

Design af software-instrumenter – en sang om diffuse designprocesser og deres refleksive potentiale i brug

Dette speciale er skrevet som afslutning på vores uddannelse i Multimedier ved Humanistisk Informatik, Institut for Kommunikation, Aalborg Universitet, 2005.

Vi vil gerne takke vores vejleder Bodil Kirstine Jensen, vores mentor Jens F. Jensen, Ilpo Koskinen, vores testere og empiriske kilder for deres medvirken, hjælp og inspiration. Og ikke mindst en stor tak til Lars og Bente for at lægge hus til vores specialelejr.

Specialets omfang er 328.010 enheder, svarende til 137 normalsider

Mads Lindgren

Mads Weitling

Resumé

Designing Software Instruments: Fluctuant Design Processes and their Reflective Potential in Use

Keywords: Music interfaces, Design, Culture, Aesthetics, Interaction, Prototyping

The thesis aims to establish a methodology for developing software prototypes within a community of experimenting musicians. The initial focus is on how information is collected and implemented through iterations in a functioning prototype. In addition to processing empiric and theoretical data, the thesis describes the construction of three prototypes which are developed in the process. The scope of the prototypes is to investigate modification of sonic material through visual and tactile interfaces. Blender and Python are the primary technologies behind the interfaces, used to ensure the synchronisation of audio and image. The prototypes are evaluated in the context of active musicians, regarding to which degree it makes sense for a musician to incorporate the design concepts in a given workflow or situation. The thesis explores topics on how online and situational test information is used to generate new iterations of music interfaces. Companies that produce music software deal with a fluctuant consumer group which often proves to be hard to analyse. The thesis investigates corporate possibilities to adapt analysis models for fluctuant tendency groups such as musicians.

Indhold

RESUMÉ	2
INDHOLD	3
INDLEDNING	7
BAGGRUNDEN FOR SPECIALET	7
SPECIALETS OPBYGNING	8
<i>Læsevejledning</i>	8
<i>Tekstens struktur</i>	8
PROBLEM.....	9
<i>Det metodiske problem</i>	9
<i>Det designmæssige problem</i>	12
METODE	14
INFORMATION.....	14
ITERATIONER.....	16
ITERATION, INFORMATION OG DESIGNSYSTEM.....	19
INNOVATION OG ADAPTION.....	20
<i>Behov for innovation og adaption</i>	21
<i>Innovation/adaption - tid, mål og metodik</i>	21
<i>Innovationssfærer</i>	22
PRIMING, KREATIVITET, INSPIRATION OG INFORMATION.....	24
<i>Framework: Kreativitet som kombinations- og forestillingsproces</i>	24
<i>Kreativ fiksering</i>	25
<i>To slags kreativitet</i>	25
OPSAMLING	26
FORMULERING AF DESIGNMETODEN	27
DE KONKRETE AKTIVITETER I DE FIRE ARBEJDSFASER	35
<i>Første fase</i>	35
<i>Anden fase</i>	37
<i>Tredje fase</i>	40
<i>Fjerde fase</i>	40
STRINGENTE OG IKKE-STRINGENTE DESIGNPROCESSER.....	41
<i>Navigation i gradueringen af stringens</i>	41
<i>Et rationale for designmetoden</i>	43
<i>Nuancering, tilnærmelse og kombinatorik</i>	43
DESIGN	45
SOFTWAREINSTRUMENTER.....	45

EMPIRI – METODE.....	46
<i>Overordnede temaer</i>	47
<i>Metoden bag empiriindsamlingen</i>	49
<i>Kvalitative metoder</i>	50
<i>Kvantitative og hybridmetoder</i>	51
EMPIRI – DOMÆNEINDKREDSNING	51
<i>Interviews, Berlin og København</i>	51
<i>Interviews, Jack Dark onlineforum</i>	58
SYSTEMFILOSOFI	60
<i>Æstetik og funktionalitet – værktøj, oplevelse og forstyrrelse</i>	60
<i>Gensidige forstyrrelser mellem omverden og designsystem</i>	60
ÆSTETIKKEN I OG OMKRING PROTOTYPEN	61
<i>Æstetikens vilkår</i>	61
<i>Nogle grundlæggende definitioner</i>	62
<i>Intern og ekstern æstetik</i>	63
<i>Systemintern æstetik</i>	63
<i>Æstetisk systemfilosofi</i>	64
<i>Metaforik, rumlighed og naturlige egenskaber</i>	65
<i>Interaktionsdesign</i>	66
<i>Instrumentdesign</i>	67
<i>Systemekstern æstetik</i>	69
<i>Funktion og form – en sang om struktur</i>	71
<i>Kunst – industri eller manifest</i>	73
FUNKTIONALITET OG MODIFICERING	76
<i>Residualberegninger</i>	77
FUZZY LOGIC.....	79
<i>Phrasetrækker som eksempel på brugen af fuzzy logic</i>	83
FRAMEWORKS OG DIALEKTIKKER – INDKREDSNING AF ARBEJDSPROCESSER	83
<i>Prækomposition – komposition – performance</i>	84
<i>Makroniveau og mikroniveau</i>	86
<i>Forventning – overraskelse</i>	86
<i>Frameworket og dets grundlag</i>	87
SCENARIER – SKITSERING AF INTERAKTIONSFORMER	88
USE CASES – INTERAKTIONSFORMERNES FORANKRING I FUNKTIONALITET	89
<i>Positive use cases</i>	90
<i>Abuse cases – negative use cases</i>	91
PROTOTYPE	94
ARKITEKTUR	94

