

Notazioni elettromeccaniche, o forse no

Andrea Valle, CIRMA/StudiUm – Università di Torino

Ricevuto il 7 Luglio 2018

Revisione del 28 Marzo 2019

Introduzione

Il problema di che cosa sia una notazione musicale è, a livello teorico, decisamente più complicato di quanto possa sembrare perché immediatamente apre, a catena, al problema di cosa sia una notazione in generale e, di lì, alla definizione del segno in quanto tale. Personalmente, ho provato in (Valle 2002) a discutere alcune prospettive teoriche generali in relazione alla musica novecentesca, a cui mi permetto perciò di rimandare. Assumerò invece qui di seguito una prospettiva per così dire etnografica, discutendo un insieme di usi della notazione in relazione ad una congerie di progetti da me sviluppati che includono sempre un soggetto umano e un soggetto macchinale. Questa mossa indubbiamente circoscrive il problema introducendo un limite perimetrale biografico e uno, certamente più interessante, di tipo tecnologico. La messa in relazione tra umano e macchinale è uno snodo cruciale nello sviluppo e nella proliferazione dei linguaggi di programmazione, a partire dal COBOL (Ceruzzi 1998), e ha portato alla distinzione fondamentale tra alto e basso livello in relazione ai linguaggi stessi (Schneider e Gersting 2007). Nei linguaggi di programmazione, “alto livello” indica la prossimità all’umano mentre “basso livello” segnala la prossimità alla macchina. Dunque, i linguaggi di programmazione costituiscono un luogo di mediazione semiotica tra umano e macchinale, con tutta la complessità semiotica che consegue da questa duplicità (Valle e Mazzei 2017). Il riferimento alla programmazione non è accessorio o riduttivo in questo contesto, perché il perimetro biografico di cui si diceva sopra limita l’insieme dei progetti che discuterò a tecnologie di tipo computazionale nell’ambito del cosiddetto *physical computing* (O’Sullivan e Igoe 2004), e rende così pertinente per la notazione musicale lo stesso problema di mediazione semiotica che si presenta nei linguaggi di programmazione. Per questo motivo, almeno rispetto al livello di controllo, i progetti che in seguito verranno discussi includono sempre una dimensione computazionale. Ed è ovviamente a livello di controllo che la nozione di notazione diventa utile. A tal proposito, è stato osservato come la notazione musicale, in generale, sia insieme “memoria” e “progetto” (Ambrosini 1979). La notazione ha dunque una relazione strutturalmente ambigua, bifronte, con la temporalità: è

insieme orientata all'indietro, verso ciò che in essa deposita (storicamente, e insieme empiricamente in relazione a uno specifico lavoro di composizione), e in avanti, verso ciò che essa rende possibile o almeno promette (la classe delle esecuzioni possibili). Come per il segno in generale, per la notazione musicale vale il motto "et est et non est" (Ferraris 1997). Da un lato, la notazione è espressione *in praesentia*, che si manifesta in una qualche natura sensibile (inchiostro su carta, pixel su uno schermo, ma anche suono nel caso di indicazioni acustiche, ecc.), dall'altro questa sua stessa natura materiale si cancella verso qualcos'altro, e nel caso della notazione musicale, come si è detto, il suo contenuto include un plesso inestricabile di passato e futuro. Cercherò nel seguito di discutere questa natura semioticamente biplanare di espressione/contenuto e questa caratterizzazione temporale del secondo (passato/futuro) in relazione ad alcuni progetti, presentati come *case studies*.

Notazione e visualizzazione dei dati

Grazie alle tecnologie computazionali e all'accessibilità di grandi quantitativi di informazione digitale, l'ambito della visualizzazione dei dati ha guadagnato negli ultimi anni una particolare rilevanza come strumento di interpretazione e, ancora maggiormente, di esplorazione dei dati. Nella visualizzazione dei dati, a partire da un'informazione di tipo numerico, si tratta allora di definire un morfismo che metta in relazione quel dominio di partenza con un co-dominio, inteso come insieme di categorie di tipo visivo. Se tradizionalmente la visualizzazione dei dati si produce in un certo output definitivo (un'immagine, cfr. Tufte 1986), grazie alle tecnologie recenti la stessa visualizzazione a schermo può diventare interattiva e cioè parametrizzabile per esplorare una certa dimensione o escursione (Fry 2008). Aderendo a questo *Zeitgeist* tecnologico e spostandosi al dominio udibile, è stato osservato come gli strumenti musicali acustici possano essere pensati come mezzi per convertire movimenti fisici in suoni musicali: la composizione musicale diventa un processo implicito di organizzazione e direzione dei gesti umani su uno strumento musicale, cioè, propriamente, una sonificazione del gesto (Delle Monache, Polotti, Papetti e Rocchesso 2008). In questa prospettiva, la composizione algoritmica, cioè l'uso di tecniche formali per la generazione della composizione musicale, può allora essere pensata come una sonificazione dei dati, ovvero come l'equivalente udibile della visualizzazione dei dati, poiché produce strutture dati che vengono proiettate nel dominio acustico (sia esso di origine strumentale o elettronica). Nel caso in cui questa proiezione avvenga attraverso strumenti acustici tradizionali, a maggior ragione si può pensare alla notazione, che media tra dati compositivi ed esecutori strumentali, come a una forma di visualizzazione dei primi, evidentemente secondo il modello della scrittura.

Un caso interessante di cui mi sono occupato ha riguardato *Imaginary Landscape No. 4* di John Cage (Valle e Casella 2016). Il lavoro originale del 1952 è stato scritto per 12 radio e 24 esecutori, due per ogni radio a controllarne volume e frequenza. Per questa composizione, pionieristica nell'uso di dispositivi elettronici quotidiani, Cage ha predisposto una partitura apparentemente scritta in forma tradizionale, ma che

in realtà prevede un insieme di peculiarità legate sia ai dispositivi impiegati sia ad un uso idiosincraticamente cageano della notazione tradizionale. L'ipotesi di lavoro, con una sorta di *reverse engineering*, è stata quella di partire dalla partitura scritta e dalle indicazioni del compositore per ricostruire una struttura dati che rappresentasse gli eventi da eseguire per ogni radio, includendo tutti i parametri previsti. Per ottenere quest'ultima, si è scelto di annotare in forma testuale la partitura. Il formato di annotazione predisposto (Figura 1) traduce dunque in forma testuale (digitale) le informazioni grafiche presenti in partitura: ad esempio, le durate hanno richiesto di misurare fisicamente le lunghezze dei righi e di verificarne la congruenza con le indicazioni di durata specificate dal compositore.

Una simile annotazione testuale può essere usata come strumento analitico, come aiuto per l'esecuzione per i musicisti, e come sorgente per una realizzazione automatizzata del lavoro. Quest'ultima è stata infatti implementata (anche se non completamente) attraverso 12 radio controllate in frequenza e ampiezza via microcontrollore Arduino (Banzi 2009).

Nel contesto di una riflessione sulla notazione, ci si può chiedere su quale sia lo statuto semiotico di una simile annotazione testuale. In effetti, dal punto di vista teorico si tratta di un'altra forma di notazione, in cui il dato alfanumerico dell'annotazione costituisce l'espressione della partitura (che era a sua volta espressione rispetto al contenuto musicale), che ne diventa così contenuto, secondo un principio di "reversibilità totale" tra i due piani del segno (Eco 1975). A sua volta, tuttavia, il formato di annotazione consente di essere visualizzato sotto forma di una partitura grafica che utilizza la *time notation*, partitura intesa come forma di controllo visivo dei dati annotati e come strumento di analisi e comunicazione (Figura 2).

