

Nummer
11

Musikkens interfaces

Af Morten Breinbjerg

Center for Digital Æstetik-forskning

Digital

SKRIFTSERIE

Center for Digital Æstetik-forskning

Nr. 11 – 2006

Morten Breinbjerg
Musikkens interfaces

Dette arbejdspapir er en omarbejdet og redigeret version af et foredrag fra Forskningsprojektet Interfacekulturens Æstetiks D-dag seminar i april 2005 på institut for Æstetiske Fag, Aarhus Universitet. Denne dag fremlagde projektet sit hidtidige arbejde for et stort veloplagt publikum. Forskningsprojektet løber frem til 2007.

www.interfacekultur.au.dk

Udgiver: Center for Digital Æstetik-forskning
IT-Parken Helsingforsgade 14, DK-8200 Århus N
e-mail info@digital-aestetik.dk • www.digital-aestetik.dk

Tryk: Reprocenteret, Det Naturvidenskabelige Fakultet, Aarhus Universitet

Copyright © 2006 Center for Digital Æstetik-forskning og forfatteren.

ISBN 87-91810-02-7

De enkelte numre kan rekvireres ved henvendelse til Center for Digital Æstetik-forskning, så længe oplaget rækker. Skriftserien forlægger også i en elektronisk udgave, der kan hentes på centerets hjemmeside.

I serien er foreløbig udgivet følgende titler:

1. Kim Cascone: *Laptop Music – counterfeiting aura in the age of infinite reproduction.*
2. Pia Wirnfeldt: *Netkunsten og sidemetaforen - transparensforestillinger og kritiske kunstneriske potentialer.*
3. Anne Sophie Warberg Løssing: *Internettet som udstillingsramme.*
4. Søren Pold: *Genrer i digital kunst*
5. Morten Breinbjerg: *Emergens – om tilsynekomstens æstetik*
6. Falk Heinrich: *Kunst som transiente kommunikationssystemer*
7. Lone Koefoed Hansen & Jacob Wamberg: *Interface eller interlace?*
8. Lars Kiel Bertelsen: *Vindue, spejl, skærm – transparensmetaforik i 'nye medier'*
9. Henrik Kaare Nielsen & Søren Pold: *Kulturkamp.com – mellem åbne værker og intellektuel ejendomsret*
10. Bodil Marie Thomsen: *Real-time interface – om tidslig simultanitet, rumlig transmission og haptiske billeder*
11. Morten Breinbjerg: *Musikkens interfaces*

MUSIKKENS INTERFACES

Interface er en betegnelse for grænsefladen mellem computeren og computerbrugeren, mellem maskinen og den ydre verden. Grænsefladen er ikke kun en fysisk størrelse, men også symbolsk. Symbolsk fordi den virker ved tegn i form af tekst, billede, lyd og grafik, og som sådan kalder den på fortolkning.

Som element i en fortolkningssituation og som en designet grænseflade er interfacet ikke neutralt. Indlejret i interfacet er konceptuelle forståelser af musikkens væsen, og de kommer til udtryk i de programfunktionaliteter vi betjener os af, og i den måde som musikkens elementer i øvrigt er repræsenteret på. Som diskuteret af bl.a. Bolter og Grusin er interfacets repræsentationsformer ofte overtaget fra tidligere medier¹. I musikalske interfaces finder vi således både nodebilledet, båndoptageren og andre ældre medier anvendt som interfacemetaforer og som konkrete repræsentationsformer. Sammenholdt med vores fortolkning af dem er disse repræsentationsformer styrende for de måder, hvorpå vi interagerer med computeren, og lægger musikken til rette på. Det vil sige for den musik, som vi frembringer.

På den baggrund er hensigten med artiklen, at udpege forskellige interfaces virkning på den måde som vi tilgår musikken kompositorisk og æstetisk. For at forstå denne virkning vil jeg undersøge, hvordan udvalgte interfaces præsenterer musikken for os, og hvordan vi forstår, ja, i det hele taget gør os begreb om musikkens væsen gennem vores fortolkning og brug af dem.

I artiklen arbejder jeg med et udvidet interfacebegreb, der inkluderer nodebilledet. Normalt taler vi ikke om nodebilledet som et interface, da det ikke på samme måde som det computer-

¹ Bolter & Grusin, 2002.

baserede interface er en grænseflade til styring af og udveksling med en maskine. Dog er det en grænseflade mellem komponisten, der skriver musikken, og musikeren der opfører den, men kun den ene vej. Musikeren meddeler ikke komponisten noget gennem nodebilledet, selvom nodebilledet naturligvis kan anvendes til at optegne, hvad musikeren spiller. Det er dog ikke det, der giver mig anledning til at betragte nodebilledet som et interface. Anledningen er, at også nodebilledet virker ved tegn og er en repræsentationsform, der tillader os at skrive musikken frem i modsætning til blot at afbilde den. Nodebilledet præsenterer os også for særegne måder at organisere musikken på ud over at indoptage tidligere mere primitive afbildningsformer i sig. Interfacebegrebet er således min adgang til, på tværs af medier, at stille enslydende spørgsmål til virkningen af musikkens repræsentationsformer, samt til at udpege forskelle og ligheder i mellem dem.

På trods af at jeg bruger interfacebegrebet som en samlande betegnelse for musikkens repræsentationsformer, er artiklen opdelt i to selvstændige afsnit, der kan læses uafhængigt af hinanden. I første del diskuterer jeg nodebilledet som et interface, mens jeg i anden del redegør for to overordnede interfaceparadigmer med udgangspunkt i en skelnen mellem computeren som henholdsvis en visuel og en generativ maskine.

I DEL. NODEBILLEDET SOM INTERFACE

Som interface er nodebilledet et medie for kommunikative udvekslinger mellem en komponist og en eller flere musikere. For at forstå dets betydning som interface må vi interessere os for, hvad det er muligt at kommunikere i nodebilledet. De spørgsmål jeg skal forsøge at besvare er, hvad nodebilledet afbilder, og hvad der kan udtrykkes med det. I den forbindelse er det vigtigt at være opmærksom på, at nodebilledet både kan bruges til at optegne

(notere ned) allerede klingende musik og til at angive musik, der senere kommer til at eksistere i klingende form. Nodebilledet kan således både fungere deskriptivt (som model af en praksis) og præskriptivt (som model for en praksis).

NODEBILLEDET SOM REPRÆSENTATION (HVAD AFBILDER DET?)

De første musikalske skrifttegn opstår i den tidlige europæiske kirkemusik som mnemoniske redskaber, der skal sikre enslydende opførelser af de liturgiske salmer. Udbredelsen af den kristne lære gennem kirkesangen skaber nemlig behov for at standardisere melodierne. De første musikalske skrifttegn, de såkaldte accenter og neumer, angiver alene retningen af den melodiske bevægelse, f.eks. om en given tone er højere eller lavere end den foregående, eller om to eller tre efterfølgende toner skal falde eller stige.² Som adastematiske³ neumer indeholder de ingen information om tonernes varigheder, tonehøjder eller om intervalforholdene imellem dem. Derfor kan vi i dag ikke være sikre på, hvordan melodierne faktisk har lydt. Den manglende information har dog ikke været et problem for datidens sangere, idet neumerne kun var en støtte for hukommelsen i en ellers mundtlig baseret læringsproces.

Fra og med Guido d'Arezzo ser vi omkring år 1000 en begyndende horisontal placering af først alfabetiske tegn som tonehøjdeindikatorer og senere, at de bliver erstattet af grafiske symboler i form af nodehoveder på optegnede nodelinier.

Med mensuralnodeskriftens fremkomst hos Franco af Køln og dens videreudvikling hos Philippe de Vitry i afhandlingen ”*Ars Nova*” fra 1320 indføres taktangivelser og et system til rytmiske

² Stanley Sadie, *New Grove Dictionary of Music*, Volume 13, p. 344.

³ Af græsk: uden mellemrum, dvs. uden angivelse af intervalforhold.

inddelinger gennem symboler for udmålte varigheder såsom maxima, longa, brevis, semibrevis, minima m.fl. Varighederne blev angivet ved nodehovedernes bredde.

Den fortsatte raffinering af notationssystemet gennem 1600- og 1700-tallet skaber et system, der tillader komponisten i stadig højere grad at fastlægge musikken, som den tænkes opført. Heri indgår ikke bare præciseringer af tonernes højder og varigheder, men også en række artikulationstegn i form af fraseringsbuer og sproglige betegnelser såsom *expressivo*, *grave*, *andante* m.fl., samt styrkebetegnelser som *crescendo*, *diminuendo*, *piano* og *forte* etc. Udviklingen betyder, at komponisten på et stadig mere detaljeret niveau kan angive variationer og forsiringer og således i højere grad beskrive musikken som intenderet. Det fremgår af forordet til Geulio Caccinis ”*Nuove musiche e nuova maniera di scriverle*” fra 1614, hvori han begejstret siger, at den nye måde at skrive musikken på betyder, at sangeren nu kan lære og praktisere musikken, uden at han har brug for forfatterens sang. Det vil sige, at sangeren nu kan tilegne sig det foreskrevne uden om den mundtlige overlevering. Sat på spidsen kan man hævde, at musikken ikke længere læres gennem at blive hørt, men ved at blive læst.⁴

Caccinis kommentar minder os om, at den klingende og den noterede musik taler til hver deres sans. Musikken bliver som klingende fænomen i tid afbildet i et visuelt udtryk. Notationens henvisningskarakter hviler derfor på en mulig analogi mellem det visuelle og det auditive indtryk. Hermed nærmer vi os en klassisk repræsentationsproblematik, der handler om, i hvilken grad repræsentationen ligner det, den repræsenterer. En repræsentation

⁴ Tak til Jette Barnholdt Hansen, der har gjort mig opmærksom på Caccinnis betragtninger. Jette Barnholdt Hansen citerer og oversætter Caccinni i artiklen ”Il Canto sulla lira – overhørte klange fra et mundtligt univers”. Se Barnholdt Hansen 2001, p.140.

