社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－DBS－143（85）
2007／7／4

ファイル検索におけるアクセスログから抽出した関連度の利用

渡部 徹太郎† 小林 隆志†† 横田 治夫†††,†

† 東京工業大学大学院情報理工学研究科計算工学専攻 〒152-8552 東京都目黒区大岡山 2-12-1
†† 名古屋大学大学院情報科学研究科 〒464-8601 愛知県名古屋市千種区不老町
††† 東京工業大学学術国際情報センター 〒152-8552 東京都目黒区大岡山 2-12-1
E-mail: †tetsu@de.cs.titech.ac.jp, ††tkobaya@is.nagoya-u.ac.jp, †††yokota@cs.titech.ac.jp

あらまし 近年, ファイルシステム内のファイルの数が爆発的に増加しており, 既存のファイルシステムでは, 利用者が望むファイルを容易に見つけられないという問題点がある. この問題点に対し, デスクトップサーチなどの全文検索による解決策が提案されているが, キーワードに関連していてもキーワードを含まない画像ファイル, データファイル等を探し出すことはできない. 特に同一ディレクトリ下に無いような場合には, それらのファイルの検索は困難になる. 本稿ではファイルサーバのアクセスログから予めファイル間の関連度を抽出し, キーワードで検出されたファイルと関連度の高いファイルを提示することで, 全文検索できないファイルを含んだ検索を可能とするシステムを提案する. また, ファイル間の関連度を算出する部分までを実装し, 被験者実験を行い関連度の有効性を確認した.
キーワード デスクトップサーチ, ファイル検索, アクセスログ解析, ファイル間関連度

Application of Relationships Derived from Access Logs to Retrieve Files

Tetsutaro WATANABE†, Takashi KOBAYASHI††, and Haruo YOKOTA†††,†

† Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan
†† Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan
††† Global Scientific Information and Computing Center, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
E-mail: †tetsu@de.cs.titech.ac.jp, ††tkobaya@is.nagoya-u.ac.jp, †††yokota@cs.titech.ac.jp

Abstract Since the number of files in a file system explosively increase in recent years, it becomes very hard for users to find desired files from them. To attack the problem, desktop search functions using a kind of full text search methods have been developed. However, the files not including the given keywords, such as picture and experimental data files, cannot still be found by such methods. It becomes harder for the files located in different directories, even if they have some association with files including the keywords. In this paper, we propose a method of retrieving such files for given keywords utilizing the relationship information derived from file access logs in a file server. The proposed method calculates the relationships from the file open and close time information in the log. We evaluate the recall of the derived set based on testers answers.
Key words desktop search, retrieving files, analyzing access log, relationship between files

1. はじめに

近年, ファイルシステム内のファイルの数, サイズ共に爆発的に増加している. Nitin Agrawal らが 2000 年から 2004 年の間, ある企業内の 6 万台の Windows PC のファイルシステムを調査した報告 [1] によれば, PC 内のファイルの数は, 5 年間で約 3 倍になり, 平均ファイルサイズも約 2 倍になっている.

現在, オペレーティングシステムのファイルシステムではディレクトリ階層構造が提供されているが, 膨大なファイル群に対して適切にディレクトリ階層を構成することは難しい. また, 全てのファイルが適切にディレクトリ階層に格納されていたとしても, ファイル名が不適切であったり, ファイルが深い階層に格納されていた場合, ディレクトリ構造を辿るだけでは欲しいファイルを探すのは困難である.

そのような背景から, ファイルシステムに対して全文検索を提供する, Google Desktop [2], Windows Desktop [3], Spot-

light [4], Namazu [5] を利用したファイル検索などが用いられてきた．しかし，全文検索の対象は主にテキストファイル，pdf ファイル，doc ファイル等の構造化された文書ファイル，電子メール等であり，それら以外の画像ファイルやデータファイル等は検索できない．例えば，論文の図の基となっている画像や，グラフの基となっている実験データなどである．また，全文検索の対象となるファイルの種類であっても，キーワードを含んでいなければ検索できない．しかしこのようなファイルを検索したいという要求は高い．

この要求に対し，画像に対するキーワード検索では Google Image Search [6] が提案されており，これは HTML 等の文書ファイルからの参照関係を用いた検索方法を行っている．この方法はウェブ上の画像に対しては効果はあるが，一般のファイルでは基本的には関連ファイル中に参照情報が無い場合が多く適用できない．

