

マンマシンシステムとしての 自動演奏システムの現状

筑波大学 笹川 瑠美, 三好 和憲, 五十嵐 滋

PSYCHEは、グランドピアノとシンセサイザの自動演奏を中心としたシステムである(図1)。このシステムでは、入力を音楽情報データ、計算機を経た出力を自動演奏として、自分の音楽を実感できること、さらにその音楽は、非常に音楽性の高いものとなっていることが大切である。そのための使い易さとは入力のし易さ、ターンアラウンドタイムが短いこと、出力の音が良いこととして言い換えることができる。以下では、音楽を記述する言語、それを生かすために入力に利用するハードウェア、ソフトウェアについて述べることにより、音楽システムの使い易さとは何かということを中心として、総合音楽情報システムPSYCHEの現状を紹介する。

1. PSYCHEの概略

PSYCHEでは、入力から出力まで少なくとも2種類の計算機を使用する。音楽情報データは、HITAC M-170上に汎用音楽記述言語EUROPAを用いて作られるが、その際、エディタとして、後に述べるように、文書編集用の行エディタをもとにして専用のもを開発中である。

コンパイラは、M-170上にPascalで書かれている。EUROPAの全仕様をサポートする、音の大きさ、長さの適当なデフォルト値を与えるといった機能が細かく実現され、かつ、実行時間が短くなくてはならない。

オブジェクトをマイコンに転送して、楽器の実時間制御を行なう。現在のところ、約200小節の楽譜をデータに直して音がきこえてくるまでには、データ作成に1時間強、コンパイルに3分、転送に5分ほどかかる。これだけの時間がかかって自分の音楽として実感できるのだろうかという疑問もあるだろうが、実際体験したところによれば、メロディーのわかっている曲は勿論、知らない曲でも音が出ると同時に、ホッと、あるいは、弾き間違いに気付いて赤面したりと、自分が演奏をした場合に感じるであろう気持ちと変わるところはない。

システムとして入力から演奏までの流れが整い心地良い音楽が得られるというのがPSYCHEの現在の姿である。

2. 汎用音楽記述言語EUROPA

音楽を再生するための入力方法には、いくつか考えられる。

1. 楽譜を計算機で読み採る。
2. 曲を弾き、アウトラインを入力する。
3. 画面に楽譜をかいていく。
4. テキストデータを作る。

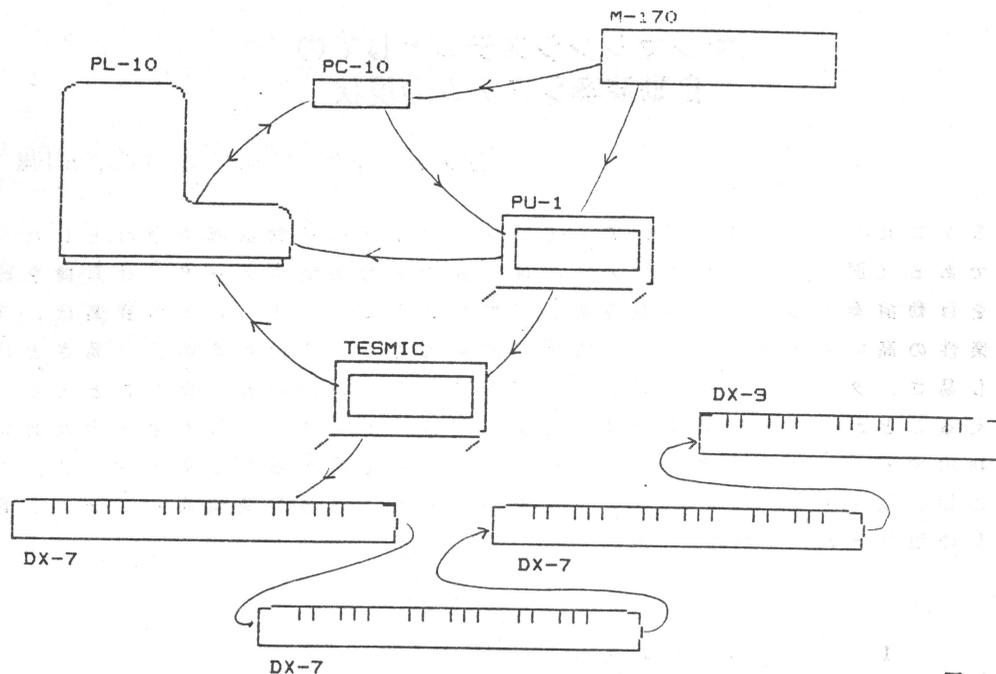


図 1 .

