

HDDの情報検索方法

Data Search Method for HDD

北 尾 英 樹 Hideki Kitao
東 山 雅 彦 Masahiko Higashiyama
福 田 晋 児 Shinji Fukuda



要 旨

今回、オーディオのタイトル検索を対象とし、リスト型のテキスト情報を扱うのに好適な情報検索ソフトウェアを開発した。

本ソフトウェアは、登録／編集／削除／検索／検索結果取得の機能を備えており、データのレコード位置を格納したインデックスを列毎に分離作成することで、検索対象と出力対象を個別に選択可能な柔軟な検索システムの実装を可能としている。また、インデックス部をメモリ上にロードすることで、ディスクアクセス回数の削減と、省メモリかつ高速な検索を実現した。

本論文では、開発した情報検索ソフトウェアの概要と基本性能について紹介する。

尚、本ソフトウェアは、2002年11月に発売されたHDD-AVNにMusicJuke機能の一部として搭載している。

Abstract

Fujitsu Ten has recently developed data search software that is targeted at audio title searching and is also suitable for handling any text data in list format.

The software is provided with functions for registering, editing, deleting and searching, and for acquiring the search results. It compiles each of various "groups" of data into a separate index containing the "record" (line) positions of the data items, thus enabling construction of a flexible system permitting search objects and output objects (results) to be selected separately. Further, the index component is loaded into the memory, realizing high-speed and memory-saving searches requiring fewer accesses to the disk.

This paper presents an overview of the newly-developed data search software plus its basic performance.

This software is included in the HDD-AVN launched in November 2002, forming part of its "Music Juke" functions.

1

はじめに

自宅のパソコン（以下PC）やインターネットを介して音楽や動画を楽しむ人が年々増加している。それに伴い、家庭で取り込んだ音楽データを楽しむことが可能な車載機器の人気も高まっている。これらの機器には、MP3データを再生可能なCDドライブやHDD(Hard Disk Drive)が搭載されており、数百～数千の曲を扱うことが可能になっている。

一方、携帯電話を利用したインターネット接続機能を持つ車載機器も増加傾向にあり、HDDに格納されるデータは、従来の地図や音楽だけでなく、POI(Points Of Interest)やE-mailなどに拡大していくと予想される。

データの数や種類が増えるにつれて、データの扱いは煩雑になり、ユーザーには「素早く必要な情報に辿り着きたい」という欲求が生まれる。このような状態では、検索機能が重要な役割を担うこととなる。

今回、オーディオのタイトル検索を対象とし、リスト型のテキスト情報を扱うのに最適な情報検索ソフトウェアを開発したので、その概要を紹介する。

尚、本検索ソフトウェアは、2002年11月に発売されるHDD-AVNにMusicJuke機能の一部として搭載されている。

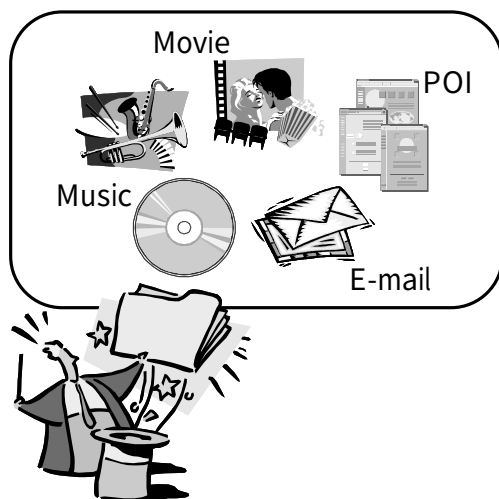


図-1 多種多様な情報から素早く検索
Fig.1 Rapid searching of highly varied data

2

開発のねらい

今回の開発では、HDDを搭載した車載機に格納される圧縮音楽の情報検索機能を実現した。

情報検索ソフトウェアは、マルチメディアデータとそのデータを利用するアプリケーションとの仲介を、ユーザデ

ータベースを通して行うソフトウェアである。ユーザデータベースおよびマルチメディアデータはファイルシステムを利用してアクセスされる。(図2)