<i>Prototype – iterativ udvikling af arkitektur</i>	94
<i>Arkitekturen</i>	94
KONSTRUKTION.....	96
<i>Teknologi</i>	96
<i>Blender</i>	97
<i>Python</i>	98
<i>MIDI</i>	99
FORMMÆSSIG FIKSERING AF PROTOTYPEN	100
EMPIRISK FIKSERING AF INTERAKTIONEN	100
TEORETISK FIKSERING AF INTERAKTIONEN	103
<i>Innovation - Adaption</i>	103
<i>Fuzzy Logic</i>	103
<i>Opsamling</i>	105
TEKNOLOGISK FIKSERING AF INTERAKTIONEN	105
<i>Venstreklik</i>	106
<i>Højreklik</i>	107
<i>Scrollhjul</i>	107
<i>Tastatur</i>	107
<i>Softwaren</i>	108
EMPIRISK FIKSERING AF ÆSTETIKKEN.....	109
TEORETISK FIKSERING AF ÆSTETIKKEN	113
TEKNOLOGISK FIKSERING AF ÆSTETIKKEN.....	113
PROCESBESKRIVELSE	114
<i>Prototypeproces: Improvisation inden for et framework</i>	114
<i>Tidslighed i udviklingen</i>	114
<i>Fravalg og konsekvenser</i>	115
EVALUERING.....	116
TEST AF PROTOTYPEN	116
TESTPERSONERNE	116
TESTFORLØBET.....	118
<i>Første halvdel af forløbet: Containerturm</i>	118
<i>Første halvdel af forløbet: Phrasetrækker</i>	119
<i>Anden halvdel af forløbet: Test med Artillery</i>	120
MODSVAR TIL BEHOV OG IDEER TIL FORMÆNDRINGER	121
UDVIDELSE AF DESIGNET – NYE PERSPEKTIVER	122
EVALUERING AF PROTOTYPEN OG TESTEN.....	124
<i>Designets relevans som helhed</i>	125
<i>Prototypen som repræsentation af et design</i>	126

<i>Muligheder med relevans for redesign</i>	128
EVALUERING AF METODEN	130
<i>Metode versus proces</i>	130
<i>Informationsbehandling</i>	132
<i>Metoden i et organisatorisk perspektiv</i>	133
<i>Revidering af metoden</i>	135
KONKLUSION	137
LITTERATURLISTE	140
ANSVARSLISTE	144
BILAG	145
ORDLISTE	145
TEST - INTERVIEWSVAR	148
<i>Bo Nicolaisen</i>	148
<i>Sune Petersen</i>	150
<i>Rasmus Allin</i>	150
TEST – NOTER TAGET UNDERVEJS.....	152
<i>Bo Nicolaisen</i>	152
<i>Sune Petersen</i>	152
<i>Rasmus Allin</i>	153
<i>Diskussion af fremtidige perspektiver</i>	153
RESPONS PÅ ONLINEFORA OG MAIL.....	154
<i>JackDark – Happy little hellhole [http://jackdark.proboards39.com/]</i>	154
<i>BuzzChurch [http://www.buzzchurch.com]</i>	160

Indledning

Baggrunden for specialet

Vi står ved vores uddannelses foreløbige afslutning, og ser det som vores mål at vores speciale kan bidrage med nye perspektiver på udvikling indenfor humanistisk informatik. Samtidig fungerer specialet som en endelig sammenfatning af de erfaringer og tanker, vi har gjort os undervejs i vores uddannelsesforløb.

