Computing Frontier I4: Software Development, Personnel and Training

Peter Elmer (Princeton)
David Brown (LBNL)
Ruth Pordes (FNAL)

Organization

- We organized periodic meetings for discussion of specific topics, typically with one preparing notes on a specific topic for a given meeting + general discussion by participants

- Active participants: David Brown (LBNL), Peter Elmer (Princeton), Ruth Pordes (FNAL), David Lange (LLNL), Liz Sexton Kennedy (FNAL), Craig Tull (LBNL), Rob Kutschke (FNAL), Gregory Dubois-Felsman (SLAC), Daniel Elvira (FNAL), Robert Hatcher (FNAL)

- In addition many of the above discussed one on one with members of the community and reported back

White papers

- We also have three white papers

- "R&D and upgrade" needs driven by processor hardware architecture evolution - P.Elmer, P.Wittich, K.Stenson, S.Rappoccio

- "A Vision on the Status and Evolution of HEP Physics Software Tools" - P. Canal, D. Elvira, R. Hatcher, S.Y. Jun, S. Mrenna (FNAL)

- SLAC Geant4 vision - M.Asai et al

Goals and specific recommendations

- The discussions we had among ourselves and with others had for the most part three main themes. We have called these out as the major "goals" we see in the coming years.

- We also have a larger number of more specific "recommendations", which support the goals

- The note for the CpFI4 subgroup is in draft form (in the Google Drive area) and we are iterating with people towards finalizing it.

Goal 1

- To maximize the scientific productivity of our community in an era of reduced resources, we must use software development strategies and staffing models that will result in products that are generally useful for the wider HEP community.

Goal 2

- We must respond to the evolving technology market, especially with respect to computer processors, by developing and evolving software that will perform with optimal efficiency in future computing systems.

Goal 3

- We must insure that our developers and users will have the training needed to create, maintain, and use the increasingly complex software environments and computing systems that will be part of future HEP projects.

Reccomedations - Software Mgmt, Toolkits, Reuse

- Continue to support established toolkits (Geant4, ROOT, ...)
- Encourage the creation of new toolkits from existing successful common software
- Allow flexible, reliable funding of software experts to facilitate transfer of software and sharing of technical expertise between projects
- Facilitate code sharing through open-source licensing and use of publicly-readable repositories
- Consolidate and standardize software management tools to minimize cross-experimental “friction”

Recommendations - New Hardware Architectures

- Significant investments in software are needed to adapt to the evolution of computing processors, both as basic R&D into appropriate techniques and as reengineering “upgrades”

- New software should be designed, and existing software reengineered, to expose parallelism at multiple levels

- Develop flexible software architectures that can exploit efficiently a variety of possible future hardware options

- Support investigation and development of tools that allow user-level code to run on multiple, diverse hardware architectures

Recommendations - Staffing

- Include software infrastructure, frameworks, and domain-specific detector-related applications early in project reviews

- Integrate software professionals with scientist developers in a collaborative effort to insure the software meets both the technical and physics performance needs of the experiment

- Maintain software professional collaborators as part of the experimental team over the life time of the experiment

Recommendations - Training

- Use certification to document expertise and encourage learning new skills
- Encourage training as a continuing experimental activity
- Use mentors to spread scientific software development standards
- Involve computing professionals in the training of scientific domain experts
- Use online media to share training
- Use workbooks and wikis as evolving, interactive software documentation
- Provide young scientitsts with opportunities to learn computing and software skills that are marketable for non-academic jobs