Extraction of Contextual Metadata from File System Interactions

Adaora Okoli[1] and Bernhard Schandl[2]

[1] Smart Information Systems GmbH, Vienna, Austria
a.okoli@smart-infosys.net
[2] University of Vienna, Department of Distributed and Multimedia Systems
bernhard.schandl@univie.ac.at

Abstract: Semantic systems improve information management by providing mechanisms to access objects by their attached semantic metadata elements rather than by their static location within a hierarchical structure. Semantic file systems apply these mechanisms to data stored in file systems in order to increase the quality of information retrieval. However, the effectiveness of semantic file systems depends strongly on the enrichment of files with relevant and comprehensive metadata, which is a tedious task if done manually. In this paper we present an approach that attains to disburden users from this expensive task by automatically generating useful metadata for files. It analyzes file system interactions and transforms them into contextual and semantic file relationship metadata, which can then be queried in order to search and retrieve information more efficiently. In this paper we give an overview on the algorithms we have used for this purpose, as well as an experimental evaluation of our prototype implementation, which demonstrates the effects of varying algorithm parameters.

1 Introduction

The rapid growth of storage capacities and the increasing number of files that users store on their hard disks pose a crucial challenge to file management—namely, of how to manage and organize the steadily growing number of files in a way that facilitates efficient file retrieval. Hierarchical, location-based file systems fail to appropriately cope with the changing situation: searching and identifying files successfully requires either information on the specific file name and location, or involve the task of having to traverse the file system directory tree and scanning all potential candidates. These methods can be very time-consuming and error-prone, in particular as users tend to remember the context and semantics of files more likely than exact file names and locations [TAAK04]. Moreover, current file systems are not able to reflect multiple file contexts sufficiently, which impedes efficient, flexible, and intuitive retrieval of files.

Semantic file systems aim to provide more user orientation in file management. Rather than identifying files by their fixed name and position in a strict hierarchy they can be addressed by their semantics, which are usually attached to files in the form of explicit metadata. This significant shift in paradigm allows for a more flexible and straightforward file retrieval processes. However, in order to be able to benefit from the inherent capabilities of semantic file systems, users are still required to enrich their files with meaningful and useful metadata, which is an expensive and complex task users might not always be willing to perform. Thus, to untie the basic usefulness of semantic file systems from explicit user participation, we propose to automatically generate metadata by analyzing file interaction sequences, i.e., tracing read and write access operations to file data.

The underlying assumption is that these interactions significantly reflect the users' temporal view on the contextual relatedness of files. Associating files with contextual metadata in the form of explicit relationships to other files enables users to locate files in a more intuitive, context-driven manner. When accessing a specific file, users are enabled to retrieve other files that are relevant in their current working contexts, based on these relationships.

Several projects have already engaged in analyzing file system interactions to generate file relationships [SG05, KP97, ALPB02]. However, as to our current knowledge, research on the inference of more specific file relationships has not been as extensive. In this paper, we intend to take existing approaches a step further. We do not only infer generic file relationships, but also determine the intent of specific interaction sequences. This allows us to ascertain even more expressive causal relationships between files.

2 Detection of File Context

The basis for the automatic generation of contextual relationships is the definition and detection of contexts, and the identification of files that are relevant in the same contexts. As users, while interacting with their files, commonly do not switch between unrelated goals and tasks or rotate unrelated files in different contexts, but stay at a task and sequentially access files that are relevant in the same context [Har06], it can be supposed that the temporal locality of file interactions corresponds to the relatedness of the accessed files. Files that have been accessed nearby in time presumably are closer contextually related than files that have been accessed within a larger temporal distance. Thus, it is the least obtrusive manner to approach the automatic generation of contextual file relationships by analyzing time-ordered file interaction sequences.

