

# 組織知識継承のための共有フォルダからの 活動単位抽出アプローチ

齊藤 典明<sup>1,a)</sup> 金井 敦<sup>2</sup> 谷本 茂明<sup>3</sup>

受付日 2015年4月9日, 採録日 2015年10月2日

**概要:** 長期間にわたって継続的に組織活動を行うには, 組織構成員が替わっても蓄積してきた知識を活用してゆく必要がある. 現在多くの現場組織では組織活動の記録として共有フォルダに資料を蓄積し, 組織の知識として活用している. しかしながら, 共有フォルダを長期間運用することにより蓄積資料は膨大になり資料の全体像の把握が困難なことや, 必要な資料を探し出すのが困難になり, 蓄積された知見を組織知識として十分に活用できないという問題がある. この問題を解決するために, 資料作成のタイムスタンプから得られた時間情報と, 共有フォルダにおけるフォルダの命名方法に着目して抽出した活動単位に基づき蓄積情報を分類整理し, 組織知識として活用することを支援するカレンダー型インタフェースの提案を行った. さらに, 実データによる方式検証, 蓄積資料の探索シーンを想定したシミュレーション評価を行い提案方式の有効性を確認した.

**キーワード:** 情報共有, 知識共有, 知識継承, 組織知識

## Activity Series Extraction Approach from Shared Folder for Organizational Knowledge Inheritance

NORIAKI SAITO<sup>1,a)</sup> ATSUSHI KANAI<sup>2</sup> SHIGEAKI TANIMOTO<sup>3</sup>

Received: April 9, 2015, Accepted: October 2, 2015

**Abstract:** To keep organizational activities continuously during long term, even if someone of the organization is moved out, his experienced knowledge should be stored in the organization. To achieve the aim, current organizations are storing many files in the shared folder as organizational knowledge. However, the stored files become too large quantities by using shared folder long term, then some problems will be happened such as: we cannot grasp of the whole of the shared folder, finding necessary files is difficult, and stored files are not used enough as organizational knowledge. To solve such problems, stored information are classified by using time information from the file timestamp and extracted activity series based on the folder naming pattern, the calendar type interface which enables the classified information to be used as organizational knowledge is introduced. Then, the effectiveness of our approach is shown by using actual sample data and a simulation test.

**Keywords:** information sharing, knowledge sharing, knowledge inheriting, organizational knowledge

<sup>1</sup> 日本電信電話株式会社  
Nippon Telegraph and Telephone Co., Musashino, Tokyo  
180-8585, Japan

<sup>2</sup> 法政大学  
Hosei University, Koganei, Tokyo 184-8584, Japan

<sup>3</sup> 千葉工業大学  
Chiba Institute of Technology, Narashino, Chiba 275-0016,  
Japan

a) saito.noriaki@lab.ntt.co.jp

### 1. はじめに

組織活動を行ううえで組織内に蓄積されている知識を活用することは重要なことである. 組織における知識とは, 教科書やマニュアルに記載されているような定式化されているものよりも, 組織内の相互作用で解釈され, 記載されている内容をどのように活用すればよいのか, という実

実践的な知識が重要である [1], [2]. たとえば, ソフトウェアの開発標準があったとき, それをそのまま推進するだけでは良いソフトウェアは開発できない. 開発標準で規定されている範囲で, その時々状況に応じて柔軟に対処する必要がある [3], [4]. このような組織内の実践的知識は, 組織内でトップダウン的に広まるものではなく, むしろボトムアップ的に広がっている. さらには「場」として組織活動の中に積極的に取り入れてゆく必要がある [5].

組織活動を長期にわたって持続してゆくには, このような実践的な知識をとらえて確実に組織内で継承してゆく必要がある [6], [7]. このような組織の知識の蓄積と継承は, 組織メンバが安定し, かつ, 定期的にメンバが出入りする場合において, その時々組織メンバから口伝で継承してゆくことも可能である. しかしながら, 現在は協働環境が多様化しているため必ずしも口伝では継承できない. 特に現在は, 様々な資料が電子ファイルとして ICT 環境上に点在しているため, ICT を活用することで組織の知識の継承が効果的になると考えられる.

一般に, 組織活動の中で生成される電子ファイルは, 電子メールや共有フォルダで共有される. 電子メールで共有された資料は配布された各自の所蔵になるが, 共有フォルダの場合は後々のメンバでも参照可能で, 再利用性の高い電子ファイルとなる. 共有フォルダの利用は多くの組織で定着しているため, 我々は, 共有フォルダを有効活用することで組織の知識の蓄積と継承を支援する方法を検討してきた [8], [9].

これまでの検討では, 組織活動の例題として研究開発組織を対象に調査を進めた. ここでは, 資料を蓄積する活動については 10 年以上にわたり定着しているが, 活用については過去の資料を十分に引き出せていないという問題が明らかになった. さらに, 共有フォルダ内に蓄積された過去の資料を調査した結果, 組織活動のイベントと紐づくものが多いことが判明した. そこで, 組織内のイベント情報を管理しているスケジューラと連動して過去の資料をカレンダー型インタフェースで引き出せる方式を実現し, 実用性を検証した.

実用性の検証では, 単独の活動の 4 年分の資料に対して実施し良好な結果を得た. ただし, 既存の情報環境や組織文化を考慮したより実用的な環境に拡大するには, イベントに紐づく時間情報の取得方法と, 複数の活動が混在した状態で大量の情報を効果的に分類できる方法が必要であることも判明した.

そこで今回の検討では, カレンダー型インタフェースをより実用的な環境に拡大する方法を検討した. ここでは, ファイル種類を特定したうえでタイムスタンプから時間情報を取得する方法と, 共有フォルダの命名方法に着目してフォルダ名を整理することによって蓄積資料を分類できる方法を提案する. これによりファイル探索時のユーザ負荷

を軽減できるため, 共有フォルダを組織知識として活用しやすくなることを検証する.

## 2. 共有フォルダによる組織知識と関連研究

研究開発の現場において, 競争力のあるコア技術に基づいて継続的な研究開発を続けてゆくには, 着想からプロダクト化までの開発期間, 開発したプロダクトの維持管理期間, 特許の維持期間などにおいて 10 年スパンの対応が必要になる. 一方, この間には組織内ではメンバの出入りや組織の改変もあり, 組織内に蓄積した知識が忘失される可能性もある. サステナビリティを意識した場合, このような状況を乗り越えて持続的な組織活動を創り出すことが重要になる.

組織的な研究開発活動では, 組織活動において様々な資料が作成され, 活用し, 組織活動の記録として共有フォルダに蓄積されてゆく. 新しいメンバは共有フォルダ内の資料を活用することにより, 組織内で蓄積した知見に基づき新しい研究開発活動ができる. このようなことから, 共有フォルダは組織知識を共有・継承するための「場」としての役割がある. しかしながら, 共有フォルダ内に資料を蓄積する活動は定着しているものの, 共有フォルダから自発的に資料を取り出し活用するにはいくつかの障壁が残っている. 具体的には, 共有フォルダの運用が長くなれば蓄積資料も膨大になり, 共有フォルダ内にどのような資料が蓄積されているのか分からない, 目的の資料が見つからない, あるいは目的の資料を見つけるために時間がかかるなどの問題がある.

本研究ではこのような状況を想定し, 10 年以上の長期にわたって共有フォルダを運用することで組織知識を蓄積・継承できる基盤の実現を目指す. 具体的には, 蓄積資料の全体像を把握することができる仕組みと, 目的の資料をより手軽に探し出せる仕組みの実現を目指す.

大量に蓄積された共有フォルダから資料を探し出すためのアプローチとして, (i) あらかじめ定めた分類方法に従って資料を蓄積し分類に従って取り出す, (ii) 資料が必要になった時点でキーワード検索によって探し出す, (iii) 動的な分類により取り出すという 3 つのアプローチが考えられる.

