多次元ツリー自動構成ツール MD-TACT の開発と評価

柿本 由気† 掛下 哲郎‡

佐賀大学工学系研究科 〒840-8502 佐賀市本庄町 1 番地

E-mail: †kakimoto-yuuki@cs.is.saga-u.ac.jp, ‡kake@is.saga-u.ac.jp

あらまし 近年多くの企業や企業で大量の情報が電子化されており、それらの整理や検索が困難になっている。我々は増え続けるファイル群を系統的に整理するべく、多次元ツリーを用いたファイル整理ツール HyperClassifier を開発している。本論文では既存の分類で良く使われる単一ツリーを HyperClassifier の多次元ツリーに変換する手間を軽減するため、多次元ツリー自動構成ツール MD-TACT を開発する。MD-TACT は単一ツリーを多次元ツリーに自動変換し、さらに手動で洗練する機能を提供する。また、ツリーの再構成情報を利用することで、それ以降の自動構成の精度向上を図る。MD-TACT の評価実験を行った結果、被験者からは良好な評価が得られた。また、MD-TACT を使用することで、手作業と比較して 3.8 倍程度の効率化が図れることが分かった。現在、評価実験を通じて得たログデータを詳細に分析中である。

キーワード 多次元ツリー、OLAP、ファイル整理ツール

1. はじめに

近年、多くの企業や団体でコンピュータが導入され、大量の情報を電子化して扱っている。その数は企業の規模などにもよるが、ファイル数にして、およそ数万から数十万ファイル以上にものぼる。これらの情報は、企業の活動によって作成およびやり取りがされることによって次々と蓄積されていく。企業が蓄積している情報の量は、時間の経過とともに増大の一途をたどっている。近年では 50～60%の割合で増加し、今後 10 年以上はこの傾向が続くといわれている。このまま大量のファイルが蓄積してくると、それらの整理や検索が困難になってくる。ある調査によると「インフォメーションワーカーは、平均で労働時間の 24 %を情報の検索と分析に費やしている」との報告[1] もあり、情報探索にかかる労力は、企業活動において大きな負担となっている。そのため、必要な情報の探索にかける時間を短縮することは、企業等にとって重要な課題である。

このような背景から、ファイルを系統的に分類・整理し、素早く検索できるシステムが求められている。企業内に蓄積された情報を検索できるようなシステムは従来から多く開発されているが、その多くは、Web ページの検索エンジンのような、キーワードを入力することによってファイルの検索を行うシステムである。しかしこの形式のシステムの場合、情報を得るためにはキーワードが必要であり、目的の情報を検索できるキーワードが不明だと検索ができない。また、ファイルサーバの中に何の情報がどのように分布しているか分からず、目的の情報が存在するかが分かりにくい。

これらの問題を解決するために、我々は多次元ツリーを用いてファイルを整理する方法を用いたファイル整理ツール HyperClassifier を開発してきた[2,3]。HyperClassifier は多次元ツリー構造と対応付けてファイルを登録し、OLAP 操作を行うことで登録したファイルを検索できる。これにより、今までのファイル整理ツールと比較して高速かつ柔軟な検索ができるようになった。

HyperClassifier の欠点として、既存のファイルサーバからの移行に手間がかかる点が挙げられる。特に、単一のフォルダ階層を用いて管理されてきたファイルを多次元ツリーによって再整理するには時間と労力がかかる。

この欠点を克服するために、本論文では既存の単一ツリーで構成されたファイル群を多次元ツリー形式に自動的に組み替えるツール、MD-TACT（Multi-Dimensional Tree Automatic Construction Tool）を開発する[4,5,6]。MD-TACT は、既存のファイル群を読み込み、ファイル名を取得して自動的に多次元ツリーに再構成する。また、生成された多次元ツリーを HyperClassifier にインポートすることで、時間のかかるファイル登録を、ファイル群ごと一括で行うことができる。

本論文では、MD-TACT の開発と評価を行ったので、これについてまとめる。まず 2 節では、本研究のファイル整理の要点である多次元分類方式と OLAP 操作について述べ、それをもとに開発されたファイル整理ツール HyperClassifier の機能、特徴、欠点について述べる。3 節では、HyperClassifier の欠点を克服するため、本研究で提案及び開発した多次元ツリー自動構成ツール MD-TACT の目的と機能、基本的な処理の流れについて説明する。4 節では 3 節で述べた MD-TACT の有用性を検証するために、MD-TACT の評価実験を行い、その結果を分析する。5 節では本論文のまとめと今後の研究課題について述べる。

