社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－DBS－141 (16)
2007－GN－ 62 (16)
2007／1／26

タグ情報に基づくファイル管理システム

阿部 淳也 出石 大志 杉上 裕一 堀 幸雄 今井 慈郎

香川大学

概要 - 伝統的な木構造を用いたファイルシステムが広範に使用されているが，各ファイルのコンテンツによる分類や関係付けの有効な手法が不足している．そのような単純なファイルシステムでは，キーワードを指定した効果的なファイル検索ができないという問題を抱えている．コンテンツに基づく情報検索を可能にするため，各ファイルに関するタグ情報を活用するファイル管理の新しい手法を設計している．本報告では，我々が作成しているファイル管理システムのGUIを紹介し，併せて，形態素解析によるファイル属性からのキーワード抽出，DBMSによるキーワード操作およびキーワードに基づく情報検索などを用いた，プロトタイプ実装についても言及する．

Proposal of an Improved GUI for File Management System with Help of Tagged Information

Junya Abe, Hiroshi Izuishi, Yuichi Sugiue, Yukio Hori and Yoshiro Imai

Kagawa University

Abstract - Conventional tree-structured file systems have been widely used, but they have lacked a useful mechanism to classify and relate their files according to the contents of each file. And such simple file systems are suffering from efficient retrieval of their files by specifying keywords. In order to perform content-based information retrieval, a new scheme of file management is designed to utilize tagged information about each file. In this report, we will introduce a GUI of our file management system. And we will describe its prototype implementation by means of keyword extraction from file attributes with morphological analysis, keyword manipulation through DBMS and information retrieval based on keywords.

1．はじめに

各人の研究活動で作成されるファイル群を研究室内において共有することは，研究についての議論・討論の円滑化を促し，研究の引継ぎにおいてもファイル移動の手間や不注意による損失を低減することが期待できる．このように，ファイル共有を効果的に実現することは，ファイル管理システムの運用を検討する上で重要な観点であり，利便性の高低を図る基準である．ところが，多くの研究室などでは，規模の大小によらず，以下に述べるような要因でファイル共有が有効に実現できていないという問題がある．

例えば，本研究室ではネットワークHDDを用いたファイル共有を行ってきた．ネットワークHDD内には，ユーザごとの専用のユーザディレクトリと共用ディレクトリを設けていた．共用ディレクトリでは他のユーザの利用を考えたディレクトリ階層を指定する必要があるが，他者も利用する性質上完成形のファイルが置かれることが多い．成果物の保管庫という扱いが中心となり，作業用ディレクトリとして複数ユーザが使用するといった利用形態が敬遠される傾向がみられる．

一方，ユーザごとに割り当てた専用ディレクトリでは，個々のディレクトリ階層がユーザごとに異なり，アクセス権限を付与された場合でも．効率よく他者のファイルを見つけることが困難な場合も少なくない．一般に，このような理由のため，共用を目的としたネットワークHDD全体や個別ディレクトリの利用に対する敷居が高くなり，結果としてファイル共有の利便性が期待できないことも多い．

本研究では，複数のユーザ間による効果的なファイル共有と，サーバ側，クライアント側のHDDを意識させない共有環境を提供する，ファイル管理システムの実現を目指している．ファイル検索（含むディレクトリ検索）の効率を上げる仕組みとして，フォークソノミーの活用という観点からキーワードに基づいた検索機能の実装を行っている．GUIの改善によるファイル管理の効率化を図り，キーワード自動抽出など，検索効率向上ついて検討している．

本稿では，既存の共有手法を紹介し，キーワード抽出にふれ，システム構成を紹介する．また，キーワード分析についても論じている．

2．既存の共有手法の調査

2.1 先行研究について

(1) Semantic File System

David らは，柔軟なファイルアクセスを実現する Semantic File System（SFS）を提唱した[1]．コマンドライン上でのファイル操作において，従来のコマンドに専用のクエリを付与することにより，ファイルアクセスの効率を高めている．

効率的なファイルアクセスを実現するため，各々のファイルには属性情報を持たせている．ファイルの種類に応じてそれぞれの Transducer を用意し，ファイル属性情報を自動抽出している．例えば，オブジェクトファイル（"*.o"）からは，作者名，エクスポートしているプロシージャ名，インポートしているプロシージャ名などを，メールファイル（"*.txt"）からは，送信者名，受信者名，件名などを，そして TeX ファイル("*.tex")からは，著者名，セクション名などを，それぞれ属性情報として抽出する．

