Team Developpement With Mercurial

April 29, 2011
Past vs present

Past: immutable = liquid
Present: mutable = frozen
History: Past + present
Why clean history:

**easy to read**
- Review process
- `hg annotate`

**easy to process**
- `hg bisect`
- `hg annotate`
- Continuous Integration tools

---

**Changeset**
- From valid state to valid state.
- Atomic
- As small as possible
clean history

Errare humanum est won’t make it right.

- Iterative
- Collaborative
- Multiple task in Parallel

**tool:** Version control System (distributed)
Partial Solutions exist

**rebase and histedit**
- limited
- hard to share
- hard get older version

**mq**
- overlay: (unknown by core and extension)
- limited: (queue only)
- fragile: (no transaction, reject, .hg/patches consistency)
- break rules: (truncate revlog)
The full solution

- **DVCS** trace changes to *set of files*,
- Must trace changes to *changesets*.

A is updated as $A'$

A and B are merged as C

Need to snapshot the whole set of files, but there is no such thing as a consistent state for the set of heads of a tree.

Changes are distinct from each other.
**Insert**

**characters:** $abc \rightarrow aXbc$

**lines:**
$\begin{align*}
abc & \quad abc \\
ghi & \rightarrow def \\
ghi & 
\end{align*}$

**CHunks:** *adding a chunk to a changeset*

**changeset:**
$\begin{align*}
o & C \\
o & C \\
/ & \\
/ & oB \\
o & B \\
/ & \\
/ & \rightarrow oD \\
o & A \\
/ & \\
/ & oA \\
- & \\
- & \\
\end{align*}$
Delete

characters:  $abc \rightarrow ac$

lines:  $abc \quad abc$
        $def \rightarrow ghi$
        $ghi$

CHunks: remove chunk from a changeset

changeset:  $o \ C$
            $/ \quad o \ C$
            $o \ B \quad /$
            $/ \rightarrow o \ A$
            $o \ A \quad /$
            $/ \quad -$
            $-$
Modify

characters: \( abc \rightarrow adc \)

lines: \( abc \quad abc \)
\( def \rightarrow jkl \)
\( ghi \quad ghi \)

CHunks: *modify a chunk in a changeset*

changeset: \( o \ C \)
\( \_ \quad o \ C \)
\( o \ B \quad \_ \)
\( \_ \rightarrow \quad o \ A \)
\( o \ A \quad \_ \)
\( \_ \quad \_ \)
\( \_ \)
Copy

characters:  $abc \rightarrow abac$

lines:  $abc$
        $abc$
        $def \rightarrow def$
        $abc$
        $ghi$
        $ghi$

CHunks:  $N/A$

changeset:  
            $o$ cherry pick of $C$
            /  
            $o$  $C$  $o$  /  $C$
            /  /  
            /  $o$ $B$  /  $o$ $B$
            /  /  $\rightarrow$  /  /  
            $o$  /  $A$  $o$  /  $A$
            /  /  
            //  //
### Move

**characters:**  \( abc \rightarrow bac \)

**lines:**  

\[
\begin{align*}
abc & \quad \text{def} \\
def & \rightarrow abc \\
ghi & \quad ghi
\end{align*}
\]

**CHunks:**  \( N/A \)

**changeset:**  

\[
\begin{align*}
o & \quad C \\
\downarrow & \quad \downarrow \\
o & \quad B \\
\downarrow & \rightarrow \downarrow \\
o & \quad A \\
\downarrow & \quad \downarrow \\
- & \quad -
\end{align*}
\]
Join

characters: N/A

lines: 
abc
-> abcdef

def
ghi

ghi

CHunks: N/A

changeset: o C

| o C
| o B |
| -> o A+B
<table>
<thead>
<tr>
<th>o A</th>
</tr>
</thead>
</table>
Split

characters: N/A

lines: abcdef
       ghi
   ->  abc
       def
       ghi

CHunks: N/A

changeset:

        o C
        /      
        /  o B  
  o A+B   ->  /    
   /      o A 
  -      /   
       -

Team Developpment With Mercurial
## Move Boundary

<table>
<thead>
<tr>
<th>characters</th>
<th>N/A</th>
</tr>
</thead>
</table>
| lines      | abc
def
ghi       | abcd
def
ghi       |
| Chunkss    | N/A |
| changeset  | o C
/o A+B    | o B+C
/o A
/        |
|           | o A+B
/          | o A
/          |
|           |   -     |   -     |
sum up

- Insert
- Delete
- Modify
- Copy
- Move

- Join
- Split
- Move Boundary
Proposed solution

- create new changeset as usual,
- New type of relation between changeset
  - update
  - delete
  - split
  - merge
  - (copy)

We can detect
- **obsolete** changeset
- **conflicting** changeset
- **Out of sync** changeset
Core change

It’s Necessary to:

- Alter existing command to hide **obsolete** changeset,
- Add new command to recognise/solve **out-of-sync** and **conflicting** changeset,

It’s recommended to have:

- Light weight changeset,
- Garbage collection.
Extensions change

- Do not alter Frozen changeset
- Add relevant link on edit
- Define hooks to update they internal state
Simple iterative work
Feature that need a bug fix
Multiple people working on the same thing
People have works base on unwanted changeset