

A day with Subversion (SVN)

Sebastian Heimann

Institut für Geophysik, Arbeitsgruppe Seismologie

February 15, 2007

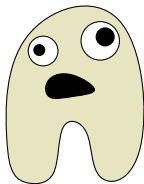
Outline

1. Concepts (by example)
2. Vocabulary
3. Benefits
4. Further reading

The initial work

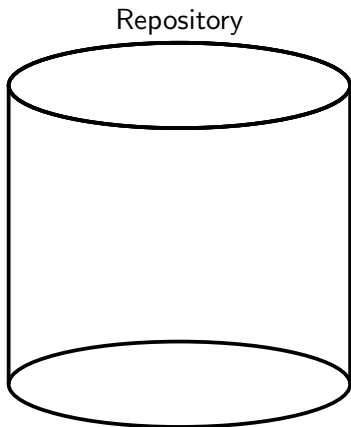
Let's assume we have just made a quick and dirty hack, which we now want to put under version control for further development:

```
> ls /my/hack
```



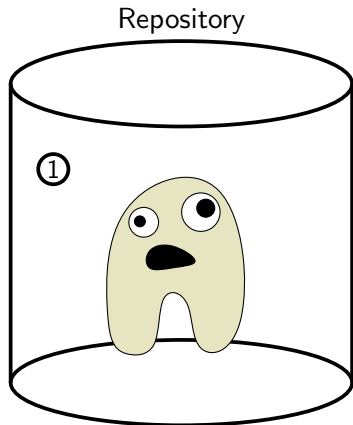
Create an empty repository

```
> svnadmin create --fs-type fsfs /my/repos
```



The initial import

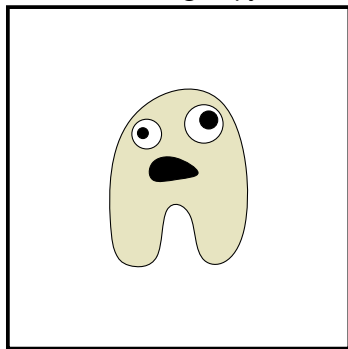
```
> svn import /my/hack file:///my/repos/hack/trunk \  
-m 'Initial import'
```



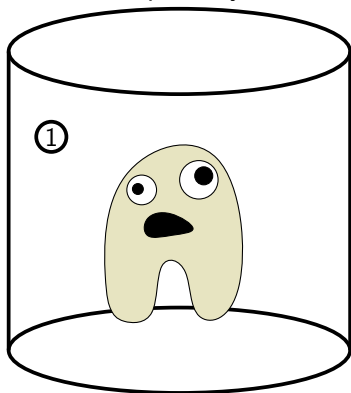
Checkout a working copy of hack

```
> svn checkout file:///my/repos/hack/trunk /work/hack
```

Working copy



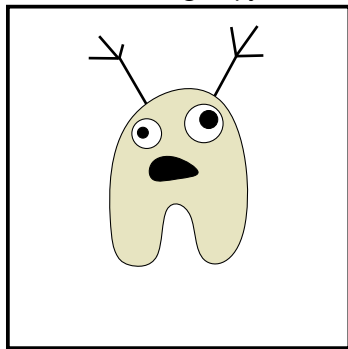
Repository



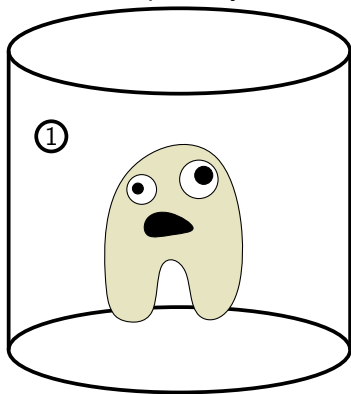
Add a new feature to hack

```
> cd /work/hack ; vi hack
```

Working copy



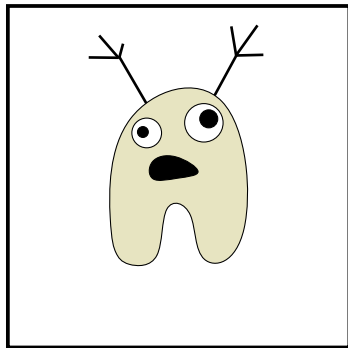
Repository



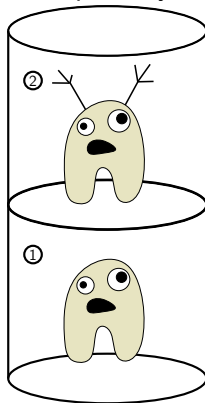
Commit your changes to the repository

```
> svn commit
```

