

Smart File Manager – A Category Based Sorting System

Jasmohan Singh Narula¹, Vrushabh Patel², Aditya Rathod³, Dr. Vinayak D. Shinde⁴

^{1,2,3,4}Dept of Computer Engineering

^{1,2,3,4}Shree L. R. Tiwari College of Engineering.

Abstract- Today, we are using many digital devices which store all our information as data in the form of files. Our files (Photos, Music, Videos, Documents, code, etc.) are spread out across multiple devices inside numerous folders amounting to GBs or TBs. We currently have to manually specify a filename and sort them, so that we can easily identify the file. Smart File Manager System is a GUI based application that we are developing that will help users organize digital files and folders in the same way we organize real world objects into categories and sub-categories. The User preferences used by the system for the folders will be stored as an “.rfs” file in XML format, inside the root of the folder. The .rfs file will have two types of rules, i.e., Detection and Sorting rules. The folder having the .rfs file in its root will be considered a Category Folder by our system. The aim of this software is to provide a simple solution for users to arrange their data in a logical format which best suits their workflow.

Keywords- SFM (Smart File Manager), RFS (Rules for Sorting), AI (Artificial Intelligence), ICO (.ico file extension), VFS (Virtual File System), XML, GUI.

I. INTRODUCTION

Currently, our data is typically is in the range of gigabytes to terabytes. However, due to the current rate of growth of information, storing and manually organising such large amounts of data that include personal emails, messages, documents, contacts, presentation slides, audios, videos, etc. using the 45-year old folder/file metaphor has become difficult. [1]

If these files are not arranged properly or using a logic, it becomes a tedious task in locating the required file as we have to remember where those files are or use a search tool. Most of the time we have to manually specify a filename so that we can easily identify the file. These are all the reasons or drawbacks which we are trying to overcome.

Traditionally, a novice computer user found it difficult to organize his/her data. The user has to manually find his/her data, create appropriate folders and then move the data. In case of images, videos and audio files, sorting

becomes more difficult because the user has to manually open up each media file to find out what they are if they are not named properly at the time of creation.

Also the data may be spread out on multiple logical partitions/physical drives and hence, finding it and ensuring that no duplicates (no redundancy) are present is again difficult. Due to all this, when the user needs some data, he/she cannot find it easily.

II. LITERATURE SURVEY

File & Folder properties across various Operating Systems like Windows, Linux, MacOS, etc. varies a lot with the file system used. Some metadata and properties are not available across all operating systems; but the common hierarchy of Files into Folder remains the same. Files are identified by the name and extension given to them. Folders are identified only by the name given to them.

- What is common across different Operating Systems?

File Folder Analogy is used by all GUI based operating systems. Files contain the actual data for the user or program. Folder is a placeholder for files i.e. Folders are used to hold group of files. Folders can be given icons to make it easier for the user to identify them. Each Operating System gives user a home folder with basic folders for each user to make organizing the files easier.

- What is uncommon across different Operating Systems?

Each operating system supports different and wide variety of File Systems like FAT32, HFS, NTFS, EXT4, etc.; some file systems may be supported by all operating systems but some are proprietary.

- Path Naming scheme

The storage media/device of the computing devices can have multiple partitions. Each partition in Windows gets assigned a Drive letter. All the files and folders in the storage device are referenced in relation to the Drive letter. This

naming scheme is known as absolute path. E.g.:-
D:\Folder\file.ext

To differentiate folders from files, backslash(\) is used by for Windows. If a path is shared without a Drive letter, than it is relation to some folder which above that folder path in the hierarchy of file system. This path is known as relative path. [2]

In Linux and MacOS, a virtual file system is used. Every folder, file and storage device will be addressed in relation to root(forwardslash /). All entities will come after root. If Root (/) is coming at the end of path, it signifies that the path is for a folder. In same manner as Windows, files and folders can be addressed with either absolute path(with relation to root) or relative path(with relation to current folder).