Et umiddelbart perspektiv er vores eftersøgning af de muligheder for *konstruktion*, som empiriske undersøgelser, analyser og kritik må åbne op for i design-sammenhæng. Ofte bliver analyse eller kritik et mål i sig selv indenfor vores fagområde (humaniora), fremfor at tjene som kilde til nye designs, innovationer og håndfaste konstruktioner (vores fortolkning af informatik). Med vores speciale ønsker vi at sætte informationsindsamlingen ind som grundlag for en konkret design- og konstruktionsproces, fremfor at information anvendes kritisk imod designs.

Samtidig forsøger vi at etablere et grundlag for en sammenkædning af kreativitet og systematik i udviklingen af multimediesoftware. Dette grundlag ser vi i *refleksion*, forstået som en selvrefleksiv tankepraksis, der foregår gennem hele designprocessen, og som omfatter såvel design som metode. Refleksionen som tankeform åbner op for en løbende reformulering af vores selvforståelse mht. design og metode, og tjener således som kilde til nye erfaringer og tanker.

Det er vores mål at afprøve hvorledes refleksion og konstruktion kan kombineres i designprocesser. Med andre ord er vi interesserede i at gøre designprocessen til en central, vidensbaseret proces bag formgivningen af software. Viden forstået både som information indsamlet ude i verden, men også som refleksion hos os som software-designere.

At vi ser os som software-designere i multimedie-sammenhæng betyder at vi ønsker at kombinere arbejdsområder som f.eks. æstetisk formgivning og funktionalitetsdesign, konkret som brugerfladedesign og programmering. Systemdesign og kortlægning af brugeres arbejdsformer er en anden kombination, vi finder givende i forhold til designets finitte form. Dette bygger på tidligere universitetsprojekter, hvor vi har tillagt os et lignende perspektiv på konstruktionen af software som en proces, der kræver tværfaglighed og evnen til at danne en sammenhæng mellem mange sondringer og konstruktionsprocesser.

Samtidig krydser vi en adskillelse mellem hhv. software-designere og musikere, idet vi selv virker som musikere, og herfra henter mange forventninger, inspiration og ideer. Vi har udenfor

universitetsstudierne en forankring i den elektroniske musikkultur i Danmark, og for Mads Lindgrens vedkommende også Berlin. For os begge handler det om et længere virke, der bl.a. har resulteret i flere udgivelser og en lang række koncerter, og som bygger på lang erfaring med eksisterende musiksoftware og –hardware i stort set alle mulige brugssammenhænge. Denne baggrund afspejler sig både i de muligheder for empiri, som vi har afsøgt i specialet, men også i de mange sondringer for design, vi gør gennem designprocessen. At være en del af denne musikkultur betyder kontakt med musikere med lignende interesser, hvoraf en del andre også arbejder med eget software-design, og hvor langt de fleste har en holdning til, hvordan musiksoftware kan forbedres. Holdninger, der bygger på praktisk erfaringer med designs, bliver dermed en platform for vores udvikling.

Forholdet til musikkulturen er således ikke blot gennem observation, men som deltagelse – ikke blot den lokale musikerkultur, men også internationale samhørigheder med designere gennem online-fora. Derfor kommer denne baggrund ikke altid direkte til udtryk, men præger specialets muligheder, perspektiv og selve skriveprocessen. Sideløbende med specialeskrivningen har vi ikke blot opereret som studerende, men også spillet koncerter, øvet med andre og eftersøgt mulighederne i andres software, hvilket uvægerligt åbner op for inspiration og udfordringer for vores eget design, og løbende giver os nye ideer. I udviklingsfasen har vi anvendt vores tilstedeværelse i kulturen som en stor ressource for at implementere så meget relevant information i designet som muligt.

Specialets opbygning

Læsevejledning

Allerførst foreslår vi, at læseren tager et kig på den vedlagte cd-rom eller hjemmesiden for projektet (<http://www.bitreactor.dk/>). Her findes det endelige design i en form, der kan installeres på en Windows-computer, og videoer, der viser designet i aktion. Allerede i næste afsnit forudsætter vi at læseren er bekendt med designets udseende.

Termer, som stammer fra vores virke som musikere, eller teoretiske termer, der tages ind undervejs, er forklaret i en ordliste, der findes i *Bilag*.

Tekstens struktur

Specialet beskriver en designproces og dennes resultat, og kan læses som en eksplicitering af den informationsindsamling, konstruktion og refleksion, der ligger bag såvel design som metode.

