Background: Sonic Sketching

Introduction

This chapter begins with a discussion on the dominant metaphors present in the modern DAW and expands on the earlier introduction to the topic. A compositional example is given to illuminate the issues and limitations that these metaphors can impose on its users. This is followed up with a survey of legacy systems that take a graphical approach to their interface and can be conceptually framed with the metaphor of sketching.

Dominant DAW metaphors

As has been introduced, analog studio metaphors of tape machines and hardware mixing desks dominate the UI approach to DAW interface design. Other prevalent metaphors often found in these interfaces are that of the *outboard effects units*, and the *piano roll* [Levin, 2000; Bell, Hein and Ratcliffe, 2015]. A description of these now follows.

Multitrack tape recorder

Multitrack tape recording was introduced into the recording studios in the 1960s and is typified by the systems produced by Ampex and Studer. These allowed producers to record multiple *tracks* of audio information which could be edited and mixed to taste. This unlocked significant creative possibilities and made albums like "Sgt Peppers Lonely Heart Club Band" possible. The underlying model of tracks typically manifests itself in DAWs as rectangular blocks stacked from top to bottom and running from left to right. Similar to editing tape, these can be spliced, cut, and pasted. Terms and techniques prevelant in DAWS like *bouncing*, *overdubbing* and *markers* (which were originally created using a physical pen), all have their roots in their analog precedents.

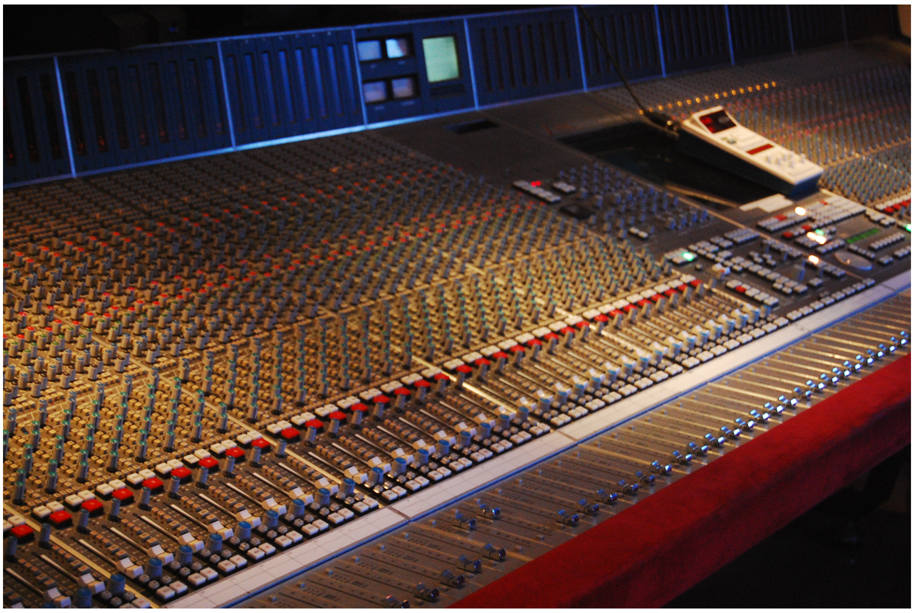


Figure 1: SSL SL9000J (72 channel) console at Cutting Room Recording Studio, NYC

Multichannel mixing desk

The multichannel mixing desk metaphor is present in the large majority of DAWs and is normally represented in a similar fashion to the sliders (or faders) found in hardware mixing desks (see figure 1). The mixing desk enabled the producer to control the relative amplitude of a finite amount of channels in addition to performing tasks such as panning to balance the signal in a stereo field. The slim vertical sliders found on most systems were codified in the 1960's by Bill Putnam [Bell, Hein and Ratcliffe, 2015]. This layout allowed the producer to manipulate multiple channels of audio simultaneously in a practice known variously as "riding the faders" and "playing the mixer". Despite the fact that the digital variants of these are largely controlled by a mouse that only affords the manipulation of a single fader at a time, they are still, largely speaking, presented in this fashion on screen.