これらのうちから、PSYCHEでは、4.の方法を採っており、そのために音楽記述言語EUROPAが設計された。1.~3.については、次の章で触れる。

音楽の情報を楽譜からテキストデータに移すことは、楽譜からよみとれる曲全体の速さ、拍子、調、各音のもつ情報（高さ、長さ、装飾記号など）、メロディーの情報（次第に大きとか、ゆっくりとか）を最低限、含まねばならない。このうち、音のもつ高さと言長さという基本的な2つの情報に限ってもその表し方にいくつかのアプローチがあるが、それらに関しては、相当大きいデータを作ることになるため、入力に要するストローク数が少ないこと、音楽的意味がわかりやすいことという点が、新しくデータを作る時と、そのデータを見直す時において主に評価されるべきであろう。

EUROPAは、楽譜のイメージをなるべく損なわずに、データ化できることが設計方針の中心となった。ここでいう楽譜のイメージとは、楽譜上にある音の高さ、長さ、様々な記号の慣用的、図式的使われ方からうけとる楽譜ということである。

楽譜にかかっているもののうち、もっとも基本的なものとして音符がある。音符は、音楽をデータに直したとき、大きい部分を占めることは想像できる。そこで、音符の表し方が、音楽記述言語では大きいポイントの1つとなる。音楽記述言語を使う人は、今のところ限られた人数ではあるが、その中でも音楽に慣れている人とそうでない人がいる。図2の楽譜をみて頭の中でスッとメロディーを歌える人ならば音楽に慣れているといつてよいであろう。そのときメロディーはどのようにしてでてくるのか。ハミングかもしれないが、階名(do, re, mi...)や音名(c, d, e... , n, o, a...)で歌うこともあるだろう。EUROPAは固定ド法で

音名唱法をすることで楽譜上の音符をデータに直す。つまり、音名は、c, d, e, f, g, a, b, それに音高をオクターヴ分割することで音名に付加して絶対音名のかわりとし音符がかかれる。長さは、4分音符や8分休符といった読み方に従い、それぞれ、4、8と表され、付点音符は、4、8..のように表す。図2の楽譜をEUROPAでかくと図3のようになる。この記法は、音楽に慣れている人に都合よく作られはしたが、慣れていない人にとって馴染めないものとはならないはずである。音楽に慣れていないということを楽し譜をみただけで曲が想い浮かばず、演奏をきいてから楽譜を追うようなこととみなすと、そのような人は、楽譜に関しては白紙であり、音楽記述言語に取り組むことで楽譜上の音符を理解するようになる。このとき音名で音符の高さを表すという作業を続けるならば、短い期間で音楽に慣れた人になれるであろう。また、このような入力をすると、大譜表上で自由に音を追うことができ、高音部譜表と低音部譜表の間を旋律に従って行き交うことができる。データを見直す場合は、音名がかかっていることから、曲を連



図 2 .

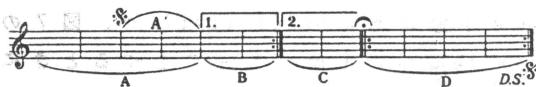
```
!r4/4
!p1.lvl a44 c+6 b c+4 ea
!p2.lvl s4 ea+p d+ ea+ aa g ga
!p3.lvl f4 aa+ fa ea
!p4.lvl aa8 g aa c+ ea+2
```

図 3 .

Allegro.



図 4 .



(奏法) A, B, A, C, D, D, A', C

図 5 .

想することは容易である。

EUROPAでは数字の使用は以下のときに限られる。音の長さ、小節番号といった、楽譜上の時刻を示すものは、headingと呼ばれ、 $! p n_1 \dots n_2 \dots v n_3$ の形をしており、第 n_1 小節の第 n_3 声部の n_2 個めの音という意味をもち、楽譜上の音符が一意に定まる。このheadingや、表情記号、装飾記号の演奏方法の指定をするときに数字の使用は限られ、数本来の意味を表しており、自然である。

