

## ユーザタスク情報に基づくファイル関連管理手法

定 免 睦 昌<sup>†1</sup> 國 島 丈 生<sup>†2</sup> 横 田 一 正<sup>†2</sup>

PCでユーザが扱うファイル数は極めて増加しており、フォルダによる分類やデスクトップ検索など、ユーザが目的のファイルを探す手法の重要性は増している。しかし、これらの手法ではユーザが実際に行っていた作業を反映した検索は充分に行えない。我々は、先行研究では考慮されていなかった複数のタスクが同時に行われるという状況を考慮して、ユーザのデスクトップ操作からファイル間の関連度を算出する方法、階層的クラスタリングによって関連の深いファイル集合としてユーザのタスクを抽出する方法などをこれまで提案し、いくつかの評価実験を行ってきた。本稿ではこれまでの提案や評価実験について報告し、ユーザが過去に行ったタスクに基づくファイル関連管理手法について考察する。

### A method for managing relationship between files based on history of user tasks

YOSHIAKI JOMEN,<sup>†1</sup> KUNISHIMA TAKEO<sup>†2</sup>  
and YOKOTA KAZUMASA<sup>†2</sup>

The number of the files which PC user uses on desktop is increasing and the importance of the method of searching the target file is increasing. However, by these methods, the search reflecting the task which PC user was actually doing cannot fully be performed. We take into consideration the situation where two or more tasks which were not taken into consideration are performed simultaneously, in precedence research, and proposed until now the method of computing the degree of association between a file from a user's desktop operation, how extracts a user's task as a deep file set of relation by hierarchical clustering, etc., and have conducted some evaluation experiments. This paper reports a proposal so far and evaluation experiment, and studies the method for managing relationship between files based on history of user tasks.

### 1. はじめに

近年、ユーザがPCで扱うファイル数は極めて増加しており、同一作業で使用したファイルは注意深く管理を行わないと他の作業ファイルに埋もれてしまい、発見することが難しくなっている。このような背景から、ファイルのあるカテゴリーで分類するフォルダ管理やフォルダ構造に依存せずに目的とするファイルを発見できるデスクトップ検索などが多く使われている。しかし、フォルダによるファイル分類では複数のフォルダに属するファイルの管理方法が難しく、デスクトップ検索では目的のファイルを発見することはできても、作業と一緒に使用したファイルの発見をすることはできない。このような動機の下に同一作業で使用したファイルを効率的に発見する手法の需要が高まっており、いくつかの研究が行われている。我々は、先行研究で考慮されていなかった複数の作業（タスク）が同時に行われるという状況を考慮して、ユーザのデスクトップ操作からファイル間の関連度を算出する方法を提案し、階層的クラスタリングによって関連の深いファイル集合としてユーザのタスクを抽出する方法を提案した<sup>1)</sup>。また、ファイルのクラスタリング精度の向上を図るためにタスク進行におけるファイルの主従関係に着目したクラスタリング手法の提案<sup>1)</sup>や、ファイルの挿入操作を考慮に入れてファイル間関連度を拡張する方法の提案と評価実験<sup>3)</sup>、タスク間で共通して使用するファイルの発見手法の提案<sup>2)</sup>と評価実験<sup>4)</sup>を行ってきた。本稿ではこれまでの提案や評価実験について報告し、ユーザのタスク情報に基づいたファイル関連管理手法について考察する。

### 2. 関連研究

文献<sup>5)6)7)</sup>では、ファイルアクセスログからファイルのオープン・クローズ時間を抽出し、ファイル間関連度を算出する。2つのファイルを同時にアクセスしていた場合、その2つは同一の作業に利用していた可能性が高いと報告している。共起時間や、ファイルの使用時刻の近さから、ファイル間関連度を算出する点では本研究と一致するが、複数のタスクを同時に平行して行う場合については考慮されておらず、ファイルサーバのアクセスログを用いてファイルの使用履歴を抽出して点で本研究とは異なる。文献<sup>8)</sup>では、デスクトップ上の