Selve teksten må derfor ses som en kortlægning af en proces, hvorved nogle ting tages ud af en kronologisk rækkefølge og i stedet underlægges den emnemæssige sammenhæng, som vores

disposition afspejler. Dispositionen bygger i stedet på en overordnet tilnærmelse af et design og en metode med relevans for os. Den følger derfor en opbygning over disse hovedpunkter:

- Formulering af et metodisk og et designmæssigt problem
- Beskrivelse af det metodiske grundlag såvel som de konkrete aktiviteter, der indgår i metoden
- Beskrivelse af designet, herunder informationsindsamling, æstetik, funktionalitet og en kobling af teoriens og empiriens udsagn til designrelevant information.
- Konstruktion af en prototype, der udtrykker ideerne bag designet.
- Evaluering af designet ud fra feedback indsamlet gennem afprøvning. Og evaluering af metoden ud fra refleksion over dennes anvendelighed indenfor vores arbejdsområde.

Undervejs vil vi samtidig knytte an til et refleksivt niveau, der handler om forudsætningerne for vores metode, design og evaluering, og således indgår som et konstruktivt selviagttagende bidrag til såvel metode som design. Denne dobbelthed bestående af konstruktion (af objekter, ideer, etc.) og iagttagelse (af forudsætninger, konsekvenser, etc.) indebærer at vi løbende etablerer en sammenhæng mellem formgivning og iagttagelse af denne form.

Det første anslag til såvel en metodisk som en designmæssig konstruktion er emnet for næste afsnit, hvor vi vil beskrive vores problem og opridse en terminologi for specialet.

Problem

Det metodiske problem

Vi formulerer det metodiske problem i vores speciale ud fra iagttagelse af nogle væsentlige vekselvirkninger indenfor designprocesser generelt og design af multimedie-software specifikt.

Den første vekselvirkning angår forholdet mellem to polariseringer, vi definerer som følgende: Design, konstruktion og test på den ene side, og metodik, refleksion og selviagttagelse på den anden. Et speciale er også et forsøg på at bidrage med ny viden indenfor vores konkrete fagområde, hvilket umiddelbart fordrer en formidling af de iagttagelser og teoretiseringer vi gør os. Specialet søger i nogen grad at forandre forholdet mellem de nævnte polariseringer og foreslå en alternativ metodisk tilgang til design og designprocesser, hvilket vi ser som et metodisk bidrag til fænomenet softwarekonstruktion. Med vores praktikophold in mente har vi fundet et konkret behov for at omskrive standardiserede udviklingsmetoder. For det første for at applicere metoder til

softwareudvikling i domænet musiksoftware, og for det andet for at udvide teoretiske antagelser og andre bevæggrunde for metodisk problemløsning. Målet for at igangsætte denne forandring på området er altså en integration af metode og konstruktion mhp. en dybere refleksion. Vekselvirkningen mellem refleksion og konstruktion ser vi som basis for designet, såvel som et refleksionsgrundlag for en revidering af specialets metode. Vi savner en sådan integration af det konstruktive og det refleksive moment i udviklingsprocesser, såvel som formulering af alternative metoder. Vi vil derfor vige udenom ren og skær teoribaseret metodeformulering såvel som rendyrket konstruktion uden et refleksivt indhold. Vi finder et potentiale i at undgå radikaliserings og i stedet arbejde kombinatorisk mellem disse ekstremer. Design og refleksion hænger hermed nøje sammen i en videnskabsende proces, hvor konstruktion afføder refleksion, der katalyserer en ny konstruktion (f.eks. viden eller nye objekter). En af specialets primære antagelser er, at design og refleksion sammenholdes af information. Information, der løbende implementeres i specialets prototype, såvel som specialets tekst. Projektet søger at etablere og afprøve en designproces ud fra denne præmis: nemlig at analytisk, teoretisk og metodisk refleksion går hånd i hånd med praktisk design og konstruktion. Heri er en gensidig afhængighed ekspliciteret, idet vores speciale gælder designprocessen og i særdeleshed dens vekslen mellem analyse og syntese. Afhængighedsforholdet er ekspliciteret gennem information, som etablerer en sammenhæng mellem specialets enheder. Udvælgelsen af relevant information sker gennem refleksion over f.eks. designkriterier og i henhold til specialets hovedantagelser.

Derfor er specialet opbygget omkring to overordnede tråde, nemlig hhv. en metodisk/refleksiv og en syntetiserende/konstruktiv tråd. Vi foranstalter en designproces, ikke blot for at blive klogere på dennes objekt (nye former for interaktion og kontrol med digital lyd), men også for at blive klogere på processens indre dynamikker og modsætninger. Med andre ord etableres designprocessen som en *case* for en refleksiv iagttagelse af designprocesser i udviklingsøjemed.

