



# A Visual Adaptive Interface to File Systems

Rosario De Chiara

Ugo Erra

Vittorio Scarano

ISISlab — Dipartimento di Informatica ed Applicazioni “R.M. Capocelli”

Università degli Studi di Salerno

84081 Baronissi (Salerno) — Italy

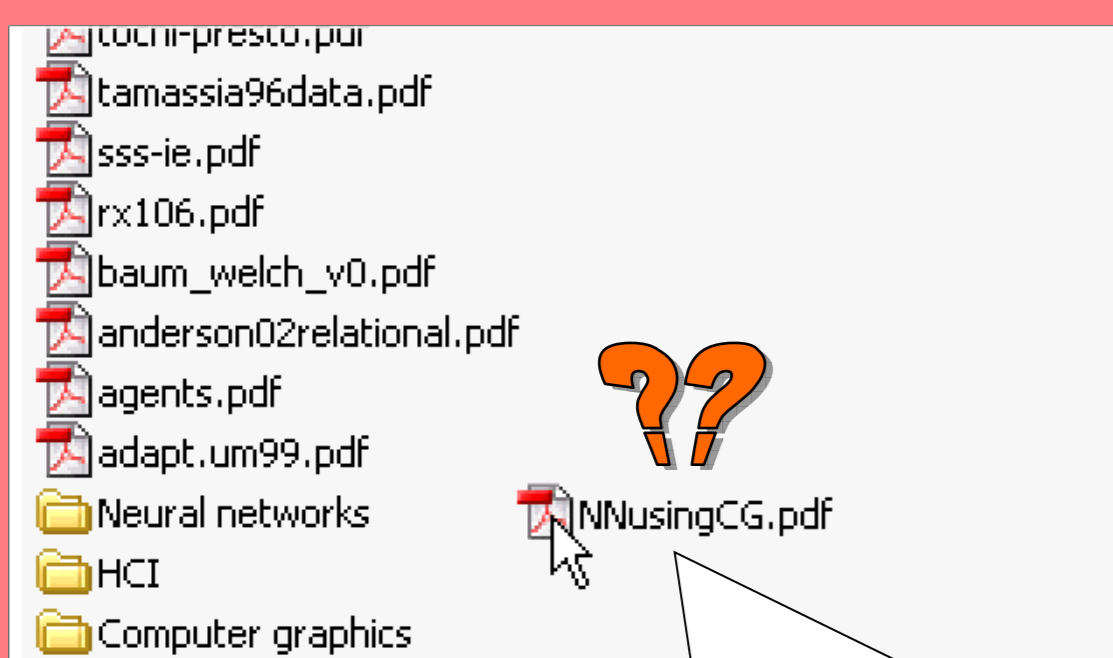
ISISlab



Euler Venn Diagrams

Markov Chains

## MOTIVATIONS



Files often should be put in different categories at once our paper should be in /Research/Practice/Visualization/AVI2004/ but also in /Topics/Theory/VennDiagram/ as well as in /Programming/C++/Graphics/ and don't forget to place it under /Programming/C++/OpenGL/

Symbolic links are not a valid solution:  
• not offered in a systematic framework  
• only a solution for a single document  
• difficult to trace back:  
Users have to manage the link consistency after operations of rename/move and delete of files and directories.

**Abstract.** We present our experience in building a visual file manager, VennFS2, that offers to users an adaptive interface toward access to files. Our file manager was originally designed to overcome some of limitations of hierarchical file systems, since it allows users to categorize files in such a way that files may belong multiple categories at once. Based on the last history of the files that were opened and modified by the user, VennFS2 graphically presents the user a small number of choices of the next file the user will modify.

100% HFS free

OpenGL Graphics

I'm not interested of these files.

**HFS** Hierarchical File System is an established reality. Developed more than 30 years ago (Ritchie and Thompson for Unix)  
In the 70s things were different:  
• a computer had 64K memory  
• little bit more of storage  
• designed keeping in mind the efficiency more than the expressivity  
• innovations are needed making information retrieval as efficient for users as it is for machines.

## File life

Once the files interesting to user have been spotted a *stimulating scenario* rises up. Interesting files often present a very common "life":  
• A small amount of them is modified in the same sequence: paper.tex → image1.eps → paper.tex → paper.dvi → paper.tex → etc...  
• This sequence is repeated during the days, weeks, months (it depends on how far is the deadline ;)

