Response to the LCG Application Area Internal Review Report

Applications Area Project Leaders

Draft 0.2, Feb 23 2004

Preamble

The Applications Area project leaders thank the reviewers for their careful assessment of AA activities as reflected in the detailed and constructive review report they provided to us. Following discussion in the AA projects and the Architects Forum, we present here the AA response to the internal review report. As the response makes clear, the majority of the recommendations of the reviewers have been taken up and are being incorporated in Applications Area plans.

Software Process and Infrastructure (SPI)

The review report is excerpted in italics and annotated with comments. The milestones discussed below assume that adequate manpower will be available. SPI resources are currently under review, as is the collaboration with the EGEE Middleware project.

The services provided by SPI are not used consistently by the groups working within LCG-AA (SEAL, POOL, PI, Simulation), leading to problems with various tools and unnecessary learning efforts.

SPI plans to increase training and prepare a tutorial for its services and tools. This could be achieved in collaboration with the EGEE project, if EGEE will use the SPI infrastructure. EGEE has training of developers is one of its work packages. An SPI/EGEE workplan should be completed in Feb 2004. Target is a Mar 2004 tutorial on the SPI services including: Services Overview, External Software, Savannah admin, Testing Frameworks, QA reporting and policies. The usage among the LCG projects depends largely on the projects themselves. We rely on the other LCG projects (SEAL, POOL, etc) to encourage their developers to follow the LCG standards.

Following agreed rules in a consistent manner would reduce unnecessary inefficiencies and duplication of effort. A central librarian – as suggested by the RTAG – would increase the efficiency significantly.

SPI with LCG, EGEE and SFT is already looking for the person to fill the "central librarian" position. This is given the highest priority. Target date to fill the position is Feb-Mar 2004.

Interaction of the other LCG areas with the SPI infrastructure should be clarified.

We will initiate discussions between SPI and the LCG area leaders. Target is to complete a comparison of the software infrastructures used in the different areas of the LCG in Feb 2004.

The SPI team is encouraged to use CERN-IT services. After satisfactory tests the IT-CVS service should be used as soon as possible.

This was already planned and the validation is scheduled by end of November, the migration will be performed beginning 2004, to avoid problems during the Christmas period. Migration to the IT CVS service should be completed in Feb 2004.

The use of Savannah as project portal is well received. Scalability studies should be made. The good collaboration with the authors should be maintained to avoid future divergence of CERN from the mainstream Savannah.

The Savannah team has now merged back with the open source initiative. Also a lot of improvements asked by the LHC projects are being implemented. The savannah.cern.ch site has now been updated to use the same version as savannah.gnu.org. The scalability of Savannah seems sufficient; the open source installation hosts more than 2000 projects and almost 23000 users, without any problem. We are currently 100 times less demanding.

The web site – in particular with the Workbook format – is very complete and impressive. It will eventually require continuous maintenance by a dedicated documenter as suggested by the RTAG.

Identifying the manpower is of course the problem. EGEE help is being pursued.

The infrastructure for project code documentation with Doxygen is well developed, but the quality varies among the projects. It is recommended to include checks of the documentation in the regular QA tests.

These checks were present in the past but were removed because the errors of doxygen often gave incorrect warnings. Now it has been improved and doxygen checks will be added again to the QA reports (target Jan 2004).

The QMTest tool provides a good framework to use the OVAL and X-Unit tools and should make it easier for the projects to run their QA tests. Fully automated testing is encouraged (nightly QA) to cope with the increasing number of tests as projects mature – eventually a central QA responsible may be required. The existing QA policies must be backed by tools to facilitate compliance.

We will improve the automatic QA reporting tools and make them available to the project leaders and developers in the LCG projects (target Jan 2004). The QA reports based on the information on Savannah will be added to the general Savannah open source at no cost for us. And hopefully will be enriched by the Savannah community.

