

出自検索：ユーザが経験した ファイルの移動と編集に基づく検索手法

栗山 拓[†], 重森 晴樹[†], 倉本 到[†],
辻野 嘉宏[†], 水口 充^{††}

現在、PC ユーザは多くのファイルを抱えている。これらのファイルにアクセスするためにユーザはファイルを分類したり検索を行ったりする。ファイルへのアクセスに用いられるキーはファイルの格納場所と名前や内容の一部などのファイル自体の情報であり、これらを覚えていなければファイルへのアクセスが難しい。一方、ユーザはファイルをどのように手に入れたかという「ファイルの出自」を自身の経験として覚えていることが多い。本研究では、この「ファイルの出自」に着目し、出自を基にしたファイル検索手法を提案し評価を行った。その結果、提案手法では従来のエクスプローラでの探索より、容易にファイルにアクセスできることが分かった。

Roots-Based Retrieval : File Retrieval Method Based on Its Moving and Editing that User Experienced

Taku Kuriyama[†], Haruki Shigemori[†], Itaru Kuramoto[†],
Yoshihiro Tsujino[†], Mitsuru Minakuchi^{††}

Today, PC users have many files. In order to access some of these files, the users retrieve them. In such a case the keys, for example, storage places, the file names, and some words in the file contents, are used to retrieve. However, if users do not remember them, it is difficult to access desired files. In order to solve the problem, we proposed a file retrieval method based on "roots of the file", how to get the file, as his/her own experience. As the result of an experimental evaluation, users can retrieve the files more easily by the proposed method than by conventional tools such as explorer, when they remember the roots of the file.

1. はじめに

現在のPC使用環境において、日々ユーザは多様なアプリケーションを使用し、多量のファイルを作成する。そして、ユーザはそれらのファイルにアクセスするためにファイルをフォルダ階層構造に分類したり、ファイル検索システム[1][2]を用いたりしている。このとき、一般に分類や検索で用いられるキーはファイルの格納場所または名前や内容の一部などのファイル自体に関連付けられた情報のいずれかであり、これらを覚えていなければファイルにアクセスすることは難しい。

この問題に対して、筆者らは、ユーザのファイルに対する体験（ユーザエクスペリエンス）に含まれる情報が、これまでにユーザがファイルにアクセスする際に用いる手がかりとなりうることを捉え、この情報をファイルにアクセスするためのキーとして用いるためのスキーム[3]を提案している。

本研究では、いろいろなユーザエクスペリエンスのうち、ユーザがファイルをどのように手に入れたかというユーザエクスペリエンスである「ファイルの出自」に着目する。例えば、「Aさんからファイルを貰った」、「Bというサイトからダウンロードした」や「Cというファイルから名前を変更した」などがファイルの出自である。つまり、ファイルの出自はファイルの移動と編集に関するユーザ自身の体験である。本研究では、ファイルの出自の要素となる情報をキーとしてファイルを検索する手法を提案する。先の例の場合、「Aさん」、「貰う」、「サイトB」、「ダウンロード」、「元の名前C」、および「名前変更」がキーである。

2. 現在のファイルアクセス手法の問題点

現在の一般的なOSに採用されているファイルシステムは、フォルダを用いた階層構造に基づいてファイルを管理する機能をユーザに提供している。ユーザは自由に名付けたフォルダにファイルを分類する。そして、ユーザがファイルにアクセスする際はその格納場所であるフォルダの名前を見てフォルダ構造を探索する。

しかし、ユーザはフォルダの名前を覚えていない場合や、ファイルの分類がうまくできていない場合に、ファイルの格納場所が分からなくなることが多い。このとき、ユーザはフォルダを手当たり次第に開いて目的のファイルを探さなければならない。その結果、ファイルアクセスに長い時間がかかる。

この問題に対する解決のひとつとして、デスクトップ検索システム[1][2]などのファイル検索システムが用いられている。これらのファイル検索システムでは、ファイル

[†] 京都工芸繊維大学
Kyoto Institute of Technology

^{††} 京都産業大学
Kyoto Sangyo University

の名前や、サイズ、更新日時などのプロパティ情報や内容の一部をキーとして用いることで格納場所を忘れたファイルにアクセスすることができる。

しかし、ユーザがファイルの内容やプロパティ情報を覚えていない場合が多い。例えば、後で役に立つだろうととりあえずファイルをダウンロードしておいた場合や、人からファイルを貰ったりしただけで中身を確認していない場合など、ファイルの名前や内容を見てすらいない場合がある。また、ファイルの内容を覚えていたとしても、探しているファイルが画像ファイルのような入力可能なキーを含まないファイルの場合もある。これらの場合、ユーザは入力すべきキーが見つからず、検索が行えないという事態に陥る。

