

## ユーザ経験としてのタスク-資源関係に基づくファイル管理手法

後藤 啓太†, 重森 晴樹†, 倉本 到†, 渋谷 雄†, 辻野 嘉宏†, 水口 充††

(† : 京都工芸繊維大学, †† : 京都産業大学)

現状の PC 使用環境では, ユーザはフォルダを用いた階層構造でファイルを管理している。しかし, この階層構造型のファイル管理では, 時間の経過と共にファイルの最適な分類構造が変化する, ファイルの保存場所を忘れる, などの問題がある。一方, ユーザはタスクを多数抱えながら生活しており, そのタスクを実行するには資源が必要である。オフィスにおけるタスクの代表的な資源にファイルがある。本研究では, タスクとファイルの間の「使う-使われる」という関係に着目し, タスクを基にしたファイル管理手法を提案し評価を行った。その結果, ユーザに柔軟なファイル管理手法を提供でき, ユーザのタスク実行の支援も可能であることがわかった。

## A File Management System Based on Task-Resource Relation as One of User-Experiences

Keita Gotoh †, Haruki Shigemori †, Itaru Kuramoto †, Yu Shibuya †, Yoshihiro Tsujino †, Mitsuru Minakuchi ††

(† : Kyoto Institute of Technology, †† : Kyoto Sangyo University)

Today, PC users manage their files with hierarchical file structure systems. However, these file systems have some problems. One of the problems is that proper classification structures will change. Another one is that users occasionally forget locations of files. On the other hand, users daily manage many tasks that require some resources. Concerning office tasks, files are typical resources. Thus, we focus that the relationship of "use - be used" between tasks and files. We propose a file management system based on tasks. As the result of an evaluation of the system, it was found that the proposed system was flexible for the file management and was effective on supporting user's task execution.

### 1. はじめに

現状の PC 使用環境において, 日々ユーザは様々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, ユーザは作成した多量のファイルをフォルダで構成された階層構造を用いて管理している。

しかし, フォルダで構成された階層構造には様々な問題点が存在する。例えば, 目的のファイルへアクセスするためにはそのファイルが格納されているフォルダ(場所)を特定する必要がある。

しかし, 階層構造ではそのフォルダを構成する要素に親子関係が存在し, 順番が異なると別のファイル集合を指す。このとき, ユーザがファイルにアクセスする際に, 手がかりとなるフォルダの要素を順番通りに思い出すとは限らないため, フォルダを特定しづらくなり, 目的とするファイルへのアクセスが困難になる可能性がある。また, 時間の経過とともにファイル数が変化し, それに伴いユーザはファイルを再分類するとフォルダ構造は変化するため, ファイルの保存場所がわからなくなるということもありうる。

そのため現状では, デスクトップサーチ[1][2]

などの検索システムを使用しファイルを探す手法が利用されている。しかし, この検索システムはファイル名やファイルの内容の全文検索のようにファイル自体の情報をキーとして検索する。しかし, ユーザが知りたいのはファイル自体の情報であることが多い。そのため, ファイル自体の情報をキーにすることには本質的な問題がある。

そもそもファイル検索のキーとなる情報はユーザの記憶にある情報であり, ユーザの記憶はユーザの経験に基づいて生成される。そこで, ユーザの経験(ユーザエクスペリエンス)にファイルアクセスのためのキーが存在しているのではないかと考えられる[3]。

本稿では, ユーザエクスペリエンスのひとつである, 「ユーザが各タスクでどのファイルを使用したか」という点に着目し, このタスクとファイルの関係に基づいたファイル管理手法を提案する。ここでタスクとは, 何らかの目的を達成するために行われる作業である。タスクを実行するには資源が必要であり, オフィスにおける PC を用いたタスクの主要な資源はファイルである。この両者の間には「(ファイルを)使う - (タスクで)使われる」という関係が存在する。提案手法によって,