本研究の大きな目的は，上述の全文検索ができないファイルに対しても，ファイル間の関連等を利用することで，検索を可能にすることにある．

この検索を実現するために，本研究ではファイル間の関連として頻繁に同時に使うファイル群に着目した．コンピュータ上である作業をする際，関連するファイルを同時に開いて，参照しながら，あるいは必要な部分をカットアンドペーストしながら進めることが多い．このような場合，作業毎に使用するファイル群は類似するため，頻繁に同時に使われるファイル群は関連が高いと考えられる．

本稿では，キーワードによる全文検索結果と，それらと同時に使用されたファイル群を提示することで，キーワードが含まないが関連性の高いファイル群を提示する検索方法を提案する．また，ファイルサーバのアクセスログから各ファイルの使用時間情報を抽出し，ファイル間の関連度を算出する手法を提案し，実験により提案した関連度を評価する．

以下に本稿の構成を述べる．まず 2. 節では提案手法の概要について述べる．次に 3. 節では評価の対象とする関連度抽出手法を説明する．そして 4. 節では関連度抽出手法の実装と，その評価実験について述べる．5. 節では関連研究について述べ，最後に 6. 節でまとめと今後の課題を述べる．

2. 提案手法の概要

2.1 提案手法の概要

本研究で提案する検索手法では，ファイルのメタデータや，アクセスログから抽出したファイルの使用情報を利用して，ファイル間の関連度を算出する．ファイル検索手法は検索要求があった時に，まずそのキーワードで全文検索を行い，その検索と関連度が高いファイル集合を提示する方法である．この方法により，実際にはキーワードを含まなくても，そのキーワードに関連したファイル群を検索することが出来るようになる．

2.2 ファイル検索方法

本稿で提案するファイル検索手法を説明する．ファイル検索の際には，まずそのキーワードで全文検索を行い，そのキーワードを含む文書ファイルを検出し，その結果のファイル集合を F とする．本手法では全文検索できないファイルを検索するので，ファイルシステム内の F 以外のファイル集合 $\bar{F}$ が検索の対象となる．またファイル x とファイル y の関連度を $R(x, y)$ とし，ファイル集合 F とファイル x に対して，F の要素の中でファイル x と関連しているファイルの集合を $P(F, x)$ とする (図 1 参照)．ある検索対象ファイル $x \in \bar{F}$ の得点は ε を定数として，以下の式で算出される．

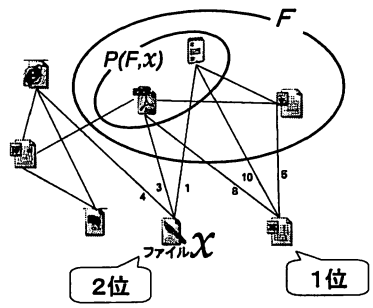

図 1 検索の例

$$Point(x) = \Big(\sum_{f \in P(F,x)} R(f,x) \Big) \times |P(F,x)|^{\varepsilon}$$

この式により，より F の要素との関連度が大きく，さらにより多く F の要素と関連しているファイルに高い得点が与えられる．このようにして全ての検索対象ファイルの得点を算出し，得点上位から検索結果として表示する．

3. アクセスログを利用した関連度の抽出

本研究では検索に必要な $R(x, y)$ を算出するためにファイルサーバのアクセスログからユーザのファイル使用情報を抽出する．しかし，ファイルサーバのアクセスログには問題が幾つかあり，ファイル使用情報を的確に抽出するためにはアクセスログデータのクリーニングが必要である．

以降ではまずファイル使用情報の抽出方法を説明し，その情報を基に $R(x, y)$ を算出する方法を説明する．

3.1 ファイル使用情報の抽出

まず，ファイルサーバのアクセスログからそれぞれのユーザ毎にファイルのオープン／クローズの情報を取得し．ファイル毎の使用時間情報を抽出する．加えて，ファイルサーバのアクセスログにはシステムファイル等のアクセスも含め多様な情報が含まれるため，対象とする利用者が実際に編集するファイルの種類のみを解析の対象とするためのフィルタリングを行う．

対象を Windows に限定し，Samba を利用したファイルサーバのアクセスログを解析した結果，アクセスログからファイル使用情報を抽出するためには以下の問題があることが分かった．

