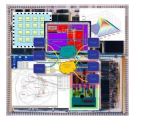
MANCHESTER 1824

The University of Manchester

Subversion code management system
(A. K. A. SVN)

Sergio Davies

APT group – School of Computer Science
The University of Manchester



SpiNNaker
a universal Spiking Neural Network architecture

MANCHESTER 1824

The University of Manchester

"*If C gives you enough rope to hang yourself, think of Subversion as a sort of rope storage facility*"

- Brian W. Fitzpatrick

SpiNNaker
a universal Spiking Neural Network architecture

Subversion features

- Revision numbers are global across the whole repository and identify how the entire repository looks in a specific instant
  (i.e. You can identify a specific build of the software in the repository just using the repository revision number)

- No additional space required for file or directory copies (i.e.: cheap copies or symbolic links)

- Directory versioning

- Atomic commits (all or nothing)

- Small database space needed (compressed diffs)

- File & directory metadata

MANCHESTER 1824

The University of Manchester

Subversion architecture

Software repository database

Currently hosted on: apollo.cs.man.ac.uk

WAN or LAN

Working copy on your computer

(folder in your hard drive)

SpiNNaker a universal Spiking Neural Network architecture

MANCHESTER 1824

The University of Manchester

Downloading repository content

USE THIS ONLY THE FIRST TIME YOU WANT TO INITIALIZE A WORKING COPY FOLDER ON YOUR PC!!!

On your computer, using a terminal window, type:

```
svn checkout file:///home/amu4/spinnaker/svn
```

This will download the entire repository tree into a working directory.
To checkout only a part of the tree use:

```
svn checkout
file:///home/amu4/spinnaker/svn/path_of_repo_needed
```

SpiNNaker a universal Spiking Neural Network architecture

MANCHESTER 1824

The University of Manchester

Basic work cycle

All the commands described are to type in a terminal window after changing directory into your working dir

- **Beginning of the day:**

```
svn update
```

- **Working on files:**

Use your favourite text editor.
Subversion is not affected by this!

- **Examine changes:**

```
svn status
```

- **End of the day:**

```
svn commit -m "describe here what you did"
```

SpiNNaker
a universal Spiking Neural Network architecture

MANCHESTER 1824

The University of Manchester

Basic work cycle – file operations

All the commands described are to type in a terminal window after changing directory into your working dir

- **Adding files to the repository:**

```
svn add file_path/file_name
```

- **Deleting files from the repository:**

```
svn delete file_path/file_name
```

- **Copying files/dirs from one folder to another:**

```
svn copy src_path/src_name dst_path
svn copy src_dir_path dst_dir_path
(always recursive)
```

- **Moving files/dirs from one folder to another:**

```
svn move src_path/src_name dst_path/dst_name
svn move src_dir_path dst_dir_path
(always recursive)
```

SpiNNaker
a universal Spiking Neural Network architecture

MANCHESTER 1824

The University of Manchester

Undoing changes

All the commands described are to type in a terminal window after changing directory into your working dir

- **Resurrecting deleted files:**

```
svn update -r repository_revision
    file_path/file_name
```

- **Undoing modifications from the latest revision**

```
svn revert file_path/file_name
```

SpiNNaker a universal Spiking Neural Network architecture

Repository architecture

```
+-Working dir
|
+-+-Project 1
| +---Branches
| +---Tag
| +---Trunk
|
+-+-Project2
| +---Branches
| +---Tag
| +---Trunk
|
...
```

- Trunk: **must** always host a compilable and "working" copy of the source code
- Tag: contains a snapshot of a project in a specific instant
- Branches: contains copies of the main project (contained in \trunk) where users are working on (development branches) and new features are not tested/stable

Branching policy (1)

The never-branch system

- Users commit their day-to-day work on /trunk.
- **Occasionally /trunk "breaks"** (doesn't compile, or fails functional tests) when a user begins to commit a series of complicated changes.
- **Pros:** Very easy policy to follow. New developers have low barrier to entry. Nobody needs to learn how to branch or merge.
- **Cons:** Chaotic development, **code could be unstable at any time**.

NO!!!!!!