En anden vekselvirkning angår det metodiske forhold mellem paradigmerne *brugercentreret* og *designercentreret* design. Vekselvirkningen mellem disse traditioner kan kort beskrives ud fra principper, der er velkendte i områder som test- eller valideringsmetoder i softwaredesign. Brugeren bliver set som en kvalitetssikring i den traditionelle opfattelse af brugercentreret design. Brugercenreret design definerer vi som en proces, der inddrager brugeren som den egentlige beslutningstager vedrørende designets form. Dette sker f.eks. gennem fokusgruppe-undersøgelser, *participatory design*¹, eller mere eksperimenterende metoder som *cultural probing*² eller empirisk vidensindsamling i online-fora. Designercentreret design indsætter derimod designeren som den

¹ Se bl.a. Bødker, Kensing, Simonsen, 2000

afgørende beslutningstager for processen i at forvandle ide til form, eksempelvis vha. ekspertevalueringer, designmetoder ud fra teoretiske anvisninger, eller viden om materialer. Denne designform er ofte kritiseret for ikke i tilstrækkelig grad at repræsentere og imødekomme brugerbehov, mens brugercentreret design på den anden side kritiseres for at overlade for mange designbeslutninger til personer, der ikke besidder egentlige designrelevante kompetencer indenfor f.eks. teknologi og æstetik. Hermed skitserer vi et modsætningsforhold mellem paradigmerne, der kan opløses eller i det mindste imødekommes ved en anderledes metodisk tilgang. En metode, der tilgodeser såvel brugerinddragelse som anvendelsen af de kompetencer, der findes i designgruppen. Metodens hovedformål er at generere så stor en mængde designrelevant information som muligt under omstændighederne, da begge paradigmer søger at varetage betydningstunge elementer i en designproces. Vigtige elementer, der overses, hvis enten brugercentreret eller designercentreret design isoleret anvendes som det eneste udgangspunkt for processen. I metoden ligger også tidlige overvejelser, i form af et ønske om at bevare muligheden for en løbende indarbejdelse af ideer og kreativitet i designprocessen. Dvs. 'forstyrrelser' som en overskridelse af et sæt forventninger, der stammer fra designgruppen eller fra designgruppens tolkninger af uventede brugerinputs indsamlet i f.eks. et forum, en fokusgruppe eller testsituation.

At overkomme modsætningen mellem disse to tilgange til design indebærer for vores vedkommende også en markering af vores dobbelte tilhørsforhold. Vi arbejder hhv. som en konstruktiv/refleksiv designgruppe (os som forfattere af dette speciale), og en vagere defineret gruppe bestående af 'musikere, der anvender computere'. Vi står med en fod i begge lejre, og også af denne grund bliver det uholdbart at tale om en rent bruger- eller designercentreret designproces for vores vedkommende. I stedet vil vi trække på de styrker, dette forhold giver os i form af kendskab til brugerbehov, teknologiske muligheder og kulturen omkring musiksoftware. Vi ser et potentiale i at forstærke vores metodiske/teoretiske udviklingsapparat med denne viden. Med andre ord er vi ikke ude på at etablere en 'glasvæg' mellem empiri og design, eller mellem os selv og de brugere, vi inddrager undervejs. I stedet ønsker vi at etablere en designproces, der i vidt omfang er åben for alle inkluderede parter, og de forstyrrelser og uventede ideer, der opstår når divergerende forventninger mødes og analyseres. Den refleksionsmæssige udfordring bliver dermed at holde rede på sådanne forstyrrelses placering i designprocessen som enten relevant eller irrelevant information, herunder hvordan og i hvilke faser, vi aktivt åbner processen for sådanne forstyrrelser.

Denne sidste vekselvirkning – mellem en lukket og en åben designproces – angår imidlertid ikke kun forholdet mellem designere og brugere, men også forholdet mellem designere og design. Det

² Gaver, m.fl., 1999

drejer sig med andre ord om at etablere en åbenhed for forstyrrelser fra materialet og konceptets side. F.eks. når ny teknologi anerkendes som en mulighed for designet, eller når komponenter ikke fungerer som forventet, og ikke varetager den funktion, som grundarkitekturen har defineret. Forstyrrelser kan også være et potentiale, eksempelvis når nye koncepter og metaforer viser sig omsættelige via de komponenter, vi endelig kan anvende. I vores tilfælde håber vi at tilvejebringe overordnede forstyrrelser af designet, når vi lader to hidtil forholdsvis adskilte teknologier, 3D *game engines* og musiksoftware, mødes. I denne kombination håber vi også at forstyrre den etablerede semantik eller metaforik i hhv. musiksoftware og *game engines*. Dette gør vi for at muliggøre en forståelig og interessant kobling af et visuelt, et interaktionsmæssigt og et auditivt domæne.

Hermed nærmer vi os det designmæssige problem, som vi vil behandle nedenfor.

