

コンテンツ埋込 DB と木状ビューの対応を使った メタデータ管理インタフェースの設計と実装

赤間 浩樹[†] 三井 一能[†] 串間 和彦[†]

近年、デジタル機器やインターネットの普及によってコンテンツの流通がさかに行われるようになった。その結果、多様化したコンテンツの流通の各過程に適用可能なコンテンツの効率的な管理法が求められている。本論文で我々は、コンテンツのコメント域や拡張域に DB を構築し、そのコンテンツ埋込 DB に格納したメタデータを木状ビュー情報と関連付けて管理する仕組みを提案する。この手法は、コンテンツ埋込 DB 内のメタデータに対する効果的な維持管理インタフェースとなるとともに、コンテンツの流通の各過程においても効率的なコンテンツの概観および管理手段となる。本論文ではその設計概要と実装の詳細を示し、また、大量画像および URL の管理に適用した例を示す。

Design and Implementation of Metadata Management Interface Using Correlation of Content-embedded Database and Tree View

HIROKI AKAMA,[†] KAZUYOSHI MII[†] and KAZUHIKO KUSHIMA[†]

Recently, as a result of prevailing digital devices and the internet, streams of contents are increasing. Therefore effective contents management techniques are required on each site on the streams. In this paper, we propose two methods. First proposition is to construct database in each content using comment area or extension area. Second proposition is to manage metadata of contents using correlation of content-embedded database and tree view interface. We applied them to images and bookmarks management.

1. はじめに

コンテンツのデジタル化が進行し、管理対象となるコンテンツが急激に増加している。また、インターネットによるコンテンツの流通も管理対象の増加に拍車をかけている。それらに対応して、コンテンツの管理方法についても各種の提案がなされている。

1.1 サーバにおけるコンテンツの管理

(1) キーワードや属性情報によるコンテンツ管理
コンテンツにキーワードや解説文などの付加情報を付与することによって管理する方法はすでに広く行われている。本論文では、これらコンテンツの内容や主題を表現するキーワード、コンテンツの日付や著者などの属性情報、コンテンツの表層的な特徴を表現する特徴量情報をまとめてメタデータと称することとする。このようにコンテンツをキーワードで管理することは一般的にはその付与に人手を介することになるため、コストが高く時間がかかるといった問題、および、複

数の付与と作業の間で一貫性の保持が難しいという問題を発生させる。我々は、これらの問題に対処するため、次項で述べる特徴量に基づくアプローチ^{1)~6)}を検討してきた。また、そのほかにも日付時刻、GPS、焦点距離などの機器による自動付与の情報、Web ページ中の画像の周辺テキストや ALT 属性から推定して自動付与された情報などの自動付与信息を利用する方法や、スポーツなどのライブ映像に音声認識やテンプレート情報の GUI を使ってリアルタイムに付与する方法⁷⁾や、ニュースやドラマのシナリオ情報をメタデータとして利用する方法⁸⁾など、各種の方法が幅広く提案されている。

(2) 特徴量によるコンテンツ管理

これまで我々は、画像目録閲覧と画像類似検索を連携した画像管理システム ImageCompass を提案⁶⁾してきた。我々の提案した画像検索システム^{1)~4)}では、まず特徴量による画像の類似検索(図1, 図2)に取り組んだ。これは、従来の画像全体に対する類似検索の精度の悪さを、画像内オブジェクト候補を階層的に自動抽出する技術(図3)を使って、画像内オブジェクト検索を可能にし、精度の向上を実現した。しかし、画

[†] NTT サイバースペース研究所
NTT Cyber Space Laboratories

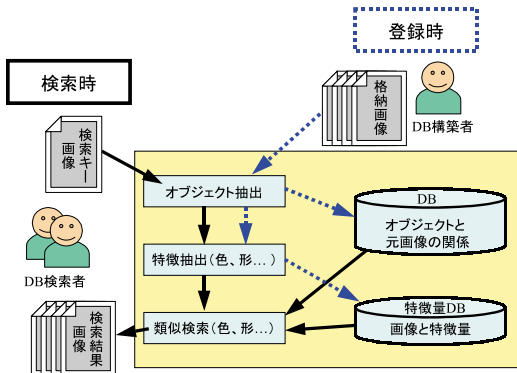


図1 類似画像検索システムの処理の流れ

Fig. 1 Flow of similarity-based image search system.

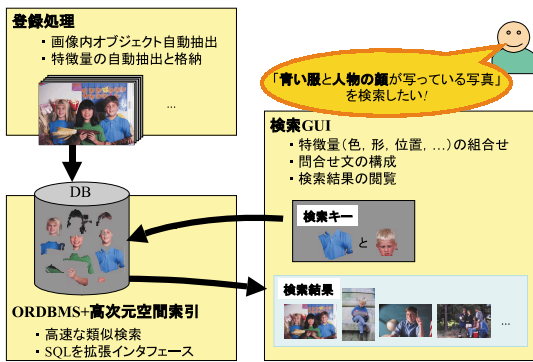


図2 類似画像検索の例

Fig. 2 Example of similarity-based image search system.

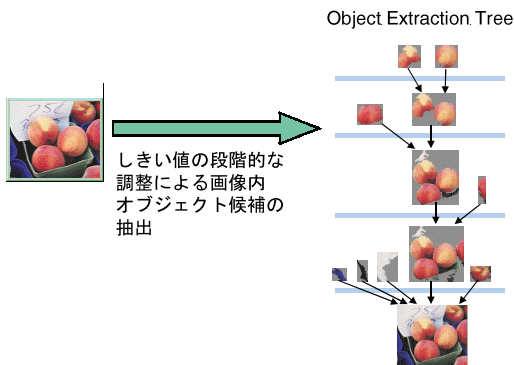


図3 画像内オブジェクトの抽出例

Fig. 3 Example of image object extraction.

像の管理システムとしては、まだ大きな問題が残った。

課題1. キー入力の効率が低い。

課題2. データベース(DB)全体の概観が把握できない。

類似画像検索では一般に画像を検索キーとして与えなければならない。しかし、スケッチ入力による画像描画では検索キーを与えにくく、効率良く類似のサン

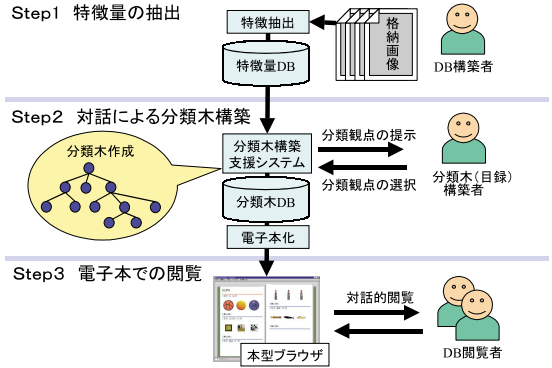


図4 画像分類木作成支援・目録閲覧システムの処理の流れ

Fig. 4 Flow of image classification-tree authoring system and image catalog browsing system.

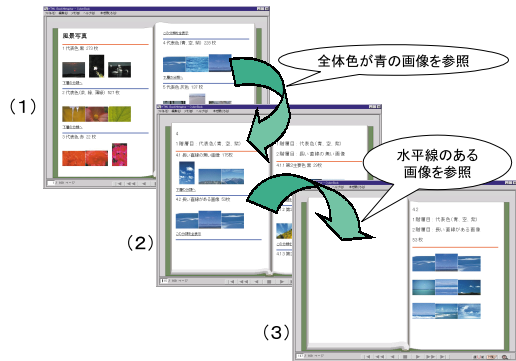


図5 画像分類木(目録)の閲覧例

Fig. 5 Example of browsing image catalog.

