

技術報告

テキスト形式フォーマットの統一的扱い方

城昌利

電子技術総合研究所

茨城県つくば市梅園1-1-4

jo@etl.go.jp

ISO提案フォーマット、その他、スペクトルを表現するためのフォーマットを統一的に扱う方法を提案する。この方法は、実際には、現在使われているフォーマットを変更することなしに、新しく「フォーマット記述ファイル」を定義し、プログラムは、これを通して元のフォーマットを扱う。これにより、フォーマットの拡張、変更が簡単になり、今まで扱えなかった情報も取り扱えるようになる。

1. はじめに

ISO/TC201/SC3委員会によって提案されたフォーマット<sup>1)</sup> (Partially encoded version of "ISO" format proposed in Common Data Processing Version 3.1, 以下ISO提案フォーマットと呼ぶ) は現在スペクトルを記述する際の標準フォーマットに事実上最も近いものであるが、ヘッダー部分の項目の選択・並ぶ順序に一貫性がない、ほかの同じような形式のものと互換性がない、フォーマットが変更されるたびに処理ルーチンを書き直さなければならない、あつかう項目の数が100個程度に限定されている、など、依然として多くの問題を抱えている。とはいえ、このフォーマットのもとになるVAMASフォーマットが提案された当時の、貧弱な計算機(PC)の処理速度、記憶容量、表示能力、使い勝手、開発環境を考えれば、それはやむを得ないことであった。この点は今や、計算機のソフト・ハード両面の劇的な進歩によって、処理能力の制限は取り払われ、高性能かつ使いやすいプログラミングツールも登場し、もっと便利に、膨大かつ詳細な情報を扱う可能性が開かれた。しかしながら、一度普及したフォーマットの変更は、多大な混乱を招くからなかなかできないのも現実である。

ここでは、このジレンマの解決策を提案する。具体的には、現行のフォーマットは一切変更せず、その代わりに、テキスト形式の任意のフォーマットを扱うためのプログラム側の対応指針を与える。この方法によると、個々のフォーマットは、ある一般化フォーマットの特別の場合と見なせる。またこの方法はきわめて柔軟であり、文字列で表せなくとも計算機が扱えるような情報(たとえば画像などの膨大なものも可)なら何でもこの枠組みの中に取り込むことができる。さらに、フォーマットの仕様が変更、拡張されたとき、プログラムの側の変更方法を系統的に与える指針となる。なお、本稿は、94年秋の

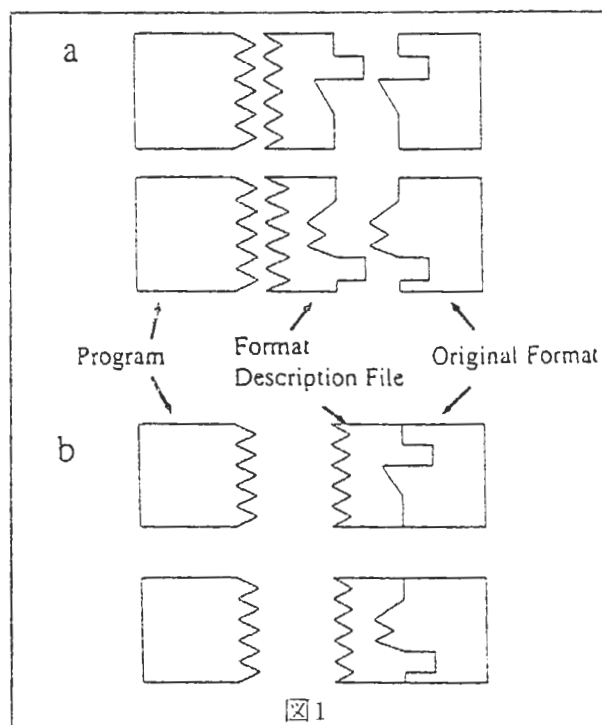


図1

表面分析研究会での発表内容に加筆し、同時に一部訂正を行ったものである。

2. フォーマット記述ファイル (Format Description File)

ここで対象とするファイルは、スペクトルデータの様々な情報を記述するフォーマットファイルで、テキスト形式で書かれ、各行に、それぞれひとつの情報が書かれているものとする。

- 項目 1
- 項目 2
- ...
- 項目 n
- ...

