CMS & LCG/AA

AA Internal Review, March 2005

Vincenzo Innocente

in behalf of CMS Collaboration

AA: core software for LHC

- **CMS believes that the goals of LCG/AA (to provide to the LHC experiments a set of coherent software packages) has been largely successful and the full integration of ROOT will bring it to completion**
  - Persistency
  - Foundation
  - Framework services
  - User interface
  - Simulation
  - Software Process (?)

- **We take the opportunity to acknowledge the key role of Torre Wenaus as project leader in this**
  - We warmly thank him for his efforts
  - We wish Pere Mato a fruitful tenure

Experiments in AA

- **Key to success of AA: (has been) the continuous involvement of the experiments, at all levels and with specific and concrete responsibilities:**
  - Specification of requirements and technologies in the RTAG
    - Realistic goal matching concrete needs for LHC
  - Day-to-day management in the AF
    - Milestone-matching between AA & experiments
    - Quick feedback and fast corrective actions
  - Participation at both design and implementation levels
    - Build a common culture
    - Knowledge spread among a large community close to end-users
- **AA Software is primarily OUR software**
  - CMS looks forward to a renovated phase of close collaboration

CMS Contribution to LCG AA in 2004

- **Reduced with respect 2003**
- **SPI : 0.3 FTE**
  - OVAL
  - SCRAM
- **SEAL: 0.3 FTE**
  - Contribution to general design
