

スペクトルデータベースの基本構造に関する提案

福島 整

科学技術庁 無機材質研究所

(1995年4月4日受理)

スペクトルデータベースの構造について、4レベルからなる階層構造をモデルとして提案した。検索しやすさ、維持管理のしやすさ、強靱さの3点が、データベースの構造に対する重要なポイントとしてあげられるが、階層構造はこれらの点を満足する一つの解と考えられる。実際に階層構造を持つデータベースを構築する上でのルールについて議論するとともに、考えうるいくつかの問題点の指摘を行った。

1. 序

近年のハードウェアの飛躍的な発展とネットワークの急速な普及は、それまで大変限られたユーザーを対象としてしか運用されていなかった様々なデータベース利用の門戸を大きく開く事となった。例えばChemical Abstractsに代表される科学技術分野の文献データベースは、すでにおよそ20年近く前からメインフレームレベルで広く運用されている。これは、近年のパーソナルコンピュータ(PC)のめざましい発達によって、CD ROM化されたデータベースとしての供給も行われている。また、ネットワークを通じた文献検索サービスを実際に行っている機関も存在する。経済分野では、様々な一般ニュースや経済情報のデータベース自体や検索サービスの提供が、専用・商用ネットをとわず日常的に利用される時代となった。インターネットの日本における普及は、これらの状況をますます加速している。

分光データにおけるデータベースも、その研究および利用の歴史はかなり古い。もっとも代表的なものは、JCPDS-International Center for Diffraction Dataによる粉末X線回折データ集であろう。以前は膨大なカード集と分厚いインデックスからなっていたが、現在ではCD ROMと検索システムによるPCのデータベースの利用が可能となっている。

日本ではSDBS(Spectral Data Base System)が広く普及している分光データ

ベースの代表的な例としてあげることが出来よう。これは、赤外吸収スペクトル、 H^1 -NMR、 C^{13} -NMR、および質量スペクトルを中心に、ESR、ラマンスペクトルを合わせた総合的な有機化合物のデータベースであり、実試料の測定データをもとにその試料に含まれる化合物の同定に大変有用である。このデータベースはネットワークでの利用サービスも提供しているが、やはりPCレベルでの利用が可能であることが実用性を高めている。

一方表面分析のデータベースでは、海外にはいくつかの例があり、最近ではSurface Science Spectra誌が刊行されるなど着々と活動実績が積み上げられている。これに対して、国内では、VAMAS-SCA-JAPAN委員会によるスペクトルデータバンク構想が進められていたのが公になっている唯一の例ではないかと思われる。この活動は、現在表面分析研究会(SASJ)の重要な活動として受け継がれている。

しかし、このスペクトルデータバンク構想の構築活動において、スペクトルデータベース自体の構造についての議論は、ほとんど重要視されて来なかった様に思われる。この原因は、過去の活動が表面分析の定量精度の向上という主題を中心になされており、スペクトルデータバンクはその為のデータプールで役割を十分に果たしていたからではないかと考えられる。

現在、スペクトルデータベース構築そのものが主目的となった活動が要求

され、また現に展開されている。その様な状況において、データベースの構造そのものに対する様々な議論もまた十分になされるべきであろう。ここでは、表面分析用に絞ったデータベース構造に対して、一つのモデルを提案する。

2. データベース仕様に対する 基本条件

データベースにおける「実用性」を定義するのは、たとえ使用目的が明確であっても大変困難である。これは、一つにユーザー個々の使い方に依存する部分が大いからである。しかし、「実用性」に必用な条件を予想し検討することは意味のある事であると考えられる。考えうる重要なものとしては、次の3点が挙げられよう。

2.1 検索しやすさ

データベースに対して、最も強く要求される点であろう。

必要とするデータをヒットするのに必用なプロセスと時間を減らした効率的な検索が可能か、熟練を要せずに検索が出来るか、という事である。

望みのデータをヒットするために、キーワードによる検索、データの照合による検索など、様々な検索法が準備されねばならない。扱うデータの性質やデータベースを扱う目的に応じて、様々な検索法が提案されており、現在も研究されている。しかし、具体的に準備される検索法のアルゴリズムが、検索目的だけでなくデータベースの構造にも大きく依存する事も、容易に想像がつく。

今のコンピュータの演算速度は極めて速くなっており、例えば最近のPCのCPUは10年前のメインフレームよりも格段に速い。このようなPCを用いたデータ処理でしばしば見られるのは、周辺記憶装置の使い方のまずさによる処理速度の低下である。

例えば、キーワードによる検索を行うときに、キーワード群を全てRAMにロードするかあるいはROM化したキーワードプールを用いるのであれば、処理速

度は検索アルゴリズムとCPUの早さで押しさえられることになる。しかし、周辺記憶装置にキーワードプールがいてあるのならば、メカニカルなアクセススピードが検索スピードを決める大きな要因となりうる。このメカニカルなアクセススピードを速くするには、装置のハードウェアに手が加えられないとしたら、その記憶媒体上での（要するにハードディスクやフロッピーディスク上での）データの並びに工夫をせざるを得ない。このデータの並びとは、どの様なパラメータを検索データに用いるか、それらのパラメータの論理的構造、記憶媒体上の物理的な配列の3つで決められる。