Dal punto di vista tecnico, il formato di annotazione testuale viene importato in Nodebox2, un software che include interfaccia grafica e un insieme di librerie per la programmazione nel linguaggio Python3: Nodebox si occupa della visualizzazione e dell'esportazione finale dell'immagine. Un punto di rilievo nella generazione attraverso la programmazione sta nel fatto che è possibile parametrizzare la visualizzazione e

*	g	1/4
1	77	h
@	2/4	@
2/4 + 2/4 + 1/4 + 1/4 + 1/4	g	0
@	88	5/4
3	1/4	1
7/4	g	1/4
h	101	0
@	1/4	1/4
66	h	#
2/4	110	

Figura 1. Annotazione della partitura di *Imaginary Landscape No. 4*.

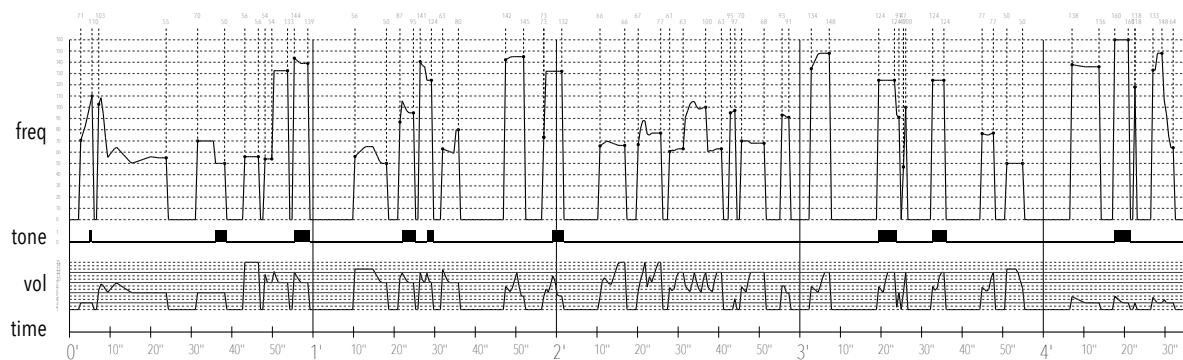


Figura 2. Visualizzazione della parte di una delle 12 radio a partire dall'annotazione testuale di *Imaginary Landscape No. 4*.

generare la partitura a diversi gradi di risoluzione dell'informazione. In questo processo, partitura originale, annotazione e partitura di visualizzazione costituiscono così una catena di trasformazioni tra espressioni e contenuti, e in ognuna delle posizioni si perde e si guadagna informazione. Ad esempio, la partitura cageana capitalizza la competenza storica dei soggetti umani, anche in un contesto inusuale come quello del controllo delle radio, ma occulta parzialmente l'organizzazione finale del brano; il formato di annotazione produce una formalizzazione analitica, a scapito della immediata percezione dell'organizzazione temporale; la partitura di visualizzazione permette di esplorare in maniera variabile una forma complessiva, ma risulta più difficile da eseguire direttamente per i soggetti umani.

La notazione come visualizzazione sottolinea evidentemente, nel suo essere a posteriori, la dimensione della memoria: non è cioè pensata come strumento di realizzazione ma come esplorazione visiva di forme generate in altro modo. In relazione al controllo di oggetti mediali, una situazione analoga si è presentata nel ciclo *Systema naturae*, scritto a quattro mani con Mauro Lanza (2013-17), che prevede, per ognuno dei quattro pezzi di cui è composto, strumenti musicali tradizionali e un vasto insieme di dispositivi elettromeccanici, costruiti tipicamente a partire da oggetti quotidiani (asciugacapelli, motori di recupero, coltelli elettrici, e così via), che sono controllati in tempo reale dal calcolatore (Valle e Lanza 2017). L'alimentazione dei motori e degli altri dispositivi elettromeccanici è modulata via microcontrollore Arduino. L'unico parametro di controllo (e dunque di composizione) disponibile è dunque il valore a 8 bit dei segnali elettrici prodotti in Pulse Width Modulation (PWM) da Arduino, che determina la quantità di corrente ai motori.

La partitura di controllo dei dispositivi elettromeccanici è sempre generata algebricamente, attraverso una varietà di tecniche (qui non rilevanti), e prende la forma di una lista di eventi parametrizzati secondo il modello dello *score* di Csound (Boulangier 2000). Come si vede in Figura 3 (sinistra), ogni evento è indicato, riga per riga, con il nome del dispositivo elettromeccanico cui fanno seguito i valori di alcuni parametri (attacco, durata, valore iniziale e valore finale). Questo formato di scambio di natura testuale funziona *a lato hominis* per garantire una leggibilità del contenuto musicale tra i compositori, mentre *a lato machinae* il formato testuale può essere parsificato

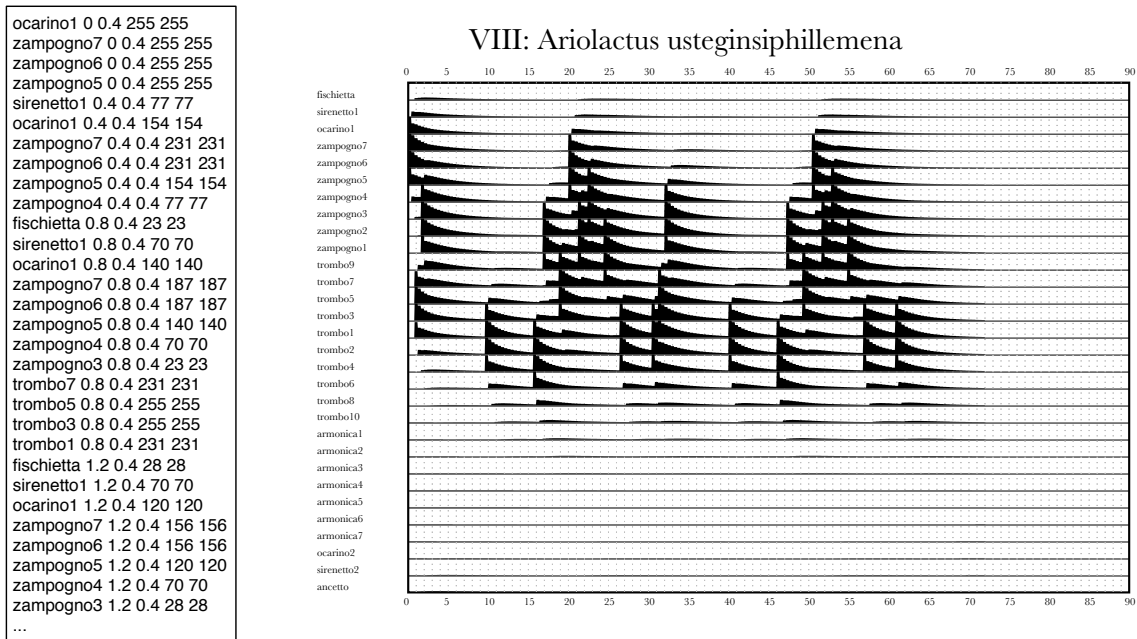


Figura 3. Notazione testuale e visualizzazione grafica via PostScript in *Systema naturae*.

automaticamente per produrre la struttura dati di controllo che viene utilizzata per l'esecuzione in tempo reale sui dispositivi elettromeccanici. Si tratta dunque di una notazione che media tra leggibilità (e scrivibilità, in alcuni casi) umana e formalizzazione macchinale. Anche in questo caso tuttavia è stata approntata una partitura grafica (Figura 3, destra), che serve come controllo compositivo a posteriori e come descrizione della forma complessiva che assume la componente macchinale. Dal punto di vista tecnico, la partitura viene generata automaticamente sfruttando PostScript (Adobe 1999), un linguaggio di programmazione per la grafica vettoriale in forma testuale.

