Kronosphere

A temporal visualization to replace the ailing hierarchal file system

Chris Harrison

chrios@rocketmail.com

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in the Department of
Computer Science, New York University

April 29, 2006

Approved: _____

Dennis Shasha, Research Advisor

Acknowledgements

Foremost, I would like to thank Professor Dennis Shasha for his guidance and encouragement. I am also grateful to my two co-researchers, Jeff Borden and Stacey Kuznetsov.

Research and Development

Kronosphere was developed by Chris Harrison and Stacey Kuznetsov over the course of the Spring 2006 semester. The text analytics package used to extract keywords was developed Jeff Borden.

Contents

1 Introduction

Today, users are bombarded with files. They not only need to sort files they create, but also manage documents received from others, perhaps via email or downloaded off the internet. Users are also taking a rapidly increasing number of digital photographs and storing them on their personal computers. Other media, like music and video, continue to fill users hard drives. Unfortunately, current systems lack an efficient mechanism to help users organize and efficiently access all of this data.

2 Problem

2.1 User Reliance

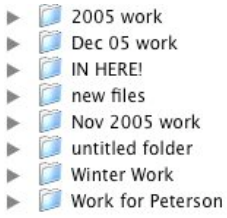
Hierarchical file systems were a natural extension of the way people file data in the real world. Early computer systems used flat or predefined file structures. These became increasingly cumbersome as technology advanced, storage increased, and the number of files stored on machines grew. Unix, initially developed in the 1960s, was the first system to debut a fully generalized hierarchical file system, now a standard feature on today's operating systems. This allowed users to "file" away documents into a hierarchy of directories. However, a hierarchical system is only an effective management structure if users are diligent and spend the time to organize documents appropriately. If documents are not sorted, or worse, mis-sorted, the system can become more unwieldy than a flat file system.

2.2 Hierarchical Overload

As directories become saturated with files, users create sub-folders to partition documents into smaller and more manageable sets. As users create and acquire more files, maintaining and navigating these increasingly deep organizational structures becomes more difficult and time-consuming.

2.3 Naming Ambiguity

Effective naming is vital to maintaining easy-to-navigate hierarchies. The user, rather than remembering the entire file structure, can scan over a list of directory names and choose the one that is most applicable to the target document (i.e. "Resumes", "Dec 05 work"). However, this system becomes unusable when directories are poorly or ambiguously named.



Ambiguously and poorly labeled folders

Beyond directories, users also rely on informative names to differentiate files. Inconsistent and vague naming leads to confusion, and several files may need to be opened before the correct one is located, even within a single directory. Most operating systems remedy this problem for image files by generating thumbnails. Images names, which are typically cryptic (i.e. IMG_1092.jpg), are no longer needed because the thumbnails provide sufficient distinction. However, current

systems do not provide a powerful was to see the contents of other media, especially text, without opening them.

Thumbnails provide distinction between image files, but other media types have no analogous mechanism.

2.4 Search Limitations

Searching files by content has become increasingly popular. Tools like Google Desktop Search and Apple's Spotlight are helping users sidestep their hierarchical file systems and locate files faster. However, these technologies are not particularly intelligent. For the most part, these systems use pre-build indexes to do full-text searches. This, of course, limits the content that can be searched to simply text files. However, users are amassing large collections of digital photographs, music and other media. Although some textual metadata can be extracted from these files, for example, an album name from an MP3's ID3 tag, the content remains unsearchable.

2.5 Versioning

Users create versions of documents for two reasons: It provides a safety net for accidental modifications, and offers a history of documents for reference. Although software exists that provides an integrated versioning solution for most operating systems, they are not designed for average consumers. Instead, users

frequently rely on simple naming conventions, like lettering or numbering (i.e. “resume1.doc”, “resume2.doc”, etc.). However, this system requires users to be diligent and accurate in creating and naming versions. It also causes additional bloat, and new directories may be created that just contain versions.