  - Implementation of Foundation classes (re-engineering of *iguana classlib*)
  - Implementation of core framework services (plug-in mechanism)
  - Mathlib
- **POOL: 1.5 FTE**
  - Responsibility of FileCatalog
  - Implementation of Collections based on root-tree
  - Contribution in the Relational storage service
  - Debug and testing
- **PI: 0.2 FTE**
  - Project Leadership
  - Interface to POOL
- **Simulation: 0.5 FTE**
  - Leadership of Generator task
  - Contribution to Simulation validation
- **3D (Distributed DataBase Deployment) : ~1.0 FTE (actually under deployment area)**
  - Responsibility in the Proxy system (FermiLab project)

CMS Software in 2004

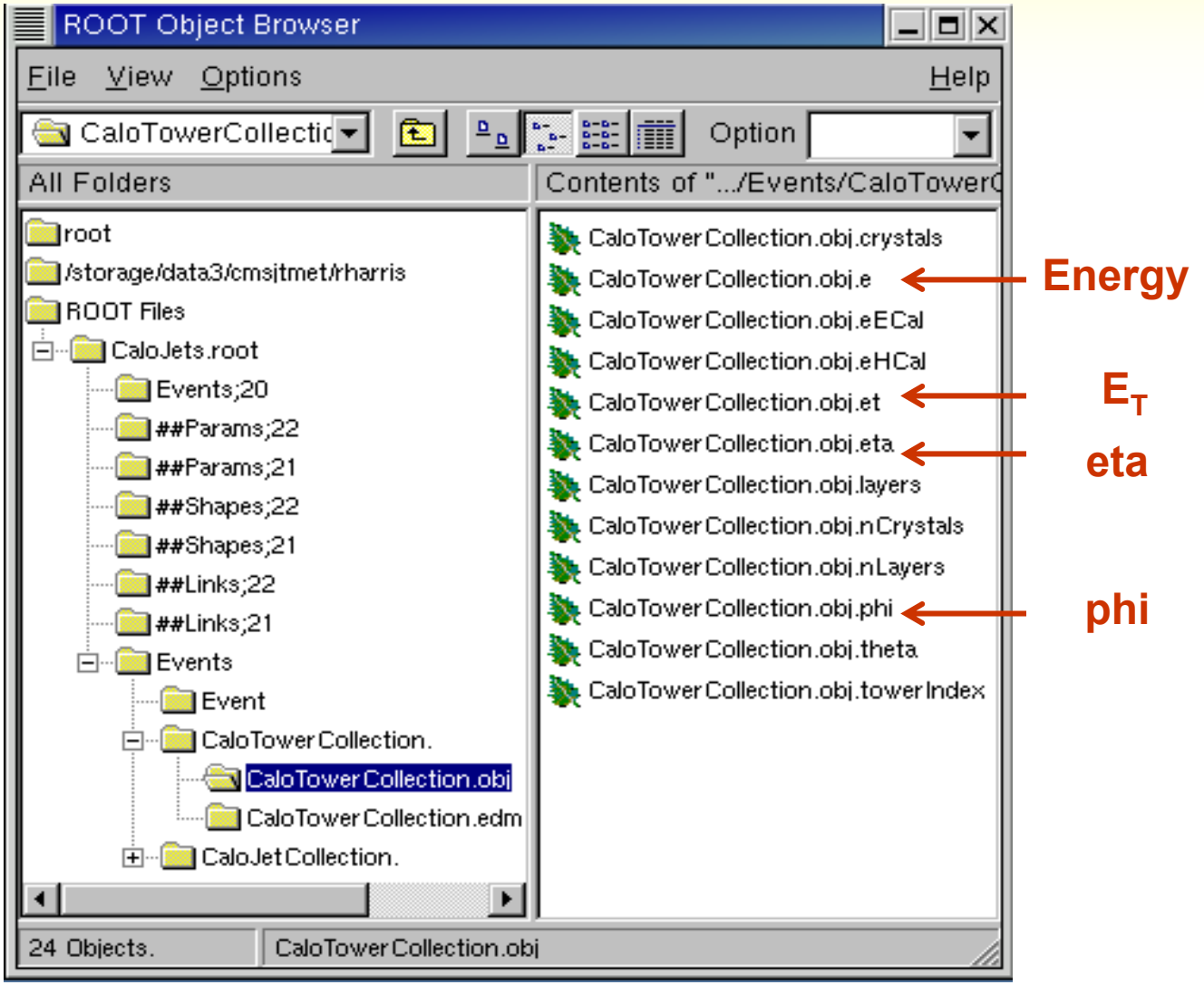
- **DC04: sustained data production at 25Hz**
  - Reconstruction at T0 (CERN)
    - 40MB/s out of CASTOR
    - 500 jobs running in parallel
    - 4MB/s out to CASTOR
  - Data Distribution and remote analysis
    - 6TB (~3.5M file replicas) distributed to Tier-1
    - 20 minutes latency for analysis to start at Tier-2
  - AA software (in particular POOL) was never a show-stopper
- **Continuous Production**
  - 80 Million events simulated and reconstructed
  - > 100 TB stored using POOL (w/o counting replicas)
  - No crashes due to AA software (Pool, G4 in particular)
- **Review of this experience**
  - Event data model too complex
  - Too many small files
  - Bookkeeping based on metadata stored in files does not scale