Come si vede, l'ambito mediale richiede costellazioni di software dedicati a svolgere compiti diversi: in questo contesto, il formato testuale è il perno intorno a cui si reggono le procedure di integrazione, sia in termini di scambio dati (an/notazioni) che rispetto alla loro elaborazione (linguaggi di programmazione). Dunque, a partire dalle partiture testuali il programma di generazione scrive file che contengono codice PostScript, i quali possono essere convertiti in formato PDF e inclusi nella partitura complessiva che comprende anche gli strumenti. Secondo il modello presentato, centrato sulla proiezione di dati in diversi domini, nei primi tre lavori le partiture testuali vengono convertite in altre partiture grafiche per essere visualizzate nell'interfaccia grafica per il controllo in tempo reale dell'esecuzione (un esempio è in Figura 4).

Nell'ultimo brano del ciclo, *Fossilia*, le partiture per oggetti vengono invece utilizzate direttamente all'interno dell'interfaccia grafica che controlla l'esecuzione in tempo reale. In Figura 5, la partitura è caricata come immagine nella finestra di controllo (che contiene, come in Figura 4, anche un cronometro, un campo di inserimento e un pulsante di avvio/fermo dell'esecuzione) e ad essa viene sovrapposta una linea verticale rossa che funziona da marcatore interattivo di avanzamento temporale. Si assiste così in ambito digitale ad una permeabilità totale tra partitura "tradizionale" e interfaccia grafica.

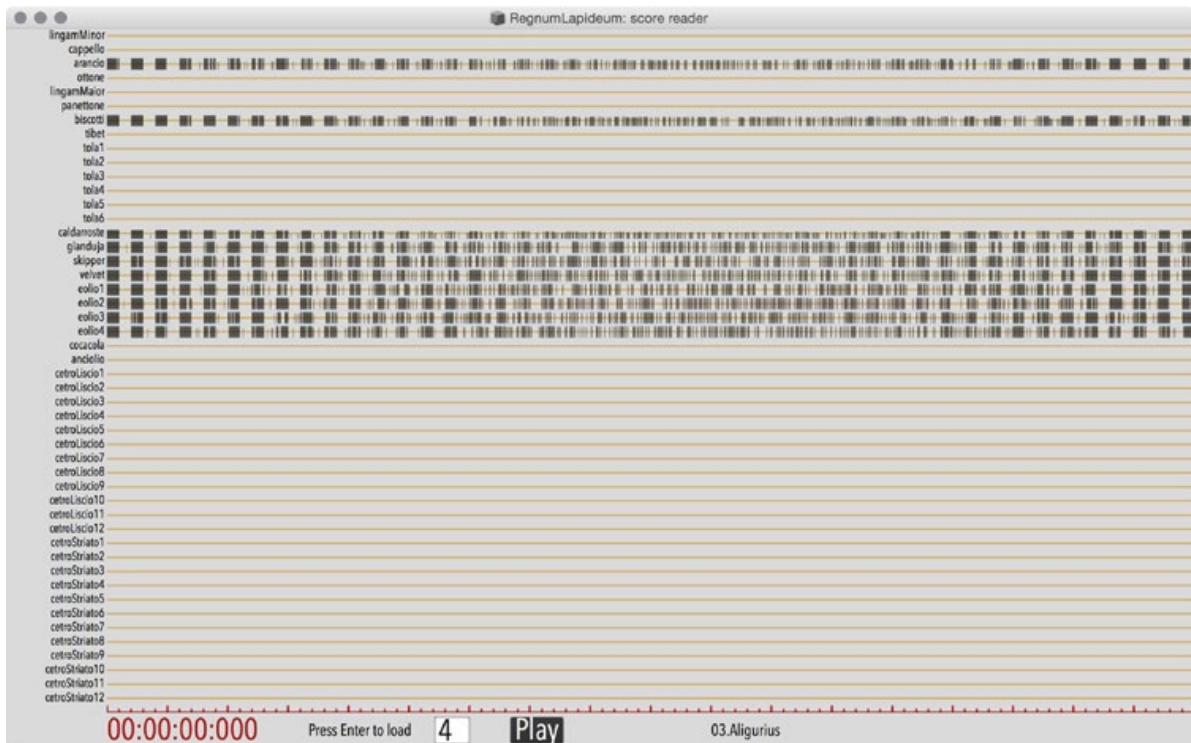


Figura 4. Interfaccia grafica a partire dalla notazione testuale.

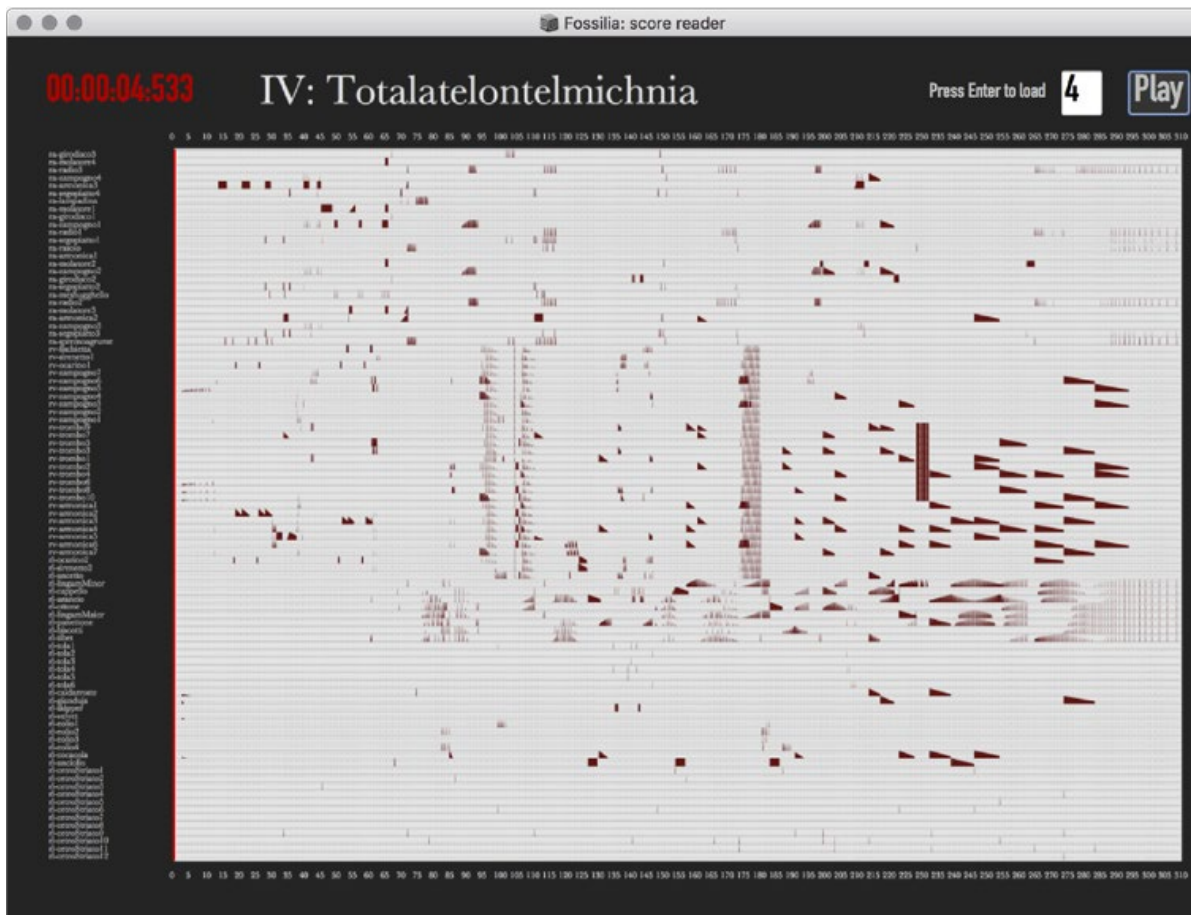


Figura 5. Interfaccia grafica che include la notazione grafica.

