

OLAP 操作を活用したファイル整理ツール HyperClassifier

HyperClassifier: A File Arrangement Tool using OLAP Operations

掛下 哲郎[†] 園木 幸寶[‡]
Tetsuro KAKESHITA Yukitaka SONOKI

1. はじめに

情報社会の進展に伴い、あらゆるオフィスにおいて大量の電子ファイルが蓄積されている。現代における組織活動（営利企業のビジネス活動だけに限定するものではなく、国・地方自治体、病院、学校、非営利組織等における活動も同様である）は情報を基盤としており、組織内で生み出されるデータの量や、組織外部から取得するデータ量は急速に増大している。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レポートもある[1]。デジタル化の重要な狙いはコンテンツの再利用であるが、これを効果的に実現するためには、蓄積されたファイルが系統的に整理されていることが不可欠である。また、最近では、企業等におけるIT統制の必要性も高まっており、特にデジタル化された情報に対するきめ細かなセキュリティ管理、アクセス管理、ファイル管理等の重要性が高い。これらを実現するためには、様々な条件を指定してファイル等を絞り込み、得られたファイル等を利用者が望む各種の観点に基づいて整理した上で提示する必要がある。

我々はこれを系統的に行うために、ファイル整理ツール HyperClassifier を開発している。HyperClassifier は、我々が提案した多次元分類方式[2,3]を活用してファイルの系統的な整理を支援する。さらに、分類されたファイルに対しては OLAP 操作[4]を活用して柔軟な検索および分析を行うための機能を提供する。OLAP 操作は従来、データウェアハウスで管理されている多次元データ[6,7]を分析するために用いられることが多かった。これに対して、HyperClassifier は、OLAP 操作をファイル整理やファイルサーバの可視化に活用した点で新規性を有する。また、通常の OLAP 操作が提供するデータ分析機能に加えて、HyperClassifier はファイルの登録機能を統合したシンプルな利用者インタフェースを提供している。

本論文では HyperClassifier の概要を紹介して、その予備的な評価を行なう。

以下、2章では多次元分類方式を、3章では OLAP 操作を基本的事項として紹介する。4章で HyperClassifier の機能と利用者インタフェースを説明し、ファイル検索処理が OLAP 操作に対応していることを示す。以上の提案手法に対する予備評価の結果を5章に示す。6章では関連研究との比較を行う。

2. 多次元分類方式

多次元分類方式は、分類対象となるエンティティを複数の木構造と対応付けて分類する方式である。エンティティは電子的に保持されている情報であれば任意だが、HyperClassifier ではファイルをエンティティとする。

それぞれの木構造は、特定の観点に基づいたエンティティの分類基準を表現しており、親子の節点間には IS-A 関連（子節点は親節点の特別な場合に対応）が成立するように構築する（IS-A 制約）。また、同一木構造の兄弟節点は互いに排他的なものとする（排他制約）。IS-A 制約により分類基準の一貫性を保つことができる。また、排他制約によって分類基準の明確化が図られる。これにより MECE な木構造が構築され、木構造の理解も容易になるため、ファイルの分類作業や検索作業の効率が高まる。上記の制約を満たす木構造は、既存の分類基準を活用すれば構築できる場合もある。ファイルを分類する際に活用できる基準について、3種類の木構造定義例を以下に示す。

プロジェクト：企業等の事業内容および各事業に含まれる具体的な案件に基づいて木構造を定義できる。大学教育の場合、教育課程に関する規程等に基づいて木構造を定義できる（例：全学教育科目、語学、専門教育科目）。また、研究を分類する際にも、科学研究費補助金におけるテーマ分類等のような既存の分類基準に基づいて木構造を定義できる。

ファイルの作成者：企業等の組織図や各種委員会の体系に基づいて木構造を定義できる。これにより、同一人物が複数の役割（例：A課の課長とB委員会の委員）を兼ねていた場合、役割毎にファイルの作成者を区別することもできる。

文書の目的：事務文書については文書管理規程に基づいて、ソフトウェア文書についてはドキュメンテーションガイドライン等に基づいて文書の種類を列挙できる。これにより木構造を定義できる。