Working copy

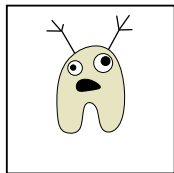


Repository

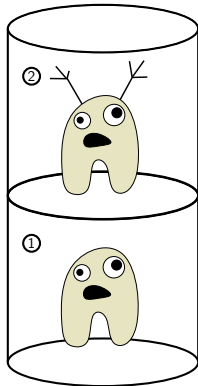


More than one person can now work on hack

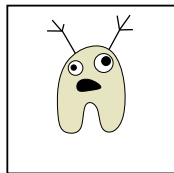
Fu's working copy



Repository

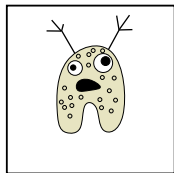


Fara's working copy

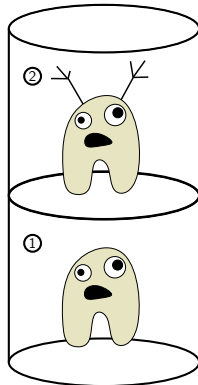


Both implement their improvements

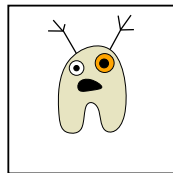
Fu's working copy



Repository



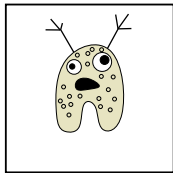
Fara's working copy



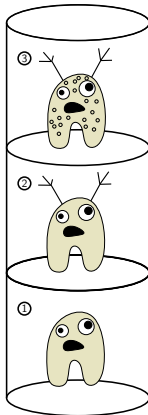
Fu commits his changes to the repository

> `svn commit`

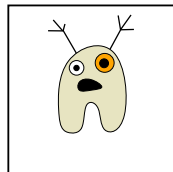
Fu's working copy



Repository

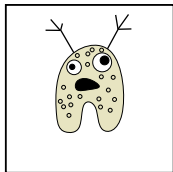


Fara's working copy

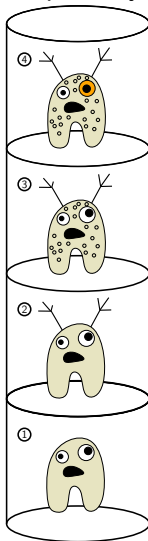


Fara commits her changes to the repository

Fu's working copy

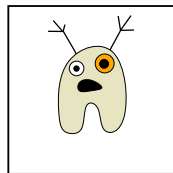


Repository



> svn commit

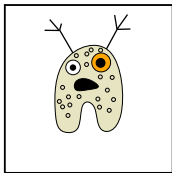
Fara's working copy



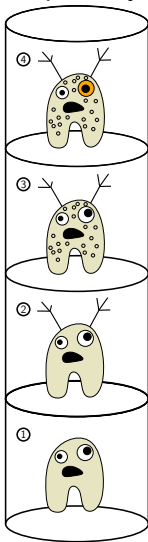
To get each others changes

> svn update

Fu's working copy

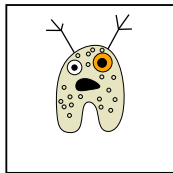


Repository



> svn update

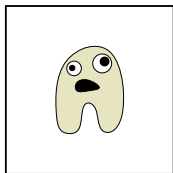
Fara's working copy



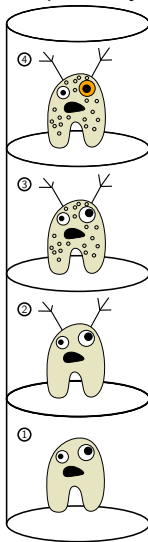
Fu and Fara can go back to any previous revision

```
> svn update -r 1
```

Fu's working copy

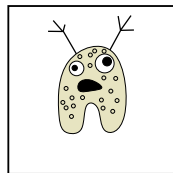


Repository



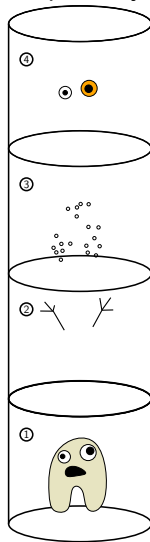
```
> svn update -r 3
```

Fara's working copy



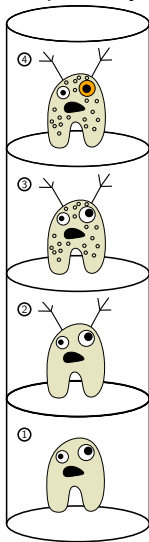
Internally the repository will only store differences...

Repository



...but it pretends to make a new copy on every commit.