UNIX のファイルシステム上で実装し，ユーザインタフェースとファイルシステムの中間に SFS サーバを配置し，バーチャルディレクトリモジュールによってクエリを解釈し，ファイルアクセスを行う．

この研究では，ファイルの種類に応じた Transducer を用いることで，より適切な属性情報を抽出している．しかし，ファイルの種類が多岐にわたれば， Transducer を対応させるためのオーバーヘッド増加が問題となる．

(2)階層的キーワードに基づく名前管理手法とそれに基づくファイル共有手法

轟木らは，階層的キーワードベースの名前管理法とファイル共有に関する研究を行っている[2]．複数ユーザによる共有環境において，各ユーザの名前空間を統合することにより，ファイル分類の自由度の増加を抑制している．

階層的キーワード名前管理では，階層的名前管理と属性ベースの名前管理の特徴を組合せて，ファイルに対するキーワード集合に階層構造を持たせている．階層的キーワード・リストの順序性を活かし，複数キーワードを並列に指定するだけよりも，検索の情報量を高めている．

複数ユーザのファイル共有では，ユーザごとに個別のファイル分類規則が，検索精度に悪影響を及ぼしかねない．そこで，各ユーザの名前空間の明示的な統合，同じ意味を持つキーワード同士の読み替え，他ユーザのファイルへのキーワード付与などを行う．

プロトタイプは，UNIX 上に実装され，通常のシェル上で実行可能である．

2.2 既存システムについて

(1)Samba

Samba は，ネットワークを通じて Windows マシンにファイル共有やプリンタ共有などのサービスを提供するための UNIX のソフトウェアである[3]．サーバ側の HDD にファイル共有のための領域を確保し，NAS としての機能を果たす．Access Control List をサポートする環境であれば，より詳細なアクセス権を設定することが可能となる．しかし，WAN を通してのアクセスが出来ない，ファイルが従来どおりの木構造で管理されるという問題がある．

(2)P2P 型ファイル共有

資源を一括管理するクライアント・サーバシステムではなく，クライアント同士が直接データをやり取りする Pear To Pear 方式によるファイル共有手法である．中央管理サーバが存在する中央サーバ型と存在しない純粋型が存在する．有名な純粋方 P2P 型ファイル共有ソフトとして Winny が挙げられる[4]．一括管理するサーバが存在せず負荷分散が実現され，また特定のサーバに依存しないため一部のクライアントマシンが停止しても共有環境を実現できる．

しかし，サービスに使用する情報がネットワーク内に分散しているため，データの一貫性を短期間で同期させるのが難しく，データの一元管理が困難であるという問題がある．また，各クライアントに共有したいファイルが分散している場合，完全な共有環境を実現するため，全てのクライアントマシンをネットワーク接続する必要がある．

(3)オンラインストレージ

オンラインストレージは，Web ブラウザを利用してファイルのアップロード・ダウンロードを行うファイル共有サービスである．最近では，MediaFire などのような様々なオンラインストレージサービスが登場している[5]．システムにログインする必要が無く，フォームにファイルを指定するだけで簡単にアップロードできる．また，インターネットに接続環境であればどこからでも利用可能で，HTTP を用いたダウンロードなので好みのダウンローダを利用できるという利点がある．

しかし，高機能なウェブサーバを通じてファイル転送するため負荷が集中しやすいなどの問題がある．

3．フォークソノミーの活用

3.1 フォークソノミーの現状

近年，SNS やウェブログなどのような Web サー

ビスが急速に普及している．これらの中には，フォークソノミーを効果的に活用している Web サービスも少なくない．フォークソノミーとは，folks（人々）と taxonomy（分類）の造語と言われ，キーワードなどから成るタグを用いて検索などに役立てるという概念である．

写真共有サイト「Flickr」や動画共有サイト「YouTube」などにおいては，個々のユーザが共有したいファイルにタグ情報を付与することにより，大量のファイルを分類している[6][7]．ウェブログ同士をつなげるウェブリング「はてなリング」では，含んでいるリングが多いタグを相対的に大きく表示するタグクラウドという機能を用いている[8]．

以上のことより，複数ユーザによるデータの共有や検索などにフォークソノミーの活用が適していると考え，ファイル管理システムへの適応を検討する．

3.2 タグ情報の構成

本システムでは，ファイル１つに付き，キーワード群を１つのタグ情報として構成し付与する．表１にタグ情報の構成を示す．

表 1 タグ情報の構成

名称	内容
ファイル ID	ファイル固有の ID
オーナーID	ファイル所有者の ID
登録日付	ファイル登録時の日付
ディレクトリキーワード	ディレクトリ名から抽出されたキーワード群
ファイルキーワード	ファイル名から抽出されたキーワード群
ユーザ指定キーワード	ユーザにより手動で指定されたキーワード群
拡張子	ファイル拡張子
ファイル名	ファイル登録時のオリジナルファイル名