Det designmæssige problem

Sondringerne i det metodiske problem leder os mod konkretiseringer. Her stiller vi os selv udfordringer i at skulle etablere en form, der skal indeholde mange attributter og egenskaber. For det første skal formen være interaktiv og rumme funktionalitet, og repræsenteres i overensstemmelse med æstetiske standpunkter. Allerede her er dele af et begrebsunivers anslået; interaktion, æstetik, repræsentation, funktionalitet mv. Vi anser interfacet for en opsummering af empiriske, metodiske og teoretiske analyser eller antagelser. En form for delkonklusioner, der i vores iterative proces har givet materialet en form, og resultatet i at idemasse gradvist transformeres til finit form. Vægtningen af specialeafsnittene viser også en vis tyngde i design og metode, der er en dybdegående diskussion af påvirkningsfaktorerne af den finite form. Forud for de mere uddybende diskussioner findes der også en del antagelser, der kronologisk ligger som det allerførste led i specialet. Her er spørgsmålene: hvilke muligheder findes i lydsoftware, og hvad kan vi etablere herudfra, der ikke findes. Derefter begynder de metodiske problemstillinger at tage form som nævnt i forrige afsnit. Indledende har vi foretaget en sondring af musiksoftware i sin helhed, ud fra egne brugsmønstre, behov og ambitioner. Finder vi et korrelat til disse behov i en given empiri/teori, anser vi det for et 'reproduceret' eller 'verificeret' behov. Det indledende kriterie, der senere er blevet reproduceret, har for os været *en ny interaktionsform*. Umiddelbart fordi vi har anset det som teknologisk muligt i forhold til tidsrammen for specialet. Videre fordi vi anser *en ny interaktionsform* som et æstetisk formsprog, der kan udvides med nye former for kontrolflader. Med andre ord mangler der måder at etablere nye arbejdsprocesser på, set i forhold til mængden af musiksoftware, der findes, hvilket er det første reelle designmål. Målet er at skabe et indtryk af hvad en kontrolflade *kan være*, i forhold til hvad en kontrolflade *er*.

Med denne indkredsning af det metodiske og det designmæssige problem, vil vi nu se på etableringen af en konkret metode.

Metode

Formuleringen af en designmetode, som vores designproces kan struktureres efter, tager udgangspunkt i en række teoretiske afsæt, som vi i det følgende vil behandle i mindre afsnit. Det drejer sig om begreberne *information*, *iteration*, *designsystem*, *innovation* og *adaptation*. *Priming og kreativitet* vil beskrive en binding af information og iteration til kreativitet, hvilket bl.a. sker i inspirationsprocesser. Dette sker for at komme nærmere ind på 'den gode ides' udspring og udvikling, som gives en metodisk basis i en efterfølgende formulering af metodens konkretisering.

Her vil vi ud fra disse begrebsafklaringer og yderligere afsæt i teorier med specifik relevans indenfor systemudvikling formulere den metode, der strukturerer vores designproces. Opbygningen af et metodisk apparat trækker som nævnt i *Problem*-afsnittet på såvel generelle begrebsafklaringer som referencer til eksisterende bud på systemudvikling, herunder brugen af *scenarier*, *use cases* og *prototyper*. Metodens indhold konkretiserer vi derefter yderligere ud fra de fire faser, der indgår i den som fire primære arbejdspakker, der er karakteriseret ved særegne aktiviteter.

Formuleringen af en metode til systemudvikling indenfor et kreativt domæne som musiksoftware, giver os endvidere anledning til overvejelser om forholdet mellem stringente og ikke-stringente metoder. Vi finder det nødvendigt at opløse stringensen i forhold til at kunne bearbejde informationsimplementeringen i designet. Informationen har bl.a. 'musiker-programmør' som kilde, hvilket er et diffust område.

Et område, hvor arbejdsprocesserne er svære at beskrive entydigt, hvilket stringensen også må justeres i forhold til. På den anden side af informationsindsamlingen kræves en større stringens, når data om arbejdsprocesser, setups, osv. behandles i forhold til gennemgående tematikker i specialet. Denne vekslen mellem stringens og ikke-stringens kræver en behandling af metodens perspektiv, der sammenfattes i en behandling af metodens præmisser i afsnittene *Et rationale for designmetoden* og *Nuancering, tilnærmelse og kombination*.

