Integrating AA software in COBRA

SC2, April 2004

Vincenzo Innocente

CERN/EP

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: **2.6 FTE**
  - 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

Few old 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
- **March 2003: first tests of FileCatalog**
  - Feedback on performances, API and command-line tools
- **April 2003: POOL_0_5_0 released**
  - First version able to support realistic use-cases
- **May 2003: first full scale integration completed**
  - 99% of persistent classes in lcg-dict
  - Missing features identified
    - All about items already supported by “Vanilla” Root
- **14 June 2003: POOL_1_1_0-theta released**
  - satisfied most of the cms requirements
  - Start of full-scale realistic tests

Use of Pool in CMS: Current Status

- **COBRA 7.7.x OSCAR 3.0.y ORCA 8.0.w**
  - Based on POOL 1.6.z
  - First public release on September 20 2003 (based on POOL 1.3)
  - In use in production
- **Usable for production, deployed to physicists**
  - 2 Million events produced with OSCAR (G4 simulation) in a week-end
  - 10 Million events reconstructed in DC04 with no crash
  - SW tutorials each Friday based on ORCA 7.5 since October 2003
- **Essentially same functionality as Objectivity-based 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
- **Many bugs, missing-features, performance problems solved since first deployment**
  - Always a very effective collaboration between CMS POOL ROOT and SEAL
  - Never more than a week to solve a problem

What CMS use of POOL?

All objects (event and metadata) are stores as root keyed-objects (no root-tree)

Only object navigation is used, no other access mechanisms

- **File Catalog**
  - Full interface
  - XML implementation in Physics Applications
  - MySQL & RLS under test for production use cases
- **Ref**
  - Full interface
- **Session**
  - Only Transaction Management
- **Few other classes and methods**
  - Mainly workaround to bug/missing-features
  - In test programs

CMS persistency paraphernalia

- **Thread-safe proxy-wrappers to pool-interfaces**
- **Scoped (exception-safe) nested-transaction**
- **Context/Thread-specific Data Services**
- **Creation and management of DataBases and Containers**
  - Including catalog, PFN, LFN and metadata
- **Object (RefBase) -based placement hint**
- **Generic "named" navigation**
  - Mono and bi-directional map<string,Ref>
- **Specialized (base) classes**
  - Smart-Proxies
  - Collections
  - …

Integration with LCG-2

- **CMS was very pleased to discover how the File Catalog interfaces (API, command-line and python) responded to all requirements related to the use of COBRA in the grid environment**
  - Unfortunately we discovered also that this was not the case for the EDG side
- **All functionalities seem to be there and working**
  - Catalog management
  - Query using all sort of attributes
  - Publishing fragments in and out a Master catalog
  - Possibility of use multiple catalogs (new in POOL 1.6)
  - Replica registration and discovery
  - Possibility to implement custom backend to “bestPFN”
- **Performance**
  - Resonable for XML and SQL implementation
  - **VERY** poor for the RLS implementation
- **All these features have been tested on LCG-2**
  - In particular at CERN and INFN

Few Comments on SEAL

CMS contributed with 1 FTE to port Iguana-classlib to SEAL-Foundation

- **SEAL looks more and more like a collection of quite heterogeneous and independent products**
  - SPI distribution hides well the complexity of the project
  - More difficult for individuals willing to test or integrate a single component
  - The dependency from Root and CLHEP makes integration difficult
- **No plan for big-bang migration to SEAL at the moment**
  - Slow replacement of foundation classes
  - dictionary used for POOL, possibly as bridge to python
  - No plan to use high level framework infrastructure (whiteboard for instance)
  - Experimenting with new Minuit: used already in physics code
  - Plugin-manager used in new packages
    - to load algorithmic geometry-builders in CMS-DDD (XML)
    - to load SEAL Storage Manager as generic root I/O plugin

Few Comments on Simulation

- **CMS is in production using Geant4**
  - Physics processes have been validated
  - Current framework and interface to G4 considered sufficient
  - Geometry translated from Geant3
- **Geant4**
  - Relationship with G4 team improved dramatically over last year
    - Excellent communication and two-way feed-back
    - Needs for an “insulation layer” almost disappeared
- **Geometry**
  - CMS is in the process of reviewing its simulation geometry description
    - Target: ready end of 2004
    - Tools are essentially in place
- **Fluka**
  - Low priority activity: use of common geometry with G4 is a must
    - Port a test-beam setup first
- **Generator**
  - CMS considers the current LCG/AA effort satisfactory
  - Concerns for non converging toward a common “Event” structure (in memory and persistent)

SPI Services

- **CMS has been between the first to uptake Savannah**
  - We are satisfied of 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
    - This has not happened, LGC/AA decided to drop SCRAM and develop a new tool
    - CMS has been forced to invest his own person-power to complete the development of SCRAM according to the original plan
- **CMS supports the strengthen of the role of a central librarian in SPI**
  - Should help in guarantee the coherence and quality of software releases
- **CMS would like SPI to support more directly experiments providing tools and service in the area of the software development process**

Scripting and interactive environment

- **COBRA provides a scripting and interactive environment in python (using boost)**
  - Seamless integration with SEAL, POOL and PI components exported to python (as for many other external software such as HippoDraw, KDE etc)
  - Plan to test python biding using LCG-Dict
- **No experience in exporting CMS software to Root/CINT**
  - No LCG/AA software exported to root/cint yet

Future

- **Freeze "schema" now for next 18 months (now only 12 left)**
  - SEAL/POOL will not support schema evolution in near future
- **Follow a minimalist approach to avoid further confrontations with bugs, missing features, performance problems**
  - Use only what is really needed and produces major benefits to CMS use-cases
  - Avoid early migration to LCG/AA software in areas were CMS has already deployed solutions
- **Focus on CMS near-term use-cases**
  - Develop/integrate only components with a wide use potential
  - Do not get involved in projects of unclear benefit to CMS
    - Involve more deeply in areas critical to CMS

Examples of “near-term use-cases”

- **Mathlib**
  - Collaboration with SEAL
- **Condition DB**
  - Collaboration with POOL to develop a RDBMS-backend
- **Scripting environment**
  - We will evaluate PyLCGDict
- **Grid Replica Location**
  - Will use Pool FileCatalog as front-end
- **User Level Event Collections**
  - Will use Pool Root-Collections (possibly enhanced by an AIDA interface)

Priorities for 2004

- **Optimization of the POOL streaming layer**
  - Single LCG-ROOT dictionary
  - Improvement of performances
- **Development of a POOL RDBMS back-end**
  - Main client is Condition DB
  - ARDA Metadata layer will also profit
- **Deployment of a MathLib for LHC experiment**
- **Complete AIDA implementation in terms of LCG/AA software**
- **GEANT4**
  - Complete integration between Geant4 and experiment framework
  - Further improve performance and reliability
  - Fully validate physics for LHC use cases
  - GEANE replacement

Concluding Remarks

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

Still a long way ahead of us

- Some critical use cases not yet supported
- LAN and WAN data access/replication still very unsatisfactory
- Tuning of performances will require more work
- Integration with LCG-2 (and Grid services in general) is still problematic

Pool (and application software in general) has never been a show-stopper for CMS Data Challenge in 2004