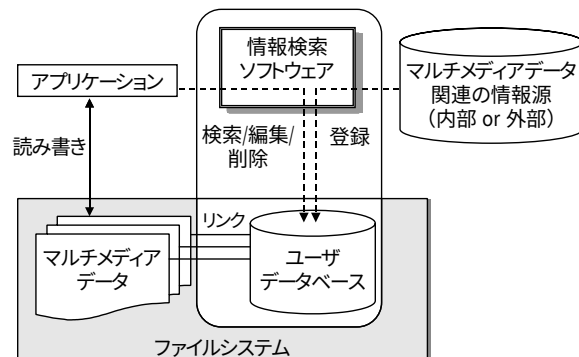


図-2 情報検索ソフトウェアの位置付け
Fig.2 Data search software's place in the setup

2.1 情報検索ソフトウェアへの要求事項

①高速であること

車載機器では車両を運転中、もしくは、信号などで一時停止している短い間で操作できることが必須条件である。一方、車載向けのCPUの演算能力はPC向けのものの1/10以下であり、演算処理を極力抑えなければならない。また、HDDは高速なデバイスと捕らえがちであるが、外部I/Fに接続されたデバイスであることと、データを読み書きするためにATAやSCSIのプロトコルによるオーバーヘッドが発生するという性質上、細かいデータでランダムにアクセスするのは苦手である(表1)。そのため、HDDへのランダムアクセスの回数を極力減らす工夫が必要である。

表-1 HDDのアクセス速度の一例 (ATA PIO mode 0)
Table 1 Sample HDD accessing speeds (ATA PIO mode 0)

転送方法	読み出し	書き込み
64KB×32回	500KB/s	300KB/s
2MB×1回	2.2MB/s	1.9MB/s

②少ないメモリで動作すること

車載機器に搭載される全メモリは、中級機で数MB、高級機でも32MB以内が一般的である。したがって、検索ソフトウェアなどの一機能では、少なくとも1MB以内で動作することが求められる。

③一つの格納データに複数の情報を付与できること

いかなる情報にも1つの格納データに複数の情報が存在する。例えばオーディオタイトルを扱う場合には、曲名、アーティスト名、アルバム名、データの格納場所などが必須である。以下ではこの複数の情報の塊を「レコード」と呼ぶ。

④情報の格納や編集の機能を備えること

一般に、格納するデータ本体からは十分な情報が得られることは稀であり、別の情報源から得られる情報を、情報源とは別の場所（図2のユーザデータベース）に保存して利用することになる。そこで、情報の登録／削除／編集／検索の機能を搭載する必要がある。

⑤検索対象と、その結果の出力対象を分離すること

「〇〇〇というアーティストの曲を探す」場面を考える。このとき検索する対象（以下グループ）は、「アーティスト名」であり、アプリケーションはこのグループの中から「〇〇〇」という名称のレコードを探すように検索ソフトウェアに要求する。一方、アプリケーションがユーザに提示するデータ（出力対象）は、検索にヒットしたレコードに含まれる曲名というグループの一覧である。この提示データは、別の場面ではアルバム名というグループの一覧に変化することもあり、検索対象とは分離して扱うべきである。

表2に、今回開発した情報検索ソフトウェアの目標仕様を示す。尚、開発当初のHDD-AVNの仕様を考慮し、CPUにはSH7709Aを、OSにはμItronを採用した。

表-2 情報検索ソフトウェアの目標仕様
Table 2 Target specifications for data search software

項目	目標仕様
検索速度	検索キーの入力から結果の出力まで1秒以内
検索対象件数	1500件
照合方法	完全一致
機能	登録、編集、削除、検索、検索結果取得
動作メモリ	500KB以下

