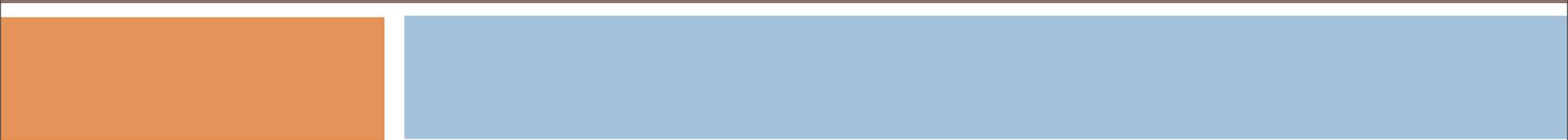


Version Control

Jim Yu



The Problem

2

- Alice, Bob and Catherine cowork in a project
 - ▣ Alice: web page
 - ▣ Bob: application flow
 - ▣ Catherine: database access
- They are struggling for the inconsistencies of the source files every day

The Problem

3

- Updates to the database schema only works on Catherine's computer, but not on Alice's or Bob's
- Bob adds a new function to the utility code, but the function is gone after syncing with Catherine
- They all have to remember what is modified since last sync to avoid code break

Version Control System

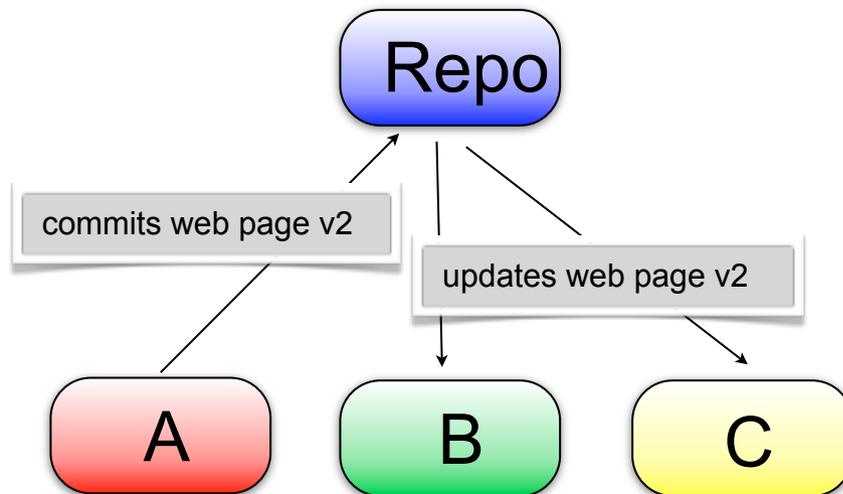
4

- A centralized system for information sharing
- A server manages the **repository**
- Any number of clients can connect to the server and read/write the files in the repository
- A write from one client creates a new **version** of the affected files

Version Control System

5

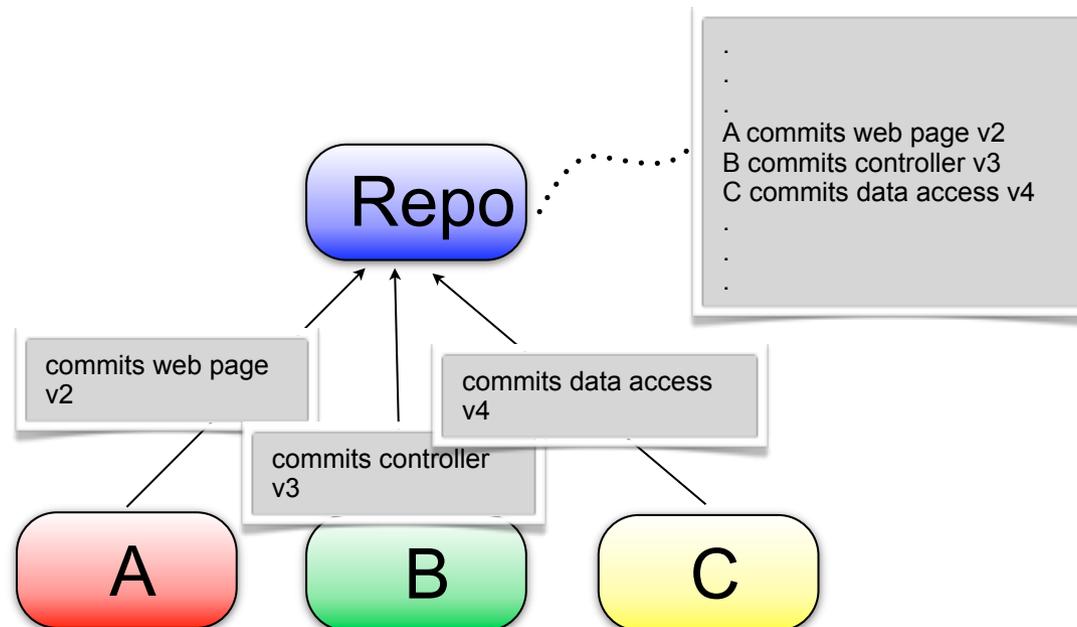
- Another client can sync the newer version of the file from the repository



