筑波大学大学院博士課程

システム情報工学研究科修士論文

ファイルシステムにおける
ファイル操作が可能な内容表示インタフェース

謝 湘平

修士（工学）

（コンピュータサイエンス専攻）

指導教員　田中 二郎

2015 年 3 月

概要

ファイルシステムのGUIにおいてユーザがファイル操作を行う際，フォルダをたどってファイルを探し出し，ファイルのアイコンをダブルクリックしてファイルを開く．印刷や削除といったいくつかのファイル操作は右クリックで出現するコンテキストメニューでも可能である．だが，ユーザがファイルの内容のコピー等の普段良く行う操作を行いたい場合，ファイルをアプリケーションで開いてメニューをたどる一連の操作を毎回行うのは煩雑である．

そこで，本研究ではユーザが簡単な操作でよく使用するファイル操作を行うことができるように，フォルダ展開とファイルの内容表示のインタフェースを提案し，プロトタイプシステムとしてIcons++を作成した．

Icons++では，ユーザがフォルダのアイコンにマウスカーソルを当てると，アイコンの近くにフォルダ内のサブフォルダやファイルが展開される．複数階層のフォルダを同じ画面で展開することができるため，ユーザは入れ子状に表示された複数のフォルダの中身を一度に閲覧することができる．また，ユーザがファイルのアイコンにマウスカーソルを当てると，アイコンの近くにファイルの内容が表示される．ユーザは表示されたファイルの内容ページに対し，マークを付けてページを強調することができる．さらに，ファイルの内容を表示した画面上には操作を示すアイコンが表示される．ユーザは操作を示すアイコンをクリックすることにより，よく使用するファイル操作をワンクリックで行うことができる．

本研究では，提案インタフェースを体現するプロトタイプシステムIcons++を実装し，Icons++の有用性を検証するために評価実験を行った．実験結果からIcons++がユーザのファイル操作において有用であることを確認した．さらに，提案インタフェースに関する改善案について考察を行った．

目 次

図 目 次

第1章　序論

1.1　Graphical User Interface について

今日のコンピュータは Graphical User Interface（GUI）を介して操作することが一般的である．GUI はデスクトップメタファというコンセプトに基づいて設計され，ウィンドウ・アイコン・メニュー及びポインティングデバイスを構成の基本要素とする [1, 2]．ユーザがキーボードからコマンドを入力してコンピュータに命令を与える Command Line Interface（CLI）と異なり，GUI では，操作対象がアイコン等のグラフィカルな表現により示され，ポインティングデバイスによる直接操作が可能である．

1.2　ファイルシステムの GUI

ファイルシステムの GUI において，何階層かフォルダをたどり，ファイルを見つけ，操作を加えるためにファイルのアイコンをクリックしてアプリケーションで開くことは一般的である．アイコンによって，ユーザはフォルダやファイルを一覧し [3]，フォルダの中身の概要やファイルの種類について把握することができる．したがって，アイコンは閲覧性に優れていると言える．一方で，アプリケーションによってユーザは様々な操作 (e.g., 編集や形式変換など) をファイルに対して行うことができる．そのため，アプリケーションは操作性に優れていると言える．

1.3　ファイルの閲覧と操作におけるニーズ

一般にユーザにはフォルダやファイルの内容について素早く把握したいというニーズがある．このニーズを示す 1 つ目の例として，サムネイル画像を含んだフォルダのアイコンやサムネイル画像を使用したファイルアイコンを挙げる．例えば，PDF ファイルのアイコンにはファイルの 1 ページ目のサムネイル画像が使用されている．これによって，ユーザが対象ファイルの内容について思い出すきっかけができる．また，写真フォルダのアイコンとして，内含する写真の画像が埋め込まれているフォルダアイコンを使用しているものもある．２つ目の例として，Gmail におけるファイルのプレビュー機能を挙げる．Gmail のメール閲覧画面では，添付ファイルのアイコンを押すことにより，アプリケーションを用いてファイルを開くことなく，ファイルの内容を同じ画面内で閲覧することができる．

また，ショートカットキーやマウスジェスチャに見られるように，ユーザにはよく使用する操作を素早く行いたいニーズがある [4, 11].

1.4 アイコン配置に考慮したフォルダ展開の重要性

階層構造になっているフォルダの表示に関して，既存のファイルマネージャではツリー表示やカラム表示を用いているものが多い．ツリー表示とは，リスト状にフォルダやファイルを表示し，フォルダの中身にあるサブフォルダやファイルをインデントをずらして，親フォルダの項目の下にリスト状に表示するものである．カラム表示では，フォルダ階層ごとを横に並べて表示する．具体的に一つのカラムには，縦一列のリスト状にフォルダやファイルが表示される．その横にさらにカラムが設置され，ユーザの選択したフォルダの中身のサブフォルダやファイルがこのカラムにリスト状に表示される．カラム表示は Mac OS の Finder で使われている．

ユーザは自分のコンピュータの使い方や現在のタスクへの必要性に応じてファイルを整理したりファイルの場所を決めたりしている [5, 8, 9] ．また，ユーザは検索機能を用いてファイルに一気にたどり着くよりも，見える形でフォルダをたどり，周りのファイルとの関連性を見ながらファイルにたどり着く方を好む [6] ．そのため，ユーザがファイルを探す際には，デスクトップやファイルマネージャ内で表示されたアイコンやアイコンの表示位置といった視覚的な情報が大きな手掛りとなる [7, 10] ．

本研究では，フォルダを展開し，中身のサブフォルダやファイルを表示する際に，フォルダの階層構造を視覚化しながら，リストではなくユーザのファイル探索の手掛かりとなるアイコンの配置をユーザに見せる必要があると考える．

1.5 本研究の位置づけ

本研究では，フォルダやファイルの内容を表示し，更にアイコンを用いてファイル操作を素早く行うことが可能なユーザインタフェースを提案し，プロトタイプシステムとして Icons++ の開発を行った．図 1.1 では閲覧性と操作性を横軸にとり，本研究で作成した Icons++の位置づけを示している．

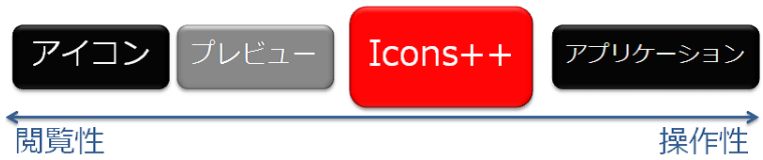


図 1.1: Icons++の位置づけ

ユーザはアイコンを見ることにより，複数のフォルダやファイルを一覧できる．そのため，アイコンは閲覧性に優れていると言え，閲覧性よりに配置される．また，プレビュー機能は

ファイルの内容を表示し，ユーザによるスクロールを許容する．そのため，プレビュー機能は閲覧性に加え，少しの操作性を持ち，アイコンの横に配置される．ユーザはアプリケーションを用いることにより，ファイルの内容の細かい編集や更新ができる．そのため，アプリケーションは操作性に優れ，操作性よりに配置される．本研究で提案するインタフェースは閲覧性と操作性の双方を兼ね備えたものを目指しているため，Icons++をアイコン（プレビュー機能）とアプリケーションの間に位置づける．

1.6 本論文の構成

本論文は本章を含め８章で構成されている．第１章では既存のデスクトップやファイルシステムの GUI について説明し，ユーザのニーズと既存の GUI の問題点を挙げた．

第２章では本研究の目的とアプローチについて述べる．第３章では本研究が提案するインタフェースについて説明し，本研究の利点を述べる．第４章では提案インタフェースを用いた利用シナリオの例を挙げる．第５章ではプロトタイプシステム Icons++の実装の仕方を説明する．第６章では関連研究や関連するアプリケーションを挙げ，本研究との共有点や相違点について述べる．第７章で Icons++を用いた評価実験と実験の結果，および実験結果を受けた UI の改善案について述べる．最後に，第８章では本研究についてまとめ，将来の展望について言及する．

第2章　目的とアプローチ

2.1　本研究の目的

本研究では，ユーザのファイル操作時に簡単なジェスチャとアイコンの利用を取り入れることにより，ユーザがファイルの内容を確認しやすく，ファイルの操作を素早くできるようなインタフェースを目指す．そのため，ファイルの内容の把握や操作を素早く行うことができるユーザインタフェースを開発することを目的とする．

2.2　本研究のアプローチ

本研究のアプローチとして，簡単な操作を用いてフォルダやファイルの内容を閲覧でき，同時にファイル操作を行うことができるインタフェースを提案する．提案インタフェースでは，対象フォルダやファイル（以降フォルダとファイルをまとめて指す場合に「アイテム」と呼ぶ）をホバーすることによって，アイテムの内容を閲覧することができる．

フォルダのアイコンがホバーされた際に，フォルダの内容が入れ子状に表示される．そのため，ユーザは複数の階層にあるアイテムを一覧することができる．さらに，階層やフォルダを跨いでアイテムの移動を行うことができる．

ファイルのアイコンがホバーされた際に，ファイルの内容がアイコンの近くに表示される．ファイルの内容が表示される画面では，表示されたページに対するマーク付けを可能にしている．これにより，ユーザは目立たせたいページと目立たせたくないページに対して，視覚的にページの見た目を変更することができる．また，ファイルの内容が表示されると同時に，ファイルの操作を表したアイコンも画面上に表示される．それを用いることにより，ユーザはよく行う操作をワンクリックによって行うことができる．

第3章　システムのインタフェース設計

ユーザがフォルダやファイルの内容を見て，操作を加えられるように，まず「コンテンツビュー」と呼ばれる内容を表示するウィンドウを画面に出す．このコンテンツビューにより，ユーザはフォルダやファイルの内容を確認し，それが自分の欲していたアイテムであるかどうかを判断することができる．

フォルダのコンテンツビューは入れ子状に表示される．それに対し，ファイルのコンテンツビューはアイコンの近くに表示される．また，ファイルのコンテンツビューでは「操作アイコン」と呼ばれるアイコンが表示されている．ユーザは操作アイコンをクリックすることにより，ファイルに対してよく行う操作をワンクリックで行うことができる．さらに，ファイルのコンテンツビュー内で内容を確認しやすいように，ユーザは表示されているファイルのページのサムネイルに対してマークを付けることが可能である．マーク付けとして，本研究では「Dog-ear」を付けることと「ページの折り畳み」を行うことを提案する．

3.1　ホバーによるアクセス

ユーザはホバーによってコンテンツビューにアクセスすることができる．

ホバーとはアイテムに対してマウスカーソルを当てる動作のことを指す．ホバーを用いることにより，ユーザはマウスジェスチャのような複雑な操作を記憶する必要がない．さらに，コンテンツビューについて全く知らないユーザでも，フォルダやファイルに対して操作を加える途中で，自然とホバーによって表示されるコンテンツビューに気付く可能性がある．

ホバーはマウスによる他の動作（ダブルクリックを用いてファイルをアプリケーションで開く動作，右クリックによるコンテキストメニューの表示など）と競合しない．そのため，ユーザはアイテムに対して通常の使い方と同様にマウスを使用することができる．

3.2　コンテンツビューの表示

3.2.1　コンテンツビューの表示・固定・消去

ユーザがアイテムのアイコンを 1 秒程度ホバーすると，コンテンツビューがアイコンの右下に表示される．ユーザがアイコンからマウスカーソルを離すと，表示されたコンテンツビューが消える．多数のアイテムに対して多数のコンテンツビューが表示されたままではユーザの作業の邪魔になる可能性があるため，コンテンツビューはユーザが必要に感じた時にのみ手

動で固定できるようにする．ホバーしたままアイテムのアイコンをワンクリックすると，コンテンツビューを固定することができ，マウスカーソルがアイコンから離れてもコンテンツビューは消えなくなる．コンテンツビューを固定することにより，ユーザがアイテムの内容をスクロールして詳しく見る，または複数のアイテムの内容を見比べることが可能となる．コンテンツビューの右上にある×印を押すことにより，固定されたコンテンツビューを消すことができる．

3.2.2　フォルダの中身の表示

フォルダのコンテンツビューは，フォルダアイコンに対する 1 秒程度のホバーにより出現する．図 3.1 の上部の図がマウスカーソルを対象ファイルに当てたときであり，図 3.1 の下部の図はホバーによってフォルダのコンテンツビューが表示されたときを示している．

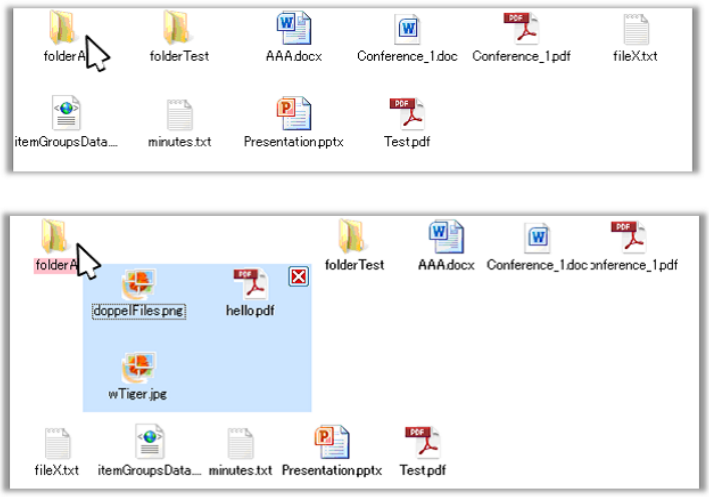


図 3.1: ホバーによるフォルダの中身の表示

図 3.1 のように，フォルダのコンテンツビューは閲覧しやすさを考慮して，近くのアイテムと重複しないようにメインウィンドウに埋め込まれた形で表示される．フォルダのコンテンツビューには，そのフォルダの中身にあるサブフォルダやファイルのアイコンが表示される（図 3.1 の色付きエリア内）．また，フォルダのコンテンツビューの大きさはフォルダの中身の要素数に応じて動的に変わる．