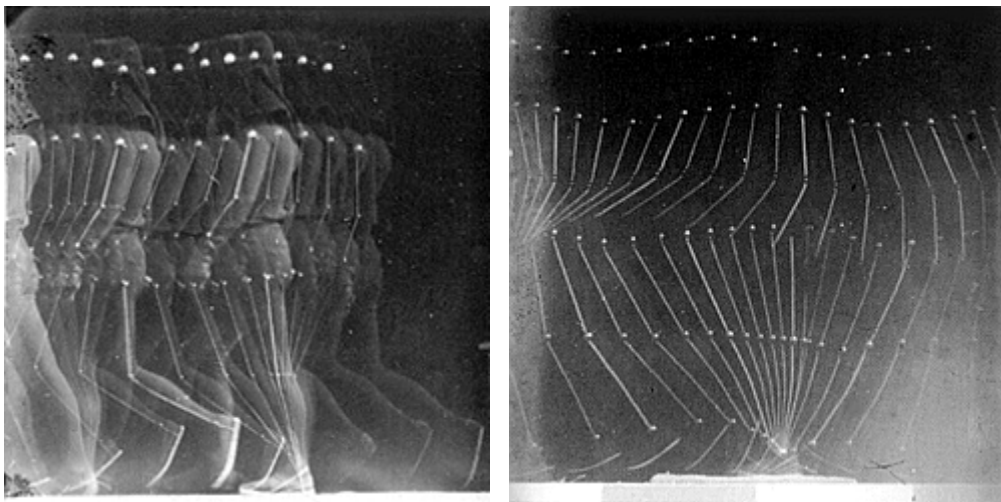
må nødvendigvis altid være en abstraktion og dermed også en reduktion af det, den repræsenterer. Det repræsenterende og det repræsenterede må i en eller anden forstand være isomorfe, men de kan ikke være identiske. Deri består deres ontologiske forskel. Således mister vi altså noget i selve afbildningen. Noget går tabt eller sagt mere neutralt; der er væsensbestemte træk ved den klingende lyd, som ikke kan afbildes i nodebilledet. Det vi oplever i lydbilledet er ikke nødvendigvis det, vi kan angive i nodebilledet.

NODEBILLEDET SOM ET METRISK RUM

Varighedsrelationer skaber rytmer. Vi kan høre, at en tone er længere end en anden tone inden for visse perceptuelle tærskler, og vi kan opstille en rytme ved at sammenstille toner af forskellige varigheder. Vores evne til at høre forskelle i varigheder er bundet til kroppen, hvilket vil sige, at rytmen er somatisk begrundet. Vi oplever selv meget små rytmiske forskydninger som kvalitative aspekter ved musikken, hvilket vi kender fra udtryk som: ”musikken swinger”, ”musikeren ligger forrest på beatet” osv. Vi kan med andre ord artikulere og percipere subtile tidsforskelle i meget små dynamisk fluktuerende enheder. Som noteret fænomen relaterer rytmen sig ikke til kroppen, men til forholdet mellem summariske varigheder, hvis kendetegn er, at en varighed kan beskrives som summen af to eller flere mindre varigheder. Den mindste rytmiske forskydning vi kan repræsentere i nodebilledet er derfor bundet til den mindste rytmiske noterbare enhed, vi opererer med, og det er i forhold til den enhed, at rytmen kvantificeres. Referencepunktet er altså ikke længere kroppen, men den mindste (tællelige) noterbare enhed. Den samme reduktionisme kendetegner også repræsentationen af tonehøjder. I virkeligheden forekommer lyden i en uendelig variation af mulige frekvenser og klangfarver. Men uendeligheden er ikke noterbar.

Kortet er ikke landskabet, så lydens frekvens må placeres i forhold til de musikalske skrifttegn.

Forholdet mellem den kvalitative oplevelse af lydbilledet og den kvantitative bestemmelse i nodebilledet opstår altså ved, at lyden som klingende fænomen i tid oversættes til et visuelt og rumligt fænomen i nodebilledet. Vi kan stoppe kameraet og se verden som en sekvens af enkeltbilleder, der hver især er en tidløs fremstilling. For billedet er et punkt/et nu, der undslipper tiden ved ikke at have nogen udstrækning. Ved at reproducere og definere bevægelsen som punktsammensætninger kan vi omvendt studere og opleve bevægelsen som tidsfænomen, hvilket tidlige fotografiske studier af Etienne-Jules Marey vidner om.



Figur 1: Étienne-Jules Marey: "Motion Capture Suit"¹

Lyden kan vi ikke fikserer i øjeblikksbilleder, for "stopper" vi lyden, er der kun stilheden tilbage. Lyden er ikke et "nu". Den har en varighed og kræver en tid at folde sig ud i, og det er i selve dens udfoldelse, at vi oplever dens kvalitet. Lyden er i den franske filosof Henri Bergsons forstand en "oplevet tid", og det er en tid, som vi har levet, og hvori vi har oplevet lyden folde sig ud. Det er denne oplevelse af lyden som en "varen", som notationen omtyder ved at repræsentere lydens kvalitative karakteristika *som tid* i form

af kvantitative karakteristika ved lyden *i tid*. Tonernes varigheder noterer vi ganske vist, men som relationelle kvantitative tidslængder (den enkelte varighed kan kun tænkes som en varighed i forhold til en referenceenhed) og ikke som kvalitative varigheder. Nodebilledet inddeler tiden triadisk i fortid, nutid og fremtid - i et før, nu og efter. Dermed skaber nodebilledet begreb om et tidsrum og om de enkelte toners placering i dette rum. Rummet er et metrisk rum. Et visuelt udtryk der omformer Bergsons kvalitativt oplevede tid til Newtons kvantitativt udmålte tid.

Forskellen mellem lydbilledet og nodebilledets repræsentation er et spørgsmål om tærskler. Tærskler der enten er forankret i kroppen eller i notationens arbitrære og konventionelle kvantificeringer. Overskridelsen af perceptuelle tærskler oplever vi som kvalitative forandringer, men disse kvalitative forandringer overskrider ikke nødvendigvis notationens tærskler, hvorved de falder uden for systemet som falske eller ikke noterbare udsving. De kvalitative forandringer kan være så små, at de ikke overskrider notationens kvantitative tærskler. I en parafrasering af Gregory Batesons definition på information som forskelle, der gør en forskel, kan man sige, at hvis de kvalitative forskelle eller udsving ikke gør en forskel, hvis de netop ikke overskrider notationens tærskler, vil vi ikke kunne notere dem. Problemet er velkendt for den musiketnologiske forskning, hvor registreringen af ikke-vestlige musikkulturer vanskeliggøres med stavnotationen som repræsentationsteknologi. Kun tærskeloverskridelser inden for systemet kan registreres og angives, hvorfor ”musiketnologen” må gå på kompromis med den akustiske virkelighed. Optegnelsen af mundtlige musikformer kan derfor vanskeligt finde sted med stavnotationen, da notationens summariske rytme ikke er forankret i kroppen, men i en opmåling af varigheder på baggrund af fastlagte mindsteenheder. På samme måde kan udmålingen af tonehøjdere relationer kun ske i kromatiske intervaller. Notationen bygger

således på selve målingens ide, der som en kulturel teknik drejer sig om at:

”sammenligne, om at drage grænser eller om at finde en balance. Som kulturelle aktiviteter er disse sociale teknikker imidlertid også historisk forbundet med centrale formelle begreber indenfor matematikken. Opmåling i forskellige former forudsætter logisk, og fremkalder dermed effektivt, en generel idé om det metriske rum, det vil sige et rum hvori gælder en ensartet afstandsfunktion, hvorved det overhovedet giver mening at måle afstande.”⁵

I nodebilledet kan vi opmåle og repræsentere de egenskaber ved den klingende lyd i forhold til hvilke, vi kan operere med en ensartet afstandsfunktion, og disse er de relationelle egenskaber, altså de melodiske/harmoniske og rytmiske dimensioner.

Som relationelle egenskaber kan vi afgøre, om en tone er højere eller lavere, kortere eller længere end en anden tone, og herved bliver det muligt at fastlægge skalaer for de to dimensioner. Og vel at mærke skalaer, der kan kvantificeres. Intervalforskellen mellem to toner kan bestemmes i et præcist antal af kromatiske halvtonetrin, mens den relative varighed mellem to toner af forskellige længde kan fastlægges som summen af mindre noterbare varigheder. Karakteristisk for tonehøjden og tone-længden er, at de er kvantitative dimensioner, og det er selve kvantificeringen, som vi kan beskrive og aflæse i nodebilledet.

NODEBILLEDET SOM DIAGRAM

Som det fremgår af ovenstående, skaber nodebilledet et metrisk rum, hvor kun de relationelle og dermed kvantitative egenskaber ved den klingende musik er repræsenteret. For at forstå på hvilken måde nodebilledet ligner sit objekt, kan vi ty til det ikoniske tegn og særligt dets karakter af diagram. Som bekendt har tegnet hos Charles S. Peirce tre fremtrædelser: Ikon, indeks og symbol. Iko-

⁵ May & Stjernfelt 1996, p.64.

net henviser til sit objekt i kraft af en lighedsrelation, indekset i kraft af en årsagsrelation og symbolet i kraft af en konventionel, vanebegrundet relation. Ikonet inddeles igen triadisk i billede, diagram og metafor. Billedet ligner sit objekt, fordi det har (sans)kvaliteter til fælles med det, f.eks. en kvalitativ lighed i form eller farve. Diagrammet er en anskueliggørelse af ikke åbenlyst anskuelige (dvs. sanselige) forhold, f.eks. relationer mellem abstrakte størrelser såsom mængder. Diagrammet ligner altså ved at have relationer til fælles med sin genstand, f.eks. en lighed i proportioner. Metaforen henviser til sin genstand via en anden genstand, som f.eks. træets henvisning til slægtskabsstrukturer. Det vanskelige ved i analytisk sammenhæng at anvende de tre tegntyper såvel som den skitserede underinddeling af ikonet er den manglende entydighed i tegnet. Definitionerne har en tendens til at overlape hinanden. Derfor vil et givent henvisningsforhold f.eks. kunne karakteriseres som både billedligt og diagrammatisk. Et tegn er ikke bare entydigt et ikon, et symbol, et indeks og/eller et billede, et diagram eller en metafor. De forskellige tegntyper optræder ofte i det samme henvisningsforhold og er således mere eller mindre fremtrædende aspekter ved den givne tegnrelation.