このうちの論理的構造と記憶媒体上の配列を決定する上で、データベースの構造が重要となる。

2.2 維持管理の容易さ

データベースに含まれるデータは、常に更新される。要するに、つけ加えられ、削除され、その作業が繰り返されるのである。

例えば、登録する必用のあるスペクトルは常に存在する。また、先に登録されていたスペクトルに問題が生じた場合には、容易に削除できなければならない（削除せずに使用しないだけにするという方針ももちろんありうる）。

また、検食用キーワードの追加や削除の要求は常に存在しうる。

削除のしやすさの必要性は、訂正だけにとどまらない。データが増えて巨大なデータベースになった場合、単に記憶媒体容量の問題だけからでも、データプールの分割や整理の要求の出現が予想できる。これに対応するには、効率良い削除機能が準備されねばならない。

構築された当時の目的が、そのまま永遠に変わらないという保証もない。したがって、データベースがカバーする領域・分野に変更が要求される事が十分ありうる。このような事態にも対応出来る構造が、求められる条件の一つと考えられる。

例えば、表面分析用のスペクトルデ

ータベースを構築したとする。最初の目的は、較正されたスペクトルとの比較検討であったとすると、それに応じたパラメータやデータ自体の並びが決まってデータベースの構造が決められ、構築される。それに対して、校正や比較の仕方に色々工夫がなされてくると、構築当初に考えられていた校正法や比較法に対応したデータベース構造では不便になってくる可能性がある。必要なスペクトルデータをヒットするのに、当初はキーワードによる検索だけで良かったのが、例えばスペクトル同士の直接比較（いわゆるサーチマッチ。相互相関を取るなどの方法が用いられる）の必要性が出た場合を考えると、構造があまりにリジッドに組まれていては対応が不可能となりかねない。

従来データを活用しながら、さらにデブスプロファイルのデータも加えたい、マッピングの為のデータも加えたい、といった要求が出される場合も考えられる。この様な場合に対応しようとする場合、新しいカテゴリーを比較的容易にそのシステムに組み込めるような構造を持ったデータベースであれば柔軟に対応できる。

以上の様な要求に対応するには、データベースの構造がより単純であればあるほど良い事は誰でも理解出来ることである。しかし、あまりに単純であると操作性が悪くなり、2.1と相いれなくなる。ここに、十分に検討されたデータベース構造が要求される一つの必然性がある。

2.3 強靭性

様々なアクシデントによるデータベースへの影響についても、十分に予測し対応しておくことが必要である。ハードウェアの故障、ソフトウェアの欠陥、操作ミス（おそらくこれが最も多いであろう）、停電、消耗等々、様々なアクシデントが考えられる。これらの影響は、データベースに含まれる様々なデータの部分的な欠落の発生である。

この問題に対して、最も単純で最も

効果的なのは、バックアップである。データベース全体を、記録媒体自体をメカニカルな機構から切り離せるもの（例えば光磁気ディスク等）へバックアップし、事故が起きたら再インストールするのが、最も単純かつ確実ではある。

しかし、データベース自体が巨大なものになった時、この方法しか準備されていないのでは大変能率が悪い。あるレベルまでの損害は自己修復が可能な構造をとっていれば、データベースが巨大化しても比較的楽に復元作業が出来て、維持管理の効率化を計ることができる。

自己修復機能を持たせるには、データベースに含まれているデータの内で重要なものが、効果的に分散されて複数記憶されている事が必要である。すなわち、強靭性を持たせるには、データの冗長度をどのように効率的に持たせるかということが重要になる。

また、障害が生じた時、それによる損害が全体に広がらないようにする構造を持たせる事も重要である。

これらの点は、データベースが小さいときでもシステムが少々大げさになり、かえって操作性を悪くしている原因と見えることもある。特に、冗長度を持たせる事により必然的に記憶容量が増えるので、扱われるデータ量が少ない場合にはかえって非効率的である場合ももちろん存在する。したがって、データ量が少ないままで良いデータベースと、巨大化することが十分予測出来るデータベースとでは、採るべき方針は異ならねばならない。

3. 階層構造を持ったデータベース

前節で議論した3点を満足させる一つの案として、階層構造を持ったデータベースのモデルを提出する事ができる。

階層構造の有利さは、なんと言ってもその強靭性である。あるレベルのデータが常にそのすぐ下のレベルによって再構成でき、そのレベルのデータはそのすぐ下のデータしか支配していない様なデータ構成を取らせることで、

障害に対する被害の伝搬を出来るだけ
くい止めることが可能になるとともに、
最下層のレベルの損害以外は自己修復
を可能にしてくれる。

また、最下層以外のどの階層でも検
索出来るように設定することが可能で
ある。したがって、任意の層から最下
層までを常に独立したデータベースと
して用いる事も可能となる。これは、
データベースの結合や分割が可能であ
る構造であることを意味する。また、
デンドログラム的な構造とみなせるた
め、ある程度の検索性の良さも保証で
きる。

もちろん、最下層以外の各階層にお
ける検索用データ群の配列にも、必要
に応じた論理構造や相互の関連づけを
持たせることが可能である。したがっ
て、リレーショナルな検索性を与える
ことももちろん可能である。

また、階層構造とデータの分類自体
を対応させることで、データ個々の追
加・削除だけでなく、新たな目的に対
する改造にも対応させられる。さら
に、管理の自動化を可能にする構造の
一つでもある。管理の自動化がはかれ
れば、スペクトルデータ個々に対する
検索用識別名（例えばファイル名）の
問題は、必然的に消滅する。