- *Path Limits*

The absolute path of file or folder in Windows must not exceed 260 characters. This is the path limit for absolute path in Windows.

In the Windows API, the maximum length for a path is MAX_PATH, which is defined as 260 characters. A local path is structured in the following order: drive letter, colon, backslash, name components separated by backslashes, and a terminating null character. For example, the maximum path on drive D is "D:\some 256-character path string<NUL>" where "<NUL>" represents the invisible terminating null character for the current system code-page. (The characters <> are used here for visual clarity and cannot be part of a valid path string.) [2]

In similar manner, there is 255 character limit for file name and 4096 for path in EXT4 file system commonly used in Linux. [3]

- *Metadata*

Metadata is data that describes other data. Meta is a prefix that in most information technology usages means "an underlying definition or description." Basic information about data can be summarized with Metadata, which in turn can make it easier for finding and working with particular instances of data. For example, author, date created and date modified are examples of very basic document metadata. In addition to document files, metadata is also used for images, videos, spread-sheets, web pages, etc. [4]

Through our testing we have found out that files have more types of metadata and folders have no or are limited in comparison. Files have name, extension, author, date, ratings, etc. Folders have just name, date, and few others as metadata. All metadata of files and folders is not cross compatible between various Operating Systems. Metadata like ratings, tags, etc. are OS specific and can be lost or be unreliable after transfer.

- *How icons are set in Windows?*

If an icon file '.ico' is selected for a folder, the absolute path for the icon file is stored inside the desktop.ini file and only the .ini file is stored in the root of the selected folder. If the icon file is moved, the folder will revert back to the default folder icon.

- *Characters acceptable for file and/or folder names in Windows?*

Through our testing and reading documentation, the following characters can be used for defining names for file and/or folders for English Language users:

- 26 alphabetic characters, A to Z
- Arabic numerals, 0 to 9
- Special symbols ~ ` ! @ # \$ % ^ & () _ + - = { } [] ; ' , .

Characters which cannot be used for defining names for files and/or folders:

- | \ / ? : " < > ?

III. EXISTING SOLUTION AND SYSTEMS

- *File Browser or Manager*

A file manager or file browser is a computer program that provides a user interface to manage files and folders. The most common operations performed on files or groups of files include creating, opening (e.g. viewing, playing, editing or printing), renaming, moving or copying, deleting and searching for files, as well as modifying file attributes, properties and file permissions. Folders and files may be displayed in a hierarchical tree based on their directory structure. Some file managers contain features inspired by web browsers, including forward and back navigational buttons. [5]

Each Operating System has a variety of file browsers/managers along with the default ones available for the user to manually search and manage their files.

Default File Managers are:

- Windows – Windows Explorer
 - Linux (Ubuntu) – Files
 - MacOS – Finder
- *Search Tool*

If user cannot find files and/or folders then they can rely on a search tool. A search tool is software which uses a Search algorithm like matching keywords or metadata to find what the user is looking for.

Desktop search tools search within a user's own computer files as opposed to searching the Internet. One of the main advantages of desktop search programs is that with use of proper indexes, results can be displayed quickly. [6]

The Default search tools available with Operating systems are:-

- Windows – Search
- MacOS – Spotlight [7]
- Linux - Search and grep

Some third party search tools are:-

- Search Everything [8]
- Google Desktop Search [9]

Why Search Tools are not a proper solution?

Desktop Search Tools rely on the names & metadata of files and folders to be effective. If it is unavailable then it may require time for the tool to find the wanted files or it may not find it at all. Search tools also give a large amount of results if the search queries are too vague.