3 情報検索ソフトウェア概要

今回開発を行った情報検索ソフトウェアを搭載する検索エンジンの全体構成を図3に示す。本情報検索ソフトウェアは、データベースにテーブルを作成しデータの管理を行い、検索用のインデックスデータを作成し検索を実行するものである。

3.1 情報検索ソフトウェアの構成

1) DBMS (Database Management System)

DBMSはアプリケーションとデータベース間のインターフェイスである。アプリケーションのデータ操作要求（登録／削除／編集／検索）を受け付け、テーブル管理、イン

デックス管理、レコード管理の各機能を呼び出し、データベースに対する操作を実行するシステムである。

2) テーブル管理

データベースに構築されたテーブルデータと検索用に作成されたインデックスデータの構成情報の管理を行い、インデックス管理、レコード管理からのパラメータの問い合わせに応答する。

3) インデックス管理

テーブルに登録された情報を、高速に検索するためのインデックスデータの管理を行い、検索要求に対してデータの格納位置を応答する。

4) レコード管理

データベースに構築されたテーブルデータを、後述するように行を1つのデータのまとまりとして管理する。このまとまりをレコードとし、アプリケーションから受け取った登録データのレコードへの格納、レコードからデータの取り出しを実行する。

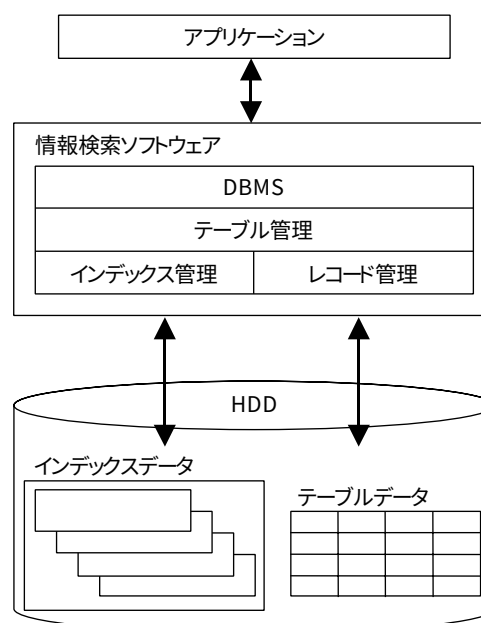


図-3 情報検索エンジンの構成
Fig.3 Configuration of data search engine

3.2 テーブル構造

①テーブルデータ

登録データの格納構造として、レコード（行）とグループ（列）で表現できる2次元のテーブルをデータベースに作成し、データの管理を行う。テーブルに格納された各データにはレコードの位置を表すRecordIDとグループの位置を表すGroupIDが定義されており、アプリケーションはこれらのIDを使用してデータにアクセスを行う。（図4）

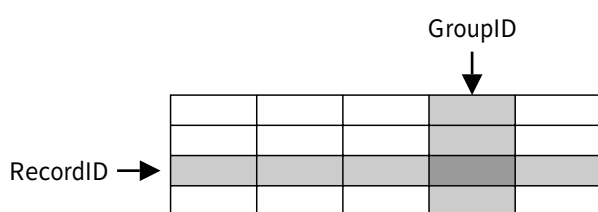


図-4 テーブルデータアクセス
Fig.4 Accessing of table data

②テーブルカスタマイズ

多数のアプリケーションで使用可能な汎用性を実現するには、登録や検索の対象となるデータの数や種類が、アプリケーションごとに違ってくことを考慮しなければならない。本情報検索ソフトウェアはアプリケーションから必要とするテーブルのレコード数、グループ数、登録データの種類（文字列、数値）といったテーブルの定義を行うデータを受け取り、データベース内にカスタマイズされたテーブルを作成する。（図5）