To detect and separate different file contexts within interaction sequences, we introduce the concept of *context scopes*. Similar to the *relation window* used in Connections [SG05] it is a clock-time based measure that specifies the maximum duration of task-related contexts. Files that have been accessed within these context scopes are considered to be related and relevant in the process of solving a specific task. This measure, if chosen appropriately, also supports the detection of context switches. In order to avoid missing relevant relationships or creating irrelevant relationships between files, we apply the following methods to improve the detection and correct delimitation of the varied file contexts within interaction sequences:

A new context scope starts at each file-opening interaction. — Instead of intersecting interaction sequences into static, global, time-based context scopes, we create a new context scope at each unique file-opening interaction. We only consider file-opening interactions

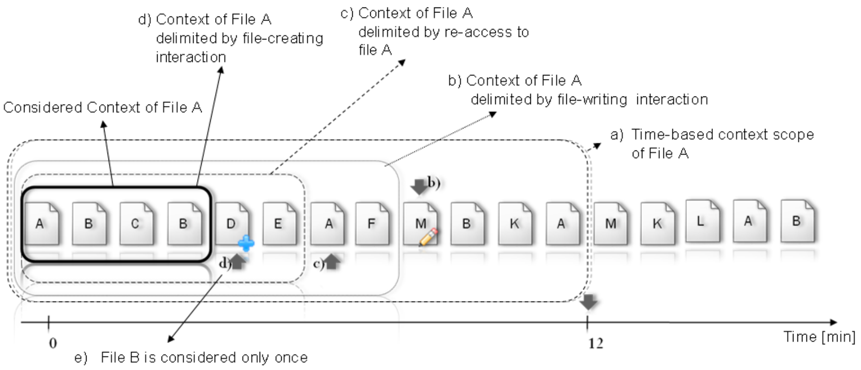
d) Context of File A delimited by file-creating interaction
c) Context of File A delimited by re-access to file A
b) Context of File A delimited by file-writing interaction
a) Time-based context scope of File A
Considered Context of File A
A
B
C
B
D
E
A
F
M
B
K
A
M
K
L
A
B
b)
d)
c)
0
12
Time [min]
e) File B is considered only once

Figure 1: Determining relevant file contexts: *(a)* context is delimited by a clock-time based measure (context-scope); *(b)* context is delimited by file-writing interactions; *(c)* context is delimited by re-accesses to the file; *(d)* context is delimited by file-creating interactions; *(e)* multiple accesses to the same file within the context of the initial file are considered only once.

(reading sequences), as these most significantly indicate the start or end of a file contexts.

Context scopes are delimited by non-reading interactions and file re-accesses. — We do not only delimit file contexts by predefined context scopes, but also by non-reading interactions, such as 'write' and 'create'. We consider each file-writing interaction (equivalent to file saving) as termination of a task and thus, the ending of all previous context scopes that still have been active. We interpret file-creating interactions as either the beginning or ending of a task, and thus as a context-intersecting interaction.

As it is usually not possible to observe file-closing interactions, we interpret re-accesses to a file within its already active context scope as file-closing and subsequent access to the file. Each file closing interaction (detected and identified by a file re-access interaction), similar to file-writing interactions, terminates the file's contextual influence and thus, its previous context scopes. Figure 1 illustrates the described context-detection mechanisms based on the analysis of an interaction sequence.

Semantic Distance (Steps) — We apply SEER's [KP97] basic concept of semantic distance to measure the relatedness of files. Each file access initiates a new context and thereby reduces the contextual relevance and influence of previously accessed files. The semantic distance or alternatively the significance of files decreases with the number of subsequently executed file accesses, which we refer to as steps.

Relationship Weight — As user behavior and file system interactions are not always context-aware, it must be considered that the generated relationships could have derived from irrelevant interactions. It is necessary to track these divergences in user behavior to provide users only with the most current and significant contextual relationships.

Figure 2 shows an example file relationship graph, whereas nodes represent files, and detected relationships are represented by directed and weighed edges. A directed edge is interpreted as the source node (source file, S) being relevant in the context of the sink (target file, T). For example, file E is relevant in the context of file B (see Figure 2).

The link weights (L_{ST}) represent the number of detected temporal relationships between source and target files.

