

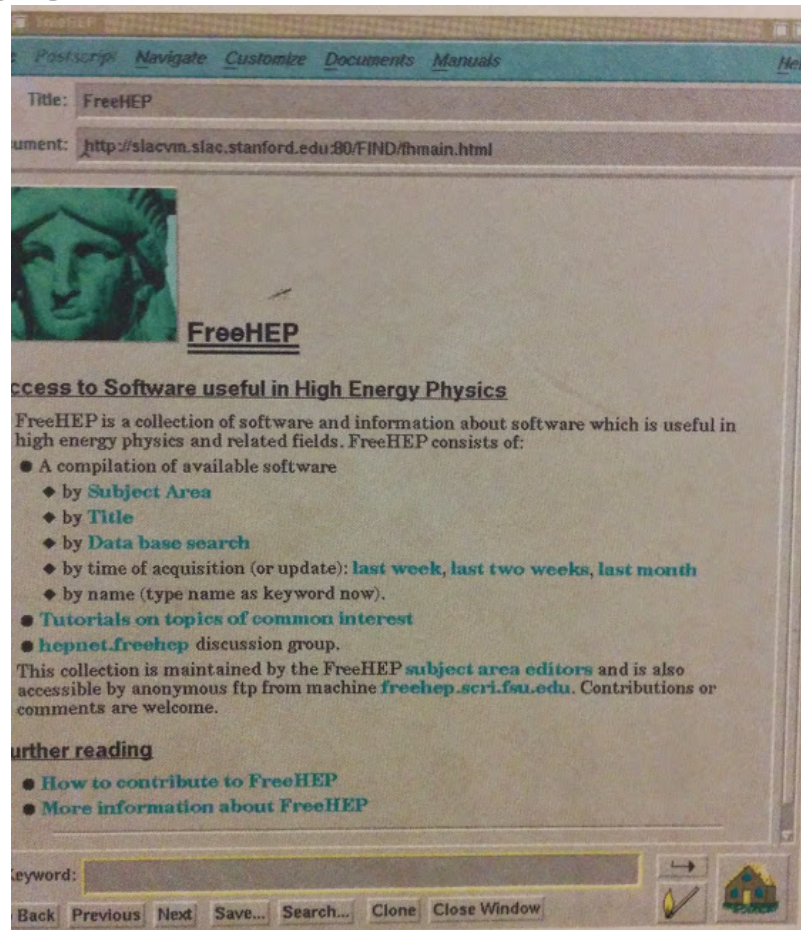
FreeHEP lessons learned

Tony Johnson
HEP Software Foundation meeting
SLAC, January 2015



What is/was FreeHEP?

- Effort to publicise/share HEP software
 - Founded in 1991
 - Used SPIRES database
 - perhaps first database driven web application
 - Curated listing of HEP software
 - Subject editors for ~10 topic areas
 - ~150 entries
 - Links to shared ftp download area
 - Searchable -- based on keywords, titles
 - Usenet discussion newsgroup

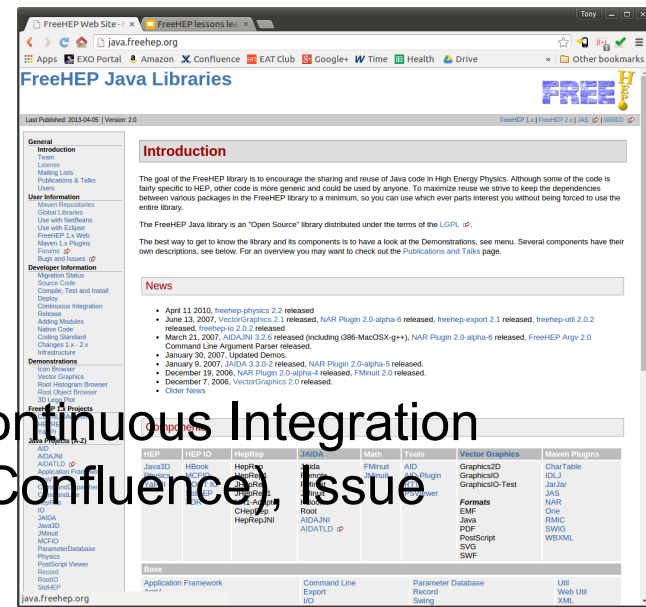


What worked/what did not

- Curators initially helped get library started
 - Enthusiasm varied over time, not easy to replace curators
- No mechanism for removing items, so over time hard to tell what was active and what was not
- Other than a database, FTP site and newsgroup did not provide any tools to developers
 - Limited community involvement



- Library of (primarily) Java based HEP software
 - Core libraries supporting a variety of topics
 - GUI, 3D, vector graphics, physics, fitting
 - Applications based on these libraries
 - WIRED event display
 - JAIDA histogramming toolkit
 - JAS analysis toolkit
 - Provided:
 - Source code repo (CVS, Subversion), Continuous Integration (CruiseControl, Jenkins), Forums, Wiki (Confluence), Issue Tracking (JIRA), Project web site



What worked/what did not

- Worked:

- freehep.org (rather than freehep.slac.stanford.edu)
- Modular components, take what you want, no lock in
- Open source model, contributions from within HEP and outside
- Adopted by experiments (BaBar, ILC, CLEO, Fermi Gamma-Ray space telescope, ...)
- Parts of freehep are embedded in commercial and open-source projects, linux distributions ...

- Didn't work

Licensing issues

- FreeHEP was licensed under LGPL
 - After discussion with SLAC tech transfer dept
- Later we were contacted by companies which would/could not use LGPL
 - Could not change license without getting signatures of all contributors (impossible by then)
- Other people took parts of our code and repackaged under different licenses
 - Caused massive confusion
- Did person who contributed code really have right to do so

Support Model

- Supporting and maintaining software needs effort
 - Reuse requires
 - good documentation (installation, use, porting)
 - time to answer questions (dumb or otherwise),
 - ability to travel to meet users and evangelize
 - Time to fix bugs, set future directions, mediate disputes
 - Prevent unnecessary forking
 - HEP seems particularly fond of reuse by “cut and paste”
 - HEP users want “institutional support”
 - Maybe we should have used `freehep.slac.stanford.edu`
 - Infrastructure needs maintenance, upgrades
- We were (and still are) unable to get funding to support software beyond immediate needs of specific experiments

Personal Conclusions

- Encourage development of modular components
 - Avoid framework/language/technology lock-in whenever possible
- Choose a single license and encourage everyone to use it
 - Obviously must be compatible with all other licenses, labs, universities, countries (even Wales)
- Adopt a clear lifecycle for projects
 - Incubator/Active/Featured/Archive/Retired
- Funding/supporting scientific software for long-term use is difficult
 - If HSF develops a solid reputation it may be able to lend legitimacy to member projects; This will take time to develop
- Don't recreate things that already exist:
 - code repos, mailing lists, plan for change (when GitHub becomes Sourceforge)
 - Cross-platform continuous integration engine may be useful
 - HEP wide licenses software like JIRA may be useful

Maybe real problem discovered by SLAC cyber-security

Web Page Blocked

Access to the web page you were trying to visit has been blocked in accordance with SLAC computer usage policy. Please contact SLAC Cyber Security at 650-926-5000 or security@slac.stanford.edu if you believe this is in error.

URL: forum.freehep.org/

Category: abused-drugs