(i) に関しては, 網羅的かつ体系的な分類方法である 10 進分類 [10] や LATCH 法 [11] などの伝統的な手法がある. 10 進分類は幅広い分野を網羅的に分類する図書に対して適しているが, 日々身の回りで生成される電子ファイルは, 特定の分類に偏り, 分類として機能しないために適していない. LATCH 法はいくつかの分類方法の総称であり位置 (Location), アルファベット順 (Alphabet), 時間 (Time), 分野 (Category), 階層 (Hierarchy) に基づいてコンテンツを整理する方法であり Web コンテンツのデザインで用いられている. 共有フォルダに対する標準的な設計方法は

ないが、かわりに Tips 的なものは多数存在する [12]。Tips には、管理組織ごとや時系列ごとに分類する方法があげられている。また、探索を効率化するために記号を用いる方法や、フォルダの階層に上限を持たせる方法なども含まれる。しかしながら、従来から事前の分類ではうまく探し出せないことが超整理法 [13] などで指摘されている。

(ii) の資料が必要になった時点でキーワード検索によって探し出す方法に関しては、オペレーティングシステム上の機能として用意されているフォルダ内のファイル検索機能 [14] や Google デスクトップやそれに代わる検索ツール [15] によって蓄積されているファイルを探し出す方法がある。キーワード検索では思ったキーワードでなかなかヒットしない、キーワードでヒットしてピンポイントで資料は探せるものの資料の全体像を把握しにくいなどの問題がある。特に、これらは情報蓄積時と情報利用時で「手がかりに関する意識のずれ」[16] や、「利益の不均衡」[17] など情報共有における非対称性がおこり、目的の情報を探し出すことが難しくなるからである。ここで「利益の不均衡」というのは、情報利用を容易にするために情報蓄積時に多大な労力を払う、あるいは情報蓄積を容易にすることにより情報利用で多大な労力を払う、という意味である。

(iii) の動的な分類により取り出す方法には、以前から提唱されている時系列により資料の管理を行う超整理法 [13] や、時系列で資料を管理する概念を発展させ、ユーザのコンピュータ上の作業を再現することにより過去のファイルを探す方法である Time-Machine Computing [18] や、ユーザのファイルに対する体験をもとにファイルを探す出自検索 [19] が考案されている。これらはユーザの記憶を頼りに時間軸を中心にファイルを探す方法である。さらに、時間軸に加えてファイルのテキスト情報とファイルアクセス情報を組み合わせることによりファイルどうしのつながりを抽出することによりファイルの所在を明らかにする方法 [20] などが考案されている。

長年運用された共有フォルダは、混沌としてしまうという問題がある。共有フォルダは複数人で利用するため、一定の整理がされていたとしても、どこにどのようなファイルあるのかについては蓄積した本人しか分からなくなるためである。混沌としてしまった共有フォルダでは、ユーザが必要とするファイルあるいはファイルにたどり着くための操作量が膨大になる。この操作量がユーザの負荷となる。そこで本研究では長期間運用されている共有フォルダの利用を活性化することを目指し、(iii) の動的な分類により取り出す方法のアプローチの延長線上として、ユーザの負荷を低減させつつ目的のファイルを探しやすくする分類構造を検討する。ここでは、資料作成のタイムスタンプから得られた時間情報と、フォルダの命名方法に着目して抽出した活動単位に基づき蓄積情報を分類整理することで、共有フォルダを組織知識として活用するカレンダー型イン

タフェースを実現する。なお、活動単位とは一連の資料が作られる活動の区分と定義する。

### 3. 時系列に基づいた共有フォルダ活用

共有フォルダ内の資料を動的な分類により取り出すアプローチを実現するにあたって、カレンダー型インタフェースによる提案方式について述べる。

#### 3.1 スケジューラ連動型共有フォルダにおける課題

先の検討 [8] の中で、共有フォルダ内に蓄積する資料は組織活動のイベントと紐づくものが多いことが判明した。この結果を基に、組織内のイベント情報を管理しているスケジューラであることに着目し、スケジューラと連動した共有フォルダを提案した [9]。このスケジューラと連動した共有フォルダの活用 (図 1) では、年と月で構成されるカレンダー形式のセルの中にファイルへのリンクが埋め込まれたファイル名を表示した。この表示形式でファイルの有無を確認することで活動の流れが把握できた。この効果は、特に複数年同時表示したカレンダー形式で高まる。しかしながら、これらの効果は比較的小さな活動単位で確認したため、より一般的な規模の環境で適用するためには、次の 3 つの課題があることが判明した。

- (1) スケジューラに依存する制約の解消
- (2) 大量の既存資料の活用方法
- (3) 活動単位が混在した状況での運用方法

1 つめの「スケジューラに依存する制約の解消」は、スケジュールソフトは個人や組織の都合で選択される。スケジュールの選択が資料蓄積の障害になることは好ましくないため、任意のスケジューラと連動できることが必要になる。また、多くのスケジューラが iCalendar 形式 [21] のデータで連動することができるものの、資料の共有という観点では機能が十分ではないため、様々な拡張が必要になる。

2 つめの「大量の既存資料の活用方法」は、システムの導入に際して過去の情報資産が活用できないことは大きな障害になる。すでに大量に蓄積されているファイルをカレンダー形式で表示するには、蓄積されているファイルから時間情報を取得する必要がある。

3 つめの「活動単位が混在した状況での運用方法」は、



図 1 カレンダー型インタフェースの活用  
Fig. 1 Calendar type interface for shared folder.

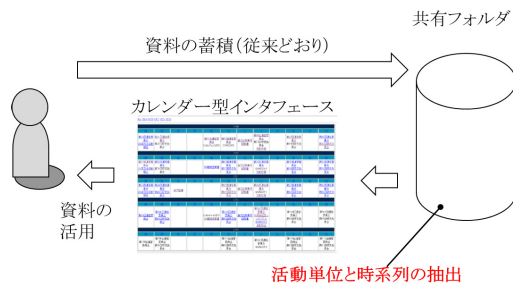


図 2 既存フォルダへの適用

Fig. 2 Calendar type interface for existed shared folder.

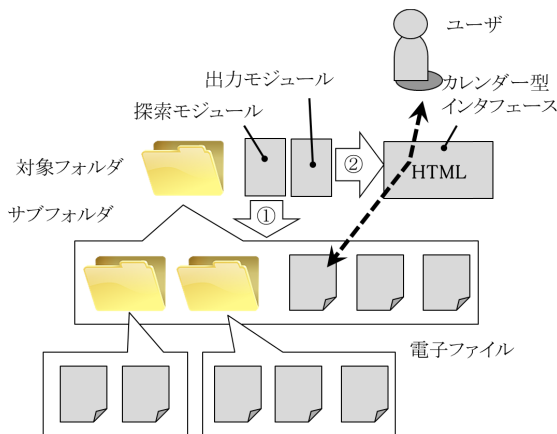


図 3 システム構成

Fig. 3 System structure.

組織やユーザの活動の中には複数の連続した活動が存在する。共有フォルダではこれらが混在して混乱してしまっている。資料を活用する観点ではこれらが必要に応じて分離できる必要がある。

### 3.2 解決アプローチ

これらの課題を解決するアプローチとして、1つめの課題については、スケジューラそのものではなく、スケジューラで用いられるカレンダー型のインタフェースを活用することとした。これによって、スケジューラの制約から解放される。その結果、情報の蓄積は従来どおりの共有フォルダで行い、資料の取り出しでは情報活用に必要な機能や表示方法などをカレンダー型インタフェースの中で実現できる(図2)。実現にあたって、蓄積資料をカレンダー型インタフェースの中で表示するにはいくつかの情報が必要となるため、これらは共有フォルダへのクローリングによって取得した。取得した情報を、JavaScriptを含むHTML形式のファイルに入力し、これをWebブラウザで操作することによりファイル探索とファイル取得を実現するカレンダー型インタフェースとした(図3)。

2つめの課題については、カレンダー型のインタフェースに既存資料を表示するためには、資料の時間情報が必要になる。そこで、一定の制約のもとファイルのタイムスタンプから自動的に時間情報を取得した。これにより大量の



図 4 ファイルの所在表示の例

Fig. 4 Operation example at the calendar type interface.

