Edgar Huckert 2009

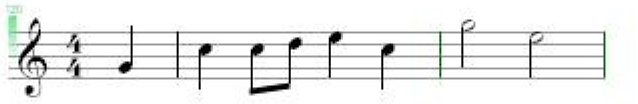
Problems with the design of a MIDI editor

Anacrusis and count-in beats

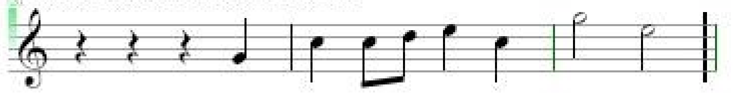
You may have seen that **HUMidi** offers a feature to add count-in beats at the start of a MIDI file portion.

Producing count-in beats seems to be a rather simple problem: put the MIDI events (normally event_on event_off events on channel 10) in front of the MIDI file and play that file. This was my first naive approach. You will find that this is not a satisfying approach with many MIDI files.

For scores starting with an anacrusis some usual MIDI converters (I use the MIDI converter in Capella V4.0) generate rests in front of a MIDI score - even if you don't enter rests in the score:



Normal music notation



Usual MIDI rendition

This is normally not a problem - both notations and MIDI renditions are valid. It may however be a problem if you add count-in beats in front of the MIDI tune: should the count-in beats replace the rests or leave the rests intact? I guess that every normal musician agrees that the count-in beats **replace** (overwrite) the rests. Things may even be more complex: the sum of rests may be so small that a single count-in beat cannot fill this measure!.

When adding count-in beats a designer for a MIDI editor must proceed as follows:

- Find the first real note event (skip rests athe start of a tune)
- Get the last measure events (saying 3/4, 4/4, 6/8 etc.) before the first note event - the duration of the count-in beats depends on this mesaure event
- Insert the count-in beats before this first note event starting with timestamp 0
- Correct the timestamps for the first note event and all other events

This is the rough algorithm. Note that this algorithm is not 100% correct: there are many MIDI file on the commercial and the PD market that have no measure events at all! Measure events are optional in the MIDI standard.

The bitter consequence is thus: it is not possible to design an algorithm that produces correct count-in events for all MIDI files.

Even if you succeed to place the count-in beats correctly in the MIDI file some problems may still remain. The musician cannot be forced to specify the "usual" number of count-in beats. In our notation sample above this would be 3 or 7 count-in beats - he can also specify 5 or 6 or.... if he likes this. You (being the designer of the MIDI editor) will now have the problem that the bar/beat count may be incorrect - which may lead to an even more complex algorithm just to produce count-in beats.

Let me remember that - even if you have solved all these problems - some questions remain open:

- Should the count-in beats remain in the MIDI file when it is saved?
- What do do with some special count-in schemas? Jazz musicians count in following the schema "1 2|1 2 3 4|" (the first bar has **half** notes) whereas classical musicians just count "1 2 3 4 | 1 2 3 4"

Mixing MIDI files

My MIDI editor *HUMidi* offers a feature that allows mixing two MIDI files. You may have experienced that this feature does not always produce the expected results.

MIDI ist not the best base to start a mix process. The follwing questions must be answered (this is not a complete list of the problems):

- Can we mix MIDI files with different tempi (Adagio file with an Allegro file)?
- What to do with different tonalities (C major with Bb minor)?
- What to do with different measures (3/4 with 6/8)?
- What to do with anacrusis?

Some of theses questions can be answered in a rather brutal way:

- Different tonalities: MIDI has no mandatory tonality information. So don't care for tonality
- Different tempi: MIDI has no mandatory tempo information. So don't care for tempo.
- Different measures: MIDI has no mandatory measure information. So don't care for measure information.

In fact things are not so simple: instead of not taking are of information like tempo, tonality, measure I prefer to produce first two clean versions of the files - i.e. to wipe out all tempo, measure, tonality information and then to mix the files. As the mixing process can not decide which of these information should dominate I leave it to the user to reintroduce these information into the mix result.

There couldbe a special case when tonality, tempo and measure information could be preserved: when both files are exactly identical in these parameters. *HUMidi* at the moment has no solution for this special case.

The analogy with mixing "normal" audio files should be noticed: a mxing process for audio files does not take care of tempo, tonality or measure information simply because audio files (WAVE, MP3 etc.) do not have these information.

When you mix MIDI files you may however encounter additional problems:

- MIDI files may have different time bases (in MIDI called "division"). It is thus necessary to normalize the time bases before you mix the files.

- MIDI has no bar information. When you say "mix in file two after bar 5 / beat 3" then you use information that you (the human user) have but that is lacking in a MIDI file. Bar/beat information must be reconstructed in MIDI files - this process can produce errors (see my short article on the anacrusis problem)
- MIDI has relative (not absolute) time information for the note events. As the mixing process may introduce new notes in the sequence of the the existing notes you must modify the time information for the events. This may lead to unexpected results.
- When you mix two large MIDI files then the resulting mixed file may show too many tracks. I prefer therefor to mix rather channels than tracks (channels and tracks are not the same!), i.e. if the first file has events in channel 5 and the second file also then all events extracted from both files for channel 5 are combined in the mix file on channel 5 - no additional track is created.