Manually browsing through numerous directories is probably the simplest but yet most frustrating task when it comes to searching for a file. Searching is possible only if one knows what one is looking for, and is typically applicable to text content. There are times when a user is looking for a particular file that may contain too many generic words shared by other files, which leads to the second problem of returning too many hits. Going through the returned list of search results may be as frustrating. In fact, there are no simple strategies to rank a collection of text documents without ready-available link/relationship information between documents; so strategies like Google's Page Rank [10] is powerless here. Third, a search simply finds the file, but does not help user organize it properly so that next time when he needs the same file he could navigate straight to the location instead of using a search

engine all over again. Another consequence is that the information is forever tied to the search engine, i.e., no search engine, no organized information. This leads to the fourth problem; the non-portability of retrieved files, i.e., the search engine simply retrieves files but do nothing to organize them. If the corpus of documents is copied to a USB disk and moved to another system, the same problem will persist on that system unless it also has a desktop search engine installed. At the end of the day, using a search engine to find files may take equal or more time as going through the various hierarchically organized directories manually. [11]

- *File System Level – Using Relational Database*

- KFS – Knowledge File System is a Stackable VFS
 - The Knowledge File System (KFS) is a smart virtual file system that sits between the operating system and the file system. Its primary functionality is to automatically organize files in a transparent and seamless manner so as to facilitate easy retrieval. KFS can be considered as a personal assistant, which can arrange every one of your documents into multiple appropriate folders, so that when it comes time for you to retrieve a file, you can easily find it among any of the folders that are likely to contain it. The actual classification can be based on a combination of file content analysis, file usage analysis, and manually configured rules. KFS organizes files using the familiar file/folder metaphor. [11]

- Gnome Storage/VFS
 - Gnome Storage [12] is an open-source effort to revolutionize the file system interface by storing everything in a relational database. It provides a virtual file system layer (GnomeVFS) for compatibility with existing applications.

- Usenetfs for newsgroup servers
 - Usenetfs is a stackable file system for large article directories [13]. It was developed to improve the file search on newsgroup servers. Similar to files on personal computers, files on servers typically reside in only a few commonly used directories over time, which makes searching inconvenient. Usenetfs changes the file structure by creating smaller directories containing fewer files, instead of the large

existing flat file structures. This improves the processing rate of the articles on the server.

- Microsoft's WinFS
 - Microsoft's WinFS file system [14] was a commercial attempt to replace the file system with a relational database. Although well invested, it has been delayed many times since it was first announced in 2002. As of 2010, it has not been released commercially due to performance issues.
 - *Some Radical and Innovative Approaches*
- Microsoft's MyLifeBits – records everything
 - Another interesting research project carried out by Microsoft is MyLifeBits [15], which aims at recording everything in a person's life. With regard to computer use, every mouse click is recorded, every web page visited (not only the link) is stored, and every IM chat is logged. MyLifeBits tries to create a "lifetime store of everything" [16]. The project is not yet available for public or commercial use.
- Stuff I've seen by Dumais – History based browser
 - A related but less ambitious approach is the "Stuff I've Seen" system by Dumais [17], which simply remembers all entities including files, web pages, emails, contacts, etc., that a user have come into contact on his computer.
- MIT's haystack – Single interface for all applications
 - MIT's haystack [18] is a different approach for information management. Haystack proposes a new concept of information management by organizing all types of information into one universal interface. Therefore, there is no need for separate applications to manage different types of data. By unifying the access to all types of information from within a single application, data from various applications can be easily associated and cross-referenced with one another.
- NEPOMUK
 - NEPOMUK (Networked Environment for Personal, Ontology-based Management of

Unified Knowledge) is an open-source software specification that is concerned with the development of a social semantic desktop that enriches and interconnects data from different desktop applications using semantic metadata stored as RDF. [19]

IV. PROPOSED SYSTEM

- *What is RFS and how it works?*

'rfs' file is a file extension being made for our system. The folder which will have this file in its root, will be considered as a Category by our system. There are two types of rules used by our system, namely Detection rules and Sorting rules, which are stored in XML format inside this rfs file.

In detection rules, user can define for a condition like file extension or file keyword, which sorting rules is to be used like basic 'if...then...' statements used in programming.

The condition (if) is written inside the <Condition> element and which sorting rules to use (then) is written inside the <Use> element.