Notazioni grafiche

Una via in qualche modo completamente speculare rispetto alla visualizzazione dei dati è quella dell'utilizzo di partiture grafiche. Il progetto *Rumentarium* (Valle 2013) è costituito da un ensemble di percussioni elettromeccaniche, costruite a partire da oggetti di recupero eccitati da piccoli motori a corrente continua, secondo il modello introdotto precedentemente. Una tra le modalità di controllo previste per l'ensemble ha incluso un insieme di partiture grafiche disegnate a mano su carta. In questo caso (18 elementi percussivi), la notazione prevede un rigo di 18 spazi, secondo il modello del rullo del piano. Il livello del grigio controlla il parametro PWM. La notazione grafica è pensata per sfruttare gli effetti di configurazione resi possibili dalla gestualità grafica e la continuità di controllo attraverso la sfumatura. La procedura compositiva comprende un insieme di passi. In primo luogo, allestimento del rigo (generato via Nodebox) e disegno a mano della notazione (Figura 6).

Dopo l'acquisizione dell'immagine via scansione, questa viene ritagliata in modo che i rigi siano riassemblati in un'unica immagine che rappresenti linearmente l'intero sviluppo temporale. Seguono il sezionamento in righe (una per ogni strumento), il campionamento (che determina la risoluzione orizzontale in pixel a rappresentare le unità temporali) e la quantizzazione (il livello dei grigi viene ridotto a 8 bit). Di fatto, la partitura grafica in entrata è convertita in una matrice, le cui colonne rappresentano le unità temporali, le righe i 18 dispositivi elettromeccanici, i livelli di grigio (i contenuti delle celle) i valori di PWM. Il processo di esecuzione legge una colonna alla volta e invia ai 18 dispositivi il segnale con il valore di PWM relativo alla cella. Si noti che la discretizzazione in realtà si traduce tipicamente, per l'inerzia dei motori che "legano" una cella alla successiva, in un comportamento continuo.

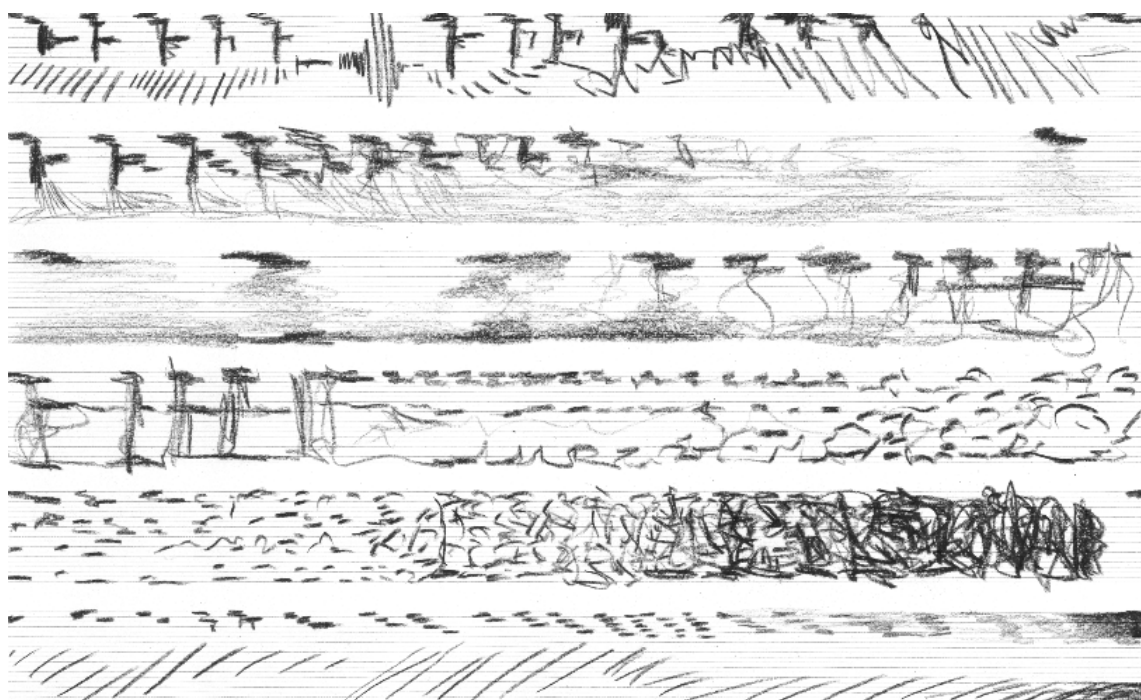


Figura 6. Partitura grafica disegnata sul rigo.

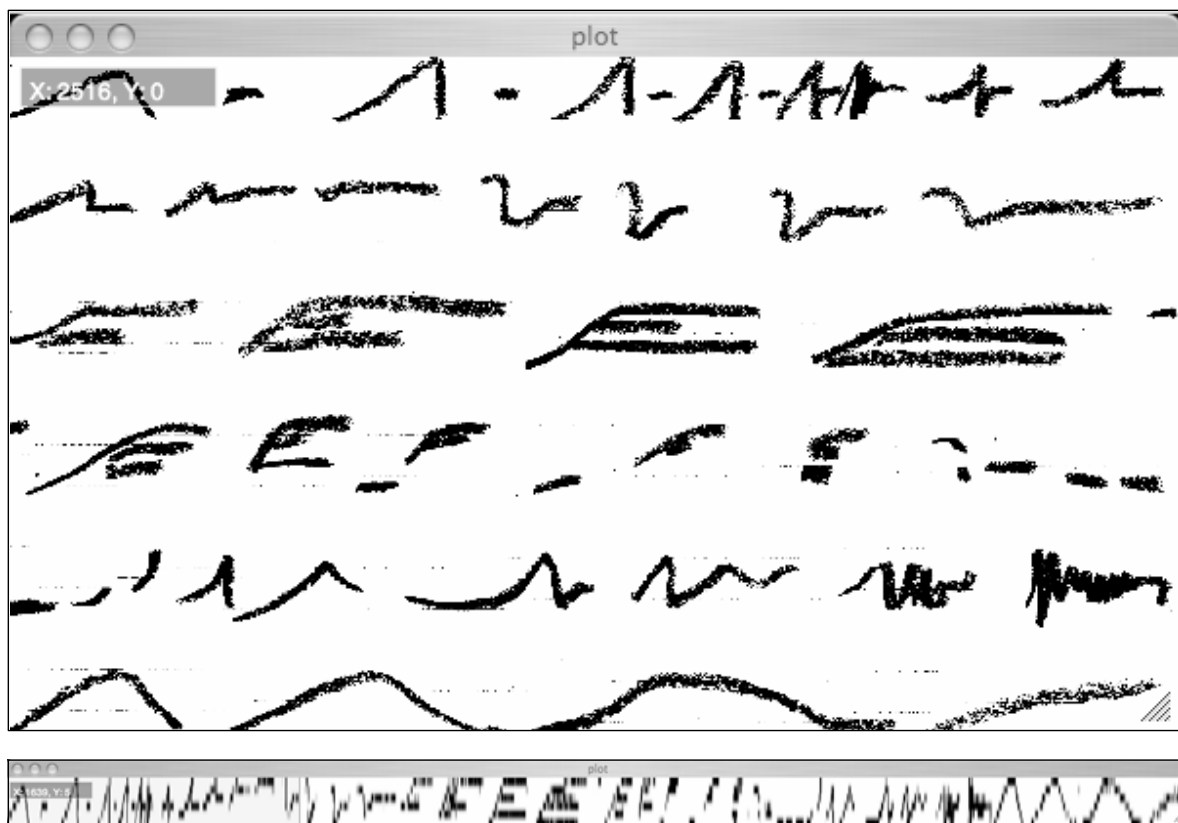


Figura 7. Partitura grafica importata in una interfaccia grafica (sopra) e rielaborata (sotto).

