

31/3/2005

Ioannis Papadopoulos, CERN IT/ADC

POOL Status and Plans

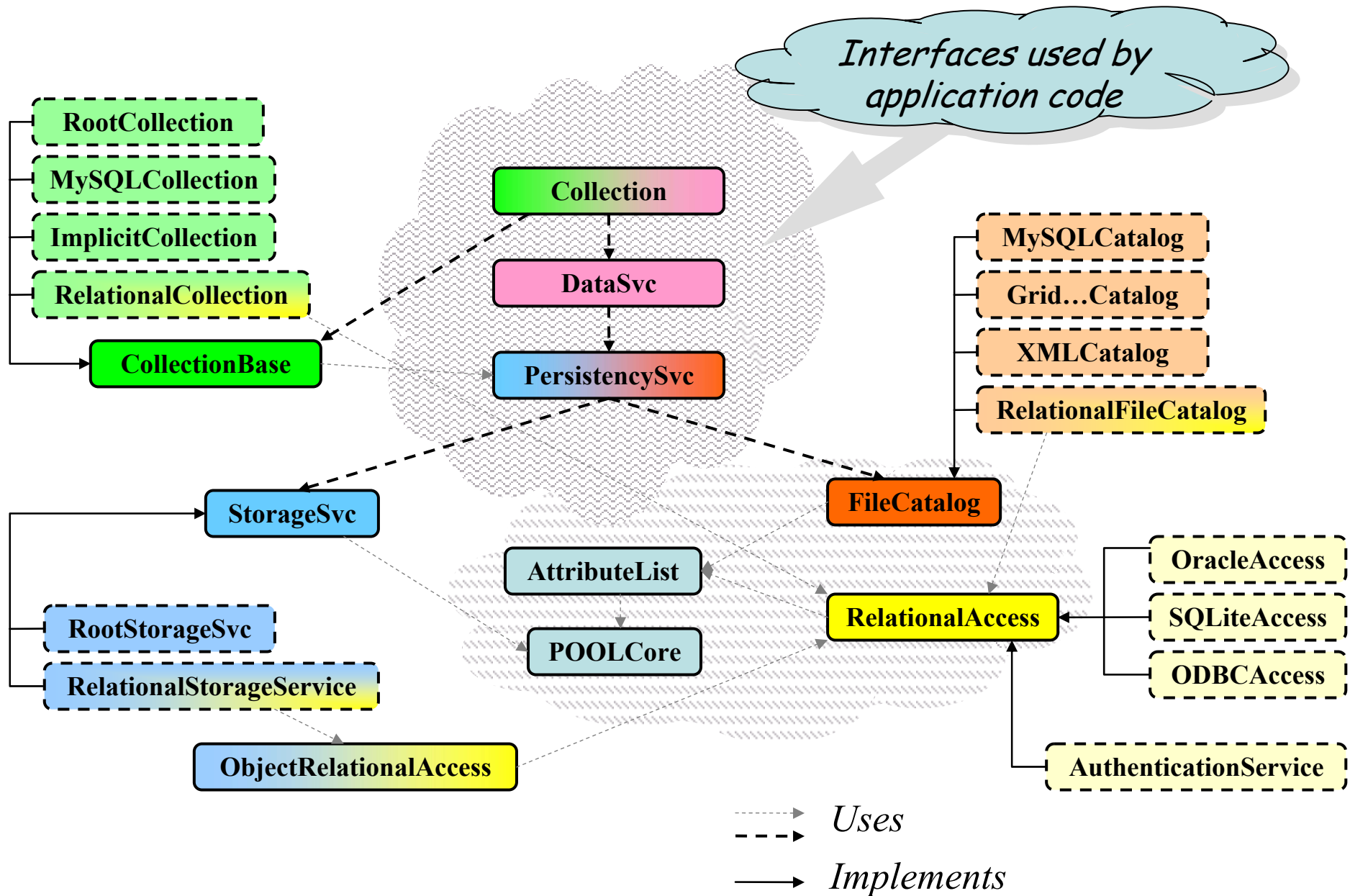
Current developments and
proposed workplan for 2005