このような場合でも、ユーザはファイルにアクセスしようとしているのだから、そのファイルを識別できる何らかの情報は持っているはずである。このような情報の中で、ファイルのプロパティや内容に比べて記憶に残りやすい情報として、ユーザがファイルをどこから手に入れたかという情報、つまりファイルの出自がある。先の例の場合、ファイルを誰から貰ったかの情報やファイルをどこからダウンロードしたかの情報などがファイルの出自である。ところが、ユーザがファイルの出自を覚えていたとしても、それを用いてファイルを検索する手法は提案されていない。

3. ユーザエクスペリエンスに基づく情報資源管理手法

筆者らの所属する研究グループでは、現在の情報資源管理の問題に対して、ユーザエクスペリエンスに基づく情報資源管理手法[3]を提案した。この情報資源管理手法では、ユーザが情報資源に対してさまざまな操作を行う中で得た体験（ユーザエクスペリエンス）に着目している。このユーザエクスペリエンスを5W1Hで記述することにより、5W1Hの各々に相当する情報を用いた情報資源管理の可能性を示唆している。本研究で提案するファイル検索手法はこの情報資源管理手法に基づいている。

多くの場合、ユーザエクスペリエンスはそのユーザの記憶と深く結び付いていると考えられる。このため、ユーザエクスペリエンスを情報資源にアクセスするためのキーとして用いることができれば、アクセスのキーが見つからない事態が減ることが期待できる。

また、ユーザエクスペリエンスに含まれている情報のうち、従来の分類や検索のキーとして用いられていなかった情報を用いて情報資源にアクセスすることができれば、ユーザは情報資源に関するより多くの記憶を利用することができる。その結果、ユーザは多くの情報資源の中から目的の情報資源をより絞り込みやすくなることが期待できる。

4. ファイルの出自に基づく検索手法

2節で述べた、ファイルのプロパティや内容を覚えていないとファイルを検索できないという問題を解決するために、ファイルの出自を用いたファイル検索手法を提案する。

4.1 ファイルの出自

ファイルを手に入れる方法として、ファイルを Web サイトからダウンロードすることや、人からファイルを貰うことがある。これらの方法はファイルを移動したことによってファイルを手に入れたといえる。また、ファイルの内容を変更した場合、古いファイルから新しいファイルを手に入れたと考えられる。このような場合はファイルを編集したことによってファイルを手に入れたといえる。

これらの、ユーザがファイルを手に入れる方法としての移動と編集はユーザが体験したことであるので、ファイルの出自は3節で述べたユーザエクスペリエンスのひとつであるといえる。この情報をファイルアクセスのキーとして用いることができれば、ファイルの内容を覚えていない状況でもファイルにアクセスすることができると考えられる。

なお、ファイルの新規作成もファイルを手に入れる操作といえるが、新規作成には次節で説明するソースにあたる情報がなく、今回提案するモデルからは外れるためファイルの出自として扱わない。しかし、新規作成に関しては、作成日時などをキーとした従来の検索手法で対応可能である。

4.2 ファイルの出自のモデル化

ファイルをどこからどのように手に入れたかというユーザエクスペリエンスを5W1Hで記述することでファイルの出自をモデル化する(図1)。このモデルを構成する要素であるソース、操作、および操作の補足情報がファイルの出自の要素となる出自情報である。

5W1HのHowに当たる部分はユーザが行ったファイルの移動と編集の操作である。また、操作には何らかのアプリケーションが用いられる。例えば、ユーザはWebブラウザでダウンロードしたり、文書編集ソフトウェアで編集したりする。このことから、どのようなアプリケーションを用いたかは操作の一部であると考えられる。そこで、操作に用いられたアプリケーションはHowにあたる操作を補足する情報として扱う。

5W1HのWhereとWhoに当たる部分はファイルがどこから移動したかの情報である。つまり、入手したファイルが以前存在した場所と以前のファイルの持ち主がそれぞれWhereとWhoにあたる。Whatに当たる部分は、ファイルを何から編集したかの情報である。つまり、編集前のファイルがWhatにあたる。本研究ではこれらのWhere、Who、およびWhatに当たる情報をソースと呼ぶ。5W1HのWhenに当てはまる部分は、操作が行われた日時の情報である。これも操作を補足する情報にあたる。