「ユーザが過去にそのファイルを使用したタスク」をキーとしたファイルアクセスを可能にする。またこの関係を利用することでタスクの開始・再開支援が可能となる。

## 2. フォルダを用いた階層構造型のファイル分類手法の問題点

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多数のファイルを分類し、必要なときに必要なファイルをすぐに取り出せるようにしようとすることが多い。

しかし、フォルダを用いた階層構造型のファイルシステムには様々な問題点が指摘されている[4]。ユーザは基本的にファイルを整理しやすいよう分類し、ファイルを取り出すときはファイルが一貫した整理方法で分類されているものと思ってファイルへアクセスしようとする。整理しやすい分類と一貫した分類とは必ずしも一致しないので、ユーザのファイルを分類するときの分類基準と、ファイルへアクセスするときにユーザが考えている分類基準が異なる可能性がある。そうすると、必要なファイルにアクセスすることが難しくなる。その場合、ユーザは目的とするファイルがありそうなフォルダを逐一探したり、検索システムを用いて目的のファイルを探すことになる。これらはファイルを一貫した分類方法で管理することが難しいことを意味している。その理由として、以下のものがある。

### 2.1 ファイルの最適な分類構造の可変性

ユーザはファイルをその作成理由による意味的な包含関係に基づいて分類しようとする。しかし、ファイル数が増えてくると、整理しやすいという理由でその関係を変更することがある。

例えば、音楽ファイルを入れるフォルダ「音楽フォルダ」の中に、お気に入りの曲を入れる「お気に入りフォルダ」があったとする。ユーザは当然このお気に入りフォルダに気に入った曲を入れていく訳だが、時間が経過すると、お気に入りフォルダ内の曲が増えてくる、あるいは、当時気

入っていた曲が気に入らなくなっている場合がある。そのときユーザはフォルダ内のファイルを再分類する。音楽フォルダ内の音楽ファイルを全て再分類するのならばよいのだが、しばしば、このお気に入りフォルダ内のファイルのみ再分類することがある。気に入らなくなった曲をお気に入りフォルダに入れたまま、お気に入りフォルダの中に「さらにお気に入りフォルダ」なるものを作り、その中に気に入った曲を入れる、ということになりかねない。これはユーザがファイル数の増加に応じて意図的に分類方法を変更していることになる。これに対応するため、ファイル数が少ない時から細かく分類したとしても使いやすくない。このように、いつまでも使いやすい分類構造を用意することは困難である。

### 2.2 格納先フォルダの一貫性の欠如

ユーザはファイルを分類するためにフォルダを作るが、フォルダ名によっては複数のフォルダに分類可能となるファイルがある。例えば、論文で使用する実験結果のファイルを分類する場合、「実験結果」フォルダと「論文」フォルダがあるとするなら、どちらのフォルダに格納しても意味が通る。

しかし、このファイルを一方のフォルダにのみ格納すると、次にこのファイルにアクセスしたいときに、ユーザはどちらのフォルダに格納されているのかをフォルダの名前から判断することができない。また、格納先として別のフォルダもありえることに気付かず分類してしまうこともありえる。

また、ファイルをコピーしてそれぞれのフォルダに格納することもできるが、その場合、一方のファイルを編集すると、その編集内容がもう一方のファイルには反映されないため、編集後に再度コピーしなおす必要がある。

どちらかにショートカットを作ることで、上で挙げた問題は解決されるが、Microsoft Windows ではショートカットが一種のシンボリックリンクであるため、リンク先ファイルの場所や名前が変更するとリンクが途切れるという問題もある。また、万能なショートカットがあったとしても、他方にショートカットを入れ忘れるという問題が残る。

### 2.3 ユーザのファイル格納場所の忘却

ユーザの記憶は時間が経つとともに曖昧になる。これはどのようにファイル分類を行ったとしても生じる問題である。フォルダを用いた階層構造では、ユーザはフォルダに名前をつけて管理しているが、フォルダ名を決めるのはユーザであり、ユーザは正しいファイルにアクセスするために、各ファイルがどのフォルダに存在しているかを覚えていなくてはならない。このように、フォルダによる分類ではフォルダ名が分類の鍵となるが、フォルダ名を決めた理由を覚えておかないと、フォルダ名だけではそのフォルダが何のために作られたのかわからなくなる場合があり、そのフォルダの中に何のファイルが入っているのかわれた場合に推測しにくくなる。