A) いくつかの理由で，あるオープンに対するクローズが取れず，正確な使用時間が計算できない場合がある．

B) ユーザはファイルを開いたまま席を立つ場合がある．

C) ファイルをロックをするアプリケーションとメモリに読み込んで，ファイルのロックを開放してしまうアプリケーションがあり，後者の場合はオープン／クローズが実際のファイルの

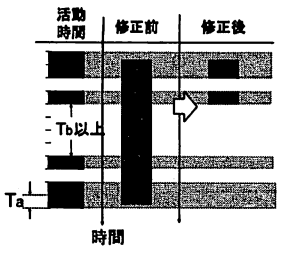

図 2　問題 (B) への対処

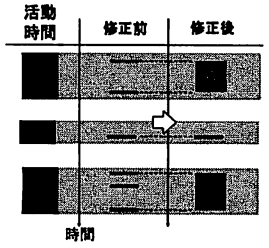

図 3　問題 (C) への対処

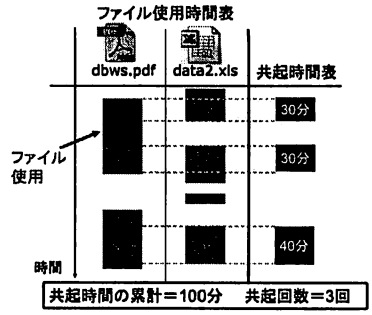

図 4　共起時間表の例

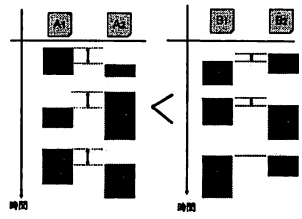

図 5　ルール 3 の例

使用時間と一致しない.

これらの問題を以下のように対処する.

(A) への対処は，ログを先頭から探索していき，オープンを見つけたら，対応するクローズを再帰的に探す．対象のログの末尾まで対応するクローズが発見できない場合は，そのオープンの直後にクローズを補完する.

(B)，(C) への対処のために，まず以下の情報をアクセスログから抽出する.

ア)　ユーザの活動時間:ユーザのアクセスログから T_a の間隔でログの有無を調べ，ログがあれば活動していたと見做し，活動の有無を時系列でリストにしたもの.

イ)　直ぐに閉じてしまうファイルタイプのリスト:アクセスログから拡張子別にファイル使用時間の平均を調べ，平均が T_c 以下のファイルタイプをリストにしたもの.

(B) のログへの対処法は，まず (ア) より活動時間外のファイル使用を消す．加えて，T_b 以上活動時間が空いた場合は，ファイルを閉め忘れて帰宅したと見做して，それ以降のログを消す (図 2 参照).

(C) のログへの対処法は，(イ) の対象となるファイルタイプに対して，(ア) の活動時間内で最初のファイルアクセスから最後のファイルアクセスまでをずっと使用していたと見做して，使用時間を拡大する (図 3 参照).

以上の操作でアクセスログを実際のファイル使用時間に類似させ，作成した使用時間を表したデータを「ファイル使用時間表」と呼ぶ.

3.2　関連度の算出

3.2.1　関連度の大小関係

前節で作成したファイル使用時間表を基にファイル間の関連度を計算する．関連度の算出方法の説明の前に，関連度の大小関係を定義する.

本研究では，関連度の高いファイル同士は長い期間，各作業の度に，同じタイミングで利用されるものと仮定する．そのため，関連度の大小関係を考慮する上で以下の 4 つのルールを導入した.

ルール 1　共起時間の累計が長いほど関連度が高い.

ルール 2　共起回数が多いほど関連度が高い .

ルール 3　ファイル使用開始パターンが類似しているほど関連度が高い.

ルール 4　共起の間隔が離れているほど関連度が高い.

これらのルールについて例を挙げて説明する.

ルール 1，ルール 2 に関して，ファイル使用時間表が図 4 の様になっている場合，ファイル使用時間表の重なっている部分を「共起時間表」と呼び，この例では共起時間の累計が 100 分であり共起回数が 3 回である．これらの値が大きいほど関連度が高いとする.

ルール 3 に関して，図 5 の A_1, B_2 の方が B_1, B_2 よりファイル使用開始時間が近いものが多く，より関連度が高いする.