In sorting rules, there are two fields. INPUTS and sorting string format stored as elements with character S followed by a number. Example: S1, S2, S3, ...

The element S1, S2, S3,... will be read as String by our system. As stated in literature survey, all acceptable characters will be treated as String and used as defined for naming of files and folders. When a character that is not acceptable for file folder names is encountered, our system will use them as follows:-

- Only these non-acceptable characters for file folder names are used by our system. Any other character will be ignored and not used for renaming. The non-acceptable character for file folder names should have 2 instances.
 - ? for Inputs from User. The element of sorting rules enclosed between ? ? will be read and used to take inputs from the user by our system. The inputs collected from the user will then be converted into strings and used for naming the selected files and/or folders.
 - \ or / for folder name. Any String or Input enclosed between \ \ or / / will be used to

create a folder and placing the selected file and/or folders with our system.

- o | for using Metadata from file. Any metadata or attribute of a file (like date) enclosed between | | will be read from the file and used as String for naming the selected file.
- o * for data that will be retrieved from our Software. Example is order number in selection queue.

```
<?xml version="1.0"?>
<RULES>
  <Detection-Rules>
    <Rule1>|
      <Condition>FILEEXT = jpg</Condition>
      <Use>S1</Use>
    </Rule1>
    <Rule2>
      <Condition>FILETEXT = *party*</Condition>
      <Use>S2</Use>
    </Rule2>
  </Detection-Rules>
  <Sorting-Rules>
    <INPUTS>
      <INPUT1>
        <InputType>Folder</InputType>
        <InputName>Folder Name</InputName>
      </INPUT1>
      <INPUT2>
        <InputType>TEXT</InputType>
        <InputName>Event Name</InputName>
      </INPUT2>
      <INPUT3>
        <InputType>Date</InputType>
        <InputFormat>yyyymmdd</InputFormat>
      </INPUT3>
      <INPUT4>
        <InputType>Number</InputType>
        <InputName>Account No</InputName>
      </INPUT4>
      <INPUT5>
        <InputType>Date</InputType>
        <InputFormat>dd-mm-yyyy</InputFormat>
      </INPUT5>
    </INPUTS>
    <S1>?INPUT1\IMG_|INPUT3| - ?INPUT2? - *ORDERNO*.|ext|</S1>
    <S2>|INPUT3| - Party - *ORDERNO*.|ext|</S2>
    <S3>Acceptable characters as Strings Numbers as String 1234 - *ORDERNO*.|ext|</S3>
  </Sorting-Rules>
</RULES>
```

Fig. 1: Sample RFS rule file

Explanation of sample RFS rule file

1. Detection Rule 1:- if file extension is jpg, use Sorting Rule ‘S1’. In Sorting Rule S1, create a folder in the selected Category folder by taking input INPUT1 from user. Then take INPUT3 and INPUT2 from user and place the file inside the created folder.

- EXAMPLE:-

Original File:-

D:\Unsorted\IMG_20170414123456.jpg

After sorting with S1:-

D:\Photos\New Years Party\IMG_Party - with school friends - 1.jpg

2. Detection Rule 2:- If file has keyword party anywhere, then use Sorting Rule ‘S2’. In Sorting Rule S2, read INPUT3 (date) as yyyymmdd format from file and put the file in selected Category Folder by renaming it.

- EXAMPLE:-

Original File:-

D:\Unsorted\IMG_20170414123456.jpg

After sorting with S1:-

D:\Photos\20170414 - Party - 1.jpg

- o System Working Concept

We have made a GUI (Graphical User Interface) base JAVA application which the user can install and use on their computing device.

- 1) We are going to make a custom file known as ‘.rfs’ which stands for ‘Rules for File/Folder Sorting’. Any folder which has a ‘.rfs’ file will be considered as a Category Folder and Folders without ‘.rfs’ file will be considered as Content Folders by our system. The ‘.rfs’ file will have XML (eXtensible Markup Language) structure in which 2 types of rules will be defined:-

- Detection-Rules – Which sorting rules is to be apply based on the conditions like keyword, extension, user, etc. will be defined here.
- Sorting-Rules – The format of renaming the files and folder with the path will be defined here. The different types of inputs that are to be taken will also be defined here.

