

7N-4

木構造に基づいたアイテムマネージャ

西村 潤二[†]

中山 健^{††}

小林 良岳[†]

前川 守[†]

1 はじめに

現在、ユーザはパソコンの記憶容量の大容量化、演算機能の高性能化により多数のファイルやメールなどを日常的に分類、操作している。ユーザがパソコンで分類、操作しているアイテムは多岐に渡る。ファイルやディレクトリをはじめ、メール、音楽などのプレイリスト、画像ファイルなどである。しかし現在のシステムで多数のアイテムを分類、操作するには、様々な制約により、できない分類、操作が存在する。本研究では、ユーザが多数のアイテムを分類、操作しやすくする体系を提案し、ファイルシステムを例とした、プロトタイプシステムを作成した。

2 背景

2.1 OS等システム上の制約

アイテム分類の表現機構の制約により、アイテムの分類、操作が阻害されている。たとえば

1. 通常のファイルシステムは同一ディレクトリに同名のファイルを複数作れない。

2.2 操作体系の不統一性

アイテムの種類により分類、操作体系が異なる。同じ操作であっても異なる動作をするため、アイテムの分類、操作にはアイテムそれぞれに操作を理解しなければならない。たとえば

1. ファイルシステムでの操作では、[削除]を行えば対象のファイルやディレクトリは実体が削除されるが、プレイリストでは、プレイリストのアイテムが削除されるだけで、ファイル自体は削除されない。

2.3 検索等の演算結果の再利用性の低さ

入力である検索対象と、出力は同じ集合ではない場合、その結果からさらに検索を行うことができない。たとえば

1. ファイルシステムに対する検索結果は、ファイルシステムとは異なる一覧として出力されることが多い。したがってこの結果に対してはさらに検索を行うことが出来ない。

2. 一般の検索機能では、ファイル名、テキストに含まれる文字列、変更日時、サイズ、種類などから検索を行えるが、検索結果からさらに検索することはできない。

3 方針

これらの問題点を解決するために、操作体系の統一と、検索結果の再利用できる構造が必要である。

- 操作体系の統一
どのような種類のアイテムを分類、操作するときでも、1つの操作に対して同じレスポンスを返す必要がある。そのためにこの体系を「分類構造に対する操作」と「アイテムに対する操作」に分け、現在のアイテムごと異なる操作性を統一させる。
- 検索結果を再利用できる分類構造
検索結果を再利用するためには、検索対象と、検索結果が同じ分類構造である必要がある。検索対象と検索結果を同じ分類構造でできるようにする。

4 アイテム集合の木構造

これから示すシステムの特徴は、1つのアイテムの集合を複数の木構造で見ることができることにある。以下アイテムとディレクトリで構成されるファイルマネージャにより、本システムにより提供される木構造について説明する。

4.1 アイテムのみで構成された木

画面に表示されたファイルマネージャの木からアイテムをいくつか選択し、別の画面に選択されたアイテムだけがいった木 Ft が表示される (図 1)。

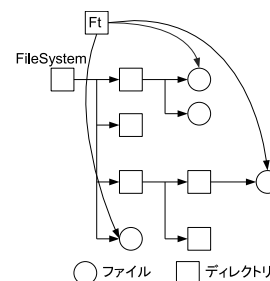


図 1: アイテムのみで構成された木構造 Ft

4.2 アイテムとディレクトリで構成された木

ファイルマネージャの木からアイテムとディレクトリをいくつか選択し、別の画面に選択されたアイテム

ItemManager based on tree structure

[†] 電気通信大学大学院 情報システム学研究所
Graduate School of Information Systems, University of Electro-Communications

^{††} 電気通信大学 eラーニング推進センター
Center for Developing e-Learning, University of Electro-Communications

1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182, Japan

ムやディレクトリが入った木 Dt が表示される。(図 2(a)) この時、ディレクトリ自体はファイルマネージャと同一のものであるので、ディレクトリの中身も同一のものとなる。

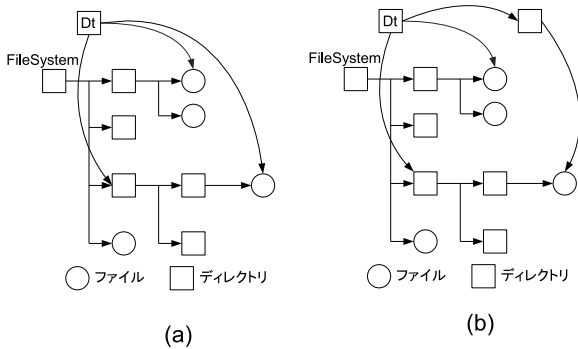


図 2: ディレクトリとアイテムで構成された木構造

4.3 木 Ft や木 Dt に新たにディレクトリを追加した木

ファイルマネージャの木から選択されたアイテムやディレクトリだけが表示されてる木 Ft や Dt に新たにディレクトリが追加されたもの(図 2(b)) このディレクトリはファイルマネージャのディレクトリとは別のものであり、このディレクトリはさらにファイルマネージャのアイテムやディレクトリを指している。

5 システムの仕様

システムは JAVA 言語で実装した。ユーザインターフェースは大別して次の 3 つのウィンドウで構成される。



図 3: システムのスナップショット

- ツリーフレーム (図 3 フレーム 1)
アイテム集合を木構造として提示する。複数の木構造を同時に扱える。
- セレクションフレーム (図 3 フレーム 2)
選択している 1 つのアイテムの情報を表示する。ディレクトリやファイルの情報、画像プレビューなど。
- テキストフレーム (図 3 フレーム 3)
ユーザの行ったアクション、現在のアプリケーションの状態などが表示される。

5.1 アイテムの入力

ファイルシステムのディレクトリを指定することで、そのディレクトリをルートとした部分木を抽出して、ツリーフレームに表示される。

5.2 検索

検索したい木構造を指定して検索を行う。ユーザには検索ウィンドウを提示する。検索結果はツリーフレームに木構造として表示される。

6 使用例

このシステムを使用して、前記の問題点の解決方法を示す。

- 要求：ファイル名”Readme.txt”を 1 つのディレクトリにまとめたい。

ファイルマネージャからファイル名”Readme.txt”を検索し、その結果を新しい画面に木として表示する。これはファイルのみで構成された木を作っている(図 1) この木はファイルシステムに依存しない木となるので、同名のファイルが複数あっても問題はない。

- 要求：ファイルマネージャで 2006 年 1 月 1 日にキャプチャーしたテレビ番組を検索したい

ファイルマネージャから読み込んだ木に対してキャプチャーファイルの検索を行う。その検索結果を新たな画面で木として表示する。さらに検索結果の木に対してファイル日付で検索を行う。

7 まとめ

アイテムへの操作性を統一させ、また絞り込み検索の行える再利用可能な検索機能の実装した。本システムによりどのようなアイテムでも分類、操作が行いやすくなると思われる。

8 参考文献

林竜輝 (2004), 木構造を操作する Script 言語, 電気通信大学大学院 情報システム学研究所 平成 15 年度修士論文