

タグツリーによる個人コンテンツ管理

Personal Contents management by Tag tree

内藤一兵衛 赤間浩樹 谷口展郎 山室雅司*

Summary. デジタル化されたコンテンツの普及と HDD の大容量化により超大量のコンテンツが蓄積可能となっている。しかし、ユーザが再利用しようとして貯蓄したコンテンツの多くは十分に再利用されていない。また、新たにコンテンツを蓄積する際には過去に関連のあるコンテンツを蓄積した覚えがあるがそれらと関連付けが行えないという状況が発生する。本論文ではこれらの溜め込んだ超大量のコンテンツに対し、タグによるコンテンツ管理を拡張し、タグツリーを導入することによって、ユーザ自身が過去に探した過程や整理した過程を再利用することを提案する。その上で、本提案をファイルシステムにマッピングした階層構造の操作を定義し、その効果について述べる。

1 タグによるコンテンツ管理の課題

Flicker やはてなブックマーク等、Web 上でコンテンツを整理する方法として、タグ付けとタグクラウドからの動的な絞込みによるコンテンツ管理がある。タグはユーザが登録するキーワードであり、ユーザが自由にタグ付けを行なえる点でユーザの記憶と結びつきが強く、コンテンツを検索する際に有効である。一方、自由なタグを付与できる環境はタグを増加させてしまう欠点がある。タグが増加すると、欲しいコンテンツにアクセスするためのタグが思い出せず、かつタグクラウドのような全てのタグを見渡すインタフェースのみからは見つからないという状況が発生する。検索の観点からはタグの増加を抑える必要があり、タグ付与の観点からは過去に作成したタグを有効に使う為の機能が必要である。また、既存のタグによるコンテンツ管理システムのインタフェースでは各コンテンツにタグを一つ一つ付与していく必要がある。一つのコンテンツに複数のタグを同時に付与することや、複数のコンテンツにまとめてタグを付与することは効率的にはできない。

2 タグツリーによるコンテンツ管理の提案

タギングシステムのタグ管理にツリー構造を導入し、そのツリー構造を既存のファイルシステムにマッピングすることで、前述したコンテンツ管理の課題を解決するアプローチを提案する。

従来のタグ管理は図 1 内のコンテンツ空間とタグ空間より構成されていた。我々が提案するタグツリーは各ノードが登録済みのタグにより構成され、各ノードとタグは N 対 1 の関係となる。タグツリーの親子関係の順序に意味はなく、ユーザが自由に作成できる。そして、このタグツリーがコンテンツの

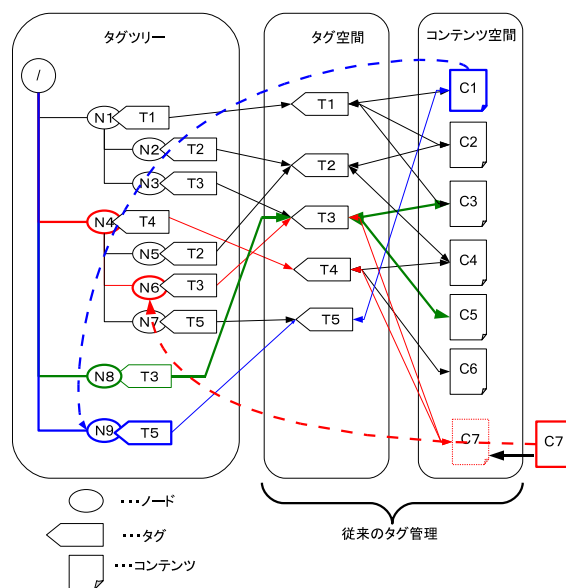


図 1. 提案アーキテクチャと操作例

フィルタリングとタギングの両面で使う方式となっている。

2.1 タグツリーを使ったフィルタリング

タグツリーのルート (root) からのパス (タグ群) がフィルタリングにおける条件式となる。例えば、図 1 に示すタグツリーにおいて、ノード N_1 タグ T_1 を選択すると、フィルタリングの結果としてコンテンツ C_1, C_2, C_3 が得られる。

ツリーで管理されたタグの上下関係はフィルタリング時には AND 関係に置換される。ノード N_2 タグ T_2 を選択すると、タグ T_1 and T_2 でフィルタリングされ、コンテンツ C_2 が得られる。

既存のタグを新たなノードとしてタグツリーに登録すると、その配下にはそのタグによってフィルタ

Copyright is held by the author(s).