Information

Vores informationsbegreb er grundstenen i vores metodiske og teoretiske konstruktion. Vi anser det derfor som et allestedsnærværende kernebegreb, og informationen som et element, hvis form og funktion kan levere en forklaring på størstedelen af processerne i specialet. Begrebet er inspireret af

George Spencer Browns informationsbegreb³, og senere formidlere af samme tankeramme som Niklas Luhmann og Christian Borch⁴. Spencer Brown⁵ har som udgangspunkt, at der skabes information i differentieringer som afgrænsninger mellem forskelle, og at undersøgelsen af sådanne forskelle ligeledes skaber information. Den metodiske konstruktion indbefatter mange sæt af differentieringer, der for os skaber mere relevant information. Ligeledes beror designprocessen på mange sondringer mellem f.eks. æstetik og funktion og andre dialektikker, der diskuteres i kapitlet *Design*. Afsøgningen af former i forhold til deres grænser bliver i vores fortolkning af formteorien en generator af designrelevant information. En afgrænsning markerer en forskel for et system og alle dets bestanddele, da grænser påkalder regulativer, specielt hvis grænsen krydses eller på anden måde afprøves. Systemet foretager en afgrænsning - et valg⁶, der vidner om systemets struktur. Struktur ses som et mønster af interne operationer i forhold til eksterne faktorer, som eksempelvis information fra andre systemer. Informationen opstår ifølge Spencer Brown som følge af at noget vælges som inderside af en afgrænsning og repræsenterer højere værdi i forhold til ydersidens lavere værdi. Det vil sige som to etablerede former med en grænse imellem dem. Værditilskrivningen blotlægger kriterier for hvad der defineres som værdi i systemet og implicit værdiens polariseringer som +/- . Værdi eller information påvirker systemets videre kriterier for operation. Luhmann formidler samme proces⁷ som en konsekvens af at mening konstant stabiliserer/destabiliserer sig gennem valg (selektioner), der skaber en kontingens. Kontingens⁸ er defineret som en streng af valg, hvor systemer gennem selektioner definerer sider og former ved at lave *distinctions* (betegnelser). Mindre abstrakt er en differentiering et valg, hvor en serie af valg skaber en retning der i sit udgangspunkt er uforventelig (heraf kontingensbegrebet). Personen⁹ eller systemet udsættes for selektionstvang på grund af et systemeksternt overskud af information og er nødsaget til at foretage kompleksitetsreduktioner som et håndteringsprincip. Kontingensen, som afspejler disse reduktioner, vil i analyse vise en generel retning for hvordan valg træffes, og blandt hvilke muligheder. Heri ligger også hvordan systemer *betegner*, og drivkraften bag denne

³ Spencer Brown, 1969

⁴ Borch, 2000

⁵ Spencer Brown 1969, p. 5

⁶ Forholdet mellem aktualisering og mulighed ifølge Luhmann (Luhmann, 1984, p. 44)

⁷ I det psykiske system. Et socialt system opererer ikke med mening som følge af individets konstruktion, men med andre kriterier, eksempelvis håndteringen af individuelle meninger i et demokrati som socialt system

⁸ Ordbogen siger om kontingens at det er en uberegnelighed, noget uforventeligt. Kontingens er i projektet her anvendt for at beskrive en serie af valg, der ikke altid er binære men altid har et udfald. Rækken af disse udfald skaber en 'retning', gennem kontingente handlinger, da valgets udfald er uforventeligt. Kontingens er en sekvens af aktualiseringer

⁹ Luhmann har et kompliceret personbegreb, der af pladshensyn ikke præsenteres her. Det er hos ham en paraplybetegnelse for subsystemer, der gennem kommunikation udgør 'en person'. Eksempelvis hvordan neurologiske netværk relaterer sig fortløbende til psykens opereren med mening

betegnelse ikke kan udtømmes med begreber som selektionstvang. Luhmann ekspliciterer også, som Spencer Brown og Borch, et moment af motivation i betegnelse som en usynlig selvreference. Det understreges samstemmigt, at det er motivationen, der driver opmærksomheden og de værditilskrivende handlinger, uden at denne motivation kan fremanalyseres eller fremelskes. Denne motivation sikrer i lige så høj grad handlekraft til at gennemføre valg og skaber en kontingens.

Med denne forståelse af information set i forhold til systemer skitserer vi en forståelse, der præger vores egen designaktivitet, forstået som en håndtering af information, der varetages af et designsystem (en kommunikativ praksis omkring designaktiviteten). Men samtidig bygger også vores forståelse af metodebrug på denne tankegang, idet metode ses som et ekspliciteret grundlag for den lange række af valg, der præger designsystemets informationshåndtering. Endelig indrammer denne forståelse vores tilgang til design mod brugere, da også forholdet mellem bruger og softwareinstrument kan analyseres i termer af information, system og motivation til håndtering af kompleksitet.

