HepForge and HEP software engineering
A new, lightweight development environment
for small HEP projects

Andy Buckley
Institute for Particle Physics Phenomenology
Durham University, UK

CHEP06, Mumbai 2006-02-16
Outline

HEP software development
HepForge
Summary
HEP software development
Before I begin. . .

- I will probably say some things that are controversial!
  In the first half, anyway
Before I begin. . .

- I will probably say some things that are controversial!
  In the first half, anyway
- Please interrupt and challenge me at any point
HEP software development divides into
  ▶ small projects by individuals and small groups
  ▶ big projects, typically by experiment collaborations

In the big projects, there is a rather unhealthy obsession with frameworks

“No silver bullet” — Fred Brooks, 197?

I want to emphasise the importance of small, well-engineered tools
HEP software overview

- HEP software development divides into
  - small projects by individuals and small groups
  - big projects, typically by experiment collaborations
- In the big projects, there is a rather unhealthy obsession with frameworks
- “No silver bullet” — Fred Brooks, 197?
- I want to emphasise the importance of small, well-engineered tools
- Compare & contrast: history of Unix tools vs. history of HEP tools!
An aside on “frameworks”

- In principle no bad thing; in practice may be **heavyweight**
  - Build system and infrastructure (e.g. Gaudi/Athena with CMT)
  - Monolithic (e.g. ROOT)
- Problem arises when doing simple things becomes hard (the “how short is ’Hello World’ ” test)
- Too abstruse: factories, ‘very smart’ references... Should **hide infrastructure**!
- **Monolithic**: tendancy to “place all our eggs in one basket”
- Not enough emphasis placed on
  - *real* modularity
  - common exchange formats and interfaces
- We need to encourage more modular design... for real! (cf. OSS, even without the hype. e.g. StatPatternRecognition)
The importance of software engineering
For large projects

- Physicists often sceptical / condescending about SE
- It is genuinely important
- Why not using standard packaging and build systems? RPM/DPKG/Portage? automake and autoconf?
- Experimental physics (i.e. numbers we publish) depends on good engineering
- Many small sub-projects implementing the same features: another case for modularity
The importance of software *engineering* for small projects

On the phenomenology side (i.e. matrix element calculators, Monte Carlo integrators, general purpose MC, precision MC):

- Still an entrenched Fortran culture — general purpose MC is a relative island of enlightenment (*Herwig++*, *ThePEG*, *Sherpa*, Pythia 8)
- Build systems tend to be convoluted hand coded Makefiles
- \( \Rightarrow \) interfacing to generators harder for experiments
- Encourage good software practice in small pheno projects
Software engineering problems for small projects

- Not using standard build/installation procedures e.g. autotools... SCons?
- Strong dependences on execution/build environment
- Informal quality control, feedback and bug tracking (by email and memory)
- Failure to use version control systems (Quote: “—— is our version control: we email our changes to him and he merges them by hand”!)
- Inadequate documentation: physics is usually documented via arXiv: software manual may be less developed
Getting started

- Everyone has written code that might be re-used...
- ...but few get round to providing it for use by others.
- The open source community has got the hang of it: many excellent free tools
- But!
  - also lots of chaff (usually based on PHP), NB. www.sf.net ≈ 80% garbage, ~1% gold
  - time spent installing them detracts from “physics time”
  - most physicists do not have deep knowledge of Web service config.
- Why painfully re-invent the wheel so often?
Existing development sites

- Good development practice #1: see if there’s already something you can use!
- Existing collaborative devel. sites:
  - FreeHEP: not open to all, facilities?
  - LCG Savannah: confusing, inflexible (my opinion... YMMV)
  - SourceForge: restricted facilities, not HEP specific
- For us, no existing system has an intuitive enough interface, combined with powerful tools and robust engineering.
- This leads us (slightly reluctantly) to HepForge...
HepForge
HepForge

- HepForge is a new collaborative development system
- Online at http://hepforge.cedar.ac.uk
- Features including:
  - Shell access with full set of dev tools
  - Web space (with several convenient features)
  - CVS and Subversion version control
  - Wiki and bug tracker
  - Mailing lists for developers and users
  - Downloads manager
- We’ve done the “boring bits”!
- “Went live” in January: ~6 months of beta activity from e.g. Herwig++ and RunMC before that.