エンティティは、各木構造について1つ以上の節点と対応付ける。すなわち、必要に応じてエンティティを複数の節点と対応付ける場合もある。すべてのエンティティを少なくとも1つの節点と対応付けることにより、迷子エンティティをなくすことができる。なお、いずれかの木構造において、エンティティと既存節点の対応が取れない場合、当該木構造（属性）に対応するエンティティの属性値が空値だと考えられる。空値には以下に示すようにいくつかの種類がある[5]。

非存在の空値：対応する値が存在しない。

不明の空値：対応する値が不明である。

無情報 (no information) の空値：対応する値は存在するが、具体的な値は公表していない。または、（機密保持等の理由により）対応する値の存否を含めて公表していない。

これらの空値を必要とする場合には、空値の種類に対応する節点を当該木構造上に定義してエンティティと対応付ける。

[†] 佐賀大学 Saga University

[‡] (株)ネクステップ Nextep Co. Ltd.

多次元分類方式の特徴としては以下が挙げられる。

1. 分類に用いる個別の木構造が一貫しており、比較的小規模になるため、利用者による木構造の理解および保守が容易である。
2. 個別の木構造や木構造の節点を利用者が自由に指定してエンティティを検索できる。木構造や節点の指定順序も自由に指定できるため、検索の際の自由度が大きい。
3. 蓄積されている情報の全体像を、利用者が容易に把握できる。これにより、所望の情報がシステムに登録されているか否かを素早く判断できる。

3. OLAP 操作

多次元データは、エンティティに対して複数の属性が付与されているデータの総称である。OLAP 操作は多次元データを様々な観点から柔軟に集計する機能を提供することで、そのデータの全体像だけでなく、利用者が興味を持つデータの特定部分についても分析を行う機能を提供する。そのためにダイシング、スライシング、ドリリングの3種類の操作を組み合わせ用いる。

3.1 ダイシング

n を属性数とすると、多次元データは n 次元空間に分布しているため、それを直接可視化することはできない。そこで、データを分析する際に用いる次元を利用者が指定して、それに基づいて集計結果を表示する。その際、次元を選択する操作をダイシングと呼ぶ。学生情報を集計する際の基準として、所属学部、入学年度、出身高校などを選択する操作がダイシングの例として挙げられる。

3.2 スライシング

多次元データに対して特定の条件を適用し、分析の対象とするデータの絞込みを行う操作をスライシングと呼ぶ。学生情報を集計する際に、「理系学部」の学生だけを対象に集計する操作などがスライシングの例として挙げられる。

3.3 ドリリング

多次元データの各次元に属する属性値は一般に階層的に構成されている。階層の上方はより抽象的な分類に、下方はより具体的な分類に対応しているが、特定の次元について、集計に用いる階層のレベルを上方に移動する操作をドリルアップ、下方に移動する操作をドリルダウンと呼ぶ。ドリリングはドリルアップとドリルダウンの総称である。ドリルダウンの例としては、学生情報を集計する際に、学部毎の集計を学科毎の集計に切り替える操作などが挙げられる。なお、学科毎の集計を学部毎の集計に切り替える操作はドリルアップの一例である。

4. HyperClassifier の機能

HyperClassifier は、(1) オフィスにおいて様々なファイルを共有したい、(2) 共有したファイルを、プロジェクト、作成者、文書の目的等の多様な基準で分類したい、(3) 分類したファイルの分布状況を様々な角度から分析・把握したい、(4) ファイル整理に要する手間をできるだけ減らしたい、といったニーズに応えることを目的として開発している。従って、HyperClassifier は、個人での利用も可

能ではあるものの、主として部署単位やプロジェクトチーム等のグループでファイルを共有する場面を想定して開発している。本章ではファイルとタグの対応付け機能およびファイルの検索機能に限定して説明するが、通常ファイル管理ソフトウェア(例: Explorer 等)と同等の機能は備えている。

4.1 全体構成

図1に HyperClassifier の全体構成を示す。HyperClassifier は、DB、HyperClassifier サーバ、HyperClassifier クライアントの3階層から構成されている。