Som sagt er nodebilledet udtryk for en teknisk udmåling af musikkens relationelle egenskaber: tonehøjde og tonelængde og dermed er nodebilledets karakter af ikon primært orienteret mod diagrammet. Diagrammer er angivelser af relationelle eller strukturelle lighedstegn: ”*der henviser til deres genstand ved en eller anden form for skelettal analyse af den i indbyrdes forbundne dele. Diagrammet udgør så at sige en skitseagtig anatomi over sin genstand*”.⁶ Typiske diagrammer⁷ er kort eller skemaer, og det er

⁶ Stjernfelt 1999, p. 255.

ikke vanskeligt at se nodebilledet som en sådan skelettal anatomi. På samme måde som bykortet mangler byens finere detaljer, har notationen kun udvalgte kendetegn tilfælles med den klingende musik. Abstraktion og reduktion er således afgørende karakteristika ved diagrammet. Det er også karakteristisk for diagrammer, at de udtrykker generelle og ikke individuelle relationer. Naturligvis er et bykort knyttet til den pågældende by, som det tegner et diagram over, men andre diagrammer, såsom det periodiske system eller en kvintcirkel, har ikke denne individuelle reference. Det samme gælder for nodebilledet, der angiver generelle intervaller, vendinger og figurer, der i sig selv er uden individuelle referencer. Ganske vist vil noderne i en bestemt konfiguration, altså som partitur, referere til et bestemt værk, men selv da som en metareference uden det individuelle særpræg, der knytter sig til enhver fremførelse. Det individuelle indgår ikke som en del af musikkens bestemmelse i nodebilledet, hvorfor det overhovedet er muligt at tale om forskellige opførelser og fortolkninger.

I og med at nodebilledet angiver relationelle egenskaber ved den klingende musik, kan det ses som et diagram over og dermed som en *model af* lydbilledet, men det er ikke nødvendigvis en model af noget, der eksisterer i forvejen. Det kan også være en *model for* noget. Diagrammet kan m.a.o. bruges som forlæg. Det system der i udgangspunktet var tænkt som en model af en praksis (et mnemonisk redskab) og dermed som deskription af noget allerede foreliggende kan ligeså vel anvendes til at skabe modeller for samme praksis, altså som præskription. Som ”model for” anvender vi modellen i forhold til en ide, som vi udtrykker gennem systeminterne relationer. Det er altså systemet selv, dets udtryksenheder og deres mulige kombinationer, der udgør rammen

⁷ Min gennemgang af diagrammets generelle karakteristika har grund i den udredning af diagrammet der fremlægges hos [Johansen & Larsen 1994, p. 55-57].

for de mulige strukturer, det kan frembringe. Det er den præskriptive funktion, som begejstrer Caccini, for den betyder, at musikken i kompositorisk forstand kan løsrives fra en mundtlig praksis. Nodebilledet indfører en litterær dimension i musikken, hvorved komponisten i bogstavelig forstand skriver sin musik ved at fastlægge interne relationer mellem musikalske skrifttegn.

Det der nu står for er at undersøge, på hvilken måde det præger musikken, at den bliver skrevet i et nodebillede.

MUSIKKENS LITTERÆRE DIMENSION

I bogen "*Orality og Litteracy*" fra 1982 udreder Walter Ong grundliggende forskelle mellem mundtlige og skriftlige fremstillingsformer. Heriblandt forskellen mellem *additive* og *subordinative* former som han eksemplificerer med skabelsesberetningen i henholdsvis "*Genesis 1-1:5 Douay version*" fra 1610 og "*the New American Bible*" fra 1970. Forskellen er, at de enkelte sætninger i den ældste version er ligeværdige i den forstand, at "og" er bindeord, mens sætningerne i den nye version er indlejret og indføjjet i hinanden gennem bi- og bindeord som "da" (then), "således" (thus) og "imens" (while).⁸

De bærende formprincipper i den mundtlige fremstilling som Duay-versionen er eksponent for, er gentagelse og addition. Som retoriske funktioner skal de sikre og lette forståelsen af det udtalte. Den mundtlige meddelelse bliver herved redundant, da den til stadighed peger tilbage på det allerede udtalte for at understrege, tydeliggøre og vedblivende aktualisere meddelelsens pointe.

Skriften ophæver nødvendigheden af dette hensyn i formuleringen af den skriftlige meddelelse, idet læsningen ikke som aflytningen er underlagt hukommelsen. Der kan m.a.o.

⁸ Ong 1982, p. 37.

arbejdes med strukturer, der ikke behøver tage hensyn til udsigelsen, og som derfor udvikler sig lineært i den forstand, at strukturerne forudsætter (kausalitet) og underordner sig (subordination) hinanden.

Den mundtlige fremstilling har lighed med den episodiske struktur, der udgør et klassisk litterært struktureringsprincip. Den episodiske struktur udgør en fragmenteret fortælling, hvor logiske principper ikke er styrende, men hvor strukturen f.eks. kan være opbygget efter formelen: ”og... og...og ” eller ”og så...og så...og så” etc. Altså netop det additive princip som Ong taler om, og som er udtryk for en sideordning af begivenheder og dermed for en vis tilfældighed og vilkårlighed i tekstens progression. Den skriftlige fremstilling kan i modsætning hertil sættes i forbindelse med et lineært struktureringsprincip, der fordrer fortællingens enhed og helhed og som i aristotelisk forstand må indeholde en begyndelse, midte og slutning, sådan at den udgør en afrundet og afsluttet form. Den lineære form forudsætter en logisk fremadskridende handling, hvor forholdet mellem de enkelte elementer i fortællingen er kausale. Handlingsafsnittene har en årsag og følger logisk efter hinanden. Den skematiske form kunne være ”og så... derfor”, ”hvis... da... medmindre”. Formen er kendetegnet ved subordination, idet de enkelte led er indlejret i og tillige forudsætter hinanden. De er kronologiske i den forstand, at de enkelte afsnit ikke kan udelades eller ombyttes, uden at fortællingens sammenhæng brydes.

Med Ong når vi frem til to overordnede struktureringsprincipper, hvoraf det ene, den episodiske, er karakteriseret ved sideordning, tilfældighed og digressioner, mens den anden, den lineære struktur, er karakteriseret ved logik, kronologi og dermed ved handlingens enhed og helhed. Den episodiske struktur forbinder vi i kraft af dens additive karakter med en mundtlig

kulturform, mens den lineære struktur, der er kendetegnet ved subordination, vokser ud af en skriftlig fremstilling.

De to litterære formtyper har deres lige i musikken i de såkaldte kæde- og udviklingsformer, som Jens Brincker omtaler som to modsatrettede formningsidealer:

" et andet inddelingsprincip er opdelingen i kædeformer og udviklingsformer - undertiden kaldet plastiske og logiske former. {... } I kædeformen (ex rondoformen) er formens inddeling i statiske afsnit, der gennem proportionering, adskillelse, kontrast og lighed afbalancerer hinanden det væsentlige moment. I udviklingsformer (ex sonateformen) spiller de enkelte formleds indbyrdes forhold, f.ex. udtrykt igennem sammenhængen mellem temaer og motiver, der afledes af hinanden, og gennem den dynamiske sammenkædning af formledene, den afgørende rolle."⁹

Hvor kædeformen er karakteriseret ved statiske afsnit, der afbalancerer hinanden ved proportionering, kontrast og lighed, er udviklingsformen karakteriseret ved, at formledene indbyrdes er kausale. Formledene afledes og udledes af hinanden og sammenkædes dynamisk. Udviklingsformen er altså netop af typen ”og så... derfor”, og i den forstand indgår de enkelte formled i en kausal sammenhæng. Formledene kan m.a.o. ikke byttes om, fordi de forudsætter hinanden i en bestemt rækkefølge. Derved får udviklingsformerne lighed med den lineære form. F.eks. må sonatens tema nødvendigvis præsenteres før end det kan udvikles, perspektiveres og repeteres i gennemføringsdelen og reprisen. Sonatens formdele er progressive elementer og kausale størrelser.

En anden af de forskelle som Walter Ong angiver mellem skriftlige og mundtlige fremstillinger er, at den skriftlige fremstilling er analytisk (*analytic*), mens den mundtlige er

⁹ Brincker 1974, p. 28.

aggregativ (*aggregative*)¹⁰. Dermed mener Ong, at den skriftlige fremstilling ikke behøver at tage hensyn til hukommelsen. Det skyldes, at man i den skriftlige fremstilling kan vende tilbage til teksten og læse den samme fremstilling flere gange, mens den mundtlige fremstilling må tage retoriske virkemidler i brug for at sikre, at tilhøreren kan følge med og forstå det sagte. Den analytiske fremstilling fordrer et overblik, som i musikkens verden netop gives i kraft af nodebilledets visuelle natur. Hvor det at fremføre eller lytte til musik er et tidsligt, lineært fænomen, er det i partituret muligt at springe frem og tilbage i nodebilledet og sammenligne vertikale og horisontale strukturer uafhængigt af musikkens tidslige forløb. M.a.o. forløber partiturlæsningen og partiturskrivningen uafhængigt af den kronologiske og sekventielle orden, der karakteriserer musikken som klingende fænomen. Det betyder, at vi kan forstå partiturlæsning og -skrivning som ikke-lineære og atemporale praksiser, der åbner for en diskurs, der løber igennem nodebilledet, og som naturligt centrerer sig om de strukturer og former, det overhovedet er muligt at lægge til rette her.

Konsekvensen ser vi allerede i overgangen fra monodi til polyfoni. Notationens opløsning af tiden i metriske enheder er grundlaget for polyfoniens vertikale og horisontale kompleksitet, der er kendetegnet ved, at komponisten bringer flere melodiske lag til at bevæge sig i forhold til hinanden. Det nødvendiggør en stram koordinering og synkronisering af de forskellige melodiske lag, da sangerne har behov for løbende at kende deres position i værket. Det kræver, at tiden kan præciseres, udmåles og kvantiseres, og det er disse krav, der bliver opfyldt med de Vitrys indførelse af taktlinier og nodeværdier med eksakte varigheder. Victor Zuckerkandl formulerer det således:

¹⁰ Ong 1982, p. 38.

“... den vestlige musik i det andet årtusinde, vores musik. Kun den har påført sig selv tidens lænker i form af et fast metrum og det netop på samme tidspunkt, som den gjorde sig klar til at tage det betydningsfulde skridt ind i polyfonien. Så længe der kun er en enkelt stemme involveret kan hver enkelt tone frit tildeles en vilkårlig varighed. Men hvis flere forskellige stemmer skal følges ad, må deres bevægelse, i medgang og modgang, reguleres af en tidsstandard. [...] Vores notation tager højde for det. [...] Symbolerne udtrykker, at tonerne underkaster sig en fælles udmåling af varigheder. Ingen andre varigheder er tilgængelige end de der kan repræsenteres som simple multipla og brøkdele af den bestemmende temporal enhed, der fremkommer når vi tæller takten.¹¹

Synkronisering og koordinering af flere samtidige stemmer nødvendiggør altså den opmåling af tiden, som notationen tilvejebringer, og det skaber plads for nye musikalske udtryksformer som polyfonien.

