Portability

David Adamcyk

Advanced Mixed Music Composition
Definition of “portability”:

It is the degree to which the electronic part of a mixed piece can be:

a) transferred to another location and performed by other people other than the composer

b) updated so that it can still work as technology evolves.

The degree of portability of a piece is a function of:

a) The choice of devices used in the piece and their:
   • Availability
   • Sensitivity to technological changes
   • Sensitivity to the use of substitute components and/or to changes in the performance environment

b) The required musical precision/repeatability of the electronic part

c) The quality of the documentation

d) The precision of the notation of the electronics
Aspects of documentation:

• Technical “rider”
• Description of electronics in the score
• “User friendlyness” of custom software
Technical rider

• Describes all the technical requirement of the piece, is as much detail as possible
• Serves as a document to help understand the piece, but also as a punch list of things to bring come concert time.
• Assume the person reading the rider has little knowledge of electronics
Description of the Electronics in the Score

• Helps the conductor/performer(s) figure out what’s going on in the electronics

• Can have an impact on whether your piece is selected for a performance

• Can be the same as the technical rider or can be a reduced version with references to the technical rider
Custom Software Design

• Should be simple for unfamiliar users

• Two options:
  a) Um-modifiable
     > Must work perfectly (no bugs, good interface, etc)
  b) Modifiable
     > Less complicated as it doesn’t have to account for different devices

• The software should have:
  a) Clear launch sequence
  b) Easy to follow DSP chain
  c) Easily readable events
  d) Clear event navigation and visual feedback
  e) Clear input/output meters and faders
A few remarks about Max patches

- Meters should be in dB (but internal level messages can be linear)
- Clear DSP chain
- Have speaker tester
- Have EQ at the input and output
- Have a good way to save your start levels
- Have a way to quickly choose all your MIDI devices
- Test it on different computers
- Have key function for usual tasks: init, panic, stop audio files, reset, stop DSP, reset MIDI, go to events, etc.
- Don’t use the preset object unless you save the info to files
- Use send object instead of send~ when possible
- Have a prevent window close function
- If your patch is DSP heavy, avoid excessive use of visual objects (meters, toggles, led, bangs, etc.)
A few remarks about Max patches (cont.)

• Have easy way to change MIDI ctlin values, as well as a way to change pedal polarity
• Allow to mix SF and RT independently (this can also be done on the mixer, but requires more interface outputs (and therefore takes up more mixer inputs)
• If you’re working on a large screen, make sure to reduce the size of patch windows for typical laptops computer screens
• Consider making an input signal calibration tool, so that you can ensure you have the same level as your simulation