†1 岡山県立大学大学院  
Graduate School, Okayama Prefectural University

†2 岡山県立大学  
Okayama Prefectural University

ファイルにアクセスした時刻の差やファイルにアクセスした順序、一定時間内にアクセスしたファイル数の密度等によって、ファイル間の関連度を算出する。デスクトップ上のファイルを対象としてファイル間関連度を算出する点では本研究と一致するが、具体的な関連度の要素を抽出する方法について述べられておらず、複数のタスクを同時に平行して行う場合についても考慮されていない点で本研究とは異なる。文献<sup>9)</sup>では、OSのイベントとプラグインを組み込んだアプリケーションから動作履歴を抽出し、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、関連度を算出する。そして、関連度を基にしたデータによる関連検索と、着目度を基にした参照していたデータを時系列に表示するビューを提供するシステム、俺デスクを実装している。OSのイベントを用いて、動作履歴を抽出する点では本研究と一致するが、複数のタスクを同時に平行して行う場合について考慮されていない点で、本研究とは異なる。

### 3. クラスタリング手法

クラスタリングは、大きな集合を共通の特徴を持つ部分集合へと分ける手法のことである。要素そのものに特徴があり、特徴が似ている要素を切り分ける手法と、要素自体には特徴がなく、要素間の関係のみを用いて関係の高い集合を発見する手法がある。我々の研究では、ファイル間の関係を用いるため前者の手法は用いることができず、後者のクラスタリング手法の1つである階層的クラスタリングを用いた。

#### 3.1 クラスタとタスク

ユーザの操作に基づいて、ファイル間関連度やクラスタ間の距離を決定しているので、タスク毎に使用されている可能性の高いファイルが関連付いている。デンドログラムを用いて、ある閾値の距離でクラスタを切断することにより、タスク毎に関係のある集合を得ることができる。

### 4. デスクトップ操作に基づくタスクの抽出

以下では、ファイル  $x$  とファイル  $y$  の間における関連度を算出し、ファイル間の距離の決定する方法について述べる。

### 4.1 概要

ファイル間関連度の尺度からファイル間関連度を算出し、ファイル間の距離を決定する。次に、ファイル間の距離に基づいて階層的クラスタリングを行い、クラスタリング結果に基づいてデンドログラムを用いてクラスタを切り分け、分割されたクラスタをタスクとみなすことでユーザタスクの抽出を行う。

#### 4.2 ファイル間の関連度

ファイル  $x$  とファイル  $y$  が同一タスクに属する可能性を数値で表したものである。値が高いほどファイル  $x$  とファイル  $y$  が同一タスクで使用された可能性が高いものとする。4.2.4節に示すように、以下に述べる共起時間、ウインドウフォーカスの遷移回数、ファイルの挿入という3つの尺度から決定される。

##### 4.2.1 共起時間 $T$

2つのファイルの使用時間が重複していた合計時間に着目した尺度である。また、ユーザが退席した場合を考慮して、キーボードとマウスの入力がない時間が閾値  $t_{inactive}^{*1}$  より大きかった場合、次にキーボードかマウスの入力があるまでの時間を共起時間から引いたものを、実際の共起時間と考えるものとする。合計時間が大きいほど、2つのファイル間において関連性が高いものとする。例えば、ファイル  $x$  とファイル  $y$  を同時に使用していた時間が  $n$  回あり、それぞれの時間が  $t_i (1 \leq i \leq n)$  であるとき、ファイル  $x$  とファイル  $y$  の共起時間  $T(x, y)$  は以下の式で表される。

$$T(x, y) = \sum_{i=1}^n t_i \quad (1)$$

##### 4.2.2 ウインドウフォーカスの遷移回数 $F$

2つのファイル間でウインドウフォーカスの遷移回数に着目した尺度である。操作の誤りでファイルを参照してしまう可能性を考慮して、ウインドウアクティブ時間の要素  $t_j$  が閾値  $t_{attend}^{*2}$  以下のものは考えない。合計回数が多いほど、2つのファイル間において関連性が高いものとする。ファイル  $x$  のウインドウからファイル  $y$  のウインドウ、またはファイル  $y$  のウインドウからファイル  $x$  のウインドウにフォーカスが遷移した場合を考える。このとき、フォーカスの移動後ウインドウがアクティブになった回数が  $j$  回あるとき、ファイル  $x$  とファイル  $y$  のフォーカス遷移回数  $F(x, y)$  は以下の式で表される。

