Moves and Renames in Apache Subversion 1.8

Stefan Sperling <stsp@elego.de>

SVN Day 2012 (Jun 14)

The 'move' operation in Subversion 1.7

- 'move' is implemented as **copy** + **delete**
- In Subversion 1.7 these commands are equivalent:

```
svn move alpha beta
svn copy alpha beta; svn delete alpha
```

- Distinction exists only at the user interface.
- Committing the `delete` independently from the `copy` is possible $\Rightarrow$ non-atomic move which spans multiple revisions

The 'move' operation in Subversion 1.7

- 'move' as `copy+delete` works fine for:
  - history tracing during `svn log` and `svn diff`
  - `svn update` and `svn merge` **if there are no conflicts**
- During update/merge with incoming moves:
  - The server sends `add` and `delete` events[1] in any order.
  - The client cannot relate incoming `add` and `delete` events.
  - The client detects tree conflicts[2] via `delete` events only.

[1] This will be fixed eventually, see Hyrum's "Editor v2" talk.

[2] conflicting changes to the structure of the versioned directory tree

Tree-conflicts involving moves

1. incoming move vs. local move
2. incoming move vs. local delete
3. incoming move vs. local edit
4. local move vs. incoming move
5. local move vs. incoming delete
6. local move vs. incoming edit

In all of these cases the `copy+delete` concept causes problems.

Example: incoming move vs. local move

- Harry wants to move `src/clock.c` to `lib/clock.c`

```
$ svn move src/clock.c lib/clock.c
A         lib/clock.c
D         src/clock.c
$ svn status
A  +    lib/clock.c
D       src/clock.c
$
```

- Sally wants to move `src/clock.c` to `core/clock.c`

```
$ svn move src/clock.c core/clock.c
A         core/clock.c
D         src/clock.c
$ svn status
D       src/clock.c
A  +    core/clock.c
$
```

Example: incoming move vs. local move

- Harry commits first

```
$ svn commit -m "move clock code into lib/ directory"
Adding         lib/clock.c
Deleting       src/clock.c

Committed revision 42.
$
```

- Sally's commit is blocked:

```
$ svn commit -m "move clock code into core/ directory"
Adding         core/clock.c
Deleting       src/clock.c
svn: E155011: Commit failed (details follow):
svn: E155011: File 'src/clock.c' is out of date
$
```

Example: incoming move vs. local move

- Sally updates:

```
$ svn update
Updating '.':
A    lib/clock.c
   C src/clock.c
Updated to revision 42.
Summary of conflicts:
  Tree conflicts: 1
$ svn status
!     C src/clock.c
      >   local delete, incoming delete upon update
A  +    core/clock.c
Summary of conflicts:
  Tree conflicts: 1
$
```

- `core/clock.c` is added to Sally's working copy.
- Conflict described as "incoming **delete** vs. local **delete**"!

Example: incoming move vs. local move

- A "delete vs. delete" conflict is flagged if:
  1. Harry and Sally want to move `clock.c` to different locations
  2. Harry and Sally want to delete `clock.c`
- Case 1 is a conflict; moves are wrongly labelled "delete"
- Case 2 is **not** a conflict but detected as false positive!

  $\Rightarrow$ Modelling a move as `copy+delete` is insufficient.

The 'move' operation in Subversion 1.8

Goals for 1.8.0:

- Improve behaviour for uncommitted moves in working copy.
- Prepare client for "Editor v2"[3] rename support.

Non-Goals for 1.8.0:

- Improve behaviour for incoming (i.e. committed) moves
- Store renames in repository filesystem

[3] See Hyrum's "Editor v2" talk.

The 'move' operation in Subversion 1.8

- Keep `copy+delete` concept but add annotations for moves
- Subversion 1.7 already has appropriate columns in `.svn/wc.db` SQLite database:
  - `moved_here` column in `NODES` table (boolean)
  - `moved_to` column in `NODES` table (relative path)
    These columns are unused in Subversion 1.7.
    Subversion 1.8 starts using them.