The repository for external software used by the projects is of professional quality with a good separation of tools, versions, platforms and compilers. However, a transparent decision-making process for provision and maintenance of external software is recommended. The recommendation is to have

- *Well identified "owners" – who requested/need/use a given package*
- *Documented procedure for interacting with users and authors to report/follow up bug fixes*
- *Management of versions and dependencies as well as approval of new external tools via the Architects Forum. With the large number of external tools used, and the impact of changes in the configuration, the version should be modified only when well justified (critical bugs or important new features)*
- *A policy for handling different structures and conventions of external packages in a consistent way (the structure of the package itself should not be modified)*

The external software service will be improved following the recommendations. Especially a clearer description of the support process is going to be implemented and documented (target end 2003). A document (end 2003) will clearly specify that (1) policy version/dependency management is done via the AF and only with a clear process and (2) structures and conventions of external packages follow those of the package itself.

While the documentation on how an external package was installed is very welcome, simple scripts for doing the compilation/installation, as well as installation logs will be helpful for outside institutes and users. Many tools come with QA tests for validation of the build and installation – the use of them should go in the installation script and the logs. Simple QA tests should be provided for external tools without such an essential validation service.

The validation of external tools will have to be improved; maybe also some tests should be run by SPI. Early in 2004 SPI will put in place a validation mechanism for the tools that SPI provides, for the testing and QA. By May 2004 external tools should have a set of validation tests in place such that SPI can validate new installations of external packages. These tasks will be part of the assignment of the central librarian position.

Concerning the distribution of the LCG software more interaction is needed with LCG grid deployment area to develop common distribution tools (Pacman is a candidate). Distribution in a world without network (CDROM) should not be forgotten. There is some concern about the granularity of the existing distribution. The installation and distribution needs to be customizable for specific components and platforms, for simple use as well as for development.

The distribution system of SPI was meant to be a temporary solution. It will be reassessed together with the grid deployment and the LHC experiments and following the work that is being done in other projects. SPI will follow and (as manpower allows) participate in GDA activity in establishing software distribution requirements, standards, tools, services etc. SPI will be involved in the discussions on pacman.

The build system and infrastructure requires some simplification and development of consistency. ... A central librarian would be of significant help.

The central librarian role will include these duties.

The nightly build model seems insufficient. Developers need simple tools to test compilations immediately on their preferred as well as on other platforms. There are doubts that the results of the nightly build are thoroughly looked at. Is the system supporting all platforms – including Windows? There is some concern about the long term maintenance of the NICOS tool.

The maintenance and usage of NICOS is being clarified. BNL is committed to the support of NICOS for both ATLAS and LCG so long as it is found to be a useful tool. The insufficiency of the nightly build model is acknowledged and a complementary service providing on-demand user builds on the LCG platforms is needed. Adding this functionality to NICOS is under consideration. EDG has its own system that is also under review for the EGEE project. SPI will participate in this discussion. A single system for LCG and EGEE is our clear goal.

The concerns with the build functionality of SCRAM are acknowledged. The investigation of the use of autoconf/automake is strongly supported. The main benefit of autoconf/automake is that they are "standard". This investigation should be continued in close collaboration with the experts in the LHC experiments. If the conclusion would be to move to autoconf/automake the change from SCRAM to autoconf/automake should be coordinated with the experiments.

The evaluation of the autoconf+gmake based system (appwork) is continuing. Experts on existing tools in the experiments (SCRAM, CMT) were invited to evaluate the system and give their input. Substantive input was not received, which we did not find particularly surprising; achieving “close collaboration with the experts” on competing systems seems unrealistic unless such collaboration was supported by the experiments, which it is not. We see the essential evaluation to be that of the build experts in SPI and the LCG projects, whose requirements must be met. This is underway. If the autoconf+gmake system is adopted, SPI is going to take it over when the author will consider the initial development debugging done. The target is to reach a decision in spring 2004.

We understand that it would be an advantage if external contributions to LCG software development did not require installation of non-standard tools. However, the tools used by ATLAS, CMS and LHCb – SCRAM and CMT – provide additional functionality, and a very careful study how to consistently cover these needs is recommended. The long term strategy should be based on autoconf/automake – it should avoid the use of several tools (autoconf & SCRAM & CMT) and ensure optimal support for the experiments.

In order to facilitate integration by the experiments, the configuration files for the experiment build systems (SCRAM and CMT) should be provided.