2. 基本事項

2.1. 多次元ツリー

多次元ツリーとは、各ファイルを複数のツリーと対応付けて分類する方式で、本研究で独自に提案する分類方式である。一般的な OS が用いるファイルシステムは、単一のツリーを使って分類を実現している。単一のツリーは、構造が比較的単純で実現が容易であるが、複数の分類観点が混在していると、ツリーの分類の一貫性を保てない。この他にも、検索の順序がツリーの根から葉への方向に固定され、検索の自由度が低い、分類観点が増えると、ノード数が爆発的に増加する等の欠点があり、検索効率が決して良いとは言えない。

これに対して多次元分類方式では、「プロジェクト名」や「ファイルの目的」などといった、ファイルの分類基準ごとにツリーを複数構築し、分類を行う。

多次元ツリーで用いるツリーは、以下の 2 つの制約を満たすように構成される。

- IS-A 制約
  親子のノード間には、IS-A 関連が成り立つように構築する。IS-A 関連とは、子ノードは親ノードの特別な場合に対応する関連である。ある概念 A と B が存在した時に、それが「A は B である（A is a B)」という関係が成り立つ関連である。
- 排他制約
  同一ツリーの兄弟ノードは、互いに排他的なものとする。この制約により、分類基準を明確化することができる。

多次元ツリーは個別のツリーが一貫しており、ツリーを比較的小さくできるため、理解及び保守が容易である。また、ツリーやノードを利用者が自由に指定して検索ができるので、全ての階層をたどらなくてもファイルを見つけることができる。さらに、カテゴリごとにツリーが作られているので、蓄積されている情報の全体像を容易に把握できる。

2.2. OLAP 操作

OLAP（Online Analytical Processing）は、利用者が直接データを検索・加工することで、問題発見や問題解決のための分析を行う、多次元データベース構成の分析型情報システムである。リレーショナルデータベースは 2 次元の表で構成されているが、多次元データベースは、それ以上の数の次元を持つことが可能である。HyperClassifier ではファイル検索の各操作にダイシング、スライシング、ドリリングの 3 種類の OLAP 操作を用いる。

ダイシング

次元を指定し、それに基づいた検索結果を表示する操作。「学年」「学部」「出身地」の 3 つの次元から作成した学生情報を集計する場合に、「学年」と「学部」という次元を指定して、それぞれを組み合わせた学生数の表を表示するときなどに用いられる。この操作により、分析の観点を変えることができる。

スライシング

特定の条件を適用し、データの絞り込みを行う操作。学生情報を集計する際に、「理工学部の学生」といった条件を付けて絞り込むような操作がスライシングである。条件を設定して絞り込むことで、必要な情報だけを抽出できる。

ドリリング

特定の次元の階層を上下させ、データの集計範囲を切り替える操作。下の階層に切り替える操作をドリルダウン、上の階層に切り替える操作をドリルアップという。例を挙げると、ドリルダウンは学部ごとのデータを学科ごとのデータに分割する操作、ドリルアップは学科ごとのデータを学部ごとのデータにまとめる操作である。

2.3. HyperClassifier

HyperClassifier は、ファイルを登録・分類する機能と、ファイルを検索する機能の 2 つを主に提供する。

ファイルの登録は、各観点のツリーから対応付けたいタグを選択し、登録するファイルやフォルダをドラッグ&ドロップすることで、登録を行うことができる。登録されたファイルはファイルサーバにアップロードされ、利用者が共用できるようになっている。

検索の際には、OLAP 操作を用いることができる。タグを選択するとダイシングを行い、そのタグに対応付けられているファイルの一覧が表示される（図 1)。ファイルを選択した状態でタグを選択するとスライシングされ、目的のファイルを素早く探し出すことができる（図 2)。各階層構造には、ドリリングによってアクセスできる（図 3)。

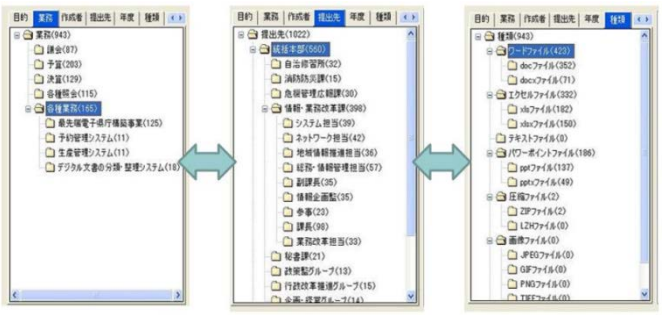

図 1. HyperClassifier のダイシング操作

従来のファイル整理ツールと違い、多次元ツリーを用いることでより検索しやすく、登録機能によりファイルの増加に対応することもできる。

上記 2 つの機能のほかに、CSV ファイルを介したツ

リーや対応関係のインポート・エクスポート機能がある。CSV ファイルを使用して所定の形式でツリーやファイルの情報を入力し、それを HyperClassifier に読み込むことで、ツリーの構築やファイルとタグの対応付けを一括変更できる。また、HyperClassifier 上に登録されているツリーや対応関係を CSV ファイルに書き出すこともできる。MD-TACT を用いて再構成した多次元ツリーは、CSV ファイルを用いて HyperClassifier にインポートできる。これにより、HyperClassifier の欠点である移行作業の手間を軽減している。