New CMS Software

- **CMS is now developing a new framework and event model**
  - Event consisting in a flat collection of “products” mapped on branches of a single root tree
    - POOL upgraded to support this model (see next)
  - Independent modules for I/O, reconstruction, filtering and analysis
    - Selected and loaded at run-time using SEAL plugin
  - Set of services (geometry, calibrations, configuration, etc) based on SEAL framework component
  - Support for three modes of operation
    - Direct browsing using ROOT
    - Analysis with minimal services and no reconstruction modules
    - Full simulation and reconstruction with all required services
- **As already happened for POOL placement in ROOT branches it may require new functionalities from AA software**

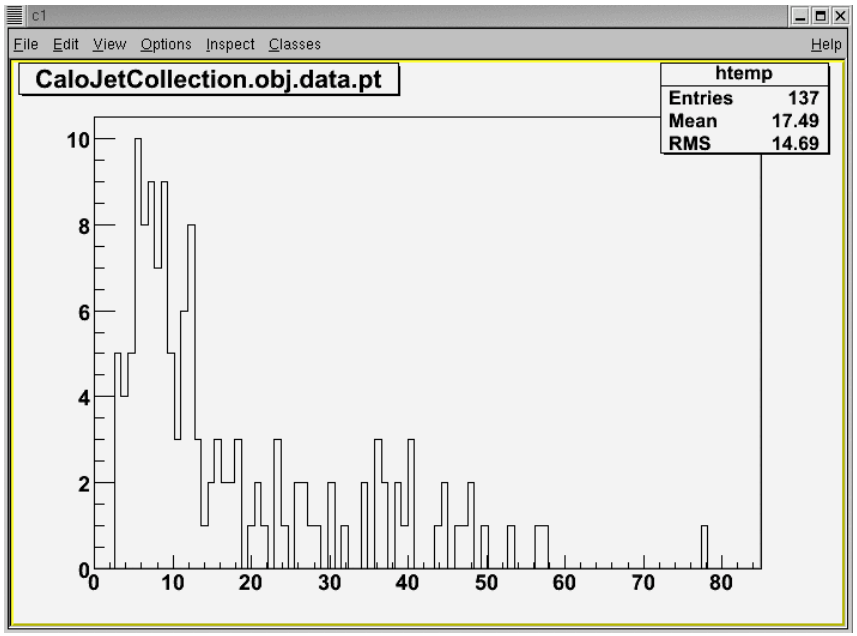
Browsing the EDM: CaloTower

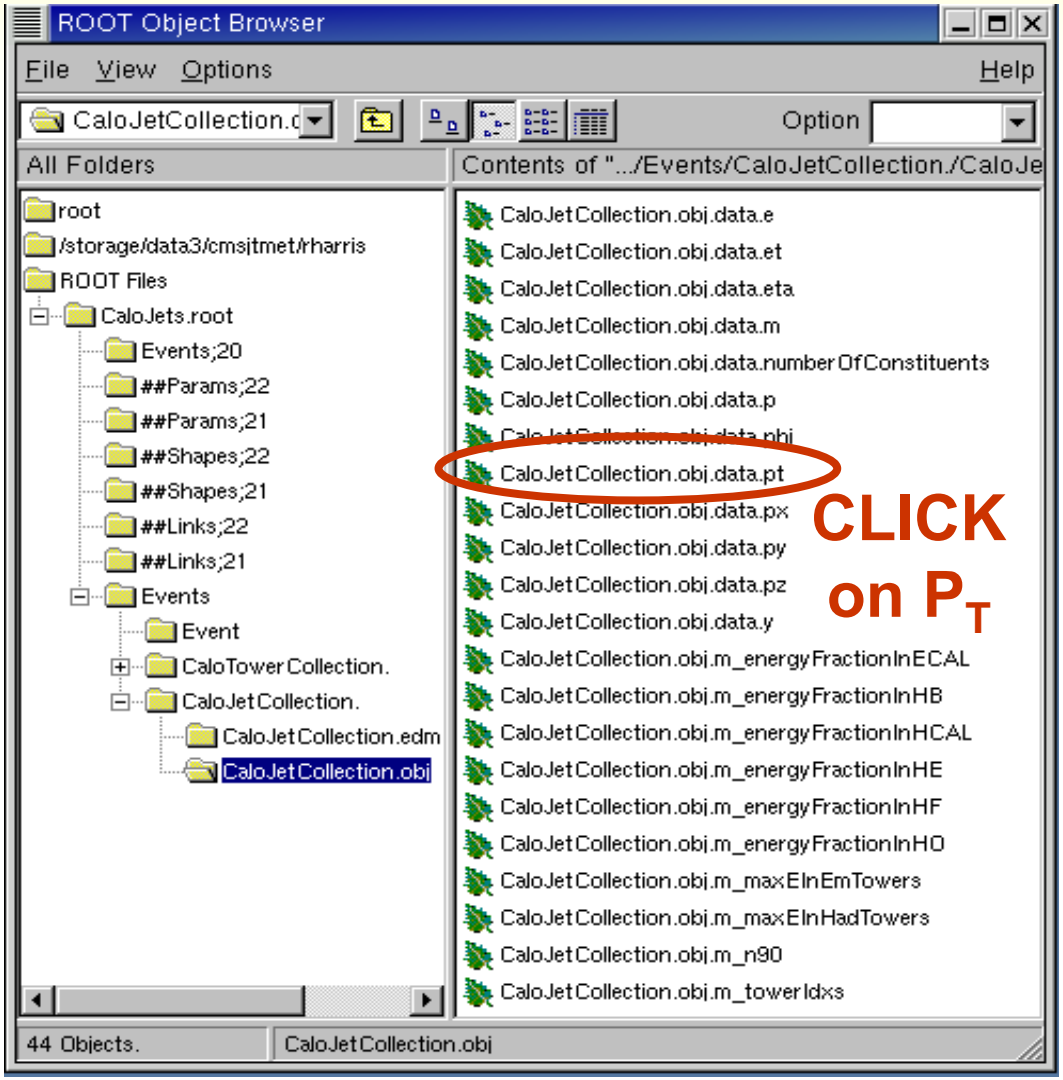
- **We click on the branch CaloTowerCollection**
- **We see leaves of CaloTower data.**
  - Discover the names!
- **Each leaf is a piece of data we can examine**
  - The “crystals” and “layers” leafs are vectors, which we cannot browse inside yet with root.
  - Feature of browsing vectors inside vectors will be added to root soon.



Data Inspection with Root Browser

- **Click on leaf**
  - Displays histogram
  - Good for quick tests
  - Histograms without writing a line of code.