\*1 閾値の決め方は検討中だが、被験者実験では 300sec とした。

\*2 閾値の決め方は検討中だが、被験者実験では 1sec とした。

$$\begin{cases} n_k = 0 & (0 \leq t_j \leq t_{attend}) \\ n_k = 1 & (t_{attend} < t_j) \end{cases}$$

$$F(x, y) = \sum_{k=1}^l n_k \quad (2)$$

#### 4.2.3 ファイルの挿入 $I$

2つのファイル間でファイルの挿入に着目した尺度である。ファイル  $x$  にファイル  $y$ , またはファイル  $y$  にファイル  $x$  の挿入が行われた場合を考える。ファイルの挿入が  $n$  回行われたとき、ファイル  $x$  とファイル  $y$  のファイルの挿入  $I(x, y)$  は以下の式で表される。

$$\begin{cases} I(x, y) = 0 & (n = 0) \\ I(x, y) = 1 & (n \geq 1) \end{cases} \quad (3)$$

#### 4.2.4 ファイル間の距離の定義

式 (1), (2) をすべて以下のように区間  $[0, 1]$  に正規化を行う。

$$R_T(x, y) = \frac{T(x, y)}{\max_{a, b \in AllFiles} (T(a, b))} \quad (4)$$

$$R_F(x, y) = \frac{F(x, y)}{\max_{a, b \in AllFiles} (F(a, b))} \quad (5)$$

式 (3), (4), (5) を用いてファイル  $x$  とファイル  $y$  の間における関連度を以下のように定義する。

$$R(x, y) = I(x, y) + R_T(x, y) + R_F(x, y) \quad (6)$$

式 (6) を用いて、ファイル  $x$  とファイル  $y$  の距離を以下のように定義する。

$$D_R(x, y) = 1 - \frac{I(x, y) + R_T(x, y) + R_F(x, y)}{3} \quad (7)$$

#### 4.3 ファイル間の距離に基づくタスクの抽出

ファイル間の距離に基づき 4 章で述べたクラスタリング手法を用いてタスクを抽出する。閾値の取り方は検討中だが、现阶段では実験の結果が一番良かったものとする。

## 5. タスク抽出の拡張

### 5.1 アクティブ時間を考慮したタスク抽出精度の向上

ユーザが多くのタスクを行うなかで、ファイルを参照しながら他のファイルを作成することはよくある。例えば、PDF ファイルを参照してスライド作成を行うタスクの場合、タスクの中心となるファイルはスライドファイル、参照されるファイルは PDF ファイルとなる。このとき、ファイル作成を行うファイルと参照されるファイル間ではファイルの主従関係が成り立つ。ファイルの主従関係に基づいてクラスタリングを行いタスクを抽出することで、複数タスク環境でも同一タスクに属するファイルの精度が高くなると考えられる。以下では、このファイル間の主従関係に着目し、ウインドウのアクティブ時間によってファイルの着目度を抽出してそれに基づいてクラスタリングを行い、クラスタリング結果を用いてタスクの抽出を行う方法について述べる。

#### 5.1.1 ウインドウアクティブ時間 $A$

ファイルが起動している間にそのファイルを参照した合計時間に着目した尺度である。操作の誤りでファイルを参照してしまう可能性を考慮して、参照時間が閾値  $t_{attend}^{*2}$  以下のものは考えない。また、ユーザが退席した場合を考慮して、キーボードとマウスの入力がない時間が閾値  $t_{inactive}^{*1}$  より大きかった場合、次にキーボードかマウスの入力があるまでの時間をウインドウのアクティブ時間から引いたものを、実際のアクティブ時間と考えるものとする。合計時間が大きいほどそのファイルを軸に作業を行っていた可能性が高いものとする。例えば、ファイル  $x$  を開いているウインドウが  $m$  回アクティブになったとき、アクティブになった時間をそれぞれ  $t_j (1 \leq j \leq m)$  とすると、ファイル  $x$  のウインドウアクティブ時間  $A(x)$  は次のように表される。

$$A(x) = \sum_{j=1}^m t_j \quad (t_{attend} < t_j) \quad (8)$$

#### 5.1.2 アクティブ時間を考慮したタスクの抽出

クラスタ内で、ウインドウのアクティブ時間が一番大きい要素を選択し、選択した要素間の距離が最小値であるクラスタを統合する手法である。ウインドウのアクティブ時間は、ファイルに対する着目度を表しており、着目度が大きいほどそのファイルを軸に作業を行っていた可能性が高いので、クラスタを統合する際に有用だといえる。クラスタ  $C$  に属するファイル  $x$  の中で、アクティブ時間  $A(x)$  が最も大きいファイルを選択し、選択したファ