La Figura 7 mostra una partitura grafica digitalizzata e importata in una interfaccia grafica (sopra) e la sua versione ricampionata e riassemblata come rullo del piano (sotto).

Dal punto di vista teorico, questo insieme di processi ha un effetto rilevante poiché converte il regime semiotico della notazione. Quello originale è un regime che con Goodman (1976) si può definire autografico. Per Goodman il regime autografico è definito dall'assenza di modello formale che ne descriva gli elementi. In altri termini, potenzialmente ambigui in questo contesto, esattamente dall'assenza di una "notazione" (cfr. anche Basso 2003). La notazione disegnata è autografica proprio perché non ha una "notazione", cioè non prevede un modello formale, laddove nel regime allografico invece esiste una specificazione degli elementi e della loro configurazione (si pensi alla notazione standard, la cosiddetta *Common Practice Notation*, o più semplicemente ancora alla scrittura alfabetica). La matrice risultante dalla digitalizzazione della partitura manuale è dunque allografica: di fatto è una notazione della notazione disegnata.

Codice come notazione

La discussione precedente ha sottolineato come l'informazione testuale di per sé possa costituire una forma di notazione. Del tutto conseguente, un programma per la generazione musicale è una forma di notazione, poiché definisce un insieme di materiali e la loro organizzazione attraverso una sintassi e una semantica formalmente

definite. Si può astrarre dalla macchina? È un punto interessante. Da un lato alcuni processi possono essere descritti esaustivamente al livello della programmazione. Data una sintassi ed una semantica del linguaggio, si fa così astrazione dall'implementazione (qualunque essa sia), che si dovrà far carico di realizzare correttamente il programma. D'altro canto, è pur vero che, se il musicista lavora interattivamente con la macchina, l'implementazione effettiva, con la sua specificazione formale necessariamente presente ma implicita o nascosta da altri strati linguistici, diventa pertinente, perché si realizza un anello di retroazione tra programmazione e risultato ottenuto (si programma in funzione del risultato). Un esempio in proposito riguarda la generazione del segnale. La semplice sintesi di una sinusoide può produrre risultati più o meno "puri" in relazione all'implementazione DSP del software in uso, e dunque retroagire sulle scelte estetiche del compositore. Se però si accetta l'idea di astrazione (alla base della programmazione) e non si considerano casi di sfruttamento specifico dell'implementazione (e in particolare dei suoi problemi: si pensi alla prospettiva estetica del *glitch*), il programma si comporta a tutti gli effetti come una notazione della musica che si vuole produrre. La nozione di programma qui utilizzata fa astrazione dalla forma espressiva che lo veicola: potrebbe perciò essere scritto in forma testuale o in forma grafica (nel cosiddetto *visual programming*). Sebbene questa distinzione di forma non costituisca di per sé una discriminante teorica, è pur vero che in termini analitici (*a lato hominis*) la scrittura mantiene una sua perspicuità specifica. In molti miei lavori l'unica forma di notazione presente è appunto il codice, tipicamente scritto in SuperCollider (Wilson, Cottle e Collins 2011). Poiché un linguaggio di programmazione è descritto formalmente ed esaustivamente da una grammatica (ed adegua per definizione il modello della Macchina di Turing), è teoricamente possibile tradurre un programma in altri linguaggi. Il codice dunque diventa la notazione della musica. Un'ulteriore aspetto di interesse dipende dal linguaggio in questione, SuperCollider, che è tecnicamente "interpretato": dunque, ogni espressione sintatticamente corretta può essere eseguita immediatamente. Data la natura interpretata del linguaggio, il codice *describe* e insieme *esegue* i processi che costituiscono la musica. Diversi sono i punti di interesse nel considerare il codice come notazione. Come osservato, in primo luogo, si tratta di una notazione astratta rispetto al risultato, che potrebbe essere tradotta, più o meno agevolmente, in un altro. Esso mantiene una sua astrazione rispetto al risultato. In secondo luogo, si tratta tipicamente (soprattutto se si fa uso di controlli di flusso e di meccanismi di incapsulamento) di una notazione di tipo sintetico e procedurale. Ad esempio, la ripetizione per 100 volte di un evento *E* prende una forma scrittoria procedurale del tipo "ripeti 100 volte *E*". Si potrebbe osservare come un precedente di questo tipo di notazione possa essere trovato in tutte quelle notazioni musicali, come quelle dell'*Ars subtilior* ad esempio, che condensano informazione e lasciano all'esecutore il compito di espanderla nuovamente. Si pensi al canone in cui al tema si aggiunge semplicemente l'indicazione "sicut cancher": la notazione include una indicazione sulla procedura da effettuare per recuperare l'intera informazione musicale, ovvero il retrogrado del tema notato. In terzo luogo, la programmazione interattiva (in SuperCollider o in sistemi analoghi) ottempera effettivamente a quel duplice sguardo della notazione musicale in avanti e all'indietro. È memoria di un lavoro fatto

```

liveMorsing.scd
3 (
4 // the DC
5 ~dc = DeeCee1.new(2);
6 // ~dc.addEvent(3)
7
8 // function to speak morse
9 ~speak = {lid = 1, text = "morse", mul = 1, dcl
10   var seq = Morse.new(text).flat*mul ;
11   {seq.do{|t| dc.addEvent(id); t.postln.wait}}.fork
12 } ;
13
14 // a def to get onsets
15 SynthDef(\dcOnset, {arg in = 0, db = 0, thresh = 0.25 ;
16   var sig = SoundIn.ar(in)*db.dbamp;
17   var loc = LocalBuf(512, 1) ;
18   var onsets, chain, input = sig ;
19   chain = FFT(loc, input) ;
20   onsets = Onsets.kr(chain, thresh) ;
21   SendTrig.kr(onsets, in+1, 0) ;
22 }).add ;
23
24 o = OSCFunc({lmsg, time, addr, recvPort| "GOT IT!: ".post ;
25   ~dc.addEvent(msg[2].postln); "\n\n".postln
26 }, \tr) ;
27
28
29 )
30
31
32 Chronometer.new ;
33
34 ~dc.addEvent(8)
35
36
37 // utterance: where, what, scale, dc
38
39 ~speak.(4, "e", 0.75, ~dc) ;
40 ~speak.(5, "h", 0.75, ~dc) ;
41 ~speak.(3, "m", 0.75, ~dc) ;
42 ~speak.(2, "h", 0.75, ~dc) ;
43 ~speak.(1, "h", 0.75, ~dc) ;
44 ~speak.(7, "h", 0.75, ~dc) ;
45

```

Figura 8. Codice interpretato in SuperCollider.

e progetto messo in esecuzione. La dimensione interattiva, per cui il codice può essere modificato ed eseguito mentre è in esecuzione, non modifica il quadro. Anche nel caso estremo del *live coding*, in cui le espressioni linguistiche vengono scritte durante e costituiscono propriamente la performance (Collins, McLean, Rohrer, e Ward 2003, ma anche il numero monografico del *Computer Music Journal*, 38/1-2014, “Live

coding”), queste devono essere formalmente ben formate per essere eseguite: si assiste dunque non a un cambio di logica, ma a una sorta di continua alternanza “in compressione temporale” dei due momenti di notazione ed esecuzione. Infine, la notazione coincide in questo caso con l’interfaccia esecutiva tra uomo e macchina. La Figura 8 riporta un esempio di codice utilizzato dal vivo. Il blocco delle righe 3-29 contiene definizioni di elementi che verranno utilizzati nel seguito. Le ultime righe (39-45) contengono invece espressioni del linguaggio che lanciano la funzione “-speak”, definita sopra e usata per generare pattern Morse che, alla fine della catena, aprono e chiudono gli altoparlanti di una serie di radio nel progetto *Trilobiti*. Queste istruzioni sono scritte, modificate ed eseguite in tempo reale durante la performance. Esse rimangono così nel programma-notazione come un deposito dell’esecuzione.