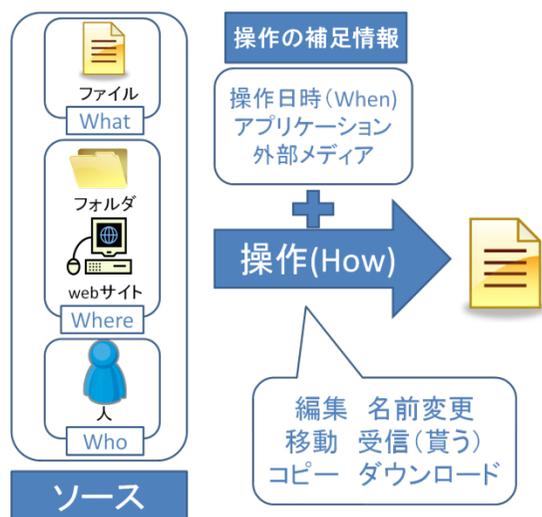


図 1 ファイルの出自のモデル
Fig 1 the Model of "roots of the file"

4.3 提案手法で用いる出自情報

ユーザが頻繁に行うファイルの広義の移動操作には、コピー、フォルダを移動させる狭義の移動、ダウンロード、および受信がある。同様に、ファイルの広義の編集操作には内容を変更する狭義の編集と名前変更がある。一般に、削除もファイルをごみ箱へ移動させる操作と考えられるが、移動先がごみ箱と決まっているので本研究では削除を対象としない。また、受信やダウンロードのような高レベルのファイル操作には他にも移動や編集を意味する操作が複数あるが、提案手法では普段ユーザがよく使うと考えられる、先に述べた6つの操作(コピー、狭義の移動、ダウンロード、受信、狭義の編集、名前変更)を扱う。以下、移動と編集は特に明記しなければ狭義のものを表す。

なお、受信という操作には、ファイルをメールで受信することの他にUSBメモリなどの外部メディアで人からファイルを貰うことを含める。この外部メディアもアプリケーションと同様に操作の情報の一部として扱う。

それぞれの操作に対するソースを以下に示す。

- コピー：コピー元のファイル，コピー元のフォルダ

- 移動：移動元のフォルダ
- 名前変更：変更前のファイル名
- 受信：送信者
- 編集：編集前のファイル
- ダウンロード：ダウンロードした URL，サイト名

5. 関連研究

ファイルアクセスについての問題の解決を試みた研究が複数ある。本節ではそれらの研究と本研究との関係について述べる。

5.1 時間中心のファイルアクセス手法

暦本らのTime-Machine Computing[4]はファイルの置き場所をデスクトップのみとし、その状態を常に記録していく。ユーザは時刻を指定することで、デスクトップを過去の状態に戻すことができる。ユーザはそこに置かれた過去の状態のファイルにアクセスすることができる。

Time-Machine Computing では時刻を指定することで過去のファイルにアクセスする手法を提案しているが、本研究では、ファイルの過去の情報である出自情報をキーとすることで現在のファイルにアクセスする手法を提案している。

5.2 ファイル間関連度を利用したファイルアクセス手法

大澤らによる俺デスク[5]は、ユーザの履歴情報の想起を効率化することを目的としている。そのための機能のひとつとして関連度検索ツールを提供している。このツールは操作履歴からファイルやwebサイトなどのデータ間の関連度を算出し、指定したファイルと関連度の高いファイルをユーザに提示するものである。また、データの相対的な着目指標として、ユーザがデータを参照しているときの様々な操作からデータ着目度を算出し、これと時間を軸としたタイムラインビューを提供している。

俺デスクでは関連度検索を用いることでファイルの内容から得られる情報やプロパティ情報を覚えていなくてもファイルを探索することができる。また、タイムラインビューでファイルを探索することは「あの時見たファイル」というユーザエクスペリエンスを用いたファイル探索といえる。しかし、俺デスクでは4.3節で述べたような特定の操作を行ったことをキーとしてファイルを探索することができない。

渡部らの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[6]は、キーワードを含まないファイルをそのファイルと関連度の高いファイルに含まれるキーワードで検索することができるようにする手法である。この手法は、直接関連があるはずのファイルの複製、移動、および改名の操作が行われる前後のファイルの間にある関連性は利用されない。一方、本手法ではこれらの操作が行われたことを出自情報として用いて検索することができる。