コンテンツビューを固定することにより，図 3.2 のように複数個のコンテンツビューを同一画面内に表示することが可能である．そのため，ユーザは複数個のフォルダの中身を同一画面で同時に閲覧することができる．

図 3.2 は，1 つ目のフォルダをホバーし，コンテンツビューをフォルダアイコンの近くに表示させた後に，２つ目のフォルダをホバーしてコンテンツビューを表示させたものを示す．

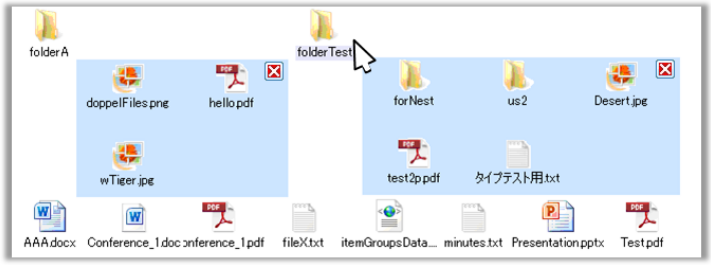


図 3.2: 複数フォルダに対するコンテンツビューの表示

3.2.3　コンテンツビューにおける入れ子状表示を用いたフォルダ展開

フォルダのコンテンツビューでは，表示されたサブフォルダに対してさらにコンテンツビューを表示させ，サブフォルダの中身をユーザに見せることができる（図 3.3(b) と 図 3.3(c) ）.

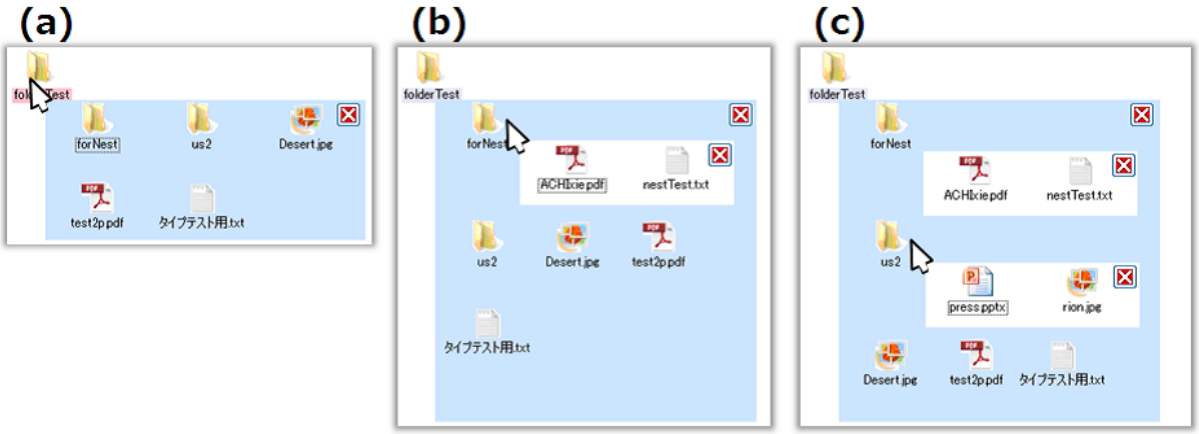


図 3.3: 入れ子状表示を用いた複数フォルダの中身の表示

図 3.3(a) では，フォルダアイコンをホバーし，コンテンツビューを用いてフォルダの中身を表示させている．その後サブフォルダをホバーし，サブフォルダの中身を表示させ（図 3.3(b) ），さらに同じ階層のサブフォルダについても中身を表示させた（図 3.3(c) ）.

図 3.3(b) と 図 3.3(c) のように，入れ子状にフォルダの中身を展開していくことにより，ユーザはフォルダの実際の階層に関係なく，同一画面で階層の異なる複数フォルダについて中身を確認することができる．また，フォルダのコンテンツビューを用いてフォルダを展開した後に，中身のファイルについても後述するコンテンツビューを用いて内容を閲覧することができる（図 3.4 ）.

図 3.4 は，あるフォルダについてコンテンツビューを用いて中身を展開した後に，中身にあるファイルをホバーし，後述するファイルのコンテンツビューを表示させたものである．これを用いることにより，ユーザはフォルダの階層に関係なく，複数フォルダ内のファイルに

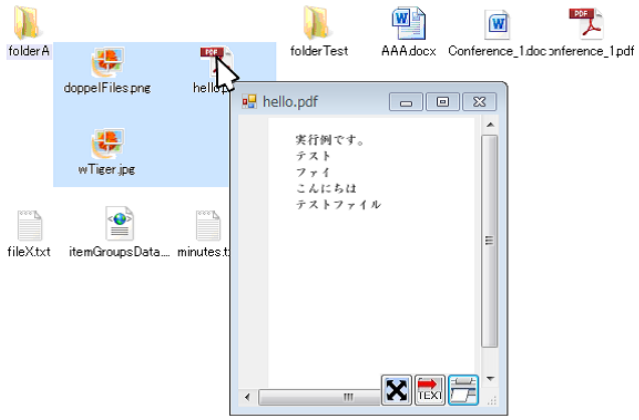


図 3.4: フォルダ展開後のファイルの内容閲覧

ついて内容を閲覧しながらファイルの確認を行うことができる．

3.2.4 ファイルの内容表示

ファイルの場合はコンテンツビューが新しいウィンドウとしてアイコンの近くに表示される．図 3.5 では，ファイルをホバーするとコンテンツビューが表示され，ホバーされたファイルの内容が閲覧可能になった例を示している．

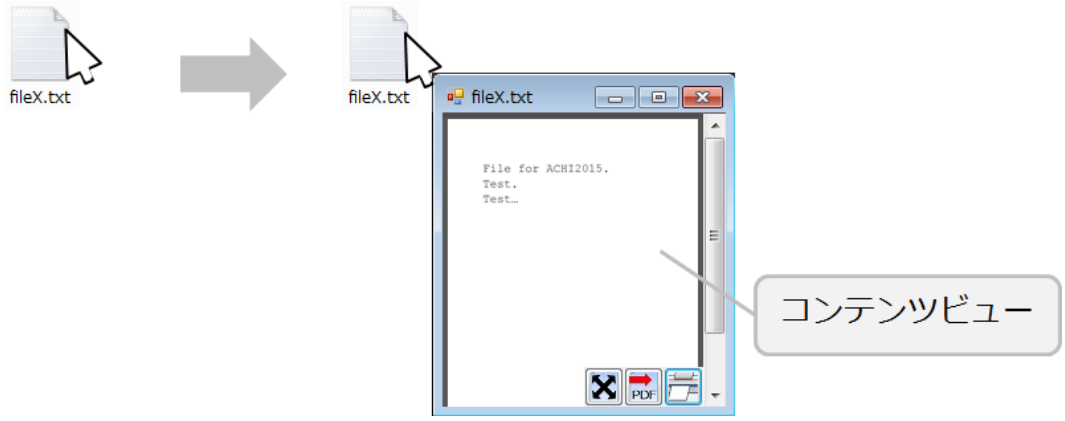


図 3.5: コンテンツビューによるファイルの内容表示

ファイルのコンテンツビューは，ファイルアイコンに対する 1 秒程度のホバーによって出現する．ファイルのコンテンツビューではユーザはファイルの各ページについて閲覧することができる．

図 3.6 では 1 つ目のファイルをホバーして，コンテンツビューを表示させた後にクリックによってコンテンツビューを固定し，2 つ目のファイルについてもコンテンツビューを表示さ

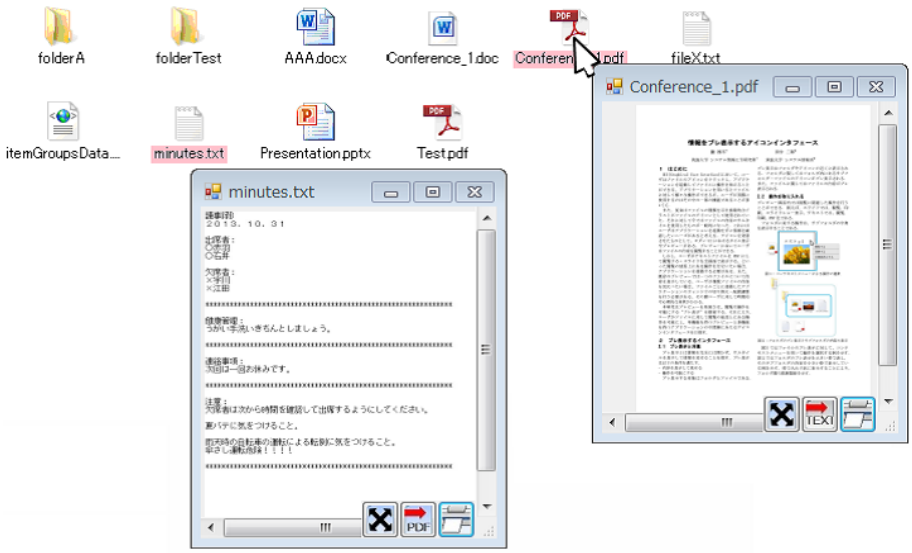


図 3.6: 複数ファイルに対する内容表示

せたものを示している．このように，ユーザは複数のファイルについてコンテンツビューを表示させることができ，ファイルの内容を確認したり見比べたりすることが可能である．また，後述するファイルの操作についても，同じ画面内で複数のファイルに対してファイル操作を行うことが可能である．

3.3 コンテンツビューにおける操作

コンテンツビューを用いることにより，ユーザはアイテムについて中身を閲覧するだけでなく，ファイル操作を行うこともできる．

3.3.1 複数フォルダ間のアイテムの移動

フォルダのコンテンツビューにおいて，ユーザは階層にかかわらず複数個のフォルダについてコンテンツビューを表示させて中身を確認することができる．その際，アイコンのドラッグ・アンド・ドロップ（以降ドラッグ&ドロップと記す）により，フォルダ間のアイテムの移動を行うことが可能である．

複数フォルダ間におけるファイル移動の例を図 3.7 に示す．あるフォルダのサブフォルダの中身にあるファイルを違うフォルダに移動させたい場合を想定する．図 3.7 では，ACHIxie.pdf というファイルのアイコンを移動先のフォルダのコンテンツビュー中にドラッグ&ドロップしている（図 3.7 上段）．移動後に各フォルダについて展開した図が図 3.7 下段である．右側のフォルダのサブフォルダ内から ACHIxie.pdf がなくなり，左側に位置するフォルダの中に ACHIxie.pdf が存在することが確認できる．

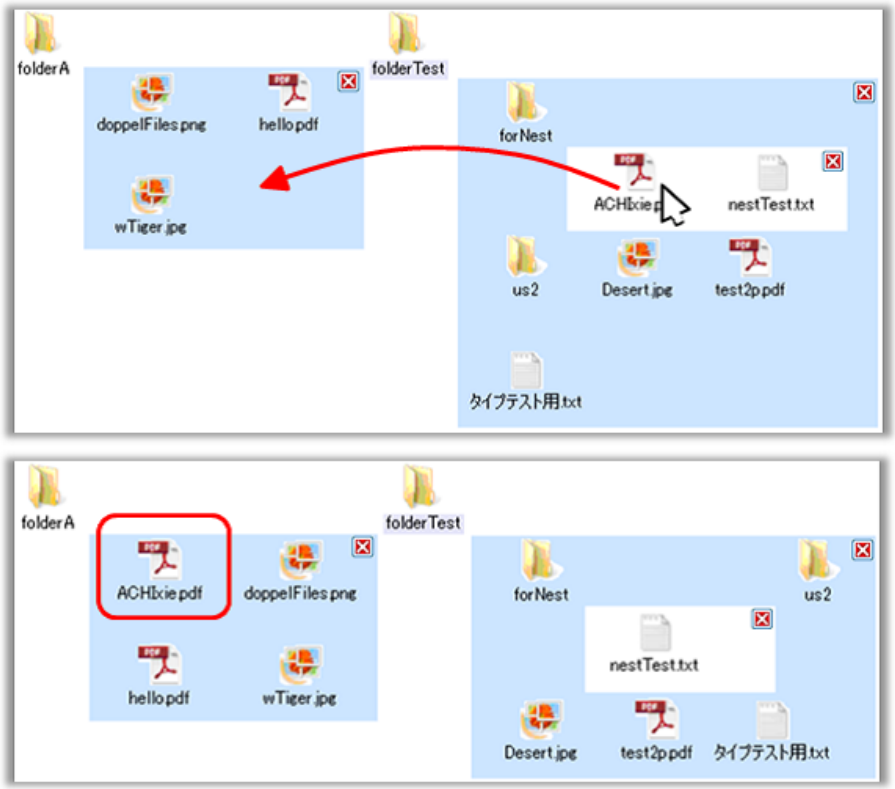


図 3.7: フォルダの入れ子状表示におけるファイル移動

3.3.2 アプリケーションへのアクセス

ファイルのコンテンツビューでは，画面上部にマウスカーソルを移動させて 1 秒程度ホバーするとアプリケーションのアイコンが表示され，選択可能になる（図 3.8 ）．