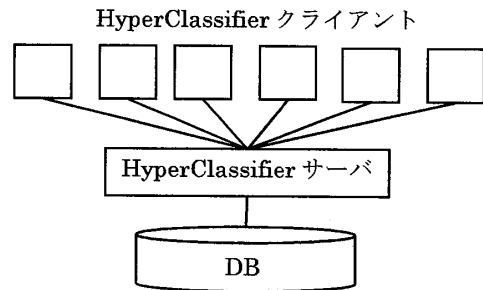


図1 HyperClassifier の全体構成

DB には、管理対象となっているファイルおよびその URI、ツリー（分類のために用いる木構造）、タグ（ツリーの各ノード）、ファイルとタグの対応関係、ユーザープロフィール等が登録されている。

クライアントは、利用者に対するユーザインタフェースを提供する(図2)。利用者は、分類に用いるツリーの閲覧と編集、ファイルの登録とタグ付け、ファイルの検索をクライアント上で行える。また、ツリーを構成する各タグには対応するファイル数が常に表示されている。これによってファイルの分布状況を利用者に表示する。

サーバはクライアントからの処理要求に基づいて DB をアクセスし、必要な処理を行う。この中には、ツリーの参照・更新、ファイルの追加・削除、ファイルとタグの対応付けの追加・編集・削除などが含まれる。また、利用者が指定した条件に基づいて各種の検索操作に必要な処理を行なう。

OLAP 操作はデータウェアハウスが管理している多次元データを分析することを目的としている。データウェアハウスが管理するデータは、「意思決定のために、目的別ごとに編成され、統合化された時系列で、更新処理をしないデータの集まり」であり、エンティティと属性の対応付けは ETL ツール (Extract, Transform and Loading) 等の別の枠組みで編集される。これに対して HyperClassifier ではファイルとタグの対応付けも一般の利用者(ファイルの登録者)が行うことを前提としており、OLAP 操作によるデータ分析と対応付け操作をクライアントが提供する単一のユーザインタフェース上で実現できるように工夫している。これにより、ユーザインタフェースの統合を図れると同時に、利用者が対応付けを編集する際に、スライシング等の OLAP 操作を組み合わせ用いることもできるので、編集作業が容易になる。

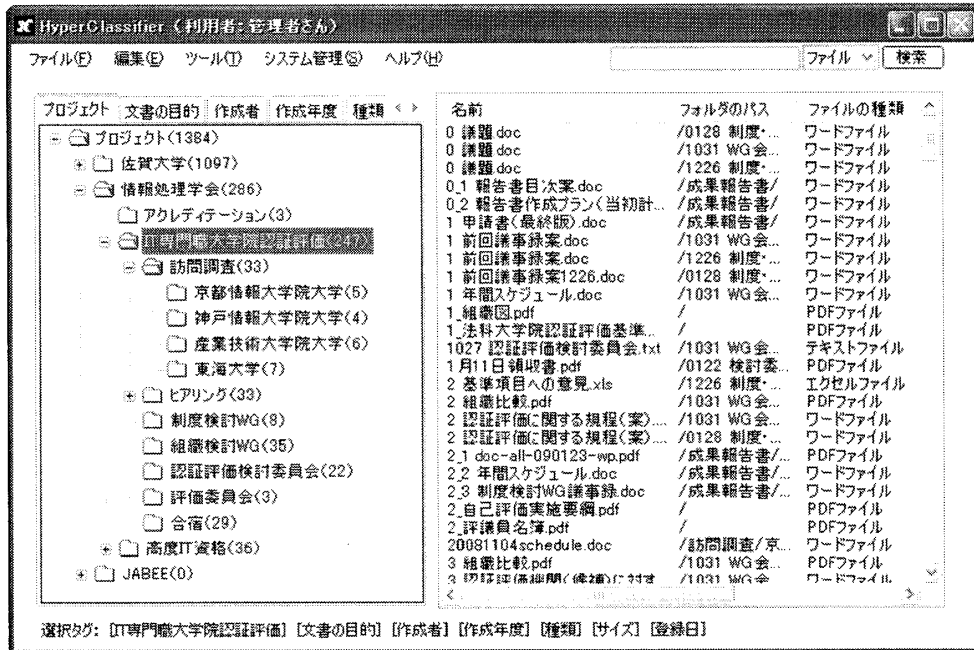


図2 HyperClassifier クライアント画面

4.2 ファイルの登録とタグ付け

HyperClassifier は、ファイルを整理するためのツリーを最大 10 個まで定義することができる。このうち、表 1 に示す 4 つのツリーは事前に定義されており、利用者がファイルを HyperClassifier に登録した時点で、ファイルの属性値を取得してタグとの対応付けが自動的に設定される。