既存の資料をそのまま活用できるようになる。

3つめの課題については、様々な活動が混在した組織活動の中から活動単位を識別する方法を実現し、活動単位に基づいてファイルを取り出すアプローチとした。ここでは、共有フォルダにファイルを蓄積する場合、ファイルに対するメタな情報はフォルダ名として蓄積している。フォルダ名の命名方法の特徴をふまえて階層上のフォルダ名から活動単位を抽出する手法を実現した。次節で上記のそれぞれについて述べる。

### 3.3 カレンダー型インタフェース

スケジューラと連動した共有フォルダの方式検証において、蓄積資料をカレンダー形式の時系列で確認できることは活動の流れが把握できること、業務の引継ぎという観点では共有フォルダよりもユーザの負荷が軽減されることが理論的に確認された[9]。特に、活動の流れの把握は複数年表示できることにより効果が大きくなる。反面、資料が多くなりすぎると様々な活動単位が混在してしまい把握できなくなるという問題点も明らかになっている。具体的には、個々のファイル名が多数表示されるだけでなく、一般的なPCの画面の表示範囲を超えるほどになり、可読性が大きく低下した。

そこで、カレンダー形式で蓄積資料を確認するインタフェースの改善として、月単位のセルの中にファイル名を直接表示していたものをやめ、いったんファイル数を表示することとし(図4-A)、セルを選択した際にファイル名一覧を表示する形式とした(図4-B)。また、複数の活動単位を分離して見せる方法は、活動単位ごとに分けたものをカテゴリとし、カテゴリ単位でファイル数を表示する方法とした(図4-C)。なお、カテゴリの作成方法については別途述べる。また、カテゴリ単位についても表示・非表示の切替えができるようにしたため、1年分の表示数が小さくなる。これにより、一般的なPCの画面の表示範囲で複数年同時に表示することが可能となる(図4-D)。

このような機能の導入により、複数の活動単位が混在し、資料が大量になっても複数年にまたがって活動の流れが把

表 1 クローリング対象ファイル  
Table 1 File types and investigation target.

種別	概要	対象
ドキュメント ファイル	組織活動で生成される資料の ファイル	○
隠しファイル等	OS やアプリケーションが使用 するファイル (ini ファイルなど)	×
自動更新 ファイル	OS やアプリケーションの動作で タイムスタンプが更新される ファイル (eml ファイルなど)	×
ダウンロード ファイル	ネットワーク経由などで取得した ファイル	×
展開ファイル	アーカイブ形式で取得し、 その後展開したファイル	×
データファイル	画像ファイルなど	×
プログラム ファイル	exe ファイル, スクリプトファイル など	×

○ : 対象とする      × : 対象としない

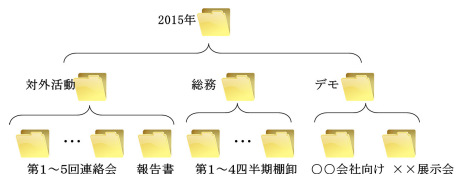


図 5 共有フォルダ例

Fig. 5 Example of shared folder.



図 6 カレンダー型インタフェースによる表示例

Fig. 6 Example of display at calendar type interface.

握できるカレンダー型インタフェースを実現した。

### 3.4 作成資料に限定した時間情報の抽出

次に、すでに大量の資料を蓄積している共有フォルダに対してカレンダー型インタフェースを適用するために、蓄積資料の時間情報を取得する方法を検討した。蓄積資料である電子ファイルの時間情報を取得する方法として、電子ファイルのタイムスタンプを用いる方法がある。UNIXベースのファイルシステムの場合、タイムスタンプには最終更新日時と最終アクセス日時がある。このうち最終更新日時を用いることとする。ファイルのタイムスタンプの取得は共有フォルダへのクローリングによって実施する。このとき、すべてのファイルのタイムスタンプを取得すると、「組織内で作成された資料を探す」という観点では不適切なものが多数存在する。

たとえば、OS やアプリケーションが自動的に生成するファイルやタイムスタンプを更新するファイル、ネットワークからダウンロードしてきたファイルは、ファイルを作成し利用した日付とは異なるタイムスタンプになる。圧縮形式のダウンロードファイルを展開したファイルは、利用した日付で管理したいところが作成した日付になる。このようにデータファイルやプログラムファイルなどは、資料を探すという目的には必ずしも適していないファイルである (表 1)。

そこで本検討では、「組織内で作成された資料を探す」ことに着目し、対象を絞ってファイルのタイムスタンプを抽出することとした。具体的には、MS-Office 系のファイルや PDF ファイルなど、対象組織内の資料作成で使用するファイルタイプに限定してタイムスタンプを取得する。

なお、利用用途によっては写真などのデータファイルやプログラムを対象にする場合も想定されるが、写真やプログラムを取り巻く活動はもう少し別の議論も必要になるため、ここでは検討の対象外とした。

### 3.5 活動単位抽出によるカテゴリ生成

次に、様々な活動が混在した組織活動の中から活動単位を抽出するためのカテゴリ生成手法について述べる。共有フォルダにおける情報の蓄積において、蓄積資料は資料の位置づけや内容を表す適切なサブフォルダを選択しながら蓄積する。そのため共有フォルダのフォルダ構造から活動単位となるカテゴリを抽出する方式とした。

#### 3.5.1 単純抽出手法

活動単位を抽出する単純な方法として、階層構造のフォルダ名をすべて連結したものをカテゴリとした。このとき、カテゴリ数は階層構造のフォルダ名の数だけになり、カテゴリ名もまた長くなり、判読が難しくなるためユーザの負担になる。たとえば、図 5 のような第 1 階層が年、第 2 階層が分類、第 3 階層が各資料の説明を表すサブフォルダになっている共有フォルダ構造があったとき、階層構造のフォルダ名を単純連結したカテゴリ分けは図 6 のようになる。ここでは、階層構造のフォルダ名を単純連結したカテゴリ一覧が左側表示される (図 6-A)。それぞれのカテゴリについて、ファイルのタイムスタンプで指定される月ごとのセルの中に、該当するファイルの個数が表示される (図 6-B)。

3.5.2 フォルダ名分類による活動単位抽出手法

階層構造のフォルダ名をカテゴリ名としてそのまま使った方法では、単語長が長くなるため可読性が悪くなる。また、カテゴリとしての分類が、共有フォルダ上の分類単位と同じため、分類方法が改善されたわけではない。そこで、より適切な分類方法を実現するために、階層構造のフォルダ名からカテゴリ生成する手法を提案する。

フォルダ名からのカテゴリ生成手法を検討するにあたり、インターネット上で公開されている共有フォルダの整理方法 10 種類 [12] と組織内の共有フォルダで 10 年以上にわたって数十人により命名された約 4,000 個のフォルダ名を確認し、フォルダ名の命名方法の特徴を分類した。

1 つめは、公開されている共有フォルダの整理法 10 種類中 5 種類で推奨されているのが「時系列」を表す単語である。具体的には、年や月だけでなく四半期などの周期的な単語、第 1 回などのような順番を表す単語が含まれる。

2 つめは、公開されている共有フォルダの整理法 10 種類中 8 種類で推奨されているのが「分類方法」を表す単語はである。具体的には、組織名や個人名、分類方法などの書かれたフォルダ名である。

3 つめは、公開されている共有フォルダの整理法 10 種類中 4 種類で推奨されているのが、「注釈や記号」を表す単語である。これは、フォルダやファイルに対するアクションを注釈として記載することや、共有フォルダ中のサブフォルダが適切な順番で並ぶような整理番号や特別な意味の目印を付与した場合である。たとえば、注釈では「○月○日メ切り」や「後日整理する」など、記号では「001」や「★」などがフォルダ名に付与されている場合である。

このほかに、実際の共有フォルダを上記で分類してみるといずれでも分類できないフォルダが多数出現する。これらは公開されている 10 種類の共有フォルダの整理法においては命名方法が自由となるフォルダ名である。実際の 4,000 個のフォルダ名に対して、いずれの分類にも入らなかったフォルダ名の特徴を確認したところ、蓄積資料の内容、または資料を用いたイベントを表す単語が用いられていた。そこで、これらを、「内容・イベント」という分類にした。

以上をまとめると、共有フォルダに用いられるフォルダ名の特徴は表 2 のように分類できることが分かった。

これらの分類のうち「時系列」を表す単語は、カレンダー型表示における表示位置になるため、資料の分類としては不要である。「分類方法」を表す単語は、資料蓄積時に有用な情報であるが、資料を探す際には必須ではない。「注釈・記号」についても共有フォルダ上で資料を探すための目印であり、カレンダー型表示では必須ではない。よって、残りの「内容・イベント」を表す単語を活動単位を表すカテゴリとして抽出した。

次に、抽出単語について、資料を分類するうえで適切に

表 2 フォルダ名の分類方法

Table 2 Folder classification method based on the naming.