ユーザが表示されたアプリケーションアイコンの中から一つを選択してクリックすると，ファイルは指定されたアプリケーションによって開かれる．選択できるアプリケーションの種類はファイルの種類によって動的に変化する．

図 3.8 (a) はユーザがマウスカーソルをコンテンツビューの上部に移動させる様子を示している．コンテンツビューの上部を 1 秒程度ホバーすると，アプリケーションのアイコンが画面上部に表示される（図 3.8 (b) ）．そこからユーザが「メモ帳」のアイコンを選択すると，メモ帳によってファイルが開かれる（図 3.8 (c) ）．

このように，ホバーによってユーザはファイルアイコンからコンテンツビューにアクセスすることができる．さらにホバーによってコンテンツビューからアプリケーションにアクセスすることができる．そのため，提案インタフェースはアイコンとアプリケーションの中間に位置し，アイコンとアプリケーションを結びつけていると言える．また，従来のダブルクリックによるファイル展開と異なり，提案インタフェースでは一つのファイルを複数のアプ

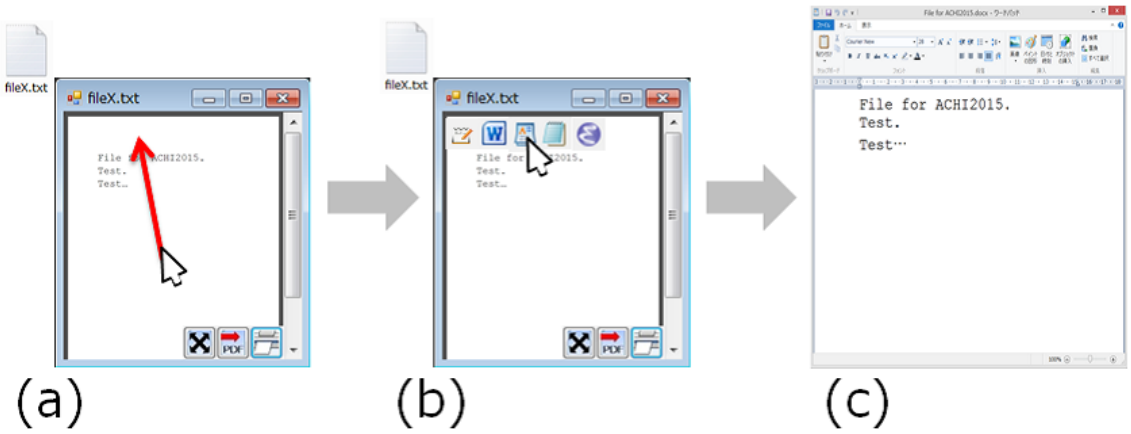


図 3.8: アイコン・コンテンツビュー・アプリケーション間のアクセス

リケーションと結びつけて選択可能にしている．

3.3.3 操作アイコンを用いたワンクリックのファイル操作

アプリケーションを用いた場合では何ステップも経てから選択可能となる操作を，ユーザは操作アイコンを用いることにより，よく使用する操作をワンクリックで行うことができる．図 3.9 ではコンテンツビューにおける操作アイコンの例を示している．また，図 3.10 はいくつかの操作アイコンの例を示している．

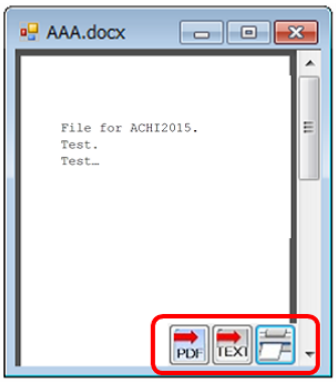


図 3.9: コンテンツビュー下部に表示される操作アイコン

図 3.9 のように，操作アイコンはコンテンツビューの下部に表示される．ファイル操作アイコンとして，ファイルの形式変換（e.g., PDF やテキストに変換する），内容コピー，フルスクリーン表示，スライドショー表示，印刷などが挙げられる．コンテンツビューに表示されるファイル操作アイコンはファイルの種類によって異なる．コンテンツビューでファイル

図 3.10: (a) フルスクリーン表示 (b) スライドショー表示 (c)PDF 変換 (d) テキスト変換 (e) 印刷

の内容の表示の妨げにならないように，画面下部に表示するファイル操作アイコンは最大で６つである．これは，画面下部で一行で表示できる上限の数である．

既存のファイル操作アイコンに加え，ユーザは必要な操作をカスタマイズしたカスタム操作アイコンを作成することが可能である．カスタム操作アイコンはユーザが行いたい操作を記録し，ユーザによってアイコンが選択されたときに，記録した操作を再現するものである．

このように，従来では複数過程を経て行われてきた操作をワンクリックで可能にしているため，提案インタフェースは操作性に優れている．

3.3.4 Dog-ear とページの折り畳みを用いたマーク付け

ファイルのコンテンツビューでは，ユーザはファイルのページの重要度に応じて，サムネイルに対してマークを付けることができる．付けられたマークにより，ユーザは素早く自分の欲しているページや部分を発見できる．本研究では Dog-ear とページの折り畳みの２種類のマークを可能にしている（図 3.11 ）．

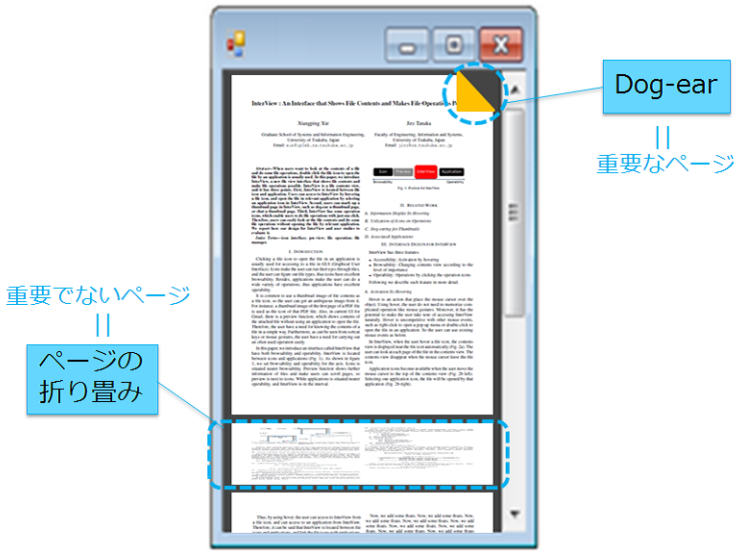


図 3.11: Dog-ear とページの折り畳み

Dog-ear はユーザが重要だと思ったページに付けることができる．ユーザが重要でないページだと判断したページに対しては，ページのサムネイルを折り畳むことが可能である．例え

ば，あるレポートでユーザが見直したい部分があった場合，その部分を含むページのサムネイルに対して Dog-ear を付けることができる．またレポートの表紙が特に役に立つ情報がないと思ったら，表紙ページのサムネイルを折り畳むことができる．

Dog-ear のインタラクション

Dog-ear とはページの端を三角形に折りたたむ動作を指す．

提案インタフェースでは，ユーザがページのサムネイルの右上の角をクリックすることにより，Dog-ear を付けることができる．Dog-ear は大小 2 種類ある．そのため，大きい Dog-ear を大事なページに目印として付け，小さい Dog-ear をメモや付箋代わりにページに付けるといったような使い分けを行うことができる．図 3.12 にあるように，シングルクリックでは小さい Dog-ear を付けることができ，ダブルクリックでは大きい Dog-ear を付けることができる．ユーザが既に付けられた Dog-ear を再度クリックすると，付けられた Dog-ear はなくなり，ページのサムネイルは元に戻る．

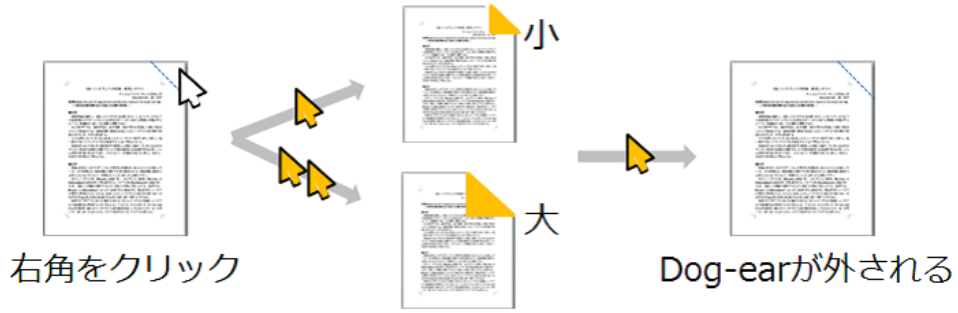


図 3.12: Dog-ear におけるインタラクション

ファイル中に Dog-ear が付けられたページがあると，その Dog-ear 付きのページがコンテンツビューが出現する際に最初に表示されるページになる．Dog-ear の付いたページが複数個ある場合，コンテンツビューに最初に表示されるページとして，Dog-ear の大小，ページ番号の若い順に優先される．

そのため，例えば，大きい Dog-ear 付きのページが 1 枚，小さい Dog-ear 付きのページが 2 枚あるファイルに対しては，コンテンツビューの最初に表示されるページは大きい Dog-ear 付きのページとなる．また，例えば，小さい Dog-ear 付きのページを 3 枚持つファイルに対しては，コンテンツビューの最初に表示されるページは，一番若いページ番号を持つ Dog-ear 付きのページとなる．

これにより，ユーザは自分が重要だと思ったページに素早くたどり着くことが可能となる．重要だと思うページの例として，レポートの後で直すべき部分を含むページや，参考文献の引用しようとするページなどが挙げられる．

ページの折り畳みのインタラクション

ページの折り畳みとは，ファイルのページのサムネイルを 1/4 のサイズに折り畳むことを指す．

ユーザがページのサムネイルをダブルクリックすると，クリックされたページは折り畳まれる．同様に，折り畳まれたサムネイルを再度ダブルクリックすると，折り畳まれたサムネイルは元に戻る．ページの折り畳みを用いて目立たなくするページの例として，レポートのタイトルページや謝辞，またはプログラムコードファイルの import 部分を含むページなどが考えられる．

このように，提案インタフェースではユーザはコンテンツビューを用いてフォルダやファイルの内容をひと目で把握することが容易となる．さらに，マークを付けることにより，ファイルのページに対して重要かどうかの重み付けを行うことができる．そのため，提案インタフェースは閲覧性に富むと言える．

3.4 提案するインタフェースによる利点

提案インタフェースの特徴として，以下の４つが挙げられる．

- 操作にはホバーとクリックを用いる
- フォルダを入れ子状に展開する
- ファイルの内容をファイルアイコンの近くに表示する
- ファイル操作を操作アイコンによって行うことができる

そのため，メリットとして以下の４つを挙げる．

- 操作方法を記憶する必要がなく，使い方を忘れても簡単に気付くことができる
- 同一画面上で，異なる階層間のファイルの移動を容易に行うことができる
- ホバーによってすぐにファイルの内容を閲覧し，確認することができる
- よく使用するファイル操作を，ファイルの内容を見ながらワンクリックで行うことができる

第4章　利用シナリオ

本研究で提案したインタフェースを利用する状況として様々な場合が考えられる．以下に，提案インタフェースのプロトタイプシステム Icons++を日常生活中で使用した利用シナリオを３つ紹介する．

4.1　ファイルの検索における利用例

Y さんは研究熱心な学生で，普段の研究時から論文をサーベイしては，気になった論文をダウンロードし，自分のコンピュータに保存していた．Y さんは忙しさに応じて，保存した論文をフォルダに分ける時もあれば，分けないでデスクトップに置く時もあった．会議に投稿するために原稿を書こうと思った Y さんは，参考文献を整理し，リストアップをする必要があった．しかし，分散されてフォルダ分けされた参考論文のファイルを探す際に，色々なフォルダを行ったり来たり，深くたどったりしなければいけなく，面倒であった．

そこで，Y さんは Icons++を立ち上げた．すると，フォルダのアイコンをホバーするだけでその場に中身が入れ子状に展開されるため，様々なフォルダの中にある論文ファイルを一覧することができた．Y さんは自分の集めてきた論文ファイルの一覧を見ながら，関連研究として載せられそうなファイルについてはアイコンをホバーし，コンテンツビューを使って論文の内容を確認した．こうして Y さんはフォルダをたどる試行錯誤をすることなく，スムーズに自分の原稿に使用できそうな関連研究のファイルを見つけることができた．

4.2　ファイル操作を行う際の利用例

議事録係の A さんは，午前午後で開かれた２つのミーテングが終わった後に，自分の記録した議事録を会議に参加したメンバーにメールで送ろうとしていた．A さんは会議のメンバーが議事録を確認しやすいように，文字化けしない PDF 形式でファイルを添付しようと思った．しかし，A さんが議事録を Word を用いて作成して保存していた．そのため，一度 Word を使ってファイルを開き,「名前を付けて保存」を選択して PDF に変えなければならなかった．この作業を２回も行わなければいけないことを考えて A さんは煩わしくなった．