The NODES table

Rows describe nodes (files and directories) in the working copy.

local_relpath	op_depth	presence
.	0	normal
src	0	normal
src/clock.c	0	normal

Understanding "op_depth"

`op_depth` says at which path component an operation occurred.

svn `add`, `delete`, and `copy` add rows with op_depth > 0

Example: `svn remove src` $\leftarrow$ one path component

`local_relpath`	`op_depth`	`presence`	
.	0	normal	
src	0	normal	
src/clock.c	0	normal	
src	1	base-deleted	$\leftarrow$ new row
src/clock.c	1	base-deleted	$\leftarrow$ new row

How `copy+delete` is represented

```
svn copy src/clock.c lib/clock.c
svn remove src/clock.c
```

local_relpath	op_depth	presence	
.	0	normal	
src	0	normal	
src/clock.c	0	normal	
src/clock.c	2	base-deleted	← new row
lib/clock.c	2	normal	← new row

New annotations for moves in 1.8

`svn move` is now different from `copy+delete`!

```
svn move src/clock.c lib/src/clock.c
```

local_relpath	op_depth	presence	moved_to	moved_here
.	0	normal		
src	0	normal		
src/clock.c	0	normal		
src/clock.c	2	base-deleted	lib/clock.c	
lib/clock.c	2	normal		true

The 'move' operation in Subversion 1.8

Moves are shown in status output:

```
$ svn move src/clock.c lib/clock.c
A         lib/clock.c
D         src/clock.c
$ svn status
A  +    lib/clock.c
        > moved from src/clock.c
D       src/clock.c
        > moved to lib/clock.c
$
```

The 'move' operation in Subversion 1.8

Committing just one half of a move fails:

```
$ svn commit src/clock.c -m "partial commit of a move"
svn: E200009: Commit failed (details follow):
svn: E200009: Cannot commit 'src/clock.c' because it was
moved to 'lib/clock.c' which is not part of the commit; both
sides of the move must be committed together
$
```

The 'move' operation in Subversion 1.8

Reverting one half of a move leaves other half as `copy` or `delete`

```
$ svn status
A  +    lib/clock.c
        > moved from src/clock.c
D       src/clock.c
        > moved to lib/clock.c
$ svn revert lib/clock.c
Reverted 'lib/clock.c'
$ svn status
D       src/clock.c
$
```

Changes in tree-conflict handling

Tree conflicts involving local moves are labelled correctly:

```
$ svn status
A  +    core/clock.c
!      C src/clock.c
       >   local moved away and edited, incoming delete
Summary of conflicts:
  Tree conflicts: 1
$
```

Changes in tree-conflict handling

Incoming edits follow local moves during `update` and `merge`:

```
$ svn status
A  +    lib/clock.c
        > moved from src/clock.c
D       src/clock.c
        > moved to lib/clock.c
$ svn update
Updating '.':
U    lib/clock.c
Updated to revision 42.
$
```

In 1.7 this was a "local delete vs. incoming edit" conflict.

Changes in tree-conflict handling

`update` auto-resolves incoming deletions on top of local moves[4]

Sally moves `clock.c` and updates:

```
$ svn status
A  +    core/clock.c
        > moved from src/clock.c
D       src/clock.c
        > moved to core/clock.c
$ svn update
Updating '.':
A    lib/clock.c    <- Harry's move wins
D    core/clock.c   <- Sally's move is undone
Updated to revision 42.
$
```

[4] This change is controversial and may not be released in 1.8.0

Work left to do for 1.8.0 release

- Discuss controversial changes in tree conflict behaviour.
  - Some tree conflicts not flagged anymore ⇒ regression!
  - Enhance `svn resolve` instead of auto-resolving during `update` and `merge`?
- Make sure nested moves[5] work correctly.
  - Need to fix problems with mixed-revision working copies.

[5] `http://wiki.apache.org/subversion/MultiLayerMoves`

1.9.0 and beyond

These are my personal goals and may not reflect goals of the entire user and developer community.

- Provide interactive conflict resolution for tree conflicts.
  - Needs to be **comprehensive** – handle any kind of tree conflict.
  - Current tree-conflict behaviour harms user productivity.
  - Try to deliver incremental improvements in every 1.x release.
- Detect server-side moves during conflict resolution.
  - Scan revision log for server-side moves?[6]
  - Enhance repository format with information about moves?

[6] `http://svn.apache.org/repos/asf/subversion/branches/moves-scan-log/BRANCH-README`

Thank you for your attention

Questions?