

## 5N-8

## I C O T o n e — 楽譜エディタ

†金井良太郎, ††斎藤隆文  
(†三菱重工業, ††東京大学工学部)

### 1. はじめに

計算機で自動演奏や作曲の支援システムなどを作成する場合、楽譜のデータをいかに入力するか、という問題がつねに生じる。これに関しては、印刷された楽譜からの読み取り、実際の演奏データ（打鍵のタイミングなど）からの採譜、さらには演奏された音からの採譜、などの方法も考えられるが、現在最も一般的な方法は、プログラムや文章と同様に、楽譜エディタを用いて入力する方法である[1,2]。

本稿では、ICOToneの楽譜エディタについて、その満たすべき条件を考察する。そしてそれをもとに筆者らが試作したエディタについて述べる。なお、以下でいう楽譜とは、五線譜を指すものとする。

### 2. 楽譜エディタの満たすべき条件

楽譜に含まれるデータを入力し、演奏用データとして蓄える場合、楽譜データの適切な計算機表現、およびマンマシン・インタフェースの向上、の二点が問題となる。

#### 2.1. 楽譜データの計算機表現

楽譜には種々の情報が含まれている。これを計算機に入力して自動演奏する場合、どこかですべての情報を数値化する必要が生じる。

五線譜の場合、音程や音長は容易に数値化することができるが、強弱(p, f, cresc.など)や速度(allegro, rit.など)は、大小または増減関係や、おおよその値が記述されているにすぎず、単純に数値化するわけにはいかない。さらに表情の指示などになると数値化することはきわめて困難である。一方、人間的な演奏をさせようとする、音長や強弱などは、楽譜に書かれていないような変動をも加えなければならない。すなわち、数値化には演奏者の「解釈」の要素が多分に含まれているのである。

従来のシステムの多くは、楽譜の情報と解釈の情報とが混在した形でしか入力できなかった。その場合、同じ曲を解釈や奏法を変えて演奏するようなど

きでも、楽譜データそのものを変更しなければならない。これを防ぐためには、楽譜データと解釈データを明確に分離する必要がある。

楽譜を計算機で表現する場合、もう一つ問題点がある。それは、楽譜はテキストとは異なり、2次元に配置されたデータだという点である。通常のスコア(総譜)には、いくつかのパート(声部)があり、これらを重ねることによって楽譜は構成されている。ある音を出すタイミングは、同じパートのそれ以前の音符の長さによって定まる。よって、音符データはパート毎にシーケンシャルに(楽譜の横方向に)持つことが自然ではある。しかし、伴奏とメロディとの間のタイミングのずれなどを表現しようとする、パート間の(楽譜の縦方向の)データのつながりも無視できない。

#### 2.2. マンマシン・インタフェース

人間にとっての使いやすさを議論する場合、対象とするユーザ、使用目的などをまず明確にしなければならない。ICOToneではユーザとして音楽は専門とするが計算機はあまり知らない人を対象としている。ワークステーションで実現するのであるから、基本的にはマウスを使ってビットマップ・ディスプレイ上に音符を並べていく、という入力方法が妥当だと思われる。しかし、ピアノ等が演奏できる人にとっては、鍵盤から直接入力できたほうがより自然であるので、将来的には採譜プログラムFSI[3]と結合させるべきであろう。

前節でも述べたが、楽譜データは2次元に配置されている。人間にとってはパート毎に入力することが自然であるが、その際、他のパートとの関係を見ながら入力できることが望ましい。また、いつでも他のパートの編集に移行できることが必要である。

その他、人間にとって使いやすい楽譜エディタにするには、応答が十分速い、誤操作の危険性が少ない、などといった基本的な条件がいくつかある。これは、文書用のエディタでも同様である。

ICOTone - Score Editor

†Ryotaro KANAI, ††Takafumi SAITO

†MITSUBISHI Heavy Industries, LTD., ††University of Tokyo

### 3. 試作システム

以上の考察にもとづいて、筆者らは楽譜エディタをSUN-2上に試作してみた。

#### 3.1. 特徴

本エディタの特徴は次のとおりである。

- 1) 楽譜の情報と解釈の情報とが分離されており、独立に扱うことができる。
- 2) 複数のパートを同時あるいは別々にアクセス（入力および表示）できる。また、各パートとも和音を扱うことができる。
- 3) 楽譜はビットマップ・ディスプレイによって実際の五線譜の形で表示される。
- 4) 音符の表示間隔を、時間間隔に比例させた。