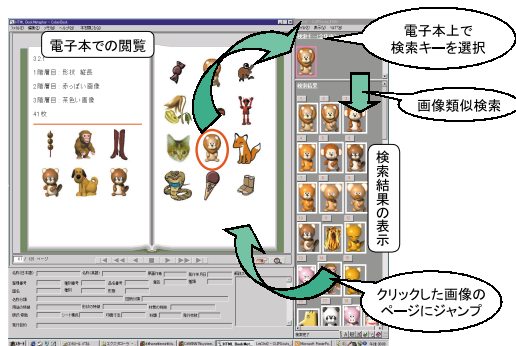


図6 目録閲覧と画像類似検索の連携例

Fig. 6 Example of integrating browsing image catalog and searching similar image.

ブル画像を与える方法が課題1として求められた。また、類似検索は相対的な比較に基づく検索であるため、検索結果が悪い場合に、それがそもそも検索キーとデータベースとの内容の不一致によるものなのか、それとも類似の観点として選んだ特徴量やその重みが原因なのかが判断できない。そこで、DBの概観によ

る課題 2 の解決が求められた。

我々は課題 1 および 2 を解決するため、画像の特微量による階層的な分類を使った分類木構築支援システムおよび目録閲覧システム⁵⁾の実現に取り組んだ(図 4)。なお、ここで分類木と目録は同一のものを指す。分類木構築支援システムでは、画像の各種の特微量を分類観点として分類木構築者に提示し、分類木構築者がその中の 1 つを選択することによって、システムは対象画像群の分類を行い、その分類を分類木情報として蓄積する。この作業を階層的に行うことによって、画像集合に対する分類木つまり画像目録が構築される。目録閲覧システム(図 5)は、その目録を電子本で閲覧するシステムである。この目録によって階層的に画像 DB 全体の傾向を把握できるようになる。さらに、検索システムと目録閲覧システムを、目録中の任意の画像から、類似検索を行って検索結果を得ること、また、その検索結果の特定の画像が含まれる目録中のページにジャンプできるように統合⁶⁾を行った(図 6)。

(3) キーワードと特微量を併用したコンテンツ管理
キーワードの利用と特微量情報の利用は、排他的なものではなく、相互に連携するとより効果的である。波多野⁹⁾は類似する特微量の画像に同一のキーワードを一括付与するなどして、効率的なキーワードの付与と一貫性管理を実現している。特にこのようなアプローチは模様画像のような、なかなか作業間でキーワードを一致させるのが難しい対象に対して、大変効果があると考えられる。また、我々のシステムでは、目録の分類観点情報にキーワードを併用することや、辞書を經由してキーワードを画像特微量に変換することを可能にしている。

1.2 流過程におけるコンテンツの管理

次に、コンテンツの流過程の変化について考える。これまでは、コンテンツがプロバイダや流通業者のサーバに蓄積され、そのサーバ上で効率的な管理、つまりメタデータに関する保守を行う集中管理モデル(図 7)が主であった。しかし、インターネットによって、コンテンツの流通が促進されたことによって、コンテンツの管理が必要な局面は、コンテンツ・ライフサイクルの全過程(図 8)にまで広がってきた。つまり、コンテンツの作者は自分の作品および素材の管理が必要となり、コンテンツのプロバイダは多数の作者から集めた作品の管理が必要であり、ユーザも自分の所有する作品の管理が必要になる。さらに、インターネットは、作者、プロバイダ、ユーザといった境界も無意味にしつつあり、作者がユーザであったり、プロバイダであることを容易に実現する。これらのことは、

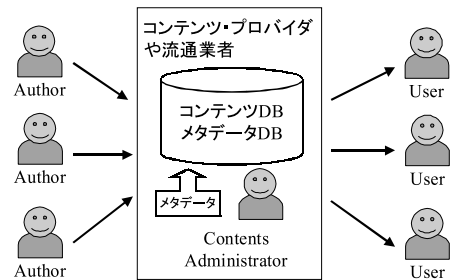
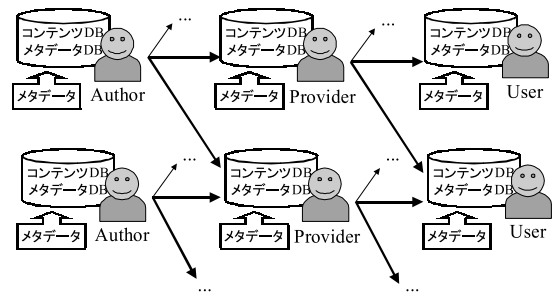


図 7 メタデータ DB の集中管理モデル

Fig. 7 Centralized management model of metadata DB.



(注) Author, Provider, User の役割が兼務され混在する。

図 8 コンテンツのライフサイクル上におけるメタデータ DB の遍在管理モデル

Fig. 8 Ubiquitous management model of metadata DB on contents life cycle.

次の課題を示している。

課題 3. 流通の全過程(コンテンツのライフサイクル)に対応できるコンテンツ管理の仕組みが必要である。たとえばコンテンツに対応したメタデータもコンテンツと同時に、かつ、効率良く流通の次の過程に流れる必要がある。また、新たに入手・発生したコンテンツと既存のコンテンツの情報は容易に統合できる必要がある。

課題 4. 各流過程において一時的または永続的に蓄積されたコンテンツ集合およびそれに対応するメタデータの概要を効率良く把握でき、かつ、効率良く管理(追加, 削除, 修正, さらに、一貫性を維持)できる必要がある。たとえば、流通するコンテンツの持つメタデータの詳細度を効率的に管理できなければならない。

本論文の狙いは上記の課題を解決するアプローチを提案することである。具体的には、まず、コンテンツ内に DB 域を構成し、そこでコンテンツのメタデータを管理することで、コンテンツの柔軟な流通への対応を可能にする。コンテンツ内の DB 域のアクセス・インタフェースとして、視覚化に優れた木状ビューを用意し、それによって、ダイナミックに変化するコンテ

コンテンツ集合の概観把握、および、それらのメタデータの概観把握、効率的な管理（追加、削除、修正）を実現する。以下、2章では、提案の骨子と周辺技術を紹介し、3章では、その設計と実装の詳細について議論し、4章で、適用事例を考察とともに紹介し、5章でまとめる。

2. コンテンツ埋込 DB とメタデータの管理インタフェース

本論文で提案する技術は、従来の2つの技術を組み合わせることで、新たな流通形態におけるコンテンツ管理に対応するものである。その1つはコンテンツ埋込情報で、もう1つは木状ビューインタフェースである。以下、各々の技術の解説と、それに対応する提案を示す。

2.1 コンテンツ埋込情報からコンテンツ埋込 DB へ

(1) 現状

最近のメディアフォーマットは、その中にコメント域を有したり、特定のメタデータを持てる構造をあらかじめ有する。最も身近な例であるデジタルカメラの規格 Exif⁽¹⁰⁾、DCF⁽¹¹⁾では、JPEG フォーマットの拡張フィールドを利用することで撮影日時、焦点距離、などの情報フィールドを有している。また、DIG35⁽¹²⁾では、それらの情報を XML で記述する試みを行っている。さらに、JPEG 画像に限ればこれらの情報を管理する PDS (Public Domain Software) などにもすでにいくつか存在している。このコンテンツ埋込みメタデータは、コンテンツと同一ファイル内に存在するため、コンテンツの流通（移動や複写）によって分離されることがなく、また、コンテンツのプレイヤー（たとえば画像ならビューア）として汎用のものが利用できるというメリットがある。

