Concurrent Version Control
CSULA CS 491B
Alfredo Tijolo III

CVS Overview

CVS Goals

- Two types of development styles
  - "Cathedral-style"
  - "A Great Babbling Bazaar"

- Ground work for CVS
  - Diff and Patch
  - Revision Control System (RCS)

CVS Description

CVS is not:
- The Lock-Modify-UnLock Model
- "File-Locking"

CVS is:
- The Copy-Modify-Merge Model

Reviewing terms:
- check out, commit, conflict, Log message, repository, revision, update, working copy

CVS keeps track of conflict

- When 2 or more developers work in the same area
- CVS alerts the conflict and it is up to the developers to communicate

Copy-Move-Merge Model

- CVS enables developers to edit simultaneously
- CVS assumes the burden of integrating changes
- CVS keeps track of conflict
- ALWAYS HAVE MASTER COPY AT RUNNABLE STATE

Other Revision Control Systems

- BitKeeper
  - scalable configuration management system
  - Distributed = every developer gets their own repository

- BitKeeper License
  - restrictions of pseudo-open-source license
Other Revision Control Systems

- Microsoft Visual SourceSafe: manages projects regardless of file type by saving them into a database.
- VSS integrates MS Access, VB, VC++, Visual FoxPro.
- Distinction between text and binary.

Procedures

Unix environment

- Other environments
  - There are Windows GUI version (www.wincvs.org)
  - Java CVS (www.jcvs.org)

Starting a new Project

- General syntax
  - Unix: `cvsimport -m "log message" projname vendor tag release tag`

Checking out working copy

- Example
  - Unix: `cvs checkout html`

Making changes

- Modify source

Finding out changes

- `cvs update` and `cvs diff`

Committing to changes

- Example
  - Unix: `cvs commit -m "my committed source"`

Status of source

- Unix: `cvs status sourcefile.txt`

Resolving and Detecting Conflicts

- Finding out who did what
- Examining and Reverting changes
Resource

- HTTP  
- PDF  