

The Bazaar Approach to Physics Education

Seth Rosenberg [UMd/CCNY]
Dick Furnstahl [The OSU]

- Problems with “conventional” PER-based reform
→ particularly for upper-level courses
- Development models for computer software:
“The Cathedral and the Bazaar”
 - evolvable systems (cf. physics research!)
 - importance of open-source philosophy
 - implementation (sociology)
- Tools for Bazaar-style development
- Prototype project

Limitations of PER: Upper Level Courses

- Cycle time is long
- Alpha/beta testing population is very small
- Limited and/or outdated connections to contemporary research
- The need to reinvent wheels
- Unrecognized subtleties in the physics
- Curriculum development and assessment is a huge time sink for an active research physicist
- Community acceptance of PER

Two Models for Software Development

- *The Cathedral and the Bazaar* [Eric Raymond]
 - Cathedral → closed source, proprietary development (e.g., Microsoft)
 - Bazaar → open-source development model
 - * massively parallel development
 - * keywords: openness, peer review, free (as in speech, not beer) software
- Development environments
 - Cathedral mode:
 - “... carefully crafted by individual wizards or a small band of mages working in splendid isolation, with no beta to be released before its time.”

- Development environments (cont.)

- Bazaar mode:

- “... great babbling bazaar of differing agendas and approaches . . . out of which a coherent and stable system could seemingly emerge only by a succession of miracles.”

- Advantage of evolvable systems

- “Centrally designed protocols start out strong and improve *logarithmically*. Evolvable protocols start out weak and improve *exponentially*.”

- “Only solutions that produce partial results when partially implemented can succeed. Evolvable systems begin partially working right away and then grow, rather than needing to be perfected and frozen.”

Examples of Open Source Projects

- **Cathedral:** Microsoft Windows 2000 and Office, Adobe Illustrator, Mathematica, etc.
 - note: these are very good products!
- **Bazaar:** Apache, sendmail, Gnu/Linux, Perl, Python, PHP, Samba, MySQL, BIND, . . .
 - Apache runs ~ 50% of world's web servers
 - Perl is behind much WWW "live content"
 - BIND provides domain name service for 'net
 - sendmail is the main email transport software
 - plus thousands of smaller scale projects
- **High quality:** many are "category killers"
- **Complex:** e.g, Linux (complete operating system)

Parallels: PER and Software Development

- Examples of Cathedral vs. Bazaar in PER
 - Cathedral → ordinary textbook development, conventional PER model
 - Bazaar → none as yet (although many projects have this flavor to some degree)
- Development community
 - Cathedral: Microsoft → textbook authors, PERG members
 - Bazaar: Hackers → Physics teachers (and students!) at all levels
 - * large (potential) pool of expertise
 - * willingness to contribute “in spare time”

- Example: Technical Support
 - Cathedral: Pay the vendor for support → support from textbook authors, PERG experts
 - * in principle, reliable support, since “paid for”
 - * in practice, support can be uneven or very limited or have a slow turnaround
 - Bazaar: The Internet! → tap the resources of the physics community
 - * in principle, unreliable since no one in charge or directly accountable
 - * in practice, tremendous resources tapped by web search engines
 - * Infoworld gave its “Best Technical Support Award” to “Linux people on the Internet”

Bazaar-Mode Lessons (Raymond)

Some maxims for open-source development

→ from Raymond's experience and observations

→ each is applicable to PER!

1. Every good work of software starts by scratching a developer's personal itch.
2. Good programmers know what to write. Great ones know what to rewrite (and reuse).
3. "Plan to throw one away; you will, anyhow."
4. If you have the right attitude, interesting problems will find you.
5. When you lose interest in a program, your last duty to it is to hand it off to a competent successor.
6. Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.
7. Release early. Release often. And listen to your customers.

Bazaar-Mode Lessons (Raymond) [cont.]

8. Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone (else).
9. Smart data structures and dumb code works a lot better than the other way around.
10. If you treat your beta-testers as if they're your most valuable resource, they will respond by becoming your most valuable resource.
11. The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.
12. Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.
13. "Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away."
14. Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected.