(2) 提案

ここで、我々は、メタデータを単に個々のメタデータ専用領域の値として個別に持つのではなく、コンテンツの拡張域やコメント域を利用して DB 域として汎用に構成し、これに合わせ汎用的な問合せ言語を提供することをまず提案する。我々はこれをコンテンツ埋込 DB と呼ぶ。このコンテンツ埋込 DB は、ネットワーク上を移動する DB であるため、DB のスケールの違いはあるものの原らの提案する概念⁽¹³⁾にも通じる点がある。しかし、我々は原らと異なり、画像などの具体的なコンテンツとの結び付きをより重視する点、および、そのコンテンツのメタデータの管理に利用する点が異なる。

コンテンツ埋込 DB は、コンテンツ埋込情報の性質を引き継ぐため、コンテンツとの一体化が可能であり、特殊なプレイヤーを必要としないというメリットも残る。これにより先に示した課題3を解決する。

本提案、つまり、DB としたことに対する一般的な疑問として、「単なるメタデータを管理するのになぜ DB などという大げさな仕組みが必要なのか？」また、「コンテンツのメタデータは1コンテンツに対して1レコードに相当する情報であって、それらは DB と呼ばれるほどの規模の情報ではないのではないのか？」などがある。これに対する我々の回答は2つある。

1つ目の回答は、1つのコンテンツには複数の同質な情報が多数含まれることが容易に想定できるということである。たとえば、画像内に写る被写体は多数存在する可能性がある。それらの各々の被写体情報を個別に管理する場合には、被写体ごとにメタデータが管理できなければならない。また、画像内の多数のオブジェクトに対する色相、彩度、輝度、形状、大きさ、位置、模様などの特徴量を画像自身が内包する場合などもデータ量が増える。また、コンテンツの更新履歴などもコンテンツと一体化して管理すれば同様に DB 化が有効になる。

2つ目の回答は、たとえ要素となるコンテンツに含まれる個々のメタデータは少なくとも、複合コンテンツになれば、それらを DB として扱うことがより自然になるということである。たとえば、ドキュメントに貼り付けられた個々のコンテンツ内のメタデータに対して問合せを行ったり、映像中の個々の画像内のメタデータについて問合せを行ったりする局面が想定できる。なお、この場合は、複数の DB に対して問合せを行って結果を統合する形態となるので、マルチ DB の一種と考えてもよい。

2.2 木状ビューによるコンテンツ埋込 DB 情報の管理

(1) 現状

Microsoft 社の Windows の Explorer に代表される木状ビューはきわめて一般的なインタフェースとなった。PDS などでもファイルのディレクトリ構成を木状ビューで表現し、そのノード内の画像ファイルを一覧できるソフトウェアが多数公開され、また市販されている。また、岩崎らは画像の木状ビューを使った画像一覧システムを提案⁽¹⁴⁾している。

(2) 提案

我々は、コンテンツ埋込 DB の管理インタフェースとしてこの木状ビューを利用し、コンテンツ埋込 DB をメタデータの管理に利用することを提案する。この

方法は従来のサーバを中心としたコンテンツの管理アプローチに比べ、きわめて簡便にメタデータを管理する仕組みが適用でき、また同時にコンテンツの概観の支援にもなる。これにより、先に示した課題 3, 4 を解決する。

ファイルの Explorer と我々のメタデータに対する木状ビューとの主な差異を以下に示す。

木状ビューは論理的なメタデータの体系を表す。よって、同一実態のコンテンツが複数のノードに現れるのが一般的状態となる。また、削除に対する考え方が論理的となり、実態が削除されるのではなく、より上位概念(上位フォルダ)に移動するようになる。

コンテンツ埋込 DB のメタデータと木状ビューのノード上のコンテンツ位置は動的に対応する。よって、木状ビュー上でのコンテンツのフォルダ位置の移動、複写操作や、フォルダ名の変更がコンテンツ埋込 DB 内のメタデータの変更に動的に反映される。

コンテンツの目録閲覧および類似検索のブラウザとして効果的に機能する。木状ビューの形状によりコンテンツ群の全体傾向の把握が可能になる。また、従来の我々の目録閲覧・類似検索システムと同様にキーワード検索や類似検索と目録閲覧が統合されている。さらに木状ビューの背景部分に概観情報を埋め込むことによる概観把握支援も可能である。

3. 設計と実装

前章での提案をもとに我々は画像を対象とした実システムを設計した。

我々のシステム ImageCompass/TagDir の機能構成を図 9 に示す。本システムでは 4 種類のプラグインを利用する構造として、各種のメディアへの対応と柔軟性を実現する。まず、画像表示用プラグインとしては、実績のある Susie¹⁵⁾ プラグイン・インタフェースの上位互換とする。コンテンツ埋込 DB アクセス用インタフェースとしては今回は SQL をベースとする問合せ言語を採用する。問合せ言語に関する詳細は 3.2 節で述べる。検索条件制御用プラグインについては、木状ビュー上の検索インタフェースの柔軟性確保のために採用する。詳細は 3.4 節で述べる。また、コンテンツ対応の自動特徴量抽出および自動フォルダ配置のために特徴抽出プラグインを利用する。

3.1 画像への DB の埋込み方法

ここでは、本論文の内容再現のための参考情報とし

て、現在最も普及している画像形式の JPEG (JFIF) を対象とした埋込 DB の具体的な構成法について解説する。

(1) JPEG フォーマットと拡張性の利用

JPEG (Joint Photograph Experts Group) の圧縮規格は、TIFF, Exif¹⁰⁾, CIFF, DCF¹¹⁾, FlashPix など多数の画像フォーマットに採用されているが、一般に JPEG と呼ばれる多くの画像はその交換フォーマットの拡張域 APP0 を使って実現された JFIF (JPEG File Interchange Format) である。今回、我々はこの JFIF を対象にコンテンツ埋込 DB を設計した。なお、コンテンツ埋込 DB の実装および構成は問合せ言語インタフェースによって隠蔽可能であるため、その実現方法はフォーマットやメディアごとに異なっても問題ない。

JPEG 交換フォーマットは 2 バイトのマーカによってブロックが構成される。まず、フォーマットは SOI (Start of Image) マーカ (FFh D8h) で始まり、EOI (End of Image) マーカ (FFh D9h) で終わる。このフォーマットにはいくつかの埋込 DB 構成法がある。1 つは、APP0 ~ APP15 まで 16 個用意された拡張域を利用することである。ただし、JFIF がすでに APP0 を、Exif が APP1 を、FlashPix が APP2 を、といったように、すでに多数の拡張フォーマットで利用されており、それらとの衝突を避けるため今回は利用を見合わせた。2 つ目は SOI の前、または、EOI の後に強制的に付加することである。この方法はかなり現実的な方法であり、拡張が規定されないフォーマットの場合などには有効な方法である。ただし、規格に適合しているとはいえない。3 つ目の方法はコメントを示す COM マーカ (FFh FEh) を利用する方法である。今回は、既存の JPEG 中のコメントも DB のデータ化することを狙って、3 番目の方式を選択した。なお、4 章で述べる URL への適用時には 2 つ目の方法を採用している。

COM マーカはマーカに続きブロック長があり、その後コメント用データが続く。長さは他のマーカの障害にならない範囲で有効であり、1 ブロックで 64 Kbyte は超えられないが、COM ブロックは複数存在してもかまわない。

DB の詳細は後述するが、行単位にデータが構成される。先頭行には識別子とフォーマット版情報からなる InfoBase マーカ行を挿入している。

第 1 フィールド：“iNFObASE-format-version”

第 2 フィールド：“2000&02&17”

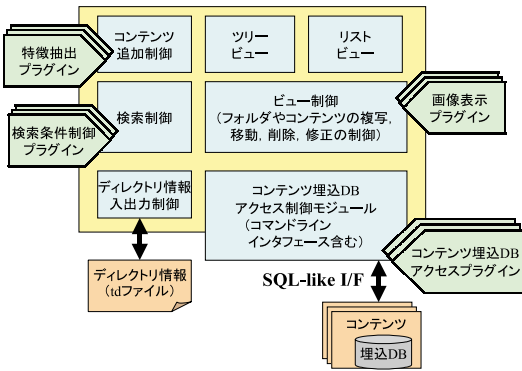


図 9 本システムの機能構成
Fig. 9 Architecture of our system.