3.3 キーワードの自動抽出

ファイルを登録するたびに，キーワードから成るタグ情報を手動で入力すると，オーバーヘッドが増大する．そこで，キーワードの自動抽出を行う．現時点では，ディレクトリ名とファイル名を対象として，キーワードを抽出している．

(1)ディレクトリ名からのキーワード抽出

現在，ユーザは木構造ファイルシステムによってファイルを管理している．多種多様なファイル群を複数の階層からなる木構造に分類し，ファイルを格納する．その際，各階層の分類を表すディレクトリ名は，ファイルの内容を端的に表している情報の１つと考えることが出来る．そこで，対象となるファイルに至るまでのディレクトリ名をキーワードとして明示的に活用することを考える．

しかし，木構造のルートから目的のファイルに至るまでの全てのディレクトリ名をキーワードとして用いると，ファイルの特徴を表すキーワードとして不適切な記述のディレクトリ名もある．これらについては，5 節において議論する．図１には，ディレクトリ名からキーワード抽出した結果を示す．

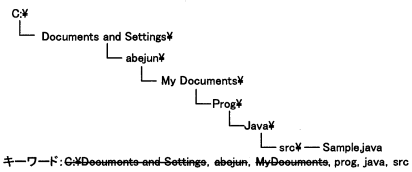

図 1 ディレクトリ名からのキーワード抽出

(2)ファイル名からのキーワード抽出

ユーザは，一般的に対象となるファイルの内容に沿ってファイル名を決定する．そこで，ファイル名からキーワードの抽出を試みる．

ファイル名からのキーワード抽出では，ファイル名に形態素解析を施す．その際，拡張子情報についてはファイル名から除去し，別途タグ情報に組み込む．表２に，「就職データベースの作成とユーザインタフェースの構成.ppt」というファイル名に対する形態素解析の例を示す．

表 2 ファイル名に対する形態素解析結果

形態素	品詞
就職	名詞-サ変接続
データベース	名詞-一般
の	助詞-連体化
作成	名詞-サ変接続
と	助詞-並立助詞
ユーザ	名詞-一般
インタフェース	名詞-一般
の	助詞-連体化
構成	名詞-サ変接続

形態素解析の結果から，名詞をキーワードとして採用し，抽出する．表２の例では，「就職」「データベース」「作成」「ユーザ」「インタフェース」「構成」がキーワードとして登録される．

4．システム設計

4.1 構成

(1)サーバ構成

サーバ側は，サーバ処理を管理するアプリケーションサーバ，ファイル転送，ファイル操作を行うSambaサーバ，タグ情報の管理を行うDBMSサーバから構成される．

(2)クライアント構成

ユーザには，図2のようなクライアントプログラムを配布する．ユーザは，クライアントプログラムを通して，ファイル登録，ファイル検索，ファイル操作などを行う．クライアントプログラムでは，ユーザに関する情報を属性として記憶する．本クライアントプログラムでは，Windowsのエクスプローラと同様のインタフェースを提供する，従来のようなファイル操作を行うだけで，ファイル登録などを自動的に実行できることが重要である．これにより，サーバ側のHDDに存在するファイル，クライアント側のHDDに存在するファイルなどとユーザに意識させないで利用できる，共有環境を実現する．

図 2 配布するクライアントプログラムイメージ

4.2 機能

(1)ファイル登録処理

ファイル登録処理は，図3のようなモジュール構成によって実現する．

クライアントプログラムの主な処理は，サーバへのファイル転送とファイル登録用タグ情報の生成である．ファイル転送は，Sambaの機能を用いることで実現する．タグ情報の生成では，ファイル名からのキーワード抽出のために，形態素解析エンジン「茶筌」を用いる[9]．サーバ側とのタグ情報の通信には，HTTPを用い，タグ情報はXML形式で送信される．タグ情報には，キーワード解析部で生成されたキーワード群のほかに，クライアントプログラムに記憶されているユーザ情報なども組み込まれる．

