

Some thoughts about note–duration representation in the ABC language Part 3

1. Introduction

In 'Some thoughts about note–duration representation in the ABC-language' , Part 1, Part 2 (1) I have presented the MCM-language as my slight modification of the ABC-language (for which Chris Walshaw deserves all credit, as I already wrote in Part 1).

In my opinion the MCM–language is more close to the standard musical notation than the ABC–language. In short:

- The MCM-language is context-independent (C16 is always a sixteenth note C, C4 always a quarter note etc.).
- Visualizing the MCM–code to a mental image of music notation is in my experience easier, i.e. in more complex situations (see 4.1 and 4.2 below).

In Part 2 of my 'Some thoughts' I wrote that the MCM-language is less compact in comparison to the ABC-language, what could be seen as an disadvantage. I suggested that compactness could be enhanced by applying Lilypond's approach to make rhythm 'sticky' (2).

I did not see at that time how to implement 'sticky rhythm' but studying newLisp (www.newlisp.org) gave me a solution. In my experimental MC Musiceditor 7.1.2 and all releases later 'sticky rhythm' is implemented (3).

2. Making rhythm sticky

'Sticky rhythm' is in fact shorthand notation. In Lilypond is e.g. C2CCC shorthand for C2C2C2C2.

In newLisp parentheses are powerful means to structure code. In the MCM- en ABC-language we can find a similar idea: chords are grouped by square brackets:

```
[A, 2C2E2G2B2d2]
```

However, I prefer the shorthand

```
[A, CEGbd] 2
```

which is not the ABC-standard but works well with abcm2ps.

I combined these two ideas by defining 'sticky rhythm' with parentheses:

```
(C D E F G) 2 equals C2 D2 E2 F2 G2  
(CDE FGA) 8 equals C8D8E8 F8G8A8
```

One example in MCM and ABC, to compare:

a. Example MCM:

`(CCCC)16 (CCCC)16 (CCCC)16 (CCCC)16`
equals
`C16C16C16C16 C16C16C16C16 C16C16C16C16 C16C16C16C16`

or alternatively

`(CCCC CCCC CCCC CCCC)16`
equals
`C16C16C16C16 C16C16C16C16 C16C16C16C16 C16C16C16C16`

b. Example ABC with L:1/4:

`(CCCC)// (CCCC)// (CCCC)// (CCCC)//`
equals
`C//C//C//C// C//C//C//C// C//C//C//C// C//C//C//C//`

or alternatively

`(CCCC CCCC CCCC CCCC)//`
equals
`C//C//C//C// C//C//C//C// C//C//C//C// C//C//C//C//`

The advantages? Shorter, more comprehensive code and less typing. Grasping groups of notes, rhythmical patterns, as meaningful entities appears in the 'sticky rhythm' variants to be quite easy. So the readability of code is enhanced.

3. How to make slurs?

Again it is not the ABC standard, but abcm2ps that gives an answer how to make 'sticky rhythm' or slurs:

- `('ABC)` gives a slur above the notes ABC
- `(,ABC)` gives a slur below the notes ABC

So:

`(CCCC)2` equals `C2C2C2C2`, i.e. 'sticky rhythm' or shorthand

`('CCCC)2` equals `('C2C2C2C2)`, i.e. a slur above the four notes C

`(,CCCC)2` equals `(,C2C2C2C2)`, i.e. a slur below the four notes C

the normal case `(C2C2C2C2)` equals a slur above/below the four notes C

4. Practical examples

To investigate the meaning of 'sticky rhythm' I show you two extreme examples:

4.1. Regular measures, no or little rhythmical variety

```
X:1
M:3/4
L:1/4
K:C clef=treble
%ABC
F F G | A A B | c d/c/B/A/ | G3 |]
```

```
X:2
M:3/4
L:mcm_default
K:C clef=treble
%MCM with sticky rhythm
(F F G)4 | (A A B)4 | c4 (dcBA)8 | G2. |]
```

```
X:3
M:3/4
L:1/4
K:C clef=treble
%ABC in MCM with sticky rhythm
F F G | A A B | c (dcBA)/ | G3 |]
```

These three small examples are similar. One can argue that 'sticky rhythm' in these cases has no or little added value. However, in the following screendump of MC Musiceditor, you will see that the color differences between notes, numbers and barlines support grouping.