例えば ISO提案フォーマット(本誌 Vol.1, No.3 p.499 参照)では、  
項目 1 = format identifier : "VAMAS Surface Chemical Analysis ..."

**List 1.****Structure of Format Description File**

Header  
File Information Section  
General Information Section  
Spectra Information Section

**List 2.****Header Section of Format Description File**

L: Total number of Lines in this file.  
F: Total number of related Files.  
G: Number of General Information Blocks  
S: Number of Spectra Information Blocks

項目 8 = experiment mode = "NORM" である。

明らかに、このようなファイルは、項目の内容やその順番によって、無数に多くのバリエーションがある。ISO提案フォーマットはそのようなもののうちのひとつである。ここで考えているのは、スペクトルに関する情報を記述するためのフォーマットであるから、項目の種類は、どのファイルも似たようなものだと期待できる。すなわち入射電子（光子）のエネルギー、分析器のタイプ、試料名、日付、データ点の個数、などの項目は、どのフォーマットにも含まれているであろう。一方、特定のフォーマットにしか含まれない情報もあるだろう。そこで、現在少なくともどれかのフォーマットに含まれる項目をすべて拾い上げれば、後は、それらに割り振る順番だけを考えればよいことになる。順番が与えられないなら、その項目は含まれない。

順序が異なるフォーマットを統一的に記述する手続きは次のようになる。

1. 考え得るすべての項目に一意的な数（識別子, Descriptor）を対応させる。
2. 目的とするフォーマットファイルの各行に書かれている内容を記述する別のファイルを用意する。すなわち、内容を表す識別子が書かれたファイル（フォーマット記述ファイル）を次節のように構成する
3. 処理プログラムは、まずフォーマット記述ファイルを読み込み、目的のファイルフォーマットを理解してから、データを読み込む。

計算機上では、識別子として、Long(32bit)型を用いれば、 $2^{32} = 4,294,967,296$ 個の項目が区別できる。これは、スペクトルの情報を記述するには十分な数であり、項目数の制限は実質的にないに等しい。

**List 3.****Example of File Information Section of Format Description File**

Name of File #1 = Name of this file.  
Kind of File #1 = Descriptor of Format Description File  
Name of File #2 = Name of Format File  
Kind of File #2 = Descriptor of Format File  
...  
Name of File #n = "sample.jpg" (If this is a movie file.)  
Kind of File #n = Descriptor of JPEG file.  
...  
Name of File #F = Similar as above.  
Kind of File #F = Similar as above.

これは、言い換えると、フォーマット記述ファイルが、各フォーマットの違いを吸収し、プログラムからは、同じものに見えることになる。これを図示すると次のようになる。図1aでは、プログラムと、もとのフォーマットの間をフォーマット記述ファイルが橋渡ししている。これを、図1bのように書くと、元のフォーマットと、フォーマット記述ファイルをあわせたものが、実は一般化されたフォーマットになっていると解釈できる。

**3. 具体的構成法**

以下のようにフォーマット記述ファイルを書くと、先頭から順に読むことにより、次々にファイルの構造とサイズの情報が得られ全体が把握できる。個々のフォーマットファイルはいくつかのブロックに分かれ、各ブロックは、互いに関連するいくつかの項目を含む。これにより各ブロックの大きさを可変にでき、各部分のサイズが変わっても同じルーチンで読み書きできる。

全体は、大きく4つの部分からなる。(List 1)

**3.1 Header**

ファイルの先頭部分で、ブロックの数、行数が書かれている。(List 2)

全行数Lはこのファイルを読み込む文字列配列の大きさを決めるのに使われる。F、G、S は以下で内容を読み込むときに使われる。

### 3.2 File Information Section

関連するファイルは自分自身と、フォーマットファイルの少なくとも2つあるので、 $F \geq 2$ である。ここでは、さらに、ほかのファイルを含んでいてもよい。ほかのファイルとして、例えば次のようなものが考えられる。