Et andet og langt senere eksempel er den dodekafone musik og dens afart serialismen, hvis strukturelle formkrav kun kan opretholdes i nodebilledet. Den dodekafone musiks ligeberettigelse af den kromatiske skalas tolv toner, dens krav om at alle toner skal bruges før en gentagelse er mulig, eller at række transformationer sker som bestemte omvendinger af den originale række, er formalismer med en indbygget automatik, som lægger musikken til rette, og som ikke er forankret i en hensyntagen til det perceptuelle niveau. De strukturelle aspekter der i en analytisk og kompositorisk praksis kan fremdrages og nedfældes som kvantitative relationer i partituret, eksisterer ikke nødvendigvis som hørbare former, altså som kvalitative fænomener. Hvem kan høre forholdet mellem originalen og dens krebsomvendning?¹² Såvel den dodekafone musik som serialismen udstiller en analytisk tankegang, der ikke blot opløser de

¹¹ Zuckerkandl 1956, p. 159-160.

¹² En krebsomvendning er inden for den dodekafone musik et udtryk for, at en given tonerække varieres ved at afspille den i baglæns rækkefølge.

musikalske fænomener i nye målbare dimensioner, men som i serialismens amerikanske variant – ”Set-teorien” - oven i købet omsætter musikalske fænomener til tal. Det accelererer og forstærker den analytiske tankegang, fordi det åbner for formaliserede beskrivelser og manipulationer gennem matematiske funktioner.

Sammenfattende kan vi fastslå, at nodebilledet:

1. som diagram indfører en kvantificerende dimension i musikken.
2. som en visuel repræsentationsform og et metrisk rum (i kraft af at være et diagram hvor relationer mellem interne symboler kan kvantificeres) gør det muligt at kaste et analytisk blik på musikken og arbejde med kausale og subordinerede strukturer til forskel fra den mundtlige fremstillingens forankring i gentagelse og addition.
3. med udgangspunkt i det metriske aspekt åbner op for nye musikalske udtryksformer som polyfoni og for formaliserede udtryksformer som vi bl.a. finder det i serialismen.

Hensigten med min gennemgang af nodebilledet har været at vise, at musik udvikler sig gennem og i kraft af nye repræsentationsformer. F.eks. giver nodebilledet, med dets kvantificering af musikkens dimensioner, mulighed for at synkronisere mange samtidige stemmer, og herved opstår der nye musikalske udtryksformer med en stigende vertikal fokusering på musikkens harmoniske dimension til følge.

I anden del skal vi se, hvordan computerbaserede repræsentationsformer både viderefører den forståelse af musik som følger af nodebilledet og tilbyder andre måder at organisere musikken på.

II DEL. SOFTWARE-INTERFACES

Blandt nutidens softwareprogrammer til musikproduktion er Audio-/midisequencerne de mest udbredte. Dominerende på markedet er Digidesign med ”Protools”, Emagic/Apple med ”Logic” og Steinberg med ”Cubase”. Andre populære programmer er ”Reason”: Propellerheads softwaresynthesizer og –sampler. Et knapt så udbredt, men dog alligevel populært program er SuperCollider, der er et programmeringssprog til lydgenerering i realtid udviklet af James McCartney.

Jeg skal efterfølgende analysere nogle af disse programmer og ud fra to paradigmer klargøre de konceptuelle modeller, der ligger til grund for programmernes udformning, samt anføre hvilke ældre teknologier, de er modelleret over. I programmernes interface kan vi således iagttage en remediering af ældre teknologier som både funktionelle modeller og grafiske udtryk. Min analyse tager afsæt i to artikler af henholdsvis Alessandro Ludovico og Andrew Gerzso, der begge giver bud på paradigmatisk forståelse af musik i forskellige typer af musiksoftware. Alessandro Lodovico taler om et sequencerparadigme, mens Andrew Gerzso taler om, hvad jeg i denne sammenhæng vil kalde for et synthesizerparadigme. Som jeg skal vise anvendes computeren i sequencerparadigmet som en visuel redigeringsmaskine, mens den i synthesizerparadigmet anvendes som en generativ maskine. Anvendt som en visuel redigeringsmaskine komponerer man sin musik gennem at manipulere grafiske repræsentationer i computerens interface. Anvendt som generativ maskine genererer og udfolder computeren selv musikalske strukturer på baggrund af at foretage beregninger ud fra regler og betingelser, der er formuleret i en algoritme. Disse regler er specificeret af komponisten.

De to forskellige måder at anvende computeren på giver anledning til to væsensforskellige måder at komponere på.

SEQUENCEREN OG DEN VISUELLE COMPUTER

I artiklen "The Sequencer Paradigm" fra 2004 hævder Alessandro Ludovico, at sequenceren er den bærende model for måden at komponere musik på i dag. Ja faktisk anskuer han sequencerens *modus operandi*: den kontinuerlige scanning af forskellige typer af data på mange samtidige spor, som en grundlæggende kulturel mekanisme. Som andre digitale udtryk er også musikken et sammensat fænomen af diskrete elementer (midi-events og lydsamples), der i sequenceren afvikles i en fastlagt rækkefølge. Som en systematisering af generelle kendetegn ved den sequencerbaserede musik, kan vi opstille følgende grundtræk:

1. Kontrol og data dikotomi: En klar adskillelse af kontrol og data
2. Loop: Gentagelsen som en grundlæggende operation.
3. Medieobjekter: En fremstilling af data som diskrete elementer.
4. Synkronisering: En præcis organisering af data i tidsdomænet.

De fire grundtræk er ikke særegne for softwaresequenceren, men nedarvede fra en række tidligere musikalske medier og instrumentteknologier såsom nodebilledet, klokkespillet, den analoge båndoptager, det mekaniske klaver m.fl. Afviklings- og repræsentationsprincipperne bag disse teknologier er indlejrede i sequencerens funktionaliteter, ligesom teknologierne er synlige i sequencerprogrammets interface som visuelle metaforer.

KONTROL & DATA

En sequencer er et apparat til optagelse og afspilning af musik med en indbygget og programmerbar hukommelse. Sequenceren består af:

1. En *afspilnings-/aflæsningsmekanisme*, der i softwaresequenceren er angivet som en markør, der kontinuerligt scanner sequencerens spor.
2. Et redigerbart indhold der er opbevaret på et *lagringsmedie*. I softwaresequenceren er det redigerbare indhold angivet grafisk som enten noder, blokke eller audiofiler.
3. En *afviklings-/kontrolmekanisme*, der i softwaresequenceren er angivet ved de ”play”, ”stop”, ”rewind”, ”forward” mekanismer, som vi kender fra den analoge båndoptager.

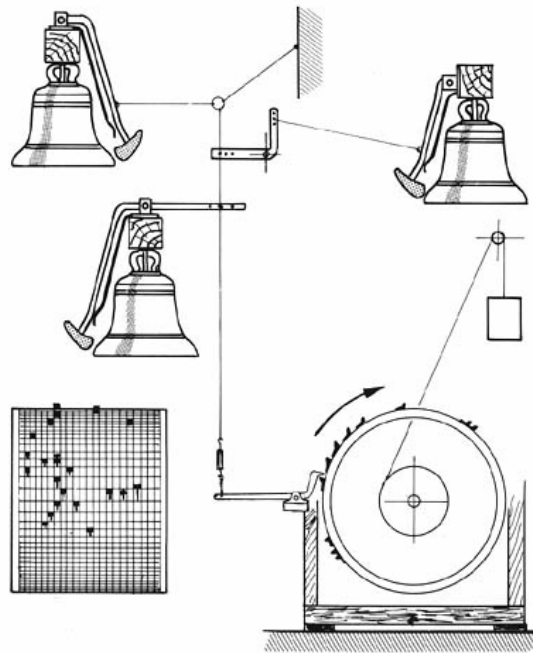
I historisk perspektiv er der flere eksempler på apparater og musikinstrumenter, der har sequencerens grundtræk.

Det første er det programmerbare klokkespil, som man kender allerede fra 1200-tallet¹³, og som lancerer et princip til mekanisk afvikling af musik, som vi senere finder i selvspillende klaverer og i spilledåser. Interessant er det også, at det er den samme mekanisme som i begyndelse af 1800-tallet udnyttes af Joseph-Marie Jacquard til automatisering af hulkortstyrede væve, og som herfra finder vej til Charles Babbage, der angiver hulkortsystemet som en metode til kontrol af beregningsprocesser.

Klokkespillet består af en roterende cylinder med lige rækker af huller, der udgør spillets lagringsmedie. I hullerne kan man placere pløkker af træ eller metal, der i dette tilfælde repræsenterer det redigerbare indhold. Det betyder, at man til stadighed kan komponere nye melodier ved at placere de enkelte pløkker i nye mønstre én ad gangen. Den måde at komponere på er indarbejdet i nutidens digitale sequencere som en ”step mode” funktion, hvor man ligeledes og på samme omstændelige vis angiver tonernes

¹³ Roads 1996, p. 662.

tidslige placering én efter en¹⁴. Når cylinderen roterer griber pløkken fat i en ”arm”, der udløser et anslag på en klokke. Armen, dens videre forbindelse til klokken, samt det drejhjul hvormed klokkespillet betjenes manuelt udgør klokkespillets afspilnings- og afviklingsmekanisme.