ル間の距離  $D(C_i, C_j) (i \neq j)$  が一番短いクラスタを統合する方法でクラスタリングを行う。選択するファイル  $S(C)$  と、選択したファイル間の距離の最小値  $D(C_i, C_j)$  は以下の式で表される。

$$S(C) = \{x | \max_{x \in C} (A(x))\} \quad (9)$$

$$D(C_i, C_j) = \min(D(S(C_i), S(C_j))) \quad (10)$$

## 5.2 共通ファイルを持つタスクの抽出

階層的クラスタリングではクラスタを排他的に統合するため、共通ファイルを含むことができない問題があった。以下では、タスク間の共通ファイルを発見する手法について述べる。

### 5.2.1 タスク間の共通ファイル発見手法

全てのファイル集合  $F$ 、ファイル  $a$  を含むクラスタ  $C_i$ 、ファイル  $x$  を含むクラスタ  $C_j$  とする ( $i \neq j, a \neq x$ )。このとき、 $D_R(a, x)$  が一定値以下である全てのファイル  $x (x \in E, E = F - C_i)$  を、ファイル  $a$  と関連の深いファイルと見なす。その尺度の一例として、

- $D_R(a, x)$  が  $D_R(a, x)$  の相加平均よりも大きい
- $D_R(a, x)$  が  $D_R(a, x)$  の相加平均よりもあまり離れていない

のいずれかを満たす全ての  $x$  を  $a$  と関連が深いと見なす。相加平均よりもあまり離れていないファイルを調べるために、 $D_R(a, x)$  の標準偏差を求める。 $D_R(a, x)$  が相加平均から標準偏差のある一定値以内であれば、あまり離れていないものと見なす。つまり、ファイル  $x$  をタスク(クラスタ)  $C_i$  と  $C_j$  の共通ファイルとする。ファイル  $a$  とファイル  $x$  のファイル間関連度の相加平均  $H(a)$  と標準偏差  $S(a)$  を以下の式で表す。

$$H(a) = \frac{\sum_{x \in E} D_R(a, x)}{|E|} \quad (11)$$

$$S(a) = \sqrt{\frac{\sum_{x \in E} (D_R(a, x) - H(a))^2}{|E|}} \quad (12)$$

### 5.2.2 共通ファイルを含むタスクの抽出

タスク間の共通ファイル発見手法でファイル  $a$  と関連が深いとみなされたファイル  $x$  をクラスタ  $C_i$  に組み込むことで、共通ファイルを含むタスクを抽出する。

## 6. 実装と評価実験

### 6.1 実装

ファイル間の関連度の尺度となる要素は、常駐プログラムでファイルのアクセスやユーザの GUI 操作を監視させ、OS のイベントをフックすることで抽出する。具体的な監視対象は、ウインドウのフォーカス、キーボード、マウス、ファイルアクセスである。マウスとキーボードの監視は主に、ユーザの退席を監視する目的で行う。なお、OS のイベント処理は OS ごとに異なるため、本研究では Windows を対象として実装と評価を行った。

#### 6.1.1 Win32API

Win32API とは、Microsoft Windows の API のなかで、特に 32 ビットプロセッサで動作する OS で利用できる API の名称である。Win32API では、.NetFramework で対応できないウインドウのアクティブ情報や、キーボードの入力情報等を取得するために用いることにした。定期的にタイマーでウインドウのアクティブ情報を取得することで、ウインドウ固有の情報を得ることができる。例えば、ウインドウがアクティブ状態になっている時間や、ウインドウがアクティブ状態になる遷移情報、ウインドウで開いているファイルの名前を得ることが可能になる。これら抽出した要素をある一定の間隔でデータベースに格納し、必要に応じてデータベースから要素を呼び出してファイル間の距離を算出する。

## 7. 評価実験

これまでに提案したファイル挿入操作を考慮したタスク抽出手法と、タスク間の共通ファイルを想定したタスク抽出手法の有効性を確かめるために評価実験を行った<sup>3)4)</sup>。本章では、実験内容と結果の考察について述べる。