4. 階層構造を構築する為のルール
階層構造をとるデータベースを構築
する場合のルールについて、表面分析
用スペクトルデータベースを想定して
説明する。なお、以下の説明中にファ
イルという用語が用いられているが、
これは「名前の付いたデータセット」
という程度の抽象的なものを示してお
り、「記憶媒体上の名前の付いた特定
の領域」という元々のよく用いられる
意味に限らない。しかし、適当な用語
が見あたらないことと、元々の意味で
の「ファイル」を想像した方が全体の
把握も楽ではないかと考えて、あえて
用いた。したがって、実際の構築を行
う際に、記憶媒体上にそのとおりの
（元々の意味での）ファイルが確保さ
れねばならない必要は無い。

ルール 1

「スペクトルデータベースは、4レ
ベルの階層構造をとる。」

図1に、その概念を図示する。

レベル（階層）が絶対に4つである
必要は無い。十分に限定されたデータ
ベース（例えばAESの絶対スペクトルだ
けのデータベース等）であるのならば、
3レベルでおそらく十分である。

レベルが4つ必要であるとしたのは、
次の様な考え方に基づく。

すなわち、最も下の階層には、スペ
クトルデータ本体が格納される。その
上の階層は、それらの検索用データの
並びが格納される。その上の階層は、
検索データのレベルで支配されている
データ群の数やシリアルナンバーを管
理するデータを格納する。このデータ
群は、様々なカテゴリーに分類される
場合があり（AESとXPS、あるいは
絶対スペクトルとその他のスペクトル、
通常のスキャンかデブスプロファ
イリングのデータか、...等々）、そ
のカテゴリーの管理をさらに上のレ
ベルで行う事を考えれば、結果的に4
階層の構造となる。カテゴリー分類を
考えなければ3階層で十分であり、カ
テゴリーにも階層を考えるのであれば
4階層以上の構造を取ることになる。

この方法の便利な点の一つは、いく
つかの別々に構築されたデータベース
を一つにまとめたり、あるいは一つの
データベースを分割したり、カテゴリー
の細分化や階層化にも対応できる
ところである。

また、隣り合った二つの階層だけを
抜き出して、それをそのままデータベ
ースとして用いることすら可能な仕様
を作りうる。

4.1 4レベルの階層構造

ルール 1-1

「あるレベルのデータでは、その一
つ下のファイルあるいはデータの並び
のみを指定できる。これにより、追加
・削除・単独利用に対するフレキシビ
リティを持たせる事ができる。」