3 Existing Systems

3.1 Lifestreams

Lifestreams was a file organization model developed at Yale by Eric Friedman [2]. It displayed files as a flat, time-order series. Users could perform searches to narrow the number of files using full-text search or by navigating to a particular time period. The remaining items would be a subset of the entire corpus of documents. There were five operations: create a new document, clone a document, transfer a document, find documents, and summarize one or more documents. A comparison between Lifestreams and Kronosphere is detailed in Section 7.

3.2 Versioning Systems

Many versioning (also known as revision control) systems exist, including the popular CVS and Subversion. These sophisticated programs allow groups of users to access files from a central or distributed repository, make changes, and them commit them back to the repository. However, they are overly complicated for single-user use. Some file systems contain underlying mechanisms that allow for file versioning, including ZFS for OpenSolaris and Fossil for Plan 9.

However, even these are complex and rarely exposed to the user for version control, instead being used primarily to maintain data integrity.

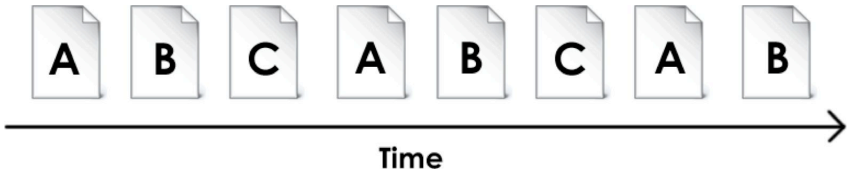
4 Concepts

4.1 File Relationships

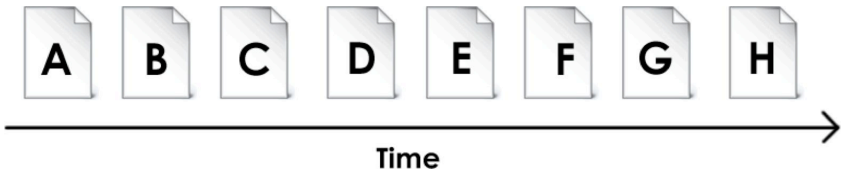
Files on personal computers are rarely unrelated. Users typically store files that are related to a limited number of topics, perhaps projects at work, hobbies, baby pictures, etc. If these relationships could be extracted and understood, they could be used to automatically group files together and facilitate access when working on a particular topic.

4.2 Temporal Relationships

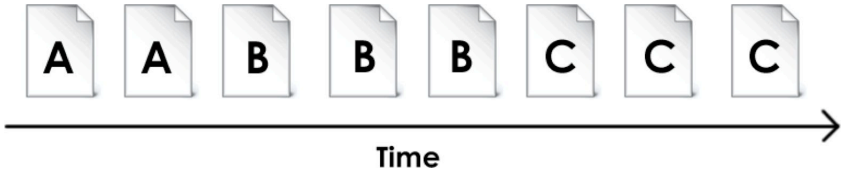
When faced with a set of tasks, humans generally group related activates together. This also true for how people use computers. It is uncommon for users to jump back and forth between many subjects. Typically, they will work on one subject, complete the work, and move on. Although users may be distracted by email or instant messages, they are likely to return to the original task. This often causes files that are modified or created sequentially to be related.



Uncommon behavior - User rotating between several topics (A, B, and C)

