

Ardour MIDI Editing

Revision 2

Thorsten Wilms t_w_@freenet.de

2007-05-27

Table of Contents

1 <u>Introduction</u>	3	5.2 <u>Conventions</u>	7
2 <u>Why MIDI in Ardour?</u>	3	5.3 <u>Modality</u>	7
3 <u>Note Event Operations</u>	3	5.4 <u>Overload</u>	7
3.1 <u>Random Access Operations</u>	3	5.5 <u>Strategies</u>	8
3.1.1 <u>Single Note Events</u>	4	5.5.1 <u>One tool per operation</u>	8
3.1.2 <u>Single or Multiple Events</u>	4	5.5.2 <u>Quasimodes</u>	8
3.1.3 <u>Multiple Events</u>	5	5.5.3 <u>Selection, Action</u>	8
3.2 <u>Real-time and Sequential Access</u>		5.5.4 <u>Keys instead of clicks</u>	8
<u>Operations</u>	5	5.5.5 <u>Smart Dragging</u>	9
3.3 <u>Frequency of Use</u>	5	6 <u>Keyboard Control</u>	9
4 <u>Representation</u>	6	6.1 <u>MIDI Cursor</u>	9
4.1 <u>Velocity</u>	6	6.2 <u>On and Off</u>	9
4.2 <u>Drums vs Piano</u>	6	6.3 <u>Navigation</u>	9
5 <u>Mouse Control</u>	6	6.4 <u>Add Notes</u>	9
5.1 <u>Target Areas</u>	6	6.5 <u>Clipboard overlay</u>	9

1 Introduction

Ardour is as of now an audio-only sequencer, but this years Google Summer of Code project by Dave Robillard will add MIDI sequencing capabilities. He prepared the infrastructure in a GSoC 2006 project.

Regarding interaction and interface design, graphical editing of MIDI is the most interesting and crucial aspect. For now I will address editing of note events, exploring the problem and solutions space.

The central question is how to make the operations necessary to edit note events available in the most usable way.

2 Why MIDI in Ardour?

Some people stated a preference for Ardour staying audio-only. They are concerned about added complexity, the required development effort and either don't need MIDI themselves or think it can be handled with separate applications.

It's common in modern music production to combine audio like drum-loops with MIDI sequenced

lines. In such a case, having 2 time-lines in 2 separate windows and applications that need to be synced is not a solution, but a problem. Being able to edit one part while seeing how it relates to another is fundamental. This is also an argument for on-canvas MIDI editing. Another is avoiding additional windows the user has to deal with.

The existing open-source audio/MIDI sequencers are all no match for Ardour's audio capabilities and none of them has on-canvas MIDI editing, so there's a gap to fill.

3 Note Event Operations

Note events can be manipulated in a number ways depending on selection and the mode of access.

3.1 Random Access Operations

Operations on note events that can and have to happen independently of transport state / playback:

- Add
- Edit
 - Move (time)
 - Transpose

- Adjust duration
- Adjust velocity (for both note ons and offs)
- Delete
- Audition

3.1.1 Single Note Events

Operations applicable only to single note events, listed with parameters and possible sources:

- Add
 - Location
 - Mouse location
 - Edit-cursor location
 - Duration
 - Drag (with or without modifier, free or with snapping)
 - Holding a key for the duration (mouse, keyboard or MIDI input)
 - Default value
 - Note number
 - Mouse cursor location
 - Edit-cursor position (assuming a cursor that has a note axis position, not time-line only)
 - MIDI input

- Alphanumeric input (note number, name and octave, tracker style keymapping)
- Default value
- Velocity (note on / off)
 - Default value
 - Drag (with or without modifier, from or in a special area)
 - MIDI input
 - Alphanumeric input

3.1.2 Single or Multiple Events

Operations applicable to single or several selected events:

- Delete
- Move
- Adjust duration (can be seen as moving note ons or offs)
- Transpose (absolute or in a scale)
- Cut (copy to clipboard and clear selection)
- Copy
- Paste
- Cut span (notes and empty space)
- Copy span (notes and empty space)
- Paste span (notes and empty space)

- Delete span (move following to leave no gap)
- Trim (only keep selection)
- Split notes at location
- Clear span (notes partly in will be cut off)
- Divide n-times

3.1.3 Multiple Events

Operations applicable only to multiple events:

- Time-stretching
- Shrink / stretch note range
- Mirror on time axis
- Mirror on pitch axis

3.2 Real-time and Sequential Access Operations

Operations that could be applied to events as they are played back or stepped through:

- Delete
- Cut out
- Delay (until the trigger is released)
- Overdub / Replace
- Insert (combined record and delay)
- Overwrite single parameter (example: note number)

- Manipulate single parameter (example: transpose notes)

3.3 Frequency of Use

Depending on the current task, the different operations are more or less frequently needed. Most frequently listed first, based on my own experience.

If not all basic operations can be made available in a straightforward way, it will have to be considered to model modes based on these sets.

Editing real-time recorded MIDI, following the question of what kind of errors need to be fixed most often:

- Move
- Adjust duration
- Adjust velocity
- Delete
- Add

Programming MIDI. Add is clearly most important, as without it, there's nothing. Otherwise the order is

- Add
- Move

- Delete
- Adjust duration
- Adjust velocity

Tweaking MIDI:

- Adjust duration
- Adjust velocity
- Move

4 Representation

The horizontal time-line is a given an only the matrix / piano-roll approach with bars representing note length allows to put audio and MIDI into relation.