表 1: ファイルとの対応付けが自動設定されるツリー

ツリー	ファイルの属性値
種類	ファイルの拡張子
サイズ	ファイルのサイズ
登録日	ファイル登録時点の時刻
登録者	利用者 ID

これに対して、利用者が定義したツリーについては、利用者がタグとファイルの対応付けを行う必要がある。ファイルの登録および、ファイルとタグの対応付けを行うために、HyperClassifier は以下の機能を提供する。

ツリーの編集機能

- ・ ツリーの新規作成
- ・ 部分木のカット、コピー、ペースト、削除

タグの編集機能

- ・ タグの新規作成、ツリーとの対応付け
- ・ タグの名前変更
- ・ タグのカット、コピー、ペースト、削除

ファイルとタグの対応付け

- ・ ファイルの登録・削除
- ・ ファイルとタグの対応付けの追加・編集・削除

これらの機能を提供するクライアントの画面を図 2 に示す。画面左側に表示されているツリーは、ツリーの名称

を表示したタブを変更すれば、自由に切り替えることができる。ツリーおよびタグに対する編集機能は、この画面で提供される。

新しいファイルを HyperClassifier に登録してタグに対応づける処理は、当初から設定されていた 4 つのツリーに関してはシステム (HyperClassifier) が自動的に行う。それ以外のツリー (利用者が定義したもの) に関しては、利用者とシステムの間で以下のやり取りに従って行われる。

1. 利用者は、各ツリー上で、ファイルに対応づけるタグをマウス等のポインティングデバイスで選択する。現在のバージョンでは各ツリー上で選択できるタグは 1 つだけであるが、1 つのツリー上で複数のタグを同時に選択できるような拡張も比較的容易に実現できる。
2. 利用者は、登録したいファイルを 1 個以上選択してファイル一覧が表示されている場所 (画面の右側) にドラッグ&ドロップする。なお、利用者はフォルダ (サブフォルダを含んでも良い) を選択してドロップすることもできる。
3. システムは、指定されたすべてのファイルを、各ツリー上で指定されたタグに一括して対応づける。なお、フォルダがドロップされた場合には、システムは当該フォルダに格納されていたすべてのファイルを、指定されたタグに対応づける。
4. 利用者がツリー上でタグを選択しなかった場合、システムはファイルを当該ツリーのルートタグに対応づける。なお、ルートタグには、「対応タグの未指定」という意味だけでなく、「すべての子孫タグと対応」という意味を与えることもできる。両者を区別する必要がある場合には、それぞれの意味に対応するタグを定義して対応付ければ良い。

ファイルとタグの対応付けに関する操作はツリー毎に独立に行う。ファイルとタグの対応関係を変更する際に

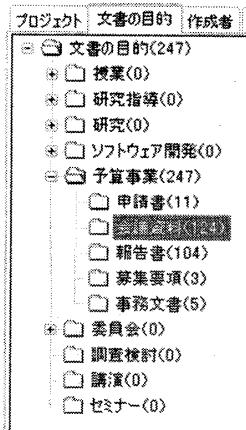


図3 ツリーの切り替え (ダイシングとスライシングの組み合わせ)

は、ファイル (1 個以上) を選択して、新たに対応付けたタグにドロップする。選択したファイルをルートタグにドロップすることでファイルとタグの対応関係を未整理状態に戻すことができる。ただし、各ファイルと各ツリーの間にはタグによる対応関係が定義されている必要があるため、対応関係をすべて削除することはできない。また、ファイルをタグに Shift-ドロップすることにより、当該ファイルに新たなタグとの対応を追加できる。ただし、ファイルを複数のタグと対応付ける場合、これらのタグの間に先祖-子孫関係があってはならない。そのような関係があった場合には警告を表示した上で Shift-ドロップされたファイルを先祖タグに対応付ける。なお、複数のタグ t1, t2 に対応づけられたファイル f があった場合、ツリー上でタグ t1 を選択した状態でファイル f をドラッグ (または Shift-ドラッグ) することで、t1 と f の対応関係だけを編集できる。