Implementation of PER-Based Bazaars

- Successful bazaar projects don't run themselves
 - bazaar characterizes dynamics of contributors
 - control at the top is essential to provide quality assurance and to avoid fragmentation
 - “hacker” culture → constraints
- Implementation models for bazaar development
 - Choose from successful models in open-source software development
 - E.g., benevolent dictator (Linux) or voting council (Apache)
 - Difficulties in maintaining a consistent style necessitate a dictator or small council

- **Critical importance of open source**
 - Stallman: “free as in speech, not as in beer”
 - computer code but also curriculum materials
 - materials must be reproducible at all stages, not just in polished, published form
- **Role of PERG members?**
 - initiators and coordinators of “Open Source” PER-based projects
- **Why not just rely on new journal(s)?**
 - pace of conventional PER is glacial
 - participation from very small fraction of physics community
 - threshold for “publishability” is too high
 - * too complete a product or result is required

- How do you make money?
 - figurative (how do PE researchers get tenure?)
 - literal (how do publishers stay in business?)
 - funding based on peer-reviewed outcomes

Pre-Rebuttals

A pre-emptive attack on some potential criticisms:

1. But curriculum materials *are* “open source” by construction! (I.e., anyone can read them.)
2. What about intellectual property rights?
3. Why should someone spend time producing materials that anyone can copy?
4. What about quality control?
5. Won't there be a million versions of everything, many of them wrong, untested, etc.?
6. Is this approach is so great, why wasn't it used before?

Tools for Bazaar-Style Development

- **Why now?** Because of internet + new tools for massively parallel collaborative development
 - Example of new technology changing how physicists work: Los Alamos preprint server
 - Here: some tools that can be adapted to physics education research
 - * Bugzilla — bug-tracking system
 - * Faq-O-Matic — “knowledge base” system
 - * SourceForge — free hosting service

Tools for the Bazaar [cont.]

- Bugzilla (<http://www.mozilla.org/projects/bugzilla/>)
 - sample bug-tracking system
 - A database for bugs that lets people report bugs via the web and then assigns them to appropriate developers
 - “bug” is a generic term for typos, software bugs, requests for enhancements, suggestions for changes
 - Bugzilla priorities bugs, coordinates schedules (“milestones”), maintains to-do lists, and tracks bug dependencies
 - Sample application: tracking errata and updates to curriculum materials

Tools for the Bazaar [cont.]

- [Faq-O-Matic \(faqomatic.sourceforge.net\)](http://faqomatic.sourceforge.net)
 - “knowledge base” system
 - a web-based system that automates an FAQ (Frequently Asked Questions list)
 - highly searchable database of “questions”
 - anyone can contribute
 - * permission system to establish hierarchy of moderators
 - Sample application: supplement to curriculum materials
 - See also: “open content” documentation [e.g., Andamooka (www.andamooka.org) or php manual (<http://www.php.net/manual>)]

- SourceForge (<http://sourceforge.net>)
 - Free hosting service for open-source projects
 - Features: CVS repository, mailing lists, bug tracking, message boards/forums, task management software, site hosting, permanent file archival, full backups
 - * CVS → Concurrent Version System
 - A tool to keep track of changes made by developers working on the same files.
 - * total web-based, secure administration
 - * to set up a project, register as a site user, login, and register your project. That's it!
 - SourceForge is itself an open source project
 - * started to remove obstacles to open source software development
 - * "A small idea that refused to stop growing."
 - Use to host open-source curriculum projects!

Prototype Bazaar Project

- A common PE “itch” → physics simulations
- Simulation software → natural starting point
 - use existing development tools and strategies
- Modern version of CUPS (and similar projects)
 - relativity, E&M, quantum mechanics, . . .
 - implement pedagogy from PER as part of PER research and development cycle
 - build in assessment and feedback tools

- Features:

- decouple hardware implementation (including graphics) from interface from simulation routines from associated curriculum materials
- platform independence
- downloadable (and updates) from the web
 - * entirely open source

- Essential initial conditions:

- a critical mass of initial code is needed
- can start off crude, buggy, incomplete, poorly documented but must have plausible promise
- coordinator has to recognize (and use) good design ideas from contributors
- coordinator sets vision rather than managing contributors → “the best innovators aren’t dictated to: they are turned loose”

- Potential development community:
 - physics teachers at all levels who would like to use the simulations in their classes
 - physics researchers simulating their specialty
 - * cf. Physics 2000 simulations
[<http://www.colorado.edu/physics/2000/>]
 - undergraduate students doing research project
→ great source of computer expertise!
 - hackers interested in physics
 - * **large** community (just read slashdot)
 - some subset → co-developers (e.g., in charge of individual modules)

Summary

- Bazaar-style development addresses:
 - short cycle times → parallel efforts
 - dissemination → same time as development
 - reinventing → reuse/extend existing materials
 - static content → continuous updates
 - time sink → lower barriers to contribute
 - physics subtleties → involve experts
 - community acceptance → get them involved!
- Tools for massively parallel collaborative development are readily available
- Physics education curriculum development and reform have largely operated in Cathedral mode; it is time to try the Bazaar!