4.1 Velocity

Sensible representations of velocity values:

- On notes
 - Numeric value
 - Colour (hue or brightness)
- Overlay: bars on the same canvas area. Dealing with polyphony is tricky.
- Special area. The common row below the note area. Comparable to automation tracks.

4.2 Drums vs Piano

Many sequencers have special drum tracks / views.

A drum editor can do away with having rows represent notes in continuation and instead allow re-ordering and assigning ports and channels per row.

As note length tends to be of no concern for percussive instruments, symbols that denote a location but no duration are used, often a rhomb.

Drum and piano style editing could be combined by always allowing rows to be reordered, having optional per row port and channel settings and using both note bars and rhombs (or similar symbol, understood as alias for notes of an adjustable equal length).

5 Mouse Control

5.1 Target Areas

Besides moving notes, it must also be possible to resize them, which can be thought of as moving either note ons or offs.

Short notes on common zoom levels are likely too small to offer target areas (handles) for 2 (whole note, note off) or even 3 operations (whole, on, off).

Approaches:

1. The user can always zoom in if he has to edit short notes
2. Resizing handles that extend to outside the actual note. But how to avoid overlap with other notes and handles?
3. Separate tools / modes for moving whole notes vs note ons and offs.

5.2 Conventions

Conventions found across several platforms applications do not just set user expectations, but encourage habituation. Habits are hard to change.

Conventions that should be followed:

- Left-clicking an item toggles its selection state (and deselects other items)
- Ctrl-left-clicking toggles item selection state without deselecting other items (items can be added to the selection)
- Shift-left-clicking allows the selection of a span of items
- Left-click-dragging from an empty space does rubber band selection

- Right-clicking opens a context menu

5.3 Modality

To make the operations necessary for editing MIDI available via the mouse, usually a number of modes is introduced in the form of tools.

Modes add context the user needs to keep in mind, as the same actions will not always lead to the same result anymore. Mode switching also leads to additional steps to be taken.

Therefore it is desirable to minimize modality.

Jeff Raskin proposed the use of quasimodes, where a mode is kept in place only as long as a key is held down. This way the user can hardly forget about it. Holding a key for a longer duration is tiresome, though.

Quasimodes should be considered for short tasks, but avoided for long tasks.

5.4 Overload

The most efficient way to add notes via mouse would be to just click where they shall appear. Or click-drag to also define their length. Just clicking on a note or dragging from empty space are expected

to work for selecting. Moving notes would be most simple by just dragging.

5.5 Strategies

5.5.1 One tool per operation

A set of tools like it can be found in some sequencers and many graphics applications.

Modifier keys could be used to temporarily switch to other tools, or to switch to a sub-mode.

Straightforward, but having to switch tools is costly and icons representing the tools take space, but should be present for discoverability.

5.5.2 Quasimodes

Use modifiers and mouse buttons to have one main mode and everything else as quasimodes. One could even differentiate between modifier keys being pressed down before or after a drag is initiated, as done in GIMP.

Could be very efficient after training, but hard to learn / memorize. Depends on documentation and users reading it . A way out could be status bar mes-

sages, but there's not even a status bar and all space is precious.

5.5.3 Selection, Action

Use the mouse to select single notes or to draw selection rectangles only. All other actions work on the selection and could be triggered by shortcuts, context menu or GUI buttons. Adding a note would happen by filling the selection (requires that white space is selectable).

Simple and fast selection in all cases (rubberbanding can start on notes), but not being able to directly draw notes or to drag them around is quite a drawback.

5.5.4 Keys instead of clicks

Use mouse for position and trigger actions with keys. Example: A to add notes.

Hard to learn. Difficult to find enough free and matching keys. Requires a one hand on mouse, one on keyboard approach for everything.

5.5.5 Smart Dragging

Use dragging from empty space for rubber band selection. Fill the selection with a note if no other note is completely covered. Use dragging on a note for moving it (or all selected notes).

This would make drawing notes that overlap other notes completely a little complicated: draw elsewhere, move. I think it's save to assume such overlaps are very rarely (if ever) wanted, so this wouldn't be a problem.

6 Keyboard Control

Just some rough ideas. Needs to be evaluated in relation to mouse control.

6.1 MIDI Cursor

As the term edit cursor is already taken for a special location mark, I will call this concept a MIDI cursor. The underlying idea is handling MIDI more like text in an editor. It marks a location on the time-line and a note row.

6.2 On and Off

Treat notes ons and offs as separately selectable items. A note is selected if both on and off are selected.

6.3 Navigation

Setting vertical location absolute: Shift-A, B, C, D, E, F, G, with # as toggle. Setting vertical location relative: Cursor keys up / down. Moving horizontally: Cursor-keys left / right.

Ctrl for bigger steps, Alt for fractions Shift for selection span

6.4 Add Notes

Add note at current position: Enter. Add note by name: A, B, C, D, E, F, G, with # as toggle.
Edit Notes

6.5 Clipboard overlay

Copy/Cut results in an overlay of the notes attached to the edit cursor.