Outboard effects unit

Outboard effects are hardware units used in studios to add audio effects to one or more channels on the mixing desk. The standard configuration of most studios allows for two different ways of applying these effects, by using insert effects and send effects. Insert effects are typically used when it only needs to affect a single channel, for

Figure 2: Skeuomorphic software FX

instance, a chorus effect applied on an electric guitar. Send effects allow the producer to send a certain amount of the signal from a channel to specialised channels to perform processing on the signal in parallel with the original signal. It is typically used to apply effects on multiple channels of audio, such as reverb. The introduction of Virtual Studio Technology (VST), by Steinberg (https://www.steinberg.net), was responsible for bringing the outboard effects metaphor to an even greater level of sophistication. This allowed third party developers to create virtual effects and instruments, and let producers expand their virtual studio beyond the built in effects. The visual interfaces increasingly paid homage to their hardware influences, emulating not only functionality but also the visual look (see figure 2). Bell et al. [2015] describe this as skeuomorphism, a design pattern where visual objects not only mimic real world objects in functionality but also incorporate superfluous visual features. The purpose of this is not only decorative but also educational, and gives connotational cues as to how it should be used.

The piano roll

The piano roll is a primary metaphor found in almost all mainstream DAWS and is typically used to represent MIDI musical note information. MIDI, which stands for Musical Instrument Digital Interface, is a standard protocol developed in the 1980s to allow control instructions to be sent between devices. It provided a standard language to, for instance, tell a synthesizer to play a particular note at a precise time and duration. These instructions could be collected into a MIDI file to, in effect, create a playable digital score. The piano roll is slightly distinct from the previously discussed examples in that it originates from a much earlier time period, the player pianos of the 1920s. The original piano rolls were operated by feeding a roll of paper with holes punched to indicate the precise timing that an attached piano should strike its notes. This provides an apt and suitable description for the MIDI musical data it normally represents. Similar to a player piano, no audible results are possible without an attached piano, and in the case of MIDI, an attached sound generating synthesizer device [Levin, 2000; Bell, Hein and Ratcliffe, 2015].

A compositional example

Rather than discussing the issues that can arise from the metaphors in the abstract, let us consider a compositional idea and how we might achieve this in a DAW. The idea is broken into the following compositional "recipe":

1. Two notes of the same timbre are played together about an octave apart for a duration of 2 seconds.
2. The first note glissandos to the frequency of the second note and vice versa.
3. The first note starts with a small amount of vibrato that quickly dissipates.
4. The second note starts with no vibrato but adds a small amount as the note nears completion.
5. When these two notes end, the same pattern is repeated except this time with different timbres and frequencies.
6. This is repeated 3 more times with different timbres and frequencies to complete this ten-second piece.

While this may seem like a contrived example this, in fact, constitutes a compositional technique called Klangfarbenmelodie [Cramer, 2002] that involves splitting a melodic line between instruments or timbres to create a timbre melody. The glissandos and altering of vibrato intensity add further complexity to better illustrate some of the weaknesses inherent in DAW metaphors.