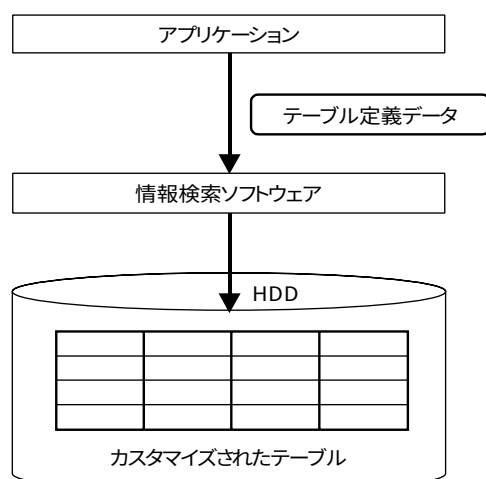


図-5 テーブルデータ作成
Fig.5 Preparation of table data

3.3 インデックス構造

①インデックスデータ

本情報検索ソフトウェアは検索処理を高速に実行するため、テーブルデータから検索に適したインデックスデータを作成する。インデックスデータは、例えるならば目次のようなデータであり、対応する登録データのRecordIDを記述したデータである。テーブルの中から目的のデータが格納されたレコードを探したい場合、全データを最初から走査し探すよりも、目次を参照しどのレコードに格納されているか調べる方が、簡単に目的のレコードを見つけ出すことができる。

②インデックス適用範囲

インデックスデータは、データベースに構築されたテーブルデータの各グループに対して作成され、各グループごとに検索することができる。アプリケーションは、登録データの属性等に従いグループを分類してテーブルを定義することにより、アーティスト名検索、アルバム名検索といった、対象を指定した検索が可能になる。（図6）

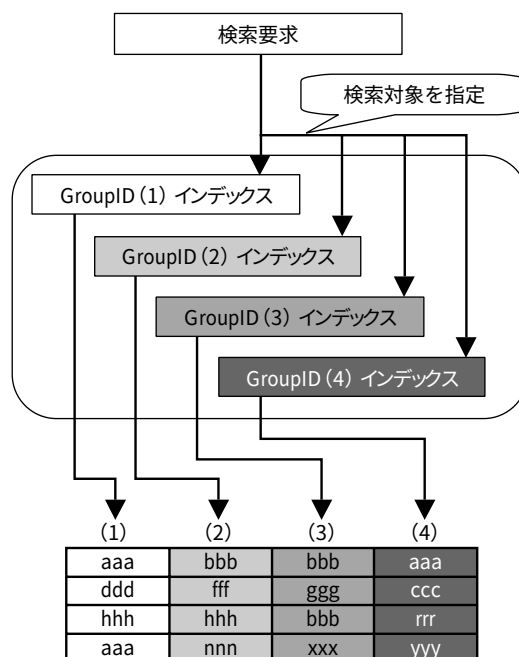


図-6 インデックス適用範囲
Fig.6 Scopes of indexes

③高速動作と省メモリ

インデックスデータを使用すれば高速な検索が実行できる。しかし、検索処理時にインデックスデータを、メモリよりも、はるかに低速なHDDから読み出していたのでは、高速な検索処理は期待できない。インデックスデータを、車載機器に実装される少ないメモリに、常時配置できる程小さくすることが重要である。本情報検索ソフトウェアでは、インデックスをRecordIDで構築することによりデータの圧縮を実現している。

④インデックス構造

本情報検索ソフトウェアはインデックス構造にハッシュ法を用いている。ハッシュ法は、登録／削除／検索の操作を登録データの個数に依存せず実行することができるデータ構造である。

この構造の原理は、キーの値xを配列の添え字へ写像する関数h(x)を導入し、データを格納される位置に直接関連付けるといものである。一般のハッシュ法では同一データを登録することはできず、検索適合数は1つとなる。本