### 7.1 実験環境

被験者に 1 時間でスライドの作成と、レポートの作成の 2 つのタスクを同時並行して進めてもらい、タスク間の共通ファイルを含めない実験(実験 1)と、タスク間の共通ファイルを含める実験(実験 2)のファイルのアクセスやユーザの操作ログを採取した。共起時間とウインドウフォーカスの移動回数の監視は自作アプリケーション、ファイルの挿入の監視は Process Monitor<sup>10)</sup> でそれぞれ行った。Process Monitor はプロセスが行った処理(ファイルシステム、レジストリ、プロセスおよびスレッドの活動)をリアルタイムで表示するアプリケーションである。

表 1 実験 1 のイメージファイル数

タスク	関連あり	関連なし
スライド作成	19	30
レポート作成	22	30

表 4 実験 2 のイメージファイル数

タスク	関連あり	関連なし
スライド作成	5	42
レポート作成	12	30

表 2 実験 1 の正解セット

	スライド作成	レポート作成
被験者 A	2	2
被験者 B	11	6
被験者 C	10	7
被験者 D	5	5
被験者 E	6	5
被験者 F	7	6

表 3 実験 1 のファイル間関連数

	ファイル間関連数
被験者 A	4
被験者 B	33
被験者 C	53
被験者 D	21
被験者 E	20
被験者 F	51

表 5 実験 2 の正解セット

	スライド作成	レポート作成
被験者 A	6	6
被験者 B	3	3
被験者 C	4	5
被験者 D	4	4
被験者 E	3	3

表 6 実験 2 のファイル間関連数

	ファイル間関連数
被験者 A	57
被験者 B	5
被験者 C	57
被験者 D	45
被験者 E	55

## 7.2 実験方法

タスクに必要なファイルは一つのフォルダに予めまとめておき、被験者はその中のファイルを使用してタスクを進める。1 時間でタスクが完成しない場合でも途中で作業を終了してもらう。スライド作成とレポート作成に必要な論文は、異なるものを 1 つずつとタスク間で共通して使用するファイルを 1 つ用意した。被験者にはサムネイル機能を使わずにイメージファイルを一枚ずつ Windows フォト ビューアーで確認して目的のファイルを探してもらう。また、タスクと関連のないイメージファイルと論文に挿入されていた関連のあるイメージファイルを名前から内容が予想できないように、名前をランダムな番号に置き換えた。被験者がタスクを進めるなかで使用したファイル（イメージファイルは挿入したもの）を正解セットとする。

## 7.3 実験 1

### 7.3.1 実験結果

実験に用意したイメージファイル数を表 1、被験者 6 人の正解セットを表 2、被験者がタスクを進めるなかで関連付いたファイルの数を表 3 に示す。3.1 節で述べたデンドログラム

で用いる閾値が 0.914 のとき、全ての被験者について 2 つのタスクの再現率は 100%となった。適合率は被験者 C を除いて 100%であったが、被験者 C のスライド作成のタスクでは 57.1%、レポート作成のタスクでは 28.5%となった。

### 7.3.2 考察

被験者 C の実験結果が悪かった原因として、タスクに関係のないファイル間の遷移が他の被験者よりも多く行われていたため、同一タスクに関連していないファイルが存在していることが考えられる。

## 7.4 実験 2

### 7.5 実験結果

実験に用意したイメージファイル数を表 4、被験者 5 名の正解セットを表 5、被験者がタスクを進めるなかで関連付いたファイルの数を表 6 に示す。3.1 節で述べたデンドログラムで用いる閾値が 0.93 のとき、5 名の 2 つのタスクにおける共通ファイルを発見しない手法と発見する手法の再現率の平均値、適合率の平均値、F 値の平均値をそれぞれ表 7 と表 8 に示す。F 値の平均値ではスライド作成の適合率の平均値にあまり違いは出なかったが、レポート作成では高くなる結果となった。

表 7 共通ファイルを発見しない手法

	スライド作成	レポート作成
再現率の平均値	0.683	0.67
適合率の平均値	0.9	0.95
F 値の平均値	0.776	0.785

表 8 タスク間の共通ファイルを発見した手法

	スライド作成	レポート作成
再現率の平均値	0.933	1
適合率の平均値	0.658	0.771
F 値の平均値	0.771	0.855

### 7.5.1 考察

表 7 と表 8 から、タスク間の共通ファイルを発見する手法では、タスク間の共通ファイルを発見しない手法に比べて再現率が高くなった。しかし、タスク間の共通ファイルのなかに関連のないファイルを含んでしまったため、適合率が低くなった。関連のないファイルが含まれた原因の一例として、ユーザが共通ファイルを参照してレポートの作成を行った後に、レポートファイルを参照しながらスライド作成を行っていたことが考えられる。また、複数タスク環境を意図的に作り出すために、イメージファイル以外の使用するファイルは実験開始時に全て開くように指示した。そのため、ウインドウフォーカスの遷移がファイル間関連度に大きく影響したことが考えられる。

