

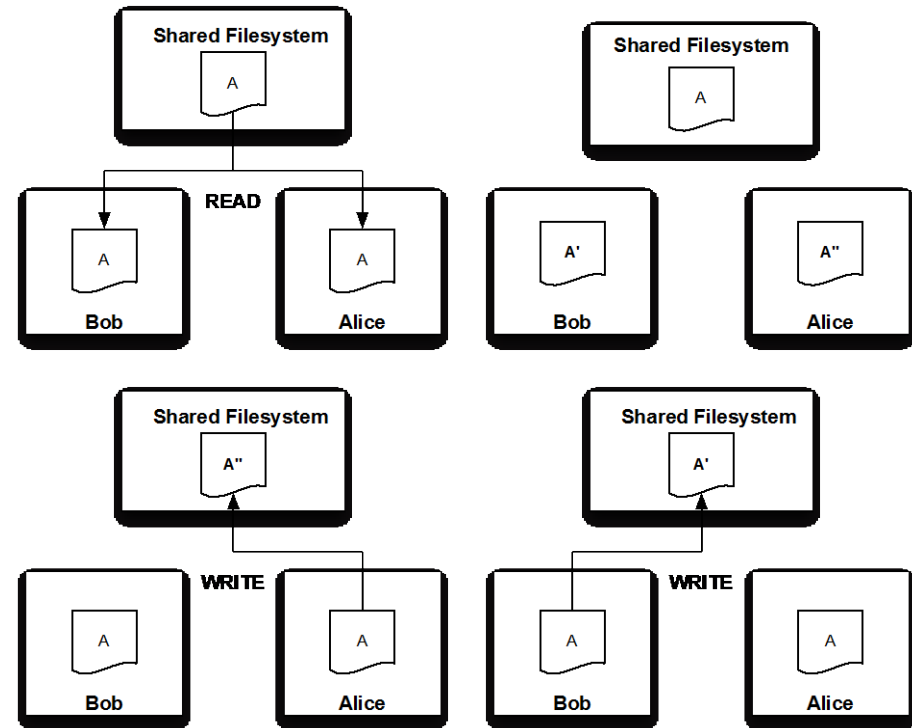
Subversion: A Brief Introduction

William R. Otte

Vanderbilt University

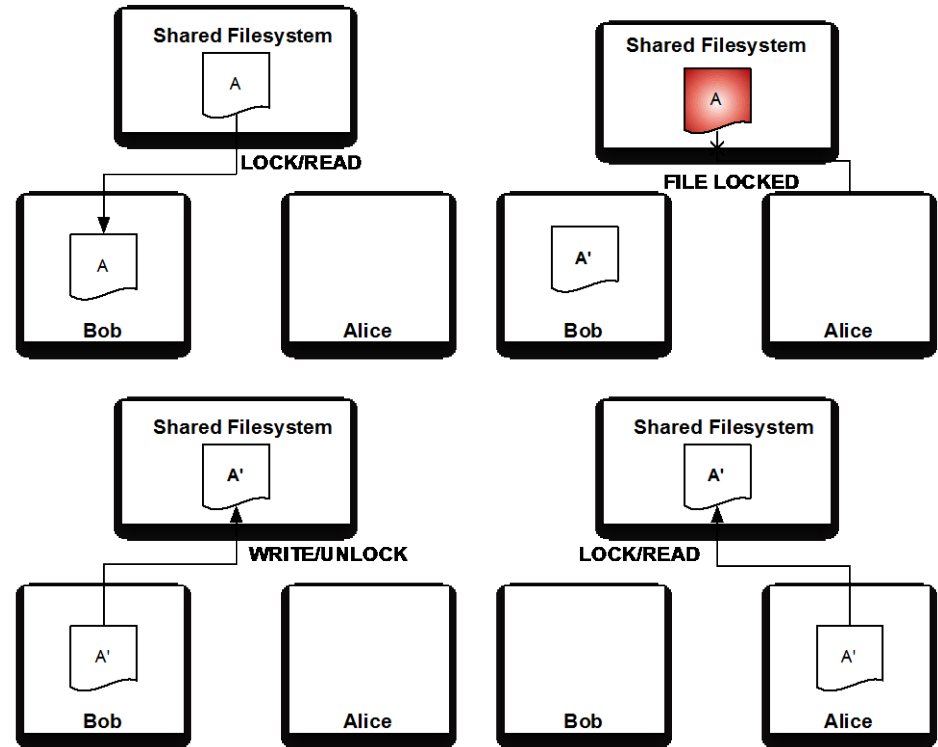
Source Control Problem

- Bob and Alice want to work on file A
- Bob and Alice both read the file
- Bob and Alice independently modify A
- Alice happens to finish first, and writes her changes
- Bob finishes second, overwriting Alice's changes