LCG Applications Area Internal Review

People

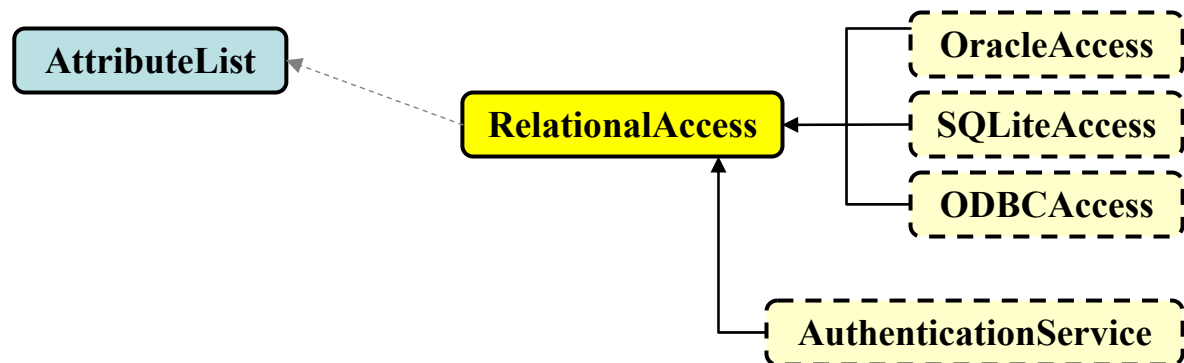
- The POOL team (lead by D.D.)
 - Storage Manager
 - Markus Frank, Giacomo Govi, I.P.
 - File Catalog
 - Zhen Xie, Maria Girone
 - Collections
 - Kristo Karr, David Malon, Helmut Schmucker
 - RAL and RAL-based implementations
 - Radovan Chytrcek, Zhen Xie, Giacomo Govi, I.P.
 - Infrastructure
 - Giacomo Govi, Radovan Chytrcek, I.P.
- Nobody contributing 100%
 - People moving back to experiment integration and physics services
 - ...faster than originally foreseen
- The “invisible” support team
 - Our beta-testers, requirement-providers from the experiments (Vincenzo, Dave, Vakho, Yulia, Sasha, RD, Peter, Nuno, Bill, Mike, Lassi, Jens, Saima,...)
 - Members of the SPI, SEAL, ROOT and COOL teams (Alberto, Andreas, Eric, Johanne, Manuel, Pere, Rene, Fons, Andrea, Sven, ...)

POOL Domain Decomposition



The Relational Abstraction Layer (RAL)

- More details in presentations of AA meetings:
 - <http://agenda.cern.ch/fullAgenda.php?ida=a043380>
 - <http://agenda.cern.ch/fullAgenda.php?ida=a051526>
- Requirements collection, design, implementation, testing, deployment, integration and usage in production software of experiments within less than a year.
- Ongoing functional review of the abstract interfaces



RAL functionality highlights

- C++ API for SQL-free, technology-neutral access to relational data (inserting, deleting, updating and retrieving rows)
- SQL-free description and management of existing or new relational schemas
 - Only C++ types in the API
- Support for bulk operations, client-side caching and SQL variable binding
- Separated connection and authentication mechanisms:
 - oracle://dbhost[:port/service]/schemaName
 - mysql://dbhost[:port]/dbName
 - sqlite_file:///directoryName/file.db
- RAL is NOT an interface for issuing arbitrary SQL statements to a database (such as perl or python DBI)
- RAL enforces "best practices" in database programming
 - Eg. Variable binding at row inserting are done by the plugin.
- RAL developers interface to the providers of the database services.

RAL backends

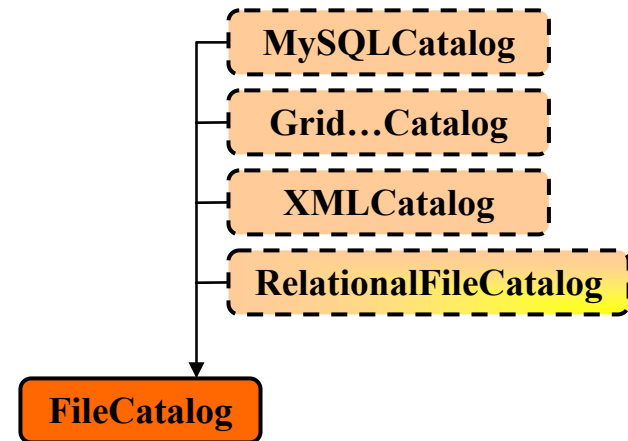
- Oracle
 - Based on OCI 10g client
 - Tested against 10g and 9i servers
 - Implements the full set of the RAL interfaces and their semantics
- SQLite
 - A light-weight embeddable SQL database engine
 - File-based (zero configuration, administration)
- MySQL
 - Implementation based on the 4.0.x MyODBC driver
 - Will soon provide an implementation based on the MySQL C-API
- Authentication services
 - First implementation
 - XML-based
 - Environment variables
 - Grid-aware implementations will come this year (3D project)