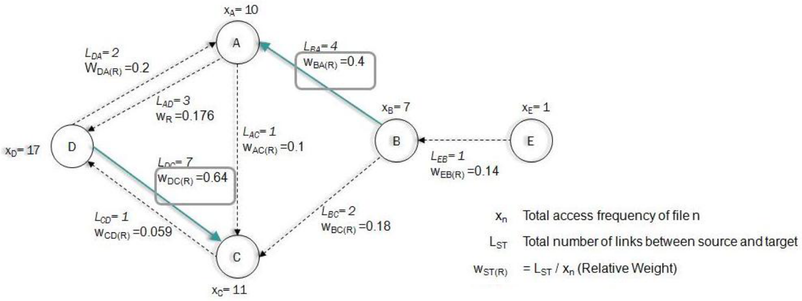

Figure 2: Relative Link-Weight Relation Graph

We calculate the weight of generated file relationships $w_{ST(R)}$ by correlating the total number of accesses to a target file n (x_n) with the number of accesses to a source file within the context scopes of the target file L_{ST}, so that $w_{ST(R)} = L_{ST}/x_n$. A source file can only be significantly related to a target file if it has been accessed within at least a minimum percentage (w_{min}) of the target file's context scopes. We consider back-to back file accesses as single file accesses to avoid distorting the computed total number of file accesses. A relationship, according to this approach, is significant, if the source file has been accessed within a given minimum percentage w_{min} of the target file's context scopes, so that $L_{ST}/x_n \geq w_{min}$. For instance, in Figure 2 a minimum percentage $w_{min} = 0.25$ leads to the identification of the relevant relationships $(B \rightarrow A)$ and $(D \rightarrow C)$[1].

The accuracy of the algorithm relies on the assigned thresholds for the context scope duration, the maximum steps and the minimum weight values. A preliminary examination of the effects of the parameters on the generated relationships is given in Section 4, and more detailed elaboration on the algorithms can be found in [Oko08].

3 Detection of Semantic Relationships

We define three types of semantic relationships between files, which can take the role of *input* and *output files* within a task and thus indicate a causal relationship. Data gained from a user study (see Section 4) that examined how users perform file-related tasks allowed us to specify the following interaction patterns that signify a semantic relationship. These derived patterns have to match exactly to produce semantic relationships between files:

Influence Relationship. A single output file (O) *is influenced by* at least one input file (I), if the input file has been accessed in a file-reading sequence in-between the interactions of opening and saving the output file. The output file must have already existed for a specified

[1] The access frequency x_n in Fig. 2 is equivalent to the sum total of the incoming and outgoing links of file n.

length of time before it was accessed within the task interaction. This algorithm conforms to the pattern depicted in Figure 3.

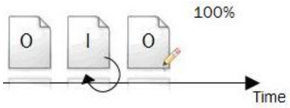

Figure 3: *Influence* Relationship pattern: at least one input file (I), one modified output file (O)

Involve Relationship. An output file *involves* one or more input files if the input files have been accessed in a file-reading sequence in-between the interactions of creating and saving the output file(s). This algorithm complies with the patterns illustrated in Figure 4(a) and (d). A considerable number of participants in the study executed the interaction pattern displayed in Figure 4(b), by accessing all input files at once before opening and writing to the output file. However, analyzing only interaction sequences does not allow to clearly isolate relevant input files from other previously accessed files. Thus, these input files cannot be assigned to an 'Involve' task. Similarly, the algorithm does not consider the first accessed input file in the interaction pattern depicted in Figure 4(c). However, as the remainder of the interaction pattern corresponds to the implemented pattern it would be still possible to associate the subsequently accessed input files to the created output file. We specify that the first input file must have been accessed within a specified maximum time after the output file has been created to be able to relate the input and output files.

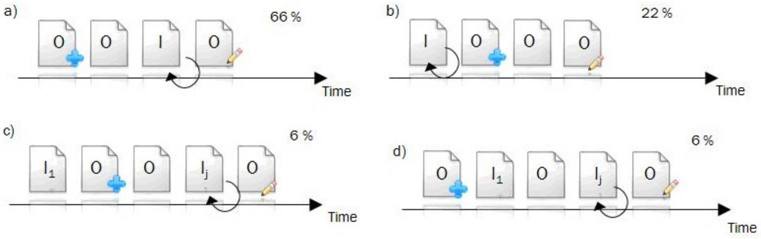

Figure 4: *Involve* relationship pattern: more than one input file, one created output file

Derive Relationship. We have implemented all revealed user interaction patterns that involved creating several output files and accessing a single input file (see Figure 5). The algorithm only considers accessed files as input files, if they that have been opened within a specified maximum time before or after creating the output files. Input and output files are considered as related only if the created output files have been subsequently modified, such as to infer that the input file had an effcet on the output file. Unlike the situation in Figure 4(b) and (c), we consider successive file-creating interactions as indicative of a specific task interaction and significant enough to deduce a causal relationship to a file that has been accessed directly before.

