

LCG Application Area Internal Review

September 2006

1. INTRODUCTION

From September 18 to 20, the LCG Application Area held an internal review of the software and infrastructure areas.

The Review Committee concentrated on the design, implementation, performance and experiment experience with SPI, POOL, ROOT and Simulation software and infrastructure. It aimed to examine the overall coherence of the software, to identify real and potential problems and risks, and make recommendations on the evolution of the Application Area projects

A particular focus of the review was on the degree to which the projects have the functionality needed by the experiments in view of the approaching LHC start-up. With production software now in heavy use within the experiments, the experiments were able to provide substantial feedback on their current experience with AA software, and to give input on the direction and prioritization of future work.

The software developers of the projects made detailed presentations allowing an in-depth technical assessment of the design and implementation of the software. The experiments have given feedback on the use of the AA software and presented their view of which functionality is needed beyond what is available, and on their preference as to the implementation schedule.

The Review Committee was composed of the following:

Sunanda Banerjee, Predrag Buncic, Paolo Calafiura, Marco Clemencic, Gerhard Raven, Maya Stavrianakou, Elisabeth Sexton-Kennedy, Vakhtang Tsulaia, Michael Doser

2. GENERAL COMMENTS

The Review Committee acknowledges the large amount of work completed in preparation for the review. The detailed and thoughtful presentations were extremely helpful for the committee. The committee has been impressed by the progress made in many areas since the last review, with several projects approaching maturity. Specific comments to the individual projects are given below, but two areas of overriding importance and common to most projects require attention:

- Communication between the Application Area and the experiments appears to work very well; nevertheless, the committee encourages expanded communication at all levels, involving the experiments as well as other stake holders (such as middleware developers) wherever possible, both within existing fora (Grid deployment, AF), as well as proactively, on a one-to-one basis, wherever an activity (carried out or planned) might have an impact on other areas. Users on their side are encouraged to submit reports through Savannah, the appropriate gateway to direct developer-to-developer contact.
- As the manpower resources in the coming year continue to decrease yet further, several of the projects have become or are close to becoming critical. Increased involvement of the experiments, also in decision processes on prioritization of subprojects or requested features,

may help to confront an increasingly difficult situation. In particular, for projects that are not taken up by several experiments, it would be desirable if involvement of the sole beneficiary were to be strengthened. In this context, the committee also encourages an optimization of the resources allocated to the different projects, in particular for those that have reached a critical level of manpower.

3. SPECIFIC COMMENTS TO THE PROJECTS

3.1 SPI

The goal of the project is to provide a software infrastructure and a process for the software projects that are being deployed in the LCG Application Area and for the LHC experiments directly.

General remarks

The continued and impressive progress made since the last review is demonstrated by the widespread adoption of SPI supported tools and systems by several experiments and other LCG projects. Most recommendations made at the time of previous reviews have been implemented with visible beneficial effects especially in what concerns the functions of the central librarian. This project is mostly mature, but the manpower situation is critical. Increased automation of the build / release process would be beneficial, although the committee acknowledges that resources would be required to implement this. There is a need to identify or establish a co-ordination body for the interface to middleware/fabric for testing and standardization.

The following subsections address specific concerns and recommendations.

Bug tracking

Savannah is a success story. The customizations brought to Savannah in collaboration with the original tool developers are commendable. The widespread use of Savannah by AA projects and experiments testifies to its strengths and further potential, and has been universally adopted.

Recommendations:

- Geant4 bug tracking is carried out by KEK, but there is a need for a gateway mechanism to allow single-entry points for end users. Although this is not a specific recommendation for SPI, implementing such a gateway would profit from support from SPI.

Configuration management

As future support for SCRAM is uncertain, migration to CMT appears as a reasonable option; however, such a migration should carefully be carried out, involving the experiments. LCGCMT needs further work and the role of configuration files should be clarified: the ability to build and distribute ONLY what one needs and to declare dependencies at such granularity that no global recompilations is triggered is essential; however, the implementation of this should be balanced against the additional workload it would entail. Timing of this migration is a concern, and input and help from experiments (CMS) is needed to minimize disruptions.