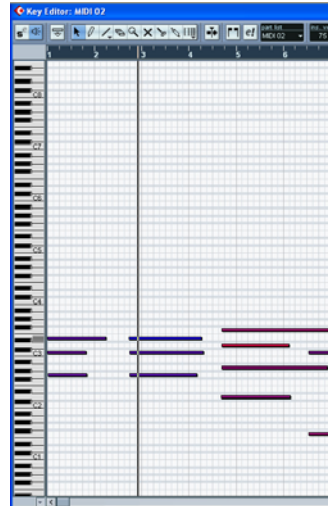


Figur 2: Figuren illustrerer princippet i det mekaniske klokkespil. Når tromlen roterer, griber de enkelte pløkker fast i en arm der er forbundet til en klokke, hvorved et slag på klokken udløses. Billedet er hentet fra: <http://odur.let.rug.nl/~koster/musicbox/musicbox3.htm>

Klokkespillets klare opdeling mellem kontrol og data kan vi iagttage i sequencerens interface. Særlig tydeligt i den repræsentationsform, der direkte mimer klokkespillet og/eller det selvspillende klaver, og som viser alle indspillede (midi)data som grafiske blokke i en 2-dimensional afbildning, akkurat som var det rækker af pløkke på klokkespillets cylinder. Den vertikale placering angiver tonehøjden som et tangentnummer fra C0 til C8. Den horisontale placering angiver midi-tonernes rytmiske

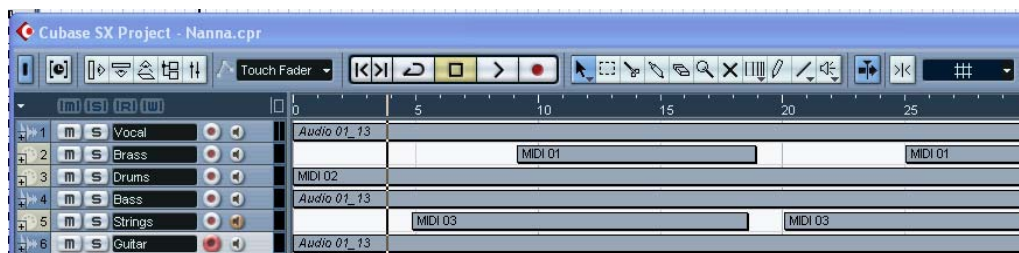
¹⁴ Roads 1996, p. 663.

organisering, mens blokkenes skiftende længder angiver de enkelte toners varighed.



Figur 3: De grundlæggende mekanismer i det mekaniske klaver og klokkespillet er synlige i softwaresequencerens interface. Her i Keyeditorvinduet i Cubase 2.0.

Det andet eksempel er flersporsbåndoptageren, hvis funktionalitet og klare opdeling mellem et redigerbart indhold og en afspilnings- og afviklingsmekanisme ligeledes er visualiseret i sequenceren. Som på det analoge bånd ligger sequencerens indspillede data (audio og/eller midi) på rækker af spor organiseret vertikalt under hinanden og med en horisontal akse, der angiver en tidsakse i enten realtid eller i takter. På samme måde som et bånd glider hen over et læsehoved, angiver markøren en sekventiel aflæsning af de midi- og audiodata, der er optaget på de enkelte spor.



Figur 4: Flersporsbåndoptagerens funktionalitet er visualiseret i det grafiske interface som vertikale rækker af spor. På de enkelte spor er de indspillede data synlige som midi- eller audiodata. Her ses projektvinduet i Cubase 2.0.

GENTAGELSE/LOOP

Den fortløbende gentagelse af melodiske og rytmiske fraser er et væsentligt karakteristikum ved den sequencerbaserede musik. Gentagelsen har form af et loop, hvorved den enkelte, ofte korte frase gentages kontinuerligt og på nøjagtig samme måde. I datalogisk forstand er sequencerens loopfunktion en løkke, hvori antallet af gentagelser er angivet på forhånd, og hvor det man gentager allerede er fastlagt i en bestemt rækkefølge. Senere skal vi se, hvordan man i algoritmisk komposition, hvor computeren anvendes som en generativ maskine, arbejder med iteration, der er en anden type gentagelig datastruktur.

Gentagelsen er en grundlæggende musikalsk form, som vi finder i al musik, og som vi også har notationstegn for. At gentagelsen er så bærende et princip i den sequencerbaserede musik har dog også at gøre med konstruktionsprincipperne bag de teknologier som softwaresequenceren funderer sig på, og med de æstetiske praksisformer som de ligger grunden for. F.eks. er gentagelsen allerede givet i klokkespillets opbygning i kraft af det cylindriske lagermedie.

Den analoge synthesizer, der fik sit kommercielle gennembrud i slutningen af 1960'erne, var opbygget af spændingsstyrede moduler (oscillatorer, filtre og forstærkere), der indbyrdes kunne forbindes med kabler. Analoge sequencere kunne ligeledes tilkobles med henblik på at automatisere lydlige og/eller musikalske forløb. På de analoge sequencere kunne man ved hjælp af drejeknapper angive en spændingsstyrke til brug for styring af et vilkårligt lydparameter, f.eks. tonehøjde, tonelængde eller tonestyrke, men i begyndelsen højst 8 eller 16 forskellige spændingsstyrker ad gangen. Sequenceren kunne indstilles til kontinuerligt at løbe igennem de indstillede spændingsstyrker og videresende dem enkeltvis og i rækkefølge til den enhed, hvortil sequenceren var tilkoblet. Herved blev det muligt at loope

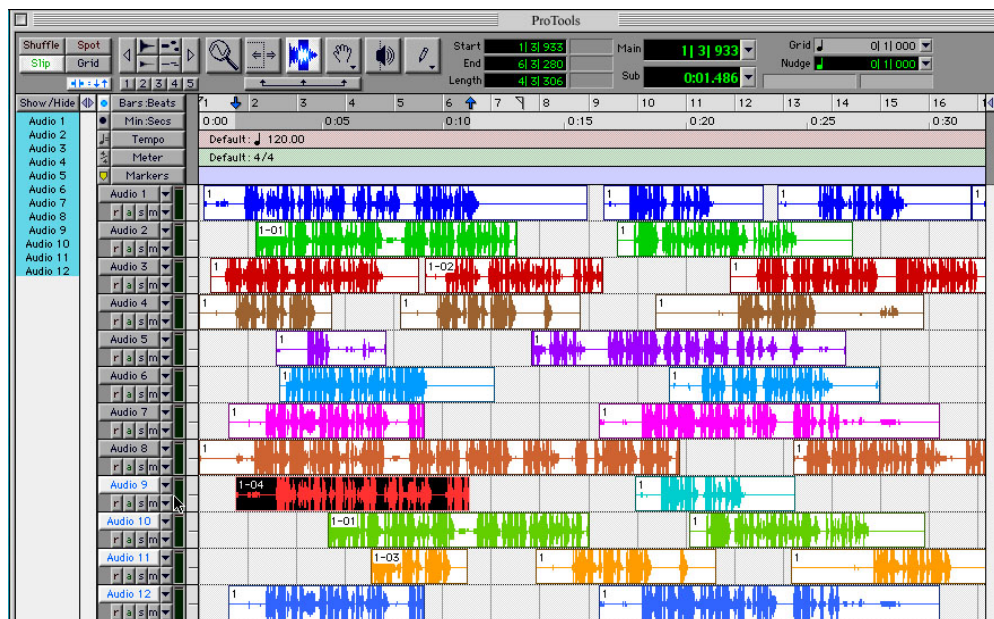
sekvensen og repetere f.eks. en rytmisk figur og gøre denne automatiserede sekvens til en grundlæggende funktion i opbygningen af musikken. Gentagelsen var således en effektiv udnyttelse af sparsomme ressourcer, for kunne man ikke gentage den indstillede sekvens, ville man være nødsaget til serielt at tilkoble flere sequencere. I tidlige digitale sequencere, samplere og synthesizere, hvor størrelsen på computerens RAM var bestemmende for, hvor lange lydfiler man kunne lagre, var loopet ligeledes en hensigtsmæssig måde at udnytte mediets begrænsninger på. Ved at loope og langsomt sænke volumen på et kort klaversample kunne man simulere en tones udklingning, uden at det var nødvendigt at sample hele den naturlige klang, og ved at optage og repetere fire takter af et trommebeat kunne man opbygge et trommespor på baggrund af en kort sekvens. Kort sagt, er loopet en hensigtsmæssig måde at udnytte de tekniske begrænsninger, men øjeblikkeligt bliver det også en ny æstetisk udtrykskraft. Det kan man observere allerede i den eksperimenterende kunstmusiks brug af båndsløjfer og/eller cirkelformede (i modsætning til spiralformede) pladeudskæringer hos komponister som Pierre Schaeffer, John Cage, Steve Reich m.fl., samt i store dele af populærmusikken i form af repetitive beats/rytmer.

MEDIEOBJEKTER

Forestillingen om det singulære lydige objekt er både i teknisk og æstetisk forstand knyttet til den analoge lydoptagelse. Muligheden for gentagende aflytninger (loop i form af båndsløjfer eller båndoptagere med flere afspilningshoveder) og simple manipulationer ved hjælp af båndoptagerens afspilningsfunktioner (variation af båndhastighed og retning) gjorde det muligt at analysere og bearbejde optaget lyd. Herved blev lyden et objekt i både fænomenologisk og konkret forstand idet den nu i

repræsenteret form kunne løsrives fra sin oprindelige kontekst og behandles som et formbart materiale.

I softwaresequencerens grafiske interface er objektmetaforen central og meget fremtrædende. Musikken er afbildet som et sammensat resultat af diskrete lydobjekter, der ligner stumper af opklippede bånd.



Figur 5: I sequencerens oversigtsvindue fremstår de enkelte lydsamples som singulære objekter, der frit kan flyttes rundt på de enkelte spor. Her taget fra Protools LE

Som grafiske repræsentationer fremstår de optagede lyde som singulære objekter, der frit og i vilkårlige rækkefølger kan placeres og flyttes rundt på de enkelte spor. Lydobjekterne kan underkastes forskellige former for analytiske nærstudier og således iagttages i forskellige opløsninger og i forskellige repræsentationsformer (spektrogram, sonogram m.fl.). Det grafiske interface skyder sig herved ind som et lag mellem den klingende lyd og den digitale repræsentation, hvorved lyden som noget nyt bliver et billedobjekt. Som følge heraf bliver vores omgang med lyden præget af et kinematografisk blik, der kommer til udtryk i de muligheder for dynamisk fokusering og perspektivering (klip,

vinkel, zoom etc.), som softwaren tilbyder i sin egenskab af visuelt interface.

SYNKRONISERING

Som navnet siger, angiver sequenceren en sekventiel rækkefølge af enkeltstående hændelser og dermed en tidslig organisering, idet nogle hændelser nødvendigvis må komme før og/eller efter andre. Sagt på en anden måde er musikken i en sequencer organiseret på samme måde som i et nodebillede, i kraft af at være repræsenteret som rækker af enkeltstående hændelser i tid. Muligheden for at arbejde vertikalt med flere samtidige hændelser i form af et vilkårligt antal spor er også til stede i sequenceren. Hændelserne kan synkroniseres, så de starter samtidigt, eller de kan bringes til at bevæge sig frit i forhold til hinanden, men altid i relation til en underliggende puls. Også det kender vi fra nodebilledet som homo- eller polyfon organisering. Det nye ved sequenceren er, som Ludovico bemærker, dens tidslige præcision. I en sequencer er det muligt at angive en tidslig positionering på samplingsniveau. Det vil almindeligvis sige 1/44100 del sekunds nøjagtighed. Det er langt ud over, hvad der menneskeligt er muligt at høre, men på samme måde som indførelsen af taktlinier var en forudsætning for organiseringen af den polyfone musik, er sequencerens stramme synkronisering, der tillader afviklingen af et i princippet uendeligt antal spor, garant for nye æstetiske muligheder. Muligheder som den canadiske komponist Paul Dolden udnytter i sit værk "The Walls of Jericho" fra 1991-92 ved at arbejde med op til 800 samtidige lydspor! Sammenfattende kan man sige, at sequenceren opererer i *tidsdomænet* på samme måde som nodebilledet og båndoptageren, men den præcision hvormed de enkelte musikalske hændelser kan fastlægges er langt større.