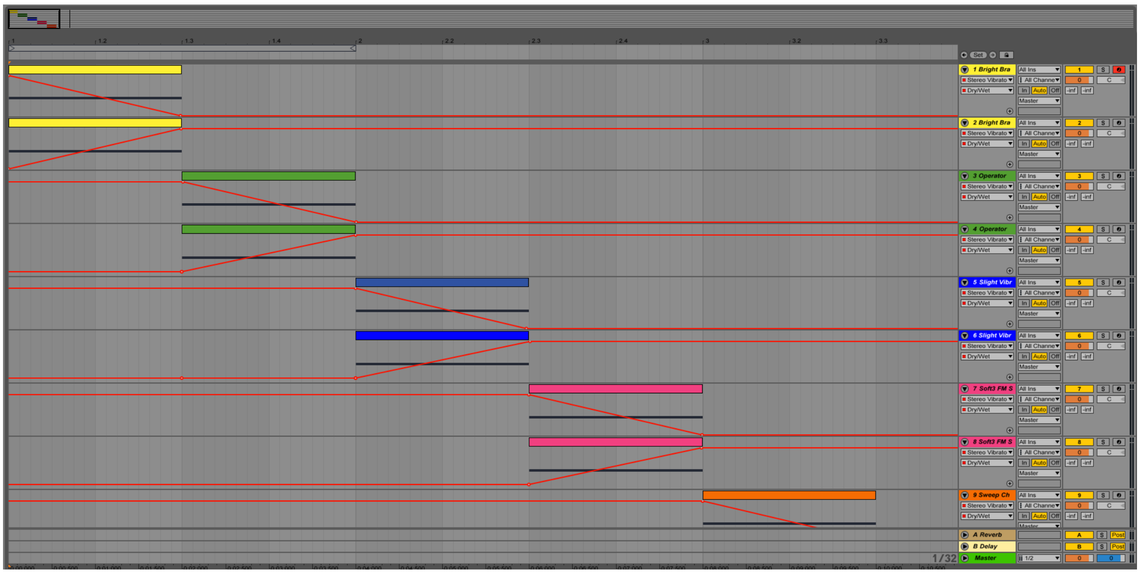
Realization in a DAW

To achieve this in a DAW we have a few different options but a possible solution would be as follows:

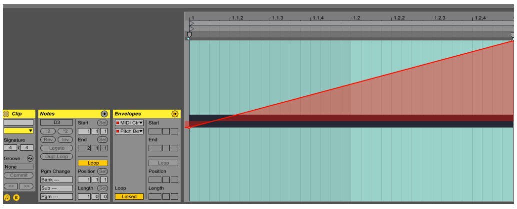
1. Working with the multitrack tape metaphor we can create ten separate tracks to house two different versions of each timbre. A vibrato plugin effect should be added to each of these by using a send or an insert effect. Two different tracks are needed for each of the timbres due to the fact that the two notes are played at the same time and both have different frequency and effect trajectories. If on the other hand, they had the same effect modulations or were played at different times, no additional tracks would be needed.
2. Working with the piano roll metaphor, create a single note in each of these tracks setting each one to the desired fundamental frequency.
3. Now edit the pitch bend automation lane by clicking into the relevant dialog
4. Similarly, open the relevant dialog to edit the intensity of the vibrato effect
5. Repeat this for each of the notes in the composition.

At this point, we may have achieved what we set out to do. However, we now may want to tweak each of these elements to taste and perhaps add more material. An explosion in track count and overall complexity is inevitable. This can lead to a slowdown in workflow, a loss of flow and cognitive overload. A common technique to combat this complexity overload is to bounce the tracks and then continue working on these simpler artifacts [Duignan, 2008]. This, of course, negates a key advantage to working in a digital environment, the fine-grained ability to freely change, tweak and undo. Locating each note in separate tracks leads to an unnatural separation of what is, in fact, closely related compositional material. This requires awkward context switching and excessive navigation through the system to focus on different details.

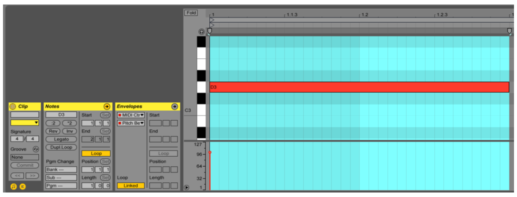
There are of course other tools in the DAW that may achieve this task more easily. For instance, a sampler may allow us to use different timbres on the same track and may work better in this case. We now have the extra task of exporting each of these samples in preparation for our composition work. Some other options present in many DAWS include aggregate instruments, multi-timbral instruments, and perhaps some midi routing options. Another option is to use an alternative, more flexible, environment such as an audio programming language. Some brief consideration of this will now be given.



(a) The timeline showing the vibrato automation on each track.



(b) The pitch envelope. Requires multiple clicks to display.



(c) Each note is contained in its own timeline clip.

(d) The instrument and effect chain. The small red dot on the DRY/WET knob indicates automation.

Figure 3: DAW realisation of composition

Realisation in code

The piece could be realised in quite a straightforward manner in an audio programming language such as *Csound* (http://www.csounds.com/). Central to *Csound* is the concept of the *unit generator* (or ugen), an abstraction to define both sound generators and processors. These can be patched together in a simple textual coding language to form instruments. A score is then specified, again in code, to define note onsets, durations in addition to other arbitrary parameters defined in the instruments. The required timbres and the vibrato effect could be made configurable on a per note basis by exposing these parameters. The Csound score could then trigger this instrument, with each note amounting to a single line of code, making the entire score a total of 10 lines. Full demonstration code is provided in the appendices. This compositional example will be revisited in a later chapter and discussed in the context of a further approach.

Table 1: CSound score represented as a table. Each row contains the data for each note.