情報検索ソフトウェアでは、同一データが登録できるよう拡張し、複数レコードに同一データが格納された状態でも、全ての適合レコードを検索することが可能になっている。

⑤検索方法

本情報検索ソフトウェアは検索キーリストを受け取り、同時に複数回の検索を実行することが可能である。各検索の実行結果としてRecordIDのリストをバッファに保持し、検索終了後に全てのRecordIDリストから、重複IDを抽出しマージすることにより絞り込み検索（AND検索）を実現している。検索終了後の重複ID抽出処理を、重複を省き検索に該当した全てのIDを抽出しマージする処理に切り替えれば、OR検索を実行することができるよう拡張可能である。（図7）

検索処理の結果として、該当したRecordIDリストをアプリケーションに返している。このRecordIDリストを用いてテーブルにアクセスすることにより、検索結果のデータを取得することができる。アプリケーションは、目的に応じて必要なデータを切り替えることにより、少ないメモリ使用量で取得データの処理を行うことができる。例えば、検索に該当した曲のタイトル表示から、録音日付の表示に切り替える場合、再度検索処理を実行する必要が無く、タイトルデータ格納バッファと録音日付データ格納バッファを同一にすることができるので、省メモリで処理することができる。

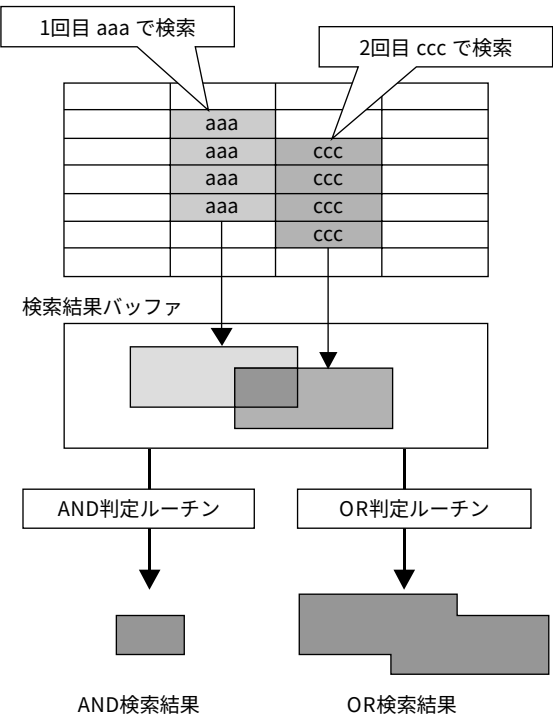


図-7 検索方法

Fig.7 Search method

3.4 インデックス構成法の選定

本情報検索ソフトウェアは、インデックスデータ構造にハッシュ法を採用している。このハッシュ法以外にも、一般的に構造としてよく用いられる方法に、「木」状の構造を持つ「B木」が知られている。

ハッシュ法はハッシュテーブルを構成する配列と、同一ハッシュ値のデータを管理する線形リストからなる単純なデータ構造をとる。ハッシュ構造へのデータの登録／削除は、線形リストの操作のみを行えば良いので高速に実行することができる。これに対してB木は、木構造の起点となる「根」、分岐点となる「節」、末端の「葉」で構成されるバランスの取れた多分木構造を構成する。しかし、節に複数の接点を持つためデータ構造は複雑になり、登録／削除は構造のバランスをとるための計算量の多い複雑な処理を行わなければならない。

B木は節に複数の接点情報を持たなければならないため、メモリ使用量が多くなり、順次HDDから部分的に節データを読み込みながら、木構造をたどることが必要になる。

これに対しハッシュ法は、データ構造が単純なためメモリ使用量が少なく、常時メモリに配置することができる。登録／削除／変更が頻繁に発生しても、メモリ上で処理することができるため、HDDへのアクセス回数が少ない高速な処理が可能になる。

表-3 ハッシュ法とB木の特徴
Table 3 Characteristics of hash and B-tree methods

項目	ハッシュ法	B木
構造	○	×
メモリ使用量	○	×
検索速度	○	○
登録・削除速度	○	×
変更速度	○	×
選定	◎	—

ハッシュ法は登録／削除／変更機能が必要な小中規模のシステムに向いており、これらの理由により本情報検索ソフトウェアは、インデックスの構造としてハッシュ法を採用した。