そこで，A さんは Icons++を使用して議事録ファイルのあるフォルダを開いた．１つ目の議事録ファイルのアイコンをホバーすると，コンテンツビュー上にファイル操作を示した操作アイコンが見えた．A さんはそこで「PDF に変換する」操作アイコンを押した．同じ操作を

２つ目の議事録ファイルのアイコンに対しても行った．こうして A さんは簡単に同じ操作を繰り返し，素早くメールに添付する PDF 形式の議事録ファイルを作成することができた．

4.3　ファイルの内容を見比べる際の利用例

料理好きな H さんは，WEB 上で気になる料理を見つけては，レシピを PDF ファイルとして料理フォルダに保存していた．この度研究室の同期で打ち上げをすることになり，H さんが料理を振る舞うことが決まった．H さんは Icons++を使用して料理フォルダを開き，同期に喜ばれそうな料理を幾つか選ぼうとした．料理のレシピを一つ一つコンテンツビューを用いてチェックする際に，見直すときのことを考えて，ファイルの内容の材料について書いているページに Dog-ear を付けた．

買い出しをする段階になり，必要な材料のリストを作る必要が出た．そこで，H さんは Icons++を用いて各レシピファイルのコンテンツビューを出した．材料の載っているページに Dog-ear を付けていたため，コンテンツビューにはすぐに材料の箇所が表示された．また，コンテンツビューを出してはクリックによってコンテンツビューを固定したため，各レシピの材料を並べてチェックすることができた．こうして H さんは並んであるレシピの材料ページを一覧しながら，必要な食材と分量を簡単にまとめることができ，買い出しをスムーズに行えた．

第5章　プロトタイプシステムIcons++の実装

5.1　開発環境と言語

提案インタフェースのプロトタイプシステムとしてIcons++を開発した．開発にはC#を使用し，Windows OS上でアプリケーションとして実装している．Windows OSにデフォルトで入っているファイルマネージャのUIをベースに使用しているため，ユーザに馴染みやすいと期待される．

5.2　Icons++の全体像

図5.1はプロトタイプシステムのIcons++の全体像を示している．

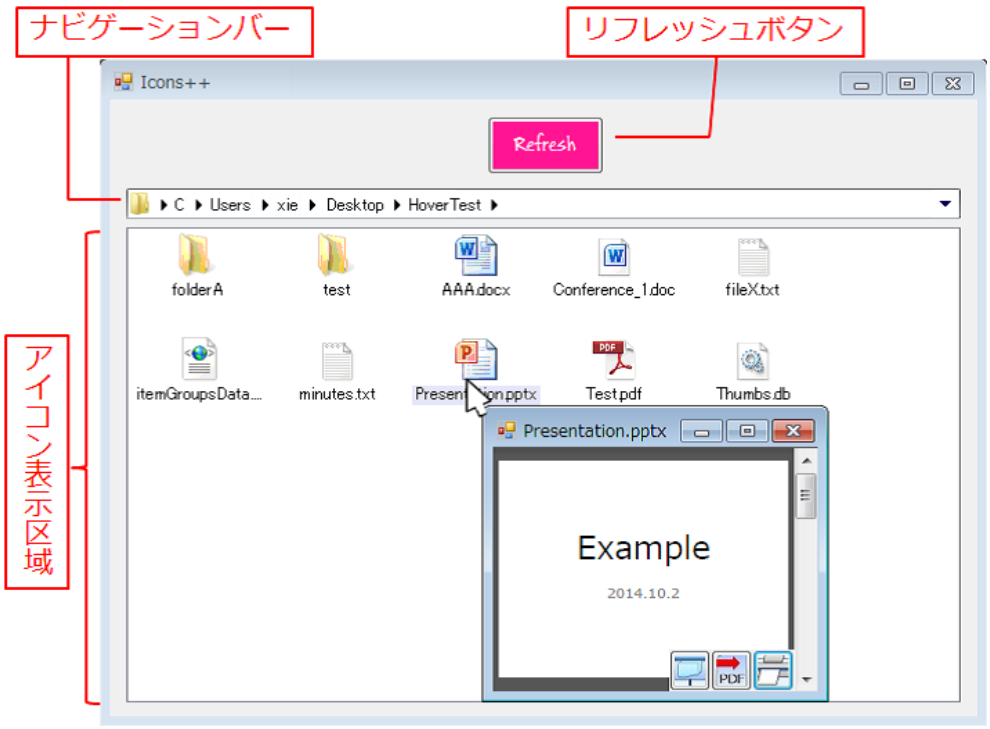


図5.1: Icons++の全体的なUI

ユーザがIcons++を起動すると，図5.1のようなウィンドウが表示される．ナビゲーション

バーを用いることにより，ユーザは現在自分がどのフォルダを見ているのかを把握することができる．また，多数のアイテムに対して表示された多数のコンテンツビューは，リフレッシュボタンを押すことにより，一度に消すことが可能となる．フォルダとファイルのアイコンは，アイコン表示区域に表示される．

図 5.1 では，ユーザがパワーポイントファイルのアイコンをホバーし，そのファイルに対してコンテンツビューが表示されたところを示している．

5.3 フォルダ展開の実現

5.3.1 フォルダ内のアイテム数を用いたサイズの計算

フォルダ展開の際，フォルダのコンテンツビューのサイズは中身のアイテム数によって変化する．Icons++ではユーザがフォルダをホバーした際に，コンテンツビューを表示する前の段階で，フォルダの中身のアイテム数を用いてコンテンツビューのサイズを計算する．

図 5.2 はアイテムの並べ方とコンテンツビューのサイズの例を示している．

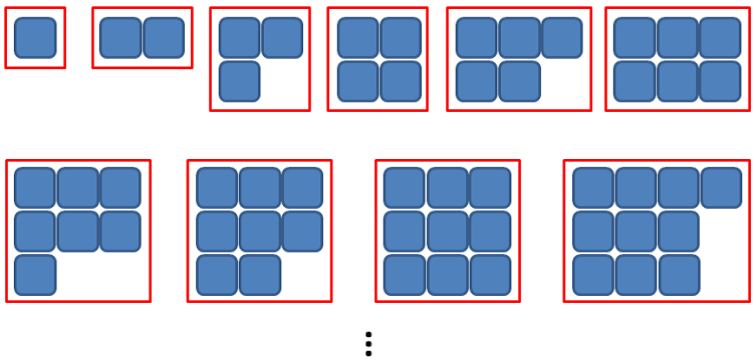


図 5.2: アイテムの並べ方とコンテンツビューのサイズ

図 5.2 において，塗りつぶされた四角はアイテムを表し，塗りつぶされていない赤枠はコンテンツビューのサイズを表している．例として，アイテムが 1 〜10 個の場合のアイテムの並べ方とコンテンツビューのサイズを図 5.2 に示す．

複数フォルダを入れ子状に展開する場合，何層にもなる入れ子の中に各層のアイテムを表示する必要がある．そのため，できるだけ小さく面積を取るようなアイテムの並べ方が求められる．Icons++では，アイテムの並べ方をできるだけ正方形に近づけようとしている．

Icons++においてコンテンツビューのサイズを計算する際に，まずアイテム数の平方根を取ったものを整数に切り上げる．これが横に表示できるアイテム数の上限となる．縦に表示できるアイテム数の上限は，横のアイテム数を全体のアイテム数で割ったものを，さらに整数に切り上げて計算する．これにより，横と縦それぞれについてアイテムの数が分かるので，アイテムのアイコンの大きさとアイコンとアイコンの間隔を計算に用いて，コンテンツビューの横と縦の大きさを求める．

例えばアイテム数が 10 の場合では以下のように計算される．
横のアイテム数：
10 の平方根 = 3.162277...
3.16 を整数に切り上げる = 4
縦のアイテム数：
10/4 = 2.5
2.5 を整数に切り上げる = 3
したがって，アイテム数が 10 個の場合，横に 3 行と縦に 4 列アイテムを配置すれば良いことがわかる．

5.3.2　フォルダのサイズ調整の流れ

入れ子状にフォルダが展開されるとき，サブフォルダに対するコンテンツビューのサイズに応じて，親フォルダのコンテンツビューのサイズが変化する．その際に，以下の図 5.3 にあるような処理の流れに沿って親フォルダのサイズ変更を行っている．

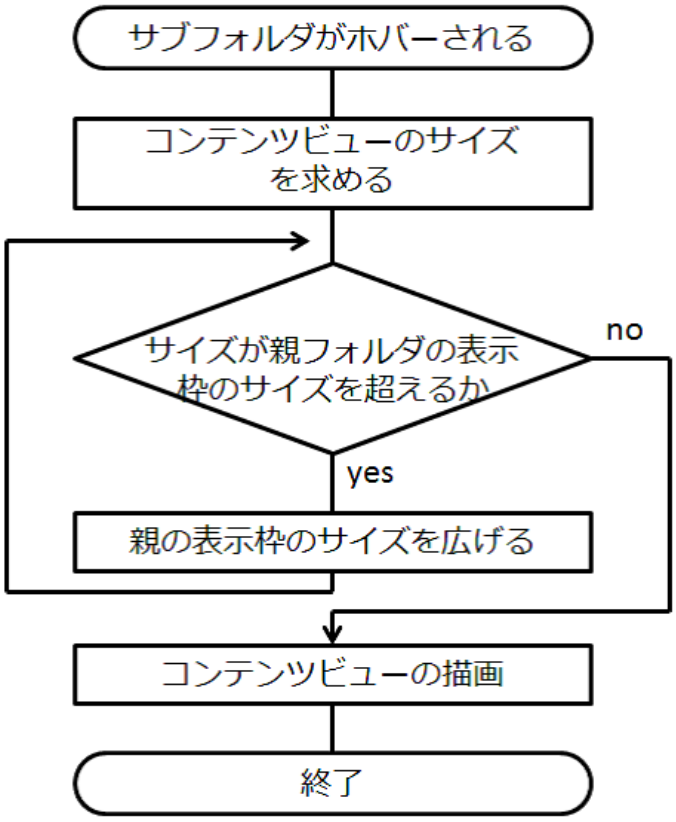


図 5.3: サブフォルダの表示による親フォルダのサイズ変更時のフロー

図 5.3 にあるように，サブフォルダに対してコンテンツビューが表示される場合，表示され

る前にサブフォルダのコンテンツビューの枠の大きさが親フォルダのコンテンツビューの枠の大きさを超えるかどうかがチェックされる．超えなかったらそのままコンテンツビューの描画に進む．しかし，超えた場合には，親フォルダのコンテンツビューの枠の大きさを広げる．その際，もし親フォルダが他のフォルダのサブフォルダにあたる場合，さらにその他のフォルダ（最初のサブフォルダにとっては２階層上のフォルダ）のコンテンツビューの枠の大きさも広げる．

このようにして，ホバーされたサブフォルダから一層ずつ上の層のフォルダについてチェックしていき，全体のファイルマネージャ内での入れ子状に表示されるコンテンツビューの大きさが決められる．

5.3.3　フォルダ展開画面におけるファイルの移動

フォルダのコンテンツビューは，対応するフォルダのフルパスを所持している．ファイルのアイコンをドラッグ&ドロップし，あるフォルダのコンテンツビュー内に移動させた場合，ドラッグ&ドロップされたファイルの所属フォルダのパスを移動先のフォルダのパスにすることにより，複数フォルダ間の同一画面におけるファイルの移動を可能にしている．

5.4　ファイルの内容表示

5.4.1　ファイルのコンテンツビューの実現

本インタフェースではシステム内に各ファイルの各ページに対応する JPEG ファイルを保持している．以降，このファイルに対応して保持された JPEG ファイルのことを「キャッシュ」と呼ぶ．

まず，キャッシュの作成について述べる．図 5.4 にあるようにシステムは予め，ユーザの持つ各ファイルについて PDF 変換を行い，各ファイルに対応する PDF ファイルを作成する．そして，PDF の各ページについて JPEG ファイルを作成する．キャッシュの作成は，ユーザが Icons++を用いて表示するフォルダを指定した際に行う．

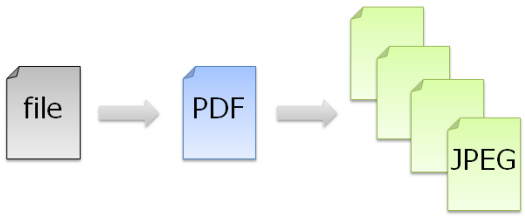


図 5.4: ファイルに対する JPEG ファイルの作成

次に，キャッシュを用いたファイルの内容表示について述べる．

ユーザが特定のファイルのアイコンをホバーした時，そのファイルに対応する JPEG ファイルを呼び出すことにより，ファイルの内容を表示している．図 5.5 では内容表示の際のウィンドウのレイヤーを示している．