ÆSTETISKE UDTRYK

Som vi har set, modelleres den moderne audio-/midisequencer i sin funktionalitet og i sit grafiske design over flere tidligere medier som det mekaniske klokkespil, nodebilledet og flersporsbåndoptageren. Musik begribes i den model som en sekvens af enkeltstående hændelser i form af lydfiler og/eller miditoner, der på vilkårlige rækker af lydspor fikseres på en tidslinie. De enkelte hændelser optræder i interfacet som individuelle medieobjekter, der horisontalt og vertikalt kan placeres frit og studeres i forskellige grafiske repræsentationer. Som objekter kan de ”skæres op” og ”skæres til”, omdøbes og tilpasses. Herved muliggør og mangfoldiggør den digitale sequencer de æstetiske operationer, som komponister som Pierre Schaeffer og Steve Reich blandt andre udviklede med de analoge medier, og som resulterede i indførelsen af montagen som musikalsk form og i en musikalsk minimalisme i form af repeterende mønstre.

Med den digitale audio-/ midisequencer får montagen fornyet aktualitet inden for musikalske genrer som techno, dance og electronica. Sequencerens grafiske interface danner her ramme om en ny form for digital montage og softwarecollage, i hvilken sammenstillingen af individuelle samples i større formale strukturer foregår som synlige operationer.

Som en sammensat struktur af individuelle og manipulerbare medieobjekter er det til stadighed muligt at rekonfigurere musikken i nye blandingsformer som mix og remix. Herved bliver det, der i én sammenhæng lægges frem som et færdigt værk omgående et materiale for andre. I sådan en montage- og mix-kultur bliver indsamling, udvælgelse og montering væsentlige kompositoriske operationer.

I bogen ”*The Language of New Media*” udpeger Lev Manovich ”*Selection*” og ”*Compositing*” som de to operationer, der bedst kendetegner de nye digitale medier og dermed den

aktuelle mediekultur.¹⁵ Som medieoperationer angiver ”*Selection*” og ”*Compositing*”, at nye medieudtryk opstår ved, at man udvælger og sammenstiller allerede eksisterende materialer i form af diskrete medieobjekter. På samme måde som man komponerer digital musik ved at sammenstille lydsamples, består et digitalt billede af sammensatte former som følge af de digitale billedebehandlingsprogrammets ”*layer*”-funktioner. Det samme princip gælder for den hypertextlitteratur, der sammenbinder tekststumper i hypermediale linkstrukturer, som vi kender det fra internettet. Manovich pointerer, at det digitale medie i kraft af den diskrete repræsentation og computersoftwarens grundliggende programoperationer reaktualiserer den historiske avantgardes formsprog. Avantgardens indførelse af montage og collage, dens arbejde med ”*objet trouvé*” og ”*cut-up*” teknikker m.m., er operationer, der er indlejret i computerens grundliggende editeringsfunktioner: ”*cut*”, ”*copy*”, ”*paste*”, ”*search*”, ”*undo*”, ”*select*”, ”*clear*” osv. Udvalget og sammenstillingen som kunstneriske udtryksformer træder i stedet for modernismens krav om originalitet. Det handler ikke længere om de ”*nie erhörte Klänge*”, men om det aldrig sammenstillede. Dermed ligger mediets signifikans i, at det til stadighed gendanner og omformer tidligere medieudtryk i stadig nye sammenhænge og blandings-former.

Inden for sequencerparadigmet og den æstetiske praksisform der knytter sig hertil, er det computerens potentiale som redigeringsmaskine, der udnyttes, og det er et potentiale, der er forankret i og synliggjort igennem sequencerens visuelle interface.

¹⁵ Manovich 2001, p.123 - 161.

SYNTHESIZEREN OG DEN GENERATIVE COMPUTER

Sequenceren er vel nok det mest udbredte musikprogram, men naturligvis er der softwareprogrammer, der kalder på andre paradigmatisk forståelser af musik.

I artiklen "Paradigmes and Computer Music" fra 1992 antyder Andrew Gerzso to overordnede paradigmer i den musiksoftware, han beskriver. Det første paradigme bindes til sequenceren og den musikforståelse, der karakteriserer andre tidsbaserede medier som båndoptageren og nodebilledet. Altså det paradigme som vi netop har gennemgået. Det andet paradigme er forankret i det forhold, at lyd som bølgeform kan udtrykkes i en matematisk funktion. Herved knytter paradigmet an til den musikforståelse, der præger den tidlige udvikling af software til syntetisk generering af lyd inden for universitære og andre forskningsbaserede institutioner som Stanford, Bell Labs og IRCAM. Gerzso omtaler paradigmet som "music as mathematical function" og refererer til programmet Music V som eksemplarisk for paradigmet. Music V er det første sprog til programmering af lyd og udviklet af amerikaneren Max Mathews i begyndelsen af 1960'erne.

I Music V foregår programmering af lyd ved kald af matematiske funktioner, der i programmet kaldes for "unit-generatorer". Unitgeneratorerne kan enten generere eller modificere et signal. Unitgeneratoren "Oscil" frembringer således en tone, når man angiver parameterverdier for frekvens, styrke, bølgeform og fase. Det er altså ved fastlæggelse og styring af parameterverdier, at man som bruger af Music V kontrollerer og genererer lyd.

Som modsætning til sequencerparadigmet og den indlejrede opfattelse af musik som diskrete hændelser (objekter) organiseret i tid, mener jeg vi med god ret kan kalde Gerzso's paradigme for et synthesizerparadigme. Musikken/lyden fremstår her som et signal,

der flyder som vand i rør rundt mellem forskellige indbyrdes forbundne enheder så som oscillatorer, filtre, envelopegeneratorer, forstærkere m.fl. Disse enheder påvirker alle lyden på forskellig vis. Nøjagtigt som vi kender det fra en hardwaresynthesizer.

Music V tilhører en type af software, der har været brugt til komposition af musik i mere klassiske computermusik-sammenhænge, men som i dag er på vej ud i bredere og mere populærmusikalske sammenhænge gennem programmer som Csound og SuperCollider (begge senere generationer af Music V), samt MaxMSP, der er et grafisk programmeringssprog til komposition af interaktiv musik. For den type af software vil jeg hævde følgende karakteristika:

1. Modulæropbygning: En fremstilling af data som et flow af informationer mellem singulære moduler
2. Ophævelse af kontrol- og data-dikotomien: Det tydelige skel mellem kontrol- og dataniveauer forsvinder.
3. En mulighed for hændelsesbaseret styring af musik.

MODULÆROPBYGNING

Csound er et scriptbaseret program, hvor man linie for linie udfærdiger et script, som programmet enten oversætter og gemmer som en lydfil på harddisken eller afvikler i realtid (rendering). Nedenfor kan man se det script, man skal skrive for at opbygge og afvikle et simpelt instrument (MyInstrument), der ved kald af en oscillator spiller en sinustone i monoformat. Volumen på sinustonen kontrolleres over tid af en envelopegenerator kaldet "aenvelope".

Scriptet er opdelt i to selvstændige filer. Et "orkester" og et "partitur". I orkesterfilen definerer og opbygger man de instrumenter, der indgår i orkesteret ved at kalde forskellige

indbyggede funktionaliteter. At tale om et instrument er en smule misvisende, for man frembringer lyd ved at opbygge enkeltlyde ud fra forskellige synteseprincipper og ikke ved at definere en lydkilde i form af et instrument. I partiturfilen angiver man hvilke instrumenter, der til givne tidspunkter eller under bestemte forhold skal spille, samt hvor længe. En måde at forstå partituret på er at betragte den som det sted, hvor vi henholdsvis tænder og slukker for signalet, åbner og lukker for ”vandet”.

ORKESTER ”MYINSTRUMENT”

```

sr = 44100                ; samplingsfrekvensen
kr = 4410                 ; kontrolrate
ksmps = 10               ; sr/kr
nchnls = 1               ; 1 = mono, 2 = stereo

instr1                  ; instrument nummer 1
aenvelope                ; envelope med et attack på 0.5
                        ; sekund, et sustain på 1 sek. Og
                        ; et decay på 0.5 sek.
atone                    ; tallene angiver amplitude,
                        ; frekvens og bølgeform.
out                      ; her angiver vi at vores lyd som
                        ; hedder ”atone” skal sendes til
                        ; lydkortets udgang.

endin                 ; afslutter instrumentet

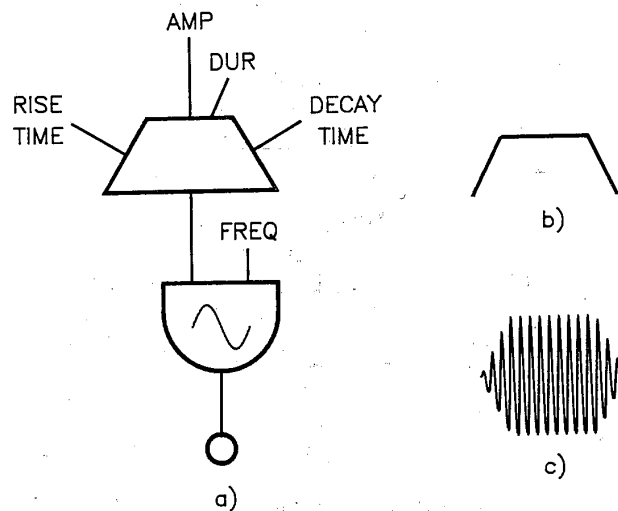
```

PARTITUR ”MYINSTRUMENT”

; Funktionsnr.	”Uploadtime”	Tabelstørrelse	Gen. Rutine	Gen. parameter1
f1	0	1024	10	1
; Parameter 1 ;(instrument nr.)	Parameter 2 Starttid	Parameter 3 Varighed		
i1	0	2		
e				

Det virker utvivlsomt noget omstændeligt at skrive et script som ovenstående blot for at generere en simpel sinustone. Imidlertid skal man forstå scriptet som et variabelt og fleksibelt instrument, der giver mulighed for at kontrollere musikken/lyden på flere detaljerede niveauer, når det først er defineret og udbygget.