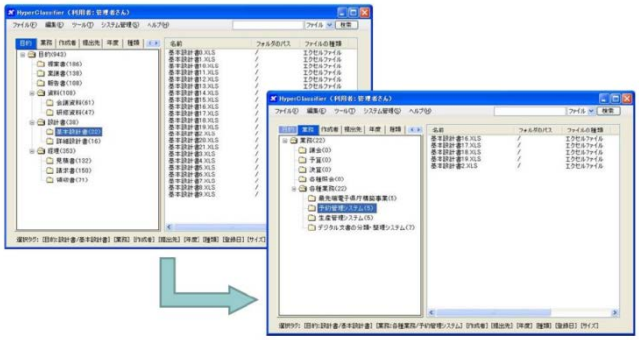

図 2. HyperClassifier のスライシング操作

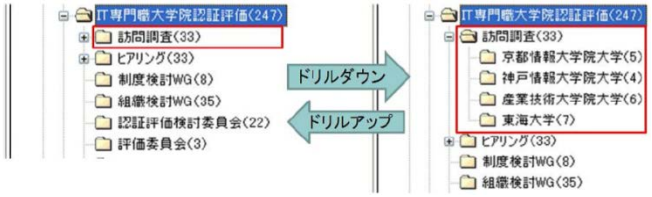

図 3. HyperClassifier のドリリング操作

3. MD-TACT

3.1. 概要

HyperClassifier では、ファイル登録の際にファイルと多次元ツリーを対応付ける必要があり、これには手間がかかる。多次元ツリー形式で構成されたファイル群なら CSV インポート機能を使うことで一括登録できるが、一般のファイル群は単一ツリーで整理されており、単一ツリーから多次元ツリーに再構成するのには手間と時間がかかる。多次元ツリー自動構成ツール MD-TACT は、この欠点を克服するために開発している（図４）。

MD-TACT は単一ツリーで構成されたデータ構造を読み込み、構成されているファイル・フォルダ名を切り出し、多次元ツリーへと自動的に変換し出力する。また、出力されたツリーに正しくないカテゴリ設定がされていた場合、手動で修正する機能も提供する。さらに再構成されたツリーの情報は CSV ファイルに出力することができ、これを HyperClassifier にインポートすることで HyperClassifier 上に多次元ツリーを再現できる。増え続けるファイルにも対応するため、このツールでも追加読み込みが可能になっている。

MD-TACT は、2 種類の情報を読み込んで処理を行う。１つは、単一ツリーの構造を示したファイルフルパスの一覧であり、これはテキストフォルダとして読み込む。フルパスは 1 行ずつ記録されており、各ノードの区切りには¥もしくは/が使われている必要がある。もう 1 つは単語辞書という CSV ファイルである。これは、MD-TACT で多次元ツリーを再構成した時に生成されるもので、どのノードがどのツリーに所属しているかを記録している。2 回目以降の読み込みでフルパス一覧と共に読み込めば、前回の構造を維持しながら新たなツリーを自動構成できる。

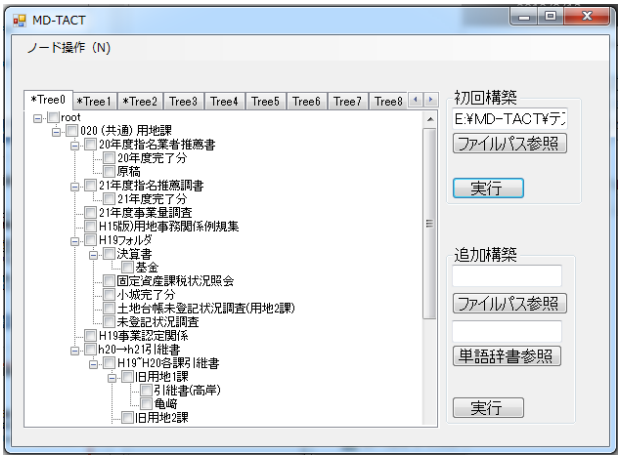

図 4. MD-TACT のユーザーインターフェース

3.2. 機能説明

MD-TACT は多次元ツリー自動構成機能と多次元ツリー手動洗練機能を主に提供する。

多次元ツリー自動構成

単一ツリー構造のファイル群から単語を切り出し、多次元ツリーの構造を自動的に構成する機能。このときに読み込まれるのはファイル群のフルパス一覧である。初回構築と追加構築の 2 パターンがあり、初回構築ではファイルパスのみで多次元ツリーを自動構成する。追加構築では、後述する単語辞書を共に読み込むことで、より精度の高い多次元ツリーを自動構成できる。