LCG AA software development and usage has reached sufficient maturity that the LCG AA development/usage environment itself provides a good testing ground for the extent to which a build tool meets LCG AA requirements (which are different from, and smaller than, experiment requirements). This is the basis on which the evaluation is proceeding: we believe we know our requirements and can evaluate tool compliance in our environment.

The decision has been taken (and practice put in place) to support CMT and SCRAM configuration files for the use of the experiments irrespective of the system finally chosen for in-house use in LCG AA, in order to ensure optimal support for the experiments. We agree that the LCG AA strategy should involve AA usage of only one tool.

SEAL

We completely agree with the statement that the success of SEAL should be measured by looking at how widespread its usage is in other LCG components and LHC experiments. We fear that is a bit too early to make this measurement taking into account that we effectively started only one year ago. In addition, since SEAL is made of a number of fairly independent parts, we do not expect that any customer will ever be using all the parts. Nevertheless, we have taken note of these concerns and we will collaborate with customer projects to integrate the already existing pieces before further development.

Foundation Libraries

- Setup tutorials, user-guides and help developers - Actions will be taken and will be added in the program of work.
- Remove unnecessary dependencies in external packages - Actions will be taken.
- Convergence between the SEAL and ROOT plug-in manager - We do not see technically a possible convergence until other basic functionalities (dictionaries, build systems, etc.) are in place to facilitate this task. No action will be taken for the time being.

Math libraries

- Concerns about the future of Minuit, GSL and CLHEP - We do not share these concerns about the future at this time. We feel that we need still to maintain, evolve and provide added value to these packages for a number of years.
- Careful testing to guarantee reliable physics results - Actions will be taken and will be added in the program of work, especially for GSL.
- Provision of a coherent set of libraries including dictionaries - We fully agree with this vision and we are working currently in defining a concrete program of work towards it. We also agree with the recommended priority given by the reviews.

Dictionary

- Concerns about the size of dictionaries - Actions have already been taken since the review and the sizes have been reduced fairly easily by a factor 5-6.
- Encourage unifying the dictionary with ROOT/CINT - We are currently having discussions in this direction. The first step will be the agreement of a common C++ reflection API reflecting the ISO C++ standard and an implementation based on CINT to be tried out. We expect, on the other hand, that ROOT will be corresponding and making efforts in adapting to use the external common dictionary.

Framework

- Discussion with the experiments to evaluate interest in framework – We agree with this recommendation. We will only further develop this area if there is a firm engagement from at least 2 experiments.

Scripting

- Evaluate the interest in the interoperability between Boost and Swig before any work is committed - We agree with this recommendation.
- Continue with the PyLCGDict automatic binding – We agree with this recommendation and we are currently finishing the new implementation mentioned during the review.
- Seek feedback from the experiment physics community in usability of python in interactive analysis – We are going to take this recommendation very seriously and collaborate with the experiments on any opportunity in this direction.

POOL

The POOL project thanks the reviewers for their very constructive comments and will take on all of the individual requests in the work plan for 2004.

Documentation

We have released the first version of a POOL user guide end of 2003 with POOL release 1.4 and will go through a complete restructuring of the POOL documentation as of POOL V1.6. The overlap between different documentation components (User Guide, Design Documents and Workbook) will be minimised and a system to generate the documentation from individual xml files under the control of the work packages is being set up. This system will produce web based documentation and pdf files for offline reading from a single source. We will also concentrate on reworking in particular the documentation of the POOL ref class and include hyperlinks to the corresponding SEAL documentation concerning dictionary generation.

Schema Evolution

POOL will go through an effort to prove that the schema evolution capabilities of the ROOT backend are in fact preserved also when using POOL. We will set up a larger suite of evolution test cases (matching the capabilities of ROOT) and confirm that in all these cases POOL data is correctly evolved to a new changed definition of the transient class. This will be done in close collaboration with the ROOT team and the experiments to define which tests should be performed for this milestone.