このため、時間が経過したときにユーザが過去に作成したファイルを発見しようとする時、そのファイルが格納されている場所を明確に思い出せなくなり、ファイルアクセスが困難になる。

### 3. タスク - 資源関係に基づくファイル管理手法

2章で述べた問題点を解決するために、オフィスでの代表的なエクスペリエンスである仕事（タスク）からファイルを検索（アクセス）するファイル管理手法を提案する。この手法は、従来のファイル検索における、ファイルの置き場所（フォルダ）、ファイル名、ファイル内容による検索ではなく、そのファイルを過去にどのように使ったかというユーザエクスペリエンスをキーとした検索の1つであり、フォルダ構造とは別のファイル分類を実現する（図1）。

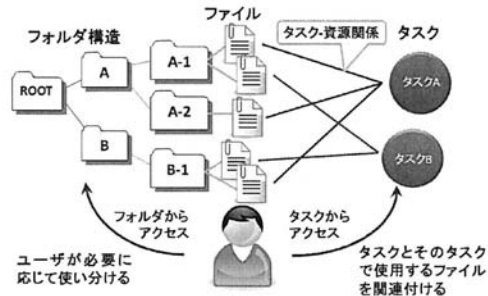


図1：タスク - 資源関係を用いたファイルアクセスの例

Fig. 1: An example of the file access using task - resources relations.

### 3.1 タスクとファイル

ユーザがPC上で何かをしているとき、それは常に何らかのタスクの一部である。そのタスクを実行するには資源が必要であり、オフィスにおけるPC環境上のタスクについては、代表的な資源としてファイルが挙げられる。このタスクとファイルの間には「(ファイルを) 使う - (タスクで) 使われる」という関係が存在する。

タスクをキーとしたファイル管理を実現するために、このタスクとファイルの関係を用いてタスクとファイルの関連付けを行う。具体的には、タスクにそのタスクで使用するファイルを登録することである。

### 3.2 一貫したファイル分類

提案手法では、タスクでファイルを分類するので、「タスクで使用するファイルを関連付ける」という分類基準が一貫したファイル分類方法である。そのため、2.1節で述べた問題を解決するファイル分類が可能になる。また、タスクへファイルを関連付けるだけなので、ユーザはファイル数に応じた分類構造の細かさを意識する必要がなくなる。

### 3.3 複数のタスクでのファイル共有

提案手法では、複数のタスクで同一ファイルを共有できるように、2.2節で述べた、複数のフォルダへ格納可能なファイルの扱いが困難であるという問題を解決する。ファイルをタスクへ関連付けることを考えた場合、複数のタスクで同一ファイルを使用するときは複数のタスクと関連付ければよく、ファイルは必ずタスクを行うために必要なため、関連付けの漏れが起こり得ない。タスクから見ればファイルはそのタスクで使用する資源なのでファイルが一意に定まる。

### 3.4 タスクをキーとしたファイルアクセス

提案手法では、ファイルの目的（タスク）という情報をキーとしたファイルアクセスが可能である。そのため、ユーザはファイルの場所を覚える必要はないので、2.3節で述べた、ファイルの場所の情報のみでファイルアクセスする場合の問題を解決する。

### 3.5 タスクの実行支援

ユーザがタスクを開始するとき、従来はそのタ

スクで使用するファイルを自分で探して開く必要がある。また、過去に中断したタスクを再開する場合、中断前に使用していたファイルを再度用意し、中断時に行っていたことを思い出し、その続きを行う。

提案手法ではタスクを実行するために必要なファイルがタスクに関連付けられているので、タスク開始/再開時にはそのタスクで使用するファイルを探すことなく自動的に準備することができる。さらに、タスクを再開するときは、中断時のウィンドウの配置の状態を復元することで、中断前に何をしていたのか思い出しやすくなる。