The screenshot displays the MC Musiceditor 7.1.2 application window. The title bar reads "MC Musiceditor 7.1.2 - www.mcmusiceditor.com - www.bestmusicteacher.com". The menu bar includes "File", "Edit", "Styles", "Export", "Tools", "Templates", and "Help".

The main editing area shows the following ABC notation:

```
1 X:2  
2 M:3/4  
3 I:mcm_default  
4 K:C clef=treble  
5 %MCM with sticky rhythm  
6 (F F G)4 | (A A B)4 | c4 (dcBA)8 | G2. | ]  
7
```

A floating window titled "MC Musiceditor SumatraPDF - temp_pre.ps" is open, showing a preview of the musical score rendered as a PDF document. The PDF preview shows a single page with a treble clef and a musical staff containing the notes corresponding to the ABC notation above.

The bottom of the interface features a toolbar with various musical notation symbols (quarter, eighth, sixteenth notes, rests, etc.) and a "short cuts" section with a treble clef icon. On the right side, there are several utility buttons: "Save", "View PDF", "Workarea", "Play", "Metronome", and "Previewer".

4.2. No measures or no regular measures, much rhythmical variety

```
X:1
M:none
L:1/4
K:C clef=bass
%ABC
D//G,,//A,,// C, D, E, ^F,/ | \
G,,//D//A,,//D//A,,// C, D, E,1 ^F,/G,/ | \
D//A,,//D//A,,//G,,//A,,//A,,// C, D, E,/^F,/G,/ | \
E//D//C//_B,,//_A,,//_G,,// E,//D,//C,//_B,,//_A,,//_G,,// _E,, D,, =E,, ^F,,/ | \
C,,4 - C,,4 |]
```

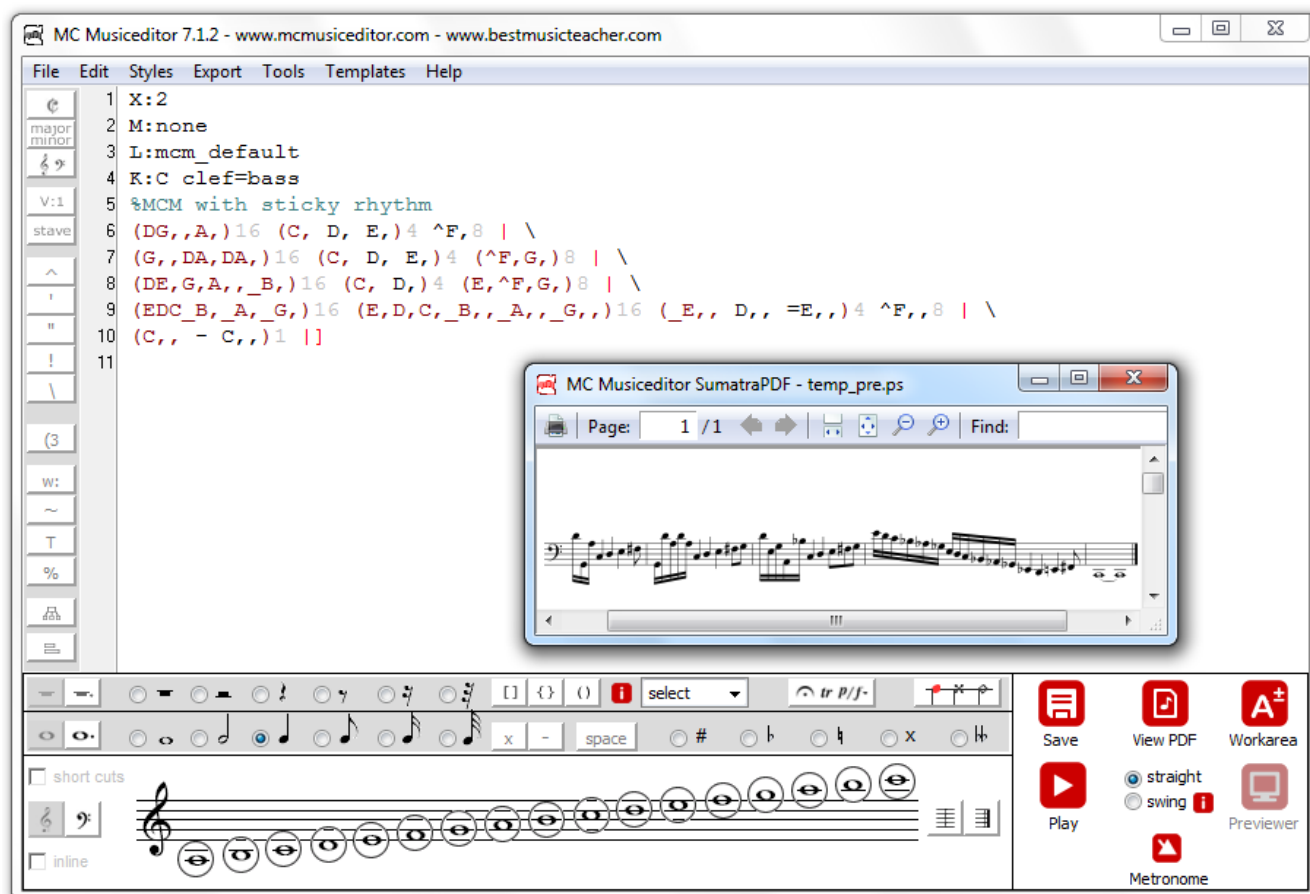
```
X:2
M:none
L:mcm_default
K:C clef=bass
%MCM with sticky rhythm
(DG,,A,)16 (C, D, E,)4 ^F,8 | \
(G,,DA,DA,)16 (C, D, E,)4 (^F,G,)8 | \
(DE,G,A,,_B,)16 (C, D,)4 (E,^F,G,)8 | \
(EDC_B,_A,_G,)16 (E,D,C,_B,,_A,,_G,,)16 (_E,, D,, =E,,)4 ^F,,8 | \
(C,, - C,,)1 |]
```

```
X:3
M:none
L:1/4
K:C clef=bass
%ABC in MCM with sticky rhythm
%note the added number 1
(DG,,A)// (C, D, E,)1 ^F,/ | \
(G,,DA,DA)// (C, D, E,)1 (^F,G,)/ | \
(DA,DA,G,,A,A,,)// (C, D,)1 (E,^F,G,)/ | \
(EDC_B,_A,_G)// (E,D,C,_B,,_A,,_G,,)// (_E,, D,, =E,,)1 ^F,,/ | \
(C,, - C,,)4 |]
```