ROOT access to POOL functionality

As requested by the review we will develop together with the ROOT team a set of plugin modules for ROOT which will allow enhancing ROOT with the additional functionality provided by POOL. In particular we plan to address the navigation using POOL refs within ROOT as an interactive analysis shell and allow to access POOL based collections from inside ROOT.

Optimisation and tuning

POOL will schedule a dedicated release cycle in addressing a consistent performance optimisation. We plan to cover two distinct areas: the CPU and storage overhead of POOL files compared to ROOT files with similar content, and secondly the efficient use of mass storage systems from POOL (optimisation of back end operations like file opens and minimisation of global system resources like open file handles and sockets).

Files and Collections

The authorisation and authentication versus file, collection and catalog access and updates with be revisited as part of the POOL ARDA combined work package.

Simulation

The Simulation Project thanks the committee for their constructive comments. We agree with almost all the recommendations and are incorporating in our work plans those which are not already present.

Generator Services

The review recommends setting up a testbed for the comparison of event generators. As the reviewers point out this requires a common environment, and while work is in progress (GENSER, standard event formats) it has to progress further before a common validation framework and testbed can be established. The project has established a milestone to have a proposal for an event generator validation framework written by July 2004.

The HepMC format for generator event data in memory has been adopted as recommended by the review. For persistent generator event storage, the approach recommended by the review has been agreed upon: one format, probably XML-based, for low statistics, and a second format, POOL/ROOT based, for high volume statistics. The specifics of these formats are currently being worked out, with a workshop on the topic scheduled for March 2004.

Simulation Physics Validation

The review encourages the experiments to increase their contributions in simulation physics validation, recognizing this as a unique opportunity for young physicists to contribute now to a physics project. We could not agree more. We have been actively soliciting such contributions from the experiments since the early days of the project. We have seen experiments criticize the low level of such contributions in the project while failing to offer contributions themselves.

Generic Simulation Framework

The review noted a lack of strong interest among the experiments in a common simulation framework, while noting that the physics validation project requires the ability to compare multiple simulation engines with the same detector configuration. The recognition of this in the project led us to focus the work for the present not on developing a common simulation framework but on applying existing tools to meet the specific, immediate needs of the physics validation project – in particular to set up an infrastructure supporting comparison of Geant4 and FLUKA in test beam simulations by making use of Geant4 based geometry descriptions and FLUGG (as recommended by the review). This work is going forward now.

The review encourages the further development of the existing ALICE/ROOT-developed VMC as a generic simulation framework. This work is continuing, outside the scope of the applications area as it has been up to now.

The review report states that in the medium and long term the VMC should become the main tool. The present level of interest among the four experiments does not support such an assertion at this time. The review recommends that a discussion among the experiments and simulation projects take place to evaluate interest in a framework and so guide priorities, development program and manpower. We believe this is the proper course. In order to inform such a discussion, the AF agreed in October that following the initial round of Geant4-FLUGG-FLUKA-based work supporting test beam validation, the next objective would be to implement a simple means of exporting Geant4-based detector geometries to the VMC (approximate timscale summer 2004) such that the VMC can be evaluated by the experiments with their own presently existing geometries, and in particular can be evaluated against the test beam validation just concluded.

Simulation tools – FLUKA

The review came shortly after a FLUKA workshop the preceding week and the pressures of priority work following the workshop prevented the planned FLUKA talk from taking place. The FLUKA team apologizes that this situation prevented them from participating in the review as originally planned. The simulation project has a good relationship with FLUKA, with the head of the FLUKA project Alfredo Ferrari involved directly in the LCG simulation

project. Reports on FLUKA activity can be found in the simulation project leaders meeting minutes (posted on the web) and most recently in Alfredo's talk to the applications area meeting Feb 18, 2004.

PI

First we would like to thank the reviewers for their detailed and in-depth evaluation of the documents we have provided. However, there are some statements in the report which could benefit from further clarification and guidance.

The sentence "WP2 seems to be now in the mandate of SEAL" could profit from a statement whether the reviewers think this is ok, or if it is recommended that this activity (Analysis Environment) should be (again) part of the PI project. As there is no comment on this in the SEAL section of the report, clarification on this would be very much welcome.