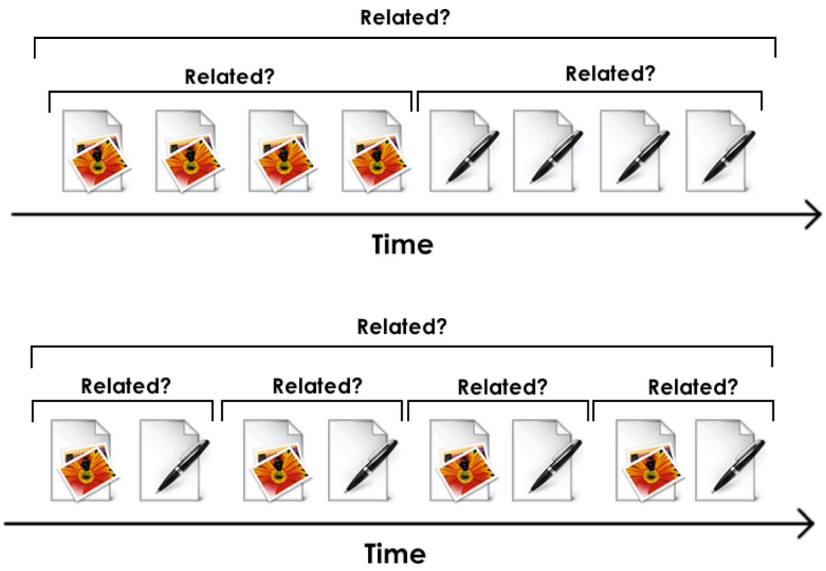


Common behavior - User moves sequentially through unrelated documents



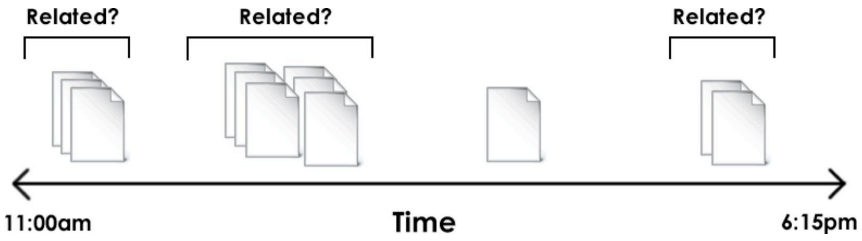
Common behavior – Users make use of several documents relating to a single topic before moving to the next topic

Interleaving of different media types can also be indicative of a relationship. For example, 20 photographs in a row might be a user's weekend fishing trip. However, if there are 10 documents interleaved with 10 photographs, there might not only be an overarching relationship, but also a close pair-wise relationship.



Possible relationships between contiguous and interleaved sets of varying media types

Of course, files are not accessed in regular intervals. When a user completes a task, there is usually some down time while files are located and material is reviewed. As work gains momentum, users create and modify documents more rapidly. This means that files relating to particular subjects are naturally clustered together in time.



Files modification and creation shown abstractly on a timeline

Photographs that are taken at roughly the same time are almost certainly of the same scene, or at the very least, captured at the same location. This unique temporal property can be used to cluster related images together reliably. Movies can also be grouped in this way.

Humans are also particularly adept at remembering chronology of items [4]. However, this skill is not leveraged during file navigation in traditional systems. Although files can be organized by modification dates, relative temporal distances are lost, and it is almost impossible to compare files stored different locations within a hierarchal structure without having to reorganize them.

4.3 Content Relationships

The mere presence of a word in a body text does not necessarily indicate its importance or relevance. For example an essay about school may mention a teacher's "leather suitcase", while an essay about the leather industry may include the word "leather" dozens of times. This does not indicate the two files are related, other than the fact they share this word. This effect often leads to confusing results in full-text searches.

What is needed is a mechanism that deduces what words best represent the text. These keywords are then saved with the file as metadata for future use. Searches using keywords will only yield documents with significant use of that word. This is not only more useful, but also quicker, as the search has to work on a smaller set of data. Determining relationships between files also becomes drastically simpler; files that share keywords (significant words) are likely to be on the same topic. Files that have closer relationships would have more keywords in common.