Således er alle parametre dynamiske. Det betyder, at vi ud over at ændre parameterværdierne til nye faste værdier kan styre amplitudeudviklingen over tid ved hjælp af en anden funktion som eksemplet viser (brugen af "envelope" til at styre tonens amplitudeforløb over tid) og som grafisk er angivet på figuren nedenfor. At interfacet betjener sig af numeriske værdier betyder endvidere, at amplitudeudviklingen kan kontrolleres præcist i modsætning til at tegne en graf eller bevæge en slider i et grafisk interface.



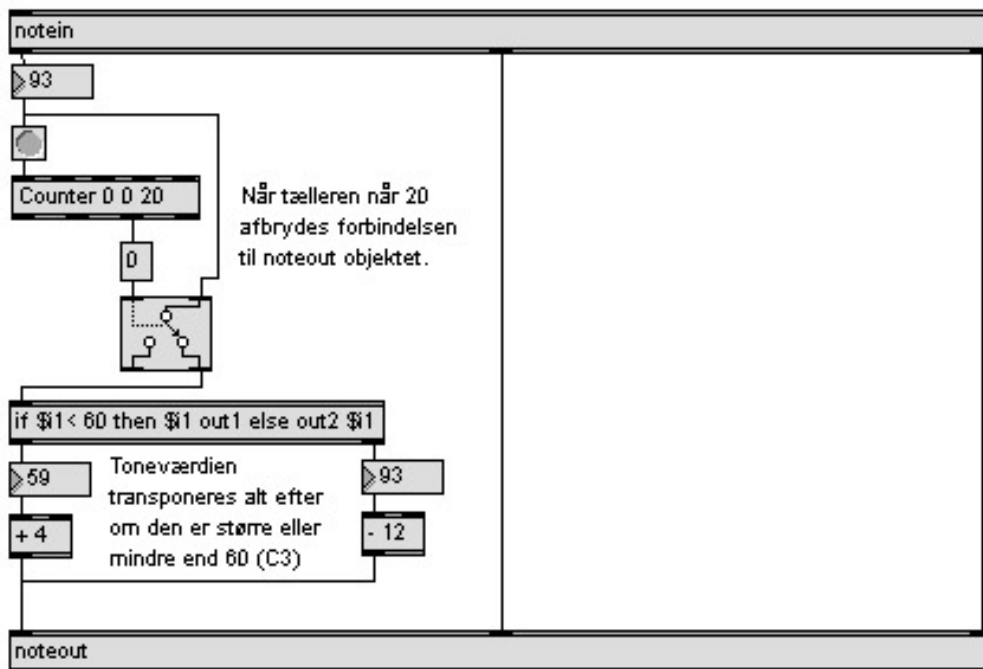
Figur 6: Grafisk illustration af Csound scriptet ovenfor. Amplituden på en sinustone kontrolleres af en envelope. Figuren er taget fra Dodge & Jerse, 1985, p.71

Programmet Max er udviklet i 1992 af David Zicarelli og Miller S. Puckette i samarbejde med IRCAM. Navnet er en hyldest til Max Matthews, der som nævnt udviklede Music V. Max er et grafisk programmeringsmiljø udviklet til realtidsbaserede og interaktive musiksammenhænge. Programmet kan i realtid modtage, procesere og afvikle midisignaler fra eksterne enheder som midikeyboards og/eller andet midiudstyr. Senere er programmet udviklet til på lignende vis at kunne håndtere audiosignaler og navnet ændrede sig til MaxMSP.

MaxMSP består af en lang række af grafiske moduler med en eller flere ind og/eller udgange. Modulerne kan indbyrdes sende og modtage beskeder, hvorved der opstår et flow af signal- og kontrolbeskeder i form af talværdier, symboler og lister mellem de enkelte moduler. Nogle moduler er simple matematiske operationer, der adderer, subtraherer eller multiplicerer værdierne i det signal eller den midihændelse, der flyder gennem modulet. Andre er mere avancerede og kan som matematiske funktioner syntetisere eller analysere et givent signal. Der er også moduler, der alene organiserer beskeder ved at starte og stoppe, om dirigere eller tilbageholde dem, og nogle moduler der modtager og/eller afsender signaler til eksterne enheder som sensorer og aktuatorer.

I MaxMSP opbygger man sit program (et Maxpatch) ved at trække kabler (patch cord) mellem de moduler, der skal henholdsvis sende og/eller modtage beskeder. Kablerne angiver i grafisk form signalvejen mellem de enkelte moduler.

Figur 7 nedenfor viser et program, hvor et midisignal fra et eksternt midikeyboard modtages i modulet "notein". Midisignalet indeholder værdier for den enkelte tones højde og varighed, samt den midikanal, den er sendt på. Alle indkomne værdier af tonenummeret, der er mindre end værdien 60 (C3) transponeres en stor terts op, ved at værdien 4 (svarer til fire halvtonetrin) adderes til det pågældende tonenummer. Alle værdier af tonenummeret over 60 transponeres en oktav ned, ved at værdien 12 subtraheres fra det pågældende tonenummer. De transponerede toner ekspederes videre til et modul, der sender værdierne ud som tonenumre, f.eks til en ekstern synthesizer. Processen fortsætter indtil programmet har modtaget i alt 20 toner, hvorefter det stopper.



Figur 7: Et eksempel på et simpelt MaxMsp-patch. De enkelte moduler rummer forskellige program-funktioner. Kablerne i mellem angiver signalvejene.

Som det fremgår af eksemplerne ovenfor, er musikken og lyden i synthesizerparadigmet et resultat af et flow af data mellem singulære moduler med særegne funktionaliteter. Musikken bygges op ved at sammenkoble moduler og dirigere signaler rundt mellem dem. Umiddelbart skulle man tro, at kald af det enkelte instrument i partiturdelen i Csound svarer til at fastlægge en sekvens af singulære hændelser i audiosequenceren, men instrumentkaldets funktion skal snarere opfattes som en mekanisme, der tænder og slukker for signalet/ ”åbner og lukker for vandet”.

Hvor det i sequenceren er de enkelte dataobjekter i form af miditoner og audiofiler, der er musikkens byggesten, er det i Csound og MaxMSP de enkelte programfunktionaliteter (moduler og unitgeneratorer). Programstrukturen i Csound og MaxMSP svarer derfor til tidlige analoge synthesizere, hvor oscillatorer,

forstærkere og filtre var selvstændige fysiske kasser, som man forbandt vha. kabler.

KONTROL & DATA:

Som tidligere anført er sequenceren grundlæggende en afviklings- og redigeringsmekanisme. Musikken bygges op ved at placere enkelte hændelser/objekter (samples og/eller noder) i et tidsligt forhold til hinanden. I sequenceren er der ligeledes en klar adskillelse mellem det indhold, vi organiserer, og de værktøjer vi har til rådighed til at påvirke og afvikle indholdet. Der er med andre ord en klar adskillelse mellem kontrolmekanismer og dataindhold. I programmer som Csound og MaxMSP er den forskel uklar. På samme måde som i en audiosequencer kan vi i Csound eller MaxMSP importere en lydfil i programmet. Lydfilen bliver hermed et dataindhold, som vi kan afvikle. I Csound og MaxMSP kan vi dog vælge også at bruge lydfilen til at kontrollere afviklingen af et andet indhold. Amplitudevariationen i den importerede lyd kan vi f.eks. bruge til at kontrollere frekvensen på et andet signal, eller vi kan bestemme os for, at en del af programmet først skal afvikles, når lydfilen er færdigspillet, eller når dens amplitude eller frekvens falder under et bestemt niveau. Det vil svare til, at en sequencer først skulle afspille musikken på spor tre, når en given lydfil på spor 1 var færdig. Det bringer os videre til en anden afgørende forskel mellem sequencer- og synthesizerparadigmet, nemlig muligheden for hændelsesstyring af musikken.

HÆNDELSESBASERET STYRING: ALGORITMISK MUSIK

Csound og MaxMSP åbner mulighed for at organisere musik på andre præmisser end de rent tidsbaserede. Fra at være *tidsstyret*, som i nodebilledet og i sequenceren, opstår muligheden for at musikken kan være *hændelsesstyret*. Det betyder, at starten på en

tone ikke nødvendigvis skal angives til et bestemt tidspunkt, men kan afhænge af, om en bestemt betingelse er opfyldt. Det er en helt afgørende forskel, der åbner nye kompositoriske og æstetiske muligheder.

I modsætning til sekvensen som jo er sequencerens grundliggende datastruktur i den forstand, at indholdet på forhånd anbringes i fastlagte rækkefølger, som programmet herefter afvikler sekventielt, gør programmer som Csound og MaxMSP brug af andre grundliggende computerbaserede datastrukturer som selektion og iteration. Herved drager Csound og MaxMSP fordel af computerens algoritmiske natur i en helt anden forstand, end audiosequenceren gør det.

Selektion betyder, at programafviklingen kan følge flere veje. Selektionen åbner for forgrening og dermed for forskellige udfald. I MaxMSP-eksemplet bestemte vi, at alle toner under C3 skulle transponeres en stor terts op og alle toner under C3 en oktav ned. Vi foretog en udvælgelse ved at spørge til størrelsen på den indkomne tone. Sagt på en anden måde skabte vi en selektiv algoritme af typen: ”**Hvis betingelse så udfør**”. En selektionsalgoritme kan udvides ved at pege på, hvad der skal ske, hvis betingelsen ikke er opfyldt. I det tilfælde angiver algoritmen hvilken af to alternative valgmuligheder, der skal gælde. Algoritmen får herved følgende udseende: ”**Hvis betingelse så udfør 1 ellers udfør 2**” som i vores eksempel.

Iteration er en anden type af datastruktur, som Csound og MaxMSP giver mulighed for at arbejde med. En iterativ struktur er en struktur, der gør det muligt at beskrive en proces af ukendt varighed i en algoritme af endelig længde. I MaxMSP-eksemplet transponerer vi en stor terts op eller en oktav ned alt efter størrelsen på den indkomne tone, indtil vi har modtaget i alt 20 toner. Hvor længe der går, før vi har modtaget 20 toner, kan vi ikke vide noget om, men da vi har betinget, at vi maksimalt vil

modtage 20 toner, ved vi, at processen på et tidspunkt vil stoppe. I stiliseret form har den iterative algoritme følgende udseende: **Gentag** ("**Hvis betingelse så udfør 1 ellers udfør 2**") **Indtil betingelse**".