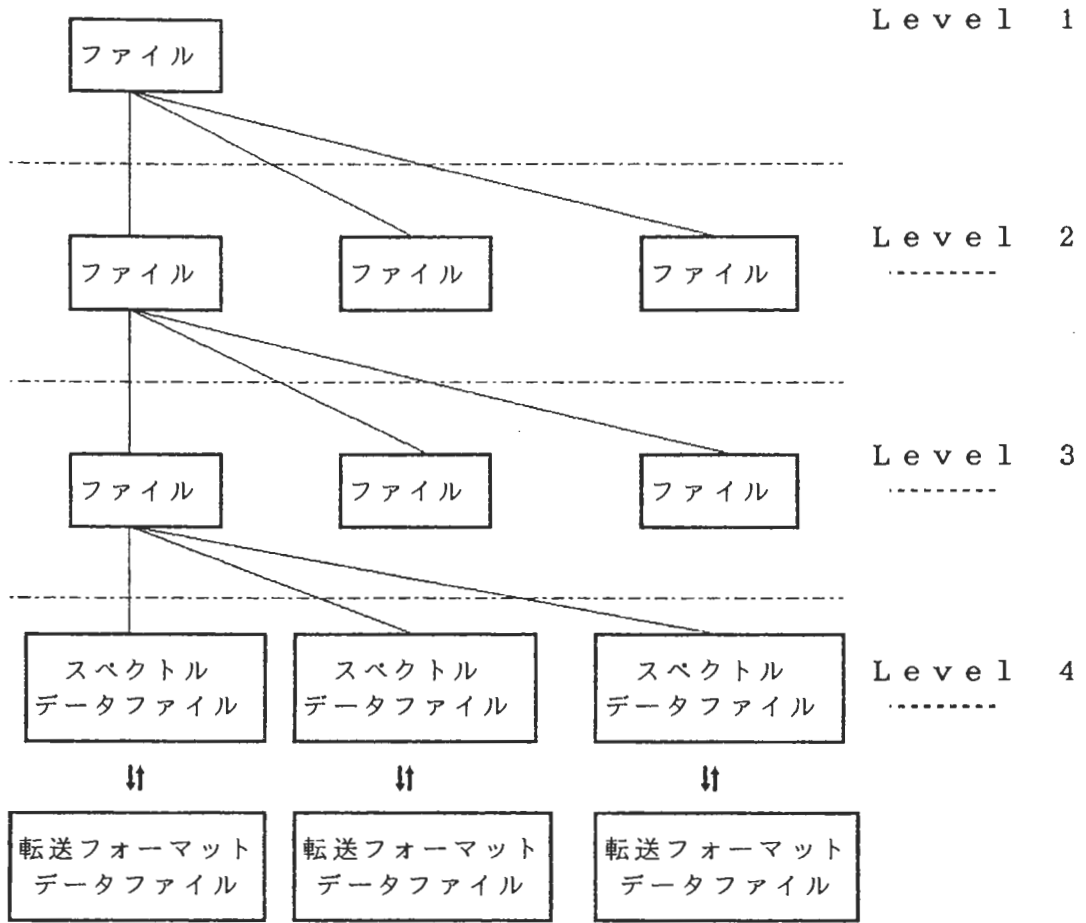


図1 4階層構造を取るデータベースのモデルの概観

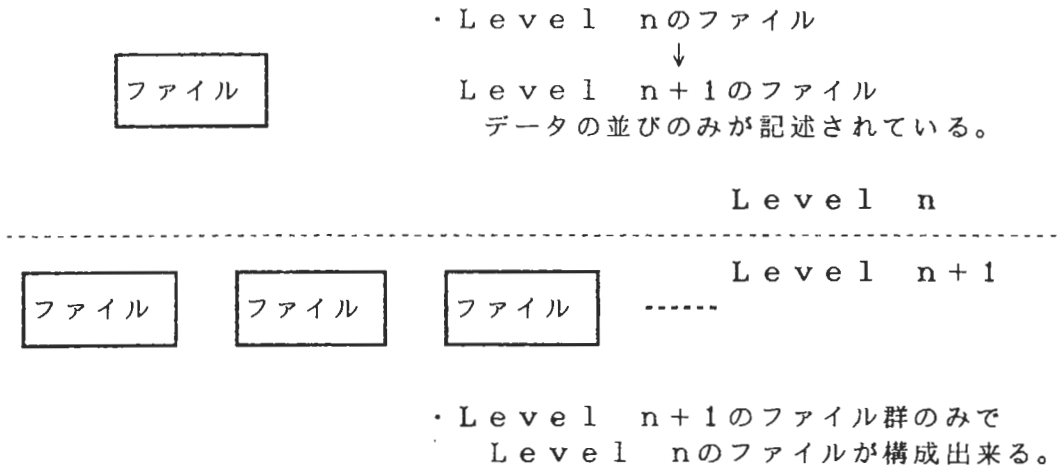


図2 隣り合った階層のファイル同士に対するルール

これは、あるレベルのデータに対応しているのは、そのすぐ下の階層のファイルやデータだということである。先の説明から、最下層にスペクトルデータ群が格納され、その一つ上のレベルに検索データ群が格納され、それより上の階層は単にデータやファイル数（あるいはシリアルナンバー等）の「数の管理」を行っているだけである。したがって、あるレベルのデータの追加や削除が行われた場合、そのすぐ上の階層の対応するファイルの中身が書き換えられるだけで対応できる事になる。

ルール 1-2

「あるレベルのみのデータおよびファイルから、その一つ上のレベルが構築できる。これにより、冗長度を上げることができ強靱さを持たせる事が出来る。」

これは、2-3で説明した事そのままである。

1-1,-2の概念をまとめると、図2の様に書ける。これが、この構造を構築するための基本的なルールとなる。

以下、各レベルの説明に入る。

4.2 第1レベル

ルール 2

「第1レベルは、1つのファイルか

らなる。」

カテゴリ自体が階層構造を取るとは、考慮されていない。したがって、ファイルは一つだけでよい。

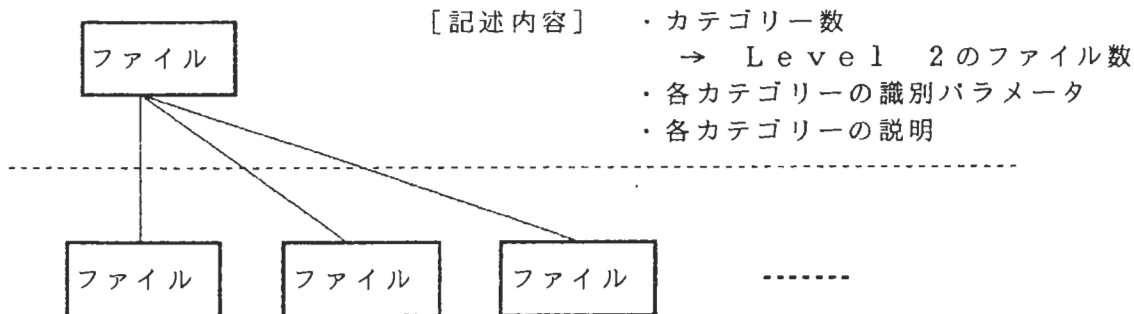
ルール 2-1

「このファイルには、データベース上のスペクトルのカテゴリ数、それぞれのカテゴリの識別子、それぞれの識別子に対応したコメントのみが記述される。識別子によって、特定のディレクトリを指定しても良い。」

本来ならば、ここでカテゴリについての議論もあわせてなされるべきであろう。しかし、それは構造と独立した議論が可能であるため、ここでは触れないことにする。

データベースを構築する上では、このカテゴリ分け（データの大きな分類）は極めて重要である。というのは、検索性能を考えると、複数のカテゴリにわたる様な検索の必要性が生じない様な分類が望ましいからである。したがって、同じデータが、複数のカテゴリに同時に存在することもあり得るであろう。検索は、カテゴリに対応したデンドログラムを後戻りせずに一方向にたどると同じ様な作業が最も効率が良いと考えられる。その機能