Recommendations:

- The migration from SCRAM to CMT will need to be supervised by the AF;
- Minimize package dependencies and distinguish between intrinsic package dependencies (packages required to compile/link) and tool dependencies (packages required for e.g. testing);

- Provide different types of distributions for different needs; use LIM meeting to address requirements such as distribution granularity, build types etc
- Ensure representation of AA in certification discussions: Linux certification, deployment of alternative compiler etc.

Release, packaging and installation

Further steps to speed up the procedure are very desirable, and are expected to have a dramatic effect on the experiment's build procedure, bug detection and overall turnaround.

Recommendations:

- Nightly builds matching the experiment's build schedule are encouraged
- Automation of the checking of test results
- Implementation of a continuous testing system
- Package and platform retirement should be looked at in the light of real usage

In particular, given the lack of manpower in the SPI area, it is recommended that the AA adopt existing solutions already deployed by some of the experiments for nightly builds and automated testing.

External software

Given the extent of adoption and use of the SPI-supported external software installations and services, several improvements should be undertaken. It must be noted that the procedure for selecting supported packages, platforms/compilers is no longer just an AF issue. Other LCG areas, especially Grid Deployment are directly concerned and affected, and should be incorporated into the information exchange process.

Recommendations:

- Consider automating the build procedure
- Maintain version publication for all external packages; this is important for experiment integration as well as for several Grid middleware tools
- Establish collaborative mechanisms with other LCG areas, especially Grid Deployment

Quality assurance

An automated high level testing facility that would constantly run high-level functional tests (some of which would/could be provided by the experiments) should be implemented, although with the present level of manpower, it is not clear how feasible such an initiative is. A higher degree of quality testing would nevertheless reduce the necessary manpower in the long run.

Recommendations:

- Automate update of build pages accessible via WWW

Training

Recommendations:

- The highly successful Python course should be reinstated with examples from recent developments (e.g. Geant4 Python interface), with external experts

Cernlib

A relatively small fraction of the full CERNlib is still needed by a certain number of packages in the AA, but generates continued support needs.

Recommendations:

- Review what is still required, and by whom (e.g. generators), as well who can maintain, package and distribute in view of reducing the level of required support. It should be indicated clearly that components outside this reduced set will not benefit from future support.

3.2 POOL/COOL/CORAL, Persistency

The POOL project has been created to implement a common persistency framework, allowing storing and retrieving of experiment data and metadata in a distributed and grid-enabled environment. POOL and CORAL have reached a mature phase, but future developments are constrained by the requirements of stability, maintenance, as well as bug fixes. In the case of CORAL, the committee endorses the recent adoption of additional databases (e.g. Frontier).

POOL

The POOL project is now in a mature phase. Three LHC collaborations (ATLAS, CMS and LHCb) use POOL as the base for persistency. Each of these collaborations has produced large numbers of events (tens or hundreds of millions); with a total volume of data stored in POOL of about 400 TB. No major POOL related problems were encountered in the production and access to the data. POOL has successfully met the requirements of the collaborations using it. Most recommendations of the previous reviews have been implemented.

POOL still needs to support schema evolution to a greater degree. Non-minimal changes to data members are not handled, and although workarounds for this exist, these require much effort in maintenance and incur performance costs. Clarification of the degree to which users require support for schema evolution for the relational backend of POOL to a similar level of functionality as that provided by ROOT is needed. This situation requires efforts both on the part of the developers as on the part of the users, and the committee encourages an open dialogue on this question.

POOL and catalogues

CMS is currently using the TFC (trivial file catalog), which is a real pool catalog; it might be worthwhile to compare this catalog to others in terms of benchmarks to assess the performances of different file catalogue implementations.