4.3 ファイル検索と分布表示

HyperClassifier におけるファイル検索は、図 2 に示すクライアント画面で行う。図 2 に示すように、画面に表示されている各タグについて、対応するファイル数が表示されるため、利用者はファイルの大まかな分布状況を常に把握できる。

利用者は、ツリーの切り替えおよびツリーを構成するタグの選択を組み合わせる検索を行う。

利用者がツリーの切り替え (例えば図 2 から図 3) を行うと、対象となるファイルの分布状況を異なる分類軸で参照できる。この操作はダイシングに対応する。複数のツリーを同時に選択してクロス集計を求めることはできないが、単一のツリーによる集計は実現している。

また、画面に表示されているツリー上で、上位のタグを表示する操作はドリルアップに、下位のタグを表示する操作はドリルダウンにそれぞれ対応する (図 4)。

利用者が画面に表示されているツリー上でタグを選択すると、指定されたタグに対応するファイルだけが右側の画面に一覧表示される。この操作はスライシングに対応する。図 2 の例では「プロジェクト=IT 専門職大学院認証評価」という選択条件によってファイルの絞り込みを行っている。複数のツリーでタグを指定した場合には、

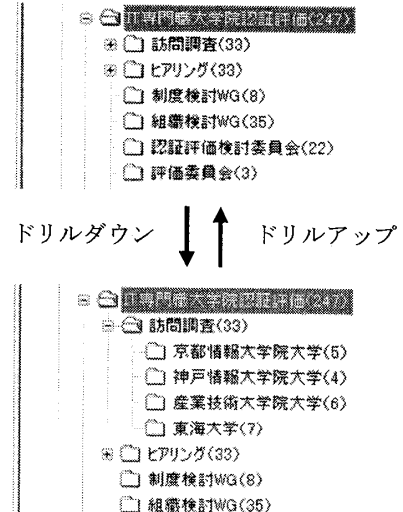


図4 ドリリング操作

それらの選択条件をすべて満たすファイルが検索される。HyperClassifier は利用者インタフェースの単純さの維持を優先し、現状では OR 条件による検索には対応していない。

なお、図 2 に示した例では、ファイル一覧には「IT 専門職大学院」タグに直接対応づけられているファイルだけでなく、その子孫タグに対応づけられているファイルも含まれている。しかし、指定したタグに直接対応づけられているファイルだけを検索するように指定することもできる。この指定はツリー毎に行うことができる。

また、タグを指定した状態でツリーの切り替え (ダイシング) を行うと、集計対象を絞り込むことができる。例えば、図 2 で指定した選択条件によって絞り込まれた 247 個のファイルを対象としてダイシングを行い、文書の目的毎に集計したものが図 3 である。

以上で示したように、HyperClassifier はクライアント側でのマウス操作により、複数のツリーを組み合わせるダイシング操作を除いて、3 種類の OLAP 操作 (ダイシング、スライシング、ドリリング) を実現している。

5. 予備評価

4 章で提案した HyperClassifier の有効性を確認するために、我々は様々な企業や地方自治体等で HyperClassifier の機能を説明して意見を求めた。聞き取り調査によって得られた意見 (15 社・団体から得られたもの) をまとめて以下で紹介する。HyperClassifier に対する機能追加の要望もあるが、4 章で述べた機能が高く評価されていることが分かる。

- 様々な属性から文書を探し出すことが可能な点が素晴らしい。Explorer と同等の操作性を持っている点も良い。
- 様々な角度からファイルの絞り込みを行いたいと思っていた。多次元ビュー、串刺し検索が可能である点で非常に有効である。
- 使い勝手が非常に良いと思う。利用者の PC スキルを考慮すると、あまり多機能にするよりも使いやすいシステムを提供してもらおうのが一番良い。
- 操作性に違和感がない点が良い。
- シンプルに使用できそうところが良い。