多次元ツリー手動洗練

初回構築で行われる自動構成は機械的な分類がされており、必ずしも精度の高い多次元ツリーが生成されるとは限らない。そこで自動構成終了後にユーザが手動で多次元ツリーを検査･編集する機能を提供する。編集結果は単語辞書に記録され、それ以降の自動構成の際には再利用される。多次元ツリー手動洗練機能は以下の操作から構成される。

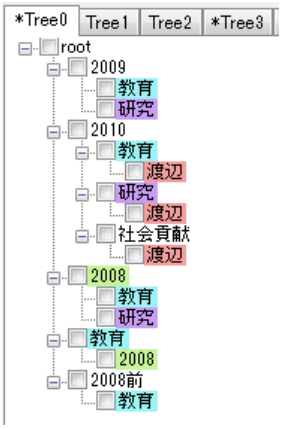

図 5. 重複警告機能

図 6. ノードの移動機能

図 7. ノードの分割機能

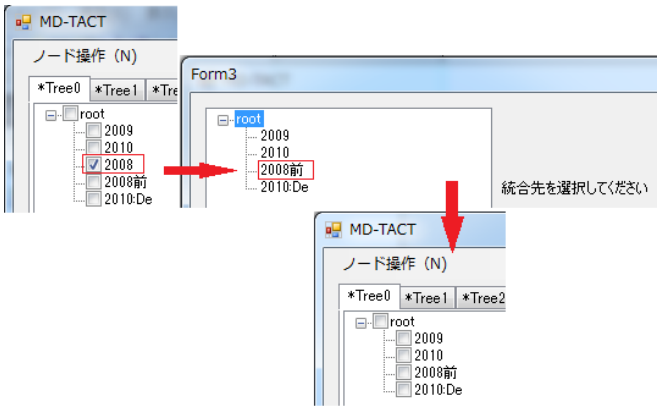

図 8. ノードの統合機能

- 重複単語の警告：同一ツリー上で同じ名前の単語が複数回出現した場合、単語ごとに色付けを行い、警告する（図 5）。MD-TACT では、同一ノードが重複して出現した場合、当該ツリーに複数の観点に属するノードが含まれていると判定する。
- ノードの移動：ツリー上の単語を別のツリーに移動する（図 6）。ノードのカテゴリを変更したいときに使う。移動先のツリー候補を示す際には、移動後に単語の重複が発生する場合は「不適切」との警告を出す。なお、利用者が警告を無視して単語を移動することもできる。
- ノードの分割：多義語を分割する。「Light」(光・右)などの紛らわしい単語を分割して登録することができる。分割された単語は、分割する単語に:DE を足したものとして追加される（図 7）。
- ノードの統合：同じ意味を持つ名前のフォルダを統合することができる。「2000 年以前」と「1999 年」などの、統合しても問題ないフォルダを統合することができる（図 8）。

これらの機能以外にも、ログの取得機能、多次元ツリーの出力機能などを提供する。

3.3. アルゴリズム

MD-TACT で単一ツリーを多次元ツリーに再構成するには、全 5 工程をこなす必要がある。3.3.1 節では多くの工程の共通操作であるツリーの再構成操作を定義する。3.3.2 節では多次元ツリー自動構成アルゴリズム（ステップ 1～3）について、3.3.3 節では多次元ツリーの手動洗練アルゴリズム（ステップ 4）について、それぞれ説明する。

最初の工程であるステップ 0 では、読み込んだファイル群のフルパス一覧から、ツール内で元の単一ツリーを構成する。この元ツリーは内部で保持され、基本的には表示されない。

3.3.1. ツリーの再構成操作

ステップ 0 で生成された元ツリーは、再構成操作を行う際に利用される。再構成操作とは、内部に保持されている元ツリーを、指定した単語のみを使い再構成する操作である。再構成操作では、上位階層から順に元ツリーのノードを 1 つずつコピーしていく。コピーしようとしたノードが指定されていない単語だった場合はコピーせず、その子ノードをコピーする。再構成操作が行われた後、直接の親子ノード間が同じ名前だった場合は子ノードを削除し、孫ノードを親ノードの子ノードとする。また、兄弟ノードが同じ名前だった場合、兄ノードと弟ノードを合併し、弟ノードの子ノードを兄ノードの子ノードとする。