- 2) Initially the user must select a "DRIVE" or "FOLDER" that he/she wants to sort. All the files and folders will be read and will be listed in the UnsortedView of the application as shown in GUI (Graphical User Interface) of the system in later parts. Only Content Folders will be added and shown and Category Folders will be ignored.

- 3) The user must then select "DRIVE" or "FOLDER" which is a Category Folder in the CategoryView. If the selected folder is not a Category Folder then the user will be prompted to make it a Category Folder by selecting a template from the application UI or manually making it. The Category & Sub-Category rules will be stored as a hidden ""rfs – Rules for File/Folder Sorting" and stored in the root of the selected DRIVE or FOLDER. Similarly, the CategoryView will be updated with only Folders having ‘.rfs’ file.

- 4) The can then select files and folder from UnsortedView and then press + button in the UI to add them to SelectionQueue. The SelectionQueueView will get updated with the files passed. The user can manage their selections with the + and – buttons in the UI.

- 5) According to the files and folders present in the SelectionQueue, rules will be matched in for the Category Folders in the CategoryView and compatible Category folders where the selections satisfy the conditions in the rules will be highlighted.
- 6) The user can then select only 1 Category Folder and then proceed to do sorting by clicking any of the three buttons:-
 - Pseudo Sort – The selections will be moved inside a folder “Unsorted” of the selected Category Folder.
 - Override Rules Sort – The selections will be moved directly/placed in root of the selected Category Folder.
 - Apply Rules Sort – The selections will be moved according to the rule format of the Category Folder.
- 7) The user will be prompted to confirm the sorting in a new Windows which shows the updated names and paths of the selections. The OK button must be clicked by the user so that the application can execute the actions.

If a file or folder is being transferred to a Category/Sub-Category folder and its content matches with an existing file inside the folder, then the file being transferred is a duplicate and a prompt will open to user to ask both files must be kept or only 1 is to be kept.

Goals

- *Set Icon of selected Category Folder*
 - Set Icon of selected folder by placing the .ico file inside the folder and using Relative path of icon; instead of the default way Windows sets the icon.
- *Create/update rules for selected Category Folder*
 - Create rules using template.
 - User manually specifies/edits rules.
 - Update previously created rules as per user preference.
- *Sort the files and/or folder into selected Category Folder*
 - Pseudo Sort - Place the file inside a folder named “Unsorted” inside the selected Category Folder.
 - Override Rules - Place in root of selected Category folder
 - Apply Rules - Place in selected Category folder by applying rules
 - Rename Files and/Folders
 - Create files and/folders by using templates and rules specified by the user.
- *Undo Sorting/Restructure files inside Category folder*

- Using log files to revert the files and/or folder to their previous unsorted location with their unmodified names.
- Using updated/Current rules, the software can be used to rename files and/or folders already sorted inside the Category folder using previous rules.

System Flow

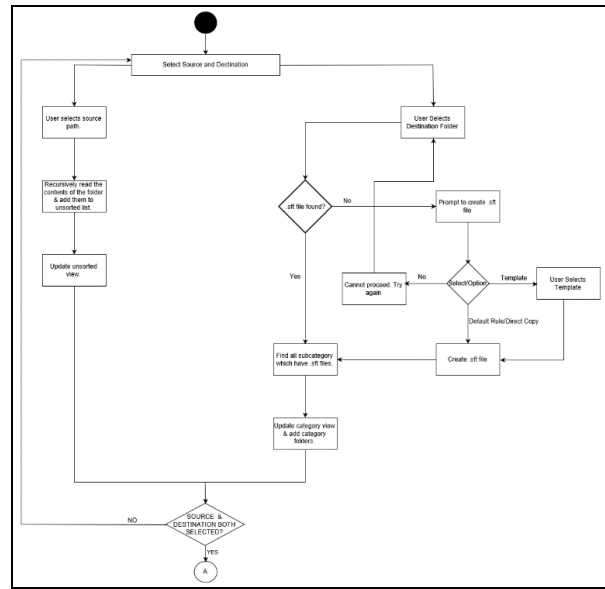


Fig. 2: System flow [Common part]