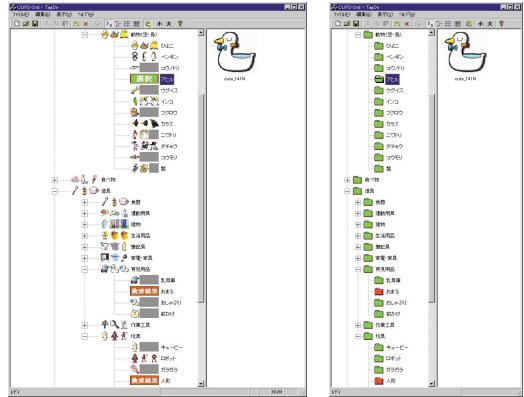


図 12 特定画像の存在フォルダの検索例
Fig. 12 Search result of folder existing key image.

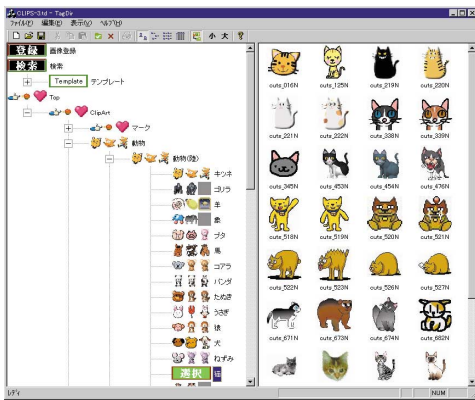


図 10 リストビューによる画像一覧の例
Fig. 10 Example of image list view.

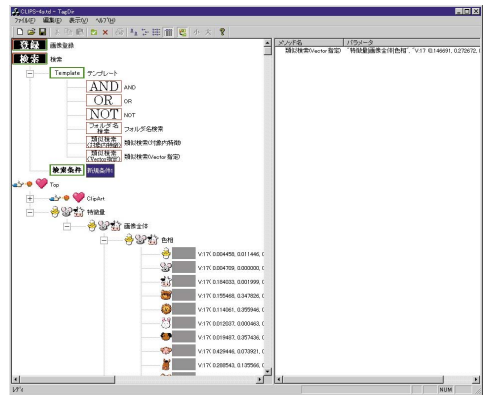


図 13 類似検索の検索条件指定例
Fig. 13 Example of search condition setting.

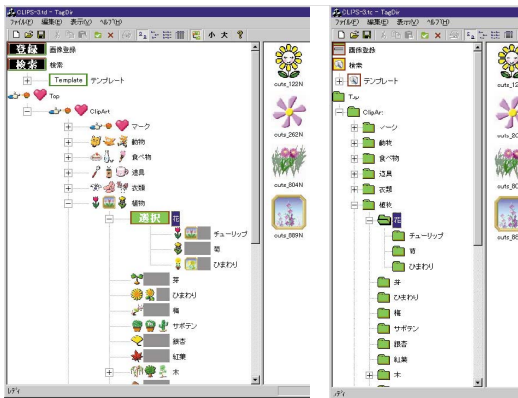


図 11 フォルダの表示切替えの例
Fig. 11 Example of folder browsing mode.

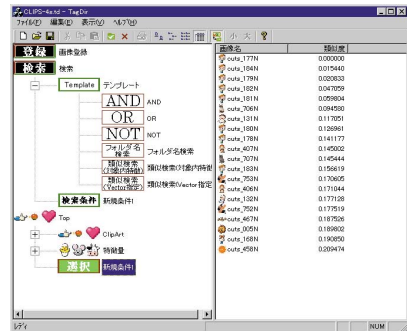


図 14 類似検索の結果表示例
Fig. 14 Result of similarity-based image search.

第 3 フィールド：文字コード種別 (省略可)
第 4 フィールド：(予約)

第 1 フィールドは、誤った変更を防ぐためにマーカーの大文字と小文字を反転して格納する。第 2 フィールド

ドは、将来のフォーマット拡張のために現フォーマットの規定日情報を格納する。さらに、第 1 フィールドと同様に誤変更を防ぐため '&' を区切り文字として使用する。

また、先に述べたようにすでに COM ブロックが存在した場合には、第 1 フィールドに以下の識別子を入

れ、コメントを第2フィールドにそのまま複写格納した行をDB内に構成する。

第1フィールド：“OriginalComment”

第2フィールド：オリジナルのコメント情報

このとき、既存のCOMブロックは破壊せずそのままの状態を保存する。

(2) DBのモデルと構成

本論文が想定する個々のコンテンツが所有するDBとは、RDBやOODBのようなきわめて柔軟なDBとは少し異なり、メタデータの管理に特化したDBである。我々の採用するDBは、その主な目的がメタデータの管理であり、木状ビューでの管理との親和性を向上させるため、下記のようなディレクトリパスの集合を基本モデルとした。

DBは前記マーカ行を含み1行以上の行からなる。行は1列以上の列からなる。各行の列数は異なってもかまわない。各行は、ディレクトリパスに対応する。列名はなくExcelのように数字で指定する。通常のRDBと同様に、行の並びの順序に意味はなく、必要であれば問合せ言語によって順序付けを行う。

データ型は文字列型を基本とする。その実際の型への解釈はアプリケーションや問合せ言語に任せる。たとえば、3次元ベクトル“V:3(1.732, 2.761, 3.14)”は問合せ言語では3つのfloatからなるベクトルの形式として解釈するが、DB内では文字列として管理する。また、各フィールドの文字列中には改行文字を含むことができる。

今回実装した具体的構成においては、JPEGのマーカと矛盾しない構成にするため、UNIXコマンドのsedで利用されている形式をベースにした形式を採用した。このsedで利用されている形式とは、行頭に区切り文字を定義し、その区切り文字によってフィールドの分離を認識する形式である。たとえば、以下の3行はフィールド中に区切り文字を含まなければ同じ意味となる。

```
:フィールド 1:フィールド 2:フィールド 3:
?フィールド 1?フィールド 2?フィールド 3?
|フィールド 1|フィールド 2|フィールド 3|
```

区切り文字としてフィールド中に出現しない文字コードをユーザが選択し利用しなければならない。さらに、改行文字への対処のため、以下に示すようなsedとの若干の機能差をつける。

入力時の改行は“¥n”で表現し、入力データ中の改行文字はLF(‘¥n’, 0Ah)に変換して格納する。出力時にはプラットフォーム(OS)に合わせて出力する。

DB格納時の行の区切りは、区切り文字と復帰文字CR(‘¥r’, 0Dh)で表現する。

本方式はアクセス処理プログラム内部で文字コードによる行区切り検出結果のリストを保持するように実装することによって、フィールド中にはシフトJIS漢字などの2バイト目が区切り文字と衝突する可能性がある文字でも利用できる。