これにより、ユーザのタスクを実行するための準備の手間を減らすことができ、効率的なタスク開始・再開を支援できると考えられる。

#### 4. タスク - 資源関係に基づくファイル管理システムの試作

##### 4.1 タイムラインナビ

本研究では、実装のベースとしてタイムラインナビ[5]を用いた。タイムラインナビは、スケジュールやタスクとその見込み時間を登録することで、ユーザの全体の空き時間を提示するスケジューラシステムである。ここで、空き時間とは、ユーザが抱えているあるタスクの締め切りまでに自由に使える時間のことであり、それぞれのタスクごとに定義されるものである。また、見込み時間とは、タスクを完了させるために必要であるとユーザが見積もる主観的な作業時間のことである。

図2にタイムラインナビの全体図、およびカレンダー上のタスクを右クリックした場合の図を示し、図3に未完了のタスクを表示するウィンドウである「現在のタスクダイアログ」を示す。図2のカレンダーには、完了、未完了を問わず、登録したタスクをすべて表示する。現在のタスクダイアログは図3の右下にあるグレーの無地のボタンを押すごとに通常表示状態と全表示状態が切り替わる。通常表示状態では現在実行可能な未完了のタスクが表示され、全表示状態では現在実行不可能な（タスクの開始時刻になってない）未完了のタスクも加えた表示になる。また、タスクの状態

を表す、「現在のタスクダイアログ」のタスク名左横に表示される記号とその意味の一覧を表1に示す。



図2：タイムラインナビ

Fig.2: Time Line Navi.



図3：現在のタスクダイアログ

Fig.3: A current task dialogue.

表1：現在のタスクダイアログの記号

Table1: Signs of a current task dialogue.

記号	意味
▶▶	現在実行しているタスク
▶	現在実行可能なタスク
■	現在実行不可能なタスク
	イベントが発生すれば実行可能となるタスク
!	空き時間が0以下となっているタスク

##### 4.2 タスクとファイルの関連付け

タスクとファイルに関連付けて扱う方法は従来から存在する。例えば、バーチャルデスクトップ機能を用いてユーザがタスク毎にデスクトップを切り替え、各タスクで使用するファイルを別々の

デスクトップに置いておくということはこれまでもあった。しかし、タスクとファイルの関連付けに関して、システム側からの支援は何も行われていない。また、バーチャルデスクトップを用いてタスク（デスクトップ）毎にファイルを割り付ける作業には手間がかかり面倒である。本研究では、タイムラインナビを用いてこの関連付けを容易にしている。

#### 4.2.1 実現方法

タスクとファイルの関連付けを行うために、タスクごとにそのタスクで使用するファイルを格納するための「作業フォルダ」を用意する。この作業フォルダは、タスクを登録する際に自動的に作成される。試作システムでは、次の3種類の方法を用いて、タスクの作業フォルダにファイルを格納することで、そのタスクとファイルを関連付けることができる。

1. 図3内のタスク名をクリックすると出現する「タスク情報ダイアログ」(図4)の中の「ファイルの関連付け」ボタンを押して行う方法
2. 図3内のタスク名を右クリックすると出現するメニュー(図5)の中の「ファイルの関連付け」を選択して行う方法
3. 図3内のタスク名の上へ、関連付けしたいファイルをドラッグ&ドロップする方法

また、現状では実装できていないが、タスク実行中に新たに作成されたファイルや開いたファイルが自動的にタスクへ関連付けられる機能が必要である。

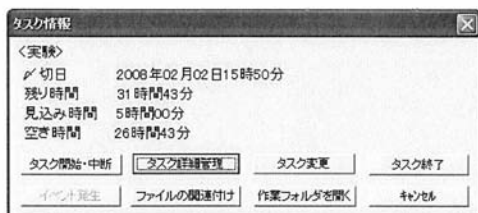


図4：タスク情報ダイアログ

Fig.4: A task information dialogue.



図5：現在のタスクダイアログでの右クリックメニュー

Fig.5: The right-click menu in the current task dialogue.