命名方法	具体例	記述※1	対象
時系列	年, 日付, 時期, 順番 など	5 個	×
分類方法	組織名, ロケーション名, 担当者名, 整理方法 など	8 個	×
注釈・記号	資料への注釈や整理上の記号など	4 個	×
内容・イベント	資料内容の性質や使われた活動を名称にしたもの	0 個※2	○

※1 公開されている整理方法 10 種類中の記述個数

※2 規定された記述は 0 個, 10 種類とも自由記述

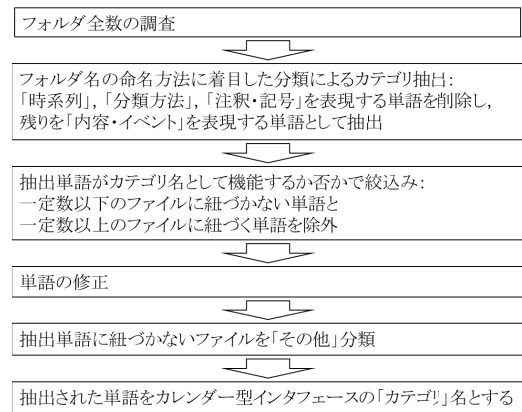


図 7 カテゴリ生成手法

Fig. 7 Category generation method.

機能するか否かを評価して絞り込んだ。つまり、ある単語で分類できるファイルが少なすぎる場合と多すぎる場合は、どちらも資料を分類する単語としては機能していない。一定数のファイルを分類できる単語だけを抽出した。

以上は、半自動的に単語を抽出することができるが、機械的な処理により判読不明な文字列や記号が残ることがある。そこで、この段階で抽出単語の修正・削除などを行う。最後に、抽出されたすべての単語に紐づかないファイルも出現するので、これらのファイルを「その他」の分類とする。このようにして得られた単語と分類をカレンダー型インタフェースのカテゴリ名とする (図 7)。

先に例示した図 5 の共有フォルダに一連の処理を加えたカレンダー型インタフェースによる表示例を図 8 に示す。図 6-A と図 8-A を比較すると、カテゴリ数が集約されていることと、各カテゴリの文字数が複数フォルダ名を連結したものから単独のフォルダ名程度に短くなること、の違いがある。

また、共有フォルダにおけるフォルダ構造ではファイルの所在が一意に決まるが、今回のカテゴリによるファイルの分類では、1 つのファイルが複数のカテゴリに属する可能性がある。そのため、複数のカテゴリから目的のファイルを探ることができるため、目的のファイルに到達する可能性が増加する。

図 8 カテゴリ生成後の表示例

Fig. 8 Example of display at improved calendar type interface after.

#### 4. 実データによる方式検証

次に、様々な資料が長期にわたって大量に蓄積されている共有フォルダを例題に、提案手法の方式検証を行った。

##### 4.1 実験対象

今回、研究所内の研究グループで継続的に 10 年以上にわたって運用され、資料の蓄積は定着しているものの資料の所在が属人化してしまっている共有フォルダを例題に、提案手法を適用し、方式検証を行った。

この組織では、資料の蓄積については活動が定着している。また、共有フォルダからの資料の活用についても一定の効果を発揮している。しかしながら、資料の所在は属人化しているため、資料が必要になるつど、資料の所在情報を電子メールなどでやりとりしながら活用している。そのため、どこにどのような資料が蓄積されているのか、という蓄積資料の全体像を把握するのが難しい。あるいは、資料が蓄積されていることを漠然と把握しているものの、目的の資料を探し出すには時間がかかるという問題を持っている。

そこで、提案手法により共有フォルダに蓄積されている資料の全体像を容易に把握でき、必要としている資料を短時間で探し出せるようにすると、共有フォルダの活用がさらに促進されると考えられるため、提案手法の方式検討の対象とした。

##### 4.2 実施結果

例題の共有フォルダは、第 1 階層が整理番号付きでカテゴリ分類されている、第 2 階層以下は、必要に応じて自由に命名してフォルダを作成している。さらに、組織メンバーそれぞれの利用者が必要に応じてサブフォルダを作成しているため、フォルダ名もサブフォルダの深さもまちまちである。

2015 年 2 月時点での共有フォルダ内の資料の蓄積状況と共有フォルダすべてのファイルに対してタイムスタンプを取得した結果を表 3 上段に示す。これに対し、提案手法であるドキュメント系ファイルだけに絞ってタイムスタンプ

表 3 共有フォルダへのクローリング実施例

Table 3 Investigation example at actual sample shared folder.

手法	対象	ファイル数	フォルダ数	最大階層	最古年
従来	全て (102GB)	46,973 個	8,885 個	16	1980 年
提案	資料のみ	21,341 個	4,232 個	14	1997 年

表 4 カテゴリ生成実施例

Table 4 Category generation example at actual sample shared folder.

処理概要	結果
クローリングで収集したフォルダ数	4,232 個
「内容・イベント」を表す単語の抽出数	390 個
抽出単語のうち 0.5%未満と 20%以上のファイルを包含する場合を除外した単語数	112 個
最終カテゴリ数 (「その他」1 個を含む)	106 個
「その他」に分類されたファイル量	7%

プを取得した結果を表 3 下段に示す。

実験した共有フォルダでは、蓄積されているファイルすべてに対して一律にタイムスタンプを取得すると、組織の存続期間に対して明らかに不自然な日付のファイルが出力された。内容を確認したところ、インターネットなどからプログラムのソースコードをダウンロードして展開したものであった。ほかに、プログラムが出力するような隠しファイルなどを除外し、組織活動で生成された資料に絞ってタイムスタンプを取得した。その結果、対象となるファイル数は半数以下の 21,341 個になり、それぞれが属するフォルダ名をリストアップすると 4,232 個になった。フォルダの階層については 16 段から 14 段とわずかに減少した。ファイルのタイムスタンプについても妥当な期間となった。

次に提案手法によるカテゴリ抽出結果を示す (表 4)。処理の初期値としてのフォルダ名は 4,232 個になる。このうち「内容・イベント」を表す単語の抽出数は 390 個であった。これらの抽出単語に対して、どの程度のファイルを包含するかを算出した。分類として少なすぎずかつ多すぎない程度として、全体の 0.5~20%のファイルを包含する単語だけに絞った結果、112 個抽出された。これまでの処理は、プログラムによるパターンマッチを繰り返す処理で実現した。このような機械的に処理をした結果、文字列として不適切なものを含んでいる場合があるため、最終調整として目視で確認し修正を加えた。今回の事例では、明らかに不適切な語として 7 個程排除し、抽出単語は 105 個が残った。これに「その他」のカテゴリを加えた 106 個を最終カテゴリ数とした。「その他」のカテゴリに含まれるファイル数を確認した結果、全体の 7%であった。

ここで、「その他」に含まれるファイルの割合は、抽出単語によるファイル包含量の閾値を狭めれば多くなり、ファ

表 5 特徴比較

Table 5 Comparison between shared folder approach and calendar type interface approach.