たとえば、E:¥a02.jpgには、以下のような拡張sed形式のメタデータを持たせることが可能である。

```
:切手:自然:風景:山:
?切手?シリーズ切手?国立公園?
?作成日?2000?09?20?
:著作権者:鈴木一郎:田中次郎:
?特微量?画像全体?色相?V: 4( 0.0, 0.0, 0.7, 0.3)?
```

本構成のDBに対する更新関連の処理は、性能よりコンテンツとしての領域使用の無駄の少なさを優先し、きわめて原始的な処理とする。行の挿入時は最後部にデータを追加し、行を削除する場合には、その領域を詰めるようにデータのシフトを行う。フィールド値の更新時には、更新前後が同じ長さであればその場所での置換を行うが、長さに差異があれば、削除後に追加を行う。

3.2 コンテンツ埋込DBへの問合せ言語

問合せ言語の基本構文としてはSQLをベースとした独自拡張言語も作成した。可変フィールド数の表構成および検索対象の汎用化に対応するため以下のような拡張を行った。

(1) 可変フィールド数対応

SELECT句およびWHERE句で列を指定する場合には、基本的に番号で指定する必要がある。そこで、column()メソッドを追加する。

(2) 検索対象ファイルの汎用化

ファイルおよびファイル群に対する検索を指定する方法として、FROM句にはファイルパスおよびファイルに対するワイルドカード(‘*’)の利用を可能にする。これらを使った問合せの例を示す。

```
SELECT column(4)
FROM ‘C:¥Windows¥Favorites¥*.jpg’
WHERE column(1) = ‘特微量’
and column(2) = ‘画像全体’
```

and column(3) = '色相'

(3) 拡張 sed 形式の利用

削除の構文は、列を column() メソッドで指定する点以外、特に変わらない。更新・挿入については、単位は必ず 1 行とし、前述の区切り文字を含んだ拡張 sed 形式を利用する。

```
INSERT
  INTO 'C:¥Windows¥Favorites¥NTT.jpg'
  VALUES ('企業：通信：NTT:')
```

(4) DB モデルと問合せ言語の能力

本 DB モデルが簡易なメタデータ管理に特化し単純であること、および、問合せ言語が基本 SQL の簡素な拡張であることから、一般的な DB の能力に比べて不足している部分がある。しかし、本論文で狙うコンテンツの管理モデルは、1 回の問合せで処理を完結するモデルではなく、次節で述べる木状ビューを用いて対話的に問合せの修正を可能にするモデルであること、および、本問合せ言語は、その強力さを目指すのではなく木状ビューを経由したコンテンツ埋込 DB に必要な最低限の機能を実現することであるため、この能力の低さが木状ビューのユーザへの直接的な制約とはならない。たとえば、スキーマやデータ項目（検索の見出し語名や位置）の曖昧さがあると 1 回の問合せで結論を得ることは困難だが、ユーザが木状ビューにより確認しながら問合せを適宜再構成すること、または、木状ビュー上での操作によってコンテンツ埋込 DB 内データを更新して統一してしまうことによって柔軟な対応が可能になる。

また、複数のコンテンツからなるマルチ DB におけるスキーマの差異の吸収については、木状ビューでメタデータの分布を確認し、対話的に問合せを修正すること、または、コンテンツ埋込 DB 中のデータ自体を木状ビュー上での操作によって修正し統一してしまうことによって解決を行うことが可能である。

ただし、いずれにしても、現在の問合せ言語は最低限の機能を実現したものであり、今後、埋込 DB の機能向上にあわせて、見直しや改良の必要性が出る可能性については否定できない。

3.3 木状ビューの基本構成

木状ビューは 2 つの目的に利用する。1 つは多数のコンテンツ埋込 DB のメタデータを効率的に管理することであり、そしてもう 1 つは、コンテンツの概観を把握し効率的な再利用を支援することである。それらについて以下で述べ、さらに検索との連携については

次の 3.4 節で述べる。

(1) 木状ビューを使ったメタデータ管理

基本的にコンテンツ埋込 DB のメタデータを各フォルダ名に対応付けてそのまま木状ビューに展開する。その様子を図 10、図 11 に示す。GUI としては Windows の Explorer に類似しており、図 10 に示すように左に木状ビュー、右にリストビューがある。

リストビューは選択した木状ビューのフォルダに対応した画像の一覧を表示する。画像の拡大や縮小、画像表示と詳細表示の切替えもできる。詳細表示に関しては、3.4 節の類似検索でその利用例を示す。

木状ビューのトップフォルダは、図 10 に示すように画像登録フォルダ「画像登録」、検索条件指定フォルダ「検索」、画像目録フォルダ「top」の 3 種類のフォルダから構成される。また、図 11 に示すようにフォルダ代表画像表示モード（左）と非表示モード（右）がある。以下、各操作イベントに対応して動作を説明する。この木状ビューについての詳細は (2) で述べる。

コンテンツの追加：コンテンツの追加は 3 つの方法を用意する。1 つ目は、木状ビューの最上層の最初にある画像登録フォルダに画像群をドラッグ&ドロップすることである。これにより、画像埋込 DB への問合せ結果で得られた各メタデータを木状ビューのディレクトリのフォルダに分解し、対応するフォルダ中に画像を配置する。これを DB 中のすべてのメタデータ（行）に対して行う。

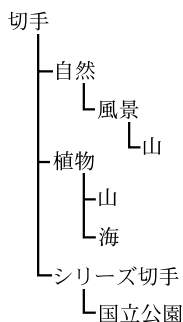
2 つ目は、画像自身を対象として画像登録フォルダに登録された特徴抽出プラグインによって特徴抽出を行い、その結果で得られたベクトルデータをもとに、特徴量による自動分類向けに定義された分類条件を持つフォルダ画像の配置を行う。分類条件としては、従来の画像分類構築支援システム⁶⁾と同様にフォルダごとに代表画像（テンプレート）のベクトルを割り当てておき、最も近いフォルダに自動配置する方法としきい値を使って判定する方法を併用する。

3 つ目の方法としては、別途、画像名とメタデータを対応させたファイル（これを td ファイルと呼ぶ）を読み込ませ、その情報で作成された木状ビューに応じたメタデータを画像内に反映させる方法を用意する。以下にその例を示す。td ファイルも拡張 sed 形式に従う。

```
| 切手 | 自然 | 風景 | 山 |E:¥a01.jpg|
| 切手 | 自然 | 風景 | 山 |E:¥a02.jpg|
| 切手 | 植物 | 山 |E:¥a03.jpg|
| 切手 | 植物 | 海 |E:¥a04.jpg|
```


| 切手 | シリーズ切手 | 国立公園 | E:¥a02.jpg |

この例では、E:¥a02.jpg が 2 つのメタデータを持つことを表す。この情報に対応して作成される木状ビューのフォルダ構成は次のようになる。



このとき「切手」-「自然」-「風景」の下の「山」フォルダを選択すると、a01.jpg と a02.jpg を右のリストビューに表示する。画像をシステムに登録する場合にも、基本的に同様の処理となる。つまり、システムは、対象画像にメタデータの一覧の問合せを行い、その結果を td ファイルと同様に解釈して、木状ビューに割り当て、表示を行う。

フォルダの削除：削除については、Windows の Explorer とは解釈が異なる。本論文の木状ビューはシソーラスに相当するメタデータの分類管理にも用いられることを想定しているため、フォルダの削除にともない行き場を失ったコンテンツ群は、その上位フォルダに移動する。これは、意味的に、より上位概念のフォルダに移動したことを表す。たとえば上記の例で「風景」とその下の「山」フォルダを削除すると、a01.jpg と a02.jpg は「自然」下に移動する。移動とともに、原則として対象画像内 DB の中の該当メタデータも修正される。

