

— FileWeaver —
Présentation séminaire du PC2, PEPR Ensemble

Julien Gori

May 7, 2024

FileWeaver: Flexible File Management with Automatic Dependency Tracking

Julien Gori Han L. Han Michel Beaudouin-Lafon

Université Paris-Saclay, CNRS, Inria,
Laboratoire de Recherche en Informatique
F-91400 Orsay, France
{jgori, han.han, mbl}@lri.fr



SORBONNE
UNIVERSITÉ

Starting Observations

Knowledge workers (scientists)

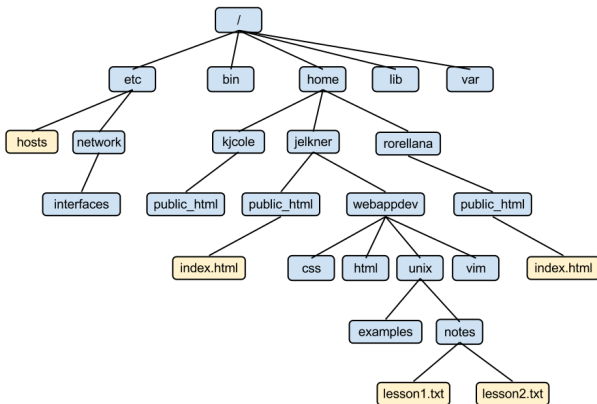
1. use different supports for information
2. transfer information across supports
3. hoard information just in case
4. create personalized strategies for re-finding information
5. use repetitive produce-communicate-reproduce cycle
6. have a limited way of expressing files dependencies



SORBONNE
UNIVERSITÉ

Files: More than a tree

Main idea of FileWeaver → Treat the filesystem as more than a tree



SORBONNE
UNIVERSITÉ

Quick view of FileWeaver windows

Short video

FileWeaver video

long video

Step back: What is FileWeaver?

1. Database that stores a graph with vertex and node attributes
2. Version control system
3. Services/features that exploit this DB/VCS

FileWeaver in a collaborative scenario

- ▶ Fileweaver helps people keep track of their files, and relationships between files → this need is even more felt when working with other people!
- ▶ viewing dependencies and histories (we believe) helps create the mental model of the project
- ▶ Example features: sharing all dependent files, adding notes/tags
- ▶ not locked in one application: everyone is free to use its own editor, tools etc.

Making a collaborative FileWeaver

- ▶ Services/features would work the same but
- ▶ DB and VCS have to be rethought

Making a collaborative FileWeaver — VCS

VCS = vanilla git

- ▶ distributed: a clone of a repo is equivalent to any other
- ▶ decentralized: no default “main” server (although we can if we want to e.g., GitHub, GitLab ...)
- ▶ asynchronous (push/pull) (semi-asynchronous e.g., when interacting with overleaf)

distributed and decentralized is positive for a collaborative FileWeaver, but the asynchronous part will lead to conflict resolution (= tedious)

automatic conflict resolution? automerge (CRDT)? (guarantee that the conflict is resolved, but not that the resolution makes sense?)



SORBONNE
UNIVERSITÉ

Making a collaborative FileWeaver — DB/graph

Currently:

- ▶ detection of parents/children built on running a file through strace (Linux only, local only)
- ▶ user can also add/remove dependencies arbitrarily (direct manipulation) and specify if edges of the graph should be updated or not.
- ▶ no clear solutions right now
- ▶ Gnome Nautilus file explorer incorporates the Fileweaver backend → virtual filesystem using FUSE



SORBONNE
UNIVERSITÉ

Making a collaborative FileWeaver — spare thoughts

- ▶ Granularity of objects: Fileweaver operates at the level of files, but could consider chunks of files instead
- ▶ what about non-text files? (everything is text in the end e.g. .docx and .pptx are zip files with XML and ressources)
- ▶ Does merging always make sense? e.g., in multilingual content, you may want to merge graphics, but not text (git cherry-picking?)