Live scripting e live writing

Come si è osservato, un punto fondamentale della programmazione è la costruzione di strati linguistici. Diventa allora possibile, a partire da un certo linguaggio, definire microlinguaggi di programmazione “al di sopra” di esso. Si perde in complessità potenziale, si guadagna in semplicità d’uso (cfr. per una applicazione alla musica elettronica Rohrhuber, Hall e de Campo 2011). Il progetto *GeoGraphy* descrive un ambiente per la composizione del suono digitale basato su grafi, distribuiti in uno spazio di controllo, che sono responsabili della generazione di sequenze temporalmente organizzate di eventi sonori (Valle 2008a). Include una interfaccia grafica di visualizzazione di questi ultimi. Il controllo (costruzione dei grafi, loro modifica e generazione del suono in tempo reale) può avvenire in forma testuale attraverso un micro-linguaggio, iXno, definito sopra SuperCollider. Propriamente, si tratta di un linguaggio di *scripting* che rende pertinente nella scrittura soltanto gli elementi di *GeoGraphy*. In Figura 9 le tre espressioni iniziano con un operatore (“e+” e “a+”), seguito da altri parametri. Poiché iXno eredita per definizione la natura interpretata di SuperCollider, esso può essere scritto ed eseguito in tempo reale, secondo un approccio che si potrebbe definire *live scripting*. In termini di notazione, lo script si configura come notazione: tant’è che potrebbe essere implementato rispetto a un altro linguaggio soggiacente.

Estendendo questa logica, in cui una certa configurazione testuale è notazione per un certo risultato sonoro, ai suoi estremi, si può pensare di usare un testo in lingua

```
e+ TapeIn 4 BetaMagnet 6 BetaKnob2 4
      BetaMagnet 6 BetaKnob 11 BetaKnob2 4 BetaKnob
e+ BetaKnob2 6 TapeOut
a+ TapeIn
```

Figura 9. Un esempio di script iXno.

naturale come notazione musicale. Si tratta di un caso di sonificazione, ma può essere letto, rispetto al percorso della discussione precedente, come un passaggio astratto supplementare, in cui al testo in ingresso corrisponde un algoritmo proiettivo di complessità superiore. Una tecnica che uso correntemente per il controllo di eventi dal vivo (tipicamente realizzati attraverso dispositivi elettromeccanici) consiste così nel definire algoritmi di proiezione tra l'insieme dei caratteri alfabetici e quello dei dispositivi. Diventa così possibile controllare pattern temporali attraverso sequenze testuali in lingua naturale. Una simile strategia è interessante per due motivi. In primo luogo, la lingua naturale prevede distribuzioni fortemente asimmetriche dei fonemi (e dunque, sebbene indirettamente, dei caratteri nelle scritture alfabetiche) e insieme forti ridondanze. Ne conseguono pattern iterativi ma estranei alle logiche di composizione tipicamente musicali. In secondo luogo, la scrittura può essere "eseguita" dal vivo, si può cioè letteralmente "suonare la tastiera" del calcolatore sfruttando la rapidità della memoria muscolare. In diversi progetti, allestisco così sistemi di frasi che digito dal vivo, improvvisando ritmicamente le durate. Le frasi vengono scelte in funzione della organizzazione interna e del risultato che l'algoritmo di proiezione produce, spesso attingendo al repertorio poetico che sfrutta precipuamente l'organizzazione espressiva della lingua (di qui la nozione di *live writing*, Valle 2011).

Ne consegue una configurazione di tre elementi. Le frasi selezionate costituiscono la notazione del brano, senza indicazione temporale, ma di per sé questo non è certo un caso raro nella notazione tradizionale, basti pensare ai preludi non misurati della tradizione clavicembalistica. In secondo luogo, l'esecuzione alla tastiera copre propriamente la dimensione performativa. Infine, gli algoritmi di proiezione costituiscono un sistema di mediazione tra i due elementi precedenti: da un lato la loro specifica può essere letta come parte della notazione (una specifica del risultato sonoro da produrre), dall'altro essi si incaricano propriamente della generazione del suono (o del suo controllo) in tempo reale (e perciò risiedono dal lato dell'esecuzione).

Live printing

Lo statuto tecnologicamente misto della notazione musicale nel momento in cui sfumano le distanze con le interfacce grafiche è altresì dimostrato dalla possibilità di produrre notazione in tempo reale. L'uso di *live notation* ha subito una forte accelerazione recentemente grazie alla disponibilità di software grafico (ad esempio, Processing, Reas e Fry 2007) e hardware (monitor, dispositivi mobili, e così via). Una *live notation* visiva (perché in effetti altre modalità sensibili potrebbero essere convocate) è in effetti un ibrido di notazione e interfaccia grafica. Il problema non è teoricamente rilevante se l'esecuzione è affidata a una macchina, poiché in questo caso è lo stesso dato di controllo che si pone come notazione ed è eseguito alla massima velocità computazionalmente possibile. Perciò, propriamente, *live notation* implica una notazione musicale rivolta tradizionalmente ai soggetti umani. In questo caso, la capacità di elaborazione in tempo reale degli stessi diventa un punto nodale. Si potrebbe altresì osservare che la notazione tradizionale in molta parte anticipa il futuro: basti pensare