Level 1 - ファイル1つ



→ Level 2の各ファイルを用いて、Level 1のファイルが再構成出来る。

図3 Level 1のファイルの性質

を、すでにこの段階でもたせてしまうわけである。

第1レベルでは、このカテゴリーの数が指定される。

また、個々のカテゴリーを管理しているファイル（第2レベルに存在している）が指定出来る様な機能が持たされる。それが、「カテゴリーの識別子」の意味である。また、それぞれのカテゴリーの説明が記述されていれば親切であろうし、説明をキーワード化するか関連のキーワードを添付することで、カテゴリーに対する検索性を持たせる事も可能となる。

第1レベルの内容が、第2レベルのファイル名と数から再構成出来るのも、明らかであろう。コメントの再構成性については、第2レベル以下のキーワードの並べ方や扱いにルールをもうければ、ソフト的に対応可能となる。

第1レベルの概念を、図3にまとめる。

4.3 第2レベル ルール 3

「第2レベルは、第1レベルの識別子で特定できる一つのファイル、あるいは第1レベルの識別子で特定されるディレクトリ中の一つのファイルからなる。」

これは、構築のルールそのままである。第1レベルの識別子一つに対して、このレベルのファイルがユニークに指定されていなければならない（1対1対応である事が必要である）。

ルール 3-1

「このファイルには、データベースに収録されているそのカテゴリーに属する全スペクトルデータ数、登録されているそのカテゴリーに属するスペクトルの通し番号の最後の番号、およびそのカテゴリーに属するスペクトルデータベース全体に対するコメントのみが記述される。」

このレベルは、これより下の二つのレベルのデータの自動管理を担当する。

そのカテゴリーに含まれているスペクトルデータの総数、およびシリアルナンバーの最後（最新）の番号が記録されていれば、データの追加あるいは削除作業を効率よく行うことが出来る。

ここで考えているデータベースでは、検索結果に対応してして表示される項目としてスペクトルデータのシリアルナンバーを考えている。したがって、シリアルナンバーは、登録時にマニュアルで付けられるのではなく、システムによって自動的に付けられ、自動的に管理されるようなシステムが望ましいと考えられる。

また、このファイルに記述されるこのカテゴリーのコメントは、第1レベルのファイル中に記述されているものと対応づけられるべきである。

また、原案には記述されていないが、第3レベルのファイルが複数存在する場合も想定するならば、その指定もこのファイル中でなされる必要がある。

第2レベルの概念を、図4にまとめる。

4.3 第3レベル ルール 4

「第3レベルは、そのカテゴリーに属する全てのスペクトルの検索用データ列で構成される。」

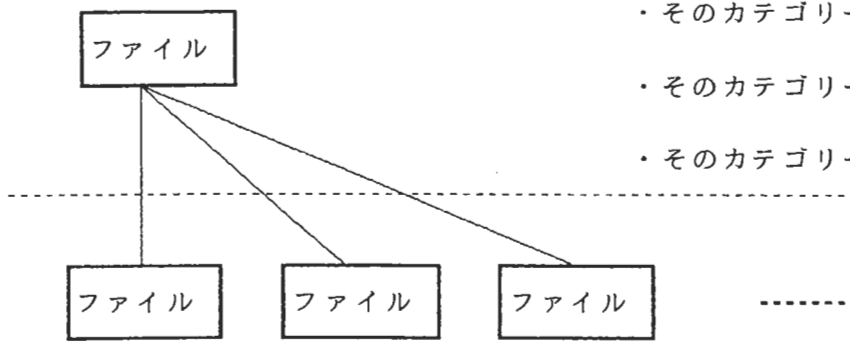
このモデルでは、全てのデータ検索はこのレベルで行われる事を想定している。検索用データ列とは、検索用ルーチンがアクセスして検索プロセスを実行するためのパラメータ群の事である。この検索用データは、一つ一つがスペクトルデータと対応付けがなされていなければならない。また、このデータの並び自体がある論理構造を取っていても良いし、互いに関係付けがなされていても良い。したがって、もちろんリレーショナルな機能を持たせることも十分可能である。

ルール 4-1

「このレベルは、単一のファイルであっても、複数のファイルあるいは複数のファイルを含むディレクトリから

- Level 2 - 各カテゴリ毎にファイル1つ
- Level 1のファイル中でユニークに指定

- [記述内容]
- ・そのカテゴリ中の全データ数
 - ・そのカテゴリのデータに付けられた通し番号の最後の番号
 - ・そのカテゴリの検索ファイル指定のためのパラメータ
 - ・そのカテゴリの説明