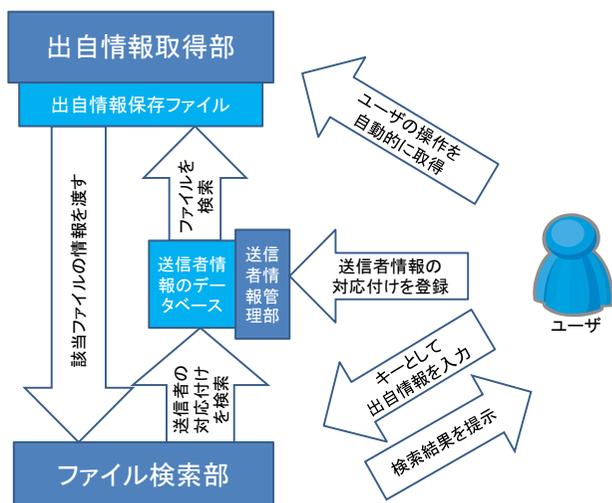


図 2 システム構成

Fig 2 the Structure of the proposed system

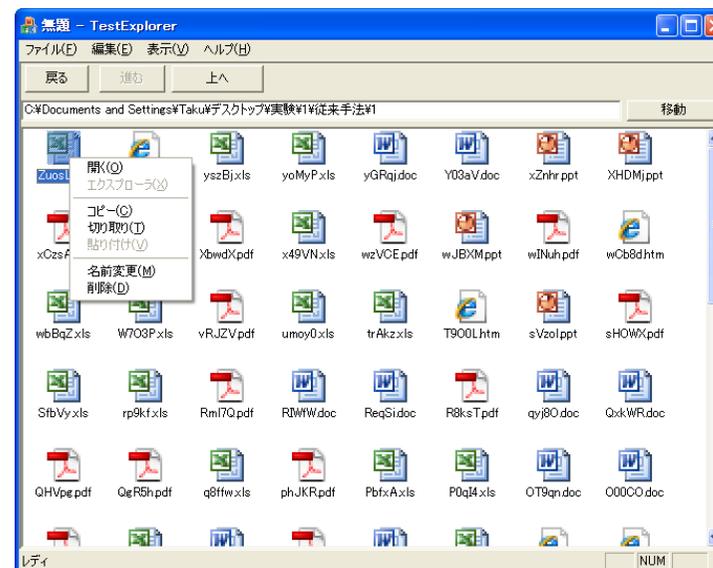


図 3 作成したファイルエクスプローラの外観

Fig 3 an Overview of the implemented file-explorer

6. 出自検索システムの実装

本節では出自検索手法を実現するために実装したシステムについて述べる. 本システムは出自情報取得部とファイル検索部, およびファイルの送信者情報管理部の3つから構成される (図 2).

6.1 出自情報取得部

本システムではユーザーが出自に関する操作を行う代表的なアプリケーションである以下のアプリケーションでの出自情報を取得の対象とした.

- ファイルエクスプローラ
- Mozilla Firefox 3[7]
- Mozilla Thunderbird 2[8]
- Microsoft Word 2007
- Microsoft Excel 2007

取得した出自情報はファイルごとに対応した出自情報記録用のファイルに保存さ

れる. ひとつの操作ごとに, ソース, 操作年月日, その操作を行ったときに利用したアプリケーションあるいは外部メディア, および操作を記録する.

ファイルエクスプローラでの出自情報の取得はWindowsエクスプローラの機能の一部を有するアプリケーション (図 3) を作成しその上での操作を記録することで行った. Firefoxではダウンロードマネージャのデータベースを監視することで, Thunderbirdでは既存のアドオン[9]を利用することで出自情報を取得した. WordとExcelではキーボードフック, マウスフック, およびMicrosoft WindowsにおいてGUIによって表現されるアプリケーション内のオブジェクトの情報を取得することができるMicrosoft Active Accessibility[10]を利用し, ユーザの行った操作を取得することで, 出自情報を取得した.

6.2 ファイル検索部

検索システムのインタフェースを図 4 に示す. ユーザは図 4 の上部にある入力フィールドに出自情報を入力し, 検索ボタンを押すことでファイルの検索を行う. 入力フィールドは 4.2.1 節で述べたモデルと対応しており, ソースである元のファイル名, 元のフォルダ, サイト名を入力するテキストボックスと, 送信者である人を入力する

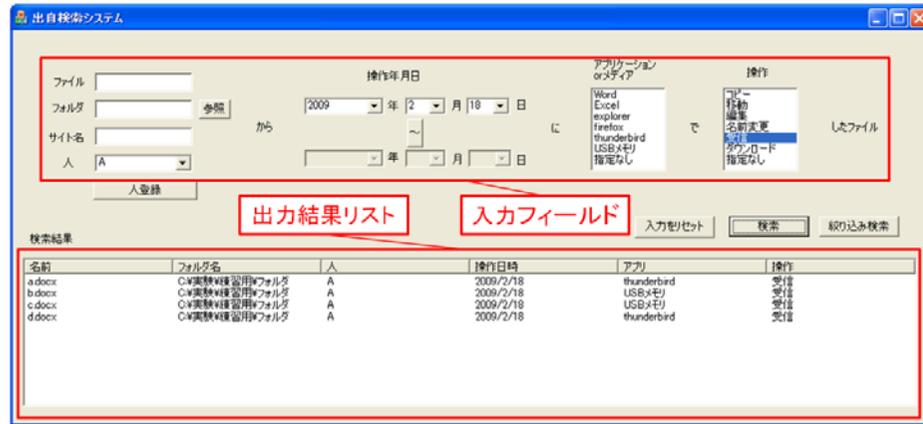


図 4 検索システムのインターフェース
Fig 4 the Interface of the retrieval system

コンボボックス、操作の補足情報である操作年月日を入力するコンボボックス、および操作を行ったアプリケーションや外部メディアと操作を選択するリストボックスがある。

検索を行うと、各入力フィールドに入力されたすべての情報と合致する出自情報を持つファイルが検索結果として提示される。この検索結果は図 4 下部のリストに表示される。表示される情報は、ファイルの名前、保存されているフォルダ、および出自情報である。