Instrument id	Start time	Duration	Start freqeuncy	End frequency	Timbre	Start vibrato level	End vibrato level
1	0	2	440	880	3	.01	20
1	0	2	880	440	3	20	0.01
1	2	2	1320	660	6	0.01	20
1	2	2	660	1320	6	20	0.01
1	4	2	330	660	2	.01	20
1	4	2	660	330	2	20	.01
1	6	2	880	1720	1	.01	20
1	6	2	1720	880	1	20	.01
1	8	2	220	440	9	.01	20
1	8	2	440	220	9	20	.01

Problems with flexible systems

Depending on the experience of the reader, the realisation of the composition in code may or may not seem like a better approach than using a DAW. The reason for this is that this approach is not beginner friendly. An approach that is more forgiving in this regard is a pattern found in game design [Overholt, 2009]. Games should be easy enough to get started without any special training or lengthy instructions but challenging enough to keep players engaged. Extremely open environments, such as that of an audio programming language, are not supportive of this initial onboarding of new users. This is not to say that it should only work for novices however (by for instance limiting pitches to simple scales). If a system is too closed it risks being more toy-like in nature and not supporting long term engagement [Wessel and Wright, 2017].

Perhaps a bigger criticism that could be made about open and complex systems, however, is that they can lead to an analytical rather than a creative way of thinking. In “Thinking Slow, Acting Fast”, Kahneman [2012] contrasts these two ways of thinking which he terms *System 1* and *System 2*. System 1 is instinctive, fast, emotional and is a mode of thinking that may not register consciously. System 2 is slow, logical, analytical and registers prominently in active consciousness. Routine tasks such as walking, opening doors etc only use system 1 thinking. These can be completed while exerting minimal cognitive effort (all the while calculating the complex motor sensory actions that must take place). Complex analytical tasks such as programming require system 2 thinking. Approaching creative tasks such as music making in this way where instinct and emotion are often crucial can slow down or stop the process. Perhaps it is best summed by John Cage: “Don’t try to create and analyse at the same time. They’re different processes” [Popova, 2012].

Sketching as an alternative metaphor

Audio programming languages offer a model that is closer to the underlying computational processes taking place than the more abstracted DAW interfaces. As we have discussed, though what is gained in flexibility can be lost in intuitiveness and ease of interaction. Rather than discarding these higher level metaphors, perhaps a better approach would be to explore alternate ones.

A rather promising but less established approach is that of sonic sketching. This has a long and illustrious historical precedent reaching back well before the, now more prevalent, studio metaphors. Graphical sound generation techniques have a long history starting with experiments beginning in the early 20th century [Roads, 1996 pg. 329]. The technique of the optical soundtrack, however, brought these ideas to a new level of sophistication. The technique, which involved placing marks via photography or direct manipulation to specify audio properties, was explored by such luminaries as Oskar Fischinger, Norman McLaren and Daphne Oram. Oram's particular take on the technique will now be discussed.

Oramics

Figure 4: Daphne Oram's Oramics machine

A primary motivating factor behind Daphne Oram's development of the Oramics machine was to bring more human-like qualities to the sounds generated by electronic means. The machine worked by playing back multiple lanes of film tape in unison, defining a monophonic series of notes as well as control signals to shape their timbre, pitch and amplitude. She details the thought process behind this in her journal style book, "An Individual Note" [Oram, 1972].

The aspects of the sound that she wishes to control are volume, duration, timbre, pitch, vibrato, and reverb. In order to do this, she describes a simple musical notation language based on the freehand drawing of lines combined with discrete symbols. The lines, which she describes as the analog control, are used to define volume envelopes. Interestingly, the default and preferred method for the parameters she wishes to control is the continuous line rather than discrete note symbols. For instance, she avoids the use of a static velocity per note and instead only specifies the use of a control envelope to change amplitude.

The discrete symbols, which she categorizes as digital control, are used to define individual pitches and are termed neumes[1]. She highlights that notes should not

[1] The term for the ancestor of the note modern western notation, which was much simpler in use and didn't specify rhythm

remain static and, thusly, an analog control of each note is also specified. Similarly to amplitude and vibrato, timbre is also defined by the freehand drawing of lines and is something that with practice the “inner ear” can develop an intuition as the sonic results of different line shapes. It is Oram’s belief that the hand drawn nature of the lines make the results slightly inaccurate and to some extent unpredictable. Herein, however, lies the possibility of bringing more humanity to the cold and precise machines generating the electronic signal.