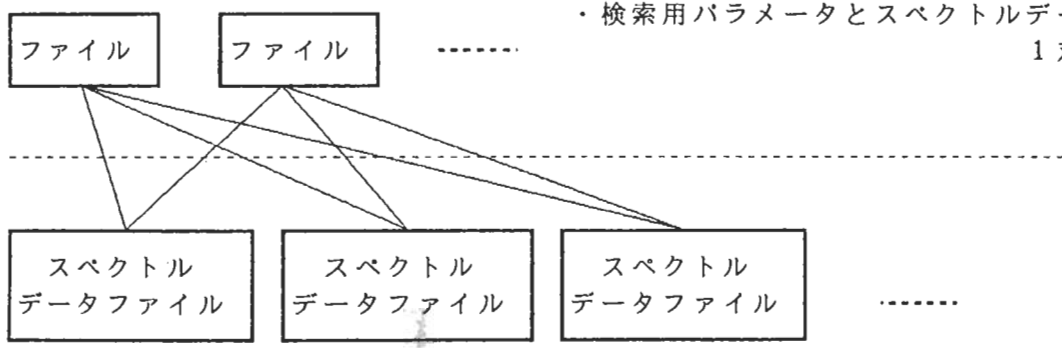


→ Level 3のファイル群を用いて、Level 2の対応するファイルが再構成出来る。

図4 Level 2のファイルの性質

- Level 3 - そのカテゴリの検索用ファイル(群)
- Level 2のファイル中で直接ユニークに指定

- [記述内容]
- ・そのカテゴリ中の全データ数の検索用パラメータファイル
 - ・検索用パラメータとスペクトルデータが1対1に対応



→ Level 4のスペクトルデータファイル群を用いて、Level 3の対応する検索用ファイルが再構成出来る。

図5 Level 3のファイルの性質

なってもかまわないし、第2レベルのファイルと同一ディレクトリでもかまわない。ただし、第2レベルのファイルから直接指定されねばならない。」

検索用データの並びが格納されているレベルであるので、場合によっては単一の領域とはならない可能性もある。その場合は、そのカテゴリーに含まれる全ての検索用ファイルが、第2レベルからユニークに指定されている必要がある。この事は、検索用データ列に持たせる事の出来る論理構造や互いの関係付けとは、切り離れた議論が可能である。

ルール 4-2

「このレベルにおけるファイル（あるいはファイル群）の中のデータのならばは、検索用ルーチンによって決定される。」

この点についての解説は不要であろう。ただし、データの並びの中に、スペクトルデータのシリアルナンバーも全て含まれている事、および検索データの並びからそのカテゴリーに含まれるスペクトルデータ総数が常に判別できる事の2点を満足する仕様でなければならぬ。

第3レベルの概念を、図5にまとめる。

4.5 第4レベル

ルール 5

「第4レベルは、そのカテゴリーに属するスペクトルを格納したファイル群からなる。」

転送されたスペクトルデータは、全てここに格納される。

ルール 5-1

「このレベルは、第2、第3レベルのファイル群と同一のディレクトリに存在してもよいし、第3レベルのファイル中のデータによって特定されるのであれば、スペクトルを格納したファイルが複数のディレクトリにわたって

いてもかまわない。しかし、記録されているスペクトルのファイルが、異なったカテゴリーに属する第3レベルのファイル中で重複して特定されてはならない。」

これは要するに、そのカテゴリーに属するスペクトルデータが、同じカテゴリーに属する第3レベルのデータからのみユニークに指定されるという事である。したがって、もしあるスペクトルデータが複数のカテゴリーに同時に分類される必要性を持っているのであれば、全く同じスペクトルデータがそれぞれのカテゴリーの第4レベルに存在することになる（同じ内容の異なったスペクトルデータファイルが存在することになる）。これによって、無用な構造の複雑化を避けることが出来、検索性を向上させられると考えられる。

ルール 5-2

「スペクトルのファイルは、原則として一つの実験からなる。一つの実験によるスペクトルとは、一回の測定で得られたスペクトル（単独、複数）、一回のデブスプロファイリングで得られたスペクトル群、一回のイメージングで得られたスペクトル群を意味する。」

このルールを構築ルールとして含めるか否かは、議論の分かれるところであろう。また、データベース構造そのものの議論からは若干離れており、分離した議論が可能であるものと思われる。むしろ、検索システムをどう構築するかという問題と会わせた議論をすべきかもしれない。

ルール 5-3

「スペクトルのファイルは、その中にレベル3の検索用データを全て含んでいなければならない。」

これは、このデータベースの冗長度を稼ぐ基礎となる部分である。スペクトルデータファイル個々に、そのデータに対応する検索用データをセットと

- Level 4 - そのカテゴリのスペクトルデータファイル（群）
 - Level 3のファイル中で直接ユニークに指定
 - 別のカテゴリのLevel 3のファイルからは
 指定されない