	従来方式	提案方式
対象	・全てのファイル ・ファイルの蓄積と利用	・作成資料に対象を絞る ・ファイルの利用のみ
入口	・第1階層フォルダ数	・年数分のカレンダー
操作	・第1階層から順番にフォルダを辿る	・年, カテゴリ, 月の順番に選択
時間情報	・フォルダ名かファイル名に記載してある場合, またはファイルの属性情報	・ファイルのタイムスタンプから自動抽出し, カレンダー型インタフェースで明示
カテゴリ	・複数のフォルダ名の組み合わせ ・全てのファイルがいずれかに属す	・内容やイベントを表す単語に限定 ・いずれにも属さないファイルが存在

イル包含量の閾値を広めれば少なくなる。そのため、「その他」カテゴリに含まれるファイルを適度に少なくするためには、抽出単語によるファイル包含量の閾値の調整を行う必要がある。今回は抽出単語によるファイル包含量の閾値が0.5%~20%であり、「その他」は7%であり閾値の範囲内であることから、抽出単語と「その他」のバランスが適切であると判断した。

以上の結果として、全フォルダ数4,232個から最終カテゴリ数は106個になった。そのため、共有フォルダ名のうち、ファイルを分類するうえで有効な単語は約1/40程度であると判断できる。

### 4.3 方式検証結果の考察

次に、提案方式であるカレンダー型インタフェースの特徴を、従来方式である共有フォルダとの対比で考察する(表5)。

まず対象とするファイルと操作について述べる。共有フォルダでは、すべてのファイルに対して蓄積と利用の両方を同じインタフェースで操作することができる。これに対し、カレンダー型インタフェースは、資料の利用に特化したインタフェースである。

資料への入り口は、共有フォルダの場合は第1階層のフォルダ群であるのに対し、カレンダー型インタフェースでは蓄積した年数分のカレンダーになる。

操作の流れは、共有フォルダの場合は第1階層からフォルダ名を確認しながら順番に深い階層をたどってゆく。カレンダー型インタフェースでは、最初に年を指定し、該当年にファイルの蓄積があったカテゴリ名が表示されるのでその中からカテゴリを選択する。カテゴリ指定の後は月別

表 6 数値比較

Table 6 Difference of measurements between two types.

	従来方式	提案方式
ア	46,973 個	21,341 個
イ	49 個	19 個
ウ	最大 15 階層・平均 5.1 階層	3 階層
エ	8,725 個	106 個
オ	末端フォルダ数=7,357 個	末端セル数=2,874 個
カ	平均 6.4 個	平均 7.5 個
キ	なし	7%

- ア. 対象ファイル数
- イ. 入口の個数
- ウ. 探索する深さ
- エ. 一意なフォルダ名またはカテゴリ名の個数
- オ. カテゴリの分岐数
- カ. 最小区分内のファイル数
- キ. フォルダまたはカテゴリに属さないファイルの割合

展開されたセルの中からファイルの含まれるセルを選択する。

資料の時間情報が、共有フォルダのフォルダ名やファイル名に記述してある場合、当該セルを確認することで知ることができる。そうでない場合は、ファイルの属性データを見て知ることになる。このように、カレンダー型インタフェースでは、カレンダー表示の中で明示的に表示される。

どのような活動で生成された資料であるのかを知るためのカテゴリ情報は、共有フォルダでは、ファイルが属する階層構造の複数のフォルダ名を確認することで判断する。カレンダー型インタフェースでは、ファイルが属する階層構造の複数のフォルダ名から抽出された「内容・イベント」を表す単語によるカテゴリ名で判断する。なお、カレンダー型インタフェースでは、いずれのカテゴリにも属さない「その他」分類のファイルがある。

次に、このような特徴に対して、実際にどのような差異が出たかを数値を用いて述べる(表6)。まず、対象としたファイル数は、カレンダー型インタフェースでは元の共有フォルダの約半分程度であった。蓄積資料の入り口となる分類は、共有フォルダは第1階層の分類が49個であるのに対し、カレンダー型インタフェースは共有フォルダの存続期間となり19個である。入り口からファイルを蓄積しているフォルダにたどり着くまでの流れは、共有フォルダの場合は第1階層から順番にサブフォルダを選択し、最大15階層となる。平均は5.1階層であった。これに対して、カレンダー型インタフェースではつねに3階層である。

共有フォルダにおいて重複を除いた種類は8,725個であった。もとのフォルダ数が8,885個であったので、重複したフォルダ名というのは8,885-8,725=160個で、これは全体の1.8%のため非常に少なかった。これに対して、カ



レンダー型インタフェースではカテゴリ数が106個である。そのため、共有フォルダでは多くのフォルダ名でファイルを分類しているのに対して、カレンダー型インタフェースでは少ない個数のカテゴリ名で分類していることになる。

このフォルダ名およびカテゴリ名による分類数は、共有フォルダの場合は最下層である末端フォルダ数で7,357個である。カレンダー型インタフェースの場合は、各年のカテゴリ数の合計で583個であった。また、カテゴリごとに月別展開されたセルのうちファイルのある末端セル数は2,847個であり、これがカレンダー型インタフェースにおける末端フォルダ数に相当する。このため、ファイル総数を末端フォルダ数で割った平均ファイル分割数は次のようになる。すなわち、共有フォルダの場合は平均6.4個で、カレンダー型インタフェースでは7.5個である。この差は1個程度であり、大きな差異はなかった。

以上のことから、従来方式である共有フォルダでは入り口となる第1階層は分類であり、入り口が運用年数となるカレンダー型インタフェースに比べ多い。これは一般的にも、第1階層をなんらかの分類にした場合、分類数が運用年数を下回することは少ないと考えられるため、共有フォルダの方が入り口の数が多くなる。さらに、共有フォルダはフォルダ階層も自由に作れるため深くなる。そのためファイルの探索において、探索ルートを入り口付近で間違えると正しいルートに近づくまでに時間がかかる構造になっている。また、全体を把握するためには多数のフォルダを開いてゆく必要があると同時に、その間に多くのファイル名を認知してゆく必要がある。

これに対してカレンダー型インタフェースでは、入り口が運用年数の個数であり、ファイル探索は、年、カテゴリ、月の順番で探すため、探索ルートは3階層の一定長である。このことから、入り口の数少なく、探索ルートも短い。そのため、ファイル探索を繰り返すのが容易な構造になっている。

さらに、カレンダー型インタフェースにおけるカテゴリ数は共有フォルダのフォルダ名より少なく、カテゴリ名も元のフォルダ名より短く簡潔になる。また、共有フォルダにおけるファイル探索では、フォルダを行き来するのにすべてフォルダ名という文字の解釈を行いながらファイルを探る。一方、カレンダー型インタフェースでは、カテゴリ名という文字の解釈の後に、年と月の時系列で表現されたカレンダー上の位置による指定でファイルを探る。このことから、レンダー型インタフェースの方がカテゴリ名である単語の出現頻度が少なくなる。一方、最小区分の平均ファイル数は、共有フォルダもカレンダー型インタフェースも同程度であることから、共有フォルダに比べカレンダー型インタフェースの方が短い単語、少ない単語でファイルを分類していると結論づけられる。

以上のことから、提案方式であるカレンダー型インタ

フェースは、従来型の共有フォルダに比べファイルを探しやすい構造になっていると想定できる。このことについて、次章では、シミュレーションにより定量的に検証してゆく。

## 5. ユーザ負荷の定量評価

ここでは、蓄積資料の探索シーンを想定したシミュレーションにより、従来方式と提案方式の理論的なユーザの負荷を検証する。

### 5.1 シミュレーション方法

シミュレーションにあたって、目的のファイルを探し出すためのユーザの負荷として、文献[22]を参考にする。ここでは、記憶の負荷、視覚の負荷、運動の負荷をあげている。このうち共有フォルダでは情報蓄積者と情報利用者が異なる場合を前提に検討するため、記憶の負荷は評価対象外とする。

視覚の負荷については、ファイル探索中に出現する項目数とする。ただし、共有フォルダからのファイルの探索では単純な表示数の比較ではなく、フォルダ名やカテゴリ名という文字列を理解するための負荷とした。運動の負荷は、フォルダを指定して開く操作回数とする。