#### 3.2. 機能

本エディタでの表示画面の一部を図1に示す。詳しい説明は省略するが、基本的には、図1の中の必要なウィンドウ内にカーソルを移動してマウスボタンをクリックするか（左または中ボタン）、あるいはクリックしてポップアップメニューで選択するか（右ボタン）のいずれかの操作で、入力、編集を行う。音符の入力のときは、五線譜上の挿入したい位置にカーソルを移動し、中ボタン（音符）または左ボタン（休符）を押す。音符の長さを変更するとき、この位置で右ボタンを押す、ポップアップメニューで選択する。Playを選択すると、編集中の楽譜を即座に演奏する。この場合、エディタはまず内部データをMLOにコンパイルし、Concurrentタタタータを呼び出すことにより演奏が行われる[4]。

#### 3.3. 内部のデータ構造

本エディタでは、一小節をひとまとまりとして扱っており、各パート毎に小節および音符データのセルを双方向ポインタでつなげる、というデータ構造をとっている（図2）。これらのセルには楽譜の情報だけが含まれている。局所的な解釈データについては、まだ詳細は決定していないが、さらに別のセルを用意してポインタでつなぐことになる。

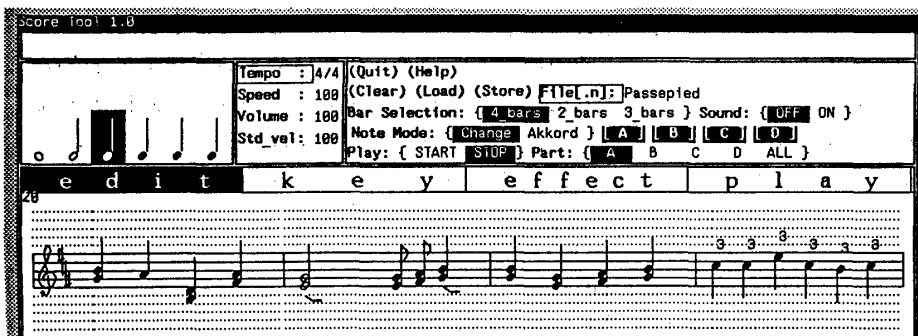


図1. 楽譜エディタの表示の一部

#### 3.4. 評価

システムは、解釈データの扱いを除いて、一応の入力ができるところまで完成している。総合的な使いやすさを他と比較することはまだできないが、現在までに気がついたことをいくつかあげておく。

- 1) 音符を時間に比例した間隔で表示したことによって、リズムが直感的にわかるようになり、音長の入力ミスが発見しやすくなった。その反面、画面スペースはかなり無駄が生じている。時間比例は一小節内にとどめ、各小節の五線譜上の長さは複雑さに応じて可変とする、などの改善を検討している。
- 2) 音長が頻繁に変わるときの音符入力は、現状では1個の音符あたり2度の異なるマウス操作が必要になり、面倒である。改善案としては、リズムによって音長の第一候補を予測する（たとえば、付点4分音符の次は8分音符にする）、マウスの左右動を音長の変更にあて、一操作で音高と音長の双方を指定できるようにする、などが考えられる。

#### 4. おわりに

ICOToneの楽譜エディタについて、その満たすべき条件を考察し、実際に試作したシステムについて述べた。今後、製作を進めていくにあたっての最大の難関は、解釈のデータを具体的にどのように処理するかである。これについては、楽譜入力の問題というよりは、音楽の本質にもかかわる問題であり、より一層の研究が必要である。

#### 参考文献

- [1] 計算機と音楽シンポジウム報告集, 1984.
- [2] Yavelow, C.: Music Software for the Apple Macintosh, *Computer Music Journal*, 1985.
- [3] 青柳: ICOTone 採譜プログラムF S I 第33回情処全大, 5N-9 (1986).
- [4] 平田: ICOTone 自動演奏プログラムConcurrent タタタータ, 第33回情処全大, 5N-7 (1986).

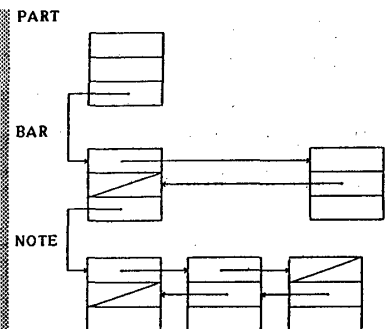


図2. 内部のデータ構造