimated
- Original development model was for frequent releases and rapid feedback
- Original integration model was that developers from the experiment working on a product would integrate it as well
 - However, this model has been found to be flawed
 - ▶ Those developers still have ongoing deliverables
 - ▶ Not every experiment has people working on all aspects of products
 - Response has been to assign integrator/liaison where appropriate
 - ▶ E.g. POOL/ATLAS - is beginning to work well
- Different priorities and timescales have driven schedules for integration as well as manpower limitations

Lessons Learned and Responses (2)

- Configuration management is hard
 - Even within a project
 - Also because of cascade of version dependencies across products
 - ▶ e.g. consistent version of CLHEP/ROOT across all products
- There is still an advantage to being resident in the same site for development and integration
- Initially the fact that LHCb and ATLAS already were collaborating on a variety of tools was not given any weight by LCG
 - This is being (somewhat belatedly) addressed with e.g. CMT support and offer of help with SEAL integration