また、視覚の負荷および運動の負荷の算出にあたって、単純化するために、ファイルではなく目的のフォルダに到達するまでの表示数および操作回数とした。具体的には、共有フォルダの視覚の負荷は、フォルダ名の個数とする。カレンダー型インタフェースでは、年単位のカレンダーの個数と選択したカレンダー内に表示されるカテゴリ名の個数とする。月単位については、カレンダー上の左右の位置として判別するため視覚の負荷から除外する。

なお、カレンダー型インタフェースはWeb上のユーザインタフェースであるため、実装方法により操作方法が変化する。そのため、ここでは、共有フォルダと同様の動作をする実装をした場合として評価する。

また、説明を簡略化するため、提案方式におけるセルは従来方式におけるフォルダと同等の機能であるため、セルについてもフォルダと表記する。

### 5.2 シミュレーションモデル

シミュレーションモデルは、長期にわたって資料が蓄積・継承されてきた事例を参考に作成した文献[8]で用いられた値を基準に作成した。ここでは、1年間で約1,000個のファイルが10年分蓄積されているとした。このとき、総ファイル数は10,000個になる。また、第1階層のフォルダ数も文献[8]のモデルに準じ10個とした。最小区分内のファイル数は表6の値を参考に5個とした。これらの数値をシミュレーションの場合分けにおける共通事項とした。

提案方式は、年、カテゴリ、月別という順番に探索する。

表 7 シミュレーションモデル  
Table 7 Simulation model.

共通事項	
期間	10年
総ファイル数	組織内作成資料 10,000 個
第1階層フォルダ数	10 個
最小区分内ファイル数	5 個
末端フォルダ数	$10,000 \div 5 = 2,000$ 個

個別事項：各階層におけるフォルダ数

階層構造	1階層	2階層	3階層	4階層	5階層
提案方式	10 個	20 個	10 個		
従来方式 1	10 個	20 個	10 個		
従来方式 2	10 個	10 個	5 個	4 個	
従来方式 3	10 個	5 個	5 個	4 個	2 個

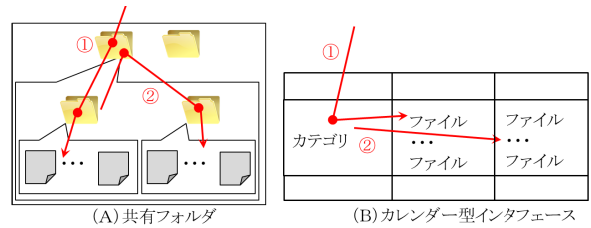
総フォルダ数が2,000 個、年が10 個であるので、各年のカテゴリ数とカテゴリごとの月別のフォルダ数は、それぞれ20 個と10 個となる。

従来方法である共有フォルダは、フォルダ構成の設計が自由であり、フォルダ階層の深さによってユーザの負荷も変化する。そこで、フォルダ階層の深さが提案方式と同等となる3階層と、表6の共有フォルダが平均5.1階層であることから4階層と5階層についてもシミュレーションモデルを作成した。

従来方式は、フォルダ階層が3階層の場合は提案方式と同等となる。このとき、第2階層のフォルダが各20 個、第3階層のフォルダが各10 個となる(表7-従来方式1)。フォルダ階層の深さが4階層の場合は、第1階層が10 個で、総フォルダ数が2,000 個になるように、第2階層、第3階層、第4階層のフォルダ数を割り振った。このとき、階層が深くなるにつれて分類が細くなるという想定で、階層が深くなるにつれてフォルダ数が少なくなるように割り振った(表7-従来方式2)。フォルダ階層の深さが5階層の場合も同様に、第2階層、第3階層、第4階層、第5階層の順番にフォルダ数を割り振った(表7-従来方式3)。

5.3 シミュレーション結果

この状態において、組織知識継承における想定利用シーンを次のように設定した。すなわち、資料探索者がフォルダ名を確認しながらあたりをつけ、目的のファイルが入っているフォルダを探索する場合を想定した。このとき、末端フォルダ内のファイルを確認し、目的のファイルが見つからなかった場合は、同じ階層の他のフォルダを順番に探索する。それでも見つからなかった場合は、1つ上の階層のフォルダに戻り、未確認のフォルダを順番に探索してゆく。一定回数探索したところで、ファイル探索をあきらめるものとし、ここでは15 回までの探索について視覚の負荷と運動の負荷をシミュレーション評価した。なお、15 回



①: 1回目の探索ルート、②: 2回目の探索ルート  
●: 視覚の負荷の測定ポイント、—: 運動の負荷の測定ポイント  
計算方法: 探索1回目=①の確認個数、①の操作回数  
探索2回目=①+②の確認個数、①+②の操作回数

図 9 ユーザ負荷の計算方法

Fig. 9 Measurement method of the load to user.

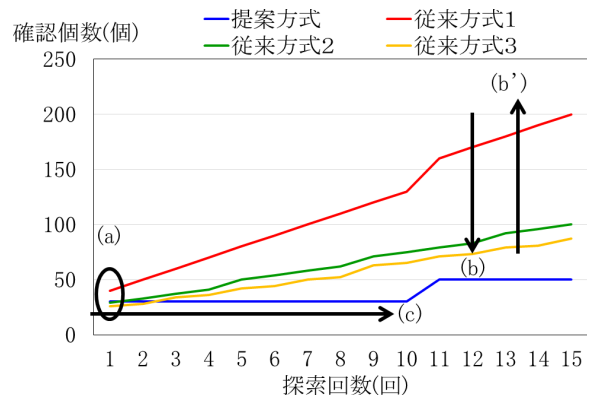


図 10 視覚の負荷

Fig. 10 Load of sight.

以上探索した場合については、15 回までの傾向の延長線上にあり類推可能である。測定方法の概要を図9に示す。

視覚の負荷は、提案方式は、1 回目の探索では、年とカテゴリ名の数となる。2 回の探索で終了する場合は、1 回目の探索が失敗し、となりの月別のフォルダを探索することになる。カテゴリとしては同じなので、視覚の負荷は変化しない。同一カテゴリ内のすべての月別フォルダを探索し終えたとき、隣のカテゴリの確認に移るので視覚の負荷が上がる。以降、一定数ごとに視覚の負荷が上がる構造である。従来方式の場合、第1階層のフォルダ名の確認、第2階層のフォルダ名の確認、第3階層のフォルダ名の確認を行うので、1 回目の探索はこれらの総和になる。2 回目の探索は1 回目の探索における負荷に隣のフォルダ名を確認して操作する負荷が加わる。以降、同様に負荷は累積してゆく。提案方式と3階層、4階層、5階層それぞれの従来方式を比較したものを図10に示す。

運動の負荷は、提案方式と3階層の従来方式は同じ構造なので、運動の負荷は同じになり、単純に探索回数とフォルダ階層の深さに応じて運動の負荷は高くなる。提案方式と4階層と5階層の従来方式を比較したものを図11に示す。

5.4 シミュレーションにおける考察

共有フォルダから資料を探すと動作に対して、対

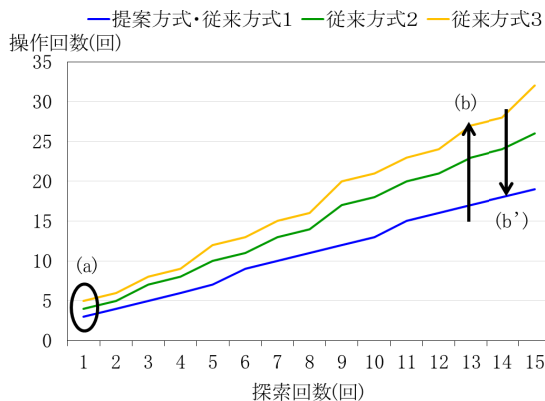


図 11 運動の負荷

Fig. 11 Load of movement.