UPIC

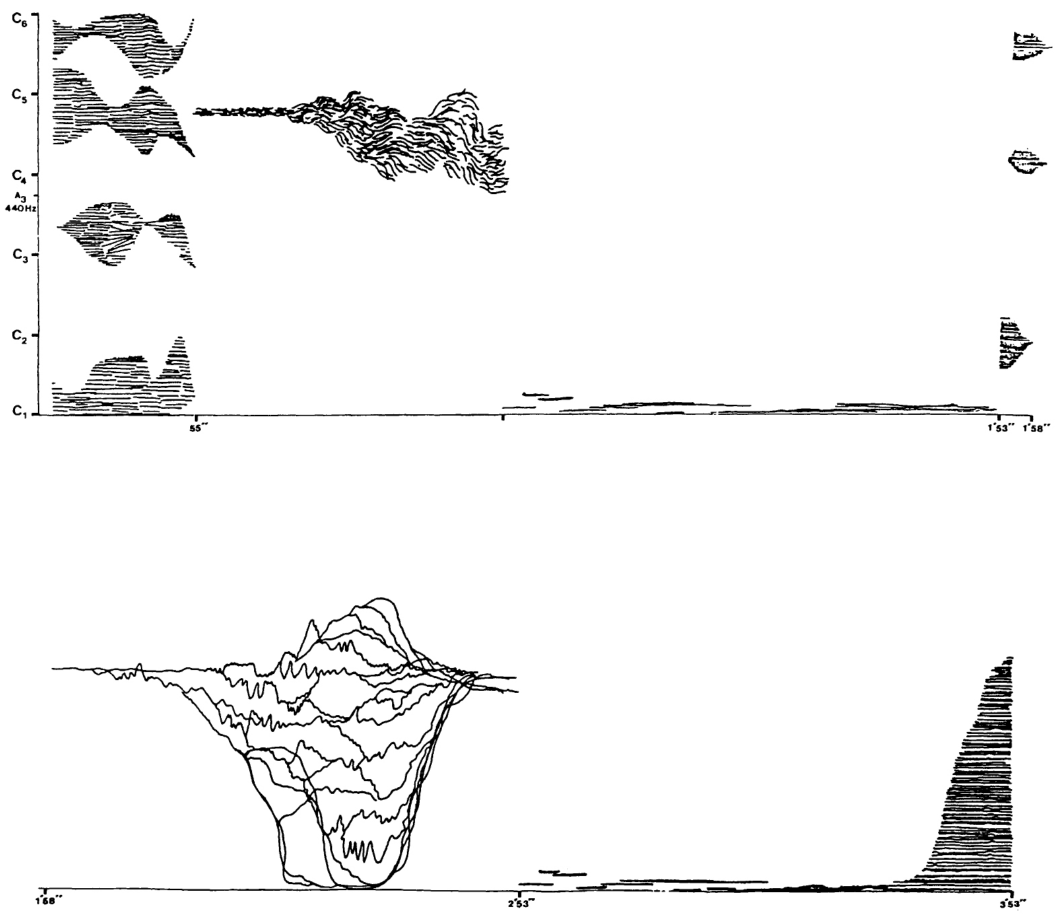


Figure 5: Iannis Xenakis - Mycenae Alpha score (Nunzio, 2014)

The UPIC (“Unité polyagogique informatique du CEMAMU”) was a graphic sound synthesis system that was designed by Iannis Xenakis and arose from his graphic approach to composition. His earliest work, “Metastasis”, was conceived using a graphic approach to describe the trajectories and sound masses that inhabits the orchestral landscape of the piece. This approach has been attributed to his

Figure 6: Iannis Xenakis showing UPIC to a younger audience (Nunzio, 2014)

background in architecture, having worked in the studio of Le Corbusier. The UPIC was first conceived of in the seventies with the realisation of the first version in 1975 and its first public showcase in 1977 [Roads, 1996 pg. 331]. The work "Mycanae Alpha" (figure 5), composed in 1978 was the first work to use the system and was a "nine-minute 38-second composition of dense and intense textures, of phase-shifting waveforms rich in harmonics that cascade, flutter, crash, and scream like sirens in a vast cosmological territory" [Tyranny, 2017] .