COOL

The reviewers note favorably the progress made on COOL and its documentation, but note that significant functionality is still required by experiments. Given the reliance of LHCb and ATLAS on COOL, and the current critical manpower situation for the project, the committee is very concerned, and encourages ATLAS in particular to pursue its attempt to identify an appropriate developer.

A further concern relates to remaining scalability issues, although it is understood that these are under investigation within the limits determined by the very low manpower.

Recommendations:

- Communication with the experiments on the subject of required functionality, feature enhancements, prioritization of the available manpower and timetable of any changes is of paramount importance to minimize the disruption to the experiments in the critical period of LHC start up.

3.3 ROOT

Much progress has been made, and the regular release schedule and good documentation are appreciated; the excellent responsiveness of the ROOT team is greatly appreciated. The committee also acknowledges the progress made on the ROOT/SEAL merger. The committee hopes that external packages (e.g. Roofit) will soon reach a similar standard.

With regards to packaging, one should investigate whether a splitting of ROOT in packages that can be released separately (e.g. CORE, I/O, MATH, ...), allowing standardized package-by-package build procedures, and more rapid deployment, as well as the ability to build minimal applications, could be feasible.

CINT/Reflex migration: this is very much needed, and must be completed. Planning is now on a more solid basis, but this is still a very ambitious project, and care must be taken to ease (via binary compatibility) the uptake by the experiments.

PyROOT: several projects are using it, but maintenance issues (a single experiment-funded developer) must be addressed

Plugin manager convergence – this continues to be important but is unfortunately not yet very advanced. Experiments must be involved by participating in setting the schedule and defining the design in the AF.

Math: much progress has been made, in particular in the new fitting package; there is a need to reduce duplications, improve naming conventions (e.g. TRandom2), deprecate and give warning for duplicated classes via AF, and make full use of the current C++ standard (e.g. Tstring vs std::string).

I/O: Issues in I/O caching and tree merging/splitting have been well addressed.

This area exhibits signs of symptoms of communication problems (SMatrix, virtual inheritance in EDM, ROOT I/O speed for ATLAS): improved communication at all levels should be pursued, and should alleviate the observed difficulties;

Care should be taken to avoid duplication of functionality between ROOT and POOL projects;

The committee encourages the campaign to improve thread-safety, but is not convinced that with the current resources, major efforts in the area of multi-threading should have high priority.

PROOF: this is currently the only parallel interactive analysis environment, and as such is worthy of further development: steps towards transforming PROOF into a “product” (schedule, docs) are encouraged. Actively seeking collaboration with experiments to establish a framework/prototype in which this can be carried out is a necessary step in this direction. Proof has potential spin-offs (xrootd, asynchronous de/compression...).

GUI/graphics: this activity is appreciated, but efforts should not interfere with those focusing on ROOT as core package for the I/O

Documentation: the committee encourages porting of existing documentation or providing links to the native ROOT documentation by generation of doxygen compatible files.

3.4 Simulation

Simulation is a crucial component for data analysis at the LHC. The Simulation Project in the LCG AA software area contains several activities, which have been discussed in turn.

Generator services

Much work has been done since the last review. Several activities have carried on or have emerged in the subproject, e.g. the database for generator data MCDB, and Generator Level. The MC generator repository subproject GENSER is already actively used by ATLAS, CMS and LHCb. Granular packaging of GENSER and modularization to permit downloading of individual packages now allow GENSER distribution to be handled efficiently. Support for HepMC has been considerably improved.

Recommendations:

- The change of project leader is an opportunity to review the communications channels between authors and the project. The mandate of the project should be clearly communicated both to authors and to users.
- Development of a validation suite is recommended

- The large number of supported platforms and versions leads to a significant load; a clear policy on deprecating and/or dropping older versions should be defined and communicated.
- As a large part of the code base relies on Fortran, development and tuning of C++ generators should be encouraged.
- The developer code repositories should be mirrored at CERN to reduce the effort required on the part of LCG AA developers. The repositories should provide access to the LHC experiments, taking care not to disrupt the contributors repository.
- MCDB is used by CMS, but not by the other experiments. In the context of decreasing manpower, a significant involvement of the beneficiary in the project is encouraged.

