

# Teaching Enterprise Open-Source Development

Darnell Gadberry, Lecturer  
UCSD Department of  
Computer Science & Engineering  
[dgadberry@cs.ucsd.edu](mailto:dgadberry@cs.ucsd.edu)

Presented: 1/8/2007 UUASC Orange County Chapter

# Teaching Enterprise Open-Source

- CSE136 - Enterprise-Class Web Applications.
  - A Capstone course open to graduating seniors and MS/Ph.D. Students.
  - A sister course to the wildly-popular CSE125 - Software System Design & Implementation. "The Video Game course."

# Teaching Enterprise Open-Source

- CSE136 Objectives

- Introduce students to professional Open-Source software development tools. (There are some REALLY SMART people out there solving the same problems as you.)
- Encourage collaborative software development skills.
- Teach some useful Unix / Java system administration skills.

# Teaching Enterprise Open-Source

- Challenges

- Few CS students ever write a “real program” before they graduate. (The typical CSE136 project weighs in at > 10K lines of Java Code.)
- Collaboration in higher education is generally discouraged. Breaking the “talking is cheating” mindset is very difficult.

# Teaching Enterprise Open-Source

- Challenges (cont'd)

- Many students have a very difficult time working without a detailed road map.

- Many university CS departments run highly-restrictive computing environments. (For example, root access to servers is not available to faculty members.) This would make CSE136 very difficult. So, I supply the servers for the course.

# Teaching Enterprise Open-Source

- Challenges (cont'd)
  - Success of the course is highly dependent upon the general skill level of the students.

# Teaching Enterprise Open-Source

- CSE136 Mechanics

- Lecture two nights per week. Discussion with the course TA. Weekly team project status meetings with me. Mandatory lab attendance once-per-week.

- Lab session is good for guaranteeing that EVERYONE gets some hands-on skills development.

- Skill-building exercises first 4 wks of Lab.

# Teaching Enterprise Open-Source

- CSE136 Mechanics

- Student teams design and develop a non-trivial J2EE web application in 8 weeks.
- Students are required to use - Apache Geronimo, Apache Struts, MySQL, Solaris 10 X86.
- Each team gets root access to their own Solaris 10 Zone on a shared server at the binaryMedia Data Center.

# Teaching Enterprise Open-Source

- Why J2EE and not .net or PHP?
  - The abundance of high-quality open-source J2EE software.
  - J2EE is a well-documented, well-supported, highly-scalable development/deployment environment.
  - Netcraft claims J2EE is the most popular environment for enterprise development.
  - Because Java really annoys the PHP and .net people.

# Teaching Enterprise Open-Source

- Why Solaris 10 x86
  - Zones provide a PERFECT environment for student development and experimentation.
  - We can easily replace a zone if it becomes completely wedged.
  - Each team gets root access to their own Zone on an X86 server running Solaris 10.
  - Pottery Barn rules - If you break your zone, you are expected to try and fix it.

# Teaching Enterprise Open-Source

- Why Solaris x86 (cont'd)
  - UNIX Rocks!
  - The Solaris administration experience seems to be very beneficial for many students.
  - Even though Operating Systems is a prerequisite for the course, few students have actually gotten their hands dirty with day-to-day UNIX administration tasks.

# Teaching Enterprise Open-Source

- How we use Solaris 10 x86
  - We typically deploy four project zones per server.
- Current server configuration:
  - SuperMicro Pentium 4 - 3GHz
  - 3GB RAM
  - 160 GB HDD
  - NFS server (RAID 5) available for disk-intensive stuff.

# Teaching Enterprise Open-Source

- Solaris 10 x86 configuration
  - The servers live behind a Cisco Firewall. (Actually, a BIG Linksys box...)
  - A library of configuration scripts make setup and housekeeping simple(er).
  - John Clingan of Sun Microsystems has provided invaluable assistance with the zone management scripts.

# Teaching Enterprise Open-Source

- Solaris 10 X86
  - Zone creation and management.
    - Make the default complete zone by hand (create `-b` option in `zoneadm`).
    - Copy the desired packages into the zone.
    - Create additional zones using the `zoneadm -z <zone> clone <newzone>` command.

# Teaching Enterprise Open-Source

- Zone creation by the numbers.  
(The current solution.)

```
• mrsun# zonecfg -z templatezone
• zonecfg:my-zone> create -b
• zonecfg:my-zone> set zonepath=/export/home/zones/templatezone
• zonecfg:my-zone> add net
• zonecfg:my-zone:net> set address=10.47.1.2
• zonecfg:my-zone:net> set physical=e1000g0
• zonecfg:my-zone:net> end
• zonecfg:my-zone> commit
• zonecfg:my-zone> exit
• mrsun# zoneadm -z templatezone install
• mrsun# zoneadm -z templatezone boot
• mrsun# <copy stuff into the template zone>
• mrsun# zoneadm -z templatezone clone team1zone
• mrsun# zoneadm -z templatezone clone team2zone
• ...
```

# Teaching Enterprise Open-Source

- Zone creation by the numbers (Spring '07)
  - Create a ZFS storage pool
  - Create student zones in the zfs pool
  - Make nightly snapshots of each zone for easy recovery.

# Teaching Enterprise Open-Source

- Software Development Environment
  - NetBeans 5.X in Labs and Student PCs.
  - Subversion or CVS repository in the project zone. Access via SSH.
  - Apache Geronimo J2EE Application Server.
  - MySQL 5 Database Server
  - Shared Asterisk Server for VoIP-IVR projects.

# Teaching Enterprise Open-Source

- Asterisk
  - GPL Licensed Enterprise PBX / Softswitch
  - Each team gets a 949 area code DID number.
  - All application code must interact with Asterisk via the Manager and Fast-AGI interfaces.
  - The Asterisk-Java library is a good development starting point.

# Teaching Enterprise Open-Source

- Asterisk (cont'd)
  - The server is configured with Festival TTS software. The voice sounds like Zardoz.
  - Students like it because speech is FUN!
  - Adding a Telephony UI to a web application encourages design modularity.

# Teaching Enterprise Open-Source

- Asterisk-Java Interface

- Uses the Asterisk-Java class libraries to communicate with an Asterisk server via the FastAGI protocol. Implements both the Manager and Event-driven interfaces.
- We provide the students with a Geronimo GBean service that implements the interface via JMS messaging and a simple XML command file format.

# Teaching Enterprise Open-Source

- Asterisk-Java interface (cont'd)
  - STCL – Simple Telephony Control Language
    - `<survey surveyID="" />`
    - `<waitForRing msgID="" preDelay="" postDelay="" />`
    - `<sayText theText="" />`
    - `<playGreeting greetMsg="" />`
    - `<validateUser roleID="" />`
    - `<askQuestion quesText="" valDigits="" />`
    - `<hangup />`
    - `<saveResponses />`

# Teaching Enterprise Open-Source

- STCL Demo
  - The Trivial message retrieval system.
    - Answers incoming SIP DID trunk
    - Prompts for password
    - Plays pre-recorded message
    - Hangs up
    - Writes call date/time/userid to log

# Teaching Enterprise Open-Source

## • STCL Sample Script

• `<survey>`

• `<waitForRing/>`

• `<validateUser testID="1234" />`

• `<playGreeting msgID="you-have-reached-a-test-number" preDelay="0" postDelay="0"/>`

• `<playGreeting msgID="goodbye" preDelay="0" postDelay="0"/>y`

• `<hangup />`

• `</survey>`

# Teaching Enterprise Open-Source

- Questions and Answers