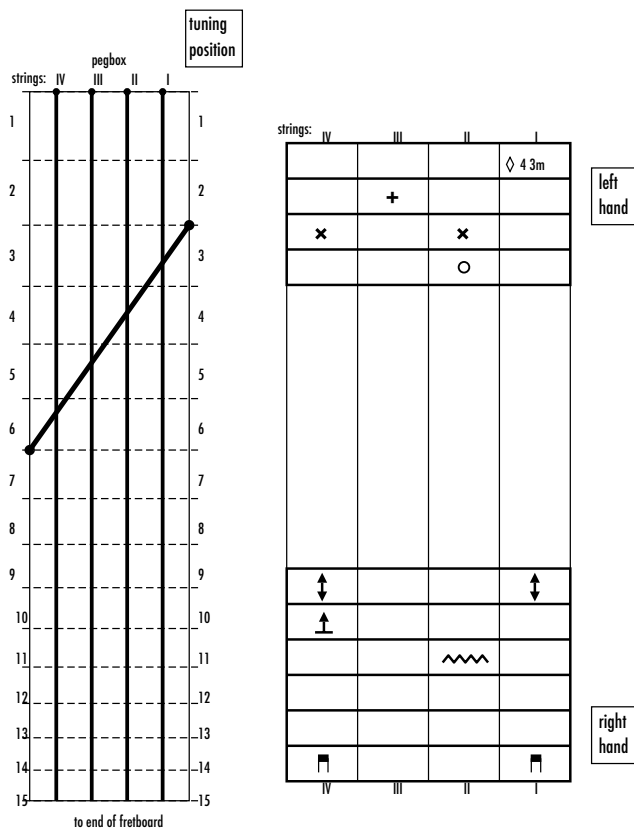
alle forcelle dinamiche che indicano propriamente un processo continuo. Più generalmente, ogni dimensione che superi quella dell'evento deve essere anticipata nella notazione proposta per non risultare problematica all'esecuzione. Questa separazione di magnitudini temporali tra macchina ed esecutore è al centro del progetto *Dispacci dal fronte interno*, scritto per archi, stampanti e live electronics. Il pezzo è basato sulla nozione di retroazione, e prevede un insieme di microfoni che catturino il suono dal vivo. Un processo computazionale si occupa dell'analisi del segnale in ingresso. Questa stessa informazione è alla base della generazione della notazione per gli archi che viene inviata alle stampanti (una per ogni esecutore). I microfoni captano il segnale ambientale complessivo, quello degli archi e quello risultante dal funzionamento delle stampanti, e il processo computazionale genera il suono spazializzato (a partire dal materiale sonoro complessivo) su 8 canali e la notazione musicale per gli archi. Dunque il brano si produce come un unico grande anello di retroazione complessivo in cui tutto il materiale sonoro è all'origine del suo sviluppo nel tempo. Questo anello però comprende strati temporali che operano a tassi differenti. La sintesi del segnale reagisce infatti a tasso di campionamento, gli esecutori prelevano il foglio e lo eseguono a velocità strumentale, infine le stampanti operano con tasso temporale minimo, poiché il processo di stampa richiede svariati secondi. Qui la macchina è dunque insieme il più rapido e il più lento degli elementi di questa configurazione a tre. La notazione in questo caso entra a far parte della configurazione complessiva non solo in quanto oggetto precipuamente cognitivo ma anche, per il tramite del dispositivo che materialmente ne genera il supporto, come componente acustica importante legata alla processualità della sua costituzione.

Dal punto di vista tecnico, il software di controllo in SuperCollider, oltre a prendere in carico l'elaborazione del segnale, genera, per ogni foglio di notazione, un file PostScript. La notazione prende la forma di una sorta di intavolatura che specifica rispetto al manico tecniche esecutive per le due mani. Il file PostScript viene inviato direttamente in stampa attraverso il terminale, poiché le stampanti sono tipicamente dotate di un interprete per il linguaggio PostScript. Un punto di rilievo è che lo statuto della notazione come oggetto a priori dell'esecuzione è parzialmente ridiscusso. Essa dipende evidentemente dalle condizioni acustiche complessive, non predeterminabili. E tuttavia diventa possibile archiviare i file prodotti nella performance e ottenere una notazione a posteriori del brano complessivo, ovvero una partitura di documentazione della performance in forma tradizionalmente grafica. In Figura 10 è riprodotto uno dei fogli (i "dispacci" del titolo) da una performance.

Astrazione procedurale

L'utilizzo di dispositivi tecnologici pone un problema cruciale rispetto alla notazione. Da un lato, si è visto come in ambito digitale la codifica stessa dell'informazione nelle diverse forme che può assumere costituisca teoricamente una notazione. È però vero che, empiricamente, un progetto che includa tecnologie di vario tipo distribuisce questa notazione complessiva tra i molti attori (software, hardware, umani) poten-

Dispatch no. 11

dynamics: *pp*tempo: *Allegro*

Edua Amarilla Zadory & Ana Topalovic @ USO project Milano, Sat Dec 1 21:12:47 2012

Figura 10. *Dispacci dal fronte interno*: notazione per archi generata in tempo reale e archiviata.

zialmente coinvolti. Questa distribuzione è insieme un seppellimento perché rende difficilmente accessibile l'informazione complessiva, frammentando la totalità del progetto compositivo in unità eterogenee. La notazione scritta tradizionale, nella forma della partitura, produce invece un'entità documentale unitaria (indipendentemente dal fatto che richieda poi una prassi esecutiva di riferimento). Non a caso, questa unitarietà può venire incrinata proprio quando la tecnologia è coinvolta, e la partitura rimanda al di fuori di essa: si pensi ai brani per strumenti e live electronics in cui ad esempio si rinvia genericamente ad una componente software non definita se non nella sua implementazione. Pur nell'uso di tecnologie da me sviluppate, sia software che hardware, e dunque altamente idiosincratice, una soluzione radicale è stata quella di astrarre dalla tecnologia. Ad esempio *Orologio da rote* è brano per pianoforte controllato dal calcolatore e 3 radio modificate. Non include perciò esecutori umani. Pensato come una riflessione sulla segnaletica acustica del tempo, prevede che le radio generino un flusso granulare (basato sul codice Morse) a partire dai segnali captati dall'etere, mentre il pianoforte esegue una musica che deriva dall'analisi di un brano utilizzato fino agli anni '90 dalla televisione nazionale per accompagnare il segnale orario. A intervalli che dipendono dall'ora recuperata dal sistema operativo del com-

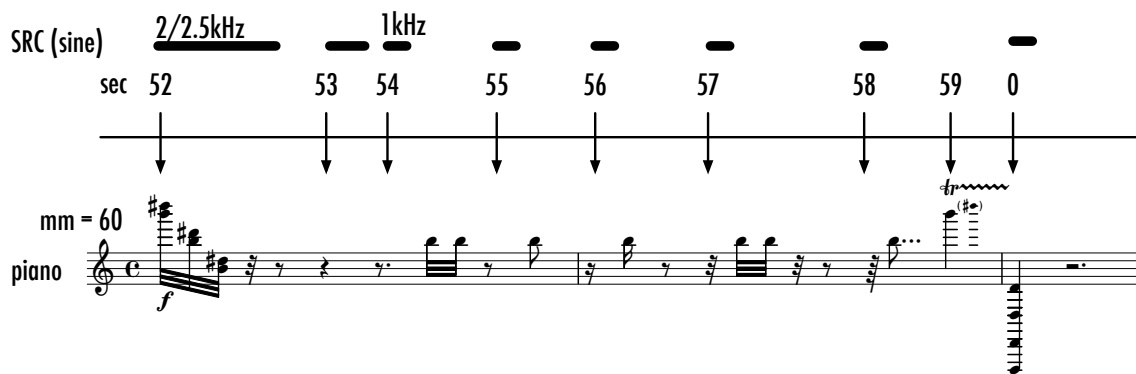


Figura 11. Notazione mista per sinusoidi e piano in *Orologio da rote*.

puter di controllo, altri segnali orari vengono emessi dal pianoforte e dalle radio. La durata complessiva (tra 2'48" e 3'40") varia anch'essa in funzione dell'ora in cui viene eseguito. Si tratta perciò di un brano che mette insieme una dimensione esecutiva e installativa, e che non richiede esecutori. Presenta tuttavia una organizzazione compositiva formalmente descritta di tutti gli aspetti coinvolti (dalla modifica delle radio alla generazione dei segnali, dalle procedure di analisi che controllano il pianoforte alla forma temporale complessiva). Di qui la possibilità di una notazione dello stesso sotto forma di un documento di 6 pagine che ne dettaglia l'allestimento (attraverso indicazioni verbali, diagrammi, notazione tradizionale ecc., Figura 11). In altri termini, il progetto, per quanto strettamente sviluppato a contatto delle tecnologie coinvolte, guadagna uno statuto allografico. Si potrebbe osservare che in questo caso si arriva al punto di dispersione tra notazione e documentazione.