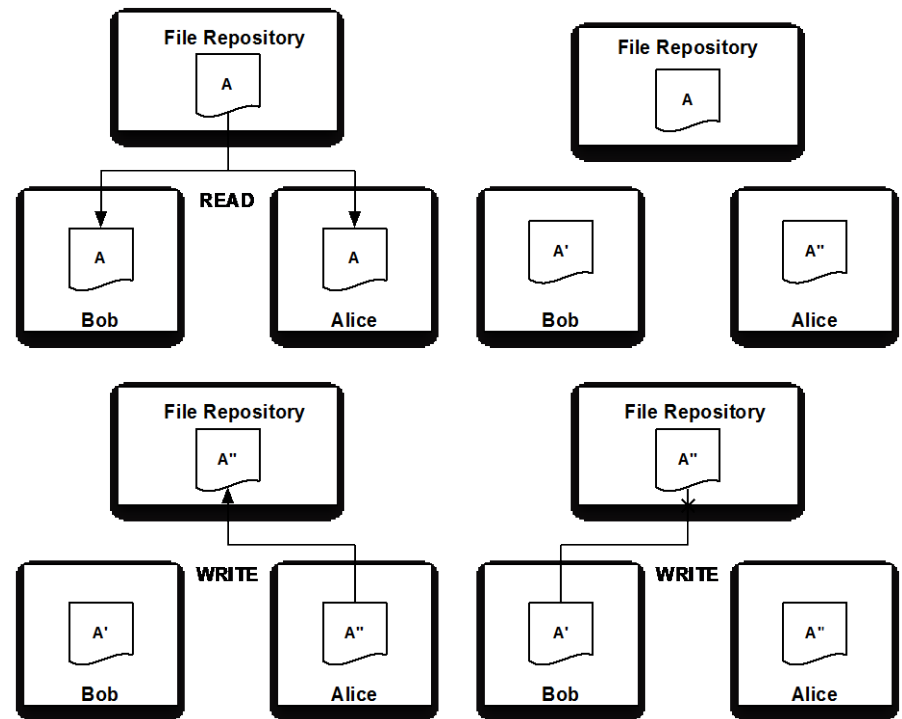
Solution Approach: Locking

- Bob “Locks” a file before reading and editing.
- Alice tries to lock the file and is denied.
- Bob finishes his edits, writes, and unlocks the file
- Alice is then able to lock the file



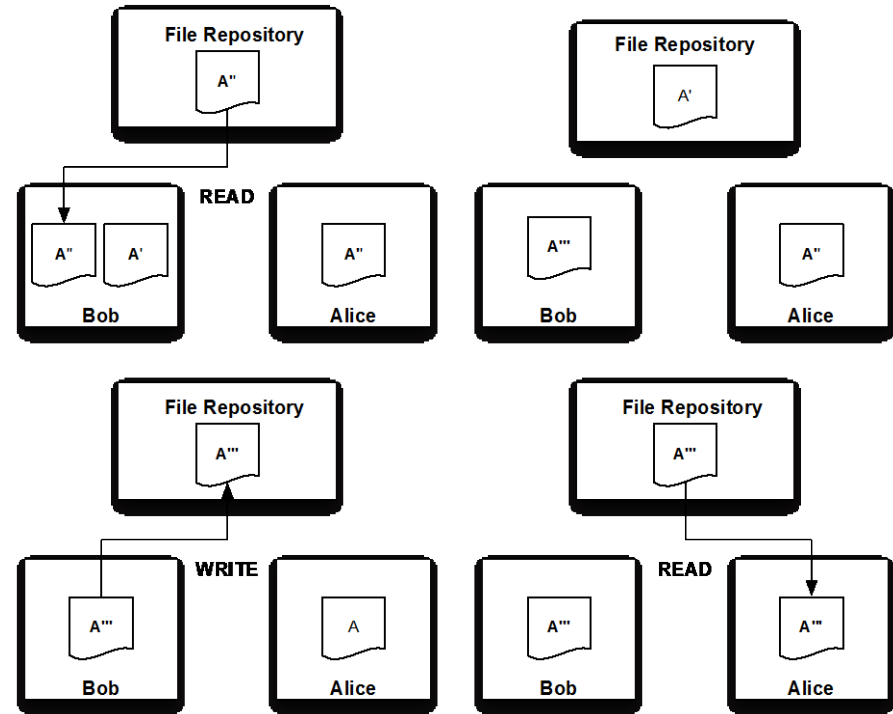
Solution Approach: Copy/Modify/Merge (1/2)

- Alice and Bob create a *working copy* of A
- Alice and Bob modify A.
- Alice (she's super fast) finishes first and commits her change
- Bob tries to commit his change, but is denied because a new version exists