The original proposal for WP3 has, to our understanding, a slightly different and actually complementary scope relative to ARDA: ARDA is dealing with issues around the border layer between the GRID middleware and software needing to use this middleware, while the WP3 of PI was intended to define the interface between this layer and the physicist. Again a recommendation on whether (and how) to continue this proposed work and if and how to interface to the ARDA project was expected.

In the context of WP1 the reviewers might have overlooked the fact that there is not only a "working system based on ROOT", but also a native implementation covering more of the AIDA interfaces. Moreover, we have provided to the user a way to make use of both implementations in parallel in the same application, enhancing the flexibility and choosing implementations according to specific features required by the user. To have this flexibility is a user requirement which we got as feedback from LHCb, Atlas and CMS, and is documented in the web page set up for the review.

We would like very much to address the stated "performance issues", however the statement as such does not give any guidance on what specifically is to address. More quantitative statements, together with the corresponding user requirements for it, would ease the work on this significantly. We have to heavily rely here on the reviewers, as the user feedback we got so far stated that there is no performance issue in the software we provided.

We observe that the statement "the system is considered as an option for LHCb and ATLAS..." does not correspond to the feedback we received from these experiments. LHCb/ATLAS are requesting PI to move fast to the new AIDA 3.2 and, to the best of our knowledge, they intend to interface to ROOT histograms using the AIDA binding provided by PI. What is considered optional at the moment is just the proxy layer.

The statement "However, for interactive analysis a direct use of ROOT ...", again does not correspond to the feedback we are receiving from ATLAS, CMS and LHCb who are interested in using the AIDA objects directly from their interactive Python-based environment. It is our understanding that all these three experiments intend to use AIDA in a Python or C++ environment and eventually use ROOT or another visualisation package such as Hippodraw to visualize and further manipulate histograms. It is also our understanding that there is a long standing request to be able to manipulate AIDA objects from the ROOT/CINT prompt. To our knowledge, work on this inside ROOT has not started.

The statement "... ROOT must be able to access data stored with POOL." seems to us misplaced in this section of the report, as it clearly addresses work to be done in ROOT and as such this is outside the scope of the PI project.

Finally it escapes us what the correlation between the PI project as a whole and ARDA is. As stated above there is a clear relation between the WP3 and ARDA, but clearly none of the other aspects of PI are in the scope of ARDA.

Conclusion

Reviewing the main issues identified by the review:

- SPI, build and configuration tool (SCRAM versus autoconf/autobuild)
  - Following recommendation to proceed with autoconf+gmake evaluation. Agree that LCG AA should use only one tool. CMT and SCRAM configuration files will be supported. Recommendation that long term tool should be autoconf+gmake based noted.
- POOL/ROOT schema evolution
  - Being addressed with priority as recommended.
- POOL collections
  - Will be addressed in the broader context of POOL+ARDA where analysis needs will be better represented.
- SEAL libraries management
  - Reducing dependencies on external packages is being pursued. Convergence with ROOT on the plugin manager is not seen as realizable in the near term. It is much harder than the areas of convergence presently being pursued (dictionary, math libraries), which present enough challenges for the moment.
- SEAL dictionary, framework and scripting
  - Dictionary convergence with ROOT is being pursued. We agree experiment engagement is crucial; it will be a priority. Scripting recommendations will be followed.
- Simulation framework
  - Work has been rescoped in light of low interest in common framework, with initial focus on physics validation needs as recommended. Premature to declare the VMC as the main tool for all experiments in the long term. Plan to enable evaluation of the VMC by all experiments to inform a discussion on longer term needs and plans.
- PI, the project may need to be redefined in the context of ARDA.
  - Agreed.

The analysis of the specific comments on each sub-project shows that most of the projects interoperate with ROOT. In order to integrate ROOT in the architecture and to optimize the performances of the services the committee recommends that the technical collaboration between the different sub-projects and ROOT evolves from a client/provider mode to a modular cooperation as agreed in the blueprint RTAG.

This is being pursued, particularly in the areas of dictionary convergence and a common math library project.

The software projects have ambitious work plans they will need the full support of the management to meet their goals.

... and, at least as important, the support of the experiments as well.