再構成操作は元ツリーを直接操作しないので、元の構造や親子関係を保存しつつ多次元ツリーの操作ができる。

3.3.2. 自動構成アルゴリズム（初回構築）

ステップ 1 から 3 は多次元ツリーの自動構成を行う工程になる。

ステップ 1 では、ステップ 0 で生成された元ツリー上にあるファイル名・フォルダ名を全て取得し、単語辞書に記録する。単語には、それぞれの単語を識別するための単語 ID と、どのツリーに振り分けるかを判別するためのカテゴリ ID が割り振られる。初回構築では、全ての単語のカテゴリ ID は 0 が割り振られる。

ステップ 2 では、元ツリー上で 2 つ以上出現した単語を抽出する。多次元ツリーでは排他制約により、同じ名前のノードが 1 つのツリー上に 2 つ以上出てきてはいけないので、再構成操作を用いて重複する単語(以降、重複単語と呼称する）を元ツリーから切り出す。

ステップ 3 では、ステップ 2 で切り出した重複単語を、重複が発生しないようなツリーに割り当てる。各ツリーは、それぞれカテゴリ ID と対応した番号を持つ。この工程ではすべてのツリーに対し、対応する番号と同じカテゴリ ID を持つ単語のみを使い、再構成操作を行う。例えば、ツリー0 にはカテゴリ ID が 0 の単語だけを使い、ツリー1 にはカテゴリ ID が 1 の単語だけを使い再構成操作を行う。もしこのステップでどこかのツリーに重複単語が出現した場合、そのツリーに対しステップ 2 の操作を行う。

このようにしてステップ 2・3 を重複単語の出現がなくなるまで繰り返し続けることで、同一ツリー内での重複単語はなくなる。これにより、自動生成された多次元ツリーは排他制約を充足するが、単語の意味を考慮した処理を行っている訳ではないため、多次元ツリーが満たすべき IS-A 制約を充足できない場合もある。

3.3.3. 手動洗練アルゴリズム

ステップ 4 では手動洗練機能を用いて、生成された多次元ツリーの洗練を行う。これを通じて、IS-A 制約を充足するように多次元ツリーを編集する。

ノードの移動では、移動させたい単語を選択し、移動先のツリーに対応したカテゴリ ID に変更する。このとき、移動先で重複単語が発生する場合は警告する。カテゴリ ID の変更後、全ての表示用ツリーに対し、対応したカテゴリ ID をもつ単語のみを使い元ツリーを再構成する。

ノードの分割では、同じツリー上と元ツリー上に、選択した単語と同じカテゴリ ID を持つ単語を追加する。追加される場所は、選択した単語の兄弟ノードの位置であり、単語 ID は一意性のあるものを与えられる。変更後は全てのツリーを再構成する。

ノードの統合では、2 つの単語を対象とし、後者の単語 ID を前者の単語 ID と同じものに変更する。変更後、全てのツリーを再構成する。

3.3.4. 自動構成アルゴリズム（追加構築）

前述した 5 工程で多次元ツリーの再構成は完成するが、現実のファイル群は次から次へと増加するため、追加登録をしなければならない。追加構築以降の再構成では、ステップ 1 にあたる工程を多少修正する必要がある。本節では、修正後のステップ 1 をステップ 1’と呼ぶ。

ステップ 1’では、初回構築で構成した元ツリーに、追加で読み込んだファイル群のツリーを併合する。併合したツリーの中に新しい単語があった場合、並行して読み込んだ単語辞書に、新しい単語として登録する。既に登録している単語はカテゴリ ID を引き継ぎ、新しい単語のみカテゴリ ID に 0 を割り当てる。あとは初回構築と同じくステップ 2、3 を繰り返すことで、前回読み込んだ単語は既に分類された状態で自動構成が進む。

4. MD-TACT の評価

4.1. 目的

MD-TACT に使われている自動構成機能は、比較的単純な機械操作によって行われる。この自動構成機能により効率的なアルゴリズムを組み込めば、さらに効率的な多次元ツリーの自動構成が可能になる。MD-TACT にはログを取得する機能があり、ユーザの操作を記録できる。このログ機能を用いて多次元ツリーの構成作業を観察すれば、どういった手順で構成すれば短時間で質の高い多次元ツリーが構成できるか、に関する知見を得ることもできる。このデータを用いれば、MD-TACT の多次元ツリー自動構成機能の精度をさらに向上させることが期待できる。

MD-TACT をたくさんの人に利用してもらうには、まず本ツールの有用性を検証する必要がある。これを証明するために、MD-TACT 利用時および非利用時に単一ツリーから多次元ツリーを生成するために必要な時間を計測する。本節では、MD-TACT の評価実験の内容と評価結果の速報を述べる。

