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

第1章 序論
1.1 Graphical User Interfaceについて .......................... 1
1.2 ファイルシステムのGUI ........................................ 1
1.3 ファイルの閲覧と操作におけるニーズ .......................... 1
1.4 アイコン配置に考慮したフォルダ展開の重要性 .................. 2
1.5 本研究の位置づけ .............................................. 2
1.6 本論文の構成 .................................................. 3

第2章 目的とアプローチ ............................................. 4
2.1 本研究の目的 .................................................. 4
2.2 本研究のアプローチ .............................................. 4

第3章 システムのインタフェース設計 ................................. 5
3.1 ホバーによるアクセス ........................................... 5
3.2 コンテンツビューの表示 ......................................... 5
3.2.1 コンテンツビューの表示・固定・消去 ......................... 5
3.2.2 フォルダの中身の表示 ....................................... 6
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 .. 7
3.2.4 ファイルの内容表示 ......................................... 8
3.3 コンテンツビューにおける操作 ................................... 9
3.3.1 複数フォルダ間のアイテムの移動 ............................ 9
3.3.2 アプリケーションへのアクセス ................................ 10
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 .......... 11
3.3.4 Dog-earとページの折り畳みを用いたマーク付け ............... 12
Dog-earのインタラクション ....................................... 13
ページの折り畳みのインタラクション ................................ 14
3.4 提案するインタフェースによる利点 ............................. 14

第4章 利用シナリオ .................................................. 15
4.1 ファイルの検索における利用例 ................................. 15
4.2 ファイル操作を行う際の利用例 ................................ 15
4.3 ファイルの内容を見比べる際の利用例 .......................... 16
第5章 プロトタイプシステム Icons++の実装 17
5.1 開発環境と言語 ................................. 17
5.2 Icons++の全体像 .............................. 17
5.3 フォルダ展開の実現 ......................... 18
  5.3.1 フォルダ内のアイテム数を用いたサイズの計算 18
  5.3.2 フォルダのサイズ調整の流れ .......... 19
  5.3.3 フォルダ展開画面におけるファイルの移動 20
5.4 ファイルの内容表示 ......................... 20
  5.4.1 ファイルのコンテンツビューの実現 20
  5.4.2 素早い内容表示と表示の安定性 .......... 21
5.5 ファイルの操作 .......................... 22
5.6 Dog-earとページの折り畳みの実現 ........ 22
  5.6.1 サムネイル画像の編集 .................. 22
  5.6.2 Dog-earとページの折り畳みの再現 .... 22

第6章 関連研究 24
6.1 フォルダの中身の表示方法に関する研究 ........ 24
6.2 入子状の要素展開に関する研究 .......... 24
6.3 ホバーによる情報提示に関する研究 ........ 25
6.4 素早い操作の実現に関する研究 .......... 25
6.5 ファイル操作時のアイコン利用に関する研究 .... 26
6.6 素早いアイテム検索に関する研究 .......... 26
6.7 関連するアプリケーション .......................... 27

第7章 評価実験 28
7.1 実験目的 ..................................... 28
7.2 被験者 ..................................... 28
7.3 実験手順 .................................. 28
7.4 実験内容 .................................. 28
  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 ..... 28
  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 29
  7.4.3 アンケート ................................ 30
7.5 実験結果 .................................. 31
  7.5.1 ユーザスタディ1の結果 .................. 31
  7.5.2 ユーザスタディ2の結果 .................. 32
  7.5.3 アンケートの結果 ...................... 32
7.6 考察 ....................................... 35
7.7 実験結果を受けてのUI改善 .................... 36
第 8 章 結論

謝辞

参考文献

付録
図目次

1.1 Icons++の位置づけ  ................................................. 2

3.1 ハバーボイによるフォルダの中身の表示  ........................................... 6
3.2 複数フォルダに対するコンテンツビューの表示  ......................................... 7
3.3 入れ子状表示を用いた複数フォルダの中身の表示  ......................................... 7
3.4 フォルダ展開後のファイルの内容閲覧  ................................................ 8
3.5 コンテンツビューによるファイルの内容表示  .............................................. 8
3.6 複数ファイルに対する内容表示  .......................................................... 9
3.7 フォルダの入れ子状表示におけるファイル移動  ........................................... 10
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  ................................ 11
3.9 コンテンツビュー下部に表示される操作アイコン  ........................................... 11
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  ........................................................................... 12
3.11 Dog-ear とページの折り畳み  ............................................................ 12
3.12 Dog-ear におけるインタラクション  ........................................................ 13