〔記述内容〕 ・ そのカテゴリに分類されたスペクトルデータ
 ・ 検索用パラメータに変換可能なデータ

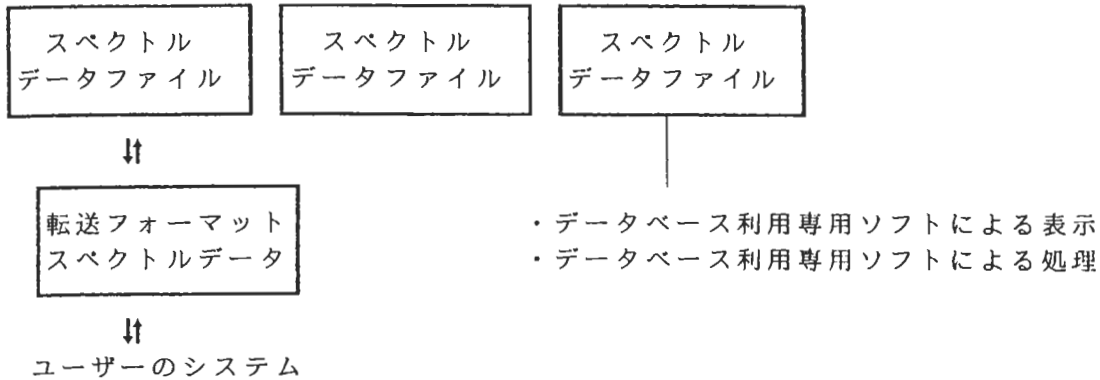


図6 Level 4のファイルの性質

して記憶しておく。そうすると、第4レベルのデータが存在していれば、第3レベルのデータが再構築出来る。第3レベルのデータが再構築出来れば、第2レベルのデータが、. . . というようにして、自己修復が可能なシステムとすることが出来るわけである。

ルール 5-4

「スペクトルのファイルは、それ単独で転送用フォーマットに変換が可能でなければならない。」

要するに、第4レベルのスペクトルデータファイルと転送用フォーマットを持つファイルとの相互変換が可能であることの保証である。

これは、2つの重要な点を含んでいる。一つは、第4レベルのデータファイルが存在すればデータベースが再構築できるという自己修復性をデータベースの外部で保証できるという事である。もう一つは、データベース自体が外部とのデータのやりとりを行うときのデータ形式を指定している事である。

しかし、第4レベルのデータファイルが転送フォーマットで書かれたものである必要は、かならずしも無い。むしろそれは、検索性や効率良い記憶媒体の利用の観点からは不利な面の方が大きい。

従って、データの外部との入出力に関して専用のルーチンを必要とすることとなるのはやむをえない。

ルール 5-5

「スペクトルのファイルは、それ単独でも利用が可能であるようなデータのならばと記録形式を持っているべきである。」

この仕様は、あると便利ではあろう。しかし、階層構造のデータベースに対して本質的に必要なものではない。

第4レベルの概念を、図6に示す。

5. データベースと、それを使用するためのソフト

データベースとは、基本的にはある目的の為に構造を工夫したデータブ

ルとも考えられる。したがって、それを使用し、あるいは整備・維持するためのソフトウェアをも含めるかどうかは、議論の分かれる点であると考えられる。

例えば、あるルールに従って名前を付けた転送フォーマットで記述されたテキスト形式のスペクトルデータファイル群で構成されているデータプールをそのまま利用するデータベースとそれを使用するソフトが、PCレベルでの利用を主眼としたシステムとして存在しうる。この様なシステムは、ルール5-4でも触れたが、規模の大きなデータベースになってくるとデメリットが大きくなっていく恐れがある。その第一は、検索性の悪化、特にスピードダウンであろう。

したがって、データベース内のスペクトルデータは、あるルールに従った（必ずしも転送フォーマットとは同一ではないが相互に変換の可能な形の）書式で記述されたファイルに変換される必要がある。この書式は、データベース内での高速な検索と高速な表示に最も適した仕様が与えられるべきである。

すなわち、検索とその結果の表示は専用のソフトを準備すべきである。これは、サーバーに付随した形で利用を考えたとしても良いし、端末として利用されるPCにインストールされているソフトウェアの機能としてつけ加えても

〔査読者との質疑応答〕

吉原： 現在、表面分析研究会で計画しているスペクトルデータベースはMaster Indexと呼ばれる検索項目からなる「ファイル」と、転送されたスペクトルデータを集めた「ファイル」からなっております。この構造は2階層構造というもので、執筆者が推薦する4層構造とはなっていません。なぜならば現在のところ、集めているスペクトルデータには、今の所デプスプロファイルやマッピングデータはないからで、

良いであろう。その一方で、必要とするスペクトルデータの選択が終了し、そのデータを自分のシステムに転送する場合には、転送フォーマットの形でデータベースから取り出せる仕様とすべきである。

6. 終わりに

以上、階層構造をとるスペクトルデータベースの一つのモデルを提案するとともに、関連したいくつかの問題点について問題提起を行った。

情報工学の分野を専門としなくても、近年の表面分析を含むほとんどの分光分析の分野が情報工学や応用数学のセンスを求められるようになってきた。これは、10年前には分光データの数値処理の分野が分析化学の研究としてなかなか認められなかった事を思い起こすと、隔世の感を否めない。データベースに関する議論においても、また然りである。

この階層モデルの大部分は、かつて小規模ながらもPCレベルで構築したXPS用のデータベースの発展型であり、実際に試用した結果を基にしている事も、附記しておく。

参考文献

- 1) 福島 整： *J. of Surf. Anal.*, 1, 61 (1995)

Master Indexをさらにカテゴリー分けをして深い階層構造を作ることは計画されていません。ただし、これはここ2年間の事で、将来は当然拡大することを考えています。この様な2階層構造で出発したデータベースを将来、4層構造に改造することは可能でしょうか。

著者： 基本的には、2階層構造データを多階層構造に取り込んでいくことは可能です。要は、2階層構造のデータベース自体をそれより上のレベルで指定出来るようにするだけですから。

階層構造の基本はスペクトルデータのアイデンティフィケーション（要するに、データに与えるシリアルナンバーのシステム等の事）をどう管理するか、という点にあります。したがって、場合によってはこの点に関して、多階層構造へ組み込む作業においてMaster Index中の検索データの並びに手をくわえる必要があるかもしれません。