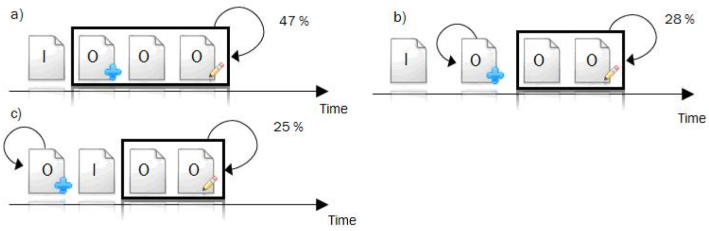

Figure 5: *Derive* relationship pattern: one input file, more than one created output file

4 Experiments

We have implemented a virtual file system, called *SileFS*, on top of our SemDAV semantic object repository [SK06, SH09]. This file system represents all file metadata (file and directory names, parent/child relationships, creation and update time, a.s.f.) as RDF triples, and allows to add semantic annotations of different kinds (tags, attributes, categories, and relationships) to these objects. It allows retrieval of annotated objects through a dedicated query API or through a standards-compliant SPARQL endpoint [CFT08]. This infrastructure allows us to capture all user interactions on files within the virtual file system (which can be accessed using the operating system's default file browser), and to perform analysis algorithms on it. The results of the analysis algorithms are then written back as file annotations and hence can directly support the user in their information retrieval tasks.

We undertook a preliminary test to examine how the variation of algorithm parameters affects the generation of contextual and semantic file relationships. The obtained experimental interaction data sets have been derived from logged interactions with the SileFS system. 16 users each had to fulfill 10 tasks like transferring information from one file to another, or combining data from multiple files. Our experiment resulted in a total of 69 interaction sequences with 11 different files and contained interaction sequences that correspond to the interaction patterns described in Section 3. The algorithm uses the measures presented in Section 2, of which we varied *context scope*, *minimum weight*, and *maximum step value*.

Running the implementation and varying context scope durations (5 minutes, 12 minutes and 30 minutes), minimum weights (0.3, 0.5 and 0.7) and step values (3, 5, and 10) led to different numbers of generated contextual and semantic file relationships.

Context Scope — By expanding the duration of the context scope, the algorithm generates a high number of relationships, as it considers files that have been accessed within a larger temporal distance. This has advantages for the identification of relevant file relationships, if the users executed few, but more time-consuming operations. In cases at which the users' interactions within a context scope are generally kept short, it is necessary to also increase the maximum step value in order to produce similar results. The chart resulting from the test run, which is depicted in Figure 6 shows that increasing the context scope duration, while keeping the maximum step value low (set to 2), has no significant impact on the number of generated relationships. This leads to the conclusion that the context scope

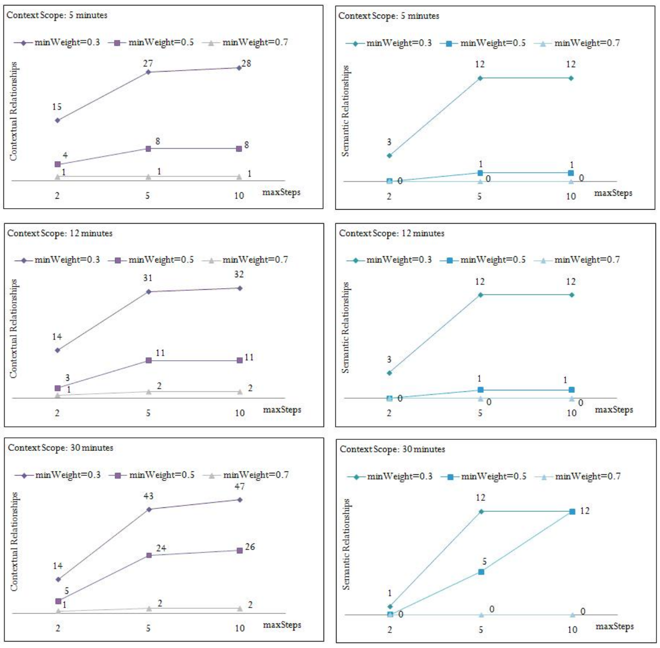

Figure 6: Experimental Analysis: Varying context scope size, minimum weight, and maximum steps