6.3 送信者管理部

出自情報のうち、ファイル送信者についての情報は、システムが送信者を特定する情報として外部メディアに記載された名前やメールアドレスなど複数存在する。しかし、ユーザが記憶している「名前」はシステムが取得した人を特定する情報ではなく、ユーザがその人を認識するために用いている名前である。この名前を検索に用いるために、その名前とシステムが取得した情報の対応を管理する必要がある。提案手法では図 4 の人登録ボタンを押すことでこの対応付けを記録するウィンドウが開く。ユーザがこれに名前の対応付けを記述することで登録する。今回、外部メディアの所有者情報は、メディア内にあらかじめ用意しておいた所有者情報を記したテキストファイルから取得した。

7. 実験

提案手法は、出自情報は覚えているがファイル自体の情報は覚えていない状況での使用を想定している。この状況で、提案手法を用いたファイルの探索が、従来行われているエクスプローラを用いた探索と比べて有効であるかを実験により確認した。

7.1 方法

被験者は 100 個のファイルが入ったひとつのフォルダの中から実験者が指定したファイルを探すタスクを行った。

本節のはじめに述べた状況をシミュレートするために、実験タスクでは、出自情報と、ファイルに含まれる画像や図を提示することで探索するファイルを指定した (図 5)。このとき、提示される出自情報は 7 種類の内のひとつとした。また、被験者が前のタスクの記憶を次のタスクに持ち越さないように、ファイル名は英数字 5 文字のランダムな文字列とし、タスクごとに名前を付けかえた。なお、各ファイルの出自情報は実験者が用意した。また、本実験で用いたファイルは一般的に用いられており、かつ画像を含むことのできるファイル形式である doc, xls, ppt, htm, および pdf とした。そして、目的のファイル以外のファイルは各タスクで同じ内容のファイルを用いた。

提案手法を用いたファイルの探索は、目的のファイルと同じ出自情報を持つファイルの数によって探索時間に差が出ると考えられる。つまり、検索結果として表示されるファイルが多いと探索時間が長くなると考えられる。このことが提案手法の有効性に影響を与えると考えられる。この影響を調べるために目的のファイルと同じ出自のファイルの数が 1 個、5 個、10 個の 3 つの条件で提案手法を用いて探索した場合、および従来行われている手法である、エクスプローラのすべての機能を用いて探索した場合の計 4 種類の探索状況を比較した。

実際の検索では、ユーザは元のファイル名、元のフォルダ、URL・サイト名、人、操作年月日、操作に用いたアプリケーションや外部メディア、および操作の 7 種類の出自情報を様々な組み合わせで入力できるが、本実験ではこれら 7 種類の出自情報それぞれが検索のキーとして有効かどうかを調べるために、それぞれを単独で使用することにした。

以上の探索状況 4 種類と出自情報 7 種類を独立変数として実験を設計した。被験者には 1 人当たり探索状況 4 種類、出自情報の種類 7 種類を組み合わせ合計 28 種類の探索タスクを行わせた。被験者は探索状況ごとに出自の種類に対応した 7 つのタスクを行った。この 7 つのタスクの順番はランダムとし、全被験者の各探索状況で同じ順番にならないようにした。また、4 種類の探索状況の順番は各状況で最初に行う被験者が 3 人ずつになるようにし、その 3 人の中で同じ順番にならないように割り当てた。

