

HAIL – Notes

Sebastian Fichtner

1 Audio Engine

Possible ways of scheduling in a sequencer:

- subclass `AEAUDIOFilePlayer`
- use `AEAUDIOUnitChannel` with the `AUFilePlayer` and CA scheduling
- write a new `AudioFilePlayer` as an `AEAUDIOPlayable`
- write a new `AudioFilePlayer` as an `AEAUDIOPlayable` that uses an `AEAUDIO` file player internally

2 Intuitiv Identifizierte Probleme

Der Anlass, ein Konzept für Musiksoftware zu entwickeln stammt aus den zum Teil frustrierenden Erfahrungen die befreundete Musiker und ich mit Programmen wie Logic Pro, Reason, Live, virtuellen Instrumenten, Samplern und Audio Editing Tools gemacht haben. Im folgenden werden Probleme aufgelistet die typisch für das Arbeiten mit existierender Software sind und damit nach einem neuen Ansatz verlangen. Mögliche Lösungen ergeben sich meist schon direkt aus der Problemfeststellung.

Die Liste entstand, so wie sie hier formuliert ist, als Initialzündung des ganzen Projektes und basierte nur auf Erfahrung und Intuition. In die Masterarbeit fließen vor allem die ins Englische übertragenen Punkte ein.

2.1 Broken Interaction

1. The user is presented with countless buttons and options, which has the following implications:
 - Comprehension and overview are impeded.
 - Only a small fraction of interaction options is actually used. Users do the same actions repeatedly. Thus, unused options waste screen space.
 - Screen space is further wasted by window- and option bars of applications and the operating system.
 - The user gets distracted from his actual creative work.
2. The dimensions of GUI-elements are fixed sizes in pixels. With high screen resolution, they become hard to read and hard to interact with.
3. The GUI is based on the WIMP-schema.
(Window, Icons, Menus, Pointers)
4. The app spreads over lots of windows which must be arranged by the user. (arrange-view, mixer, signal routing, audio files, instruments, sampler, MIDI editor, external applications connected through ReWire ...)

5. When editing (arranging) content, the user often has to zoom and scroll horizontally and vertically. That consumes time and impedes overview. (Example: 4 input elements in the arrange-view of Logic Pro)

2.2 Broken Functional Model

1. Metaphors are not related to music itself but to real studio hardware. That causes unnecessary indirections and restrictions and is also an antiquated approach.
2. Arranging, MIDI editing, creating sampler instruments and similar tasks are presented as profoundly different, in different windows, with different metaphors and interaction concepts. The reasons for that are more technical and less content related. The user is confronted with artificial and superfluous distinctions because these tasks share the same principles.
3. For the presentation of content, hardly any data gets aggregated. A consistent hierarchical model does not exist. Hierarchical approaches remain shallow, cumbersome and irrelevant to the overarching concept, although music is hierarchically structured (for example: recording, tone, chord, bar, cadence, song part, song, album).
4. The way that audio material goes from recording to arranging and output is modeled clumsily and presented inconsistently and unclear.
5. Project files are not based on an open XML format. Projects can only be edited with the software with which they were created. It is impossible to just edit them in a text editor. Other developers cannot program additional editors.
6. When mixing, the volume of a single source can only be changed absolutely not relatively, often leading to a mix that does break or not exploit the available volume range. Such a mix must then be corrected by adjusting every single source that contributes to the mix. During the creative phase, which takes much longer than the final mixing, these adjustments are an annoying waste of time.
7. The graphical display of volumes as absolute levels misleads the user to make a rather abstract judgement of the sound. The evaluation of a

mix should be done with the ears and not be biased by the knowledge of gain levels. That is why experienced sound engineers often mix with their eyes closed.

8. Creating chords from your own recordings and then treating them as atomic tones is impossible or cumbersome.
9. Working with your own samples or even creating sample instruments is cumbersome, heavily application dependent and demands quite some learning.

2.3 Kollaboration

1. Das kollaborative Arbeiten an Projekten, etwa in Bands, wird nicht unterstützt, weil mindestens eine der folgenden Voraussetzungen nicht erfüllt ist:
 - Alle können und wollen sich die Software leisten.
 - Auf allen Rechnern ist die Software lauffähig.
 - Alle können die Software bedienen und haben auch Spaß daran.
 - Konsolidieren verschiedener Projekte/Ideen ist für alle einfach möglich.
 - Die Software unterstützt kollaborative Entscheidungsfindung.
2. Samplebibliotheken und Projekte sind dezentral auf jedem Benutzerrechner gespeichert, mit folgenden Implikationen:
 - Lange Installationszeiten bei jedem Neuerwerb oder Neuaufsetzen des Rechners
 - Hoher Speicherplatzverbrauch, obwohl nur ein Bruchteil der Samples benutzt wird
 - Verlust hoher ideeller- und teilweise kommerzieller Werte bei Rechnerschaden
 - Umständliche Distribution für Künstler die Samples anbieten wollen
 - Fehlender Überblick über Angebot von Sample-Bibliotheken, geringe Auswahl, kein Erwerb einzelner Samples
 - Erschwerte Netz-Kooperation

- Bearbeiten eines Projektes von verschiedenen (mobilen!) Endgeräten ist unmöglich oder sehr umständlich.
3. Exportieren, Importieren und vor allem das Kombinieren verschiedener Teile aus verschiedenen Projekten ist unmöglich oder sehr umständlich.
 4. Das Mitnehmen (USB-Stick) oder Mailen der Software und sofortige Ausführen auf anderen Rechnern ist aus vielen Gründen unmöglich.

2.4 Umgang mit Audiodaten

1. Das Management der tatsächlich auf der Festplatte liegenden Klangdateien ist umständlich, unzureichend und zu wenig in den Arbeitsfluss integriert. Reale Daten und Referenzen werden unzureichend unterschieden. Das Volumen ungenutzter Daten wächst, ohne manuelle Verwaltung über das Betriebssystem (Finder, Explorer), unbemerkt an.
2. Trotz Undo-Funktion sind Audibearbeitungen meist destruktiv und laden nicht zum Experimentieren ein. (Bsp: Reverse, Silence, Gain, Pitch in Logic Pro)
3. Automatisches adaptives Vorberechnen der Klangbausteine eines Projektes wird nicht durchgeführt, unter anderem, weil kein hierarchischer Aufbau existiert. Es wird höchstens eine Freeze-Option für ganze Spuren angeboten, die manuell benutzt werden muss.
4. Audio Stretching im Zusammenhang mit Tempoänderungen ist nicht automatisch, destruktiv und umständlich.