Two files tagged with keywords that have a strong relationship

5 Kronosphere

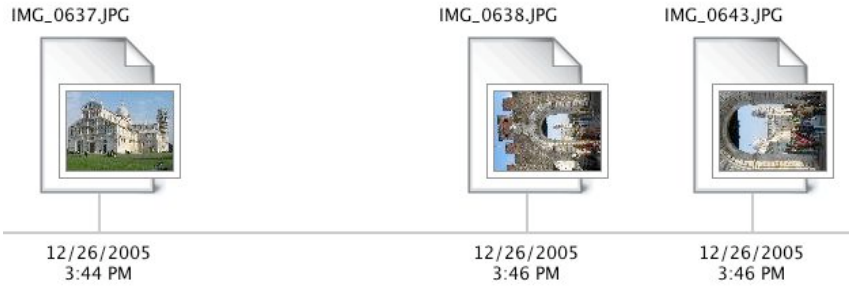
5.1 Objective

Kronosphere hopes to provide a compelling alternative to the hierarchical file system by providing a powerful and intuitive file navigation and search system. In order to minimize impact to the user, the current version of the application is designed to run along side the user's existing operating system and hierarchical file structure. Kronosphere can be used to search and locate files, and open them with traditional applications. The hope is, that one day, the default file system implementation will be like that of Kronosphere.

5.2 Timeline

The cornerstone of the application is its representation of the entire file system as a flat, timeline. Each time a file is saved to the system, either a new file being created or an existing file being modified, a new entry is created and displayed on the timeline. This was chosen for several reasons.

Foremost was the natural ability to accentuate the temporal locality between files relating to projects, as they are often modified together. Also, projects that have finite time span can be accessed quickly by navigating to that period. All of the files created or modified during the course of the project would be displayed. Photographs and movie clips can also be seen in chronological order, preserving temporal spacing, and allowing users to quickly reference groups of pictures taken at the same time.



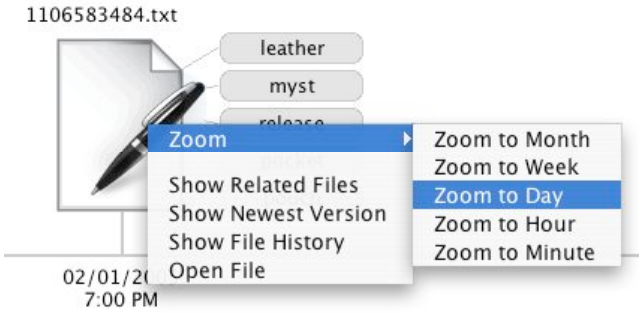
Pictures taken within a few minutes of each other

A timeline can provide an intuitive and non-obtrusive versioning mechanism. Each instance is essentially considered a new file, with data, including content, being separated from other versions. One can navigate the timeline to see when a file was modified. The earliest instance would be the file's creation. Kronosphere provides the ability to show the history of a file, or only the most recent version of a file.

Versions of a file edited throughout the day

The span of the timeline is controlled by several means. Users can use the horizontal scroll bar to quickly move between periods on the timeline. Users can also click to center the application on a particular time, including clicking a

document, which centers the span using the documents timestamp. Users can change the extent (zoom) using the scroll wheel.



The contextual menu when right clicking a file in the timeline. Users have the ability to zoom to a variety of extents, which are centered on the file. Users can also search for related files, display only the latest version, or show all of the versions.

Users have the ability to view time in a linear or exponential mode. This feature was developed under the assumption that newer files (and versions of files) are more important than older items. The exponential view affords users the ability to more readily recognize and access the most important information.