From: Subversion Best Practices - http://svn.apache.org/repos/asf/subversion/trunk/doc/user/svn-best-practices.html

Branching policy (2)

The always-branch system

- Each user creates/works on a private branch for every coding task.
- When coding is complete, someone (original coder, peer, etc...) reviews all private branch changes and merges them to /trunk
- **Pros:** /trunk is guaranteed to be extremely stable at all times.
- **Cons:** **more merge conflicts than necessary. Requires users to do lots of extra merging**.

TOO COMPLEX!!!

From: Subversion Best Practices - http://svn.apache.org/repos/asf/subversion/trunk/doc/user/svn-best-practices.html

Branching policy (3)

The branch-when-needed system

- Users commit day-to-day work on /trunk
- /trunk must compile and pass regression tests at all times.

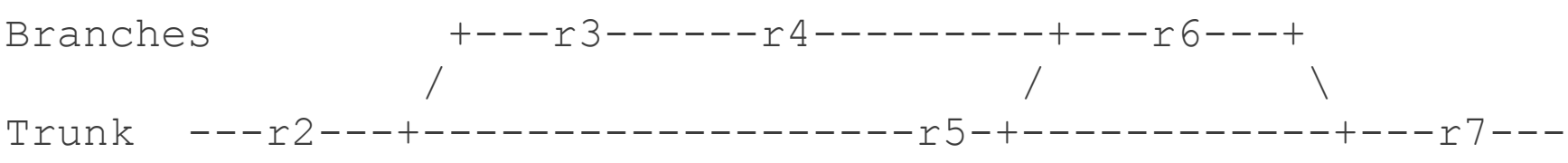
(Committers who violate this rule will be publicly humiliated.)

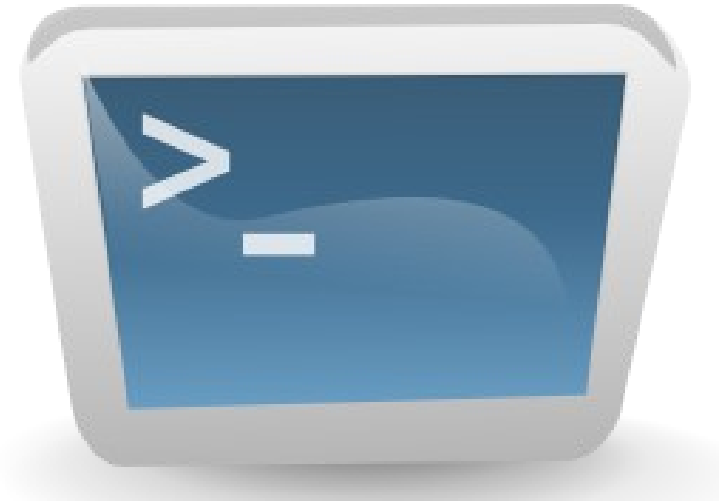
- If a commit is needed before testing (e.g. changes very wide in the software), users generates a branch
- **Pros:**/trunk is guaranteed to be stable at all times. The hassle of branching/merging is somewhat rare.
- **Cons:** Users must compile and test before every commit.

OK!!!

From: Subversion Best Practices - http://svn.apache.org/repos/asf/subversion/trunk/doc/user/svn-best-practices.html

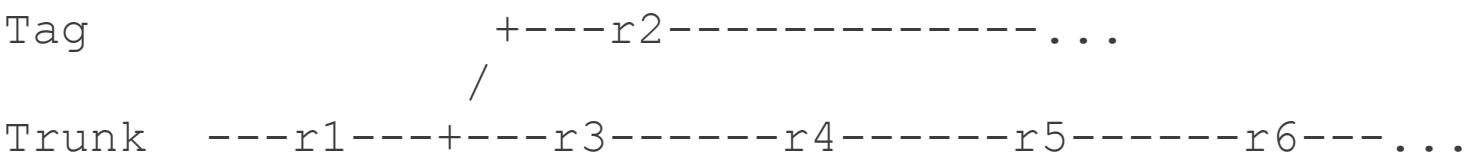
How to use branches





Live example

How to use tags



MANCHESTER 1824

The University of Manchester

Repositories location

In our group there are (at least) two SVN repositories:

- **Software repository:**
  `/home/amu4/spinnaker/svn`

- **Paper repository:**
  `/home/amu4/spinnaker/svn_papers`

SpiNNaker
a universal Spiking Neural Network architecture

MANCHESTER 1824

The University of Manchester

For strange situations

ASK ME, PLEASE!!!

SpiNNaker
a universal Spiking Neural Network architecture