ルール 4 に関して，図 6 の C_1, C_2 と D_1, D_2 を比較すると，D_1, D_2 の方が 2 番目の共起と 3 番目の共起の間に長い間隔がある．これは長期間経過後も二つのファイルを一緒に使っていたことになるので，関連度は高いとする.

3.2.2　関連度の算出

上述の大小関係を考慮した関連度の算出方法を説明する．二つのファイルの共起時間の合計を T，共起回数を C，ファイル使用開始パターンの類似度を P，共起の間隔度を D としたとき，関連度 R は $\alpha, \beta, \gamma, \delta (0 \leqq \alpha, \beta, \gamma, \delta \leqq 1)$ を定数として

$$R = T^{\alpha} \cdot C^{\beta} \cdot P^{\gamma} \cdot D^{\delta}$$

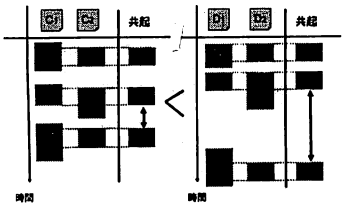

図 6 ルール 4 の例

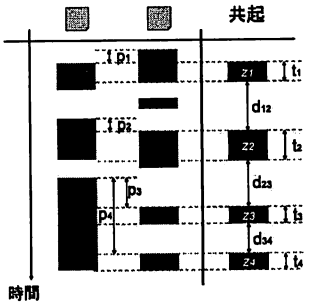

図 7 関連度計算で用いる値の例

となる．また同時に使用したことがない二つのファイルの関連度は 0 とする．加えて二つのファイルが同じディレクトリに属している場合，すでに使用者によって関連があると分類されているので，検索対象から除くために関連度を 0 とする．各定数の決定法は後の 3.2.4 節に記す．

各値の算出方法を説明する．図 7 を参照されたい．二つのファイルの共起時間表の各共起を $\{z_1, z_2, \cdots, z_n\}$ とし，各共起の時間を $\{t_1, t_2, \cdots, t_n\}$，共起の間の時間を $\{d_{12}, d_{23}, \cdots, d_{(n-1)n}\}$，さらに各共起の基となっている二つのファイル使用の使用開始時間の差を $\{p_1, p_2, \cdots, p_n\}$ として，各値は下記のように計算する．

$$T = \sum_{i=1}^{n} t_i \qquad C = n$$

$$P = \begin{cases} 1 & \forall i \ \ p_i = 0 \\ (\sum_{i=1}^{n} p_i)^{-1} & o.w. \end{cases} \qquad D = \begin{cases} 1 & n = 0 \\ \sum_{i=1}^{n-1} d_{i(i+1)} & o.w. \end{cases}$$

3.2.3 関連度閾値の算出

ある全文検索結果のファイル集合の各要素に対して，共起しているファイルは大量にあり，共起しているファイルがすべて関連するわけではないため，全ての要素をファイル間の関係として利用すると，ユーザが目的のファイルを見つける際に効率が悪くなる．そこで，本研究では以下に述べる方法で閾値 $\bar{G}$ を算出し，関連度が $\bar{G}$ 未満の関連をフィルタリングする手法を提案する．

提案手法では，$\bar{G}$ を教師付学習により計算する．学習セットとしてファイル集合 F を用意する．各 $f \in F$ に対して，利用者が与えた関連性の正解情報を用い，後述する方法でファイル別

関連度順位	C(f)内のファイル	ユーザーによる評価	fとの関連度
1	C2	○	R(f,C2)=10
2	C8	○	R(f,C8)=5
3	C5	○	R(f,C5)=1
4	C3	×	R(f,C3)=0.5
5	C7	○	R(f,C7)=0.2 G(f)
6	C4	×	R(f,C4)=0.15
7	C5	×	R(f,C5)=0.1
8	C6	×	R(f,C6)=0.05

図 8 閾値の決め方の例

閾値 $G(f)$ を計算し，その平均値 $\bar{G}$ を以下のように計算する．

$$\bar{G} = \frac{\sum_{f \in F} G(f)}{|F|}$$

次に $G(f)$ の算出方法を説明する．$C(f)$ をファイル f と共起しているファイルの集合とし，$C(f)$ の各要素 c_i に対してファイルの使用者が見て関連があるものを○，関連がないものを×の二値に分類する．次に $C(f)$ の要素で最も f との関連度が最も低く，○の評価をついている要素 c_k を求め，f に対する閾値 $G(f) = R(f, c_k)$ と求める．言い換えると，$C(f)$ 内の関連度 $G(f)$ 以上の要素の集合が○の再現率を 100 %とするような $G(f)$ を求めている．

