PH-SFT questions about the NA61 software upgrade proposal



Migration strategy

•Outline plan for transitioning from current to new software.

Develop new core, develop DSPACK I/O wrapper to interface old clients with new core, replace the core, gradually update clients.

•Show timeline and steps involved.

Core meeting of two months in March/April. Most of the implementation of the core is expected at that time.

Are the various components be treated independently or must everything be done in one step?

New data structures and wrappers need to be done in one step, then other components (e.g. simulation) can follow.

•Explain role of wrappers giving some details of design and complexity.

Two main wrappers: the DSPACK I/O wrapper for the new core, and geometry conversion from GEANT3 to TGeo. The latter is easy, as it uses established tools - indeed, it already works. The former - the design would be to stick with the bare minimum of the DSPACK API required to provide correct I/O, with complexity mostly in identifying this minimum and deciding how to handle other functionalities.

•What is impact of any increase in package dependencies (i.e. for both old and new software) during the migration?

No significant increase is expected to take place. Everything the old software needs is already there, on the other hand one of the major planned features of the new software is to make use of widespread solutions and tools.

ROOT, STL, CLHEP (?), boost (?), Eigen (?), SQLite (?)

Core software

•Are changes planned to the client server processing architecture?

In general, no. Indeed, the proposal states we want to keep things modular. We would like to change the way the two communicate with each other, though.

•What application framework has been chosen? Pierre Auger? Which others were considered?

We have considered ROOT4STAR and Aliroot. One big advantage of the Auger framework is that we've got people with working experience developing it willing to work on SHINE software; that said, we should by no means blindly copy everything regardless of what source we choose.

•What core libraries will the new C++ implementation be based on? i.e. explain how you can benefit from re-use of existing software packages

Related to the package dependencies issue above and mentioned in the proposal. STL - very common, well known worldwide. ROOT - de-facto standard of our field. Same for Geant4, plus here we can re-use some things from GEANT3.

Auger Framework depends on the following libaries: ROOT, GEANT4, boost, CLHEP, Xerxes, mysql, cppunit

•Give some details on critical components such as the plans for the event data model, geometry model etc.

Unified data model (STL + ROOT), TGeo.

•What has been developed already (or can be reused from elsewhere) and what still needs to be developed?

STL, ROOT, Geant4. Own development: NDST as a pre-development of the new data model.

Miscellania

•What are the platform dependencies now and in future and what role do you see for using virtualisation for managing these?

At present: Linux/x86 (or Linux/amd64 with 32-bit compatibility mode), Scientific Linux 5, GCC, PGI Fortran. In the future, we will get rid of PGI Fortran as well as hopefully support Linux on other CPU architectures (or at least be able to run natively in 64-bit mode). As mentioned in the proposal, virtualisation is an important aspect of keeping our environment stable - given how much effort has been devoted to migrating from SLC4 to SLC5, it would be great not having to do that again whenever IxPlus gets upgraded. Moreover, using virtual machines makes it easy to obtain an identical software environment even outside CERN, where native operating systems may be different (non-SLC Linux, Mac OS X, even Windows).

•*Have alternative approaches been considered for ensuring long-term maintainability of the NA61 software?*

As far as run-time maintainability is concerned, I would say virtualisation is the best option we have got - even at IxPlus, what OS it runs is beyond our control. On the other hand, maintainability of the code itself is likely to be helped by having it more organised and consistent, introducing documentation procedures etc. - which will help in the middle run when it may still be necessary to modify out software.

10 February 2011

•What would be the impact of not doing this project?

As mentioned in the proposal: we have already been forced to introduce some kludges, workarounds et cetera, and as new components get added to NA61 hardware and software it would become even less of an organised, wellunderstood system and more of a black-box patchwork. Especially given the progressing loss of knowledge of how this thing works in the first place.

•Do you have specific requests for support from the SFT group?

We do not have very specific requests from PH-SFT in the sense that special tools etc. are not needed to developed to fulfill the project. However, we are open, we encourage participation of anybody who may be interested. We also count on PH-SFT as an advisor.