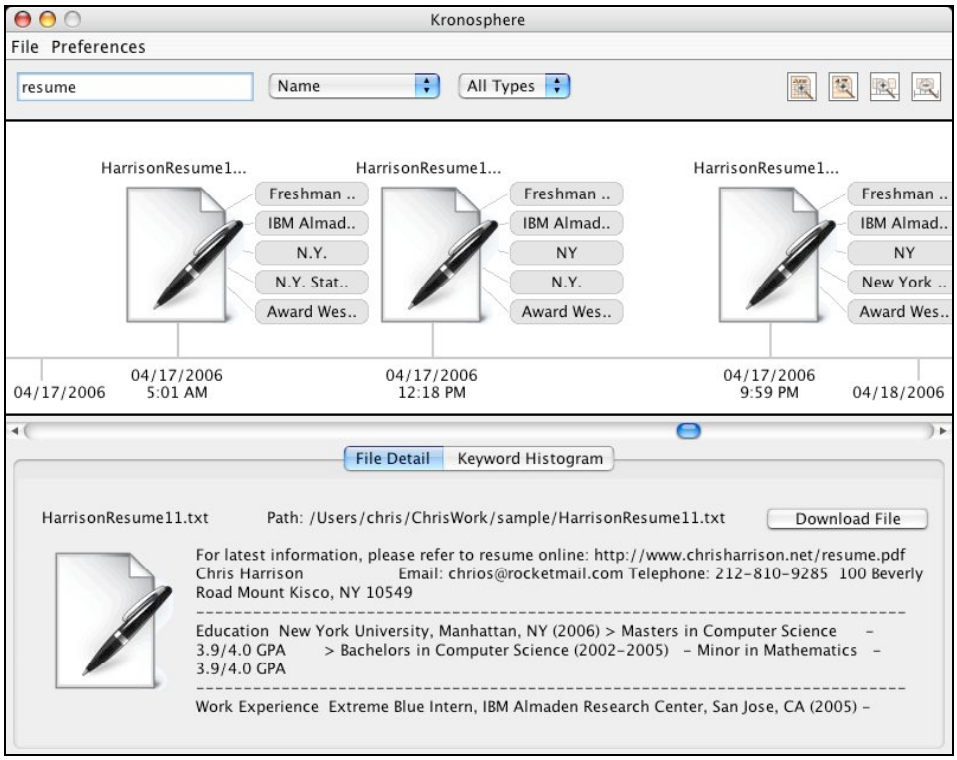
[linear and exponential pics]

5.3 Keywords

Files are processed when first loaded into the database. During this procedure, keywords are extracted that are representative of the file's content. Each media type uses a particular method to generate these keywords. Significant keywords and phrases are extracted from text documents using a text analytics package developed by Jeff Borden. Since keywords only represent the most significant

words in a text file, they can often be used to recognize the contents of the file. This helps disambiguate files on the content level, instead of solely relying on naming.

Some audio formats include textual data, such as genre, album, artist and year. MP3s, for example, contain this information in their ID3 tags, which can be read if present. Images can be described by keywords as well. Although automatically understanding image content is difficult (i.e. horses, cars, sunset), colors are easy to determine and provide some information. Thus, images are tagged with keywords that describe their most predominant colors.