図 8 に例を挙げて説明する．ある f に対して $C(f) = \{c_1, c_2, \cdots c_8\}$ であり，各共起ファイルに対して，ユーザの評価，関連度が付加されている．また，簡単のため関連度の順位で並べてある．この時，関連度が最も低く，○の評価をついている要素は C_7 となり，$G(f) = R(f, c_7) = 0.2$ とする．

3.2.4 関連度のパラメータの決定

前節で算出した閾値 $\bar{G}$ を基に関連度のパラメータセット $(\alpha, \beta, \gamma, \delta)$ を決定する方法を説明する．まず前節と同じ学習セットのファイル集合 F の各要素 $f \in F$ に対して，$C(f)$ を考え，$\hat{C}(f) = \{c | c \in C(f), R(f,c) \geqq \bar{G}\}$ とする．つまり $\hat{C}(f)$ は $C(f)$ の中で関連度が $\bar{G}$ 以上の要素の集合である．次に各 f に対して再現率 $Recall(f)$

$$Recall(f) = \frac{\hat{C}(f) \text{ 中の○の数}}{C(f) \text{ 中の○の数}}$$

を計算し，全ての $f \in F$ に対し，$Recall(f)$ の平均値を $\overline{Rec}$ とする．ここでパラメータセット $(\alpha, \beta, \gamma, \delta)$ で関連度を計算したときの $\overline{Rec}$ を $\overline{Rec}(\alpha, \beta, \gamma, \delta)$ と書き直すと，それを最大化するパラメータセット $(\alpha^*, \beta^*, \gamma^*, \delta^*)$ は以下のように計算する．

$$(\alpha^*, \beta^*, \gamma^*, \delta^*) = argmax(\overline{Rec}(\alpha, \beta, \gamma, \delta))$$

4. 評価実験

本研究では，関連度の抽出までを実装した．また関連度の評価法として，あるファイルに対して共起している閾値以上のファイルの再現率と適合率を用いた．

4.1 実験環境

ファイルアクセスログの採取には，現在広く用いられているWindows互換ファイルサーバのSambaを用いた．本実験では，ユーザの環境に特別なアプリケーションをインストールしなくても，ファイルの使用時間を抽出することが出来るためSambaを用いたが，Samba以外のファイルサーバーであっても，ファイルのオープン／クローズ情報が取得出来るものなら同様に提案手法を適用することが出来る．

実験では，ファイルサーバSamba2.2.3aをログレベル2で実行し，それをWindowsXPから二人の被験者に約4ヶ月間使ってもらい，アクセスログを採取した．被験者はこの期間に学会発表の準備などがあり，それに向けての論文のテキストファイル，画像ファイル，実験のデータファイルなどのアクセスログを中心に採取した．システムファイル等のアクセスは無視するように，解析対象の拡張子はbib, doc, gif, htm, html, jpg, mpg, mpeg, pdf, ppt, tex, txt, xlsとした．また3.1節の各定数は$T_a = 30$[分]，$T_b = 5$[時間]，$T_c = 10$[秒]とした．

4.2 実　　験

まず，それぞれの被験者のログからファイル使用時間表を作成した．次に全ファイルの中からランダムに選び，被験者に○×評価を行ってもらった．評価を行ったファイル数と全ファイル数は以下の通りである．

	被験者A	被験者B
評価を行ったファイル数	104	134
全ファイル数	420	351

上記の評価を行ったファイル集合の中からランダムに半分のファイルを選択し学習用セットとし，そこから計算された関連度のパラメータと関連度閾値は以下の通りである．

$(\alpha^*, \beta^*, \gamma^*, \delta^*)$	(0.2, 0.3 , 0.3 ,0.1)	(0.1 , 0.6 , 0.8, 0.2)
$\bar{G}$	0.628	0.150