フォルダの移動や複写：フォルダの移動や複写も、削除と同様に実態ではなくそのメタデータの変更または複写に対応する。なお、この移動や複写を容易にするため、木状ビューの特定のフォルダ配下を別の Window として表示し、親 Window を経由して複数の Window 間でのドラッグ&ドロップを可能にする。移動とともに、原則として対象画像内 DB の中の該当メタデータも修正される。

フォルダ名を更新：フォルダ名を更新した場合には、そこに含まれる画像埋込 DB の対応するメタデータも原則として一括更新する。

同期の問題：メタデータは木状ビュー（プロセス）

中とコンテンツ埋込 DB の 2 か所に存在することになる。一般には、木状ビュー上での操作と同時にコンテンツ埋込 DB が更新されるリアルタイム更新のモードが基本であり、その場合、更新プロセスを限定することで、多くの同期ズレの問題を回避することができる。しかし、たとえ 1 プロセスであってもコンテンツ埋込 DB のアクセス処理は負荷が高く、スペックの劣るマシン環境では操作性に問題が発生する可能性がある。そこで、コンテンツ埋込 DB への一括更新モードも設け、リアルタイム更新モードとの切替えを可能にする。そして、木状ビューとコンテンツ埋込 DB 間に情報の不一致があった場合には、それらを自動補正するのではなく、双方の情報をマージした情報を双方に反映し、改めて木状ビューで修正してもらうという方針とする。

(2) 木状ビューによる画像目録閲覧システム

この木状ビューは我々の従来の研究である目録閲覧システムを包含するシステムとして設計した。つまり、最初に述べた課題 1, 2 に対処することも目的としている。そこで、従来の目録に相当する td ファイルを読み込むことで、画像の分類結果を即表示することができるようにする。この場合はシステム起動時間を短縮するため、前節で述べた td ファイルとコンテンツ埋込 DB の内容確認（とマージ処理）は行わない。td ファイルとコンテンツ埋込 DB との一貫性が必要な場合には、さらに前節で述べた一括更新操作が必要になる。

また、各フォルダ内画像の効率的な内容把握を目的として、フォルダ代表画像表示モード（図 11 左）ではその配下のフォルダ内の画像中の代表 3 枚を選択し、フォルダ表示に利用する。この表示により、フォルダ配下のコンテンツを概観することが可能になる。なお、画像数が 3 枚に足りない場合にはグレー領域となる。また、下位のフォルダのコンテンツが移動した場合には、その表示はリアルタイムに変更される。さらに、木状ビューの背景部分に木状ビューの大局情報を表示することも可能とする。

3.4 木状ビューと検索との連携

我々は、目録閲覧と類似検索を連携することが効率的なコンテンツ管理になるという結論⁶⁾を得ている。前述のように木状ビューは目録閲覧システムとしても動作するため、検索との連携について検討し、木状ビュー、つまりフォルダ・インタフェースを利用した検索と結果表示のインタフェースも設計した。

(1) 特定画像の持つメタデータの検索

特定画像が持つメタデータの一覧を得る方法として、一般にはメタデータをリスト状に一覧したイン

タフェースが考えられる。しかし、この表示法は木状ビューとはまったく独立で、木状ビューの操作に対する効果的な支援とはいえない。そこで、特定の画像をキーとして検索したとき、その画像が含まれるフォルダパスのみを選択状態（開いた状態）にする機能を設ける。たとえば図 12 でリストビューに示すアヒル（「動物（空・鳥）」-「アヒル」）の画像をキーとして検索した場合に、それが含まれるフォルダ（「道具」-「育児用品」-「おまる」、および「道具」-「玩具」-「人形」）が、図中で「検索結果」と表示される。この「検索結果」表示は、そのフォルダをいったん開くと通常の代表画像表示に戻る。この機能は、複数の項目に対してメタデータがある場合に、そのメタデータを把握し、かつ、木状ビューでの速やかな修正を行うことを可能とする。なお、図 12 右のように代表画像の非表示モードの場合には、フォルダ色が変化（緑 → 赤）することで「検索結果」フォルダを示す。

(2) 類似検索

類似検索に関してもフォルダに基づく検索制御インタフェースとする。これは、検索の条件および検索結果をそれぞれフォルダ経由で受け渡すインタフェースであり、木状ビューの 2 番目のフォルダ配下に検索条件を構成する。また、検索に利用するフォルダ条件は検索条件制御プラグインとしてシステムに与える。

従来の我々の画像検索システムでは拡張可能 OR-DBMS LiteObject¹⁶⁾の拡張 SQL に類似検索関連のユーザ定義メソッドとオペレータをプラグインすることで対応していた。この経験から、類似検索の構文が画像全体、画像内オブジェクト、画像内直線、各種フィルタ条件などを想定した場合にはきわめて多岐にわたることが分かった。そこで、それに柔軟に対応する問合せ条件を設定できる必要があると考え、フォルダをベースとして各種の問合せ条件を柔軟に表現できる構成を目標とした。フォルダ構成は検索順序の制御に対応する。つまり通常の条件式の括弧に相当し、下位のフォルダは優先的に計算する。また、フォルダは、そのフォルダ内の情報を使って検索制御を行うメソッドを 1 つ保持する。そのために以下のような検索条件プラグインを用意し、それらを組み合わせたい問合せを可能にする。なお、メソッドを持たないフォルダは AND メソッドを持つフォルダと解釈する。

AND メソッド：子フォルダ計算結果のすべての画像 ID の AND 集合を返却する。

OR メソッド：子フォルダ計算結果のすべての画像 ID の OR 集合を返却する

フォルダ名検索メソッド：メソッドのパラメータ

フィールドで指定したキーワードをフォルダ名としても持つ画像の ID を求める。

Vector 指定による類似検索メソッド：メソッドのパラメータフィールドには、フォルダのキーワード条件、ベクトル値を表す文字列、類似検索の返却件数 k を与える。このキーワードとしては一般にフォルダ名として表現されている特徴量名称を与える。システムはキーワード条件を満たすフォルダの特徴ベクトルと与えられたベクトルの類似度計算を行い、その上位 k 件分の画像 ID を類似度属性とともに返却する。その検索条件の指定例を図 13 に示す。検索結果も図 14 に示すようにフォルダとして表現する。検索条件フォルダには名称が自動付与されるが、それと同じ名称を持ったフォルダが top フォルダ配下に作成され、そこに検索結果画像が格納される。また、同時に各画像の属性に類似度が付く。

キー画像指定による類似検索メソッド：メソッドの名称フィールドには、フォルダのキーワード条件、類似検索の返却件数 k を与える。また、メソッドのフォルダには検索キーとなる特徴ベクトルを埋込 DB 内に有する画像を登録する。システムは画像埋込 DB に対しキーワード検索を行い、キーワード条件に合致したフィールドの次のフィールド値をキーベクトルとして返却し、それをもとに、上記 Vector 指定による類似検索と同様の処理を実行する。

類似検索結果の統合メソッド：メソッドの名称フィールドには下位フォルダ名と重みを使った統合重み計算式と、返却件数 k を与える。システムは、各々の下位フォルダの対応する画像の類似度を重みに従って再計算して求め、その上位 k 件の画像 ID を結果として返却する。

なお、これらの条件の組合せは、検索フォルダ配下のテンプレートフォルダ中にテンプレートとして格納しておき再利用を促す。また、検索後は一括更新モードになり、検索結果がただちに画像埋込 DB に反映されることはない。