4

性能測定

本情報検索ソフトウェアは、高速な検索を実行するため、テーブルデータのインデックスを作成し、インデックスから該当するデータを見つけ出す機能を実装した。検索処理におけるこのインデックスの有効性を、検索速度を測定することにより確かめた。

4.1 測定システム構成

測定システムの構成を図8に示す。

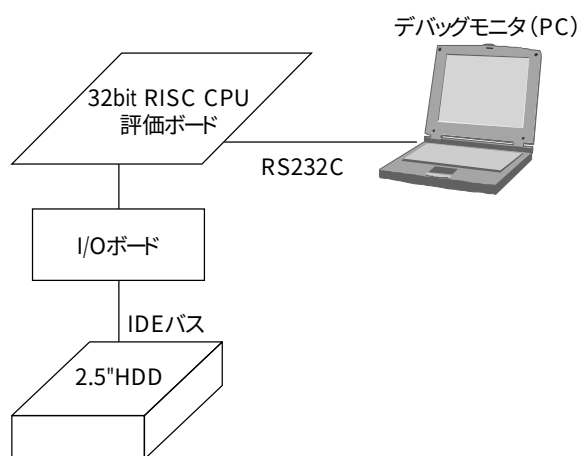


図-8 測定システム構成

Fig.8 Configuration of performance measuring system

4.2 測定方法

インデックスの有効性を確認するため、管理データを持たない検索方法を用いるソフトウェアを別途作成し、測定結果の比較に用いた。この別途作成した管理データを持たないソフトウェアは、検索処理時に順次HDDからテーブルのレコードを読み出し比較を行う。本情報検索ソフトウェアと別途作成したソフトウェアを以下の手順で性能測定を行った。

①測定用登録データの準備

性能測定を行うため、データベースに構築したテーブルにデータを登録する必要がある。最大検索時間を測定するため以下の3つのタイプのデータを作成した。グループ規定を表4、レコード構造を図9に示す。

1) タイプA

先頭グループと末尾グループの各レコードに1～1500の数値データを格納。

例： 0001, 0002, 0003, . . .

2) タイプB

任意グループのすべてのレコードに格納可能最大幅の同一文字列データを格納。

例： "あいうえお", "あいうえお", . . .

3) タイプC

任意グループの各レコードに格納可能最大幅の異なる文字列データを格納。

例： "A0001", "A0002", "A0003", . . .

表-4 グループ規定

Table 4 Group prescriptions

グループ	データ	タイプ
1	数値データ (2)	タイプA
2	同一文字列 (256)	タイプB
3	一意文字列 (256)	タイプC
4	同一文字列 (256)	タイプB
5	一意文字列 (256)	タイプC
6	同一文字列 (256)	タイプB
7	一意文字列 (256)	タイプC
8	同一文字列 (256)	タイプB
9	一意文字列 (256)	タイプC
10	数値データ (2)	タイプA

※()内はByte数

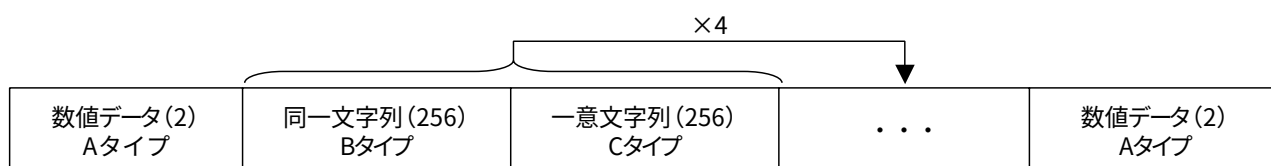
上記の登録データを用いた際のデータベースサイズとメモリ使用量を表5に示す。

表-5 データベースサイズとメモリ使用量

Table 5 Database size and memory consumption

	情報検索 ソフトウェア	比較用 ソフトウェア
インデックス	156	0
テーブル	3012	3012
メモリ使用量	284	128

※単位KByte



()内はByte数

図-9 登録データのレコード構造

Fig.9 Structure of registered data "records"