次に，これらの値を用いて評価セットの各ファイルfに対して3.2.4節と同様$Recall(f)$を求め，また下記の式で適合率$Precision(f)$を求めた．

$$Precision(f) = \frac{\hat{C}(f) \text{ 中の○の数}}{|\hat{C}(f)|}$$

これらから再現率の平均と適合率の平均を求めた結果は以下の通りである．

	被験者A	被験者B
再現率の平均値	81.0 %	82.6 %
適合率の平均値	66.2 %	45.0 %

4.3 考　　察

関連度のパラメータについては，共起時間の累計だけを考える(1,0,0,0)の組み合わせや，回数だけを考慮する(0,1,0,0)の組み合わせよりも，回数，使用開始パターンの類似，共起の間隔を被験者に合わせて組み合わせの方が，より有効な関連度をより計算出来ることがわかった．

パラメータがユーザーごとに変化する理由は，ユーザのファイル使用の違いによるものだと考えられる．例えば一度開いたファイルを閉じずに他の作業をすることが多いユーザは，関係ない二つのファイルの共起時間が多くなると考えられる．そのため共起時間の累計への重みは減ると予想される．

適切なパラメータをユーザごとに算出した結果，関連度の評価実験において被験者A，B共に再現率の平均が80 %以上であった．この結果より，提案する検索手法を実現する際に，この関連度を用いることが可能であると考える．

ログのクリーニングに関しては評価実験は行っていないが，具体的な事例として，(A)(B)の問題を修正したことによって本来関連の無い二つのファイルの共起が除去されたことを確認している．また，(C)の問題を修正したことにより，ある論文のTexファイルのログが修正され，論文の図の画像ファイルや，グラフの基となっている実験データファイルと論文のTexファイルの共起が抽出できた．このような事例が観測されたため，ログのクリーニングは有効であったと考える．

5. 関連研究

本研究では既存のディレクトリ階層構造の問題点の解決を目的として，ユーザの動作履歴を基にファイル間の関連度を算出し，ファイル検索へ利用している．そこで，関連研究を「階層構造の問題点への解決」，「ユーザの動作履歴利用」，「データ間関連度」の3つの観点から議論する．

5.1 階層構造の問題点への解決

既存のファイルシステムが提供するディレクトリ階層では，増え続けるファイル群に対して対応しきれない．Griffordらの Semantic File Systems [7] では，各ファイルはディレクトリに所属しない代わりに複数の属性を持ち，この属性を用いてファイルを管理するファイルシステムである．これにより，ファイルシステムの抽象化が可能になり情報共有等に役立つと報告している．また，石川らの研究 [8] では属性の表現にRDF [9] を用いることにより，セマンティックウェブ関連の技術の応用を容易にしている．また，ファイルの参照関係などをRDFで表現することにより，ファイルの整理や一貫性の管理を実現している．

既存の階層型分類ではなくファセット分類によるデータ利用の研究も多くなされている．Yeeらの研究 [10] では大量の画像データに対してファセット分類を用いたインターフェースを提供し，被験者実験により従来の階層型分類よりもファセット分類を用いたインターフェースが，ユーザに好まれると報告している．

これらの研究は階層構造の問題点への解決という点では本研究と一致するが，キーワードに関連するものを探すことはしていないので，その点で本研究と異なる．

5.2 ユーザの動作履歴利用

Memory-Retriever [11] はオペレーティングシステムのイベントログと，ブラウザなどのアプリケーションに組み込んだプ

ラグインから動作履歴を抽出し，その動作履歴を基にウェブ閲覧体験時の記憶を想起させるツールである．俺デスク [12] は Memory-Retriever と同様にイベントログと組み込んだプラグインから動作履歴を抽出し，それを基にウェブページやファイル (以降まとめてデータと呼ぶ) の着目度，データ間の関連度を算出する．そして，関連度を基にしたデータ名による関連検索と，時系列のビューアを提供するシステムである．

これらの研究では，動作履歴を抽出するためにオペレーティングシステムのイベントログを収集するアプリケーションと，特定のアプリケーションごとのプラグインをインストールする必要があるが，本研究ではアクセスログを利用するのでユーザの環境に何も手を加えることなくシステムを実現出来る．

また，俺デスクは主目的が情報想起を支援することではあるが，データ検索も提供している．しかし，その検索はデータ名を入力し関連しているデータを提示するものであって，本研究で提案するキーワードによるファイル検索とは異なる．関連度の算出においても，俺デスクではデータアクセス開始時間のみを考慮しているのに対し，本研究では共起時間の累計，共起回数，間隔等を考慮しており，関連ファイル間の識別能力に違いがある．