4. 適用事例と考察

4.1 実装の詳細

今回、前章での提案および設計に基づくシステムの実装を行った。画像埋込 DB への問合せプログラムは Solaris2.6 以上と Windows 98/NT4.0/2000 で動作する。また、木状ビューは Windows 98/NT4.0/2000 で動作する。エンディアンは Intel に合わせた。

プラグインとしては、画像の表示に関しては Susie のプラグインも利用できる構成としているため、すでに多種の画像表示に対応できる。画像埋込 DB のアクセスプラグインとしては JPEG (JFIF) 対応版および後に述べる URL 版を用意した。検索条件制御プラグインとしては、前章で述べたうち、典型的なものとして AND メソッド、フォルダ名検索メソッド、Vector 指定による類似検索メソッドを用意した。登録のための特徴抽出メソッドについては今回は実装を行っていない。

4.2 適用実験

コンテンツとして、切手 3,841 枚、クリップアート 900 枚¹⁷⁾について本システムの適用を確認した。その適用の様子は図 10～図 14 に示したとおりである。また、本画像埋込 DB に情報を埋め込んだ JPEG ファイルが、Adobe Photoshop および Internet Explorer で正常に表示されることを確認した。画像内 DB のアクセスは、画像自身の読み込みに比較すると一般には小さいため、起動時の時間は原理的にはそれほどはかからないが、それでも大量に画像の入ったフォルダを移動した場合などにはかなりの処理時間が必要となった。そこで、3 章で述べたように一括更新モードを設けた。なお、更新時間は本論文の本質ではないため詳細については割愛する。

一方、この論文での検証すべき重要な点は、本枠組みが画像に限らず多様なコンテンツの管理に利用できることであると考えられる。そこで、画像ではなく、URL リンクファイルの管理に適用を試みた。Windows の URL タイプドキュメント (または、「お気に入り」) は、一般には Microsoft Internet Explorer 上で管理できるが、OS 上は Favorites フォルダ内に URL タイプのファイルとして存在している。その管理に画像用に設計したシステムを適用した。その際、システムの変更点は 2 つであった。1 つ目は、URL タイプファイルの埋込 DB アクセス・プラグインの作成である。URL ファイルは [見出し] からなるいくつかのテキストブロックで管理されている。そこで、コンテンツ埋込 DB ブロック [InfoBase] を構成し、JPEG の場合と同様に拡張 sed 形式を使って、メタデータを使った管理が行えることを確認した。実際に構成したコンテンツの中身を図 15 に示す。JPEG 画像の場合には既存の COM エリアの内容を複製したが、この URL タイプファイルの場合には [InternetShortcut] ブロック中の URL 属性を DB 域に複製した。2 つ目の変更点は、そのリストビューの表示である。URL については画像が表示対象ではないため、リストビューでは詳

```
[DEFAULT]
BASEURL=http://www-db.isl.ntt.co.jp/~exdemo/

[InternetShortcut]
URL=http://www-db.isl.ntt.co.jp/~exdemo/
Modified=209E30162721C0010D

[InfoBase]
:INFOBASE-FORMAT-VERSION:2000&02&17:
?URL?http://www-db.isl.ntt.co.jp/~exdemo/?
:部門:DBサーバ:画像内容検索システム:
:名称:ImageCompass:
:提供元:NTT:サイバースペース研究所:
:販売価格:?
```

図 15 URL リンクファイルに埋込 DB を構成した例
Fig. 15 Example of URL link file-embedded DB.

細表示モードを利用し、埋込 DB 中の URL 情報をリストの属性として表示させるようにした。

なお、より効果的に URL の管理を行う場合には URL を画像で管理すること、つまり、URL に基づいてホームページにアクセスし、そのページそのものを JPEG のサムネイル画像として取得し、その画像を中心に URL を管理することが望ましいと考えるが、今回は、そこまでの実装は行っていない。

4.3 効果

以下に、本システムの効果をまとめる。

(1) メタデータとの一体化

コンテンツとメタデータが一体として構成されることで、コンテンツの流通過程に対応し、課題 3 を解決した。

(2) 多数のコンテンツ中のメタデータの概観

コンテンツ流通過程の中の任意の過程で、特にサーバなどの設定を行わなくてもただちにメタデータの木状ビューを構成することができ、その時点でのコンテンツの集合のメタデータがどのような傾向を有するのかが確認することができる。また、効率的にメタデータの管理ができ、一貫性の管理も容易に行える。これにより課題 4 を解決した。

(3) 多様なメディアへの展開

多様なコンテンツ内に DB を構成することが可能である。また、汎用性の高いインタフェースでその中のメタデータをアクセス可能にすれば、異なるメディアに対する共通の管理操作が可能になる。

4.4 類似技術との比較

(1) 電子透かし技術との比較

画像や音楽の中に周波数成分として情報を埋め込む電子透かし (watermark) 技術がある。この技術もコンテンツとメタデータを同時に流通させることができる。この技術は、不正利用の検出などに用いられるが、埋め込める情報量に制限が強いという問題がある。また、情報をたくさん埋め込むとコンテンツが劣化する

という問題もある．さらに効率の良いメタデータの更新処理が難しい．

(2) コンテンツカプセル化技術との比較

コンテンツに対して利用条件を付加しネットワーク上での不正利用を排除した安全な流通を実現する技術として Matryoshka などのコンテンツ・カプセル化技術¹⁸⁾がある．本システムもコンテンツの流通の側面を重視している部分が似ているが、プレイヤーはコンテンツと切り離された汎用的なものを想定していること、利用条件やセキュリティなどは考慮していないことなど、目的も含め互いに独立な技術である．よって、埋込 DB を有する JPEG 画像を Matryoshka によってカプセル化して配送することも可能である．

4.5 今後の課題

(1) DB 基本機能の充実

現状のコンテンツ埋込 DB および今回作成した SQL ベースの問合せ言語は、一般的な RDB や SQL ほど強力な機能を持っていない．これは、木状ビューで対話的にコンテンツ埋込 DB を管理するのに最低限の機能しか実現していないためである．索引とアクセスパス制御、トランザクション制御、セキュリティ、履歴管理などを今後どの程度までサポートすべきかを検討する必要がある．また、今回は性能やメディアフォーマットとの兼ね合いから見合わせたか、コンテンツの流通性を考えた場合、埋込 DB 情報の圧縮も重要な課題である．

(2) XML 対応

我々の提案したコンテンツ埋込 DB の 1 フィールドは可変長で、かつ、改行文字を含むことが可能なので、その情報として XML データを格納することは可能である．また、td ファイルの構成を XML に変更するのも容易である．それ以外の部分、たとえば、XML データ自身をフォルダ・インタフェースで管理すべきなのか、問合せ言語として XML をサポートすべきなのか、XML と類似検索の構文の整合性をどう実現するのかなど、XML にどう対応すべきかについては今後の重要な課題であると考えている．

(3) マルチ DB 対応

コンテンツ内に DB を構成した大きな理由の 1 つが 2.1 節でも述べたマルチ DB 対応であった．しかし、現状では、動画やより高次のメディアへの対応までには至っていない．今後は、コンテンツ内の特定の部分コンテンツに対する問合せ、たとえば Microsoft Word 内の特定の条件を満たす画像に対する問合せなどの高度な処理方式の検討やプラグインの充実が必要である．

また、WWW を経由した分散 DB に対する検討も

興味がある．今回、FROM 句に URL を利用することも考えたが、http: では画像埋込 DB の更新ができないため、実現を保留した．sql: などのような、より DB 向きのプロトコルの利用の検討が必要である．