探索開始時、被験者に「探索開始ボタン」(図 6) を押させた。被験者は目的のファ

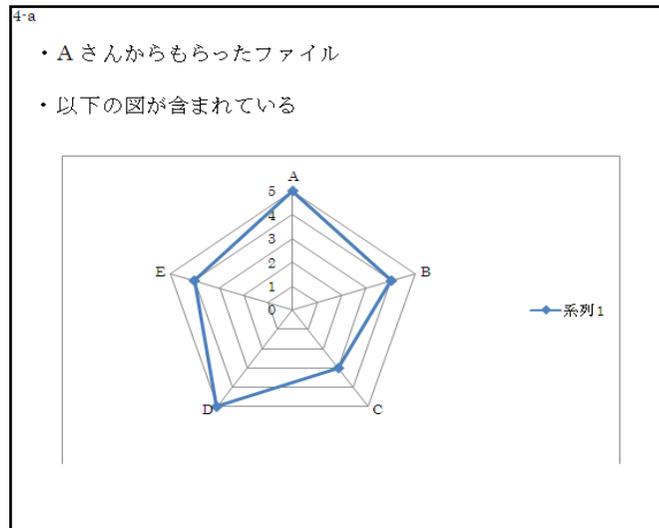


図 5 タスク課題の例
Fig 5 an Example of the task

イルが見つかったと思ったら「探索完了ボタン」を押し探索をやめるように指示した。その後、実験者が被験者の見つけたファイルが正しいファイルかどうか判定した。もし、間違っていれば、被験者に「探索再開ボタン」を押して探索を再開させた。正解の場合はタスク終了として次のタスクへ移った。なお、3分以内にファイルを発見できなければそのタスクはエラーとした。この場合もタスクは終了とし、次のタスクへと移った。

従属変数は探索時間とエラー率とした。探索時間は探索性能の比較の指標とし、エラー率は常識的な時間で探索が完了できるかどうかの指標とした。探索時間は被験者が「探索開始ボタン」を押してから正しいファイルを見つけて「探索完了ボタン」を押すまでの時間とした。このとき、実験者が被験者の探索したファイルが正解かどうかの確認に要した時間は、探索時間には含めない。また、エラーとなったタスクの探索時間は3分として扱った。エラー率は、28種類のタスクごとに、全被験者数のうちそのタスクがエラーとなった被験者の割合とした。

被験者は20代の大学生と大学院生計12人であった。



図 6 探索開始/完了/再開ボタン
Fig 6 Start/finish/restart-button

7.2 手順

まず、被験者に提案システムの使用方法について説明し、実際に使用してもらった。次に、エクスプローラのアイコンの種類、アイコンの整列、および検索について説明した。そして、被験者に探索タスクについて説明した。以上の説明について質問がないか被験者に確認し、質問がなければ探索タスクを行わせた。28種類の探索タスクが終了すれば実験終了とした。

7.3 環境

実験で用いたPCは、OSとしてWindows XP Professional、CPUにCeleron D 2.40GHz、1GBのメモリを搭載したデスクトップ型PCであった。ディスプレイは解像度1280×1024ピクセル、対角43.18cm(17インチ)のものを用いた。

また、被験者が実験を行う様子をビデオカメラで撮影した。

7.4 結果と考察

全体で336回のタスクのうち、エラーとなったタスクは37回であった。エラータスクを含めた全タスクの探索時間の平均は56.5秒(標準偏差53.3秒)であった。

まず、探索時間の結果を図7に示す。この結果に対して分散分析を行った結果、独立条件である探索状況と出自情報の種類の主効果と、交互作用が有意(それぞれ $p<0.01$, $p<0.01$, $p<0.05$)であった。そこで、単純主効果検定を行った結果、提案手法を用いた3つの探索状況での探索時間は、どの出自情報を指定したときにおいても、エクスプローラでの探索時間より有意(すべて $p<0.01$)に短かった。また、どの出自情報の種類を用いたかによって、有意差があったのは、エクスプローラを用いた場合の、元のファイルを指定した場合と元のフォルダ以外を指定した場合の間と、元のフォルダを指定した場合とサイト名を指定した場合の間であった。このことから、エクスプローラを用いたときのみ出自情報の種類による効果があるという交互作用が現れたといえる。この有意差が表れたのは、用意したファイルやその指定の仕方の違いによって、目的のファイルの種類が予想できてしまったことが原因だと考えられる。

次に、エラー率の結果を図8に示す。この結果に対して分散分析を行った結果、探索状況の主効果と交互作用が有意(それぞれ $p<0.01$, $p<0.05$)であった。そこで、単純主効果検定を行った結果、提案手法を用いた3つの探索状況でのエラー率は、出自