4.2. 評価実験の内容

MD-TACT の評価実験では、単一ツリーを多次元ツリーに再構成する作業を 10 名の被験者に行わせた。被験者は本学・知能情報システム学科の学部 3～4 年生であり、情報分野の専門教育を一通り受けている。本評価実験では MD-TACT 利用時・非利用時の 2 パターンのデータを採取した。MD-TACT 利用時はログ機能により出力されるログを収集した。一方、非利用時には Microsoft Excel のワークシートを用いて多次元ツリーを表現することとし、Excel のコマンドを用いてツリーの編集を行わせた。一例として、図 5 のツリー構造を Excel で表現したものを図 9 に示す。編集に要した時間は Excel マクロを用いて収集した。これらのデータは、多次元ツリーの構築時間と多次元ツリーの品質の 2 点を比較するために取得したものである。

テストの際には、被験者に多次元ツリーに関する講義を行い、多次元ツリーの構成方法、MD-TACT および評価用 Excel ワークシートの使用法を習得してもらった。次に MD-TACT を使用・非使用の 2 パターンで多次元ツリーを生成してもらった。10 人の被験者は 5 人 2 組のグループに分け、一方のグループ（以下、チ

ーム甲）には MD-TACT 使用、非使用の順で多次元ツリーの構築を行わせた。もう一方のグループ（以下、チーム乙）には MD-TACT 非使用、使用の順で多次元ツリーの構築を行わせた。これは多次元ツリーへの理解が深まることで公平なデータが取れなくなる事態を考慮した工夫である。また、再構築してもらうサンプルの単一ツリーは、1 人 1 人違うサンプルを使ってもらった。今回利用したのは、1649 行のファイルフルパス群である。MD-TACT 使用・未使用ではそれぞれ 2 時間・9 時間の制限時間を設けたが、どちらも制限時間内に再構成を完了した。非使用は膨大な時間がかかることが想定されていたので休憩時間も設けた。なお、休憩時間は構築時間には含まないものとした。休憩中はマクロの時間計測機能を中断させている。

最後に、被験者の 10 人には MD-TACT の使用感について感想文を書いてもらった。

図 9. Excel で表現した多次元ツリー

4.3. 評価と考察

評価実験は無事に終了し、良好な結果を得ることができた。表 1 は多次元ツリー作成時間の一覧表である。

4.3.1. MD-TACT による作業時間低減効果

まず MD-TACT を使うことでどれほどの時間が短縮できたかを確認する。表 1 の全体の平均時間を比較すると、MD-TACT 利用時は 1 時間 24 分 50 秒、Excel ワークシートを用いた手動再構成作業では 5 時間 21 分 46 秒かかっており、MD-TACT を利用しない場合、3.8 倍程度の時間が必要になることが分かる。これは、MD-TACT の多次元ツリー自動構成機能や手動洗練機能の有効性を示す結果だと考えられる。

また、両チームが手動再構成を行った時間の平均値を比較すると、甲チームは 4 時間 40 分 26 秒、乙チームは 6 時間 3 分 5 秒となっており、甲チームの方が 1 時間以上短くなっている。これは、MD-TACT を先に利用することで、多次元ツリーの完成形を既に確認できていたので、甲チームによる編集作業がスムーズに進んだためと考えられる。

一方、MD-TACT を使用した再構成作業時間の平均値は、甲チームが 1 時間 24 分 54 秒、乙チームが 1 時間 24 分 45 秒となっており、両チームともほとんど同じであった。この理由は現在詳しく調査中であるが、仮説の 1 つとして、MD-TACT の多次元ツリー自動構成機能により、完成形に近いツリー状態から手動洗練作業を始めることができたため、多次元ツリーの完成形をイメージできなくても作業がスムーズに進められたことが考えられる。

甲チームの被験者 4 は、MD-TACT を利用する際に、何度も最初からやり直しを行っていた。表の※の記録は、試行錯誤を経た後の記録であり、MD-TACT の修練を積めば、この程度の時間で多次元ツリーの再構成が可能になるとの傍証である。なお、※の時間は作業時間の一部の時間なので、甲チームや全体の平均計算には用いていない。

テスト後の感想文では、Excel を用いた手動再構成と比べると MD-TACT を利用した方が非常に楽であるという意見がほぼ全員から得られた。しかし、手動洗

表 1. 被験者毎の多次元ツリー再構成作業時間の比較

グループ	被験者	再構成作業時間			
		MD-TACT 利用		手動再構成	
甲	1	1:31:20	平均値 1:24:54	5:49:17	平均値 4:40:26
	2	1:35:47		6:02:25	
	3	1:10:34		3:22:51	
	4	0:36:21（※）		3:05:45	
	5	1:21:54		5:01:50	
乙	1	1:11:53	平均値 1:24:45	7:11:40	平均値 6:03:05
	2	1:28:17		5:47:30	
	3	1:14:04		6:43:35	
	4	1:41:00		5:29:43	
	5	1:29:06		5:02:56	
平均値		1:24:50		5:21:46	

