

HEP Software Foundation

Input from the UK GridPP collaboration.

Introduction

With decreasing levels of effort and increasing diversity of hardware platforms, the proposal for a HEP Software Foundation is timely. Fifteen years of standardisation on x86 architecture is coming to an end and large volumes of code will need to be rewritten to make full use of future processing power. The code needs to run efficiently, adapt to new computer architectures, adhere to open standards, and avoid duplication.

This foundation should help to provide the environment and tools to assist the production of efficient HEP software, often by geographically separate developers. It should foster interoperability between HEP software projects and with software projects from outside of HEP. The foundation should provide a mechanism for interacting with other communities. Skilled developers are required to carry out the development of efficient HEP code; to attract and retain such staff, it is important that there is a well-defined career path is recognised.

Goals

- To help align HEP software projects to promote interoperability, code re-use, common processes and existing open standards.
- To promote the use of relevant software components from beyond HEP where there is a clear advantage in terms of support, function, or synergy.
- To provide a supportive environment for new projects combined with a low barrier for entry.
- To promote the production of efficient code through education/training/best-practice and community guidelines.
- To raise the profile and recognition of software production within the HEP community.
- To also recognise the importance of stress/scale testing of software in real use-cases by encouraging and acknowledging those who participate in such activities.
- To be a public face of HEP software interacting with other communities and promoting our achievements.
- To promote this work to funding agencies and provide a framework that can help attract funding.

Achieving the goals

In order to achieve these goals the Foundation should:

- Ensure that code is open source and advise on the range of acceptable licenses.
- Promote interoperable pieces of HEP software by agreeing interfaces/standards and promoting component and interoperability testing.
- Suggest, but not mandate, coding and documentation standards.
- Provide a forum for the HEP experiments to discuss their needs. This forum could identify common requirements to guide existing projects, seed new ones, and stimulate collaborations.

- Provide (optional) tools and processes to facilitate code development. These might include:
 - Source code repositories.
 - Automatic build and test suites.
 - Community building and communication tools (wikis, email lists etc).
 - Binary repositories for different operating systems.
 - Issue tracking.
 - Etc.
- Organise training courses and to publicise courses being given by others. This will be especially important as the hardware ecosystem and programming languages becomes more diverse.
- Recognise and support a “best of breed” approach that, at times, may see the parallel development of more than one project for some period.

Governance and membership

The governance of such a foundation should be as lightweight and transparent. It should not become a project that attempts to dictate what should and should not be developed (that is between the projects and their funders) and individual projects must retain the control over their own work. The governance structure should be nucleated by the key stakeholders, consisting of the larger software projects and user groups. Membership should be open to any HEP software project willing to share their code and collaborate with other HEP software projects. Although, projects will be expected to provide an appropriate level of quality control.