The code management tool, CMT, and the repository system, CVS, seem to be appropriate choices, given the growing size of the code and the number of involved developers; CMT (as CVS) is used by other long term experiments, so the support/maintenance should not be an issue. User support is in a good shape, thanks to the recent efforts, particularly through help pages and dedicated checkout scripts. VCMT seems to be a reasonably stable tool in the Microsoft environment, though it would probably be good to have some help for Toby in maintaining it. The mismatch between availability and interoperability of tools that access the repository still requires resolution. In particular, more robust tools need to be developed for Linux users (and perhaps for Solaris users as well).

Gaudi is a modern event oriented programming framework developed within strong collaborations such as LHCB and ATLAS. The core group plan is to migrate the whole simulation-reconstruction-analysis chain within this architecture. This is encouraged. There is good progress in the definition of TDS objects, the persistence mechanisms and the Recon algorithms. The recent effort to improve documentation is appreciated, however the version and releases need to be defined, in order to clarify and identify to which framework the development is dedicated: Gismo/glatsim, Gaudi, G4, etc! This migration involves some other issues:

- a non negligible overhead of s/w engineering, both for the core group and physicists, while we have limited manpower, and very short term milestones.
- Gaudi is not a finalised tool, and is evolving; compatibility and interfacing with modules such as G4, or GLAST adapted persistence mechanisms need extra effort; it is not clear to the committee what could be realistically done “at home”, and what should be waited for to be done by other collaborations. This could be clarified given the near term milestones.

Root is also an evolving tool. As underlined by the core team, backward compatibility may become an issue, especially for large data sets.

The status and future of Gismo is clearly related to the status of the migration to G4; until G4 is in place Gismo provides the only means for performing simulations. The migration to G4 has started quickly for the balloon support and is progressing well regarding the hits definitions. How this work is included within the overall G4 effort was not clear to the committee, neither the plans nor the manpower involved. The recommendation is to settle the organisation for G4 migration asap, including a better defined extra-USA component (with clearly defined responsibilities). Regarding the platform support, there could be an issue for G4 maintenance on Windows NT; this should be investigated with the G4 collaboration. As the collaboration favours the elimination of any dependence on Gismo in the near future, it seems unnecessary to the committee to go through the migration of Gismo under Gaudi. Nonetheless few upgrades (heavy ions, de/dx) need to be considered for the PDR and especially the balloon support.

The current method of geometry description has to be changed for the G4 migration. Other strong reasons to change the current method were mentioned as well. The requirements for a new geometry description method were clearly defined and a variant of XML language is proposed (development by ATLAS); this evolution is recommended by the committee, while underlying the need for the definition of an instrument and calibration database, and its interaction with the geometry description files.

Current plans are to provide strong support for Windows and Linux with the possibility of providing some support for Solaris if it turns out that significant Solaris CPU is made available to GLAST at SLAC. As stated above, that support on the different platforms must be better integrated and more tools need to be developed for Linux users. The distribution of binaries could be envisaged for non-developer users.

One of the major concerns of the core group is adequate manpower. Given this concern, reasonable milestones and a reasonable schedule seem to be in place but this is one of the more difficult areas to assess currently. Certainly improved co-ordination between the various subgroups would benefit the core software group as well.