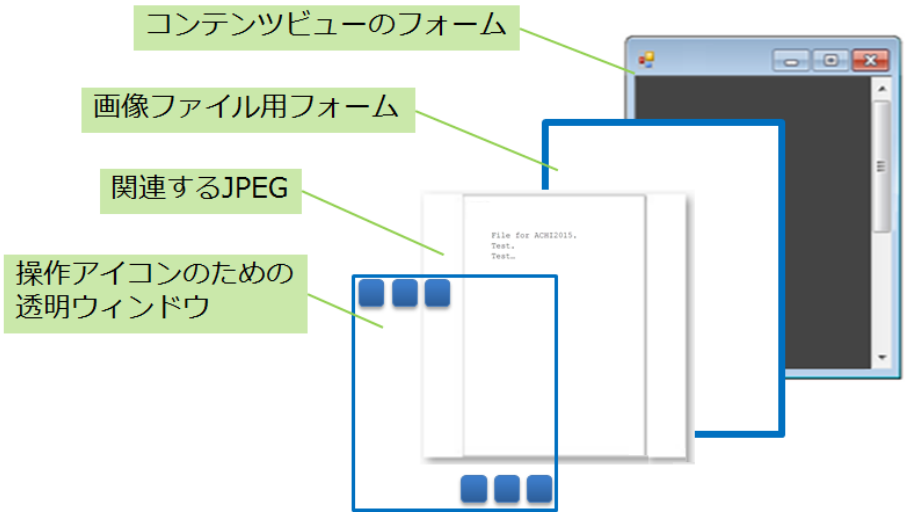


図 5.5: ファイルの内容表示の実現

図 5.5 で示されたように，ファイルの内容を表示する際，システムではまずコンテンツビューのウィンドウとして，Windows のウィンドウフォームを表示する．そして，ウィンドウの上に画像ファイルを表示できるフォームを重ねる．さらにそのウィンドウ上に JPEG ファイルを表示し，最後にファイルの操作のための操作アイコンを載せた透明ウィンドウを上に被せる．

5.4.2　素早い内容表示と表示の安定性

Icons++ではシステム内に，ユーザのファイルに対応する JPEG をキャッシュとして保持している．そのため，ユーザが特定のファイルをホバーした際には，JPEG を表示することによりファイルの内容を見せている．これは，ファイルをダブルクリックしてアプリケーションで開いて閲覧するよりも素早く内容の閲覧を可能にしている．

新たにファイルが作成された場合には，ユーザがその新たにフォルダに追加されたファイルをホバーしたときに，ファイルに対してキャッシュを作成し，ファイルの内容表示として JPEG の表示を行う．そのため，容量が大きいファイル（e.g., 20 スライドを持つ PowerPoint ファイルなど）に関しては，作成する際に何秒かの待ち時間を要する．

また，Icons++では画像を使用してファイルの内容を表示しているため，元のファイルの形式に関わらず，文字化けを防止することができる．

5.5　ファイルの操作

ファイルの操作では，操作アイコンに対応するコマンドが予めシステム内に登録されている．ユーザが操作アイコンをクリックした際に，登録されたコマンドを実行することにより，ファイルの操作を実現している．カスタマイズ操作では，一連のショートカットキーやメニュー選択などのユーザの行う操作を記録する．記録した操作を再現することにより，ファイルのカスタマイズ操作を実現している．

5.6　Dog-ear とページの折り畳みの実現

5.6.1　サムネイル画像の編集

ページに対して Dog-ear を付ける，あるいはページを折り畳みたいときに，ユーザはページのサムネイルに対してシングルクリックあるいはダブルクリックのマウスイベントを起こす．

ダブルクリックは大きい Dog-ear を作るときでも，ページを折り畳むときでも使用されるマウスイベントである．そこで，Icons++では，サムネイルの右上端をダブルクリックされたら Dog-ear，それ以外の全体をダブルクリックされたらページの折り畳みだと見なしている．

コンテンツビューにおいてユーザが特定のページに対して Dog-ear を付けたり，ページを折り畳むためにマウスイベントを起こした際には，Icons++ではまずユーザのクリックしたページがどの JPEG ファイルに当たるかをチェックする．続いて，該当 JPEG ファイルに対して，付けられたマークに応じて操作を行う．Dog-ear が指定されているならば，JPEG に対してオレンジ色の三角形の画像を追加する．ページの折り畳みが指定されているならば，JPEG を 1/4 のサイズに縦に縮小する．このとき，元の JPEG ファイルは削除せずに保持したままにする．これは，マーク付けが解除された際に，元の画像とマークの付いた画像の切り替えを行うことにより，即座にユーザにマークの解除を視覚的に見せることが可能となるためである．

5.6.2　Dog-ear とページの折り畳みの再現

システムによって作られた Dog-ear が付いた JPEG と折り畳まれた JPEG（以降マーク付き JPEG と呼ぶ）はそれぞれシステム内に収められる．ユーザのクリックによってマーク付き JPEG が作成されたときに，コンテンツビューにおいて，マーク付き JPEG とクリックされたページの JPEG を入れ替えて，コンテンツビューの再描画を行う．

また，ユーザがあるファイルについてコンテンツビューを開いたときに，まずシステム内でマーク付き JPEG が収められた場所を検索する．もしホバーされたファイルに関するマーク付き JPEG がある場合には，マーク付き JPEG と元の JPEG を差し替えてコンテンツビューに表示する．さらに，Dog-ear 付き JPEG がある場合には，Dog-ear 付き JPEG がコンテンツビューの最初に表示されるページになるように，自動スクロールして表示位置をずらす．一つのファイルについて複数の Dog-ear 付きページがあるときには，ページ番号の若いものをコンテンツビューで最初に表示されるページとする．

ユーザがマーク付きページにおいて，マークを外した場合には次の 2 ステップの操作を行っている．まず，ユーザの見ているコンテンツビューにおいて，元のマークのついていない JPEG とマーク付き JPEG を差し替え，コンテンツビューの再描画を行う．次に，マークを外されたページについて，対応するマーク付き JPEG を破棄する．これにより，次回にユーザがそのファイルをホバーし，コンテンツビューを表示させてもマーク付きページは表示されなくなる．

第6章　関連研究

6.1　フォルダの中身の表示方法に関する研究

フォルダの中身の表示方法に関して，階層構造を視覚的に示したものとしてツリー表示やカラム表示がある．

また，ファイルを階層的にフォルダ分けせず，まとまりとして捉える研究もされている．例えば，実世界の机上に紙のファイルや書類を積み重ねるのと同じインタフェースをデジタルに実現している研究として，Mander らが提唱している Pile Metaphor [12] と Agarawala らの Keepin'it real [13] がある．また，Watanabe らは Bubble Clusters [15] において，ファイルをフォルダ分けせずに，ファイルのアイコンをデスクトップ上にまとめて表示する手法を提案している．

Hakala ら [14] はフォルダやサブフォルダを線で繋いだ状態で二次元に表示し，斜め上からフォルダの入った引き出しを覗くようなインタフェースを提案している．Hakala らの開発した Space Manager では，フォルダが開かれると，フォルダのアイコンに垂直になるように中身のファイルのサムネイルアイコンが表示される．

本研究では，ユーザへの親しみやすさを考慮し，既存のファイルマネージャに多く使用されているツリー構造のフォルダの階層構造を使用している．そして，フォルダの表示の際に階層関係が視覚的に分かるように入れ子状のに中身のアイテムを表示している．

6.2　入れ子状の要素展開に関する研究

入れ子状にアイテムを表示する手法はいくつかの研究中で述べられている．

Johnson ら [16] の研究では Venn diagram や Nested treemap といった階層構造になっている要素を入れ子状に可視化する方法が述べられている．また，Shneiderman [18] の研究でもツリー構造を入れ子を用いて表示する手法が述べられている．

Engdahl ら [19] はモバイルデバイスなどの小画面でもツリー表示は有用だと示し，ツリー表示の発展として，入れ子状に情報を提示する手法を提案している．三末ら [20, 21] は D-ABDUCTOR という図的発送支援システムを開発し，ツリー構造において，階層間を線で繋いだ表示方法と入れ子状の表示方法の間の表現形式の変化操作について述べている．Weiland らの Facet Folders [17] ではフォルダの中身のツリー表示に対して，メタデータ（時間や場所など）を用いてフィルタリングを行う．フィルタリング結果を示すとき，入れ子状にファイルのサムネイルを表示している．

本研究では，フォルダの展開時に入れ子状の表示方法を使用している．また，入れ子状の表示では，入れ子はフォルダの階層構造にしたがって作られる．

6.3 ホバーによる情報提示に関する研究

情報を表示するきっかけとして，ホバーという動作を使用した研究がある．例えば，Fitchett らが提案しているファイルマネージャにおける Hover Menu [22] がある．Hover Menu では，ユーザがフォルダのアイコンをホバーすると，フォルダの内容の中で最近使用されたサブアイテムのリストがフォルダアイコンの近くに表示される．また，Terry らは Side Views [23] というシステムを提案した．Side Views では，ユーザが特定のメニューの選択肢をホバーした際に，その選択肢を実際に選択した場合に得られる実行画面のプレビューが表示される．Matejka らによる CommunityCommands [24] や Voida らによる Share and Share Alike [25] では，アイコンをホバーされたファイルについての付加情報が表示される．付加情報として，協調作業を行っているメンバーが書き残したメモやファイルを共有しているメンバーのリストなどが挙げられる．Alexander らによる Revisiting Read Wear [26] はメニューに対するホバーではなく，スクロールバーに対するホバーをシステム内で用いている．Alexander らのシステムでは，ユーザがファイルのスクロールバーにマウスカーソルを当てると，各ページのサムネイルが表示される．さらにサムネイルをホバーすると，サムネイルが拡大され，大きい画像を見ることができる．

これらの研究は本研究の提案インタフェースとはホバーという動作を用いて情報を表示するという点で似ている．しかし，提案インタフェースでは情報の表示だけでなく，情報を用いた操作（e.g., コンテンツビューにおけるファイル操作やアプリケーションへのアクセス）も可能にしている点で異なる．

6.4 素早い操作の実現に関する研究

ユーザが望む操作を素早く行えるようにすることに対し，Appert ら [27] はメニュー選択におけるショートカットとして，ストロークジェスチャを用いた方法を提案した．しかし，ジェスチャを用いたショートカットでは，ユーザは複数種類のジェスチャを記憶する必要がある．また，Ephemeral Adaptation [28] や Bubbling Menus [29] のようなメニュー選択時にメニューのアイテムを強調する研究もある．これらの研究ではユーザの求めるメニューのアイテムは見つけやすくなるが，メニューからアイテムを探し出すまでに複数ステップを経る必要がある．増井ら [30] はユーザが繰り返している操作を自動的にマクロと定義し，実行キーを押すと操作を実行するシステムを開発した．

これらの研究に対して，本研究はユーザがよく使用する操作を素早く行えるようにしたという点では同じであるが，操作の際に操作をイラストで示したアイコンをクリックするマウスイベントを用いた点で異なる．本研究のアイコンを用いた方法では，ユーザはファイルの

内容を閲覧しながら 1 ステップで希望の操作を行うことができる．また，イラストを用いたアイコンを使用しているため，何の操作であるかが絵で示されるようになっている．

6.5 ファイル操作時のアイコン利用に関する研究

いくつかの研究で，操作の際にアイコンを利用することが挙げられている．Block らによる Touch-Display Keyboards [31] では，プロジェクターを用いてコンピュータのキーボード上に操作アイコンを映し出し，ファイルの操作を可能にしている．ユーザはキーボードのキーを押して操作を選択することができる．Yeh らによる Sikuli [32] や Chang [33] の研究では，スクリーンショットをベースにしたスクリプト中でアイコンを使用している．ユーザはスプリクト中にアイコンとして表示されているスクリーンショットを用いて，マウスやキーボードイベントを行うことができる．Scarr ら [34] はユーザがファイル操作についてのコマンドを素早く行えるように，アプリケーションウィンドウに重ねて表示される CommandMaps を開発した．CommandMaps ではユーザはそのアプリケーションで使用可能なコマンドを一覧することができる．

これらの研究と異なり，提案インタフェースは従来親しまれたマウス操作を利用する．さらに，提案インタフェースでは一つのファイルが対象になるだけでなく，ファイルマネージャの同じウィンドウ内にある複数のファイルに対して操作を行うことが可能である．

6.6 素早いアイテム検索に関する研究

ユーザが自分の望むアイテムやファイルを素早く見つけられるようにするために，いくつかの手法が既存の研究で提案された．一つの手法として，ファイルアイコンやファイルのサムネイルに何かしらの効果を付けて，対象を目立たせるものがある．例えば，Fitchett らの Finder Highlights [35] では，ファイルマネージャで表示されたファイルのアイコンにハイライトが付けられる．テキストやテキストのサムネイルに対して色とりどりのマーカーを引くことでハイライトを付け，アイテムを目立たせた研究として Popout Prism [36] と Woodruff [37] らの研究がある． また，Dog-ear を付けることにより，対象を目立たせる方法もある．NiCEBook [38]，WebView [43]，Kaasten らによる研究 [39]，Hoeben らによる研究 [40] では Dog-ear が用いられた．

ファイルの内容のサムネイルにではなく，ファイルのアイコンに対して情報を追加して，視覚的にアイコンを目立たせる研究もある．小澤ら [41] の研究ではファイルマネージャ内でアイコンに対して付箋を貼ることにより，ファイルやファイル中の重要なページをユーザに見せている．また，Setlur らによる Semanticons [42] ではシステムのアイコンの UI に対して，ファイルの内容に関連したイラストを付加してユーザに見せている．