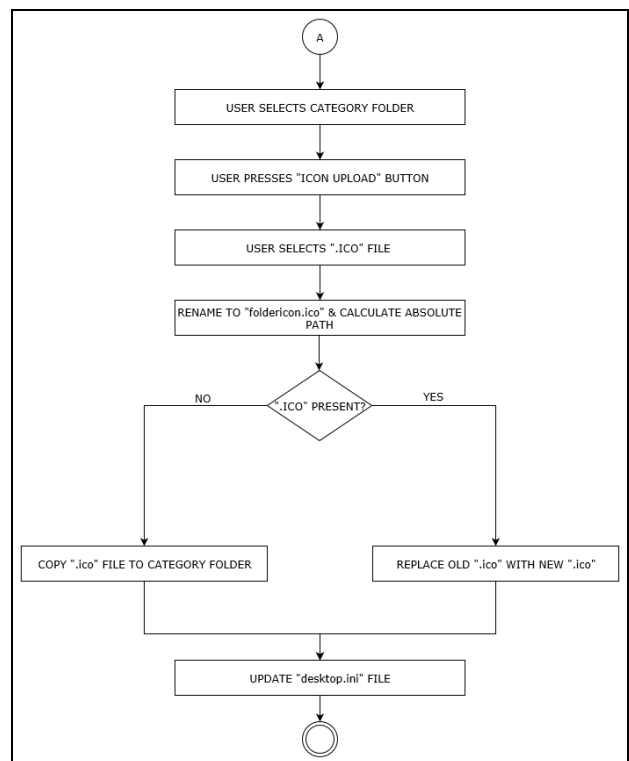


Fig. 3: Setting icon for Windows

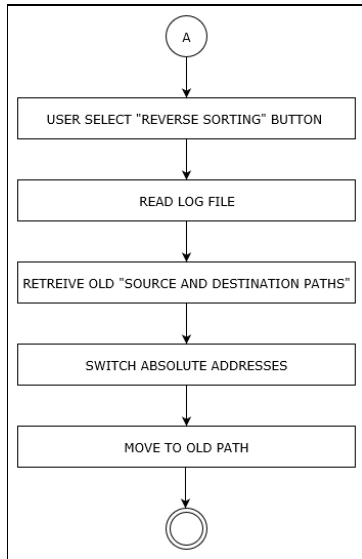
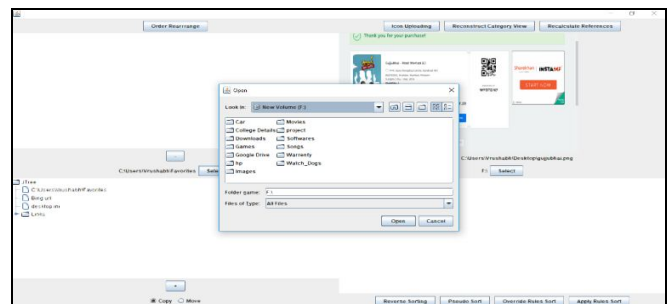
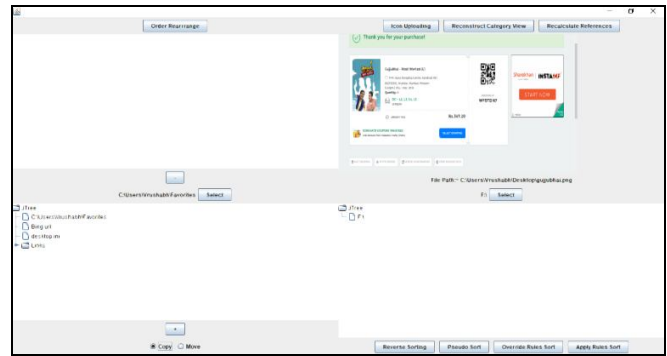


