





Ultra Light

## Abstract

A key feature of collaboration is having a *log* of what and how is being done - for private use/reuse and for sharing selected parts with collaborators in today's complex, large scale scientific/software environments.

Even better if this log is *automatic*, created on the fly while a scientist or software developer is working in a habitual way, without the need for extra efforts. The CAVES (Collaborative Analysis Versioning Environment System) and CODESH (COllaborative DEvelopment SHell) projects address this problem in a novel way, building on the concepts of *virtual state* and *virtual transition* to provide an automatic persistent logbook for sessions of data analysis or software development in a collaborating group.

Repositories of sessions can be configured dynamically to record and make Virtual state: can be reproduced on demand available in a controlled way the knowledge accumulated in the course of a scientific or software endeavor.

### Case1: Simple

User 1 : Does some analysis and produces a result with tag analX\_user1. User 2: Browses all current tags in the repository and fetches the session stored with tag analX\_user1.

## **Possible scenarios**

Case2: More Complex etc

User 1 : Does some analysis and produces a result with tag **analX\_user1**.

User 2: Browses all current tags in the repository and fetches the session stored with tag analX\_user1.

User 2: Does a modification in the program obtained from the session of user1 and stores the same along with a new result with tag analX\_user2\_mod\_code.

User 1: Browses the repository, finds that his program was modified and decides to extract that session using the tag

### analX\_user2\_mod\_code.

This scenario can be extended to include an arbitrary number of steps and users in a working group or groups in a collaboration.



## The Metaphor

- A cave is a secure place to store stuff
- Usually you need a key to enter

• Stuff can be retrieved when needed (and if the temperature is kept constant, usually in good shape)

 Small caves can be private, larger are usually owned cooperatively

• When a cave is full, a new one is build

• To get something, one starts at the local cave and, if needed, widens the search ...



# Virtual States and Transitions, Virtual Sessions and Collaboration Dimitri Bourilkov

University of Florida, Gainesville, FL 32611, USA



## • Concentrate on the interactions between scientists collaborating over extended periods of time Automatic and complete logging and reuse of work or analysis sessions (between checkpoints) • Extend the power of users working or performing analyses or developing software in their habitual way,

- giving them virtual data capabilities • Build functioning collaboration suites (stay close to users!)
- Seamlessly log, exchange and reproduce results and the corresponding methods, algorithms and programs
- First prototypes use popular tools: C++, Python, ROOT, MySQL and CVS; e.g. all ROOT/shell
- V.Khandelwal; CAVES: DB v1, D.Bourilkov & M.Kulkarni v0

## Glossary

## |State> = |Logged part, Environment> this (somewhat) arbitrary) split is called a collaboration contract

**Virtual transition:** all actions to go from initial |I> to final |F> state - |F> = T |I> recorded (logged) automatically

**Virtual session:** delimited by virtual states (checkpoints)

## **CAVES / CODESH Architectures -**Scalable and Distributed



- Only CVS user accounts with password authentication, no UNIX accounts on the server (gridmapfile uses same idea)
- read/write access control lists (per user & directory)

## **Extensible Command Set**

- Session commands
- open < session >
- close <session>
- During analysis
- help <command>
- browse <tag>
- inspect <tag> <b|c>
- startlog
- log <tag> <annot>
- annotate <tag>
- extract <tag>

- Administrative tasks
  - copy <tag> <from> <to>
  - move <tag> <from> <to>
  - delete <tag> <from>
  - archive <tag> <to>
  - retrieve <tag> <from>

## **CODESH** commands:

- run, shell getenv
- getalias etc



## **CAVES / CODESH Projects**

commands and CAVES/CODESH commands available; lead developers: CODESH: D.BOUrilkov &

## **Example - CVS Server Backend**

• Sandbox programming – work on per session basis

- CVS provides version control by tagging releases
- CVS tags act as unique IDs for virtual sessions (the namespace can be structured by a collaborating group e.g. one big cave or many barrels in a cave, selected on a session basis)
- Both local and remote modes of working
- CVS pservers (secure, efficient remote stores):





- We are looking forward to **user feedback**
- Future directions:

- GSI security
- Automatically convert session log to workflow
- Tune on smaller samples, schedule on grid for larger tasks

A picture is better than 1000 words: Try out the releases ! More info at http://cern.ch/bourilkov/caves.html

## ICCS 2005, LNCS 3516, pp. 342-345, 2005.







virtual data language [1].



• Work in progress – **first** CAVES and CODESH releases out

• Different back-ends: web/grid service oriented (e.g. Clarens), sql servers • Extend remote data access: e.g. UltraLight services, Clarens, xrootd .

```
CAVES white paper arXiv: physics/0401007; http://arxiv.org/abs/physics/0401007
CODESH / CAVES paper arXiv: physics/0410226; http://arxiv.org/abs/physics/0410226
```



- http://xxx.lanl.gov/abs/physics/0401007
   Brun, R. and Rademakers, F.: ROOT An Object Oriented Data Analysis Framework. Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81–86 5. Grid-enabled Analysis Environment project: http://ultralight.caltech.edu/gaeweb