Andy Buckley
CEDAR
**CEDAR: Collaborative e-Science Data Analysis Resource**

Hep Forge is a component of **CEDAR**:

- **JetWeb**: global tuning of Monte Carlo generator parameters
- **HepData**: archival of published experimental data
- **HepForge**: development environment for HEP software
- **HepCode**: centralised repository of pheno code/programs
- **HepML**: set of XML data formats for data sets and MC config

Hep Forge exists (officially) to implement **HepCode**

www.cedar.ac.uk
Who should be interested in HepForge

- Who’s it for? Small–medium size projects.
- Stand-alone preferred but not reqd. Aids re-use.
- Intentions to use standard external systems
  Don’t re-invent or *break expected behaviour* (UI design)
- Intention to document properly
- General responsible development, essentially

In return, HF provides powerful software management infrastructure with minimum entry level and learning curve
HepForge backend (1)
For the interested...

- Python-based with shell scripts for acc. management
- Various Subversion things, e.g.
  - re-write of CVS: almost complete UI compatibility
  - support for symlinks, metadata, dirs, mv/cp!
  - anon read access over HTTP; rw dev access over SSH
  - use it! Migration from CVS is easy.
- ViewVC with multi-site hack (also via SSI)
HepForge backend (2)
For the interested…

- Trac bug tracker and wiki:
  - SQLite backend and SVN integration
  - Moin-compatible wiki
  - excellent tool!
- Web system with post-processing scripts
  - HTML Tidy
  - transparent header and footer handling
- Download manager (personalisable via SSI)
HepForge tour

Home page

CEDAR HepForge

HepForge is a development environment for high energy physics software development projects. Some of the benefits offered by HepForge are:

- Shell account with up to date development tools
- Web page hosting
- CVS and Subversion code management systems
HepForge tour
Home page (W3C auto-validation)

News

HepForge goes live
posted 2006-01-26

HepForge has now "gone live": we've completed all the functionality we planned for the first release milestone. This includes version control, account management scripts, mailing lists, bug tracking and wiki systems, minimal-maintenance web space and robust shell accounts. We're quite enthusiastic about HepForge 1.0: we hope you will be, too :-)

Hosted projects

You can browse the list of projects using HepForge from our projects page.

Last updated: Thu Jan 26 17:01:30 2006
HepForge tour

Project list (1)

HepForge projects

Here is the current list of projects using HepForge to do their development. (We will be introducing keyword sorting of projects in time.)

- **ExHuME**: C++ generator of central Exclusive Hadronic Monte-carlo Events
- **FastNLO**
- **FeynML**
- **Fortran Herwig**: Fortran-based Monte Carlo event generator with parton shower
- **HepData**: HepData database and Web interface development project
- **HepForge**: Project to build and maintain the HepForge system
HepForge tour

Project list (2)

- **HepML**: Proposed interchange formats for MC parameters and HepData records
- **HepTeX**: Collection of HEP-specific TeX/LaTeX packages
- **Herwig++**: IO and steering utilities for HZTool
- **HZTool**: Data-MC comparison histogramming
- **JetWeb**: Web-based system for MC event generator validation
- **Jimmy**: Multiple interations implementation for Herwig
- **KtJet**: C++ implementation of the kT jet clustering algorithm
- **LHAPDF**: Les Houches Accord PDF library and interface
- **Professor**: Tuning tool for MC generators
- **PyFeyn**: Python library for programatically drawing Feynman diagrams
- **Rivet**: C++ re-implementation of the HZTool MC validation tool
- **RivetGun**: Generator interfacing system for Rivet
- **RunMC**: C++ front-end of Monte Carlo models
- **ThePEG**: Platform for using and building C++ event generators
HepForge tour
Subversion/CVS (VC) listing
HepForge tour

Project VC listing (also personal copy via SSI)
HepForge tour

Downloads listing (projects with files highlighted)
# HepForge tour

Project downloads listing (also personal copy via SSI)

## HepForge downloads

**RunMC project:**

Back to project list...

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>packages</td>
<td>1.0</td>
<td>packages-1.0.tar.gz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>runmc_win32</td>
<td>4.0</td>
<td>runmc_win32-4.0.tar.gz</td>
</tr>
<tr>
<td></td>
<td>3.3</td>
<td>runmc_win32-3.3.tar.gz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>runmc</td>
<td>4.0</td>
<td>runmc-4.0.tar.gz</td>
</tr>
<tr>
<td></td>
<td>3.3</td>
<td>runmc-3.3.tar.gz</td>
</tr>
</tbody>
</table>
HepForge tour

Documentation: user guide (note scrollbar!)