サーバ側では，ファイル格納のためのディレクトリを指定し，Sambaを通じてファイルを保存する．XML形式で送られてきたタグ情報はDBMSに保存される．

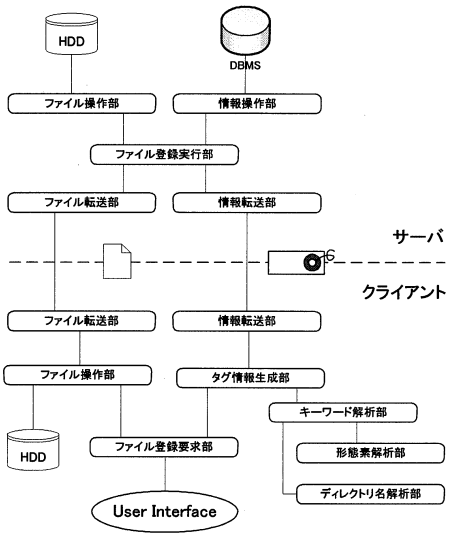

図 3 ファイル登録処理モジュール構成

(2)ファイル検索処理

ファイル検索処理は，図4のようなモジュール構成によって実現する．ユーザは，クライアントプログラム上で目的のファイルに関する情報をキーワードとして入力する．入力されたキーワードを含むファイル群のリストをDBMSから取得し，ユーザに提示する．これを対話的に繰り返し行うことで，ユーザは目的のファイルを見つけ出すことになる．

しかし，本システムは日常的に利用することを想定している．ユーザに検索フォームのみを提示するだけでは，1つのファイルに辿り着くまでに何度もキーワードを入力することになり，利便性に欠く．そこで，ユーザの入力したキーワード群のログを利用することを考える．ある検索キーワード群からファイルへのアクセスがあった場合，その検索キーワード群は目的のファイルを絞り込むための十分な情報を持っていると考えられる．このような検索キーワード群を自動的にユーザに提示することで，ユーザ側の入力負担の軽減を図る．

検索キーワードと完全に一致するものを提示するだけでは，複数ユーザ間での命名規則の揺れに対応できない．そこで，5.3節で述べる，キーワードに対する「曖昧化処理」を施すことで，他のユーザが命名したファイルへのアクセス効率を高めるよう工夫している．

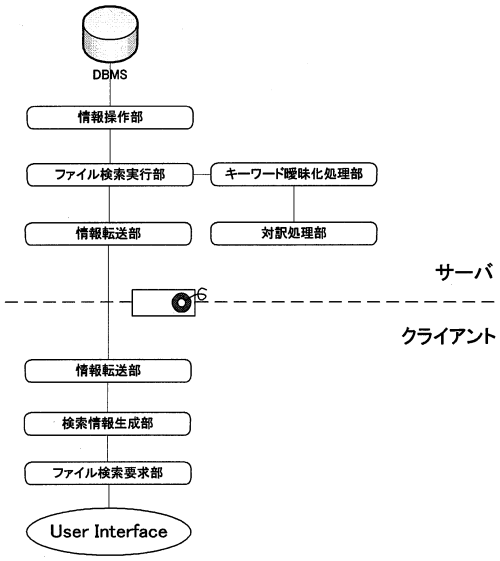

図 4 ファイル検索処理モジュール構成

5．キーワード分析

5.1 キーワード抽出

本システムでは、タグ情報を構成するキーワードを、ディレクトリ名およびファイル名から機械的に抽出している。その際，余分に生成されるキーワードを除去しておく必要がある．以下では，自動的にキーワードを抽出した際，具体的にどのようなキーワードが除去の対象になるか検討する．

今回は，研究室の学生が作業用として用いているWindowsマシン内に保存されている，研究に関係するファイル群を対象として調査し，キーワードの傾向を確認した．具体的にはRubyスクリプトにより，対象ファイル全てのフルパスおよびファイル名を取得し，キーワード自動抽出を行った．ファイル名からの自動抽出には，システム実装にも用いる「茶筌」を利用した．表3に抽出結果を示す．

表 3 抽出結果

ファイル数	3557
キーワード種類数	894
キーワード総数	28808

実際に得られたキーワードを種類ごとに抽出数の多い順に並べ，上位20位を抜粋したものが表4である．研究室でのキーワード抽出の傾向としては，個人名(研究継承のためディレクトリが作成)や拡張子などが上位にランキングされた．これは，その関係資料をまとめたディレクトリが存在し，そこに多数のファイルが存在することを意味する．

表 4 キーワード抽出数ランキング

キーワード	抽出数
semi	2047
Documents and Settings	1514
admin	1293
デスクトップ	1293
school	1182
卒業研究ネタ	1161
Thesis	783
H18年度シラバス	714
xls	681
00T	546
Presentation	484
修正後	384
txt	383
pdf	382
（新）専門	372
シラバス（学科別）	349
PDF	325
JPG	275
20051104_ACM	259
sadayuki	248