これらの研究と異なり，本研究では対象を目立たせるたけでなく，必要のない情報を目立たなくする方法も挙げている．

6.7 関連するアプリケーション

提案インタフェースと関連したアプリケーションもいくつか挙げられる．例えば，Mac OS の Quick Look 機能では，ユーザはファイルの内容を閲覧することができる．しかし，Quick Look 機能では閲覧のみできるため，ファイル操作は行うことができない．さらに，Quick Look を利用する際の動作はキーボードのスペースキーを押すことである．しかし，スペースキーによるアクセスは知識がなく初見のユーザにとっては自然に気付くことが困難な動作である．本研究ではユーザが自然に気付く可能性が高いホバーという動作によるアクセスを用いている．また，別の例として Gmail のプレビュー機能が挙げられる．Gmail のプレビューと本研究との違いとして文字化けへの対処が挙げられる．Gmail のプレビュー機能では，ファイルの形式によっては文字化けが発生する時がある．一方で，本研究の提案インタフェースではファイルに対応する PDF ファイルと画像ファイルを保持しているため，文字化けを防ぐことができる．

第7章　評価実験

Icons++のメリットについてより良く理解し，さらに既存のファイルマネージャの UI と比較するために，２つのユーザスタディとアンケートを実施した．

7.1　実験目的

評価実験を通して，Icons++と既存の UI を比較し，Icons++のメリットやデメリットについて把握することを目的とする．

7.2　被験者

被験者は 22 歳〜27 歳（平均 24 歳）大学生・大学院生の計７名である．そのうち２名は女性，残り５名は男性である．また，被験者は７人とも普段からコンピュータを使用し，ファイルマネージャを用いたファイル操作に十分慣れている．

7.3　実験手順

まず初めに，被験者に対して説明なしで Icons++を３分間使用してもらった．その後，Icons++の使い方や機能について説明し，Icons++の使い方に慣れてもらうためにさらに３分間 Icons++を使用してもらった．

被験者が Icons++に慣れてから，２つのユーザスタディを行ってもらった．最後に，Icons++の使用やユーザスタディを踏まえて，被験者にアンケートに答えてもらった．図 7.1(a) は被験者が実験を行っている様子である．以下に２つのユーザスタディとアンケートについて説明する．

7.4　実験内容

7.4.1　ユーザスタディ1：特定の文字を持つファイルの検索

このユーザスタディの目的は Icons++と Windows 7 にあるファイルマネージャを比較することである．ユーザスタディ１のタスクは，システムによって指定された特定の文字を内容に持つファイルを探し出すことである．

実験に際して，検索対象のファイルとして図 7.1(b) のように，1 文字のみを内容に持つファイルを多数用意した．

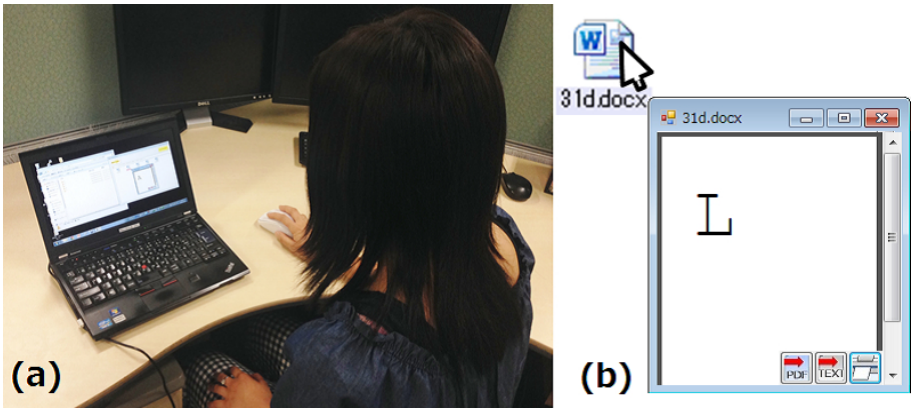


図 7.1: (a) 被験者が実験を行う様子 (b) 実験のために用意した 1 文字のみを内容に持つファイル

ファイル検索はファイルのアイコンが見える画面から開始する．被験者にはファイルのアイコンに対して，以下 3 つの方法を用いてアクションを起こしてもらい，システムの指定するファイルを探してもらった．

- ダブルクリックでアプリケーションを開き，ファイルの内容を確認する
- Windows 7 のファイルマネージャにあるプレビュー機能を用いて，ファイルの内容を確認する
- Icons++でファイルのアイコンをホバーして，ファイルの内容を確認する

各方法を用いて，被験者はそれぞれ 5 回の試行を行った．1 回の試行において，被験者は 5 つのファイルの中から指定された文字を持つファイルを見つけ出す必要があった．そのため，被験者は 5 つのファイル × 5 回の試行 × 3 つの方法 (合計一人あたり 75 ファイルに対してファイル検索を試行) を行った．それぞれのファイル検索タスクにかかった時間はシステムによって自動的に記録された．

7.4.2 ユーザスタディ 2：スライドショーを用いた PPT ファイルの閲覧

このユーザスタディの目的は，Icons++を用いた方法と既存のファイル操作のやり方を比較することである．ユーザスタディ 2 では，ユーザは同じファイル操作を行うのに以下の 2 種類の方法を用いて行った．

- 被験者自身が通常使用する方法を用いてファイル操作を行う
- Icons++でファイルのコンテンツビュー上に表示される操作アイコンを用いてファイル操作を行う

このユーザスタディのタスクは，特定の PowerPoint ファイルをスライドショーで閲覧することである．タスクはファイルのアイコンが見える画面から開始させた．

被験者自身が通常使用する方法に関しては，実験者から指定や言及はせず，制限は設けなかった．被験者が使用し得る方法として，以下のいくつかが想定された．

方法１ PowerPoint でファイルを開いた後にメニューをたどり，スライドショーを選択する

方法２ PowerPoint でファイルを開いた後にウィンドウ下部に表示されたスライドショーのアイコンを押す

方法３ PowerPoint でファイルを開いた後にショートカットキー（F5）を押す

方法４ ファイルのアイコンを右クリックし，出現するコンテキストメニュー中からスライドショーの項目を選択する

被験者は Icons++を使用する方法と自分の通常使用する方法をそれぞれ用いて，一回ずつタスクを行う．また，それぞれのファイル操作のタスクにかかった時間はシステムによって自動的に記録された．

7.4.3 アンケート

全てのユーザスタディを終わったあとに，被験者に Icons++の使用感についてのアンケートに回答してもらった．アンケートの回答欄では，５段階のリッカート尺度と回答の理由を記述する欄を設けた．リッカート尺度では，１が最も評価が低く（「そう思わない」に相当），５が最も評価が高い（「そう思う」に相当）．

付録 A にあるように，まずユーザスタディ 1 に関して，被験者にそれぞれの手法についての使い心地を尋ねた．設問としては，各手法のアクセス方法が説明がなくても自然と使い方に気付くかどうかを設定した．

続いて，被験者に対して，Icons++の使い心地とこれからも使っていきたいと思うかについて尋ねた．設問としては以下のように設定した．

(a) ホバーによるファイルの内容表示を使いやすいと感じるか
(b) 操作アイコンを用いたワンクリックによるファイル操作を使いやすいと感じるか
(c) Icons++の全体的な UI を使いやすいと感じるか
(d) 今後もホバーによるファイルの内容表示を使っていきたいか
(e) 今後もワンクリックによるファイル操作を使っていきたいか
(f) 今後も Icons++の全体的な UI を使っていきたいか

また，各設問に加え，アンケートの最後に被験者が自由に意見を書くことができる欄を設けた．

7.5　実験結果

２つのユーザスタディ及び被験者に回答してもらったアンケートの結果を以下に示す.

7.5.1　ユーザスタディ１の結果

図 7.2 はユーザスタディ 1 における 7 名の被験者の 3 つの方法それぞれを用いた際の平均達成時間のグラフである．ここで，１回の試行とは被験者が 5 つのファイルの中からシステムに指定された 1 つのファイルを探すことを指している.

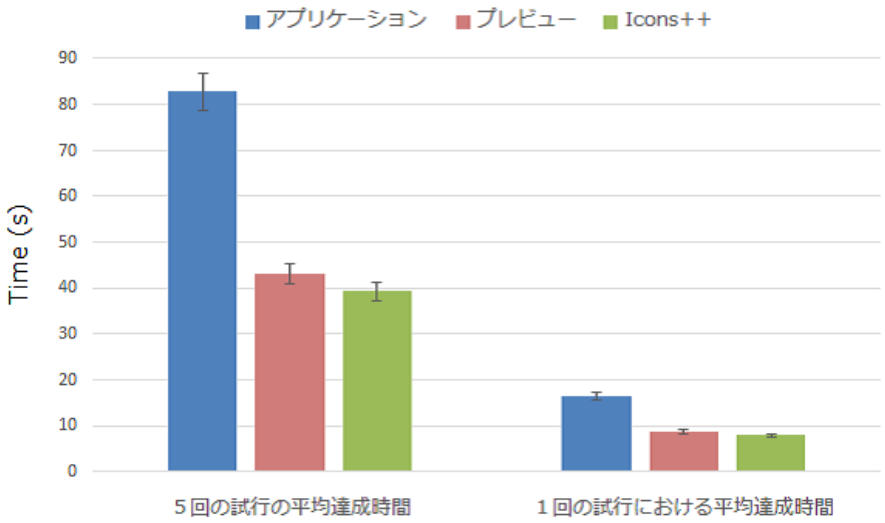


図 7.2: 7 名の被験者のアプリケーション・プレビュー機能・Icons++を使用してファイル探索をする際の平均達成時間

図 7.2 の左側のグラフは，それぞれの方法（アプリケーションを用いてファイルを開く・プレビュー機能を用いてファイルの内容を閲覧する・Icons++のホバーによる内容表示を用いてファイルの内容を閲覧する）について 7 名の被験者が 5 回の試行を行った際の，５回の試行の達成時間の平均値を表している．また，図 7.2 の右側では，それぞれの方法を用いた際の各試行における達成時間の平均値を表している.

Icons++においてファイルのアイコンをホバーし，ファイルの内容を閲覧する方法を用いた試行では，５回の試行の平均達成時間は 39.2 秒（s.d. 5.7）であった．これはアプリケーションを用いてファイルを開く手法を用いた試行の平均達成時間（平均 82.7 秒，s.d. 18.1）に比べて 53%速かった．したがって，この結果に基づいて，Icons++を用いた試行の達成時間とダブルクリックによってファイルをアプリケーションで開く方法を用いた試行の達成時間において有意差が見られた（$T(12) = 5.6, p < .001$）.

一方で，Icons++においてホバーを用いてファイルの内容を閲覧する方法における 5 回の試行の平均達成時間は，Windows7 のファイルマネージャにあるプレビュー機能を用いる方法（平均 43.1 秒， s.d. 8.6）に比べて 9%速かった．したがって，Icons++を用いる方法とプレビュー機能を用いる方法では有意差は見られなかった（T(12) = 0.7, p= 0.2）.

7.5.2 ユーザスタディ 2 の結果

図 7.3 は 7 名の被験者がユーザスタディ 2 を行った際の平均達成時間を示している.

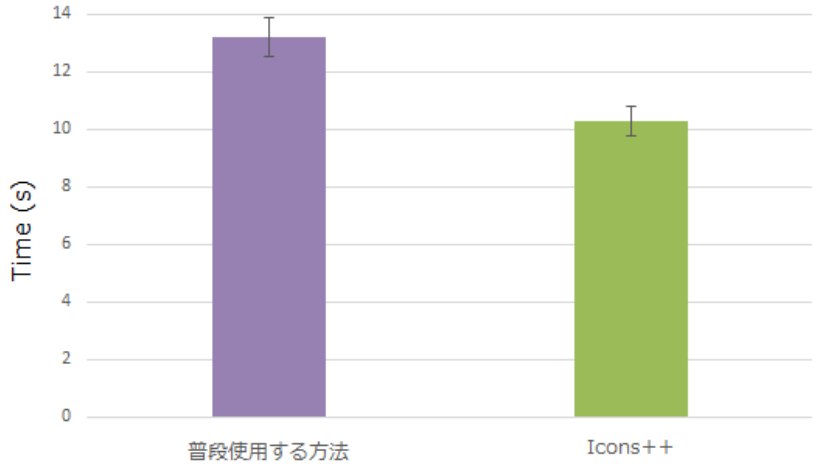


図 7.3: 7 名の被験者の Icons++と普段使用する方法を用いた PowerPoint ファイルをスライドショーで閲覧するまでの平均達成時間

被験者が Icons++を用いて PowerPoint ファイルをスライドショーで閲覧するまでに掛かった平均時間は 10.3 秒である．これは，普段使用する方法を用いた方法（平均時間 13.2 秒）よりも 22%速い（T(12) = 0.9, p = 0.19）.

普段使用する方法を用いて PowerPoint ファイルをスライドショー方式で閲覧する際に，被験者はファイルを PowerPoint を用いて開いてから，スライドショーを表すアイコンを選択した．また，PowerPoint のメニューをたどり，スライドショーで表示する項目を選んだ被験者もいた．さらに，PowerPoint におけるショートカットキー（i.e., F5）を用いてスライドショーで表示させた被験者もいた．しかし，ファイルマネージャ内でファイルのアイコンを右クリックし，コンテキストメニューからスライドショーを選択する被験者は一人もいなかった.