CMS & POOL

- **CMS has established a fruitful collaboration with the POOL team since the very beginning of the project**
  - Direct participation to the project itself: (up to **2.6 FTE)**
    - Catalog, Collection, Root-Storage, test
      - Latest contribution: placement in branches (see previous two slides)
  - Efficient communication
    - Savannah Portal
    - Direct mail (and phone) exchange among developers
    - In-person meetings when required
  - Continuous and prompt feedback
    - CMS typically feedbacks on any new pre-release in few hours
    - POOL responds to bug reports in 24/48 hours
      - Only few bugs took more than a week to be fixed in a new pre-release
      - “Percolation” to ROOT, when required, always effective
    - Participation to reviews and involvement in long-term planning

Few POOL-related milestones

- **Dec 2002: dictionary built for typical CMS data classes parsing original header file with gcc-xml**
  - dictionary moved to SEAL, no further direct involvement of CMS
- **May-June 2003: first full-scale integration completed**
  - satisfied most of the cms requirements
  - Start of full-scale realistic tests
- **September 2003**
  - First public release (based on POOL 1.3)
  - Start to be used in production
- **April 2004: DC04**
  - Sustained reconstruction and data distribution at 25 Hz
- **Since Sept-2003: continuous production**
  - More than 80 Million events simulated and reconstructed stored using POOL
  - To be used at least until March 2006

CMS & SEAL

- **CMS considers SEAL as the core C++ library for all its physics applications**
  - Involved in early architectural and design discussions (blueprint)
  - Contributed with 1 FTE to port Iguana-classlib to SEAL-Foundation
  - Involved in design and review of Mathlib and Minuit
- **Key component of the new event-processing framework**
  - Foundation and Utilities classes
  - High-level framework infrastructure (component model)
  - Plug-in Manager
  - dictionary as used in POOL

CMS&Simulation (I)

- **CMS is in production using Geant4 since end of 2003**
  - 6.2 used in production, 7 integrated in the framework and under test&validation
  - Physics processes have been validated
  - Current framework and interface to G4 considered sufficient
  - Geometry “translated” from Geant3 using our tool (DDD)

- **Geant4**
  - Relationship with G4 team improved dramatically over last two years
    - Excellent communication and two-way feedback
    - No more crashes, performance improved (both CPU and memory)
    - Need for an “insulation layer” disappeared

CMS&Simulation (II)

- **Geometry**
  - CMS uses its own tool (DDD) to describe the detector geometry and material properties
    - G4 geometry built at run time automatically
    - Persistent geometry under test
- **Fluka**
  - Second priority activity: use of the same geometry as in G4 is a must
    - Port a test-beam setup first
- **Generator**
  - CMS considers the current LCG/AA effort satisfactory
  - Support integration effort between software-infrastructure and “Theory” communities

SPI Services (I)

- **CMS has been among the first to uptake Savannah**
  - We are satisfied with the service and its development

- **SPI had taken up two CMS products: Oval and SCRAM**
  - It was expected a major development of SCRAM under SPI
    - LGC/AA instead decided to drop SCRAM and develop a new tool
      - No news about this since long time
    - CMS have since invested its own person-power to complete the development of SCRAM according to the original plan
      - Released for production last February
      - CMS asks AA to adopt this new version

SPI Services (II)

- **CMS supports the strengthen of the role of a central librarian in SPI**
  - Should help in guarantee the coherence and quality of software releases
  - Should cover all external products needed by the experiments

- **CMS, as all other experiments, is developing tools and serviced in the area of the software development process**
  - We would like LCG/AA to reconsider to support this type of tools and services for the experiments' needs

Future

- **Root, POOL & SEAL at the base of the old & new CMS framework**
  - Current data and framework required till ~ March 06
    - Old versions should continue to be supported on current platforms
    - Backward compatibility should be ensured CMS relies on current functionalities and architecture
  - AA products must continue to support the evolution of experiment software
    - Collaboration with experiment software teams must continue
- **Simulation project was essential to focus G4 on LHC needs**
  - Excellent results obtained
  - Maintain focus on current and emerging experiments' needs
- **CMS welcomes a rationalization of Core libraries as envisaged in the Root/SEAL reorganization**
  - Along the lines experimented with MathLib (see CMS recommendations)
    - Clear modular structure
    - Coherent interfaces and consistent behavior in all environments
    - Sound and robust implementation of all key components

Priorities for 2005

- **Complete POOL program of work**
  - Single LCG-ROOT dictionary
  - Automatic run-time loading of dictionaries
  - Provide pool functionalities at root-prompt level
    - Support of pool::Ref and of main interfaces
  - RDBMS back-end
- **GEANT4**
  - Further improve performance and reliability
  - Fully validate physics for LHC use-cases
    - Including shower parameterization
  - Persistent geometry and physics-lists
- **SPI**
  - Adopt new version of SCRAM (in AA releases)
- **Complete MathLib**
  - Investigate integration of CLHEP, re-engineering of Minuit…

Concluding Remarks

CMS has ported to AA all applications that were previously based on Geant3, Objectivity, LHC++ for all previously supported use cases.

- AA Software used in production since September 2003
  - It has never been a show-stopper
- At the base of CMS Software for LHC Start-up

LCG/AA has provided to CMS excellent products.

We wish to see a continuation of its program of work in close collaboration with the experiments along the lines formulated in the RTAGs that defined it