②性能測定

以下の項目の処理時間の測定を行った。

1) 登録

テーブルに1500レコードのデータを一括登録し、レコード辺りの登録時間を測定。

2) 検索Ⅰ

Aタイプの最初のレコードに格納されているデータ、最後のレコードに格納されているデータの検索時間を先頭グループと末尾グループで測定。

3) 検索Ⅱ

Bタイプに格納されているデータで検索をかけ、結果が返ってくるまでの時間を全てのBタイプグループで測定。

4) 検索Ⅲ

Bタイプに格納されていないデータで検索をかけ、見付からないことを示すまでの時間を全てのBタイプグループで測定。

5) 検索Ⅳ

Cタイプの最初/最後/任意の10点のレコードに格納されているデータの検索時間を全てのCタイプグループで測定。

6) 削除

1500全レコードを一括削除し、レコード辺りの削除時間を測定。

4.3 性能測定結果

性能測定結果を表6に示す。この結果から本情報検索ソフトウェアは、比較用ソフトウェアよりも約100倍高速であることがわかった。結果を分析すると、両ソフトウェアとも総処理時間の内、ディスクアクセスに要する時間が大半を占めている。

情報検索ソフトウェアは、まずインデックスによる検索を実行し、その後検索結果が正しいかテーブルにアクセスし照合するという手法をとっているため、大幅にディスクアクセスが少なくなっている。このことが今回の高速化につながったと言える。計算量を少なくする検索方法は今までに良く研究されてきているが、HDDに記録されたデータの検索には、より少ない情報で検索を実行することが有効である。今後もこの点に重点を置き開発を行いたい。

今回開発したソフトウェアでは、

- ・登録/編集/削除/検索/検索結果取得の機能
 - ・登録件数1500件に対して使用メモリ284Kbyte
 - ・平均10msecの検索速度
- を実現し、目標仕様を達成することができた。

5 HDD-AVNタイトルデータ管理への応用

HDD-AVNタイトルデータ管理には、本情報検索ソフトウェアの検索エンジンにタイトルプレイリスト機能を追加し、検索対象件数を3000曲に拡張して開発を行った。HDD-AVNタイトルデータ管理の構成を次項図10に示す。以下にHDD-AVNタイトル管理の主な機能を紹介する。

1) プレイリスト作成機能

アルバムプレイリスト、アーティストプレイリスト等の各種のプレイリストが作成可能。

2) プレイリスト順序操作機能

プレイリストから選択したタイトルを、リストの先頭へ移動させ並び替えることが可能。プレイリスト自体の並びを替えることも可能。

3) 並び替え解除機能

並び替えたリストをソートし、デフォルト並びに戻すことが可能。

表-6 性能測定結果

Table 6 Results of performance measurement

	情報検索ソフトウェア				比較用ソフトウェア			
	総処理時間		ディスクアクセス時間		総処理時間		ディスクアクセス時間	
	平均	Max	平均	Max	平均	Max	平均	Max
登録	30	—	20 (6)	—	370	—	340 (752)	—
検索Ⅰ	10	10	10 (1)	10 (1)	780	1060	750 (1500)	1020 (1500)
検索Ⅱ	10	10	10 (1)	10 (1)	1420	1660	1070 (1500)	1340 (1500)
検索Ⅲ	10	10	10 (0)	10 (0)	1370	1660	1080 (1500)	1350 (1500)
検索Ⅳ	10	40	10 (2)	30 (2)	1160	1530	990 (1500)	1330 (1500)
削除	10	—	10 (1)	—	10	—	10 (1)	—