Fig. 4: Reversing changes



V. RESULTS

With the implementation and testing done at various stages during the design and implementation phases, we were able to achieve the intended results that we were looking for. The small modules implemented in the first phase gave us the understanding of what is achievable and where our concept needed changes. Accordingly the changes were made. The design of the rule file was finalized to add more functionality and of the system was developed to achieve the goals.

VI. FUTURE POSSIBILITIES

- In the future, the process can be automated by implementing an AI which can track user behaviour & storing habits. The system can learn user preferences & sort accordingly.
- The system can also be made to run in the background, where if new files and/or folders are detected sorting will be done automatically.
- A UI can be made to make the task of generating rfs file easier. Instead of typing and learning XML, user can drag and drop elements to write their own rules.

Advantages

- The files will be sorted according to the preference and workflow of the user.
- In case, if the user wants to revert back to the previous state then for that logs (log files) will be

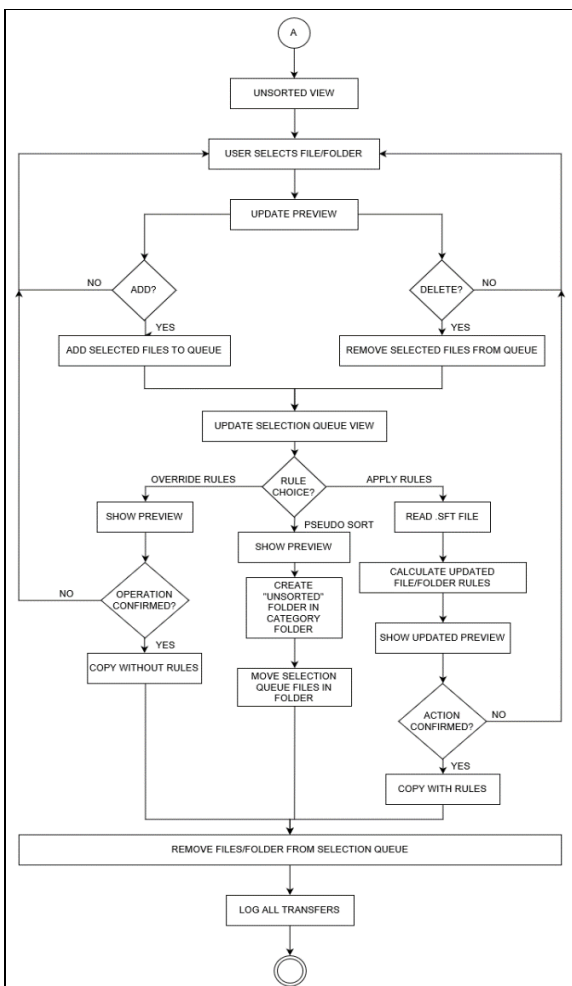


Fig. 5: Sorting Flow

Designing OF Smart file manager

maintained. So changes made by our system are reversible to an extent.

- Folder icons set with our system will keep their icons, even if their location in changed.

Disadvantages

- Computer may operate slowly when sorting process is going on.
- Certain new or unknown file formats may not be recognized by our system. If no rule is specified for handling them, they would be placed inside selected Category Folder without renaming or default rule may be applied.
- A novice user has the learn how to write his own rules in .rfs file. The rules definitions at the current might be difficult for the normal user to create but they can use templates made by other users.

Applications

1. The software can be used by the user to manage their personal files.

Example 1:- A user can sort his family photos by Date and tag the people inside the photos as metadata.

2. The software can be used by employees of businesses to manage their files and catalogue them to be used with their own Content Management Systems.

Example: - A Photographer can use the software to make a catalogue of all his/her commercial shoots according to events/types/dates/clients.