parameter is highly interlinked with the concurrent variation of the maximum step value. However, increasing context scope and maximum step values at the same time would most likely cause an increase in the number of false positives. A parameter setting with a large context scope and a low maximum step value can also lead to the miss of relevant relationships (false negatives), as it effects the computation of higher step values between files that actually are related. The given disadvantages of large context scopes imply that it is more appropriate to assign a mid-ranged value to the context scope parameter and to fine-tune the other two parameters in order to achieve more accuracy.

Maximum Steps — Increasing the maximum step parameter and leaving the context scope, as well as the minimum weight unchanged, yielded a relatively high number of relevant contextual and semantic file relationships. The tests showed that choosing a moderate context scope value is useful at avoiding and reducing the number of irrelevant results, which might derive from a high maximum step threshold.

Minimum Weight — The minimum weight value influences the number of generated semantic and contextual relationships proportionally and should ensure the exclusive generation of significant relationships. However, it was apparent that high weight values in fact did lead to the generation of significant, but also led to the generation of very few file relationships. The conclusion of the preliminary tests showed that the maximum step and the minimum weight values have a stronger impact on the number of identified file relations than the context scope. Choosing a medium value for these parameters could render a tolerable ratio between accuracy and number of produced relationships.

5 Conclusions

In this paper we have presented an approach how to analyze file interaction events in order to reduce the need for manual creation of semantic relationships between files. It is able automatically derive produce useful, user-oriented metadata from interactions with the file system without interrupting the user's interaction flow. The generated metadata are persisted in the form of explicit relationships between files, which can later be retrieved in order to enhance search and retrieval. Contrary to other approaches, our algorithm does not only infer generic contextual file relationships but is able to determine the intent of interactions and in consequence is able to deduce causal, semantic file relationships. In the future we plan to extend our system by using more features to detect file relationships, e.g., by considering file annotations like tags, or file contents, and to improve our prototype implementation w.r.t. its performance and its ability to analysis on-the-fly in order to generate file relationships in real time.

References

[ALPB02] A. Amer, D. D. E. Long, J.-F. Paris, and R. C. Burns. File access prediction with adjustable accuracy. In *PCC '02: Proceedings of the Performance, Computing, and Communications Conference, 2002. on 21st IEEE International*, pages 131–140, Washington, DC, USA, 2002. IEEE Computer Society.

[CFT08] Kendall Grant Clark, Lee Feigenbaum, and Elias Torres. *SPARQL Protocol for RDF (W3C Recommendation 15 January 2008).* World Wide Web Consortium, 2008.

[Har06] Chris Harrison. Kronosphere - A temporal visualization for file access. Master's thesis, Department for Computer Science, New York University, USA, May 2006.

[KP97] Geoffrey H. Kuenning and Gerald J. Popek. Automated hoarding for mobile computers. In *SOSP '97: Proceedings of the sixteenth ACM symposium on Operating systems principles*, pages 264–275, New York, NY, USA, 1997. ACM.

[Oko08] Adaora Okoli. Extraction of Contextual Metadata from File System Interactions. Master's thesis, University of Vienna, 2008.

[SG05] Craig A. N. Soules and Gregory R. Ganger. Connections: using context to enhance file search. *SIGOPS Oper. Syst. Rev.*, 39(5):119–132, 2005.

[SH09] Bernhard Schandl and Bernhard Haslhofer. The Sile Model – A Semantic File System Infrastructure for the Desktop. In *Proceedings of the 6th European Semantic Web Conference (ESWC 2009), Heraklion, Greece*, 2009.

[SK06] Bernhard Schandl and Ross King. The SemDAV Project: Metadata Management for Unstructured Content. In *CAMA '06: Proceedings of the 1st International Workshop on Contextualized Attention Metadata: Collecting, Managing and Exploiting of Rich Usage Information*, pages 27–32, New York, NY, USA, 2006. ACM Press.

[TAAK04] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. The perfect search engine is not enough: A study of orienteering behavior in directed search. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422, New York, NY, USA, 2004. ACM.