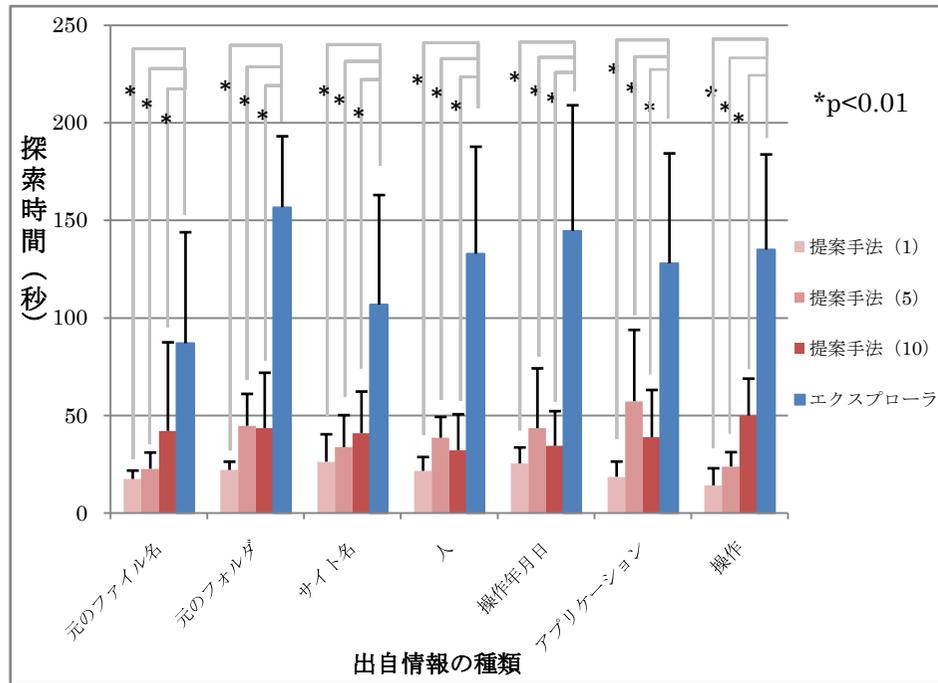


図 7 結果 (探索時間)
 Fig 7 Result (explore-time)

情報としてファイル名以外を指定したとき、エクスプローラでのエラー率よりも有意 (すべて $p < 0.01$) に低かった。また、どの出自情報の種類を用いたかによって、有意差があったのは、エクスプローラを用いた場合のみで、操作年月日を指定した場合と元のフォルダ以外を指定したときの間と、元のファイル名を指定した場合と元のフォルダを指定した場合の間であった。このことから、エクスプローラを用いたときのみ出自情報の種類による効果があるという交互作用が現れたといえる。この有意差が現れた理由も探索時間の場合と同様と考えられる。

なお、提案手法を用いた状況でエラーが起こったのは1回だけであり、これは出自情報としてファイル名を指定し、その条件に当てはまるファイルが目的のファイル以外に10個ある条件であった。

提案手法を用いた状況で起こったエラーが1回だけであったことから、提案手法を用いた場合は、常識的な範囲の時間でファイルを見つけることができるということが

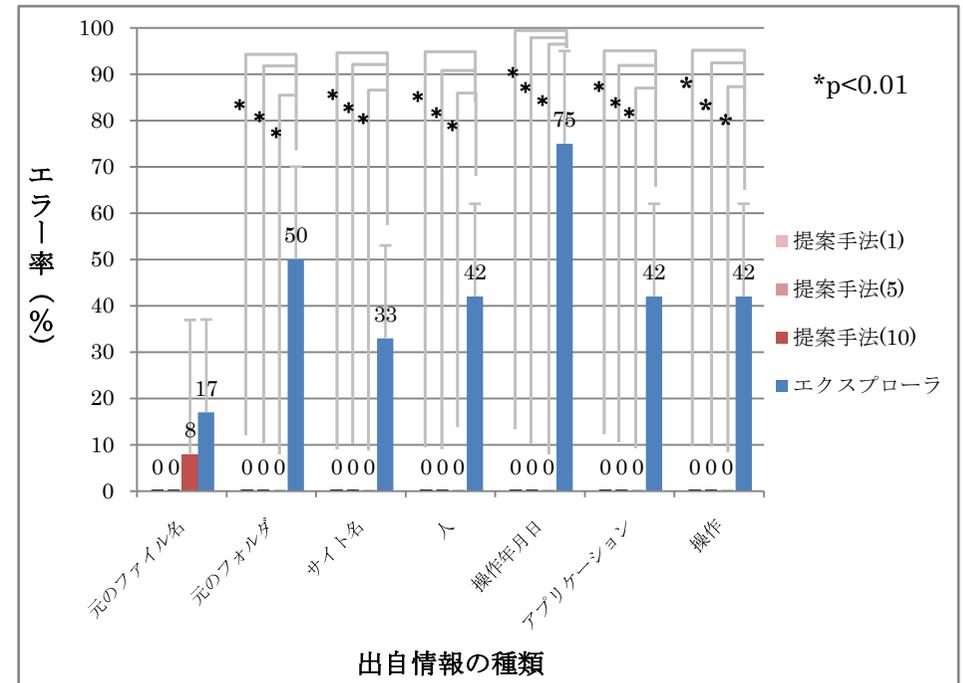


図 8 結果 (エラー率)
 Fig 8 Result (error rate)