5.3 データ間関連度

データの関連度を算出する既存の研究の多くは，データ内容を解析しデータ間の関連度を算出する．テキストデータや，画像データ，音声データ，動画データなどのそれぞれのデータに対して，類似度を算出するさまざまな研究がなされているが，本研究ではデータの使用時間のみを基に関連度を算出するので，データの種類にとらわれることはない．

増田らの近傍検索システム [13] では，(1) ファイルを含むディレクトリ構造，(2) ファイルの更新時間，(3) ファイルの内容の３つの情報を基に，ファイル間の関連度を算出し，ファイル間の連想アクセスシステムを提案している．しかし，(3) のファイル内容の関連度計算は文書ファイルに限っているので，それ以外のファイルに対しては (1) ディレクトリ構造と (2) 更新時間の情報からのみの算出になる．これに対して，本研究では文書ファイル以外の画像ファイル，データファイル等のファイルに対しても等しく関連度を算出出来る．

6. まとめと今後の課題

既存の全文検索ではキーワードを含んでいないファイルは，そのキーワードと関連があっても検索できないという問題点があった．その問題点に対し，本稿では，キーワードで検出されたファイルと関連度の高いファイルを提示することで，全文検索できないファイルを含んだ検索を可能とするシステムを提案した．

また，同時に用いられるファイル間の関係に着目した関連度を，ファイルサーバへのアクセスログから抽出する手法を提案，実装し，その関連度を用いてあるファイルと関連度の高いファイル群を提示した所，そのファイル群の正解 (ユーザが関連があると判断した) の再現率が 80 %以上であった．この結果より，提案する検索手法を実現する際，この関連度を用いることが可能であると考える．

今後の課題としては以下の 3 点が挙げられる．

- 検索手法の実装を最後まで行い，その評価を行う．
- より多くのユーザを被験者の対象として，手法の評価を充実させる．
- ファイルの使用時間の情報のみでなく，ファイルのコピーや移動をアクセスログから抽出し，その情報を元に関連度の算出をより効果的にする手法を考案する．

謝　辞

本研究の一部は，文部科学省科学研究費補助金特定領域研究 (19024028)，東京工業大学 21 世紀 COE プログラム「大規模知識資源の体系化と活用基盤構築」および独立行政法人科学技術振興機構戦略的創造研究推進事業 CREST の助成により行なわれた．

文　献

[1] Nitin Agrawal, William J. Bolosky, John R. Douceur, Jacob R. Lorch: A Five-Year Study of File-System Metadata, 5th USENIX Conference on File and Storage Technologies(FAST'07), 2007.

[2] About Google Desktop, http://desktop.google.com/about.html

[3] Windows Desktop Search, http://www.microsoft.com/windows/desktopsearch/default.mspx

[4] Spotlight, http://www.apple.com/jp/macosx/features/spotlight/

[5] 全文検索システム Namazu, http://www.namazu.org/

[6] Google Image Search, http://images.google.com/

[7] D.K.Gifford, P.Jouvelot,M.A.Sheldoon, J.W.O'Toole,Jr: Semantic File Systems, 13th ACM Symposium on Operating Systems Principles, 1991.

[8] 石川憲一, 森嶋厚行, 田島敬史: 大規模ドキュメント空間管理のための意味ファイルシステムの構築, 信学技報 DE2006-115 pp.139-144,2006.

[9] W3C.Resource Description Framework(RDF),http://www.w3g.org/RDF

[10] Ka-Ping Yee,Kirsten Swearingen,Kevin Li,Marti Hearst: Faceted Metadata for Image Search and Browsing, In Proc. of CHI ' 03, 401-408, 2003.

[11] 森田哲之, 日高哲雄, 田中明通, 加藤泰久: 記憶想起支援ツール『Memory-Retriever』, INTERACTION 2007, 2007.

[12] 大澤亮, 高汐一紀, 徳田英幸: 俺デスク:ユーザ操作履歴に基づく情報想起支援ツール, 情報処理学会第 47 回プログラミング・シンポジウム, 2005.

[13] 増井俊之, 塚田浩二, 高林哲: 近傍関係にもとづく情報検索システム, 11th Workshop on InteractiveSystems and Software (WISS2003), 2003.