I prefer the last two examples above the first one: groups of notes can be easily discerned. In the following screendump of MC Musiceditor, you will see that the color differences between notes and number support these grouping. Note that the barlines are here only orientation lines.

By the way, is the following better to read? Maybe more than a matter of taste.

```
X:2
M:none
L:mcm_default
K:C clef=bass
%
(DG,,A,)16 (C,D,E,)4 ^F,8 | (G,,DA,DA,)16 (C,D,E,)4 (^F,G,)8 | \
%
(DE,G,A,,_B,)16 (C,D,)4 (E,^F,G,)8 | \
%
(EDC_B,_A,_G,)16 (E,D,C,_B,,_A,,_G,,)16 (_E,,D,,=E,,)4 ^F,,8 | \
%
(C,, - C,,)1 |]
```



5. Concluding remarks

Compact code is one thing. Code structured in meaningful groups is another thing. Both are essential in making code readable and comprehensive. In my opinion 'sticky rhythm' can in some cases be a valuable addition to MCM or ABC. One has to consider that.

Chris Walshaw has to consider to implement into the ABC-standard

- [notes]rhythmicalvalue as shorthand chord notation
- ('notes for a slur above the notes
- (, notes for a slur below the notes

In my experimental MC Musiceditor 7.1.2 you can enter MCM- or ABC-code in the way you want: with or without 'sticky rhythm'.

Footnotes

(1)

<https://mcmusiceditor.com/download/maliepaard-abc-note-duration-representation-part-1.pdf>

<https://mcmusiceditor.com/download/maliepaard-abc-note-duration-representation-part-2.pdf>

(2) *In Lilypond C16 C C C C C C C represents eight sixteenth notes, in two groups (beams automated). I think that for better readability this should be better: C16CCC C16CCC.*

(3) *After the experimental MC Musiceditor 7.1.2 the next release will be 8.0.0, recoded with newLisp, which significantly enhances the performance.*

About the author:

Reinier Maliepaard is psychologist, software engineer, organist and teacher at the ArtEZ Conservatorium Netherlands (music theory, music history, composition).

His free/open source (GPLv2) music notation program MC Musiceditor (Windows) can be downloaded at www.mcmusiceditor.com