もっとも、構築に試用されたデータベース用ソフトウェアの仕様をそのまま生かせるかどうかは、そのソフトウェア自体の持つ仕様と構築時のデータベースの構造に依存しますから、なんとも申し上げかねます。

吉原： ルール5-3ではスペクトルのファイルと第3レベルのデータファイルの関係について述べておられます。ここでは、スペクトルデータファイルに第3レベルの情報は全て入れておくと記述されておられますが、むしろ第3レベルの情報は全てスペクトルデータファイルより構築されるという記述にされる方が現状に近いし、またそうでないと第3レベルのデータファイルは作成できないのではないのでしょうか。

著者： 主張したかった事は、スペクトルデータファイルに含まれている検索用（に用いることの出来る）データ群の一部もしくは全部を用いることで、第3レベルのファイルが出来ると言うことです。さらに、第3レベルのファイルを構築するとき、スペクトルデータファイル以外から検索用パラメータを補充してはならないという点を強調したいが為にこの様な表現をとりました。

多分、ご指摘の表現の方が誤解が少ないと思います。

関根： 「記憶媒体上でのデータの並びに工夫．．．」とあるのは、物理的な並びではなく、整理・分類上の並びということですね？

著者： もちろん整理・分類上の並びもそうですが、物理上の並びも重要になります。

査読者のご質問もあって本文に少々加

筆致しましたが、このデータの並びとは、どの様なパラメータを検索データに用いるか、それらのパラメータの論理的構造、記憶媒体上の物理的な配列の3つで決められます。

データベースの様な議論の場合、記号論理学と同様、実用上の意味や内容とは切り離れた入れ物だけの論理構造やそれを実現するのに適した物理構造が議論出来るはずで、すなわち、構造を先に決めて、それにあわせて整理・分類を行う事が可能だという考え方は、

関根： 「検索に多様性を持たせる．．．」は例えどんな時必用となりますか。

著者： ご指摘の部分は、再度検討したところ論理的にあいまいな記述であったため、本文の様に訂正させていただきました。

当初考えていた多様性とは、カテゴリーや分類、スペクトルを得るための実験の種類などの検索項目に対する検索や、スペクトルに付随して取り込まれる情報に対する検索等を考えておりました。これらは、最下層における必用のない検索仕様が可能であるからです。

また、スペクトルデータが子構造を持っている場合においても（例えば校正用スペクトルとセットになっている、あるいは一連の時系列的变化を追った実験のスペクトルセットなど）、この検索データの構造を工夫事で可能になります。

関根： 第3レベルについて、検索用データ列のイメージをもっと具体的に与えて下さい。それは、リレーショナルな検索性を具備することを説明して下さい。（前の方の記述で）最下層（4層）とその一つ上のレベル（3層）がデータとありますが、ここでは第3層は検索用のデータ列としています。一つの実験が複数のデータからなるとき、どの様になりますか？

著者： まず、ご指摘の点を考慮し、

第3レベルのルールの説明と一緒に「検索用データ列」に関する説明を加筆致しました。また、第3層と第4層に含まれるデータの相互の関連は、先の査読者のご質問と同じ内容と思われるので、それに対する回答をご参照下さい。

一つの実験が複数のスペクトルからなる場合の対処は2通り考えられます。

一つは、第4レベルのデータ構造の定義で対処する場合です。すなわち、ルール5-2で示した部分です。しかし、これはかならずしも必用なルールではありません。むしろ、この定義を厳しくしすぎると構造的にリジットになり、利用の自由度が落ちる可能性があります。利用の自由度を確保するには、利用するデータは出来る限り単独で引き出す事が可能になるような構造にすべきでして、なるべくセットは作らないようにすべきであると考えられます。

したがって、検索データ列の論理構造や相互の関連づけに工夫を与えることで対処するのが、より実用性が高いと考えられます。例えば、NPLフォーマットですと、マルチリジョン、マルチブロックの構造、実験パラメータの種類の数で、含まれている個々のスペクトルを識別できます。したがって、ある3次元の配列を考え、それにそれぞれのスペクトルを対応させることができます。そしてこの3次元の配列の

各要素を、検索用ファイルの検索項目と対応させておくことで、連れ子構造や一続きの実験のデータセットの関連づけを検索データ列の構造に取り込む事が可能です。またこの様にしておけば、一続きのデータのうちの一部分を抜き出して使用することも簡単になります。

関根：（図5のレベル3とレベル4の対応付けに関して）多重指定は複雑さのたこつぼに落ち込むのでは...!?

著者： 図中の説明に、「検索用パラメータとスペクトルデータが1対1に対応」と記述してありまして、誤解は無いと思ったのですが...。

関根：（図6のスペクトルデータファイルに対して）初めはVAMAS+ISOフォーマットを使えないか？

著者： 将来サーチマッチによる検索まで考えるのならば、少なくともデータはサーバーのハードウェアに従ったバイナリの方が扱いやすいと考えられます。したがって、構築当初から出来れば転送フォーマットとは別の専用仕様を考えるべきでしょう。

しかし、個人が必用なスペクトルデータを抜き出した上でPCベースで構築するサイズの小さなデータベースであれば、転送フォーマットベースでも実用上は問題ないと考えます。