象となるファイルが1回で見つかる場合と、複数回の探索を繰り返して見つかる場合が考えられる。前者の場合はいずれの方式でもユーザの負荷の差は小さい(図 10-a, 図 11-a)。しかしながら、後者の場合は、どの方式をとったかでユーザの負荷は大きくなってゆくと、必要なファイルを探しやすい構造の検討が重要になる。

従来方式の場合、フォルダ階層を深くしてゆけば視覚の負荷が下がるが(図 10-矢印 b)、運動の負荷は上がる(図 11-矢印 b)。反対にフォルダ階層を浅くしてゆけば運動の負荷は下がるが(図 11-矢印 b'), 視覚の負荷は上がる(図 10-矢印 b')。そのため、視覚の負荷と運動の負荷はトレードオフの関係にある。

これに対して提案方式では、年とカテゴリ名については文字の選択、月についてはカレンダー上の左右の位置によってフォルダを探す構造になっている。ここでは視覚の負荷は選択可能な文字項目数としたため、従来方式に比べ視覚の負荷を低く抑えることができる。特に、従来方式によるファイルの探索では、フォルダ名を確認しながら複数のフォルダを行き来することになる。提案方式の場合は、カテゴリ名を1つ指定すると、あとはカレンダー型インタフェースを左右に移動することで時系列によりフォルダを探索できる。そのため1回の視覚の負荷でより多くのフォルダを探索できる構造になっている(図 10-矢印 c)。

また、提案方式において、目的のフォルダに到達するには、年、カテゴリ、月の指定という3段階となる。そのため、運動の負荷は、従来方式における運動の負荷の一番小さい3階層のフォルダ構造と同じになる。その結果、視覚の負荷を下げることで、運動の負荷を下げることを両立できる構造になっている。

## 6. 考察

共有フォルダからファイルを取り出す際に、「ユーザの負荷を低減させつつ目的のファイルを探しやすくする分類構造を実現する」という目標に対して、どのように実現できたかを具体例を用いて考察する。

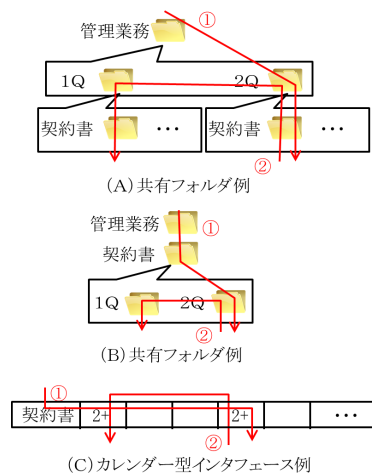


図 12 ファイル探索方法の比較

Fig. 12 Folder tracing examples.

多数のフォルダを有する共有フォルダを探索する際のユーザの負荷について、従来方式では視覚の負荷と運動の負荷のトレードオフが発生するが、提案方式では視覚の負荷および運動の負荷の両方を同時に抑えたフォルダ探索が実現できる。特に、ファイル探索は複数回繰り返すことになるため、複数回探索してもユーザの負荷が少ないことが望ましい。

たとえば、3階層のフォルダ構造として図 12-A のようなものがあつた場合、蓄積ファイルの直近の資料を探すのに図 12-A-①のような流れになる。あわせてもう少し古いファイルも確認が必要になった場合、上位のフォルダに戻って隣のフォルダに移る(図 12-A-②)。その間に複数の関係ないフォルダ名を目にすることになり、これが視覚の負荷を増加させる。また、フォルダを行き来するための操作があり、これが運動の負荷になる。視覚の負荷や運動の負荷を下げるには、末端のフォルダ構造が図 12-B のような構造であることが望ましい。カレンダー型インタフェースでは、図 12-C のような流れになる。これは、図 12-B と同様の流れで探すことができる。すなわち、カレンダー型インタフェースは資料を探すためのフレームワークであるため、ファイル探索に理想的なフォルダ構造で探すことができる。

このことから、提案方式は、ファイル探索において従来方式の共有フォルダよりもユーザの負荷を抑えた構造が実現できる。

特に、資料を蓄積する際、組織メンバそれぞれが最適なフォルダ構造を意識し、サブフォルダを作り資料を蓄積してゆくことは難しい。実際は、その時々で思いで作成している。つまり、情報蓄積時と情報利用時の「手がかりに関する意識のずれ」や「利益の不均衡」という情報共有における非対象性の問題が発生する。共有フォルダは、様々な資料を気軽に自由に入れられることで組織内に定着しているものである。この利便性を損なわずに情報利用を促進す

SEARCH												
2015+												
	01	02	03	04	05	06	07	08	09	10	11	12
共同研究+		12+										
2014+												
	01	02	03	04	05	06	07	08	09	10	11	12
共同研究+	12+	41+	16+	1+				1+		4+		
2013+												
	01	02	03	04	05	06	07	08	09	10	11	12
共同研究+		4+	41+		1+	1+	31+	27+	14+	28+		3+
2012+												
	01	02	03	04	05	06	07	08	09	10	11	12
共同研究+		24+	22+			2+		7+	22+		2+	3+
2011+												
	01	02	03	04	05	06	07	08	09	10	11	12
共同研究+		6+	4+	11+	3+	10+	3+	2+	2+	6+		2+
2010+												
	01	02	03	04	05	06	07	08	09	10	11	12

図 13 複数年表示例

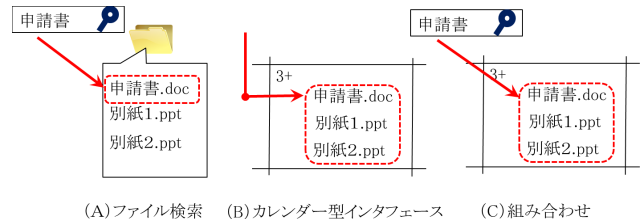
Fig. 13 Simultaneous display example for two or more years.

るには、情報共有における非対称性を解消する必要がある。このことは、今回提案したような情報利用に特化したインタフェースの活用が有望となる。このほかに、情報利用に特化したいくつかの機能の追加も期待できる。

たとえば、今回追加した機能の1つに選択したカテゴリのみを表示する機能がある。言い換えると、特定のカテゴリについて複数年同時参照ができる(図13)。これにより、年をまたがって同じカテゴリの資料を容易に探し出すことが可能になる。このような機能を発展させ、同じ周期で必要となるファイルを取り出せる機能の実現も有望と思われる。

他に、情報利用の1つのシーンとして、ファイルの検索機能を活用したファイルの探索がある。OSで推奨される方法[14]や、発展的な検索機能[15]の活用が考えられる。ファイル探索において、ファイル名が明確な場合は、これらの検索機能によって探し出すほうが効率的であり、利用シーンによってはファイル検索も有効と考えられる。そのため、開発したインタフェースにおいても、動作確認の意味で簡易的なファイル検索機能をHTML内のJavaScriptのパターンマッチングで実装した。

ここで、ファイル検索と提案方式の差異と組み合わせ方法について述べる。ファイルの利活用では、必要なファイルが1つとは限らない場合もある。ファイル検索であればファイル名のキーワードが分かればピンポイントで目的のファイルが探せる(図14-A)。しかしながら、利活用すべきファイルが異なるファイル名で複数あった場合、必要なファイルの見落としが起る。これに対して、カレンダー型インタフェースでは、同じ活動で同時期に作成したファイルは同じセルに出てくるため(図14-B)、関連ファイルをまとめて探し出せる。このことから、両者を組み合わせると、ピンポイントで目的のファイルを探し出せるのと同時に関連ファイルも探し出せるため、より効果的な



(A)ファイル検索 (B)カレンダー型インタフェース (C)組み合わせ

図 14 ファイル検索との比較

Fig. 14 Example of comparison with file search.