Un passaggio astratto successivo consiste nel non prendere in considerazione la dimensione macchinale *tout court*. Ad esempio, *Minute di cronometria* (2018) è un lavoro scritto per due strati, uno che richiede di generare materiale sonoro dotato almeno approssimativamente di altezza, l'altro che invece prevede di gestire con precisione l'informazione ritmica, organizzata in un pattern di base. Ognuno dei 60 pezzi che compongono la partitura consiste allora in una notazione di questi due strati, che serve come indicazione per l'improvvisazione. In Figura 12, gli strati sono distinti in verticale dalla linea orizzontale. Gli strati possono essere realizzati da un numero non specificato di strumenti e/o esecutori. Nell'esecuzione realizzata, lo strato ritmico, come si vede in Figura 12 notato in forma del tutto tradizionale, è stato implementato attraverso un ensemble di 8 percussioni costituite da altoparlanti su cui sono stati posizionati oggetti, quattro delle quali impiegate per la realizzazione del pattern notato, quattro invece dedicate a realizzare variazioni sullo stesso, come previsto in partitura (il segnale audio inviato agli altoparlanti è di tipo impulsivo e ha la sola funzione di sollecitare meccanicamente gli oggetti). Dal punto di vista empirico, questo dispositivo a 8 elementi è stato in realtà il punto di partenza nella composizione. Quest'ultima tuttavia, attraverso la notazione, si è poi completamente astratta dall'implementazione che utilizza dispositivi elettromeccanici e ne ha realizzato così solo una delle possibili esecuzioni. Ciò che si guadagna in generalità si perde necessariamente in specificità.

19

 - Free -

dur.:
2'

sync.: — piv. - - reg. ■ dyn.: *ff*

perc

♩ = 79

Figura 12. *Minute di cronometria, no. 20.*

Conclusioni

Visualizzazione dei dati, notazione come interfaccia, codice come notazione, notazione come documentazione: nello sviluppo di progetti che includano configurazioni di oggetti diversi intorno o insieme a un calcolatore si producono così situazioni notazionali molto diverse in termini di statuto semiotico e di funzione, che si ibridano con elementi certamente estranei alla notazione tradizionale. In questi casi, quello che si potrebbe chiamare il rimosso della notazione tradizionale, cioè il supporto materiale, ritorna prepotentemente in gioco, perché è esattamente il suo statuto di oggetto che ne permette o impedisce l'integrazione con altri oggetti mediali. Se nella notazione tradizionale si può tipicamente (anche se non sempre) fare a meno di prendere in considerazione questa dimensione materiale (la carta, l'inchiostro), essa ridiventa invece pertinente in un contesto di integrazione mediale. In maniera del tutto opposta, la dimensione materiale fondante nei progetti presentati (che includono sempre – salvo nel caso di *GeoGraphy* – dispositivi elettromeccanici) è a ben vedere totalmente irrilevante nel determinare i tratti pertinenti della notazione. Una notazione infatti si costituisce a partire dalla determinazione di una certa pertinenza (Prieto 1976): cioè, da ciò che si può o si deve notare. Dunque, della natura fisica del dispositivo rimane nella notazione soltanto la dimensione informazionale, cognitiva, legata al controllo di quest'ultimo, cioè la sua interfaccia, come si dice sia in semiotica (Zinna 2004) che in programmazione ad oggetti (Gamma, Helm, Johnson, e Vlissides 1995).

Riferimenti bibliografici

- Adobe 1999, *PostScript Language Reference*, 3a ed., Addison-Wesley, Reading, Mass.
- Ambrosini, C. 1979, 'Musica contemporanea e notazione', *Studi Musicali* VIII, pp. 303 ss.
- Banzi, M. 2009, *Getting started with Arduino*, O'Reilly.

- Basso, P. 2003, *Il dominio dell'arte*, Meltemi, Roma.
- Boulanger, R. (a c.). 2000, *The Csound Book*, The MIT Press, Cambridge, Mass.
- Ceruzzi, P. 1998, *A History of Modern Computing*, The MIT Press, Cambridge and London.
- Collins, N., McLean, A., Rohrhuber, J., Ward, A. 2003, 'Live coding in laptop performance', *Organised Sound* 8(3), pp. 321-330.
- Delle Monache, S., Polotti, P., Papetti, S. Rocchesso, D. 2008, 'Sonically augmented found objects', in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Genoa, Italy, pp. 154-157.
- Eco, U. 1975, *Trattato di semiotica generale*, Bompiani, Milano.
- Ferraris, M. 1997, *Estetica razionale*, Cortina, Milano.
- Fry, B. 2008, *Visualizing Data*, O'Reilly, Sebastopol.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. 1995, *Design Patterns: Elements of Reusable Object Oriented Software*, Addison-Wesley.
- Goodman, N. 1976, *I linguaggi dell'arte*, Il Saggiatore, Milano.
- O'Sullivan, D., Igoe, T., 2004, *Physical Computing. Sensing and Controlling the Physical World with Computers*, Course Technology, Boston.
- Prieto, L.J. 1976, *Pertinenza e pratica. Saggio di semiotica*, Feltrinelli, Milano.
- Reas, C., Fry, B. 2007, *Processing: a Programming Handbook for Visual Designers and Artists*, The MIT Press, Cambridge, Mass.
- Rohrhuber, J., Hall, T., de Campo, A. 2011, 'Dialects, Constraints, and Systems within Systems', in *The SuperCollider Book*, The MIT Press, Cambridge, Mass., pp. 635-656.
- Schneider, G.M., Gersting, J. 2007, *Informatica*, Apogeo, Milano.
- Tufte, E.R. 1986, *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT, USA.
- Valle, A. 2002, *La notazione musicale contemporanea. Aspetti semiotici ed estetici*, De Sono-EDT, Torino.
- Valle, A. 2008a, 'GeoGraphy: a Real-time, Graph-based Composition Environment', in *NIME 2008: Proceedings*, pp. 257-260.
- Valle, A. 2008a, 'Integrated Algorithmic Composition. Fluid systems for including notation in music composition cycle', in *NIME 2008: Proceedings*, pp. 253-256.
- Valle, A. 2011, 'Live Writing with the Rumentarium', *Computer Music Journal*, DVD 35(4).
- Valle, A. 2013, 'Making acoustic computer music: The Rumentarium Project', *Organised Sound* 18(3), 242-254.
- Valle, A. 2014, "'Musica per un anno". A computational reconstruction of Enore Zaffiri's analog algorithmic composition', in A. Valle, S. Bassanese, *Enore Zaffiri Saggi e materiali*, AIMI - Associazione Informatica Musicale Italiana, DADI - Dip. Arti e Design Industriale. Università IUAV di Venezia, pp. 118-152.
- Valle, A., Casella, A. 2016, 'Imaginary Landscape No. 4: study and annotation of the score', in A. Terzaroli A. Valle, (a c.), *Extending interactivity - Atti del XXI CIM - Colloquio di Informatica Musicale*, AIMI - Associazione Informatica Musicale Italiana, DADI - Dip. Arti e Design Industriale. Università IUAV di Venezia, pp. 69-76.
- Valle, A., Lanza, M. 2017, 'Systema naturae: shared practices between physical computing and algorithmic composition', in T. Lokki, J. Pätynen e V.Välimäki, eds, *Proceedings of the*

14th Sound and Music Computing Conference, Aalto University, Aalto University, Espoo, pp. 391-398.

Valle, A., Mazzei, A. 2017, 'Sapir-Whorf vs Boas-Jakobson. Enunciation and the semiotics of programming languages', *Lexia. Rivista di semiotica* 27-28, pp. 505-525.

Wilson, S., Cottle, D., Collins, N. (a c.) 2011, *The SuperCollider Book*, The MIT Press, Cambridge, Mass.

Zinna, A. 2004, *Le interfacce degli oggetti di scrittura*, Meltemi, Roma.