(4) 特徴量による自動分類とメタデータの自動付与

3.3 節のコンテンツの追加で最初に述べた登録フォルダへのコンテンツのドロップによる自動追加処理、つまり、フォルダにドロップされたコンテンツに適用可能な特徴抽出処理を自動的にを行い、得られた結果からコンテンツを適切なフォルダに自動配置する機能は、画像分類木構築支援システム⁶⁾では実装したものの、本システム上では、現在、未実装である．今後、汎用性のある特徴抽出メソッドのプラグイン・インタフェースを含めた実装方式を検討しなければならない．

(5) 背景の有効活用

木状ビューをより効果的に利用するために、2.2 節で述べたように、その背景に大局情報を表示することは大変有効だと考えている．しかし、どの情報を表示すべきかについて我々の結論がまだ出ていないため、本論文での記載は見送った．また、現在の実装では、その実行速度にも問題がある．今後のシステムでは、実装の改善とともに、概観把握のための効果的な提示情報について検討しなければならない．

5. ま と め

本論文では、主に以下の 2 点を提案した．

コンテンツのコメント域や拡張域に DB を構築すること．また、それを汎用性のある DB インタフェースでアクセス可能にすること．

コンテンツ埋込 DB と木状ビューとを関連付けてメタデータを管理すること．

これらの手法は、コンテンツのメタデータに対する効果的な維持管理インタフェースとなるとともに、コンテンツの流通の各過程においても効率的なコンテンツの概観および管理手段となる．その手法を利用したシステムの設計を示し、実験として、約 4,000 枚の画像群に適用した．また、画像以外のコンテンツにも適用できることを示すため、Microsoft Internet Explorer の「お気に入りの追加」で作成される URL タイプファイルに対しても同様に適用できることを実験で検証した．最後に、本アプローチに関する今後の検討課題を示し、今後の研究の方向性に関して指針を示した．

謝辞 実験対象の切手素材を提供していただいた、ていばーくに感謝します．本論文の図中で利用しているクリップアート CG 画像は SIFCA¹⁷⁾の作品です．査読者の方々には大変重要なお指摘をいただきました．

心から感謝いたします。

参 考 文 献

- 1) 赤間浩樹, 三井一能, 紺谷精一, 串間和彦: 画像内オブジェクトの自動抽出を使った画像検索システム ExSight, 電子情報通信学会, データ工学ワークショップ論文集 (DEWS'97), pp.107-112 (1997).
- 2) 串間和彦, 赤間浩樹, 紺谷精一, 木本晴夫, 山室雅司: オブジェクトにもとづく高速画像検索システム: ExSight, 情報処理学会論文誌, Vol.40, No.2, pp.732-741 (1999).
- 3) 串間和彦, 赤間浩樹, 紺谷精一, 山室雅司: 色や形状等の表層的特徴量にもとづく画像内容検索技術, 情報処理学会論文誌: データベース, Vol.40, No.SIG 3 (TOD 1), pp.171-184 (1999).
- 4) Kushima, K., Akama, H., Konya, S. and Yamamuro, M.: ExSight: Highly Accurate Object Based Image Retrieval System Enhanced by Redundant Object Extraction, *Proc. 1st Int. Conf. On Web-Age Information Management (WAIM2000)*, LNCS 1846, pp.331-343, Springer (2000).
- 5) 串間和彦, 佐藤路恵, 赤間浩樹, 山室雅司: 大量画像の閲覧を目的とする階層的分類支援機能—画像目録の実装と評価, 情報処理学会論文誌: データベース, Vol.41, No.SIG 1 (TOD 5), pp.54-63 (2000). (平成 12 年 9 月 20 日受付)
- 6) Kushima, K., Satoh, M., Akama, H. and Yamamuro, M.: Integration Hierarchical Classification and Content-based Image Retrieval — ImageCompass, *IFIP WCC2000 (World Computer Congress), Proc. IIP2000*, pp.179-187 (2000). (平成 13 年 1 月 27 日採録)
- 7) 角谷和俊, 川口知昭, 土居明弘, 赤迫貴行, 田中克己: インデックス付きライブ映像ストリームによる動的番組生成と配信, 情報処理学会論文誌: データベース, Vol.41, No.SIG 1 (TOD 5), pp.87-99 (2000).
- 8) 三井一能, 寺西裕一, 小島 明, 赤間浩樹, 佐藤哲司, 寺中勝美, 鎌原淳三, 下條真司, 西尾章治郎: ニュースコンテンツの検索と編集機能を融合するニュースオンデマンドシステムの設計と実装, 電子情報通信学会和文論文誌 D-II, Vol.J81-D-II, No.3, pp.516-528 (1998).
- 9) 波多野賢治, 亀井俊之, 田中克己: 多段階自己組織化マップによるビデオ映像記述支援と類似シーン検索, 情報処理学会論文誌, Vol.39, No.4, pp.933-942 (1998).
- 10) 日本電子工業振興会 JEIDA, デジタルスチルカメラ用画像ファイルフォーマット規格 (Exif) Ver 2.1. <http://www.jeida.or.jp>, (1998).
- 11) 日本電子工業振興会 JEIDA, カメラファイルシステム規格 (Design rule for Camera File system (略称 DCF)), <http://www.jeida.or.jp>. (1998).
- 12) Digital Imaging Group, DIG35. <http://www.digitalimaging.org/>
- 13) 原 隆浩, 春本 要, 塚本昌彦, 西尾章治郎, 奥井 順: 広帯域ネットワーク上のデータベース移動に基づく動的複製配置法, 電子情報通信学会論文誌 D-I, Vol.J82-D-I, No.8, pp.1049-1058 (Aug. 1999).
- 14) 岩崎雅二郎, 両角清隆: 大量画像データベースへのアクセス手段の提案, 情報処理学会研究報告 99-DBS-119, pp.43-48 (1999).
- 15) susie. <http://www.digitalpad.co.jp/~takechin/>
- 16) 岡田 敏, 鬼塚 真: 画像検索アプリケーションにおける問い合わせ機構, 情報処理学会研究報告 98-DBS-116, pp.25-32 (1998).
- 17) SIFCA, ホームページ素材集 HOT&プライベート, インプレス (1999).
- 18) 谷口展郎, 森賀邦広, 久松正和, 櫻井紀彦: マルチメディア情報ベースとその格納単位 Matryoshka, 情報処理学会マルチメディア・分散・協調とモバイルシンポジウム論文集 (DICOM'99), pp.207-212 (1999).

(担当編集委員 安達 淳)



赤間 浩樹 (正会員)

1988 年東海大学理学部情報数理学科卒業。1990 年同大学院理学研究科数学専攻修士課程修了。同年日本電信電話株式会社入社。以来、インターネット・ネットワーク向け

DBMS の開発、ニュース・オン・デマンドの研究開発、マルチメディア情報検索の研究開発に従事。2000 年 10 月より東日本電信電話株式会社研究開発センタに勤務。現在は WebGIS システムの研究開発に従事。人工知能学会、日本ソフトウェア科学会、ACM 各会員。



三井 一能 (正会員)

1991年京都大学理学部卒業。1993年同大学院工学研究科修士課程修了。同年日本電信電話株式会社入社。以来、ニュース・オン・デマンドの研究開発、マルチメディア情報検索の

研究開発に従事。



串間 和彦 (正会員)

1980年京都大学工学部電子工学科卒業。2001年博士(情報学, 京都大学)。1980年日本電信電話公社(現NTT)入社。知識処理用プログラミング環境の研究、大規模クライ

アントサーバシステムの実用化等を経て、現在はマルチメディアデータベースの研究開発に従事。