図4の楽譜(Scarlatti Sonata)には、速度記号(Allegro)、音部記号(♩ , ♮)、拍子記号(♩)、発想記号(強弱に関するものとして p , $>$)、装飾記号(♯ , \sim , ♯)、臨時記号(\flat , \sharp)、調号(♩)、運指、五線、加線、縦線と大括弧それに音符がある。このScarlattiのピアノ譜をいわゆるクラシックものではごく標準的なものとみなした場合、ほんの3小節のうちに上にあげただけ記号が含まれているのである。これらのうちから、自動演奏のためのデータを作る時に意味をもたせる必要のあるものは先に述べた音符の外には、運指および楽譜の骨組となる五線、加線、縦線、音部記号、大括弧以外のすべてのものである。リズムをつける場合に小節すなわち縦線が意味をもつこともあるが、それは別の表し方をするものとする。意味をもつそれらの記号をデータ化する場合にもとの意味を素直に連想できるものがよろしい。記号をデータ化するということは図形を文字列に直すことも含まれる。これらのことから、EUROPAでは、様々な表情記号、装飾記号を従来楽譜上にあるのとなるべく似た形で、あるいは記号をすぐに連想できるような呼び名でデータ上を実現しようとした。例えば、Scarlattiの楽譜中の記号は次のようになる。

Allegro	...	Allegro
p	...	piano
>	...	>
~ (モルデント)	...	mordento
♯, ♯ (前打音)	...	vorschlag

楽譜には、省略記号が多くあり、EUROPAでは、その一部がデータ化できるようになっている。最も単純といえる省略に調号を譜表の各五線の頭につけることがある。これは、EUROPAでは、調号指定によりなされ、次の調号指定がくるまで有効ということになる。これが単純といえるのは、調号が単に各音の高さを指示するだけで、曲の流れ、表情を直接に指示するものではないからである。代表的な省略記号として反復記号がある。EUROPAでは、図5のような反復奏法が可能であり、楽譜上で例えば、DDのように同じところを2回弾く場合、1回めと2回めとで曲に異なる表情をつけることが可能になっている。

EUROPAでは現在、図6の第4小節にあるpedal simileや、図7のespr., cantabileは仕様がない。pedal simileについては、データを作るときにエディタのコピー機能を用いれば問題はないのだが、一般にカノンのように旋律のパターンを見わけての演奏をEUROPAで実現することはできるが、自動演奏をさせるために音楽をデータ化せねばならない時、難しい問題となることの1つである。espr. やcantabileといった曲想に関する発想記号はEUROPAでは実現していない。このような記号をデータ中におくことができると、高級言語の感が強くなるが、曲想を具体的に各音の強さと長さによりにどのように展開していくかが曲それぞれによって異なり、演奏方法を一ヶ所で指定して後で何度かその記号を

Moderato (♩ = 152)

Op. 69 Nr 2

10
BIS

p
simile.
L.H. * *L.H.* *

図 6

p
L.H. espr.
L.H.
cresc.

図 7

!p1.lvl c+b. d+13 c+8 e+4 e+b b. c+16 bb d+4 d+3
 !p3.lvl a4 a5 b4 b5 c+4 e+16 c+ [a/c+]4 [g/b]8
 !p6.lvl c+b. b+1b c+8 e+4 e+8 bb. c+16 bb d+4 d+3
 !p7.lvl a4 bb c+ [f/g/a+]sf.# [e/a/c+]4 [a/e/g/b]8 [c/e/a]4 s0
 !p7.lvl e+b. f+1b e+8 f+4 f+8 f+32 g+ a+b g+16 f+.8 f+8 e+. e+.
 !p11.lvl e+sf. b c+. a. e+sf. d+. b. e+sf. c+. a. [a/c+]4 [g/b].8
 !p13.lvl c+8. b+1b c+8 e+4 e+8 bb. c+16 bb d+4 d+3
 !p15.lvl a4 bb c+ [f/g/a+]sf. b [e/a/c+]4 [e/g/b]8 [g/b]4 [a/c+]8
 !p17.lvl [e/a/c]8 [e/g/b]8 [e/a/e+]4 f+16 g+32 a+32 a4 c+16 b a4 s0

Andante grazioso
Mozart
p
cresc.