7.5.3 アンケートの結果

アンケートの設問と回答結果をまとめたものを表 7.1 に示す.

表 7.1: アンケートの回答結果（各設問の人数と回答の平均値）

設問	(1) そう思わない (人)	(2) あまりそう思わない (人)	(3) どちらでもない (人)	(4) 少しそう思う (人)	(5) そう思う (人)	平均値
ユーザスタディ 1 で使用した 3 つの方法について						
アプリケーションで開く方法は直感的か	0	1	1	1	4	4.1
ファイルマネージャのプレビュー機能は直感的か	2	1	2	1	1	2.7
Icons++のホバーによる内容表示は直感的か	0	0	0	4	3	4.4
Icons++のUIについて						
ホバーによる内容表示は使いやすいか	1	0	0	4	2	3.9
ワンクリックによるファイル操作は使いやすいか	0	0	2	3	2	4
全体的なUIは使いやすいか	0	1	2	0	4	4
ホバーによる内容表示を今後も使っていきたいか	1	0	0	3	3	4
ワンクリックによるファイル操作を今後も使っていきたいか	0	0	2	2	3	4.1
全体的なUIを今後も使っていきたいか	0	0	4	0	3	3.9

表 7.1 では，各設問における回答項目ごとの人数と，設問ごとの回答の平均値を示している．回答の平均値は，回答項目の値 × 回答人数を回答項目ごとに求め，その和を，被験者数で割った値である．

また，アンケート前半でユーザスタディ 1 で使用した 3 つの方法について尋ねた設問に対する回答の平均値をグラフにし，図 7.4 に示す．

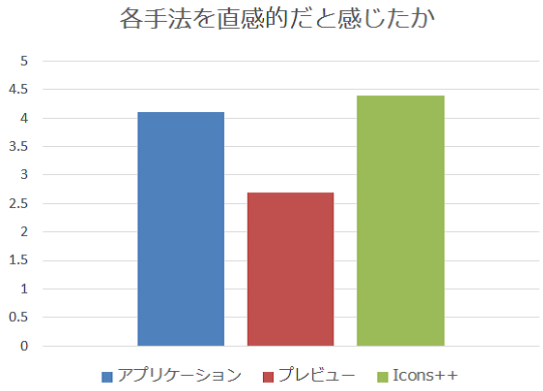


図 7.4: ユーザスタディ 1 で使用した 3 つの方法の直感性に対する回答の平均値

以降，表 7.1 の回答人数について述べる際には，回答項目で 3（どちらでもない）以上を選択した人数を好意的に感じた人数として捉える．

表 7.1 を見ると，ユーザスタディ 1 で使用した 3 つの方法についての設問に関しては，アプリケーションでファイルを開く方法を 7 名中 6 名の被験者が直感的だと感じた．また，プレビュー機能を使用した方法については 7 名中 4 名が直感的だと感じた．最後に，Icons++におけるホバーを使用した方法については 7 名全員が直感的だと感じた．

図 7.4 では左からアプリケーションを使用した方法，プレビューを使用した方法，Icons++のホバーによる内容表示を使用した方法，についてそれぞれが直感的であるかどうかの設問に対する回答の平均値を示している．これを見ると，アプリケーションを使用した方法では平均値が 4.1 であり，Icons++を使用した方法における平均値の 4.4 と大差なく，どちらも直感的だと好意的に捉えられたことが分かる．それに対して，プレビュー機能を使用した方法では平均値が 2.7 となっており，直感的だと捉えられなかったことが分かる．これは，Windows 7 のファイルマネージャにあるプレビュー機能は，プレビュー機能を示すアイコンを押すことにより使用可能になるのだが，被験者が自分から自然にそのアイコンに気付くことが容易ではないためだと考えられる．

これにより，ユーザスタディ 1 の結果において Icons++を使用した方法とプレビュー機能を使用した方法では，試行時間において有意差はなかったが，被験者は操作性の点からプレビュー機能よりも Icons++を好むことが分かる．

続いてアンケート後半の Icons++の UI に関する設問の回答結果について述べる．

使いやすさについて尋ねた設問では，7 名中 6 名の被験者がホバーによる内容表示を使いやすいと感じた（回答の平均値は 3.9）．また，7 名全員が操作アイコンを用いたワンクリックによるファイル操作を使いやすいと感じた（回答の平均値は 4 .0）．全体的な UI に関しては，7 名中 6 名の被験者が使いやすいと感じた（回答の平均値は 4.0）．

今後も使っていきたいかどうかについて尋ねた設問では，ホバーによる内容表示については 7 名中 6 名が使っていきたいと回答した（回答の平均値は 4.0）．また，操作アイコンを用いたワンクリックによるファイル操作については 7 名全員が使っていきたいと回答した（回答の平均値は 4.1）．全体的な UI に関しても 7 名全員が使っていきたいと回答した（回答の平均値は 3.9）．

被験者による自由記述では，Icons++について好意的な意見が多かった．

例えば，ホバーによる内容表示に対しては,「簡単に内容を見られるのは便利」「特別な操作をしなくても表示されるので使いやすい」「カーソルを乗せただけで表示されるのは良い」「時間の節約になる」といったコメントがあった．

また，操作アイコンを用いたワンクリックによるファイル操作については,「使う操作はある程度決まっているので，作業が早くできると思った」「使いやすそう」「アプリケーションを起動せずに操作できるから良い」といったコメントがあった．

全体的な UI に関しては,「アプリを起動するまで時間がかかるが，簡単に操作できるのは良い」「誤操作も最初からしなかったから，多くの人が使えると思った」「直感的」「説明なしでわかった」「新鮮な感じ」といったコメントがあった．

これにより，多くの被験者は，使い方を学習したり記憶したりする必要がなく，直感的な

UI であるから，Icons++は使いやすいと感じていることが分かった．

一方で，Icons++に対してネガティブな意見もあった．ホバーによる内容表示に対しては，「内容を消す&ウィンドウを固定する方法がもっと使いやすくなると良い」「結構いいけど，ホバーしたときに右下のアイコンが見えなくなるのが少し気になった」「中身が小さくて見えない」「重くなかったら良い」といったコメントがあった．

また，操作アイコンを用いたワンクリックによるファイル操作をについては，「通常のプレビュー表示と右クリックメニューの併用との差分が分からない」「すぐに操作したいときには便利だが，右クリックの方が馴染みがある」「機能が少ない」「使いたい場面があまり思い浮かばない」といったコメントがあった．

全体的な UI に関しては，「普段使っているアプリケーションの UI と若干違うため，少しとまどう」という意見があった．

２名の被験者は Icons++の欠点を，コンテンツビューの表示によって下段のファイルのアイコンが見えなくなることだと指摘した．なぜならば，コンテンツビューはそれまでに表示されていた下の段のファイルアイコンにかぶさって表示されるためである．また，ある被験者は，本当にファイルの内容を見たいと望む場合，コンテンツビューを用いて見ることができるものでは小さ過ぎて，内容の確認がしづらいと指摘している．さらに，別の被験者は，Icons++はとても役に立つが，ファイルの内容を見る意思がない場合では，自動で表示されるコンテンツビューをうるさく感じる可能性があると指摘した．これらの指摘により，Icons++の改善すべき課題が明らかになった．

7.6 考察

アプリケーションを起動する，またはプレビュー機能を用いるといった既存の手法に比べ，Icons++はより直感的であることがわかった．被験者は説明がなくても，自ずと Icons++の使い方に気付くことができた．Icons++の UI が好まれた主な理由として，ホバーによるコンテンツビューの素早い出現と操作アイコンを用いた簡単なファイル操作が挙げられる．ホバーによって素早くファイルの内容が表示されるため，内容を確認するためにアプリケーションを用いてファイルを開く必要がなくなる．

ホバーという動作はクリック（ファイルの選択時などに使用）ダブルクリック（ファイルを開く際に使用）を行う動作の途中に自然に存在する．そのため，Icons++ではユーザは，ファイルに対して何らかの動作を行う際に，アイコンのホバーによるファイルの内容表示を自然に発見することができる．２つのユーザスタディと回答してもらったアンケートの結果から，被験者はホバーやワンクリックを用いる操作アイコンというような，簡単な使い方を好ましく思うことがわかった．

しかしながら，コンテンツビューによる素早いファイルの内容表示は，ユーザがファイルの内容について知りたいと思わない場合には，ユーザのファイル使用の妨げになり得ることもわかった．したがって，Icons++のインタフェースをより役に立つように改善する余地があ

る．例えば，コンテンツビューの表示時間に制限を設けることやコンテンツビューの表示を半透明なものにするといった解決方法が考えられる．

7.7　実験結果を受けての UI 改善

評価実験のアンケートによって，ユーザがファイルの内容について知りたいと思わないときに，コンテンツビューの表示はユーザのファイル使用の妨げになることが判明した．それに対し，一つの解決方法として，本研究では Icons++のウィンドウ上にコンテンツビューの表示時間を調整できるスライダーを設置した．図 7.5 のように，ウィンドウの上部に設置されたスライダーによって，ユーザはコンテンツビューの表示の ON と OFF，そして表示する場合には表示が消えるまでの時間を設定することができる．

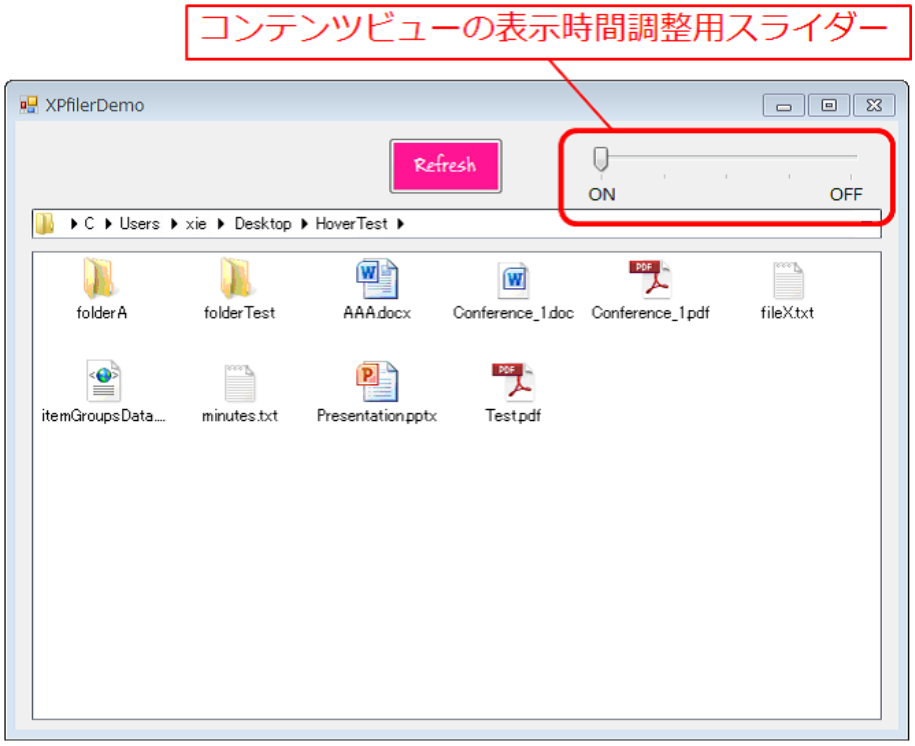


図 7.5: コンテンツビューの表示時間調整用スライダーを付けた Icons++の全体 UI

図 7.5 にある Icons++のウィンドウの上部のスライダーは，デフォルトでは左端の ON 状態になっている．ON 状態では，コンテンツビューはユーザがアイコンにマウスカーソルを当てている限り表示され，自動的に消えることはない．右端の OFF 状態では，ユーザがファイルのアイコンにマウスカーソルを当ててもコンテンツビューは表示されなくなる．また，ON と OFF の間にある 3 段階の区切りにつまみを合わせると，それぞれ ON に近い方から 2 秒・3 秒・5 秒後にコンテンツビューが消えるようになる．

この表示時間調整用のスライダーを用いることにより，ユーザはコンテンツビューを邪魔に感じたらコンテンツビューの表示をなくすことができる．また，マウスカーソルを当ててもある程度の時間が経ったら自動で消えるように設定できるようになる．

第8章　結論

本研究では，ファイルシステムにおけるファイル操作が可能な内容表示インタフェースを提案し，プロトタイプシステムとして Icons++を開発した．

Icons++では多階層からなるフォルダについて，マウスカーソルによるホバーによって，フォルダを入れ子状に展開する．それにより，ユーザは異なる階層に存在するサブフォルダやファイルを同一画面で閲覧することができる．また，Icons++ではホバーによって表示されるコンテンツビューを用いて，ファイルの内容をユーザに見せ，ファイルの内容を確認できるようにした．さらに，コンテンツビュー上では，操作アイコンを用いることにより，簡単にファイルの操作を行うことができる．

Icons++と既存の手法を比較し，Icons++のメリット・デメリットについて認識するために，評価実験として 2 つのユーザスタディと 1 つのアンケートを実施した．実験の結果から，Icons++を用いてファイルの内容を見る方法では，アプリケーションを用いてファイルを開く方法よりも 53%素早く行うことができることがわかった．また，同じ操作を行いたい場合，アプリケーションを用いてファイル操作を行うよりも，Icons++の操作アイコンを用いてファイル操作を行った方が 22%速くできることがわかった．さらに，被験者からのコメントにより，Icons++が役に立つと感じられたことがわかった．同時に，Icons++の UI において解決すべき問題点も判明した．

将来の展望としては，被験者からのアンケート結果にあった，コンテンツビューを用いてファイルの内容を見る場合に，画面が小さいため内容を見づらいという問題に取り掛かりたい．また，コンテンツビューを邪魔に感じる問題に対して，一つの解決策としてスライダーを付けたが，いくつか解決方法を実装し，どの解決方法が最良かを調べたい．

謝辞

本研究を進めるにあたり，指導教員である田中二郎先生をはじめ，志築文太郎先生，高橋伸先生，三末和男先生，Simona Vasilache 先生，嵯峨智先生には，ゼミなどを通して丁寧なご指導や貴重なご意見をいただきました．深く御礼申し上げます．また，学会発表などの研究発表の際には，数多くの助言や励ましの言葉をいただき，大変勉強になりました．博士前期課程に在学した 2 年間，これらのご指導があったおかげで研究を楽しく続けることができました．

ことに田中二郎先生にはテーマの選び方，目的の決め方から研究の進め方，論文の書き方に至るまで，始終丁寧かつ熱心なご指導をいただきました．ゼミや面談を通して，自分の中での曖昧な点や不明瞭な点について気づきを与えていただき，常に本質の問題は何かを意識するなど，多々のことを学びました．心よりお礼申し上げます．

参考文献

[1] 大谷和利. GUI の起源と進化の概略 (情報とデザイン). 情報の科学と技術, Vol. 49, No. 12, 1999, pp. 632-638.