These are interesting files

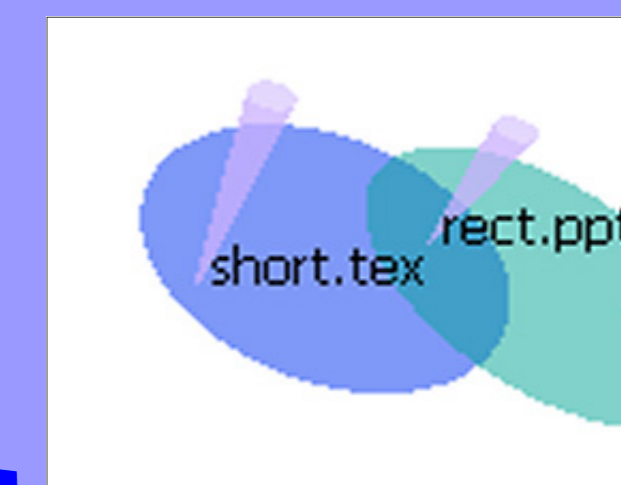
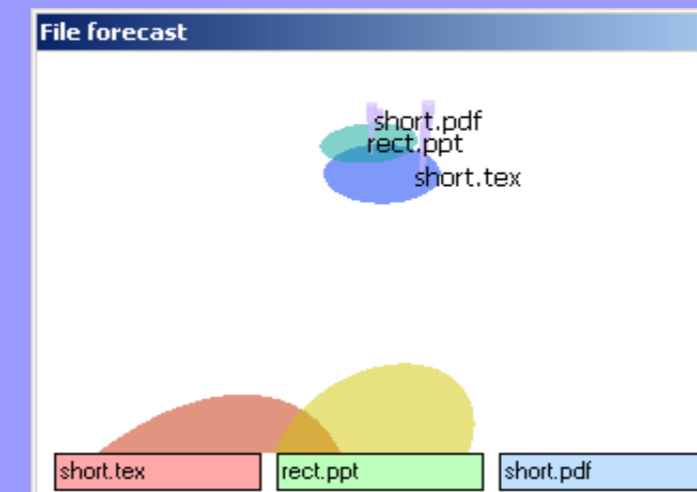
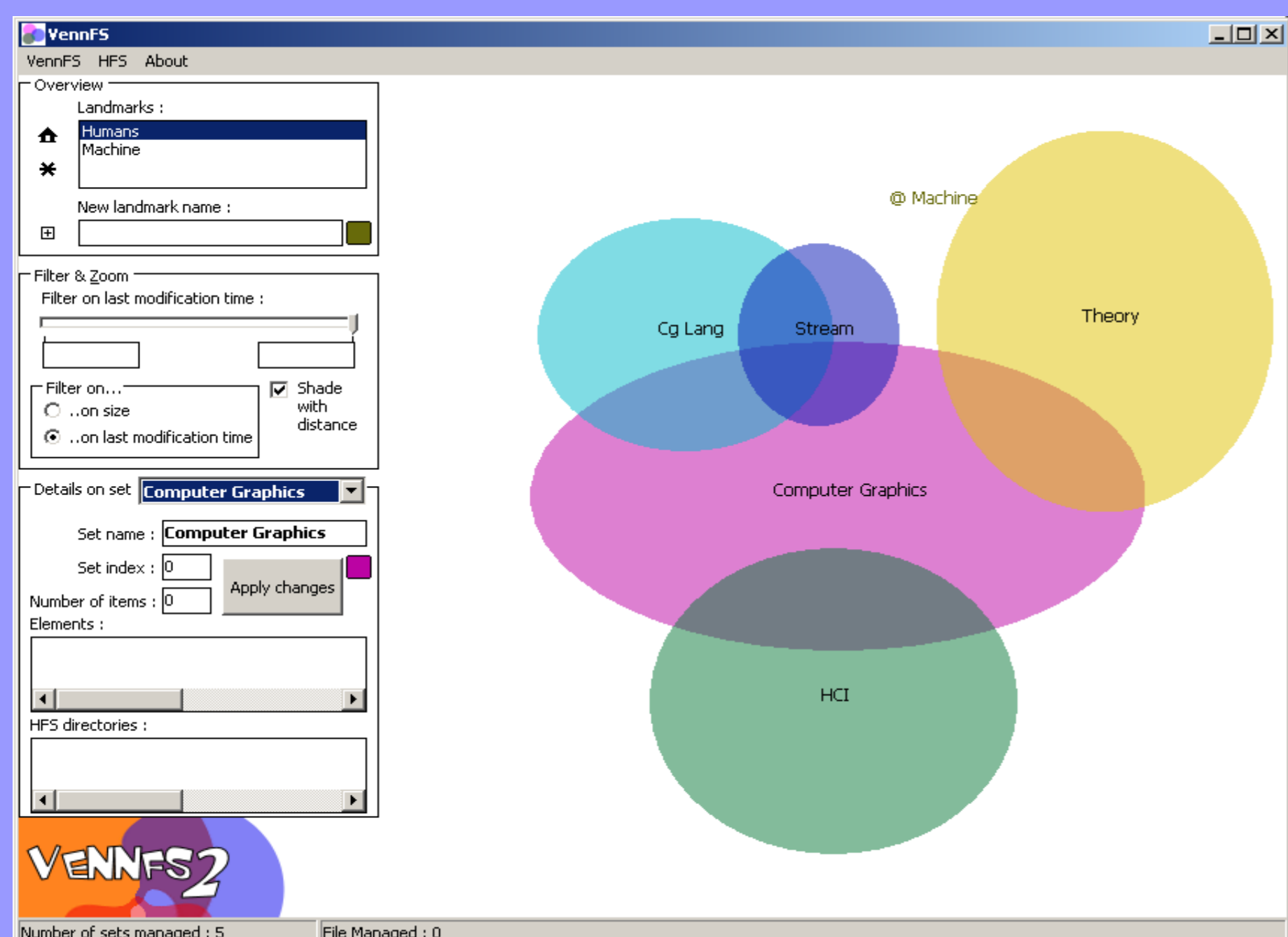
## Adaptivity