## 8. タスク情報に基づくファイル関連管理

本研究の目標は、ユーザが過去に行ったタスクの履歴に基づいて、あるファイルと同一タスクで使用したファイルを発見することである。我々がこれまでに行ってきた研究は、同一タスクで使用したファイルを発見することが目的であり、それらのファイルを管理する方法まで十分に考察できていなかった。そこで、ユーザからあるファイルに関連したファイルの検索がしたいという要求が発生した場合に、これまでに提案してきた手法を用いて関連ファイルの導出を行う。以下では、ユーザのファイル検索の要求内容とユーザが行ったタスクの

履歴から関連のあるファイルを導出する手法について述べる。

### 8.1 タスク情報の保存

あるファイルを使用した一連のタスクを抽出するには、ユーザが行っていたタスクの履歴を保存しておく必要がある。我々がこれまでに行ってきた研究ではタスクの終わりを検出できていなかったため、タスクの終わりを監視アプリケーションに検出させる。ユーザがタスクを終える一つの目安として、PC をシャットダウンさせるときが挙げられる。そこで、PC がシャットダウンする直前に 4 章で述べた方法でユーザが行ったタスクの抽出を行い、タスク情報としてデータベースに格納することにした。タスク情報は、データベースに格納した日時、ファイル名、ファイルのアクティブ開始・終了日時、ファイルの起動・終了日時、遷移先ファイル名、遷移先ファイルに遷移した回数である。

### 8.2 タスク情報に基づいた関連ファイルの発見

ユーザからの検索要求として「あるファイルと同一タスクで使用していたファイル情報（ファイル名など）を知りたい」といったものが挙げられる。さらに細かい要求として以下のようなものがあると考えられる。

- 日付を指定して、その日に同一タスクで使用したファイルを検索
- 期間を指定してその期間内で同一タスクで使用したファイルを検索
- 期間を指定せずに同一タスクで使用したファイルを検索

これらの検索要求からタスク情報を用いて同一タスクに関連のあるファイルを求める方法は次の通りである。データベースに格納されている全てのタスク集合  $T$ 、ファイル  $a$  を含むファイル集合  $T_i (T_i \in T)$ 、 $T_i$  に属するファイル  $x (x \neq a)$  とする。指定された日付もしくは期間内にデータベース格納されたファイル集合を  $T_j (T_j \in T)$  とする。検索の際に日付もしくは期間が指定されていた場合、ファイル集合  $T_i (i = j)$  に対してファイル  $x$  を抽出したファイル集合を  $F_1$  とし、ファイル  $a$  に関連のあるファイルを得る。また、検索期間が指定されていなかった場合、ファイル集合  $T_i$  に対してファイル  $x$  を抽出したファイル集合を  $F_2$  とし、ファイル  $a$  に関連のあるファイルを得る。

### 8.3 同一タスクで使用したファイル発見の拡張

タスク情報から同一タスクで使用したファイルの発見はできるが、同一タスクであっても検索のもととなるファイルを含んでいないタスクからは関連ファイルを発見することはできない。そこで、タスク情報のアクティブ時間を用いて異なるタスク間で同一タスクを発見し、同一タスクで使用したファイル発見の拡張を行う。タスクのなかでアクティブ時間が一番長いファイルをタスクの主となるファイルとみなし、異なるタスクで主となるファイルが同じ

タスクのものを同一タスクである可能性が高いと考える。アクティブ時間を用いて同一タスクで使用したファイル発見の拡張をする方法は次の通りである。データベースに格納されている全てのタスク集合  $T$ 、ファイル  $a$  を含むファイル集合  $T_i (T_i \in T)$ 、 $T_i$  に属するファイル  $x (x \neq a)$ 、ファイル集合  $T_i$  と同一タスクである可能性の高いタスク  $T_k (k \neq i, T_k \in T)$ 、 $T_k$  に属するファイル  $y (y \neq x)$  とする。指定された日付もしくは期間内にデータベース格納されたファイル集合を  $T_j (T_j \in T)$  とする。検索の際に日付もしくは期間が指定されていた場合、ファイル集合  $T_i (i = j)$  と、ファイル集合  $T_k (k = j, j \neq i)$  に対してファイル  $x$  とファイル  $y$  を抽出した集合を  $F_3$  とし、ファイル  $a$  に関連のあるファイルを得る。また、検索期間が指定されていなかった場合、ファイル集合  $T_i$  と  $T_k$  に対してファイル  $x$  とファイル  $y$  を抽出したファイル集合を  $F_4$  とし、ファイル  $a$  に関連のあるファイルを得る。