* NAITO Ichibe, 日本電信電話株式会社 NTT サイバースペース研究所

表 1. タグツリー操作の定義と従来ファイルシステムとの比較

対象	操作	タグツリー上での階層構造操作	従来ファイルシステム
ファイル	作成	ファイル名をもとにルートに書き出す	与えられたパスにファイルを書き出す
ファイル	削除	ルートではファイルは消せない 選択されたフォルダ内ではその上の階層タグの一つを削除する	ファイルの実体を削除する
ファイル	移動	指定されたパスのタグを付与する	ファイルを指定されたパスに移動
ファイル	コピー	指定されたパスのタグを付与する	ファイルを指定されたパスに実体ごとコピー
フォルダ	作成	タグを追加する 自動的に指定したパスを含むファイルは作成したフォルダ内に見える	フォルダを作成する
フォルダ	削除	フォルダが表示から消える．フォルダ内にあるファイルとそのタグに影響はない	フォルダに属する全てのファイルを削除する
フォルダ	移動	フォルダ階層のみが移動先へコピーされる	フォルダに属するファイルごと指定されたパスに移動する
フォルダ	コピー	フォルダ階層のみが移動先へコピーされる	フォルダに属するファイルごと指定されたパスに新しい実体を作る

リングされた結果を得ることができる．既存のタグ $T3$ を持つノード $N8$ を作成すると、 $T3$ タグを持つ $C3, C5$ のコンテンツがフィルタリングの結果として得られ、ノード $N8$ のフォルダの中に入る．

2.2 タグツリーを使ったタギング

フィルタとして使ったタグツリーを、タギングする為のインタフェースとしても利用する．ユーザがコンテンツをフォルダに投入することでフォルダ名がタグとして付与される．例えば、新たなコンテンツ $C7$ をノード $N6$ に投入すると $root$ までのノード集合 $N4, N6$ を取得し、それらに対応付けられたタグ $T4, T3$ がコンテンツ $C7$ に付与される．

また、既存のコンテンツに対してタグを追加することも同様に可能である．既存コンテンツ $C1$ がタグ $T1$ のみが付与されている状況で、コンテンツ $C1$ をノード $N9$ タグ $T5$ に投入するとコンテンツ $C1$ に付与されたタグは $T1, T5$ になる．

2.3 タグツリー上の操作

タグツリー操作の定義と従来ファイルシステムとの比較を表 2 にまとめた．操作はタグとファイルの関係管理のみに影響し、操作によりファイルの実体が消されたり、ファイルのコピーが作成されたりするような事はなく、ユーザはタグとコンテンツ間の整理にフォーカスできる．

また、タグツリーの操作ではファイルのコピー・移動の概念はなくタグの付与のみである．あるパスから他のパスへのコンテンツ移動の操作が移動、コピー、リンクの 3 つの意味がある従来ファイルシステムの操作に比べ、一つの操作にまとまっている．

2.4 実装

タグツリーをファイルシステムのフォルダ階層にマッピングし、FUSE[1] を使い Linux のファイルシステムとして実装した．

3 考察と今後の課題

本方式は、同じタグを付与するコンテンツを大量に蓄積しようとしたとき、一度タグツリーを作成すると、効率的にタグを付与できるというメリットがある．一方、新規タグを作成するのにフォルダの作成が必要であり、従来の思いついたときに書き込むインタフェースより手間が大きい．また、逐次新規タグをツリー構造として残すとツリーが巨大化し、不要なノードの削除が必要になるというデメリットがある．

従来、検索結果を予め登録しておき、仮想的なフォルダに見せるインタフェースは存在した．検索式の対象がコンテンツ実体である場合、検索フォルダにファイルを投入することはコンテンツ実体の変更を意味するため、検索フォルダにファイルを投入することはできなかった．本提案ではタグの階層管理とタグ・コンテンツ間の管理を分けることで、自動的なタギングがあった場合、コンテンツ実体に変更無く、動的に中身が変わる可能性のあるフォルダにファイルを投入することができる．特に入力するファイルが多いとき自動タギングルール作成とその修正は逐次入力するファイルの分類を行なうより簡単である．

今後は、長期的な利用者による評価を行っていきたい．

参考文献

- [1] Filesystem in Userspace.
<http://fuse.sourceforge.net/>
- [2] D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. W. O. Jr, Semantic file systems, Proceedings of 13th ACM Symposium on Operating Systems Principles. pp. 16-25. Association for Computing Machinery SIGOPS, 1991,
- [3] Edward Cutrell, Daniel C. Robbins, Susan T. Dumais, Raman Sarin . Fast, Flexible Filtering with Phlat, Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 261-270. ACM Press, 2006.