

楽譜エディタ emucom の使い勝手

高田正之 渡邊哲史 小谷善行
東京農工大学

emucom は、計算機支援作曲・編曲環境の核として開発した WYSIWYG 方式の五線譜エディタである。その使い勝手を次の2点から評価した。(1) 利用者による評価・要望を、使用中にその場でファイルに記録してもらった。(2) 利用者のキー入力を時間情報付きですべて記録した。これを分析することでキー割当てやコマンド体系の妥当性を知ることができる。

AN ESTIMATION OF MUSIC SCORE EDITOR "EMUCOM"

Masayuki Takata Satoshi Watanabe Yoshiyuki Kotani
Tokyo University of Agriculture and Technology
2-24-16 Nakacho Koganei Tokyo 184 Japan

We designed and implemented a WYSIWYG music score editor as the kernel of computer-aided music system. User friendliness of the editor is evaluated by the two way: (1) on-line collection of users' feeling, opinion and claim on it, and (2) record of users' time chart of key strokes.

Analyzing those data, we investigate adequacy of key position and command system such a man-machine interface of music.

1. はじめに

emucom は、計算機支援作曲・編曲環境の核として開発した WYSIWYG 方式の五線譜エディタである。その使い勝手を次の2点から評価した。

(1) 利用者による主観的・定性的な評価をmonku.emu というファイルに克明に記録した。これは、利用者と開発者の情報交換の場となり、エディタ自体のデバッグに大いに役立った。実際にエディタを使っていて気が付いたことをその場で書き込むようになっているので、設計会議だけでは得られない様々な要望や提案が得られた。なお、利用者は今のところ開発者の身近にいる音楽研究班の数名だけである。

(2) 利用者のキー入力を時間情報付きですべて記録した。これを分析することでキー割当てやコマンド体系の妥当性を知ることができる。今回は、主に打鍵レベルの利用者模型を明らかにするという観点からの分析を行なった（1990年5月のヒューマンインタフェース研究会で発表予定）。また、今回の実験では、楽器鍵盤で音高を指定する方式を一部に取り入れたが、このような他の入力方式との比較にも定量的指標を与えることができよう。

2. emucom の入力方式

2.1 キー割当て

emucom のキー割当てに関する基本的な考え方は、

- ・楽譜上の基本的な記号（音符、休符）入力については、ブラインドタッチの可能な簡素化されたキーに割り当てる（図2.1）。
- ・特殊キー（[ctrl], [grph], [shift], [caps]）使用によりコマンドを体系化（図2.2参照）する。

である。

[BS]	挿入 [INS]	削除 [DEL]
音符類の 決定 [CR]	↑	
[SHIFT]	←	→
音価変更 [XFER]	↓	

図2.1 基本的なキー割当て（PC-9800シリーズ）

ここで、

(1) 矢印キーで方向を指定すること

と、

(2) ブラインドタッチを可能にすること

とを両立させるために、キー配置を工夫している。すなわち、図2. 1のキーの上に右手のホームポジションが置かれるようにしている。

・カーソルを一つ上の縦座標へ移動する	↑
・記号を一つ上の縦座標へ移動する	SHIFT+↑
・カーソルを一つ上の段へ移動する	CTRL+↑
・記号を一つ上の段へ移動する	SHIFT+CTRL+↑
・カーソルを1オクターブ上に移動する	CAPS+↑
・記号を1オクターブ上へ移動する	SHIFT+CAPS+↑

図2.2 特殊キーの併用によるコマンドの体系化

図2. 2は、いずれもなにかを上へ移動する操作を示している。これにシフトキーを併用することにより「何を」移動するのかを表現する。つまり、次のような関係がある。

↑：上へ移動

↓：下へ移動

シフトキーなし：カーソルだけが移動

シフトキーあり：カーソルと、その位置にある音符などが同時に移動

コントロールキー、CAPSキーなし：二度だけ移動

コントロールキーあり：段（五線）間の移動

CAPSキーあり：1オクターブの移動

音符類などの基本的な入力手順は図2.3に示す通りである。すなわち、音符などの音価および、音符の位置を何度かのキー操作によって自由に変更することができる。最後に、記号決定（改行キー）で音符などの音価と位置を確定する。これはワープロにおける仮名漢字変換で漢字を確定する作業に似ている。

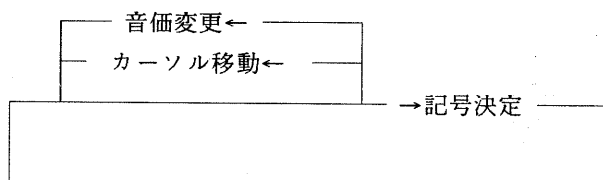


図2.3 基本的な入力手順

2.2 プリミティブな編集機能

emuc om の編集機能のうち、基本的なものを次に示す。図 2. 2 以外にも編集コマンドが体系的に設計されているのが理解されるであろう。

- ・二度の単位で移動する(↑, ↓)
- ・縦組単位で移動する(←, →)
- ・小節単位で移動する(GRPH+←, →)
- ・前後の記号へ移動する(GRPH+↑, ↓)
- ・段を移動する(CTRL+↑, ↓)
- ・記号削除をする(DEL)
- ・縦組削除をする(SHIFT+DEL)
- ・1 縦組の挿入をする(INS)
- ・記号を移動する(SHIFT+↑, ↓, ←, →)

2.3 楽器鍵盤で音高を指定する入力方式

パソコンのキーボードだけからの入力に加え、楽器鍵盤を利用した入力方式を次のように設計した。これにより、音程の指定が楽器のインターフェースと一致する。これは、演奏の自動採譜を目指すシステムではないので、音価の情報は楽器からは取らない。