Physics validation subproject

The collaboration between the experiments and the validation subproject has significantly increased during the past 18 months. Geant4 physics list have been compared in detail to LHC test beam data, and a special application for Geant4-Fluka-experimental data comparisons has been developed, although this application has not yet been used in production.

Recommendations:

- A continuing common effort together with the experiments to validate the packages against test beam data and simple benchmark processes.
- Continuation of the Fluka validation, especially of the hadronic physics, against test beam data and Geant4 results
- To install a central repository for the validation results. In particular, archiving test beam data (distributions) for future validation tests is encouraged.
- To continue to address together with the experiments the effect on systematic uncertainties for future measurements, due to the present simulation uncertainties, as observed by the ongoing validation studies.

Geant4

Geant4 has proven that it has reached a level of maturity in terms of performance, physics results, reliability and stability. The (arithmetic) stability of the package is greatly increased, but any issues encountered by the experiments should be used as input to further reduce any instabilities.

The communication and interaction with the LHC experiments is also noted to be very good. Geant4 has become the main simulation engine for the experiments ATLAS, CMS and LHCb. All have expressed their satisfaction with the present state of the subproject, but are concerned by the CPU cost per simulated event. Developments taking place within the experiments to produce a much faster simulation code, albeit at the price of loss of generality, should encourage the investigation of the inclusion of possible short-cuts or parametrizations as alternatives to a complete simulation into Geant4.

The hadronic shower shape is still an issue, although continuous efforts are made by the Geant4 team to improve the situation. Also of concern is the emerging link between quality of the simulation and performance, and steps to improve both areas are encouraged.

Potential issues connected with different strategies on the usage of CLHEP libraries by Geant4 and other LCG applications should be addressed, and may require immediate solutions. Better integration of Geant4 with other products developed by the AA, such as ROOT (analysis, I/O, etc.), is also encouraged by the committee.

Recommendations:

- Geant4 bug tracking is carried out by KEK, but there is a need for a gateway mechanism to allow single-entry points for end users.

FLUKA

The committee notes with pleasure that past issues with FLUKA have been resolved, and that source code is now available for the community at large. This will allow radiation studies to be carried out as needed, and reinforces Alice's use of FLUKA as its main simulation engine. Comparisons of Geant4 with FLUKA should also concentrate on performance aspects, in particular in the case of complex detectors, where little information is available.

Recommendations:

- If a significant increase of FLUKA users were to materialize, discussion should be started with FLUKA developers on how best to meet the needs of deployment and training.

Generic simulation framework project

The simulation framework packages have so far been mainly used internally within the simulation project, although the components are generic and ready for wider usage.

GDML has reached a level of maturity and provides very useful functionalities for various simulation studies, by guaranteeing geometric consistency across application boundaries, even using those that use different simulations engines. Geometry persistence with ROOT has been implemented, and could find a specific use as a possibility to deal with long initialization times in the case of complex geometries.

The MCTruth has split off from Geant4, and an example of MCTruth usage in Geant4 applications has been developed by the simulation framework subproject. Including MCTruth in the Geant4 examples suite is strongly encouraged.

Most of the milestones in the project have been accomplished, and only few short-term activities remain; the manpower concerns of the previous review thus no longer apply.

3.5 Miscellaneous

The experiments are developing their distributed data analysis frameworks and get support from ARDA. POOL and PROOF are AA projects and clients of grid services.

The review noted the absence of a technical forum to discuss client/services issues between the experiments, the AA developers and the grid service providers.

Recommendation:

Organize regular extended Architect's forum meetings to establish better communication and working relationship between the relevant parties.

CONCLUSION

The Review Committee was impressed by the work that has been done since the last review. Most recommendations made at the time have been implemented or are in the plan. To a very large extent – with a few exceptions – the software provided is at the production level required for the LHC start-up.