HepForge user guide

This is the guide to the services provided by HepForge. Please check here and in the FAQ if you have a problem with the system. If you can’t find an answer there, email us at hepforge@cedar.ac.uk.

Shell account and filesystem

Your HepForge account gives you full Unix shell access to our server, with user privileges. You will have personal webspaces (if you want to use it) at http://www.cedar.ac.uk/~<username> which can be accessed by creating a personal public_html directory containing HTML files: mkdir ~/public_html; touch ~/public_html/index.html should do the trick. These pages will be visible on the Web at http://www.cedar.ac.uk/~<username>/.

Project information

Each project’s filesystem contains a directory called project-config in which various
HepForge tour

Documentation: FAQs (note scrollbar!)

HepForge FAQs

Here are some questions and answers relating to HepForge. Please ask us some more questions to make this document more useful.

Version control

How do I use a HepForge project’s Subversion repository?

First-off, you might want to check the project’s online repository viewer at http://hepforge.cedar.ac.uk/projname/vc/svn/. If you just want to browse the repository or download a tarball, this is perfectly good.

If you want to use Subversion directly, then you have the choice of using the anonymous-read-only system over HTTP or the full read-write system via SSH. To use the full permissions system you will need to have a HepForge shell account and be a registered developer on the relevant project.
**HepForge tour**

**Project Web page: LHAPDF**

**LHAPDF** _the Les Houches Accord PDF Interface_

**Home**

LHAPDF provides a unified and easy to use interface to modern PDF sets. It is designed to work not only with individual PDF sets but also with the more recent multiple "error" sets. It can be viewed as the successor to PDFLIB, incorporating many of the older sets found in the latter, including pion and photon PDFs. In LHAPDF the computer code and input parameters/grids are separated thus allowing more easy updating and no limit to the expansion possibilities. The code and data sets can be downloaded together or individually as desired. From version 4.1 onwards a configuration script facilitates the installation of LHAPDF.

**Contents:**
- Installing LHAPDF
- List of all available PDF sets
- On-line user manual
- A wrapper for C++
- A little bit of theory

**Downloads:**
- Latest released version:
  - 4.2 (full): lhapdf-4.2.tar.gz
  - 4.2:(no pdfsets): lhapdf-4.2-nopdf.tar.gz
HepForge tour
Project Web page: Herwig++

The Herwig++ Event Generator

Overview

Herwig++ is a completely new event generator, written in C++. It is built on the experience gained with the well-known event generator HERWIG. The aim is to provide a multi-purpose event generator with similar or improved capabilities (like angular ordered parton evolution and the cluster hadronization model). At some point the ongoing development of the Fortran version will terminate and Herwig++ will take over.

Herwig++ is based on ThePEG and CLHEP.

Download

Herwig++ 2.0 beta

The Herwig++ 2.0 beta release has been tested with CLHEP-2.0.2.2 and ThePEG-2006-01-31.

The release note.
HepForge tour
Project wiki

Herwig++ Quick User Guides

Installation and running
- HerwigInstallation
- FirstRun

Writing new modules
- NewMatrixElement
- NewDecayer?
- NewAnalysisHandler?

Misc Examples
- Analysis with PAW
HepForge tour
Project bug tracker: milestones

Roadmap

Milestone: **JetWeb Accessing HepData**
Due in 2 months

88%
Closed tickets: 6 Active tickets: 1

Also a ♦ CEDAR Milestone.

Milestone: **JetWeb and HepData development version**
Due in 3 months

100%
Closed tickets: 3 Active tickets: 9
HepForge tour
Project timeline (integrated with SVN)

Timeline

10/02/06:

战略性| Changeset [516] by buckley
| Adding errors as expected ...

战略性| Changeset [515] by buckley
| Making progress, thanks to new use of XSLT 2.0 and the Saxon ...