RAL in use and outlook

- First clients of RAL
 - Implementation of all the three POOL base components (FileCatalog, Collection, StorageSvc)
 - Development of COOL (see Andrea's talk)
 - ATLAS geometry database
 - CMS online conditions
- Outlook
 - Ongoing internal review to facilitate best the needs of COOL, the 3D project and specific experiment applications
 - RAL will start being released and packaged independently of the rest of POOL

File Catalog

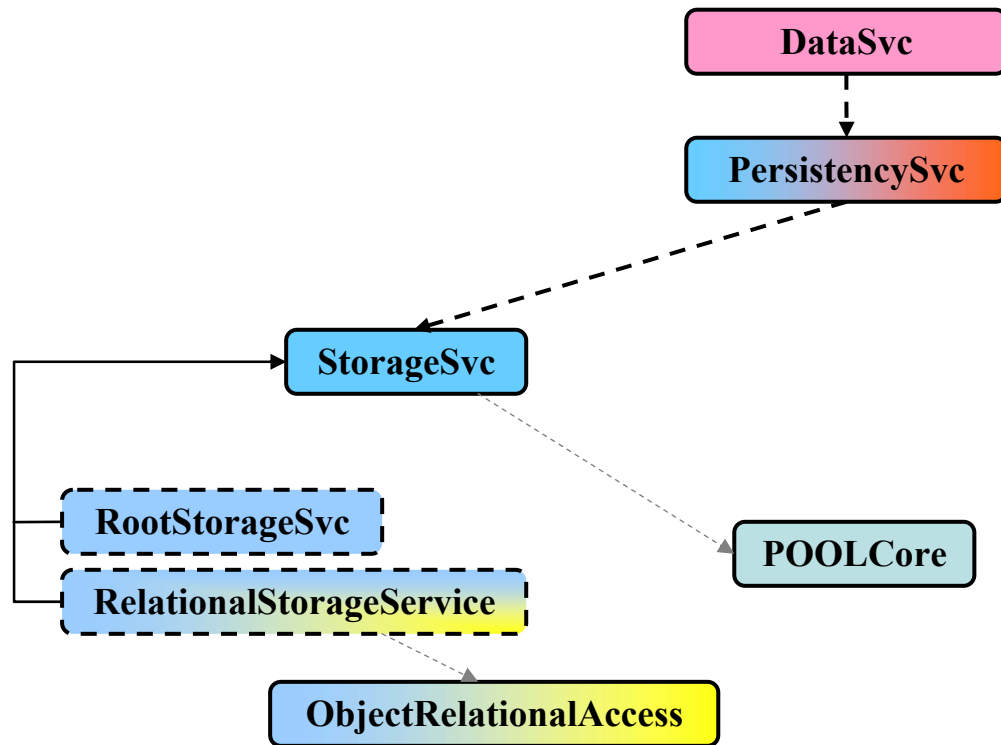
- Goals achieved in 2004:
 - Support of the LCG-2 RLS
 - Composite Catalogs
 - Single Writer, Many Readers
 - Definition of new set of interfaces splitting the file lookup and meta-data management functionalities
 - Automatic recreation of catalogs from a set of files



FileCatalog Implementations

- RelationalFileCatalog
 - Based on RAL, first release already with POOL 1.7
 - Enhanced functionality wrt native MySQL-based implementation
 - First performance/scalability tests performed by CMS (using the Oracle backend) **with very encouraging results!**
 - Currently working on the adoption of the new interfaces
- Current implementations
 - EDGCatalog will be frozen with POOL 2.0 (no further development)
 - MySQLCatalog will be gradually replaced with RelationalCatalog (MySQL backend)
- New grid-aware catalogs
 - Based on Globus, Glite (Fireman), LFC
 - Implementations of the new split interfaces
 - **To be released withing the coming weeks (main target for POOL 2.1.0)**

Storage Manager



- Reasons to celebrate during 2004:
 - Migration to ROOT 4 (RootStorageSvc)
 - Support for schema evolution
 - Definition and implementation of a RAL-based implementation of the StorageSvc interfaces
- Deferred for 2005:
 - On demand dictionary loading
 - Follow the evolution of the SEAL dictionary (Reflex)

RootStorageSvc (developments)

- Integrated with ROOT 4 (Main target for POOL 2.0)
- Demonstrated that the schema evolution cases handled by vanilla ROOT can be handled by POOL as well
- Allows for object storage in branches of ROOT trees
 - Contribution of Bill Tanenbaum, CMS.
- Currently addressing issues related to cases of data written with ROOT 3 (POOL 1.8) evolved and read using ROOT 4 (POOL 2.0)

RootStorageSvc (outlook)