## 9. まとめと今後の課題

本稿では、先行研究では考慮されていなかった複数のタスクが同時に行われるという状況を考慮して、これまでの提案や評価実験について報告し、ユーザタスク情報に基づいたファイル関連管理手法について考察を行った。今後の課題として、タスク情報に基づいたファイル関連管理手法の実装と長期的な被験者実験を行い、評価や考察を行う必要がある。また、検索結果をユーザにどのように提示するかについても考察が必要である。

現段階では、デスクトップ上の保管されているファイルに限定してファイルに関連付けている。近年ではインターネット上に存在するデータ量が爆発的に増加しており、Web を利用して情報を取得しタスクを進めることが多く行われている。このような背景から、Web ページとデスクトップ上のファイルの関連性を抽出する研究が行われている<sup>11)</sup>。しかし、Web ページは流動的であり URL のリンク切れや Web ページの更新といった特性を持っているため、そのときに利用した情報が URL 先の Web ページに残っていると限らない。今後、我々は Web ページの情報保持も考慮に入れ、同一タスクで使用したデスクトップ上ファイルと Web ページの双方を関連付けることを考えている。

## 参 考 文 献

- 1) 定免睦昌, 國島丈生, 横田一正, 複数タスク環境におけるユーザ操作に基づくファイル間関連度の導出, 第 2 回データ工学と情報マネジメントに関するフォーラム (DEIM2010), D1-3, pp.1-6, 2010.
- 2) 定免睦昌, 國島丈生, 横田一正, ファイル間関連度を用いたタスク間の共通ファイル発見手法, 平成 22 年度 (第 61 回) 電気・情報関連学会中国支部連合大会, 情報処理-(3),

- 26-17, pp.382-383, 2010.
- 3) 定免睦昌, 國島丈生, 横田一正, ユーザのファイル挿入操作を考慮したタスク抽出手法の提案とその評価, 平成 23 年度 (第 62 回) 電気・情報関連学会中国支部連合大会, 情報処理-(5), 26-29, pp.53-54, 2011.
- 4) 定免睦昌, 國島丈生, 横田一正, タスク間の共通ファイルを想定したタスク抽出手法とその評価, 第 13 回 IEEE 広島支部学生シンポジウム, Nov.11-12, 2011.
- 5) 小田切健一, 渡辺陽介, 横田治夫, アクセス履歴を用いたユーザの作業に対応する仮想ディレクトリの生成, 第 8 回日本データベース学会年次大会 (DEIM2009), 電子情報通信学会, DEIM2009 論文集, 2009.3.
- 6) 渡部徹太郎, 小林隆志, 横田治夫, 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 (情報抽出, 夏のデータベースワークショップ 2007(データ工学, 一般)). 電子情報通信学会, 電子情報通信学会 信学技報, Vol. 107, No. 131, pp. 503-508, 2007.7.
- 7) 渡部徹太郎, 小林隆志, 横田治夫, キーワード非含有ファイルを検索可能とするファイル間関連度を用いた検索手法の評価, IEICE, データ工学ワークショップ 2008 論文集, 2008.3.
- 8) 井ノ口伸人, 吉川正俊, アクセス履歴を考慮したファイル間の関連度を用いたデスクトップ検索 (履歴応用, 夏のデータベースワークショップ dbws2006). 電子情報通信学会, 電子情報通信学会 信学技報, Vol. 106, No. 148, DE2006-41, pp. 115-120, 2006.7.
- 9) 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, 2006. 第 47 回プログラミング・シンポジウム報告集.
- 10) Process Monitor, <http://technet.microsoft.com/ja-jp/sysinternals/bb896645>.
- 11) 宋強, 渡辺陽介, 横田治夫, ファイルと Web ページの共起頻度に着目した関連性抽出手法の評価, DEIM Forum 2011, 2011.3.