5.1 Icons++の全体的な UI .................................................................. 17
5.2 アイテムの並べ方とコンテンツビューのサイズ ........................................ 18
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  .................... 19
5.4 ファイルに対する JPEG ファイルの作成  ............................................... 20
5.5 ファイルの内容表示の実現  ............................................................... 21

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  ........................................................................... 29
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  ................................................................ 31
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  ........................................... 32
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  .............................. 33
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  ........ 36
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ー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・メニューや、ポインティングデバイスを構成の基本要素とする [1, 2]。ユーザー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拡げる。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、内容が写真の画像が埋め込まれた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ことができる。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。

1
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ことが多い、ツリー表示は、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ニシエクトすらして、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、フォルダ階層ごとに横に並べて表示する。具体的に一つのカラムには、数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けられ、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係性を見ながらファイルにたどり着く方法を好む [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かりとなる [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 では関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。

![アイコン] [プレビュー] [Icons++] [アプリケーション]

関覧性 操作性

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、アイコンは関覧性に優れていると言え、関覧性よりは配置される。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くないページを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リ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x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텐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
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せ
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別の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）。移動後は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。
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くと言える。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リケーションに開くことも可能である。
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

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](image)

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 シナリオ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、分けていなく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文献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、深くどったりしなければいけない、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たよ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るような関連研究のファイルを見つけたことができる。

4.2 シナリオ操作を行う際の利用例

議事録係の A さんは、午前後で開かれた2つのミーティ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、前を付けして保存」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回も行わなければならないことを考えて 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。この度研究室の同僚で打ち上げることになり、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、同僚に喜ばれるよ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。

![Icons++の全体的な UI](image)

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
5.3 カルダ展開の実現

5.3.1 カルダ内のアイテム数を用いたサイズの計算

カルダ展開の際、カ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++ではユーザがカ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、カ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内容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。

複数カ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を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にあらゆ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

19
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まる。

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5.4.1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++を利用し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。

20
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、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 Dog-ear、それ以外の全体をダブルクリックされたページの折り畳みだと言えている。

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、元の画像マークの付け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できるとあるためである。

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。もしこ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ロラ表示がある。

Hakala ら [14] はフォルダやサブフォルダを繰り返していく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き、フォルダのアイコンが重なり立つ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の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状で表示する手法が述べられている。

本研究では、フォルダの展開時に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


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


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。ユーザはスクリプ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よっ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可能である。

これらの研究と異なり、提案インタフェースは従来使わ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


ファイルの内容のサムネイルではなく、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だけでなく、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（「思うわけではない」に相当）、5が最も評価が高い（「思うと思う」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い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 3 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・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・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方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）。

31
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回の試行の平均完成時間は、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名の被験者がユーススタディ2を行った際の平均完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完成時間

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フ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）

<table>
<thead>
<tr>
<th>設問</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>平均値</th>
</tr>
</thead>
<tbody>
<tr>
<td>ユーザスタディ１で使用した３つの方法について</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4.1</td>
</tr>
<tr>
<td>アプリケーションで聞く方法は直感的か</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ファイルマネージャのプレビュー機能は直感的か</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2.7</td>
</tr>
<tr>
<td>Icons++のホバーによる内容表示は直感的か</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>4.4</td>
</tr>
</tbody>
</table>

| Icons++のUIについて                        |     |     |     |     |     | 3.9    |
| ホバーによる内容表示は使いやすいか          | 1   | 0   | 0   | 4   | 2   |        |
| 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いたいか       | 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いたいか | 0   | 0   | 2   | 2   | 3   | 4.1    |
| 全体的なUIを今後も使いたいか                  | 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ます。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扱える。

33
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++のホパ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、Windows7のファイルマネージャー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、被験者が自分自身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（回答の平均値が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」「使いやすくそう」「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だから、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、使い方を学び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らっと良い」「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、右クリックの方が駄目が多い」「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

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のは小気短過ぎて、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++はより直接的であることが分かった。被験者は説明がなくても、自分と 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に、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た操作アイコンというような、簡単な使い方を好ましけじゅうすることが分かった。

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分かった。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段階の区切りに含まれると、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を明示した。

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、志篤文太郎先生、高橋伸先生、三末和男先生、Simona Vasilache先生、暁蛾智先生には、ゼミなどを通じ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


[41] 小澤怜, 大槻保, 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.

付録A 評価実験に用いたアンケート用紙
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)

テスト1 アプリケーションで聞く：

テスト2 ファイルマネージャのプレビュー機能：

テスト3 (提案手法) ホバーによる内容表示：

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：

ワンクリックによるファイル操作：

全体的なUI：

3. 今後も使いていきたいか（提案手法に関して）

ホバーによる内容表示：

ワンクリックによるファイル操作：

全体的なUI：

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