

Xerxes : an advanced search system for your desktop

Claudia Tanase

Razvan Alecsandrescu

SPBA
UPB
Bucharest, Romania

SPBA
UPB
Bucharest, Romania

Abstract

This paper presents an advanced mechanism used to organize and search files. The system uses a combination of semantic labeling, file parsing and Virtual Folders.

1. Introduction

One of the most important things for a computer user is to be able to organize a large number of files and search for a specific one in a natural manner.

Most of the current search applications can be used to find files using information about the file name, file type or time/date information. But this only allows for simple searches.

There is a lot to learn from web search engines like Google search engine. Not only does it allow for simple searches but also for more complex searches: including content analysis.

This paper details the architecture of an advanced system for file organization. Section 2 describes existing applications, Section 3 details the implementation of our system and Section 4 presents the results of tests performed, conclusions and future work.

1.1. Previous Work

Most of the searches that we can perform using standard OS tools are limited to directory namespaces. For example in Windows we have the Search program that can perform searches filtering the results by name, type, the time it was last modified, Unix shell wildcard patterns are useful for matching against small numbers of files, and GNU locate for finding files in the directory hierarchy. The biggest issue with these applications is that to perform queries on the **data** inside a file can be a laborious procedure because of the linear complexity involved. Fast and reliable searches require an indexing structure.

To create more advanced search systems there are 2 approaches that can be taken. The first one which is the most commonly used is to develop applications that use the file

system to obtain informations about files and then process, index and allow the users to perform queries.

The second one is to implement *file indexing inside the file system*.

There are various pros and cons to each method of implementing a search system. For example file indexing inside the file system assures that the index is synchronized with the reality of the file system. The system assures us that once a file is modified the changes will show up in the index. Also there can be a small performance boost caused by the fact that processing a file that already exists in the cache is fast.

A application separated from the file system must keep track of all changes (when files are moved, deleted, renamed, modified) which is a little more complex.

2. Summary and Conclusions

This template will get you through the minimum article, i.e., with no figures or equations. To include those, please refer to your LaTeX manual and the IEEE publications guidelines. However, for a vision conference you will probably want the following equation somewhere:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}$$

Good Luck!

Acknowledgments

bla bla bla

References

- [1] Mukund Gunti, Mark Pariente, Ting-Fang Yen, Stefan Zickler, *File Organization and Search using Metadata, Labels, and Virtual Folders*, December 15, 2005
- [2] Srinath Sridhar, Jeffrey Stylos and Noam Zeilberger, *A Searchable-by-Content File System*, May 11, 2005