図 8

使えるというものでもない。

音楽記述言語は、自動演奏に音楽性を十分盛り込むことができなければならないことは第一の条件であるが、データを作るとき、見直すときに、音楽を連想しやすいことで、音作りの能率が上がるようになっていなくてはならない。

3. 入力方法

入力方法は、2. で掲げたいくつかが考えられるが、1. 楽譜を計算機で読み探ると2. 曲を弾きアウトラインを入力するというのは、それぞれ、スクリーンやパソコンインタフェースをもつミュージックキーボードという新たなデバイスを用意せねばならず、一般的でない。演奏入力は一見理想的かもしれないがこれによって容易に得られる適当な情報は、音の高さしかなく、また、音の高さすら演奏を間違えると得られず、後でどうしても長さや大きさの情報を加えねばならず、大変面倒で、自動演奏で素晴らしい演奏を得ようとするときの入力方法としては不適當である。3. 画面に楽譜をかいていくこのとき問題になるのは、入力の速さと、楽譜で演奏の全情報を賄うことができるかということである。マウスやタブレットを用いて楽譜をかいていくのと、タッチタイプでデータを作っていくのでは、断然、後者が速い。また、楽譜上の情報は、演奏情報としては不足分が大きく、それらを補おうとするとき、楽譜がゴチャゴチャとなるか別のファイルを用意するかで統一されない。

新たなデバイスを用意せず、速く、大きい入力データを作ることができるという点で楽譜という従来の視覚に訴える音楽から離れてはしまいが、適当な音楽記述言語でデータを作るのが今のところ最も望ましい入力方法と考えられる。

4. エディタ

音楽記述言語を用いて音楽データを作る時に、楽譜が失われるため、入力している人が苦痛を感じないようなツールを与えてその代償としなくてはならない。それには、音楽データ専用のエディタが作られるべきである。このエディタは、音楽データの特性をつかんだものでなくてはならないが、使う人の音楽の素養が様々であるため、各人がエディタに要求するものが異なり、何が必要なのかがなかなか定まらなかった。

EUROPAは、専用のエディタができるまで、M-170上のエディタを用いて使われていた。専用エディタは、テキストエディタをもとに、Tektronix4025グラフィックス端末上で用いて使われるものとして作られた。

データは図8のようになるのがふつうである。つまり、各行頭にheadingがおかれ、各行に入っている小節数は一定になり、heading中の小節指定の数だけがふえていく。headingは、特殊記号と英字、数字が入り混り、入力しにくい。そこで入力モードでプロンプトとしてheadingを自動的に出し、データの一部になるようにした(図9)。これにより、入力の能率が大変上がった。

この専用エディタは構造エディタとし、1行入力する度にシンタックスをチェ

84-07-16

```
1:|p1.1v1 c+0. d+16 c+0 e+4 e+8 b8. c+16 b8 d+4 d+8
2:|p3.1v1 a4 a8 b4 b8 c+4 e+16 d+ [a/c+14 [g/b]8
3:|p5.1v1 c+0. d+16 c+0 e+4 e+8 b8. c+16 b8 d+4 d+8
4:|p7.1v1 a4 b8 c+4 [f/b/d+]sf.8 [e/a/c+14 [d/e/g/b]8 [c/e/a]4 s8
5:|p9.1v1 e+0. f+16 e+8 f+4 f+8 f+32 g+ a+0 g+16 f+.8 f+8 e+.
6:|p11.1v1 e+sf.8 ct. a. e+sf. dt. b. e+sf. ct. a. [a/c+14 [g/b]8
input:|p13.1v1_
```

図 9 .

ックするようになっている。このことは、コンパイラを通してからデータを見直すことを減らすことになる。

データアナリシス用にデータを部分的に楽譜化する機能も用意した。そうすることで、音の長さや、入力間違いが一目りょう然にわかる。音楽データでは同じ音符や和音が繰り返されるという特徴があり、行コピーだけでなく、入力モードで記号を繰り返し入力できるようにすることも考えている。

5. 今後の課題

P S Y C H E は、入力方法はこれまでに述べてきた方針で今後も続けていく予定であるが、現在のところ最も改良の余地のある M - 1 7 0 の専用エディタの強化をはじめ、新たなツールを導入して能率をあげていこうと考えている。

参考文献

- 笹川、三好、五十嵐 : 汎用音楽記述言語の設計試作、第25回プログラミング・グシンポジウム報告集、pp.16~25 (1984)
- 笹川、三好、五十嵐 : 総合音楽情報システム P S Y C H E、計算機と音楽シンポジウム報告集、(1984)
- R.Nelson : A Graphics Text Editor for Music - Part 1, Part 2,Byte pp.124~138,104~118,(April,May,1980)
- J.Raskin : Using the Computer as a Musician's Amanuensis - Part 1 Part 2,Byte,pp.18~28,120~128(April,May,1980)

本 PDF ファイルは 1985 年発行の「第 26 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場（＝情報処理学会電子図書館）で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>