The early version of the UPIC worked by drawing on a large digitizing graphics tablet which was interpreted by a high-powered computer (for that period) and converted into audio signals. The graphic approach to sound specification worked on a synthesis level by allowing the composer to draw and audition waveforms. Larger structures could be drawn in by switching to a "score" page and drawing lines, or "arcs" as they were denoted, on a pitch-time canvas. The final version of the application ran on personal computers and allowed for real-time interaction with a 64 oscillator synthesizer. At this stage, the input means had changed to a computer mouse but nevertheless retained the graphic approach of interaction. [Nunzio, 2014]

A primary goal of the UPIC project was that of pedagogy. Xenakis reasoned that the universality of sketching meant that it could provide an excellent teaching tool for a wide audience, even for young children (figure 6). Another goal of the system was to encourage composer autonomy. At the time of its conception in the seventies, the technical barrier to entry into electronic music creation was very high and interfaces to help with this were rare or non-existent. Though the UPIC is not available to the general public currently, it has inspired a number of other systems that are available today. [Nunzio, 2014]

A Golan Levin's AVES

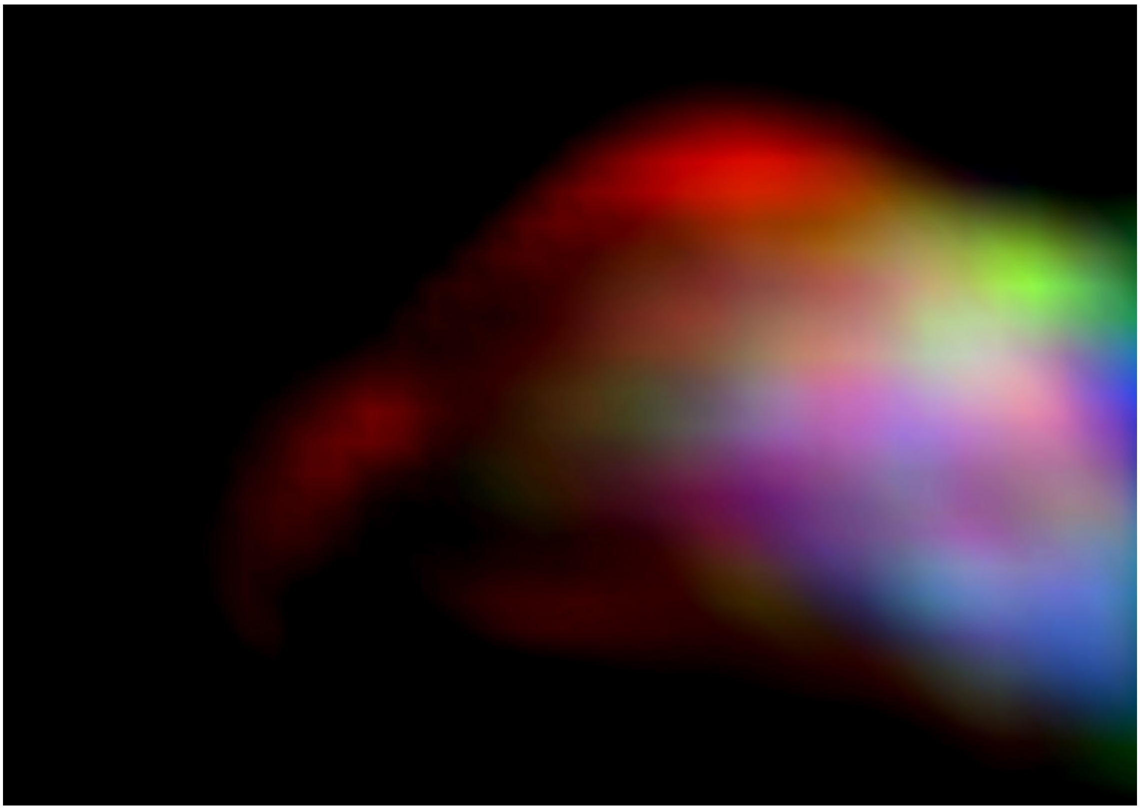


Figure 7: Golan Levin's Aurora (part of AVES)