- 各種のデジタルデータの管理にはずっと頭を悩ませている。HyperClassifierは、とても良いシステムだと思う。
- ファイル整理のために、フォルダ構成を改める必要性を感じていた。HyperClassifierを提供してもらえば、使用部署を募って試用してみたい。
- 現状ではサーバ内のファイルが全くの未整理状態で、文書整理の必要性が多いにある。
- 一般的なファイルサーバより間違いなく便利だと思う。
- ファイルに対してアクセス制限を容易に設定できるシステムを望んでいた。各種のOLAP操作で絞り込んだファイルにアクセス権が一括設定できる点が非常に素晴らしく、求めていたものである。
- バックアップ機能、グループ単位での管理機能、タグの付け替え作業、ファイル更新時のログ保存について等、多数の質問があった。
- 実装済みの機能に加えてActive Directoryと連携したユーザ管理機能や全文検索機能が提供できると強力である。
- 便利だと思う、あったら良いと思う、非常に面白い、とてもユニーク、便利で使えるシステムだと思う等の意見が多数あった。

6. 関連研究

Unix系、Windows系、Macintosh系などを問わず、OSが管理するファイルを整理するための一般的な手法は、ディレクトリやフォルダによる階層的分類である。しかし、単一の木構造でファイルに対する多様な分類基準を表現すると、木構造中に異なる視点の分類基準が混在してIS-A制約等を満たすことができないため、一貫したものにならない。その結果、利用者が木構造を理解するのが難しくなり、ファイルを探す際にも手戻りが発生する原因になる。

また、単一の木構造でファイルを分類した場合、木構造が大規模化する傾向にある。例えば、K個のファイルを単一の木構造で分類した場合、1つの葉フォルダに格納されているファイル数を10個程度(認知心理学の成果[8,9]により、人間が一目で全体像を把握できる限度)に抑えるためには、木構造の葉フォルダが $K/10$ 個程度($K=10000$ の場合、1000程度)必要になる。これに対して、 n 次元の木構造を用いた場合、各次元が互いに独立していれば¹、各次元の木構造の葉フォルダ数の積と等しい分類区分が提供できる。各木構造の葉フォルダ数が等しいと仮定すれば、各木構造の葉フォルダの数は $(K/10)^{1/n}$ 個程度($K=10000$, $n=3$ の場合、10程度)で単一の木構造を用いた場合と同等の分類ができる。そのため、HyperClassifierにおける個別の木構造はより小規模なもので済む。この性質は、木構造の保守性や理解容易性を確保する上で有効である。

一方、Google等が採用しているアプローチは、階層的な分類を行う替わりに、ファイルにラベル付けを行うことで整理を促進しようとするものである。HyperClassifierにおけるタグとファイルの対応付けは、このアプローチと類似している。しかし、利用者による自由なラベル定義を行うと、全体としての分類基準の一貫性の維持や、

ラベルの総数が大きくなった場合の系統的なラベル管理が難しい。これに対して、HyperClassifierでは、既存のラベルを一貫した木構造にまとめて系統的に提示している。提示したラベルの階層は、利用者がファイルに対応付けるラベルを探す際はもちろんのこと、新たなラベルを定義する際の指針としても有用である。

また、HyperClassifierと同様、ファイル整理やファイルサーバの可視化を目的とする商用システムもいくつか存在する。

このうちSAVVY/EWAP[10]は、独自開発の全文検索システムSAVVYと多視点ツリーを利用したファイル検索によって企業内情報の「見える化」を図る。「多視点ツリー」で構築されるツリーは1つだけであり、フォルダの各階層と分類基準が対応している。検索時には、階層(=分類基準)を自由に組み替えて多視点ツリーを構築する。これによって、利用者はフォルダの階層構造にとらわれず、自由な順序でファイルを検索できる。ただし、既存のフォルダ階層の各レベルをそのまま分類基準に対応づけているため、同一階層に異なる分類基準が混在した場合(そのような場合が現実的には多い)には多視点ツリーのメリットが失われる。また、多視点ツリーで指定した階層の順序を変更したい場合には、ツリーの上方まで戻って再設定する必要があるため操作に手間がかかる。

Document Storage[11]は「業務」「担当者」「アプリケーション」「カレンダー(データの更新年月)」「任意項目(取引先など)」の5つの観点をういてファイルを分類することができる。しかし、分類の観点が5つに固定されている、1ファイルにつき各観点1項目ずつしか対応付けられないなど、HyperClassifierと比較すると、ツリーによる分類・検索の自由度が低い。