5.2 不要キーワードの分類

キーワード抽出結果から，不要だと思われるキーワード群を次の3種類に分類する．

(1)Windows特有のディレクトリ名

表5に示すキーワード群は，Windows特有のディレクトリ名であり，ファイル内容との意味的関連のない，不要なキーワード群と考えられる．

表 5 Windows特有のディレクトリ名

キーワード	抽出数
Documents and Settings	1514
デスクトップ	1293
My Documents	221

(2)アカウント名

表6に示すキーワード群は，Windowsにログインしているアカウント名である．ファイルの所有者を表すキーワードとして利用可能であるが，本システムではクライアントプログラムにユーザ情報が組み込まれているため，不要キーワードとみなす．

表 6 アカウント名

キーワード	抽出数

admin	1293
izuishi	221
s02t202	145

(3)その他

(1), (2)のほかに，キーワードとして不適切なものが存在する．それらについては，適時，手動で不要キーワードのリストを更新する必要がある．

表 7 その他

キーワード	抽出数
semi	2047

5.3 キーワード曖昧化

複数ユーザでファイルを共有する場合，ユーザごとの分類規則の揺れが生じる可能性があり，結果として検索効率が低下する．また，同一ユーザ内でも分類規則を間違え，適切な分類やファイル命名が行えていない場合などもある．そこで，自動的に抽出されたキーワードのほかに，ユーザが検索しやすいキーワードを用意する必要がある．

以下では，対訳辞書を用いた翻訳検索機能を検討する．ユーザごとに命名を日本語で行ったり，英語で行ったりする場合がある．5.1 節で行ったキーワード抽出においても，論文を表すキーワードとして「論文」や「Thesis」が用いられていた．このような場合への対策を考える．

本機能は，図４における「キーワード曖昧化」の処理を拡充する形式で，対訳処理を追加する．入力されたキーワードに対して英語，日本語の両方のキーワードで検索できるようにする．対訳辞書としてモナーシュ大学で作成された「EDICT」を用いる[10].

検索時に本機能が常に実行されると，必要としない場合でも，対訳キーワードを準備して検索を行う．これは，検索結果を無駄に増やす恐れがある．そこで，「キーワード曖昧化」処理についてはオプションとし，必要に応じて利用できる形態が望ましい．

6．おわりに

本研究では，複数ユーザによるファイル共有と，サーバ側・クライアント側を意識させない共有環境の実現を目指し，タグ情報に基づくファイル管理システムを提案し，その設計を行った．

自動抽出の際，キーワードのノイズとして拾ってしまう不要キーワードの除去について検討を行い，ユーザによる検索の利便性を高めるキーワード曖昧化処理として翻訳検索機能について述べた．

今後の課題として，実際の検索においてユーザの利便性を高めるために，検索ログを用いたキーワード提示，ユーザ間のキーワードの揺れなどを吸収するキーワード曖昧化処理について，効率的な実装を進める必要がある．さらに，登録されたキーワード同士の関係に着目し，検索時の補助に活用する手法についても検討していく．今後はシステム全体のプロトタイピングと，ユーザアンケートなどに基づくシステム評価，できれば定量的評価を試みたい．

参考文献

[1] David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. O'Toole, Jr., "Semantic File System", 13th ACM Symposium on Operating Systems Principles, October 1991.

[2] 轟木伸俊, 多田知正, 樋口昌宏, 谷口健一, "階層的キーワードに基づく名前管理手法とそれに基づくファイル共有手法", 情報処理学会 研究報告 GN, Vol.2000, No.97, 2000.

[3] 日本 Samba ユーザー会, “日本 Samba ユーザー会”, http://www.samba.gr.jp/, (2006 年 12 月アクセス).

[4] 金子勇, “Winny の技術”, 株式会社アスキー, (2005).

[5] MediaFire., “Media Fire”, http://www.mediafire.com/, (2006 年 12 月アクセス).

[6] Yahoo! Inc., ”Flickr”, http://www.flickr.com/, (2006 年 12 月アクセス).

[7] YouTube, Inc, ”YouTube - Braodcast Yourself.”, http://www.youtube.com/, (2006 年 12 月アクセス).

[8] 株式会社 はてな, “はてなリング”, http://ring.hatena.ne.jp/, (2006 年 12 月アクセス).

[9] NAIST Computational Linguistics Lab., “ChaSen’s WIki”, http://chasen.naist.jp/hiki/Cha Sen/, (2006 年 12 月アクセス).

[10] Jim Breen, “THE EDICT Dictionary File”, http://www.csse.monash.edu.au/~jwb/edict.html, (2006 年 12 月アクセス).