I modsætning til et loop, der som sagt er karakteristisk for den sequencerbaserede musik, er iteration en type af gentagelse, hvor antallet af gentagelser ikke er kendt på forhånd, men afhænger af, om en given betingelse er opfyldt eller ej. Hvor loopet er en bekvem udnyttelse af sequencerens automatik, der fratager os arbejdet med at folde strukturen ud i sin fulde form, er iterationen en gentagelse, der afhænger af en beregning. I modsætning til loopet kan vi ikke selv folde den samlede struktur ud uden at foretage beregningen og dermed gennemløbe algoritmen. I modsætning til sequencerens loop er den iterative algoritme således en udnyttelse af computerens generative potentiale.

Forskellen mellem den algoritmiske og den sequencerbaserede måde at komponere på er, at mens man i sequenceren organiserer et indhold i tid, som man herefter lader computeren afvikle, angiver man i den algoritmiske musik trin for trin de betingelser, som ligger til grund for musikkens udfald og den arbejdsproces, som computeren skal udføre for at realisere musikken. Algoritmen bliver herved en opskrift, som computeren følger, og som den lægger til grund for musikkens måde at folde sig ud på. Algoritmen kan opstilles på forhånd som i Csound eller i realtid modificeres som i MaxMSP eller SuperCollider. I begge tilfælde bliver algoritmen det "håndtag", som komponisten hiver i, hvis han vil ændre den musik, der følger af algoritmen. Hvor sequenceren lægger op til, at man ændrer i rækkefølgen af de elementer, man på forhånd har lagt til rette, er det de regler og betingelser, der ligger til grund for algoritmen, som man udtrykker sig igennem i MaxMSP og Csound. Hvor den sequencerbaserede

musik er tæt knyttet til montageformen, der indebærer, at man lægger et musikalsk materiale til rette, er den algoritmiske musik baseret på generative processer, hvorefter der følger et musikalsk udtryk.

ÆSTETISKE UDTRYK

Som sagt er Csound og MaxMSP ikke repræsentanter for en ny type af musiksoftware. De er senere generationer af programmer, der fra og med 1960'erne blev brugt til komposition af musik, og som var i brug længe inden de første softwarebaserede audio-sequencere kom frem. Algoritmisk komposition er således ikke af ny dato og har længe været en disciplin inden for den klassiske computermusik. I dag har den algoritmiske musik fået et bredere kommercielt og populærmusikalsk gennembrud inden for techno og ambientmusikken og i særdeleshed i den musikform, der går under navnet "live-coding". Der foregår altså en opblomstring inden for musikmiljøet selv, der naturligvis har at gøre med den generelle samfundsudvikling, hvor computeren fra og med slutningen af 1970'erne glider ud i kulturen og bliver et medie, som stadigt flere bliver fortrolige med og møder såvel på arbejdspladsen som i fritids- og kulturlivet.

En anden faktor der kan være en medvirkende årsag til, at den algoritmiske musik får et bredere gennembrud, er computerspillet. Computerspillet er et medie, der qua sin interaktive og ikke-lineære natur umuliggør en tidlig strukturering af musikken. Her er afviklingen af såvel musikken som spillets øvrige elementer per definition hændelsesstyret. I kraft af computerspillets store udbredelse og kulturelle betydning får vi et medie, der gør en ny generation af musikere og komponister fortrolige med den algoritmiske musik. M.a.o. kan man hævde, at computerspillet i et kulturelt perspektiv er nutidens interface til computeren som algoritmisk maskine.

LIVE CODING

Live coding er en forholdsvis ny performansbaseret musikform. Den er karakteriseret ved, at musikerne på scenen er udstyret med laptop-computere, hvorfra de programmerer musikken i realtid. Ved brug af software som f.eks. MaxMSP, Csound og Super-Collider opstiller og/eller modificerer de under koncerten den programkode, som genererer musikken. Som et scenisk element projiceres musikernes skærbilleder ofte på bagvæggen.



Figur 8: Live-coding session, Read_Me 2004, Århus.

Når musikere vælger at arbejde med algoritmisk komposition og tillige bringer det ind i en performans/koncertsammenhæng, så skyldes det, at de kan kontrollere musikken på niveauer, hvor audiosequenceren ikke kan være med og tillige lægge musikken til rette på andre måder end den sekventielle og tidsstyrede struktur, som sequenceren kan tilbyde. I artiklen "Live Coding in laptop performance" skriver forfatterne, at deres interesse for at program-

mere computeren selv ligger i muligheden for at afskrælle de metaforiske lag, der kendetegner audiosequencerens grafiske interface og derved komme i kontakt med computerens særegne potentialer¹⁶. Disse potentialer kommer til udtryk i computeren som en generativ mere end en redigerbar maskine.

Programmeringssprog som Csound og MaxMSP gør det muligt at opbygge og definere musikken fra bunden af. Både den enkelte lyd i sig selv og den måde som de enkelte lyde skal håndteres og organiseres. Komponisterne udvikler i en og samme proces, såvel det instrument der frembringer musikken som det musikalske indhold, der følger af instrumentet. De er i gammeldags forstand både instrumentbyggere, musikere og komponister, og det er de igennem at forholde sig til en og samme maskine. Ved at programmere musikken orienterer musikerne sig både mod maskinen som et værktøj, de opbygger, afprøver og modificerer, samt mod det musikalske resultat der følger af deres evne til at udvikle og kontrollere værktøjet. Sagt på en anden måde samler deres interesse sig ikke om det musikalske udtryk isoleret set, for softwaren er ikke i simpel forstand et middel til opnåelse af et mål. Udviklingen og kontrollen af softwaren og dermed beherskelsen af maskinen er i sig selv et afgørende element. Softwaren er den genstand som interessen samler sig om og selve programmeringen det ekspressive udtryk.

KONKLUSION

Afslutningsvis vil jeg pege på, at sequencerparadigmet trives i programmer, der udnytter computerinterfacets visuelle potentiale, mens synthesizerparadigmet trives i interfaces, der er tættere på computerens algoritmiske og skriftlige natur og dens egenskab af generativ maskine. Hvor musikken i sequenceren fremstår som

¹⁶ Collins 2003, p. 322.

sekventielle ordninger af singulære objekter og alene giver mulighed for at fastlægge musikken i tidsdomænet, er musikken i synthesizerparadigmet at forstå som et signal, der flyder mellem selvstændige moduler, der på forskellig vis modificerer signalet. Udnyttelsen af computerens algoritmiske og generative natur betyder endvidere, at der arbejdes med hændelsesbaserede strukturer, der nok udfolder sig i tid, men som ikke er tidsstyrede.

Den sequencerbaserede musik trækker i vid ustrækning på æstetiske praksisser fra tidligere medieteknologier som nodebilledet og den analoge båndoptager foruden selvfølgelig de mekaniske principper, som indføres allerede med klokkespillet. De æstetiske praksisser er knyttet til montage og collage og det er de i kraft af at udnytte computerens visuelle interface til redigering og organisering af musikken.

Den algoritmiske musik synes derimod i højere grad at udnytte de potentialer, der er særegne for computermediet. For selvom interfaces som Csound og specielt MaxMSP også rummer grafiske elementer såsom slidere, knapper og grafer, så er den grundlæggende tilgang baseret på abstrakte symboler (f.eks. tal) og formaliserede beskrivelser i form af datastrukturer som "if, then, else" sætninger, og det er baggrunden for at udnytte computerens generative potentiale. Det er dog en tilgang, der kan ses i forlængelse af nodebilledets metriske, diagrammatiske og litterære dimension, da de betingede datastrukturer netop giver adgang til at arbejde med kausalt begrundede og subordinerede strukturer, endskønt det foregår på en anden måde og denne gang også uden for et tidsdomæne.

Til trods for at computeren har været her mere end et halvt århundrede og siden 1950'erne har været brugt til komposition af musik, så er det dog først de senere år, at den som algoritmisk maskine synes at have fundet større udbredelse i musikalske kredse. En årsag kunne være computerspillet, der, som nutidens

populæreste kulturelle underholdningsform, for alvor har åbnet vores øjne for den algoritmisk genererede musik, og som tillige har anvist os en æstetisk ramme at tænke den ind i.

LITTERATURLISTE

Barnholdt Hansen, Jette (2002). *Ill Canto sulla lira – overhørte klange fra et mundtligt univers*, in *Cæcilia*, (Musikvidenskabeligt Institut Århus Universitet), p. 111-148.

Bolter, Jay David & Grusin, Richard (2002). *Remediation: understanding new media* (Cambridge, Mass. London: MIT Press).

Brincker, Jens (1974). *Musiklære og Musikalsk Analyse*, V del: Form, (København: Engstrøm & Sødring).

Collins Nick (2003). *Live Coding in laptop performance*, in *Organised Sound* (UK: Cambridge University Press)

Dodge, Charles & Jerse Thomas A. (1985). *Computer Music – synthesis, composition and performance* (New York: Schirmer Books).

Gerzso Andrew (1992). *Paradigmes and Computer Music*, in *Leonardo Music Journal* (Oxford: Pergamon Press)

Johansen, Jørgen D. & Larsen, Svend E (1994). *Tegn i brug* (Danmark: Amanda)

Ludovico, Alessandro (2004). *The Sequencer Paradigm, Read_Me* Software Art & Culture (Århus: Aarhus University Press)

Manovich, Lev (2001). *The Language of New Media*, (Cambridge Mass.: MIT Press).

May, Michael. & Stjernfelt, Frederik (1996). ”måling, diagram, kunst – overvejelser over ikonets rolle i videnskab og æstetik”, in *Billeder fra det fjerne*, (red.) Anders Michelsen og Frederik Stjernfelt (Oslo: Akademisk Forlag).

Ong, Walter J. (1982). *Orality and Literacy – the Technologizing of the Word* (London: Routledge).

Roads Curtis (1996). *Computer Music Tutorial* (Massachusetts: MIT Press).

Standley, Sadie (Ed.) (2000). *The New Grove Dictionary of Music and Musicians* (Groves Dictionaries Inc).

Stjernfelt, Frederik (1999). "Den levende Kristus og det forsvindende kvadrat", in *KUNSTTEORI – positioner i nutidig kunstdebat*. Red. Hans Dam Christensen et. al. (Valby: Borgen).

Zuckerlandl, Victor (1956). *Sound and Symbol*, (London: Routledge & Kegan Paul).