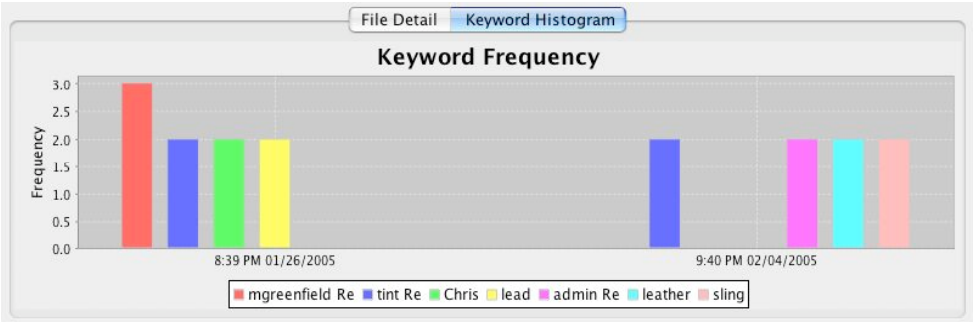


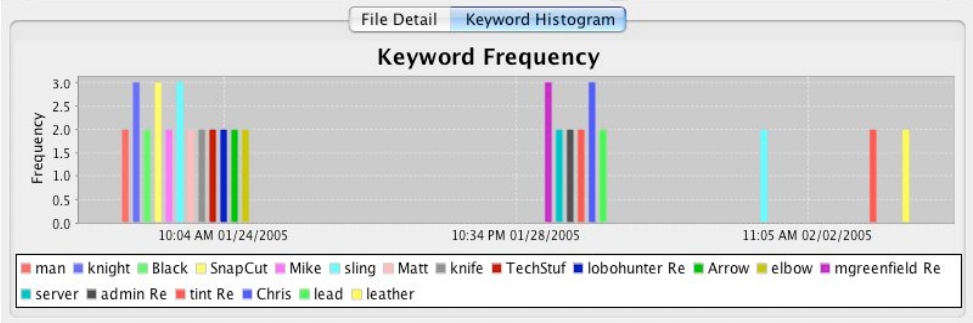
Kronosphere displaying three files on the timeline, tagged with keywords. Note keywords differ slightly between versions. The visibility of keywords in the interface can be set in the preferences.

Images are tagged with keywords describing their predominant color. Audio files containing textual metadata can be tagged with genre, album name, artist, and other information.

Keywords can also be used to find files similar to a source document. This can be achieved my accessing the database looking for other files in the system that share keywords with the source file. The more keywords they have in common, the more likely their content is similar. This can also be done with full-text, but is computationally expensive.

The frequency of keyword usage may be interesting to the user. As various projects are completed, certain keywords may peak in usage, and then drop off, giving way to new terms. Photographs, which are tagged by color, may show a prominence of red and yellow during fall and white during winter. This data can be visualized using a built in keyword frequency histogram.





Two histograms generated from different periods on the timeline. Keywords with frequencies less two are ignored.

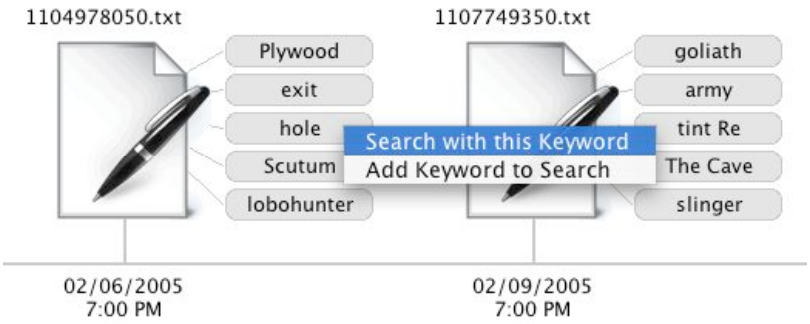
5.4 Search

Beyond the timeline visualization, Kronosphere has a robust search engine. Users can search the full content of text documents, as well as keywords associated with any media type. Name can also be searched alone, or in combination with the latter. Users can optionally narrow their search to a particular category of files, such as images or movies. The searchable time span is set by the user, which is accomplished by navigating the timeline to a particular period. As the user adjusts the timeline, either by panning or zooming, the search is dynamically updated.

The Kronosphere search bar - Users can enter the desired search term in the text field and use the two drop down menus to select what fields they would like to search as well as what file types they are looking for. Searches are invoked automatically. The four icons on the side allow users to jump to the current month, jump to the current day, expand the results to fill the, and zoom out to full extent.

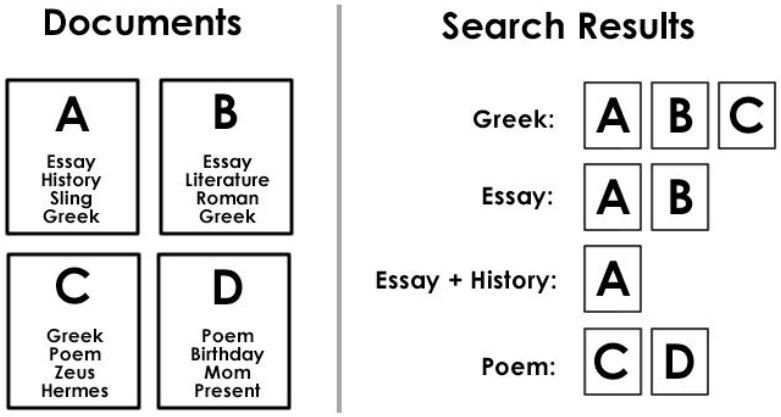
Keywords are shown radiating from files they are associated with. Users can right click a keyword and select “Search with this Keyword”, which then limits

