POOL Status & Release Plan for V0.4

Dirk Duellmann

POOL V0.4 content

Main Changes for V0.4

- "SPI compliance"
  - CVS, #include policy
- SEAL integration
  - will release against early SEAL release
- Dictionary
  - complete the chain from .h via LCG-Dictionary to persistent objects in ROOT I/O
  - support for std::vector and std::list
- File Catalog
  - functional completeness wrt typical production use cases
- Collections
  - typesafe interface to collection meta data (tag attributes)
- and more…

POOL and SPI compliance

- CVS structure has been changed according to the agreed SPI proposal
  - in fact only minor changes to CVS were still necessary
- The #include policy has been adapted
  - Example: #include “FileCatalog/IFileCatalog”
  - Every pool source was affected
- SCRAM BuildFiles have been significantly cleaned up
  - Unecessary inter-package dependencies are removed
  - Dependencies are explicit in each package BuildFile rather than in the top-level POOL BuildFile
    - This was only possible after we dropped CMT support
  - Toplevel POOL SCRAM configuration now only needs to be changed by the POOL release manager
  - Developer changes are constrained to their package subtree

POOL and SEAL

- The V0.4 release will be integrated with SEAL
  - Dictionary - but also MessageService, Exception base, etc…
- POOL V0.4 will require a SEAL release in the installation area (expected later this week)
- Both projects will share a common SCRAM ToolBox and SCRAM installation area
  - => any future compiler/platform dicussions need to be done on LCG level rather than in an individual project
- SCRAM provides the cross project magic
  - include pathes, library pathes, indirect dependencies
- On Source code level
  - only minor changes (namespace seal) as most sources where just moved between the two repositories
- Still needs careful planning to not make developers stuck during the merger

Data Service (Object Cache)

- Significant Redesign
  - Support reference counting to allow automatic cache memory management
    - user reads objects (automatic allocation in the cache)
    - all references to the same object refer to the same cache copy
    - once the last reference is destroyed the cache space is freed by POOL
- Now support two working models
  - eg LHCb : all objects in a dataservice will be deleted explicitly by the experiment framework
    - no Refs on the application side exist after a transaction commit
  - eg CMS : cache object may have different lifetimes and come and go as required by the application
    - Refs may continue to exist (and stay valid) across transaction boundaries

FileCatalog Browser

- First GUI prototype available in contrib
  - mainly addressing pool developers and experiment admins
- End user access to be provided mainly via higher level abtractions than files (eg event collections)
  - close interaction between POOL and PI projects
- Explorer like interface to browse catalog by either logical or physical file name
  - basic support for renaming logical or physical file name
  - atattching subsets of files…
  - …but first release will come up read-only by default
- Query interface for file meta data will come as soon as provided by core File Catalog component
  - scheduled for POOL V0.5

File Catalog Components

- Move to newer version of the EDG back end (edg-rls-client 1.1.3)
  - now supporting file meta data
  - used internally by POOL but not yet exposed for user meta data
- Merged command line interface for registration, renaming, deletion for all back end implementations
  - based on component interface
  - checking against concrete experiment use case for completeness
  - aim for a functionally complete interface by V0.4
    - to be picked up for early integration by CMS
- Prepared a questionaire which is currently passed through the experiments for review

File Catalog and Meta Data

- So far the POOL file catalog only maintains lfn<->fileID<->pfn mappings
  - lfn typically contains concatenated meta data for redundancy
  - meta data queries for larger (> 10**6) catalogs are inefficient if based on sequential scan over all catalog elemnts
  - propose to provide poper meta data support based on RDBMS storage and queries with V0.5
- Discussions about use cases just started

Storage Service

- The crucial POOL component for complex type support
- Several meetings between POOL and SEAL developers have taken place to agree on the detailed interaction
  - between extraction-dictionary-storage manager
- Prototype code based on current SEAL developments already working for std::vector and std::list
  - in the developer area …
  - workarounds for the connection with RootI/O are currently used – expect a cleaner solution with time

V0.4 Release Plan

- Platform & external package has been defined
  - will pick up edg-rls 1.1.3 and the most recent Root release
  - otherwise unchanged wrt to POOL V0.3
- Aim for internal release first incorporating SEAL components by Wed 19th (next week)
  - Goal: confirm that CVS/SCRAM/#include changes are correct
- Component Code Freeze: 21st
- POOL V0.4 release still scheduled for 28th
  - not much room for slippage – as we need to prepare for CHEP '03

Release Platform

- Still only one supported platform RH7.3/gcc3.2
- Test releases to insure that POOL code does compile with a larger set of compilers can and will be done as soon as the external libraries and SEAL exist
- Propose to couple request for additional supported platforms to concrete development or production activities by the experiments
- May need to setup a prioritised list and a LCG calendar for this

General Issues

- Platform and external package discussion needs to be scheduled early into the overall release schedules
  - Constraints inposed by one global SCRAM ToolBox may become an issue if more and more projects join and their sucessive releases span
- Some difficulties to maintain regular commit/tags from the developers
  - started scheduling internal releases to minimise the risk of big bang integration
- Meeting schedule de-synchronised as a side effect of ROOT tutorials
  - otherwise the tutorials were much apreciated
  - right length and content – Thanks!

Bejond V0.4

- Meta Data and Query interface to the File Catalog
- Overall Performance optimisation
  - Propose to make V0.5 mainly a bugfix and performance release
  - More efficient/compact Token implementation
- Extensions to the full chain
  extraction->dictionary->storage manager->RootI/O
  will probably still be required
  - Test suite of example classes submitted by the experiments to POOL/SEAL will guide the priotisation
- Need to stabilise data representation on disk
  - Need to know data types and data lifetime for data which is planned to stay acessible for
- Currently preparing POOL release plan for rest of 2003
  - To be presented to the PEB on February 25th
  - Any input welcome!

Summary

- A significant set of functionality extensions will be provided with POOL V0.4
  - mostly working already for their developers ;-)
- POOL is more and more integrated with (and dependent on) SEAL and SPI services
  - good progress on defining the configuration issues
  - should be SPI compliant with V0.4
- POOL V0.4 seems on track for a release on time
  - More confidence after the SEAL release integration