表 2. 被験者毎の無駄時間の比較

グループ	被験者	無駄時間	
甲	1	00:04:32	平均値 0:30:39
	2	00:17:38	
	3	01:01:17	
	4	00:03:44（※）	
	5	00:39:11	
乙	1	00:13:29	平均値 0:17:24
	2	00:18:03	
	3	00:13:40	
	4	0	
	5	00:41:46	

練機能の使用中に間違った操作を行った場合、リカバリーのために時間がかかったという意見もあった。移動先を間違えて移動したノードを元のツリーに戻すのに時間がかかったり、統合機能でルートノードと統合して元に戻せなくなり最初からやり直す必要が出たりした。こういった問題に対応するために undo 機能がほしいという意見も出た。

4.3.2. 再構成プロセスによる無駄時間の相違

次に、テストによって得られたログから無駄時間がないかを確認した。MD-TACT の洗練操作を行う際、一度操作したノードが間違った洗練操作と気づき、元に戻す被験者が良く見られた。また、ノードの移動操作を行う際、予定した移動先で重複警告が出て、そのまま元のツリーに再移動する被験者も見られた。この操作は多次元ツリーの最終構成には不必要と判断できる。これらの無駄時間をログから計測し、実際の再構成時間がどれほどかかっているかを確認した。

MD-TACT は使用後にログの CSV ファイルを出力する。ログには行われた操作の種類、操作を行ったノード、操作後のノードの位置、開始時間、終了時間が操作ごとに行われた時間順に記録されている。無駄時間の計測は、このログの操作を行ったノードと終了時間を使い計測する。既に確認したノードが操作を行ったノードとして出現した場合、その操作の終了時間と、その操作の一行上の操作の終了時間を確認する。前者から後者を減算することで、操作内容を考える時間を含めた無駄時間を算出できる。

取得した無駄時間の集計を、表 2 に示す。前述した通り、被験者甲 4 のデータは平均値の計算には含めていない。再構成操作所要時間と同じく、乙チームは先に手動で多次元ツリーの再構成を行っていたため、甲チームの半分ほどの時間で作業を完了していることが分かった。両チームの被験者 4 は他の被験者と比較すると、少ない無駄時間で再構成が完了している。これは利用したファイルパスによって作られる単一ツリーが、少ないカテゴリで分類できるノードで構成されていたのではないかと推測できる。

4.3.3. 自動構成機能による省力効果

今回利用したファイルパス群は、過去に我々が手作業で多次元ツリーに再構成し、データの提供元によるレビューを受けている。この多次元ツリーを模範解答とし、自動構成にどれほどの省力効果があるかを考察する。

自動構成した多次元ツリーと元の一次元ツリーに対し、MD-TACT の分割操作と移動操作を用いて、模範解答を再現する最小限の操作数（移動が必要なノード数）を求める。これに基づいて、以下の式により作業量削減率を定義する。

$$\left(1-\frac{\textit{自動構成ツリーからの移動ノード数}}{\textit{元ツリーからの移動ノード数}}\right)\times 100$$

現在、ファイルパス群 1 および 2 に対する解析が完了している。ファイルパス群 1 では 3 つのツリーが自動構成された。このうち分割操作が必要だったノードは 57 個、移動操作が必要なノードは最少で 152 個だった。ファイパス群 2 では 4 つのツリーが自動構成され、このうち分割操作が必要だったノードは 68 個、移動操作が必要なノード数は最少で 172 個だった。

続いてファイルパス群 1、2 の元ツリーの確認を行った。結果は表 3 に示すとおりである。移動操作が必要なノード数の項目の（）内には、自動構成ツリーからの移動が必要な最少ノード数を示す。

ノードに対する分割操作数は元ツリー、自動構成ツリーのどちらも同じだった。移動操作が必要な最少ノード数は若干自動構成後の方が少ないが、手動洗練の手間が大幅に省けるほどではなかった。作業量削減率にして、ファイルパス群 1 では約 19%、ファイルパス群 2 では約 23%、自動構成後の作業量が削減されている。このことから、自動構成機能には一定の省力効果

表 3. 元ツリーと自動構成ツリーから模範解答を再現する手間の比較

	ファイルパス群 1	ファイルパス群 2
総ノード数	224	279
重複ノード数	20	43
分割操作が必要なノード数	57	68
移動操作が必要なノード数	186（152）	223（172）
残せるノード数	99	96