Version Control System

6

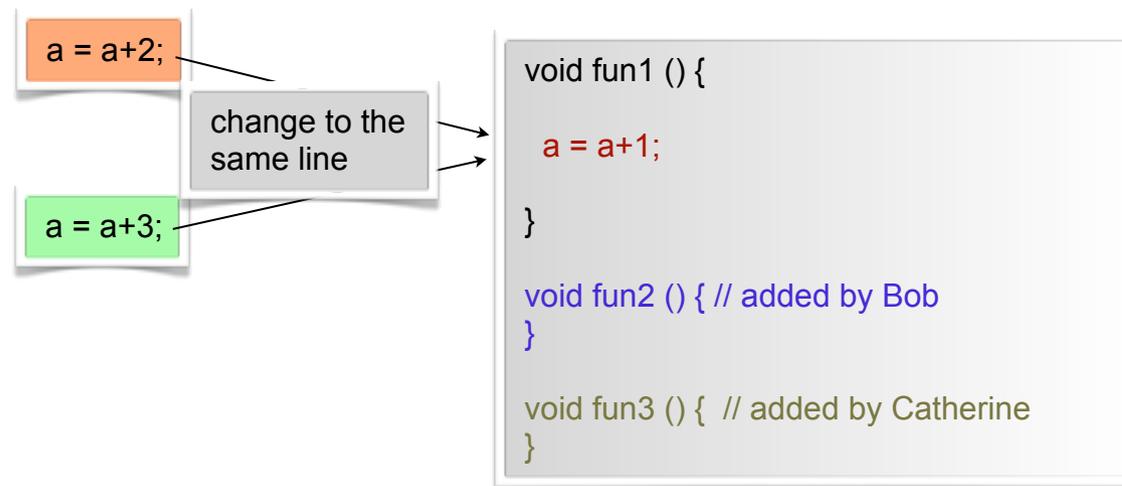
- Update history is kept in the repository
- Any previous version can be retrieved



Version Control System

7

- Non-overlapping updates to the same file can be made safely
- Overlapping updates needs manual resolution



Operations

8

- Import: initial addition to the repository to make it versioned
- Checkout: make a sync-able copy of the files from the repository
- Update: get the updates by others from the repository
- Commit: send your changes to the repository

Operations

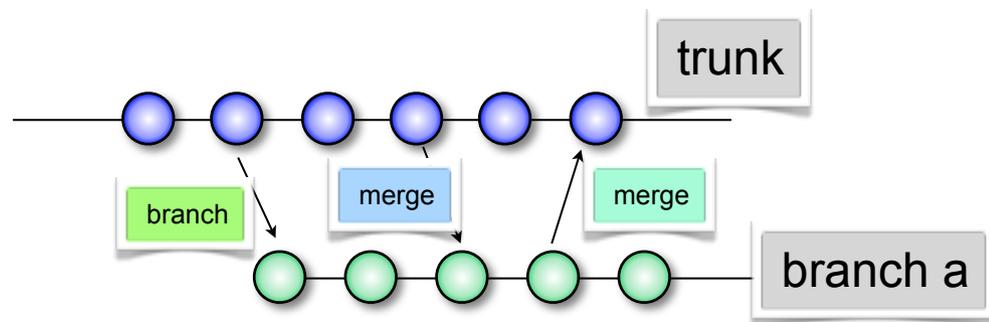
9

- Add: add new files to the repository
- Delete: delete files and propagate to the repository
 - ▣ Can deleted files be restored?
- Diff: view your local (uncommitted) changes.
 - ▣ Diff 2 versions in the history?
- Revert: discard your local changes
- Resolve conflict

Operations

10

- Resolve: decide the final result to be committed when conflicts happen
- Branch: make a copy of the whole project to be developed in parallel
- Merge: propagate the changes in one branch to another branch



Version Control in Practices

11

- Version control **all** artifacts of your project, including documents and unit tests
- Write summary comments in each commit so others can see why you make the change without looking at the code
- Use branches for feature development in parallel and merge the branches back to trunk after development completes

Version Control in Practices

12

- Integrate your version control system with bug tracking and other project management systems
 - e.g. Bug #12345 is fixed in revision #200001. You can cross reference the bug information or revision from the either system

Demo

13