the result to files containing that keyword. Users can also right click keywords and select "Add Keyword to Search" to narrow their results. This action can also be achieved by double clicking a keyword. Users also have the ability to find related items using the "Find related documents" menu option.



Two files tagged with keywords. Right clicking a keyword bring up a contextual menu with two options, as shown above.

Keyword search is particularly useful, and can act as a high-level categorization and hierarchical organizational system, even in a flat database. Files can be present in distinctly different result sets as they can be described by many keywords. Users have the ability to tag files by overriding the automatically generated keywords. This allows users to tag documents in a similar way to successful systems like del.icio.us and Flickr [1][3].



A set of documents tagged with keywords and the results from different searches. Note how keywords allow items to be categorized in different groups, and that mutli-keyword searches essentially create sub categories within larger categories.

Leveraging the Rapid Content Based Image Search technology developed in a parallel project, users can find images that are similar to a source image. This feature is accessible by right clicking on a image in the timeline, and selecting "Find similar colored images" or "Find similar images". This creates a special query, which returns only images that have a strong similarity to the original image.

The contextual menu for an image file. Users can search for images that are similar in color or content in addition to the options available for all file types.

6 Architecture

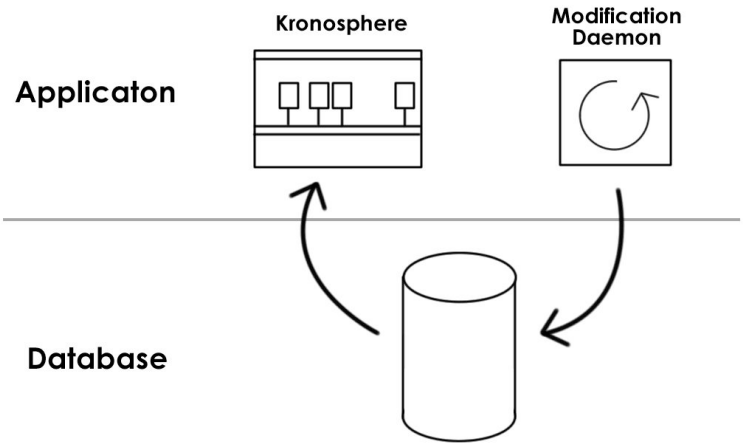
Kronosphere is comprised of three applications. A setup program must be run when Kronosphere is first installed. This application initializes the database, creates the appropriate tables, and scans the user's hard drive for files. This occurs in a two-stage process.

The first pass is required to initialize data structures, which are later used to perform keyword extraction for text documents. During this initial stage, an index containing the words found in all the text documents is constructed. When the pass is completed, the index can provide a list of files that contain a particular word. This is useful in two ways; it provides an efficient full-text search mechanism and the ability to compute the number of times a word is used in the set of documents. This data is used in the second stage of the setup.

The second stage operates on every media type, including text, images, and music. With the text index built, a Text Frequency Inverse Document Frequency (TFIDF) algorithm can be used to extract significant keywords from text files. This is done by comparing the usage of a word in a single document relative to the word's frequency in the corpus of documents. The relative frequency can be computed for every word in the document. The five most significant words are selected to be the text file's keywords. Keywords for other media types are extracted during this stage as well. Thumbnails, as well as search metrics (see Rapid Content Based Image Search for details) for images are created as well. A

record is created in the database for each file. Name, path, last modification date, file type, keywords, and other information is stored.