#### 4.2.2 タスクと関連付けしたファイル

本システムでは、タスクにファイルの関連付けを行うと、ファイルの実体ではなくファイルのショートカットが作業フォルダ内に置かれる。こうすることで、同じファイルを異なるタスクに関連付けた場合に、一方のタスクで変更を加えたのち他方のタスクでそのファイルを参照すると変更がなされていないといった事態を防ぐことができる。もし異なるタスク間でお互いの変更を反映しなくてはならぬ場合、ユーザが手動でファイルのコピーを作業フォルダに入れることもできる。

現状では実装できていないが、将来的にはショートカットではなくハードリンクで実装することで、リンク先ファイルの場所や名前に変更があったとしても対応できるように考えている。

#### 4.2.3 タスクをキーとしたファイルのアクセス

ユーザは試作システムに登録してあるタスクから作業フォルダ内のファイルへのアクセスが可能である。また、エクスプローラから作業フォルダにアクセスすることも可能である。タスクの作業フォルダへは以下の方法でアクセスする。

1. タイムラインナビのカレンダー上のタスク名を右クリックすると出現するメニュー(図2)の中の「作業フォルダを開く」を選択する方法

2. 図 3 内のタスク名をクリックすると出現する「タスク情報ダイアログ」(図 4) 中の「作業フォルダを開く」ボタンを押して行う方法
3. 図 3 内のタスク名を右クリックすると出現するメニュー (図 5) 中の「作業フォルダを開く」を選択して行う方法
4. タイムラインナビのメニューバーの中の「表示」のプルダウンメニューの中の「終了したタスク一覧」を選択する方法
  - ・選択すると終了したタスクが一覧となって表示される (図 6) .この一覧の中からアクセスしたいタスクを探して、そのタスクの作業フォルダを開くことでファイルにアクセスする

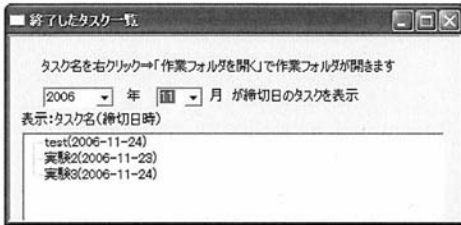


図 6 : 終了したタスク一覧

Fig.6 : A list of finished tasks.

#### 4.3 タスク実行支援

タスクを開始した場合、そのタスクの作業フォルダ内のファイルをすべて開く。タスクを中断した場合、その時点で開かれていたタスクに関係しているウィンドウを最小化し、次のタスクへ移りやすくする。このとき、どのウィンドウを最小化したのかをログファイルへ書き込む。ユーザが過去に中断したことのあるタスクを再開すると、前回中断したときのログから、中断時に最小化したウィンドウをすべて元に戻す。

タスクを開始、中断、あるいは再開する方法は 2 つある。

1. 図 3 内のタスク名をクリックすると出現する「タスク情報ダイアログ」(図 4) 中の「タスク開始・中断」ボタンを押して行う方法
2. 図 3 内のタスク名を右クリックすると出現するメニュー (図 5) 中の「タスク開始・中断」を選択して行う方法

## 5. 試作システムのユーザビリティ評価

### 5.1 実験目的

試作システムの以下の点におけるユーザビリティを主観的に評価することを目的とする。

- (1) ユーザの負担なくタスクとファイルの関連付けが行えるか
- (2) タスク実行支援は有効であったか

### 5.2 実験の構成と方法

被験者はまず今回の実験の目的と内容を記述した実験の概要を熟読した後、以下の作業を行った。

1. タイムラインナビにタスクを登録
2. 登録したタスクにファイルに関連付け
3. 1 つ目のタスクを開始
4. 途中でタスクを中断, 2 つ目のタスクを開始
5. 途中でタスクを中断, 1 つ目のタスクを再開
6. 1 つ目のタスクが終了したら, 2 つ目のタスクを再開
7. 2 つ目のタスクの終了後アンケートの記入