- 試料処理の詳細を記録した画像（静止画または動画）ファイル。

- フォーマットに収まりきれないコメント、試料履歴、などを記したテキストファイル。

これらのファイルを用いることにより、今までは研究者のノウハウに属し、実際に見せるしかなかったものを相手に伝えられる可能性を持つ。

この記述ファイルに関連するF個のファイルの名前（文字列）と、そのファイルの表す意味・属性（整数の識別子）をこの部分に書く。(List 3)

プログラムは、これらのファイルが揃っているかどうかチェックできるから、複数のファイルを取りこぼしなく扱える。

### 3.3 General Information Section

一般情報のブロックには、実験の一般的なパラメータが書かれる。この例をList 4に挙げる。

このブロックの数がGである。

各ブロックの情報の種類と、サイズをList 5のように書く。

### 3.4 Spectra Information Section

スペクトルデータ自身は、ISO提案フォーマットと同様に、電子の運動エネルギーまたは、結合エネルギーの一定の増分（= abscissa increment）に対して信号強度が並んでいるとする。そうでない形式もあり得るが、本稿は、一般情報部分の記述法が主目的なのでとりあえず考えない。このようなスペクトル領域がいくつかあるとき、それぞれの領域に固有な情報を一般情報ブロックと同様に書く。領域に固有な情報とは、スキャン範囲の上限、下限、スキャン回数、エネルギー増分、などである。この領域

#### List 4. Examples of Block

```

block "Identifiers"
    format identifier
    institution identifier
    instrument model identifier
    operator identifier
    experiment identifier

block "Sample-Geometry"
    analysis width x (m)
    analysis width y (m)
    analysis axis take off polar angle (degree)
    analysis axis take off azimuth (degree)
    
```

#### List 5. General Information Section of Format Description File

```

Descriptor of 1st Block
LG1 : Number of lines of 1st block
    Descriptor of 1st line of 1st block
    Descriptor of 2nd line of 1st block
    ...
    Descriptor of LG1-th line of 1st block
Descriptor of 2nd block
LG2 : Number of lines of 2nd block
    Descriptor of 1st line of 2nd block
    Descriptor of 2nd line of 2nd block
    ...
    Descriptor of LG2-th line of 2nd block
...
Descriptor of G-th block
LGG : Number of lines of G-th block
    Descriptor of 1st line of G-th block
    Descriptor of 2nd line of G-th block
    ...
    Descriptor of LGG-th line of G-th block
    
```

の数がSである。(List 6)

### 4. 識別子の決め方。

識別子に対応する整数値は処理プログラムの側で設定する。以下、最もポピュラーで、comproの開発言語でもある、Visual Basic ver. 4の文法に則した例をList 7に示す。

値をとびとびにするのは、もちろん後の拡張を簡単にするためである。

### 5. 入出力法

**List 6. Spectra Information Section of Format Description File**

```

LS1 : Number of lines of 1st block
      Descriptor of 1st line of 1st block
      Descriptor of 2nd line of 1st block
      ...
      Descriptor of LS1-th line of 1st block
LS2 : Number of lines of 2nd block
      Descriptor of 1st line of 2nd block
      Descriptor of 2nd line of 2nd block
      ...
      Descriptor of LS2-th line of 2nd block
...
LSS : Number of lines of S-th block
      Descriptor of 1st line of S-th block
      Descriptor of 2nd line of S-th block
      ...
      Descriptor of LSS-th line of S-th block
    
```

**List 7.**

**Examples of definitions of constants for Descriptors in Visual Basic**

```

Public CONST DESCRIPTION_FILE = 101
Public CONST FORMAT_FILE = 102
Public CONST ADD_JPEG_FILE = 110
...
Public CONST BLOCK_IDENTIFIERS=10100
Public CONST INSTITUTION_IDENTIFIER = 10150
Public CONST BLOCK_GEOMETRY=10030
...
...
Public CONST SEE_JPEG_FILE =100010
    
```