The second application is a daemon that runs invisibly in the background. It periodically sweeps the file system looking for changes. A snapshot of the database is downloaded periodically. If the file is not contained in the snapshot, it must be a newly created file, and thus added to the database. If the file is found in the database, the last modification dates are compared. If the timestamps are different, it is also added. The new record will contain all the necessary data, including newly extracted keywords.



Kronosphere provides an interface to access and visualize the files contained in the database. The modification daemon is responsible for monitoring the file system and adding records to the database as files are created or modified. The setup program, not shown, uploads the initial set of data.

Lastly, Kronosphere, the main application in the suite, provides an interface to access and search the database. It retains no information locally, instead relying on a constant stream of data from the database. The timeline application has two primary threads of execution: the graphical user interface and the database query

system. The GUI is a thin layer, providing a visualization for the information packaged and provided by the database thread. The database thread reads the interfaces state (time span, keywords, and other options) and constructs a query. When data is received, it is packaged into a data structure and passed to the GUI, which will continue operating on this list of files until a new set is supplied. This allows the GUI to remain interactive and smooth despite network congestion or latency. Certain interface elements can generate the creation of a new search, for instance, a search that finds related files or similar images. This is only overwritten when a new search is created.

Kronosphere periodically re-executes searches even if the user has not modified the query. This is done so that new or modified files that have been added by the modification daemon are displayed.

Each record in the database includes a last modified timestamp. The query engine uses this data to limit the files it operates on to those that fall within the span of the timeline. This speeds access to the database significantly, as file that are not in the current view, and would not be displayed, can be eliminated from the query immediately. As the user navigates the timeline, the start and end dates change. Each time a query is executed, the last modified parameters are updated to reflect the latest time span. This causes searches to always provide the most relevant results within the current view. For example, when searching for related documents, the query will only return the most related files within the current

span. To see the most related documents overall, one would have to zoom out to full extent. This feature is useful as a limited number of files can be displayed on the screen at once. As users zoom in, and files that are no longer in the time span disappear, the query thread will fetch new sets of files using the new time span, a repopulate the timeline.

7 Comparison to Lifestreams

In many ways, Kronosphere is a practical and functional implementation of the core features embodied by Lifesteams in that both systems provide a searchable, chronological representation of files. However, Kronosphere extends this concept in several significant wants.

Kronosphere implements these features directly on top of modern operating systems, which handle a variety of media through a wide assortment of applications. There is no "new document" feature in Kronosphere – document creation is left to other applications. Kronosphere merely provides a portal from which to access this content, but does not restrict the user in anyway. Users can continue to use their hierarchal directory structure, or store all files in a single root directory. Kronosphere's visualization is independent of the underlying structure.

Kronosphere's visualization differs in several distinct ways as well. Foremost is that files are shown with relative temporal distances, instead of a stack-like representation. As noted earlier, this can provide important relational clues

between files. Also, attaching files to a physical timeline allows for more intuitive navigation (pan left right using mouse or slider, zoom using scroll wheel, click on file to center).

In addition, it provides a powerful and working mechanism for keyword extraction, and thus a primitive, but useful form of summarization, as well as the ability to find related content. Keywords are also useful for search, as sets of files tagged with the same keyword(s) can be accessed immediately. Kronosphere is also extracting textual, and thus searchable, data from audio and image files, as well as common text documents like TXT, HTML, PDF and Microsoft Word. Also of note is the ability to search for images with similar color composition or content.

8 References

[1] del.icio.us – http://del.icio.us

[2] S. Fertig, E. Freeman, and D. Gelernter, *"Lifestreams: An Alternative to the Desktop Metaphor."* In ACM SIGCHI Conference on Human Factors in Computing Systems Conference Companion (CHI '96), pp. 410 - 411, ACM Press, 1996.

[3] Flickr – http://www.flickr.com

[4] M. Lansdale, "*The psychology of personal information management."* Applied Ergonomics, 19(1) (1988).