- Immediate priority
 - Migrate to Reflex and Cintex
- Impact from the SEAL+ROOT merging:
 - Package is expected to become very thin, a simple adapter of the StorageSvc interfaces

The RelationalStorageService

- Activity originally launched to serve two main use cases:
 - Reading existing relational data as objects through the POOL framework
 - Store condition/configuration/calibration, event meta-data in an RDBMS through the POOL framework
- Main challenges faced:
 - Bridge the differences between the object and relational worlds
 - Resolve object identity
 - Map C++ aggregations to relational constraints
 - Need for software to define/store/materialize object/relational mappings.
 - Allow for user-defined views of the relational data as objects and vice-versa

RelationalStorageService (current capabilities)

- I/O of most of types of objects:
 - STL containers (nested containment as well), POOL references, 64-bit integer types
 - Not yet supported: C-arrays, pointers or C++ references
- Two types of POOL containers
 - RDBMS_HOMOGENEOUS (equivalent to ROOTTREE)
 - RDBMS_POLYMORPHIC (equivalent to ROOTKEY)
- Consistent concurrent write accesses to the same container
 - Extensive tests performed using the Oracle RAC setup in IT
- Implicit handling of several schema evolution cases
 - Change of type and order of data members, removing data members, changing name of base class(es), moving data members within class hierarchy,...
- User-customizable object/relational mapping

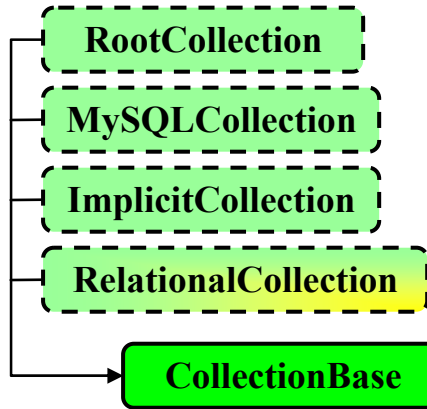
RelationalStorageService (outlook)

- Pending items:
 - Populating POOL containers from existing relational data
 - Underlying software infrastructure is ready
 - Requires 2-3 person-weeks to complete
 - Implement the “multicolumn object binding” functionality (StorageSvc semantics)
 - Required for the implementation of the StorageSvc-based collections

Storage Manager: Open items

- Ref<T> support in vanilla ROOT
 - Work in progress (currently testing Cintex)
 - Expect to benefit from the usage of Reflex/Cintex
- Frontier integration
 - Prototype exists based on proxy implementation of the IPersistencySvc interface
 - Need to help the Frontier developers implement the relevant StorageSvc interfaces instead

Collections



- Collection cataloguing
 - Implementation based on the existing FileCatalog interfaces and implementations
 - Support for collection logical identification
- RelationalCollection
 - RAL-based implementation
 - Enhanced functionality wrt native MySQL implementations
 - Integration testing will start soon
- Not yet done (for reasons beyond the control of POOL)
 - Collection plug-in for ROOT
 - Integration with ARDA services

POOL Documentation

- Identified as POOL's weakest aspect in the last AA review
- Component description documents and POOL User Guide automatically built as part of the release procedure from the docbook fragments of the various packages
- Enforced implicitly simultaneous tagging of source code with up-to-date documentation
- SCRAM-, CMT-free tutorials from the web
- There is still room for improvement, especially in the documentation of the developer-level interfaces

Build and release procedures

- Our build system allows for several platforms to be built and validated (running of the tests) in parallel to each other and the documentation compilation in the release area
- Regression testing added in the standard validation procedure
 - Real cases from the experiments
- Support for QmTest added
- Support GCOV maintained
- Continue the use of frequent internal releases which allow users to give early feedback (even though there have been some cases of abuse...)
- Regularly building on eight platforms: rh73_gcc32(_dbg), rh73_gcc323(_dbg), slc3_ia32_gcc323(_dbg), win32_vc71_dbg and osx103_gcc33

Immediate top priorities

- Release of the FileCatalog with the new “split” interfaces
- Migration to Reflex of the Storage Manager packages (and to Cintex for RootStorageSvc)
- Implementation of the reviewed/improved RAL interfaces
 - Upgrade of the RAL plugins
 - Migration of the RAL clients to the new interfaces
- Full adoption of the SEAL component model
 - Also required for the proper implementation of the StorageSvc-based collections
- Complete the RelationalStorageService implementation

Summary and Conclusions

- There has been a year with big development efforts
 - Many thanks to the experiments for providing early feedback
- There are a few but big developments ahead of us
 - mainly a consequence of the SEAL evolution and its proposed merge with ROOT