ConceptBase V [12]は文書に含まれる属性やメタデータを観点ごとに表示し、それらをプルダウンリスト・ラジオボタン・チェックボックスなどから選択していきにより検索が行える「ファセット・ナビゲーション」という機能を持つ。しかしツリー構造には対応していない。

また、上記の既存システムに共通する課題として、ファイル登録機能は別システムに分離されている点が挙げられる。OLAP操作を行うための利用者インタフェースの研究はいくつかある[13,14,15]が、データ操作と対応付けの編集を分けて扱っているものが多い。これに対してHyperClassifierでは両者を統合した利用者インタフェースを提供している。

7. おわりに

本論文では、多次元ツリーとOLAP操作を活用してファイルの整理を系統的に行うためのツールHyperClassifierを開発した。予備評価の結果も良好であり、今後、定量的な評価を行なってツールの有効性を確認する予定である。

HyperClassifierを導入する際には、既存のファイルシステムを多次元ツリー構造に変換し、個別のファイルを多次元ツリー構造と対応付ける作業が必要になる。この作業を効率化するために、我々は移行支援ツールを開発している。移行支援ツールを用いると、既存のファイルシステムの階層構造からフォルダ名を切り出すことができる。利用者がフォルダ名とツリーのノード名の対応を指

¹ 木構造の次元が独立しておらず、木構造Aと木構造Bの間に対応関係が定義できる場合、その対応関係を活用して木構造とファイルの対応付けの自動化が可能になる。

定すると、移行支援ツールはそれによって多次元ツリー構造の生成や、多次元ツリー構造と既存ファイルとの対応付けを自動的に行う。

今後の研究課題としては、検索操作の高速化や大規模な組織への適用も考慮したスケールアップへの対応を進めているところである。また、ファイルとタグの対応付けの自動化やツリーの自動生成などにも取り組み、利用者の負担を抑えると同時に、より客観性の高いファイル分類が行えるように改良したいと考えている。

参考文献

- [1] The Hidden Costs of Information Work, IDC (2006).
- [2] 掛下 哲郎, 原楨 稔幸, “多次元分類: 木構造分類とキーワード分類の複合的アプローチ”, 情報処理学会論文誌: データベース, Vol. 42, No. SIG 1(TOD 8), pp. 131-139 (2001).
- [3] 井ノ口 励, 掛下 哲郎, “多次元分類方式における木構造構成の自動化”, 情報処理学会論文誌: データベース, Vol. 44, No. SIG 12 (TOD19), pp. 33-42 (2003).
- [4] Codd, E. F., Codd, S. B., Salley, C. T., “Providing OLAP (online analytical processing) to user-analysis: an IT mandate”, Technical Report, E. F. Codd and Associates (1993).
- [5] 上林 弥彦, データベース, 昭晃堂 (1986) .
- [6] Agrawal, R., Gupta, A., Sarawagi, S., “Modeling multidimensional databases”, Proc. ICDE, pp. 232 - 243 (1997).
- [7] Pedersen, T. B., Jensen, C. S., “Multidimensional database technology”, *IEEE Computer*, Vol. 34, No. 2, pp 40-46 (2001).
- [8] 御領 謙, 江草 浩幸, 菊地 正, 認知心理学への招待—心の働きとしくみを探る, サイエンス社 (1993).
- [9] 道又 爾, 大久保 街亜, 山川 恵子, 北崎 充晃, 今井 久登, 黒沢 学, 認知心理学—知のアーキテクチャを探る, 有斐閣 (2003).
- [10] ジップインフォブリッジ, SAVVY/EWAP (<http://www.info-brdg.co.jp/savvy/ewap.html>)
- [11] エプソン, Document Storage (<http://www.epson.jp/products/offirio/sw/document/storage.htm>)
- [12] ジャストシステム, ConceptBase V (<http://www.justsystems.com/jp/km/product/cb5/index.html>)
- [13] Choong, Y.-W., Laurent, D., Marcel, P., “Computing appropriate representations for multidimensional data”, DOLAP (2001).
- [14] Maniatis, A. S., Vassiliadis, P., Skiadopoulos, S., Vassiliou, Y., “Advanced visualization for OLAP”, DOLAP (2003).
- [15] Pasumansky, M., “Microsoft OLAP and Microsoft Analysis Services”, <http://www.mosha.com/msolap/> (Last Update 2008).