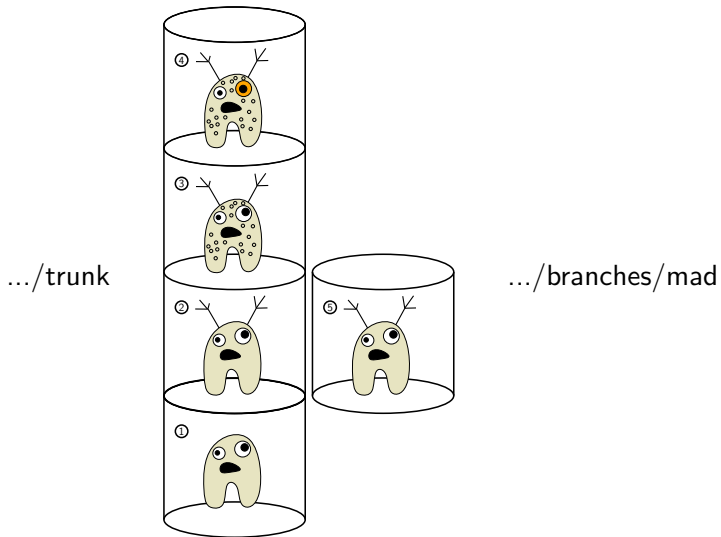
Repository



Sometimes it may be useful to branch the project...

```
> svn cp -r 2 file:///my/repos/hack/trunk \  
              file:///my/repos/hack/branches/mad
```

...this creates a copy directly in the repository.



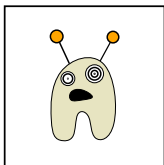
Check out a working copy of the branch

```
> svn checkout file:///my/repos/hack/branches/mad \  
/work/mad-hack
```

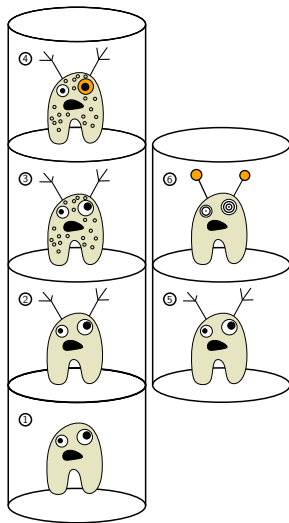
You can now safely experiment with mad-hack, while maintaining a stable version of hack in the trunk.

```
> cd /work/mad-hack
```

```
> # edit it
```



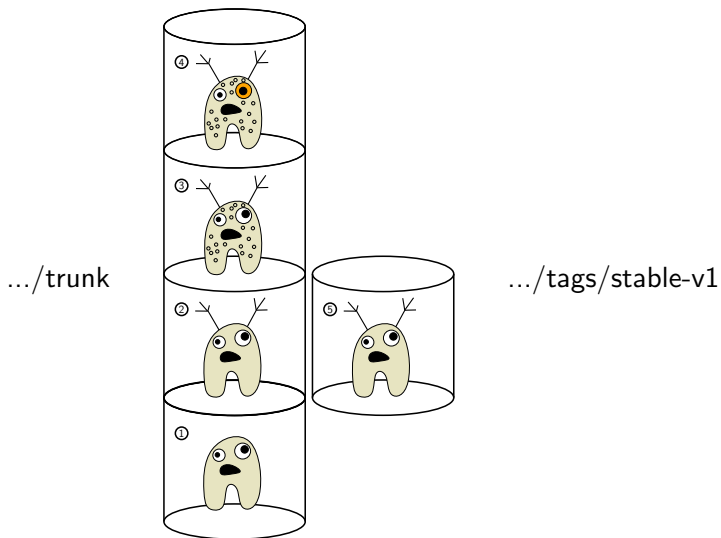
```
> svn commit
```



Tagging: You can also use the branching mechanism to mark a specific revisions as somehow special.

```
> svn cp -r 2 file:///my/repos/hack/trunk \  
file:///my/repos/hack/tags/stable-v1
```

Tagged stable version 1 of hack



Vocabulary

- ▶ repository
- ▶ revision
- ▶ working copy
- ▶ to check out
- ▶ to commit
- ▶ to update

Vocabulary

- ▶ to branch
- ▶ to merge
- ▶ to tag

Conventions

- ▶ hack/**trunk**
- ▶ hack/**branches**/mad
- ▶ hack/**tags**/stable-v1

Benefits

- ▶ Unlimited undo!
- ▶ Collaborate with others!
- ▶ Enforces structured development
- ▶ Makes you document what you are doing
- ▶ Automatic changelog!
- ▶ Helps you organize your mess in \$HOME

Tutorial

- ▶ <http://artis.imag.fr/~Xavier.Decoret/resources/svn/>

Documentation (A good intro is in Chapter 2)

- ▶ <http://svnbook.red-bean.com/nightly/en/>

Online help

- > `svn help`
- > `svn [command] help`