07/02/06:

战略性| Changeset [514] by buckley
| Getting HzTool? Fortran headers working
### HepForge tour

#### Project bug listing

#### JetWeb Accessing HepData

<table>
<thead>
<tr>
<th>Ticket</th>
<th>Summary</th>
<th>Component</th>
<th>Version</th>
<th>Type</th>
<th>Owner</th>
<th>Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>#2</td>
<td>Create XSL transformer for data HepML → HzTool Fortran headers</td>
<td>hepdata</td>
<td>1.0</td>
<td>task</td>
<td>buckley *</td>
<td>01/11/05</td>
</tr>
</tbody>
</table>

#### Object model and db persistency refactoring

<table>
<thead>
<tr>
<th>Ticket</th>
<th>Summary</th>
<th>Component</th>
<th>Version</th>
<th>Type</th>
<th>Owner</th>
<th>Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>#42</td>
<td>Refactor the object model / db interaction</td>
<td>model</td>
<td>1.0</td>
<td>task</td>
<td>buckley *</td>
<td>17/01/06</td>
</tr>
<tr>
<td>#54</td>
<td>Improve error representation in the object model</td>
<td>model</td>
<td>1.0</td>
<td>task</td>
<td>buckley *</td>
<td>01/02/06</td>
</tr>
<tr>
<td>#46</td>
<td>Consider using SQL double for data values</td>
<td>migration</td>
<td>1.0</td>
<td>enhancement</td>
<td>buckley *</td>
<td>21/01/06</td>
</tr>
<tr>
<td>#26</td>
<td>Separate HepData model from migration code</td>
<td>hepdata</td>
<td>1.0</td>
<td>defect</td>
<td>buckley *</td>
<td>25/11/05</td>
</tr>
<tr>
<td>#53</td>
<td>Use enums for error type</td>
<td>model</td>
<td>1.0</td>
<td>enhancement</td>
<td>buckley *</td>
<td>01/02/06</td>
</tr>
<tr>
<td>#56</td>
<td>Separate stat/sys and</td>
<td>model</td>
<td>1.0</td>
<td>enhancement</td>
<td>buckley *</td>
<td>02/02/06</td>
</tr>
</tbody>
</table>
HepForge tour

Project bug details (1)

Ticket #2 (task)

Create XSL transformer for data HepML -> HzTool Fortran headers

Status: assigned

 Reported by: buckley
 Assigned to: buckley (accepted)
 Priority: major
 Milestone: JeWeb Accessing HepData
 Component: hepdata
 Version: 1.0
 Keywords: Cc: jmb

HZTool requires a Fortran header file for each paper. The existing XSLT stylesheet for the HepML -> Fortran transformation is very incomplete and needs work.
HepForge tour

Project bug details (2)

Attachments

Attach File

Change History

17/01/06 17:55:28: Modified by buckley

- description changed.
- milestone set to JetWeb Accessing HepData.

I've re-written the XSL transformer classes somewhat. The next steps are:

- abstract the re-formatter classes to be more generic: HepMLReformatter, HepMLSpltReformatter, HepMLTextReformatter etc. (need better names)
- provide a superclass/interface common to
HepForge tour
The HepForge registration form! Think about it...
Features still to come

- Keyword and category project metadata
  (for HepCode & general user convenience)
- Web interface to project metadata
  (keywords, description, “pretty” project name. . . )
- FAQ handler (SSI again)
- More Web post-processors:
  - selective email obsfucator (whitelist for pass-thought)
  - Markdown/SmartyPants, ReStructuredText, Moin, others?
Join us!

If you have a re-useable HEP project and you would like to maintain it, please think about using HepForge!

- Current users include: Herwig++, ThePEG, LHAPDF, RunMC, FastNLO, Jimmy, KtJet. . .
- Plus all the CEDAR sub-projects, of course!
- Requirements:
  - has to be for a **re-useable** HEP project
  - not for processor-intensive use
  - commitment to document and support your project
  - encouraged to use standard build procedures etc.
  
  We can and will help with this!

Visit [http://hepforge.cedar.ac.uk](http://hepforge.cedar.ac.uk) to register
Summary
Summary

- HEP *needs* to consider *modularity* and *interfaces* more
- HepForge is a small spin-off from CEDAR’s main thrust (building and operating a MC generator tuning system)
- HepForge will be used to implement the HepCode system
- HepForge is available for HEP software development now!
- Feedback has all been very positive: system is powerful but very easy to use
- If you have written / are going to write a small–medium re-useable HEP application, please consider HepForge!