Golan Levin created the interactive audio-visual system, AVES, in the late nineties. This series of audio visual installations represented a landmark in the field of visual music. It is an attempt to move away from the diagrammatic approach to musical interfaces and to present an interface that is "painterly" in approach. Taking strong influence from visual artists such as Paul Klee, he presents a system that maps user input from a graphics tablet and mouse to visuals and audio. The intention is to create a strong visual correlation between these two modalities. A variety of approaches are taken to achieve this, all of them involving an algorithmic approach to a certain degree. For instance, in the piece "Aurora", he maps visuals of vast quantities of particles to a granulated sound synth sound source. He didn't take the approach of an exact mapping of visual particles to audio particles, however, and instead used a statistical control approach to approximate the correlation between the visual and aural. [Levin, 2000]

For Levin, the digital pen input in combination with its infinite variability represents an ideal instrument for creative expression in his digital temporal audio visual paintings. [Levin, 2000] The reason he gives for this is, similar to a musical instrument such as a violin, the pen is instantly knowable in that a child can pick it up and start creating marks but infinitely masterable through practice and hard work, and ultimately a vehicle for creative expression after a certain amount of mastery. A set of criteria that he and John Maeda arrived at to evaluate the success of their experiments was:

- Can you use the instrument with no instructions?
- How long can you use the instrument?
- Does your personality come through?
- Can you get better at using the tool?

– Snibbe and Levin [2000]

Levin's work is largely realtime and transitory in nature with gestures giving rise to visual and audio reactions that rise, fall and dissipate. A description that he uses of some of work is that of creating ripples in a pond. Therefore his work is very much geared towards an instrument like experience. It is not concerned with the recording or visualization of a score or timeline of musical events as would be the function of a compositional tools such as a DAW. Indeed it is a conscious design decision to avoid such representations. Many of the principles and ideas of his work can, however, be applied in the context of a composition tool.

William Coleman's sonicPainter

SonicPainter by William Coleman is a novel musical sequencer that seeks to address some of the shortcomings of traditional approaches to music sequencing found in commercial DAWs [Coleman, 2015]. The focus of the line and node based interface

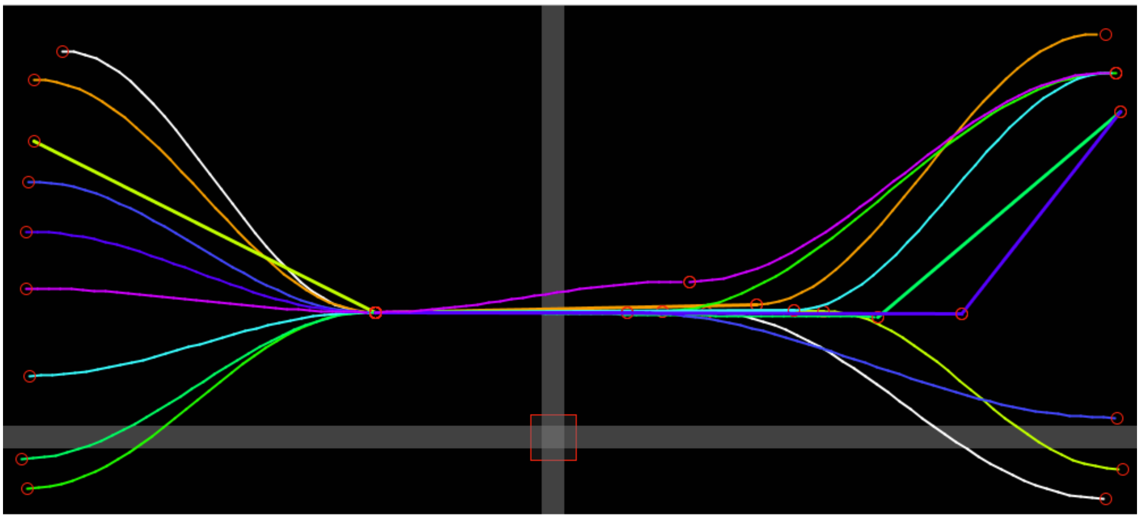


Figure 8: SonicPainter by William Coleman

(see figure 8) is to bring timbral shaping to the fore rather than being hidden away in miscellaneous automation lanes. The design takes influence from legacy musical systems, in particular, the UPIC and incorporates ideas from visual music and embodied cognition.

Similarly to traditional sequencers, the x axis represents time and the y-axis, pitch. Note information is input via keyboard and mouse. A click starts a note and can be followed with additional clicks to continue to shape it. It can be ended by clicking a keyboard shortcut. By drawing notes as lines in this manner, the unfolding of the note can be explicitly represented visually. Other timbral aspects such as vibrato are represented by further visual manipulation of the line. For instance, an overlaid sine wave line indicates the timing and amplitude of the vibrato. In addition, the system allows for freehand input of notes.