#### 5.2.1 タスクの登録

まず、初めにタイムラインナビに「文書入力タスク」と「グラフ作成タスク」の 2 つのタスクを登録する。

#### 5.2.2 ファイルの関連付け

次に、5.2.1 項で登録した 2 つのタスクに、それらのタスクで使用するファイルに関連付けする。関連付けするファイルは、予め終了したタスクとして登録しておいた「ファイルタスク」という名前のタスクの作業フォルダ内に保存してあるので、その作業フォルダを探して関連付けする。

#### 5.2.3 タスクの開始

ファイルの関連付けを終えたら、実際にタスクを行う。まず、「文書入力タスク」から開始する。

#### 5.2.4 タスクの中断と再開

タスク開始から一定時間後に、タスクの中断を促すメッセージが出現するので、それを確認後タスクを中断する。中断後、「グラフ作成タスク」を開始し、これも同様に中断メッセージが出現するので、中断した後「文書作成タスク」を再開する。

#### 5.2.5 タスクの終了

「文書作成タスク」を再開し、文書を入力し終了したらタスクを終了する。その後、「グラフ作成タスク」

ク」を再開し、それも終了するとアンケートを記入、実験終了となる。

### 5.2.6 アンケート

アンケート内容は以下の通りである。質問ごとに評価の理由を書く自由記述欄を用意した。

1. ファイルの関連付けは容易に行えたかどうか (1: 困難 - 5: 容易)
2. タスクを始める準備として開始時にファイルが自動的に開くことで準備の手間が削減されたかどうか (1: 増加した - 5: 削減した)
3. タスクの中断時、ウインドウが最小化することで次のタスクへ移りやすかったかどうか (1: 移りにくかった - 5: 移りやすかった)
4. タスクの再開時、ウインドウの配置が中断時の状態に戻ることによって容易にタスクが再開できたかどうか (1: 困難 - 5: 容易)
5. このシステムを使用することで、従来のフォルダ構造でのファイル分類より手間が削減されたかどうか (1: 増加した - 5: 削減した)
6. このシステムをまた使いたいと思ったかどうか (1: 使いたくない - 5: 使いたい)
7. その他、気づいた点・感想など (自由記述)

### 5.3 実験環境

被験者は、以前にタイムラインナビを使用したことのある大学生・大学院生 10 人である。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

### 5.4 実験結果と考察

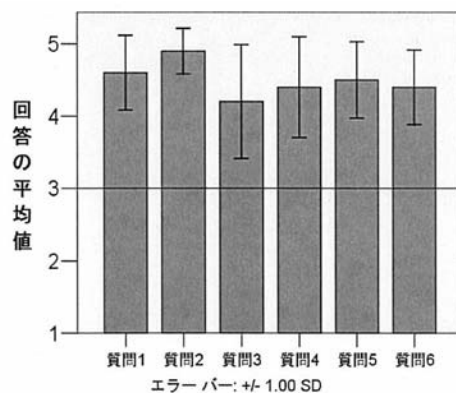


図7: 実験結果

Fig.7: An experiment result.

実験結果を図7に示す。また、5段階評価での中央値である3を評価の基準値として、実験結果の値と基準値でのt検定を行ったところ、すべての質問項目において基準値より試作システムの方が有意に良い結果となった ( $p < 0.05$ )。

質問1の回答の理由には「ドラッグ&ドロップだけでの操作だったので楽だった」という意見が7名と多かった。しかし、「関連付けの際に関連付けできたかできていないかのフィードバックがないのが不安」という意見も6名と多かった。これは関連付けの行為そのものは容易にできたため、平均値は高い値が出たものの、ユーザインタフェースに改善すべき点があることを意味している。

質問2の回答の理由には4名が「ファイルを探す必要がなかったから」、5名が「ファイルの一つ一つ開かなくて済んだから」と準備の手間が削減されたという意見が多かった。しかし、「自動的に開くのはいいが、開かなくてよいファイルまで開くので、少し邪魔だと感じた」という意見もあった。このことから、タスク開始の際の手間は省けているが、開始/再開時点では開かなくてよいファイルを見つける手法を考える必要があることを示している。