Solution Approach: Copy/Modify/Merge (2/2)

- Bob must download the latest version of A
- Bob must merge his version (A') and the latest (A'')
 - If their changes are unrelated, this can be done automatically
 - Otherwise there is a *conflict* that must be resolved manually
- Bob is then allowed to write the change to the repository
- Alice may then get the latest version



Subversion: Working Copy

- A Working Copy is a copy of all the files in the repository *at a particular version*
- Your own private work space: changes are never shared or applied to/from the repository until you explicitly tell it to
- Created with the *svn checkout* command

```
svn checkout https://svn.example.com/repo/trunk
```

NOTE: There is a special directory in your working copy named `‘.svn’`. This contains important meta-data and should **never** be deleted

Subversion: Pushing/Pulling Changes

- Changes may be *pulled* from the repository using the 'svn update' command:

```
svn update
```

- Changes are *pushed* to the repository using the 'svn commit' command:

```
svn commit -m "Useful Comment"
```

Subversion: Adding Files & Directories

- Subversion only tracks files/directories if you tell it to.
- 'svn add' tells Subversion to track a file

```
svn add foo.cpp foo.h
```

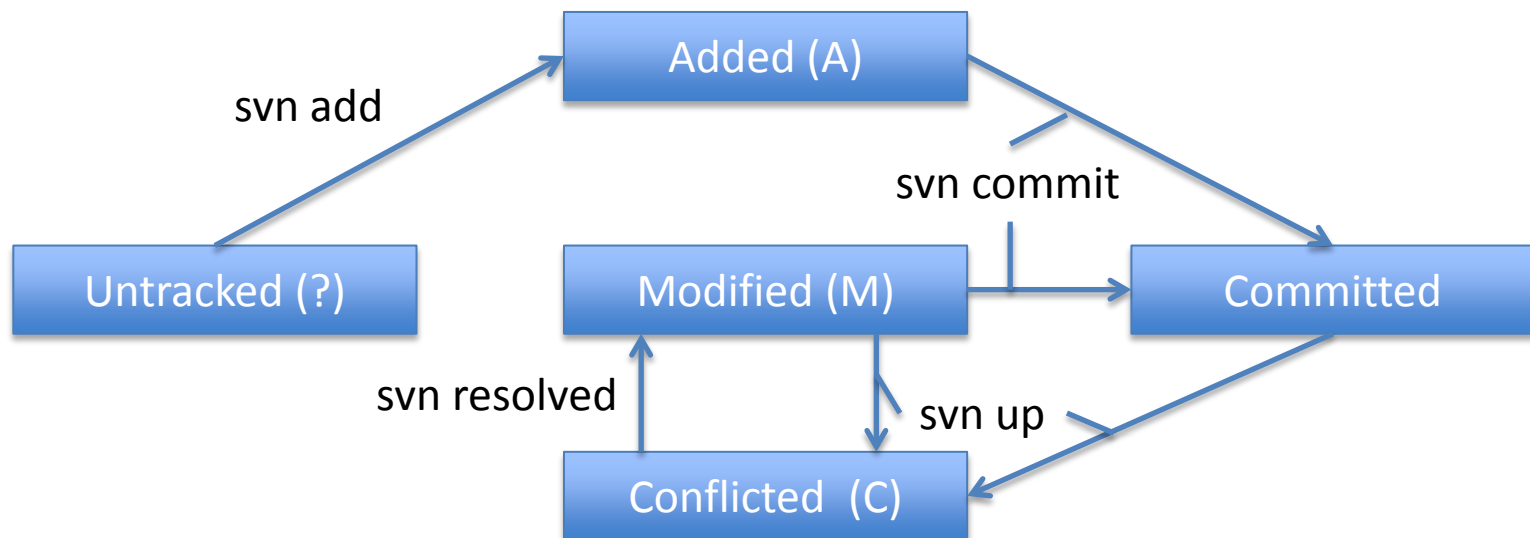
- 'svn mkdir' tells Subversion about new directories

```
svn mkdir assignment2
```


Subversion: Status of the Working Copy

- 'svn status' will tell you what the status of your working copy is

svn status



Subversion: Resolving Conflicts

- If there is a conflict, the state of the file becomes *conflicted* when you update
- The two versions of a file are *merged* into a single file, which you must *de-conflict* by hand
- Conflicted regions are marked with the following:
 - >>>>>>>>>>>>>>>>>>>
 - =====
 - <<<<<<<<<<<<<<<<<<<
- If any files are conflicted, you are not allowed to commit until they are marked as resolved:

svn resolved foo.c

Subversion: Getting Help

- Subversion is well documented and understood.
- You can use 'svn help' to get information about commands:

```
svn help
svn help commit
```

- <http://svnbook.red-bean.com> is an excellent resource.