※単位 msec, 測定最小単位 10msec

※登録, 削除は1500件に対し一括測定のためMaxの表記なし

※ディスクアクセス時間の()内はアクセス回数

4) ユーザプレイリスト作成機能

ユーザがカスタマイズした独自のプレイリストを作成することが可能。

5) プレイリスト削除機能

選択した不要なタイトルをプレイリストから削除することが可能。

6) 曲情報取得機能

プレイリストから選択したタイトルの固有情報を取得することが可能。

7) ランダム取得機能

プレイリストに登録された曲情報をランダムに取得することが可能。

8) 曲情報変更機能

登録された曲のタイトル名やプレイリストの名称を変更することが可能。

9) 登録済み情報検索機能

すでに登録されている曲のトラック番号を検索することが可能。2重登録を防止することができる。

6

おわりに

今回開発した情報検索ソフトウェアは、オーディオ等のタイトルリストを対象としたものであるが、検索ソフトウェアの基盤技術として、他のアプリケーションにも応用できるものである。

今後は、E-mailや保存されたWebコンテンツなどに検索対象を拡大するために、文章中のキーワードを高速に検索する機能の追加や多種多様なメディアを同一ソフトウェアで扱うための仕組み作りを推進していくことで、さらなる汎用的なソフトウェアを築き上げていきたい。

参考文献

- 1) 滝沢誠著：RDBMS技術解説：株式会社ソフト・リサーチ・センター（1996）
- 2) Robert Sedgewick：ALGORITHMS, 2nd ed.：Addison-Wesley Publishing Company, Inc（1988）

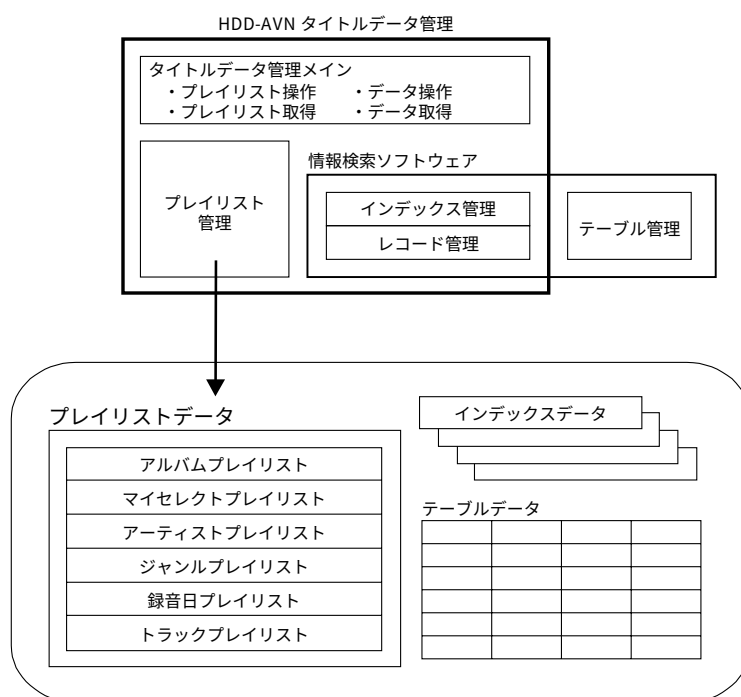


図-10 HDD-AVNタイトルデータ管理構成
Fig.10 Configuration of HDD-AVN title data management

筆者紹介



北尾 英樹
(きたお ひでき)

1992年入社。以来、騒音制御、音声処理などデジタル応用機器のソフトウェア開発に従事。現在、AVC本部システム開発統括部ソフトウェア開発部に在籍。



東山 雅彦
(ひがしやま まさひこ)

2001年入社。以来、情報検索ミドルウェアの開発に従事。現在、AVC本部システム開発統括部ソフトウェア開発部に在籍。



福田 晋児
(ふくだ しんじ)

1979年入社。以来、車載用情報機器の開発に従事。現在、AVC本部システム開発統括部ソフトウェア開発部チームリーダー。