質問3の回答の理由には2名が「最小化する手間が省けたから」、2名が「今どのウインドウを使うべきなのかを考えずに済んだから」という意見があった反面、「どちらでもいい」、「次のタスクとは関係がない」、「最小化したときにタスクバーにファイルが多くなるのが嫌」という意見もあった。この質問は被験者の選択のばらつきが大きく、ユーザの考え方の差が大きかった。このことから、ウインドウを強制的に最小化する以外に、

- ・タスクで使用していたファイルのアプリケーションをすべて終了させる
- ・表示されているウインドウをタスクバーではなくタスクトレイにまとめてひとつのアイコンで表示させる。これは、ファイルを閉じるわけではなく、最小化の特殊な形とも言える
- ・何もしない

という3つの選択肢をオプションとして追加する

という改善策が考えられる。

質問 4 の回答の理由には「何をしていた所なのかをすぐに思い出しやすいから」、「中断前の状態が自分の一番使いやすい状態だから」などの意見があった。このことからタスク中断時の状況を再現することによって、タスクの再開を支援することができていることがわかる。

質問 5 の回答の理由には「複数のフォルダに同一ファイルをそれぞれコピーしておく必要がないから」「普段も同様のこと（タスクごとにファイル分類）を行うが、それを自動で行ってくれるから」などの理由があった。このことから試作システムにおいて、タスクによるファイル分類操作を実行でき、使用価値があることを意味している。

質問 6 の回答の理由には「開始（再開）が楽だから」「バージョンを変えて文書を作るときにフォルダごとの管理はやりにくいから」などの意見もあった。一方で、「タスク実行支援の部分は利用したいが、ファイル分類はフォルダ構造の方がよい」「タスクで使用するかどうかが確定しないファイルの関連付けをどうするのか」などの意見もあった。このことから試作システムを使いたいと思う人は多いが、関連付けに関する改善点が存在することも示している。

質問 7 の自由記述には「タスクを始める際、開くファイルを選べたら便利」、「開くファイルの中でも最小化された状態で開くなどを選べたら便利」などの意見があった。このことからタスク開始時のファイルの開き方に改善すべき点が存在することがわかる。

## 6. まとめ

本研究では、フォルダ階層構造を用いた分類のみでのファイル管理ではなく、タスクとその資源であるファイルの関係に基づいたファイル管理を可能にすることで、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たファイル管理手法を提案した。そして、試作システムを実装し、そのユーザビリティを実験により評価した。その結果、試作システムを用いることでタスクでのファイル分類が可能となった。また、ユーザに柔軟なフ

イルアクセス手法を提供でき、ユーザのタスク実行支援も行うことができることがわかった。

以下に、今後の課題を示す。

- 4.2.1 や 4.2.2 で述べた現状未実装部分の実装
- タスクにファイルを関連付けする際のユーザへのフィードバックの表示
- タスク開始時のファイルの開き方の指定
- タスクの中断時の中断方法の選択
- 質問 6 の結果における、関連付けに関する改善点の改善方法の発見と改良
- 上で述べたこれらの機能の追加や改良を加えたシステムをユーザが長期的に使用した上でのユーザビリティの評価

## 参考文献

- [1] Google : “Google デスクトップ” ; <http://desktop.google.com/> (2008/07/07).
- [2] Microsoft: “Windowsデスクトップサーチ” ; <http://www.microsoft.com/japan/windows/desktopsearch/default.mspx> (2008/07/07).
- [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管理スキーム”, 情報処理学会研究報告, 2008-HCI-129 (2008).
- [4] Stephan Bloehdorn, Max Volkel : “TagFS - Tag Semantics for Hierarchical File Systems”, WWW2006, May 22-26, 2006, Edinburgh, UK. (2006).
- [5] 堤 大輔, 倉本 到, 渋谷 雄, 辻野 嘉宏: “空き時間とタスク間関係を利用したユーザのスケジューリング支援手法”, 情報処理学会論文誌, Vol.48, No.12, pp. 4064-4075 (2007-12).