わかった。

エクスプローラを用いた場合よりも提案手法を用いた場合の方が、探索時間が有意に短かったことと、出自情報としてファイル名以外を指定したとき、エクスプローラよりも提案手法を用いた場合の方がエラー率が有意に低く、提案手法を用いた場合にエラーがほぼなかったことから、エクスプローラよりも提案手法を用いる方がファイルを容易に見つけることがわかった。また、目的のファイルと同じ出自情報を持つファイルが10個あったとしても、提案手法の方がエクスプローラよりも容易にファイルを見つけることができることもわかった。

8. おわりに

本研究では、ファイルアクセスのキーとして現在用いられているファイル自体の情

報だけでなく、出自情報を用いることでより柔軟なファイルアクセス手法を提供することを目的とし、これを達成するための手法を提案した。そして、試作システムを実装し提案手法の有効性を確かめる実験を行った。その結果、提案手法が用いられると想定される状況で、提案手法がエクスプローラでファイルを探るより容易にファイルを見つけることができることが確認できた。

今後の課題として、出自情報を取得するアプリケーションを代表的なアプリケーションに絞ってシステムを実装したが、出自検索システムをより実用的にするためには、より多くのアプリケーションで出自情報を取得できるようにする必要がある。

また、本研究の実験では初期的な実験として、出自情報を単独で使用した場合について調べたが、実際に行われる複数の出自情報を組み合わせて使用する場合の提案手法の評価も行う必要がある。

本研究の実験では仮の出自情報を実験者が用意したが、実際にユーザが体験した出自情報を用いた場合の提案手法の評価も行う必要がある。

参考文献

- 1) Google : Google デスクトップ ; <http://desktop.google.com/ja/index.html> (2009/4/13).
- 2) Microsoft : Window デスクトップサーチ ; <http://www.microsoft.com/japan/windows/desktopsearch/default.aspx> (2009/4/13).
- 3) 重森 晴樹, 後藤 啓太, 倉本 到, 渋谷 雄, 辻野 嘉宏, 水口 充 : ユーザエクスペリエンスに基づく情報資源管理スキーム, 情報処理学会研究報告, 2008-HCI-129, pp.43-48 (2008).
- 4) Rekimoto, J.: Time-Machine Computing: A Time-Centric Approach for the Information Environment, UIST '99 : Proceedings of the 12th annual ACM symposium on User interface software and technology, New York, NY, USA, ACM, pp.45-54 (1999).
- 5) Ohsawa, R., Takashio, K. and Tokuda, H.: OreDesk: A Tool for Retrieving Data History Based on User Operations, ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia, Washington, DC, USA, IEEE Computer Society, pp. 762-765 (2006).
- 6) 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, DEWS2008 E10-6 (2008).
- 7) Mozilla : Mozilla Firefox 3 ; <http://mozilla.jp/firefox/> (2009/4/13).
- 8) Mozilla : Mozilla Thunderbird 2 ; <http://mozilla.jp/thunderbird/> (2009/4/13).
- 9) EvilJeff : Attachment Extractor ; <http://www.eviljeff.com/?page=moz-extensions> (2009/4/13).
- 10) Microsoft : Microsoft Active Accessibility 2.0 ドキュメント ; <http://msdn.microsoft.com/ja-jp/accessibility/bb960468.aspx> (2009/4/13).

- 1 Google : Google デスクトップ ; <http://desktop.google.com/ja/index.html> (2009/4/13).
- 2 Microsoft : Window デスクトップサーチ ;
<http://www.microsoft.com/japan/windows/desktopsearch/default.aspx> (2009/4/13).
- 3 重森 晴樹, 後藤 啓太, 倉本 到, 渋谷 雄, 辻野 嘉宏, 水口 充 : ユーザエクスペリエンスに基づく情報資源管理スキーム, 情報処理学会研究報告, 2008-HCI-129, pp.43-48 (2008).
- 4 Rekimoto, J.: Time-Machine Computing: A Time-Centric Approach for the Information Environment, UIST '99 : Proceedings of the 12th annual ACM symposium on User interface software and technology, New York, NY, USA, ACM, pp.45-54 (1999).
- 5 Ohsawa, R., Takashio, K. and Tokuda, H.: OreDesk: A Tool for Retrieving Data History Based on User Operations, ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia, Washington, DC, USA, IEEE Computer Society, pp. 762-765 (2006).
- 6 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, DEWS2008 E10-6 (2008).
- 7 Mozilla : Mozilla Firefox 3 ; <http://mozilla.jp/firefox/> (2009/4/13).
- 8 Mozilla : Mozilla Thunderbird 2 ; <http://mozilla.jp/thunderbird/> (2009/4/13).
- 9 EvilJeff : Attachment Extractor ; <http://www.eviljeff.com/?page=moz-extensions> (2009/4/13).
- 10 Microsoft : Microsoft Active Accessibility 2.0 ドキュメント ;
<http://msdn.microsoft.com/ja-jp/accessibility/bb960468.aspx> (2009/4/13).