[2] 久保田晃弘, 宮崎光弘, 永原康史, 松田純一, 大谷和利, 杉山久仁彦. GUI を再発明するポスト・ウィンドウ・インターフェイス序論. 多摩美術大学研究紀要, Vol. 23, 2008, pp. 163-170.

[3] D. Barreau and B. A. Nardi, Finding and Reminding: File Orgamization from the Desktop. ACM SIGCHI Bulletin 27.3, 1995, pp. 39-43.

[4] J. Tidwell, Designing interfaces. O'Reilly Media, Inc., 2010, 578 pages.

[5] B. Nardi, K. Anderson, and T. Erickson, Filing and Finding Computer Files. Proceedings of the East-West HCI, 1995, pp. 162-179.

[6] W. Jones, A. J. Phuwanartnurak, R. Gill, and J. Bruce, Don't Take My Folders Away! Organizing Personal Information to Get Things Done. In Ext. abstracts CHI '05, 2005, pp. 1505-1508.

[7] O. Bergman, S. Whittaker, M. Sanderson, R. Nachmias, and A. Ramamoorthy, How Do We Find Personal Files?: The Effect of OS, Presentation & Depth on File Navigation. In Proc. CHI '12, 2012, pp. 2977-2980.

[8] A. Kamaruddin and A. Dix, How Do You Make Your Folder?, MRG 27-2 Poster presentation, 2006.

[9] O. Bergman, S. Whittaker, M. Sanderson, R. Nachmias, and A. Ramamoorthy, The Effect of Folder Structure on Personal File Navigation. Journal of the American Society for Information Science and Technology 61. 12, 2010, pp. 2426-2441.

[10] T. B. Brude and D. L. Scapin, What do People Recall about their Documents? Implications for Desktop Search Tools. In Proceedings of the 12th international conference on Intelligent user interfaces (IUI '07), 2007, pp. 102-111.

[11] S. Greenberg and I. H. Witten, Supporting Command Reuse: Mechanisms for Reuse. International Journal of Man-Machine Studies, Vol. 39-3, 1993, pp. 391-425.

[12] R. Mander, G. Salomon and Y. Y. Wong, A 'Pile' Metaphor for Supporting Casual Organization of Information. In Proc. CHI '92, 1992, pp. 627-634.

[13] A. Agarawala and R. Balasrishnan, Keepin'it Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. In Proc. SIGCHI, 2006, pp. 1283-1292.

[14] T. Hakala, J. Lehikoinen and A. Aaltonen, Spatial Interactive Visualization on Small Screen. MobileHCI '05, 2005, pp. 137-144.

[15] N. Watanabe and T. Igarashi, Bubble Clusters: An Interface for Manipulating Spatial Aggregation of Graphical Objects. In Proc. UIST '07, 2007, pp. 173-182.

[16] B. Johnson and B. Shneiderman, Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. Visualization '91, Proceedings., IEEE Conference on. IEEE, 1991, pp. 284-291.

[17] M. Weiland and R. Dachselt, Facet Folders: Flexible Filter Hierarchies with Faceted Metadata. In Proc. CHI '08, 2008, pp. 3735-3740.

[18] B. Shneiderman, Treemaps for Space-Constrained Visualization for Hierarchies. ACM Trandactions on Graphics (TOG) Vol. 11, Issue 1, 1992, pp. 92-99.

[19] B. Engdahl, M. Koksal, and G. Marsden, Using Treemaps to Visualize Threaded Discussion Forums on PDAs. In Proc. CHI '05, pp. 1355-1358.

[20] 三末和男, 図的発想支援システム：D-ABDUCTOR のための図の操作について. 情報処理学会第 41 回全国大会, 1990, pp. 237-238.

[21] 三末和男, 杉山公造, 図的発想支援システム D-ABDUCTOR の開発について. 情報処理学会論文誌 35.9, 1994, pp. 1739-1749.

[22] S. Fitchett, A. Cockburn, and C. Gutwin, Improving Navigation-Based File Retrieval. In Proc. CHI '13, 2013, pp. 2329-2338.

[23] M. Terry and E. D. Mynatt, Side Views: Persistent, On-Demand Previews for Open-Ended Tasks. In Proc. UIST '02, 2002, pp. 71-80.

[24] J. Matejka, W. Li, T. Grossman, and G. Fitzmaurice, CommunityCommands: Command Recommendations for Software Applications. In Proc. UIST '09, 2009, pp. 193-202.

[25] S. Voida, W. K. Edwards, M. W. Newman, R. E. Grinter, and N. Ducheneaut, Share and Share Alike: Exploring the User Interface Affordance of File Sharing. In Proc. CHI '06, 2006, pp. 221-230.

[26] J. Alexander, A. Cockburn, S. Fitchett, C, Gutwin, and S. Greenberg, Revisiting Read Wear: Analysis, Design, and Evaluation of a Footprints Scrollbar. In Proc. CHI '09, 2009, pp. 1665-1674.

[27] C. Appert and S Zhai, Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support. In Proc. CHI ’09, 2009, pp. 2289-2298.

[28] L. Findlater, K. Moffatt, J. McGrenere, and J. Dawson, Ephemeral Adaptation: The Use of Gradual Onset to Improve Menu Selection Performance. In Proc. CHI ’09, 2009, pp. 1655-1664.

[29] T. Tsandilas and M. C. Schraefel, Bubbling Menus: A Selective Mechanism for Accessing Hierarchical Drop-Down Menus. In Proc. CHI ’07, 2007, pp. 1195-1204.

[30] 増井俊之, 大和田誠, 操作の繰返しによるマクロの自動生成. 情報処理学会ヒューマンインタフェース研究会研究報告, 1993, pp. 65-72.

[31] F. Block, H. Gellersen, and N. Villar, Touch-Display Keyboards: Transforming Keyboards into Interactive Surfaces. In Proc. CHI ’10, 2010, pp. 1145-1154.

[32] T. Yeh, T. Chang, and R. C. Miller, Sikuli: Using GUI Screenshots for Search and Automation. In Proc. UIST ’09, 2009, pp. 183-192.

[33] T. Chang, Using Graphical Representation of User Interfaces ad Visual References. In Proc. UIST ’11, 2011, pp. 27-30.

[34] J. Scarr, A. Cockburn, C. Gutwin, and A. Bunt. Improving Command Selection with CommandMaps. In Proc. CHI ’12, 2012, pp. 257-266.

[35] S. Fitchett, A. Cockburn, and C, Gutwin, Finder Highlights: Field Evaluation and Design of an Augmented File Browser. In Proc. CHI ’14, 2014, pp. 3685-3694.

[36] B. Suh, A. Woodruff, R. Rosenholtz, and A. Glass, Popout Prism: Adding Perceptual Principles to Overview+Detail Document Interfaces. In Proc. CHI ’02, 2002, pp. 251-258.

[37] A. Woodruff, A. Faulring, R. Rosenholtz, J. Morrison, and P. Pirolli, Using Thumbnails to Search the Web. ACM SIGCHI ’01, 2001, pp. 198-205.

[38] P. Brandl, C. Richter, and M. Haller, NiCEBook-Supporting Natural Note Taking. In Proc. CHI ’10, 2010, pp. 599-608.

[39] S. Kaasten and S. Greenberg, Integrating Back, History and Bookmarks in Web Browsers. In Proc. CHI ’01, 2001, pp. 379-380.

[40] A. Hoeben and P. J. Stappers, Flicking through Page-based Documents with Thumbnail Sliders and Electronic Dog-ears. In Proc. CHI ’00, 2000, pp. 191-192.

[41] 小澤怜, 大瀧保広, 新堀道信, 鎌田賢. ファイル内の付箋情報が見える作業効率化ファイルマネージャ. 第 18 回一般社団法人情報処理学会シンポジウム (インタラクション 2014), 2014, pp. 172-175.

[42] V. Setlur, C. A. Buehler, A. A. Gooch, S. Rossoff, and B. Gooch, Semanticons: Visual Metaphors as File Icons. In Computer Graphics Forum. Blackwell Publishing, Inc, 2005, pp. 627-656.

[43] A. Cockburn, S. Greenberg, B. McKenzie, M. Jasonsmith, and S. Kaasten, WebView: A Graphical Aid for Revisiting Web Pages. In Proc. OzCHI '99, 1999, pp. 15-22.

付録 A　評価実験に用いたアンケート用紙

被験者アンケート

2014年　月　日

氏名＿＿＿＿＿＿＿＿＿＿＿＿（　　才）

所属＿＿＿＿＿＿＿＿＿＿＿＿

普段使用するOS　1:　　2:

実験のご協力ありがとうございます。以下のアンケートにご協力ください。

5：そう思う　4：だいたいそう思う　3：どちらでもない　2：あまりそう思わない　1：そう思わない

1．アクセス方法は直感的か

(説明がなくても自分でどう使用するかに気づけるか)

			そう思わない				そう思う	
テスト1	アプリケーションで開く	：	1	2	3	4	5	理由：
テスト2	ファイルマネージャのプレビュー機能	：	1	2	3	4	5	理由：
テスト3	（提案手法）ホバーによる内容表示	：	1	2	3	4	5	理由：

2．使いやすいと感じるか（提案手法に関して）

	ホバーによる内容表示	：	1	2	3	4	5	理由：
	ワンクリックによるファイル操作	：	1	2	3	4	5	理由：
	全体的なUI	：	1	2	3	4	5	理由：

3．今後も使っていきたいか（提案手法に関して）

	ホバーによる内容表示	：	1	2	3	4	5	理由：
	ワンクリックによるファイル操作	：	1	2	3	4	5	理由：
	全体的なUI	：	1	2	3	4	5	理由：

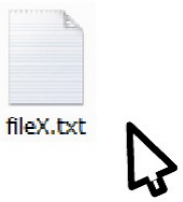
4．全体的にコメント・改善点・感じたことなど

ご協力ありがとうございました！

提案UIの使い方

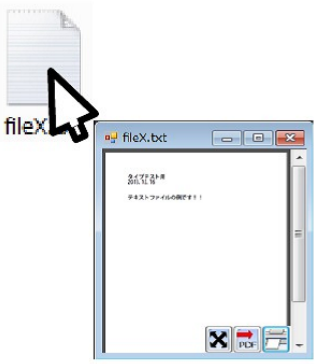
内容の表示

アイコンをホバーする



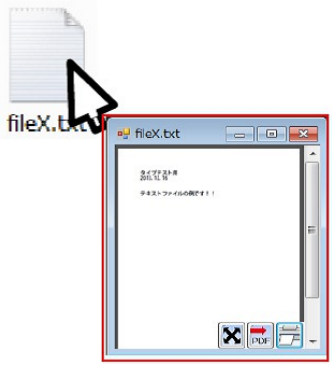
内容の非表示

アイコンからマウスを離す



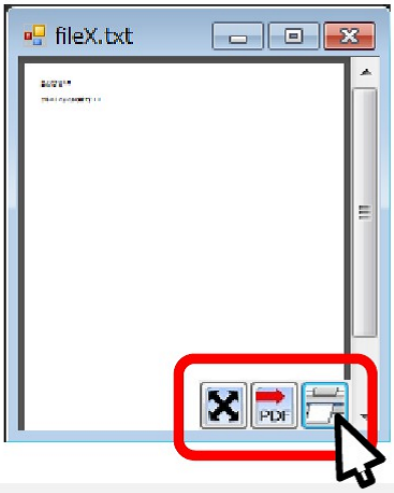
内容表示の固定

アイコンをClickする



1 Clickファイル操作

操作アイコンを選択



操作アイコン例

スライドショー　全画面表示　PDF作成　印刷