左手…楽器（縦座標の変更）

右手…コンソール（音価変更、音符決定など）

- ・カーソルは、最後に押された鍵盤の縦座標にある。
- ・従来通りカーソルキーでの縦座標変更もできる。

3. 利用者による主観的・定性的な評価

emuc om の利用者がシステムの使用中に書き込んだ monku.emu というファイルから得られた主観的、定性的な評価について述べる。利用者には、emuc om のセッションの最後にシステムに対する感想を求められる。これがこのファイルに収録される。また、記入は編集集中に随時行うこともできる。

3.1 画面配置、図形表示

おもに、カーソルの動きについての要望が多かった。例えば、図 3.1 のようにカーソルは、内部データ上では五線の第 1 線上の 4 分音符を指しているのだが、画面上では、図のような和音（2 度音程を含む和音）のときはずれてしまう。カーソルの位置情報は、画面データのレベルで管理し、それを内部データと対応付けるようにすべきである。



図 3.1 カーソル位置がずれる例

3.2 システムのレスポンス

emuc om のレスポンスは、通常の記号入力、消去に関しては問題はなかった。しかし、画面更新のスピードについて利用者は特に不満であった。この問題を解決するために、画面更新中にもコマンドを受け付け、画面よりも先に処理をしてしまう方法をとった。そして、何もキー入力がないときに、残りの画面更新を行うようにした。

3.3 機能、キー配置

図2.2を見てわかるように、特殊キーの併用は、一見すっきりとしたコマンド体系のようにみえるが、実際の利用者は二つ以上のキーを同時に押さなければならない。その手間が筆者の予想を越え、キー配置変更の要望がずいぶんあった。

利用者が、テキストエディタを普段使っている関係上、テキストエディタと同じキー配置を望む声も聞かれた。これらの問題は、利用者定義のキー配置を可能にすれば解決する。実際、emuc om では利用者定義が可能となっているのだが、今回の打鍵情報収集実験のために、この機能は使用しなかった。

4. キー入力の時間情報

emuc om 操作中の打鍵情報を、次の2点について収集した。

- ・打鍵キーの種類
- ・直前の打鍵から対象打鍵までの時間

これらの情報から、emuc om の利用者の行動を打鍵レベルで予測した。今回は、打鍵情報から得られた使い勝手に関する結果について述べる。

4.1 カーソル移動の頻度とエディタの習熟度

利用者のデータで共通しているのは、どれもカーソル移動(↑, ↓, ←, →)が全体の5割前後を占めていることである。また、emuc om の習熟度とも関係しているようである。emuc om の習熟度とカーソル移動の占める割合を表4.1に示す。

表4.1 習熟度とカーソル移動の占める割合

利用者	カーソル移動の占める割合	習熟度の目安
A	48.21%	emuc om の開発者
T	57.14%	2年前から使用
D	62.35%	今年度9曲入力
E	67.07%	今年度3曲入力

表4.1から得られる習熟度は、

$$E < D < T < A$$

であり、emuc om に慣れているものほど、カーソル移動の割合が少ないことがわかる。

ここで、利用者Aだけ1位が[shift]+↑(記号を上に移動する)となっているのは、お

そらく入力の手順の違いからくるものであろう。通常の音符入力は、入力する位置をカーソルで完全に決めてから行われるのに対し、利用者Aの場合は、目的の位置付近でとりあえず入力してしまい、あとは移動させながら修正するという方法をとっている。そのため、カーソル移動を超える数となったのであろう。被験者Aがこのような方法をとっているのは、画面上で位置を視認するのに、通常のカーソルの形(+)よりも、音符で行った方が楽だからである。今後、カーソルの形状の改善が必要である。

4.2 楽器鍵盤による入力

打鍵時間の測定結果(別途発表予定)によると、楽器鍵盤を用いたときは、記号決定キー([cr]キー)打鍵までの時間がカーソルキーを用いたときに比べて、200ms~300ms速くなった。これは興味深い事実である。

楽器鍵盤による縦座標の指定には、カーソルキーによる指定のように現在のカーソル位置を視認する必要がないのが原因と思われる。つまり、楽器鍵盤の場合、「習熟」という要因があるにせよ、一つの打鍵操作に音程の指定作業が込められる。これは効率的なマシン・インターフェースの要件であろう。

5. まとめ

emuc om の使い勝手の評価を行った。ファイル monku.emu によって、開発者には予想もし得ない評価・要望を随時知ることができた。収集した打鍵情報は、利用者の編集行動の予測を打鍵レベルで可能にただけでなく、使い勝手に関する定量的評価の指標となることがわかった。

なお、楽譜エディタの打鍵に関する実時間測定と、打鍵時間模型について、さらに興味深い結果が得られている。

6. 参考文献

- [1] 粕川, 木村: パーソナルコンピュータ用打鍵データ収集プログラムとその応用, 情報処理学会第30回全国大会, 3G-8(1985.4), pp.1645-1646.
- [2] 山崎直子: 共通楽譜データ形式, 情報処理学会第29回プログラミングシンポジウム報告集
- [3] 渡邊哲史: 作曲支援のための楽譜エディタの開発, 東京農工大学数理情報工学科1987年度卒業論文
- [4] 田子正明: 楽曲の特性分析 - 楽譜入力系と和声分析系 -, 東京農工大学工学部数理情報工学科1985年度卒業論文, 1986
- [5] 山崎直子他: 図形情報を基本とする共通楽譜データ形式, 情報処理学会第38回全国大会講演論文集, 3K-8, 1988
- [6] 山崎直子他: Coda から節形式への変換系, 情報処理学会第38回全国大会講演論文集, 1989