があることは分かるが、改善の余地も大きいと判断される。また、重複ノード数が多ければ多いほど作業量削減率が上がることが予想される。

5. MD-TACT の改良に向けた考察

MD-TACT の評価結果と DEIM2014 での質疑応答を踏まえ、MD-TACT の改良点や新機能について考察する。

ノード分割時のリネーム機能の追加

MD-TACT の手動洗練機能にリネーム機能の追加が提案された。これは表示されているノードの名前を変更する機能である。現在 MD-TACT の分割機能で生成されるノードは、名前を自由に変更できない。しかし、1 つのノードが複数の観点の単語を含む例はしばしば見られる（例：図 4 の「20 年度指名業者推薦書」）。このような場合、ノード分割後にノード名を個別の単語に合わせることで重複単語として処理できる。また、ノードに対して形態素解析を行うことで単語に自動分割する機能も、多次元ツリーの自動生成を促進する上で有効だと思われる。

複数ノードの統合機能の改良

複数ノードの統合機能では、1 つのノードを選択後、統合機能の中で統合対象ノードを選ばせていた。しかし、統合したいノードを複数選択した後で統合機能を選ぶように改良することで、統合対象ノードの選択や 3 つ以上のノードの統合が容易に指定できるようになる。

自動構成機能の強化

重複ノードの少ないツリーの場合、複数の分類観点の単語が混在するなど、多次元ツリーの自動構成機能の効果が低下する。こうした状況を改善するためには、頻繁に出現する分類観点を表現するツリーをリポジトリに保持しておき、そのツリーとのマッチングを通じて多次元ツリーの自動洗練を行う機能が有効だと考えられる。

また、複数の利用者によるリポジトリの共有や、利用者によるリポジトリへのツリーの登録、リポジトリに登録されたツリーの評価などの機能を提供することで、コンテンツとしてのリポジトリの価値が高まることが期待できる。

次元集約

多次元ツリーを構成する複数のツリーにおいて、一方のツリーとファイルの対応関係が、他方のツリーとファイルの対応関係と相関関係にあるケースが見られる（例：住所と郵便番号、身長と体重など）。このような場合、利用者のニーズと合致するツリーのみを表示することで次元集約を行い、ファイルの検索能力を低下させることなく、利用者に提示する多次元ツリーを単純化できる。

設計改良

MD-TACT の単語辞書には単語 ID、カテゴリ ID、単語名が記録されているが、これに「表示名」を追加することで、ノード分割時等の新たなノード名を自由に指定できるようになる。また、多次元ツリーを表現するデータ構造をビューとモデルに分離することで、より系統的な設計が可能になる。

6. まとめ

本論文では、MD-TACT の開発について述べ、評価実験の結果を分析した。MD-TACT により、単一ツリーを短時間で多次元ツリーに再構成することが可能になり、再構成を反復適用することで自動的に多次元ツリーを洗練できる可能性が判明した。MD-TACT のログ機能を使い、熟練者による多次元ツリーの再構成作業を記録・分析すれば、より精度の高い多次元ツリーの自動構成アルゴリズムを工夫できる可能性がある。

より多くの利用者に MD-TACT を使ってもらうためには、このツールの有用性を証明する必要がある。今回の評価実験では、時間の短縮の面において、確実に有用であることが証明された。自動構成アルゴリズムには一定の省力効果があるが、ユーザーインターフェースの面からも undo 機能がない等、操作性の面で問題もあることが分かった。今回の評価実験を通じて、MD-TACT には改善の余地があることが明らかになった。

今後は評価実験を通じて得られたログデータ等の詳細な分析を行い、多次元ツリーの品質の観点からも MD-TACT の有用性を検証する。この有用性を証明できれば、情報系の職員や教員に使ってもらい、利用データから効率的な多次元ツリー構成アルゴリズムを開発する。また、ユーザーインターフェースの改善も行っていく予定である。

参 考 文 献

[1] IDC,「The Hidden Costs of Information Work」, 2006

[2] 山口章太,「ファイル整理ツール HyperClassifier における移行支援ツールの評価とその改良」, 平成 21 年度佐賀大学理工学部知能情報システム学科卒業論文

[3] 掛下哲郎, 園木幸寶,「OLAP 操作を活用したファイル整理ツール HyperClassifier」, 第 8 回情報科学技術フォーラム(FIT 2009), 2009.

[4] 柿本由気.「ファイル分類のための多次元ツリー手動洗練ツールの開発」. 佐賀大学理工学部知能情報システム学科卒業論文,2013.

[5] 山口章太,掛下哲郎,「ファイル整理ツール HyperClassifier における多次元ツリー自動構成ツールの開発と評価」,第 3 回データ工学と情報マネジメントに関するフォーラム D6-3,2011.

[6] 柿本由気, 掛下哲郎、「系統的なファイル整理を目的とする多次元ツリー構成ツール MD-TACT」、電気関係学会九州支部第 65 回連合大会 06-2P-01、2013.