このファイルを読むためのVisual BasicのコードはList 8のようになるだろう。

何か新しい項目を付け加えたとすると、新しい識別子を表す定数を定義し、Select文の中に、新しい処理する部分を付け加えればよい。このとき、他の部分を変更する必要はない。いちいちSelect文があって、煩雑だが、今のPCなら大して時間はかからないはずである。

**6. その他コメント**

ここでは、テキスト形式のファイルに限って説明したが、バイナリ形式でも項目が順番に並んでいるのなら拡張は簡単と思われる。実際は、

スペクトルは、ある特定のデータベース上のデータとして存在し、ある固有のフォーマットで格納されている。したがって、ここで述べた処理プログラムとは、単なるスタンドアローンのプログラムよりむしろこのデータベースアプリケーションを指す。このデータ構造を設計する際、この一般化フォーマットを考慮すると、異なったデータベース同士に互換性を持たせることができるのではないかと期待している。

**参考文献**

1) J. Surf. Anal. 1,495(1995).

**List 8.**

**Examples of Visual Basic Codes for File I/O Input**

```
Do While Not(EOF)
  LINE Input , #FileNumber, TempStrings$(I&)
  Select Case Val(TempStrings$(I&))
    Case INSTITUTION_IDENTIFIER
      register institution name
    Case ANALYSIS_WIDTH
      set analysis area
    ...
  Case Else
    ...
  End Select
  I&=I&+1
End Do
```

**output**

```
For I&=1 to L
  Print #FileNumber1, TempStrings$(I&)
  Select Case Val(TempStrings$(I&))
    Case INSTITUTION_IDENTIFIER
      save institution name
    ...
    Case ANALYSIS_WIDTH
      save analysis width
    ...
  Case Else
    ...
  End Select
Next I&
```

## **General ASCII format for spectrum data**

Masatoshi Jo

Electrotechnical Laboratory

Umezono 1-1-4, Tsukuba, Ibaraki 305, Japan

I present the general way of expanding spectra format file without changing the format itself. In order to do this, a format description file that can describe the structure and the contents of original format file written in ASCII format has been introduced. The change in the original format is absorbed in the format description file, hence the program need not be changed. This also makes possible to expand the ISO format [J. Surf. Anal. 1, 495 (1995)]. In addition, the present formalism has such a flexibility as can take almost all information that is treated on the computer into itself.

査読者との質疑応答

査読者 吉原一紘(金材研)

吉原 : 本論分はフォーマットの提案というよりは、むしろ、異なったフォーマットで書かれたスペクトルデータをいかに共通の形に変換するかというソフトウェアの提案と思われます。したがって、副題(ISO 上位互換フォーマット試案)を削除されることを勧めます。

著者 : ご指摘の通り、これはソフトウェア側への提案ですが、図1 bにあるように、ある、一般的なフォーマットの提案にもなっています。ただ、その「フォーマット」は、現行のような、形のはっきりしたものとは異なり、その具体的な

「形」は処理プログラムのコーディングに依存します。小論の意図は、現在検討されているISO フォーマット案に直接置き換わるものを提案することではなく、むしろ、今後それ自身やほかのフォーマットを効率よく利用し、また新たなフォーマットを設計する上での留意点について述べることでした。その意味で、副題は、誤解を与えるので削除しました。

吉原 : 記述の中に、「ISO フォーマット」という字句が見られますが、ISO では、まだフォーマットを決定していませんので、「ISO 提案フォーマット」という表現に変更されることを勧めます。

著者 : ご指摘に従い変更しました。

## Generalization of ASCII format for spectrum data

Masatoshi Jo

Electrotechnical Laboratory

Umezono 1-1-4, Tsukuba, Ibaraki 305, Japan

I present the general way of expanding spectra format file without changing the format itself. In order to do this, a format description file that can describe the structure and the contents of original format file written in ASCII format has been introduced. The change in the original format is absorbed in the format description file, hence the program need not be changed. This also makes possible to expand the format proposed as the ISO format by ISO/TC201/SC3 Japanese Domestic Committee [J. Surf. Anal. 1, 495 (1995)]. In addition, the present formalism has such a flexibility as can take almost all information that is treated on the computer into itself.