ファイル探索が可能になると考えられる(図14-C)。よって、共有フォルダからのファイル探索という目的に対して、ファイル検索とカレンダー型インタフェースは、どちらが優れているというよりも、使い分けや組み合わせが良い。そのため、ファイル検索と組み合わせで目的のファイルに効率的に到達するための機能も有望と思われる。

また、今回の検討では、長期にわたって運用されてきた共有フォルダの利用状況と、Tipsとして一般的にいわれている共有フォルダの整理方法の特徴から、共有フォルダをより効果的に活用するための仕組みを提案した。ここでは、組織内でボトムアップ的に長期間運用するうちに中身がよく分からなくなってしまった共有フォルダを想定した。このようなフォルダの構成は複雑であり、現在のカテゴリ生成方法では一部の処理について目視確認と手作業による補正で対処している。さらに、提案手法は、様々な組織の共有フォルダで使えることを想定しているが、他組織での評価は未実施である。

そこで、1つめの今後の課題として、様々なパターンの複雑なフォルダ構成を一般化したカテゴリ生成手法を実現する必要がある。たとえば、同一内容にもかかわらず別々のカテゴリになってしまうようなフォルダ名の表記のゆらぎの吸収方法、複数階層のフォルダ名によって活動単位が表現される場合のカテゴリ生成手法、同一セルに大量のファイルが偏ってしまう場合のカテゴリ分割方法の検討などがある。

2つめの課題として、今回と同様の運用方法をしている他組織での検証がある。ほかに、定式化した運用方法がある組織、共有フォルダ内の命名規則を細かく設定している組織、画像データやプログラムなどを主に扱う組織などでは、制限にあわせて設計する必要があるので設計方法の定式化を検討する必要がある。

## 7. まとめ

長期間にわたって継続的に組織活動を行うには、組織構成員が替わっても蓄積してきた知識を活用してゆく必要がある。現在、多くの現場組織では、組織活動の記録として共有フォルダに資料を蓄積し、組織の知識として活用している。しかしながら、共有フォルダを長期間運用することにより蓄積資料は膨大になり資料の全体像を把握できない

ことや、必要な資料を探し出すのが困難になり、蓄積された知見を組織知識として十分に活用できないという問題がある。この問題を解決するために、資料作成のタイムスタンプから得られた時間情報と、フォルダの命名方法に着目して抽出した活動単位に基づき蓄積資料を分類整理し、共有フォルダを組織知識として活用するカレンダー型インタフェースの提案を行った。

提案方式に対して、実データによる方式検証、シミュレーションによる定量的な評価を行った。その結果、ファイルの蓄積とファイルの利用という非対称な情報共有の動作に対して、従来方式の共有フォルダは同じインタフェースで実施する必要がある。一方、提案方式では、ファイルの蓄積は従来からの共有フォルダのインタフェースで行い、ファイルの利用ではカレンダー型インタフェースで行う。このことにより、ファイルの提供は定着した方法で実施でき、ファイルの対策は共有フォルダでのデメリットを補った方法で実施できる。特に、ファイルの探索は複数回繰り返すことになるため、複数回探索してもユーザの負荷が少ないことが望ましい。提案方式は、ファイル探索において従来方式の共有フォルダよりもユーザの負荷を抑えた構造が実現できる。

以上のことから、共有フォルダに蓄積された資料を活用する場面において、従来からの共有フォルダに比べた提案方式の有効性を確認した。

#### 参考文献

[1] 野中郁次郎：知識経営の戦略，情報処理学会誌，Vol.47, No.4, pp.547-552 (2006).  
 [2] 梅本勝博：ナレッジマネジメント：最近の理解と動向，情報の科学と技術，Vol.62, No.7, pp.276-280 (2012).  
 [3] 野中帝二，安部純一：組織における知の継承—知の継承における五つの誤解，特許庁技術懇談会，No.268, pp.34-42 (2013).  
 [4] 福田國彌，水野博之，加納剛太：起業工学，幻冬舎ルネッサンス (2012).  
 [5] 野中郁次郎，紺野 登：知識創造経営のプリンシプル，東洋経済新報社 (2012).  
 [6] Wenger, E., McDermott, R. and Snyder, M.W.: *Cultivating Communities of Practice*, Harvard Business School Press (2002).  
 [7] 犬塚 篤：ビッグデータ時代の知識共有，情報の科学と技術，Vol.62, No.7, pp.302-307 (2012).  
 [8] 齊藤典明，金井 敦：組織知識継承を実現する死蔵されない共有フォルダ構成法，情報処理学会論文誌，Vol.54, No.1, pp.295-308 (2013).  
 [9] 齊藤典明，金井 敦：業務の引継ぎを容易にするスケジューラ連動型組織知識継承基盤，情報処理学会論文誌，Vol.55, No.1, pp.127-142 (2014).  
 [10] 日本十進分類法新訂9版分類基準，国立国会図書館 (2010).  
 [11] Wurman, S.R.: *Information Anxiety 2*, Que (2000).  
 [12] 齊藤典明：組織知識継承のための混沌フォルダ整理法，人工知能学会，第15回知識流通ネットワーク研究会，SIG-KSN-015-08 (2014).  
 [13] 野口悠紀雄：「超」整理法，中公新書 (1993).  
 [14] エクスプローラーでファイルを検索する (オンライン)，入

手先 (<http://windows.microsoft.com/ja-jp/windows-8/search-file-explorer>) (参照 2015-03-23).

[15] Everything Search Engine (online), available from <http://www.voidtools.com/> (accessed 2015-03-23).  
 [16] 国藤 進，加藤直孝，門脇千恵，敷田幹文：知的グループウェアによるナレッジマネジメント，日科技連 (2001).  
 [17] 松下 温，岡田謙一：コラボレーションとコミュニケーション，共立出版 (1995).  
 [18] Rekimoto, J.: Time-Machine Computing: A Time-centric Approach for the Information Environment, *Proc. 12th Annual ACM Symposium on User Interface Software and Technology (UIST '99)*, pp.45-54 (1999).  
 [19] Yamamoto, K., Kuriyama, T., Shigemori, H., Kuramoto, I., Tsujino, Y. and Minakuchi, M.: Provenance Based Retrieval: File Retrieval System Using History of Moving and Editing in User Experience, *Proc. 36th Annual IEEE Computer Software and Applications Conference (COMPSAC2011)*, Munich, Germany, pp.618-625 (2011).  
 [20] 福井秀徳，森田哲郎，岡野真一，沼尾正行，栗原 聡：ファイルネットワークに基づいた情報の抽出と可視化，第22回人工知能学会全国大会，2H-3 (2008).  
 [21] Dawson, F. and Stenerson, D.: Internet Calendaring and Scheduling Core Object Specification (iCalendar), IETF RFC 2445 (1998).  
 [22] Weinschenk, S.: *100 Things Every Designer Needs to Know About People*, Person Education Inc. (2011).



齊藤 典明 (正会員)

1988年法政大学工学部卒業。1990年同大学大学院工学研究科修士課程修了。同年日本電信電話(株)入社。1999年奈良先端科学技術大学院大学博士後期課程修了，博士(工学)。現在，日本電信電話(株)セキュアプラットフォーム

フォーム研究所主任研究員。情報ネットワークを用いた知識共有およびセキュリティに関する研究開発に従事。著書に『インターネット広場の仲間たち』等。電子情報通信学会，人工知能学会各会員。本会シニア会員。



金井 敦 (正会員)

1980年東北大学工学部通信工学科卒業。1982年同大学大学院工学研究科情報工学科博士前期課程修了。同年日本電信電話公社電気通信研究所入社。ソフトウェア開発プロセス，ソフトウェア分散開発環境，Webサービス開発技術，ネットワークコミュニティ，情報セキュリティ，ネットワークセキュリティの研究開発に従事。2008年から現在，法政大学理工学部応用情報工学科教授。博士（情報科学）。著書に『攻めと守りのシステムセキュリティ』等。電子情報通信学会シニア会員，IEEE会員。本会シニア会員。



谷本 茂明 (正会員)

1982年徳島大学工学部電気工学科卒業。1984年同大学大学院工学研究科電気工学専攻修了。同年日本電信電話公社入社。主にプライベートネットワークシステムにおける研究開発に従事。2009年より千葉工業大学社会システム科学部准教授。2012年教授。現在，情報セキュリティマネジメント，特にPKI応用，クラウドセキュリティ等の研究に従事。博士（工学）。電子情報通信学会シニア会員，プロジェクトマネジメント学会，日本経営工学会，IEEE各会員。本会シニア会員。