The adaptivity is provided in a *focus+context* flavour in which categories are shown in a 3d view. The adaptation is implemented using Markov's chain. The basic idea is that one or two most recently changed files can provide an idea of which files the user, will need to modify next. The "training" of the chain is provided in a transparent way while VennFS runs.

## Venn diagram

In a Venn diagram categories are ellipses on a plane. Ellipses can overlap each other creating in an intuitive way the place for files belonging to two or more categories.

## VENNFS



## Focus + Context

The *file forecast* window contains the panoramic view of the diagram. The point of view is strongly deformed using a particular OpenGL "camera". This deformation provides a wide view of the sets position and allows to comfy view of the "razor shells":  
Focus : is the probability that a file will be modified  
Context : is the diagram in which the files are placed  
The focus+context feature is provided in a non-invasive nifty visualization that allows user to know which files will be modified without looking for them around the file system.

## HFS is dead ! Long live HFS !

In VennFS the HFS has not disappeared, but it has been took to new life. The intuitive diagram can be "exported to" HFS. The diagram the user design in VennFS is exported in a newly created directory tree where:  
• Every category is a directory  
• The intersection between categories is expressed using the symbolic links  
No real files is moved in VennFS but automatically created symbolic links are used. The HFS is used like a flat database of files, the classification of files in categories is done inside VennFS.

## The future ? One click away.

In the file forecast window the files are not "just visualized". The textboxes showing forecasts are clickable using two keys: TAB to choose the files and ENTER to touch it with the associated application.

## MARKOV CHAIN

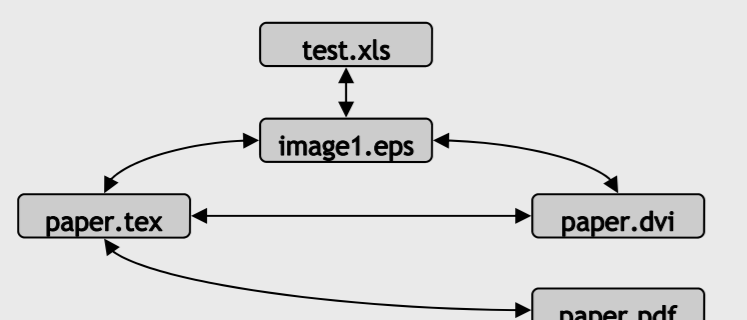
A *Markov chain* is a *stochastic process* having the so-called *Markov property*:  
• The process consists of a sequence  $X_1, X_2, X_3, \dots$  of random variables taking values in a "state space".  
• The value of  $X_n$  being "the state of the system at time  $n$ ".  
The discrete-time Markov property says that the conditional distribution of the "future"  $X_{n+1}, X_{n+2}, X_{n+3}, \dots$  given the "past"  $X_1, X_2, X_3, \dots$  depends only on  $X_n$ .  
Each particular Markov chain may be identified with its matrix of "transition probabilities" (transition matrix).  
• The entry  $p_{ij} = P(X_{n+1} = j | X_n = i)$  that is the probability of moving to state  $j$  from the state  $i$ .

Our Markov chain is labeled using file identifiers, because we are interested in predicting sequences of file modifications. We have tested two kinds of Markov chain.

- *One label state memory*: the next file that will be modified depends on the last file modified by the user. Every state in the chain is, therefore, labeled with one file identifier.
- *Two labels state memory*: the next file that will be modified depends on the last two files modified by the user.

Every state in the chain is labeled with a couple of identifiers from the last two modified files.

Another critical decision to be taken how many time we query the Markov chain in trying to guess the next file the user will modify. Experimental results shows that these parameters are critical in terms of percentage of correct previsions. Preliminary experiments seem to suggest that presenting 3 choices is enough to make the tool useful.



## ADDITIONAL INFOS

### NIFTY GRAPHICS & OPENGL

VennFS2 uses OpenGL for visualization:  
• High quality hardware accelerated images  
• Provides a useful facility "for free":  
The query "in which sets a point lies" is entirely answered using hardware without involving any geometric calculation



### FILE ACCESS SAMPLING

The file modification sampling is accomplished using a thread that monitors the files managed by VennFS. It keeps track of the last modification date. This is a portable feature whenever VennFS will be implemented under others OSes.

PROJECT WEB SITE : <http://isis.dia.unisa.it/projects/vennfs>  
AUTHORS' EMAIL : {dechiara, ugoerr, vitsca}@dia.unisa.it  
ISISLAB WEB SITE : <http://isis.dia.unisa.it>