Conclusion

The dominant analog studio metaphors present in DAWs were discussed which included details on their origins and their reincarnation in digital form. A short compositional example was given and the process to realise this in a DAW was described. The piano roll, multitrack mixer, and outboard effects metaphors were shown to be a poor fit for this particular compositional idea and resulted in an excessive amount of tracks and, therefore, complexity. A simpler solution was described in the Csound audio programming environment. The lower level abstractions provided here allowed for a more succinct and more straightforward implementation of the piece. Some potential pitfalls to this approach were given. This includes a steep learning curve for novice users and a potential bias towards an analytical rather than a creative mode of thinking. Rather than abandoning the high-level metaphors present in DAWs it was posited that another approach could be to explore other metaphors more suited to certain compositional ideas. To this end, the metaphor of sketching as an interface to audio systems was explored by tracing its early roots in the optical soundtracks of Oram to the realtime synth sketching of Xenakis's UPIC through to the contemporary approaches of Golan Levin's AVES system and William Coleman's SonicPainter.

Bell, A., Hein, E. and Ratcliffe, J. (2015) *Journal on the Art of Record Production » Beyond Skeuomorphism: The Evolution of Music Production Software User Interface Metaphors*. Available at: http://arpjournal.com/beyond-skeuomorphism-the-evolution-of-music-production-software-user-interface-metaphors-2/ (Accessed: 23 April 2017).

Coleman, W. (2015) *sonicPainter: Modifications to the Computer Music Sequencer Inspired by Legacy Composition Systems and Visual Art (MMT thesis).*

Cramer, A. (2002) 'Schoenberg's Klangfarbenmelodie: A Principle of Early Atonal Harmony', *Music Theory Spectrum*, 24(1), pp. 1–34. doi: 10.1525/mts.2002.24.1.1.

Duignan, M. (2008) 'Computer Mediated Music Production: A Study of Abstraction and Activity'. Available at: http://researcharchive.vuw.ac.nz/handle/10063/590 (Accessed: 22 April 2017).

Kahneman, D. (2012) *Thinking, fast and slow*. London: Penguin Books.

Levin, G. (2000) *Painterly interfaces for audiovisual performance*. Massachusetts Institute of Technology. Available at: https://dspace.mit.edu/handle/1721.1/61848 (Accessed: 23 April 2017).

Nunzio, A. D. (2014) *UPIC | musicainformatica.Org*. Available at: http://www.musicainformatica.org/topics/upic.php (Accessed: 18 August 2017).

Oram, D. (1972) *An individual note: Of music, sound and electronics*. London : New York: Galliard Ltd; Galaxy Music Corporation (A galliard paperback).

Overholt, D. (2009) 'The Musical Interface Technology Design Space', *Organised*

Sound, 14(02), p. 217. doi: 10.1017/S1355771809000326.

Popova, M. (2012) *10 Rules for Students, Teachers, and Life by John Cage and Sister Corita Kent*, *Brain Pickings*. Available at: https://www.brainpickings.org/2012/08/10/10-rules-for-students-and-teachers-john-cage-corita-kent/ (Accessed: 2 May 2017).

Roads, C. (1996) *The computer music tutorial*. Cambridge, Mass: MIT Press.

Snibbe, S. S. and Levin, G. (2000) 'Interactive dynamic abstraction', in *Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*. ACM, pp. 21–29. Available at: http://dl.acm.org/citation.cfm?id=340919 (Accessed: 30 April 2017).

Tyranny, G. (2017) *Mycenae-Alpha (1978)/ in 'Electroacoustic Music: Classics' - Iannis Xenakis | Songs, Reviews, Credits*, *AllMusic*. Available at: http://www.allmusic.com/album/mycenae-alpha-1978-in-electroacoustic-music-classics-mw0000877787 (Accessed: 18 August 2017).

Wessel, D. and Wright, M. (2017) '2001: Problems and Prospects for Intimate Musical Control of Computers', in *A NIME Reader*. Springer, pp. 15–27. Available at: http://link.springer.com/chapter/10.1007/978-3-319-47214-0_2 (Accessed: 16 August 2017).