Example: - A programmer can sort his projects based on the technology used, programming language used, clients, dates, etc.

VII. CONCLUSIONS

For a user to find their data/files easily they must organise it in an efficient manner that suits their workflow which in turn makes the retrieval of files faster and more reliable. The best way to achieve this is to sort the data/files into folders by renaming them such that the user can easily identify and associate with; this is done by most of us manually. Our concept and project will provide a way to do this task in an efficient manner & simplify the task of renaming batches of files so that users can spend less time organising their files and more time in using them. Our system

will not make the user rely of our software for retrieval and use of data. The user will be able to use any of their File Managers/Browsers and Search Tools to effectively find their files.

VIII. ACKNOWLEDGMENT

We are grateful to all the wide panorama of people who provided us help and motivation to complete our paper, "Smart File Manager – A Category Based Sorting System". We owe a sense of gratitude to our honourable faculty, our H.O.D. & group guide Dr. Vinayak D. Shinde and co-guide Prof. Sunny B. Sall for stimulating our interest in the topic and supporting us in this endeavour.

REFERENCES

- [1] C. A. N. Soules and G. R. Ganger, ""Why can't I find my files? New methods for automating attribute assignment"," *Proceedings of the Ninth Workshop on Hot Topics in Operating systems, USENIX Association*, May 2003.
- [2] "MSDN Microsoft," [Online]. Available: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa365247\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa365247(v=vs.85).aspx).
- [3] "Stack Exchange," [Online]. Available: <https://unix.stackexchange.com/questions/32795/what-is-the-maximum-allowed-filename-and-folder-size-with-ecryptfs>.
- [4] "Techtarget," [Online]. Available: <http://whatis.techtarget.com/definition/metadata>.
- [5] "Wikipedia - File Manager," [Online]. Available: https://en.wikipedia.org/wiki/File_manager.
- [6] "Wikipedia - Desktop Search," [Online]. Available: https://en.wikipedia.org/wiki/Desktop_search.
- [7] "Apple - Spotlight," [Online]. Available: <https://support.apple.com/en-in/HT204014>.
- [8] "Void Tools - Search Everything," [Online]. Available: <https://www.voidtools.com/>.
- [9] "Google Desktop Search (Discontinued)," 2011. [Online]. Available: <http://desktop.google.com/>.
- [10] L. E. Page and S. M. Brin, "The Anatomy of a Large-Scale Web Search Engine," *Proceedings of WWW*, 1998.
- [11] "Knowledge File System -- A Principled Approach to Personal Information Management," *2010 IEEE International Conference on Data Mining Workshops*, 2010.
- [12] S. N. "A Cognitive Defense of Associative Interfaces for Object Reference - Draft," [Online]. Available: <http://people.gnome.org/~seth/storage/associativeinterfaces.pdf>, 2010. And Gnome Evolution Project, <http://projects.gnome.org/evolution>.

- [13] I. B. E. Zadok, "'Usenetfs: A Stackable File System for Large Article Directories'," 1998.
- [14] R. Grimes, "'Code Name WinFS: Revolutionary File Storage system Lets Users Search and Manage Files Based on Content'," *MSDN Magazine*, Jan 2004.
- [15] G. B. R. L. J. Gemmell, "MyLifeBits: A personal database for everything," *Communications of the ACM*, vol. 49, Jan 2006.
- [16] S. Cherry, "Total Recall," *IEEE Spectrum*, Nov 2000.
- [17] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin and D. Robbins, "Stuff I've Seen: A system for personal information retrieval and re-use," *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003.
- [18] K. Bakshi, D. R. Karger, D. Huynh and D. Quan, "Haystack: A customizable general-purpose information management tool for end users of semistructured data," *In Proc. CIDR*, 2005.
- [19] "Wikipedia - NEPOMUK," [Online]. Available: [https://en.wikipedia.org/wiki/NEPOMUK_\(software\)](https://en.wikipedia.org/wiki/NEPOMUK_(software)).