Iterationer

Tanken om iterationer har udgangspunkt i bl.a. en generel model, der stammer fra systemteori og kompleksitetsteori. Teoriretningerne behandler forhold mellem enheder som systemer, hvor alt lige fra os som designere, selve prototypen, brugere mv. tænkes som systemer der betjener sig af information. Modellen beskriver en iterativ proces, hvori outputs fra én iteration gøres til input for den næste iteration. Dermed etableres en selvreference i systemets operationer, der er med til at forme fremtidige iterationer. Men derudover kan disse fremtidige operationer også forstyrres udefra af information, formgivet som såkaldte perturbationer (irritationer). Eller af information indefra som de ovennævnte motivationsfaktorer. Dette sker såfremt systemet lader disse forstyrrelser indgå konstruktivt i dets videre iterationer og aktualiserer muligheder på baggrund af forstyrrelserne, som informationen genererer. Denne proces sker ifølge Luhmann bl.a. gennem observation af systemets omverden¹⁰. Ifølge Spencer Brown anvendes iagttagelserne af former til at sondre mellem grænser af allerede etablerede former overfor nye horisonter – med andre ord flyttes grænserne for formen iterativt gennem afprøvning af formens grænser. Dette er en udlægning af systemer som (iterativt) selvorganiserende størrelser, i kraft af deres dynamiske informationsudveksling med deres omverden. Organiseringen af information bliver reguleret med et system/horisont konditionale, hvilket er en regulering af systemers operation. I organiseringsprocesser betjener systemer sig af en operationel lukning, f.eks. som usynlige kriterier for hvilken information der indoptages/afvises og

¹⁰ Luhmann, 1984, kapitlet *Observation*

om denne information kan påkalde en ny iterativ proces. På samme tid er der en kommunikativ åbning, i kraft af at kriterier løbende forskydes på baggrund af indoptaget information. En åbning der muliggør en forandring, og en lukning hvor denne mulighed vurderes og eventuelt aktualiseres på et senere tidspunkt. Dette må ses som systemets punkt for selvtilstrækkelighed – at der opstår mønstre gennem kontingente handlinger, hvor et system afgrænses gennem aktualiseringer af nogle muligheder frem for andre. Heinz von Foersters¹¹ *order from noise principle* beskriver systemers opereren ud fra observation/distinktion og den 'støj', systemerne frembringer i denne konstruktionsproces. Støj – selve ordet smager af et biprodukt, noget undværligt eller noget forstyrrende. Støj kan beskrives som et overskud af henvisninger, da der er meget information i den støj, det enkelte system skal selektere iblandt før en struktur opstår. Systemets omverden er derfor det overskud af henvisninger, som gør at kriterier opstår internt i systemet for at reducere i overskuddet. I relation til information er udgangspunktet, at det er tilfældigt og uforudsigeligt at informationen bliver forstået, da der er meget 'støj' der skal filtreres fra. F.eks. forholdet mellem information og form (informationsmedie), der kan forskyde den mening, en modtager/horisont konstruerer – som en misforståelse. Der konstrueres dog en mening på baggrund af information, der er valgt ud blandt anden information, hvilket er en orden, der konstrueres. Gennem denne selektionsproces opstår en orden ud af støj, eller mening ud af meningsløshed, og løbende opbygning og revidering af kriterier internt i systemet. Egbert Jürgens¹², der er tidligere neurofysiolog, illustrerer *order from noise*-princippet med en sværm af fluer. Alle fluer opererer under kriteriet 'at søge midten', hvilket på et makroniveau skaber en figur/form – en sværm. På et mikroniveau er midten imidlertid aldrig fikseret, da alle fluer bevæger sig på samme tid, og ingen observatør på noget tidspunkt kan betegne den præcise midte. Iagttagelse af formen 'sværm' er et generaliseringsprincip i forhold til en kaotisk horisont og *order from noise* etableres, når tusind fluer bliver til en sværm. Kontingens er mikroniveauets streng af vilkårlige (ubestemmelige) valg, der i sin helhed på et makroniveau danner en retning – en form om man vil. Punktet for selvtilstrækkelighed er i denne tænkning aldrig fikseret og vigtigst af alt tidsligt ustabil som fluernes/systemernes konstante bevægelse. På makroniveauet er der imidlertid en mulighed for at observere tendenser, hvilket metaforisk kan udlægges som at se en sværm, der flytter sig i en bestemt retning. Optikken, et system observerer med, må justeres i forhold til kompleksiteten i det, der observeres. F.eks. observerer vi musikere, hvis behov ændrer sig i forhold til individuelle selektioner. Dette ser vi som et mikroniveau, vi må iagttage med en optik. Denne optik udvides

¹¹ Se <http://www.univie.ac.at/constructivism/HvF.htm>. Der er bygget software der illustrerer *order from noise*-tanken, som kan ses på enten <http://www.calresco.org/sos/calressw.htm> eller http://www.cogs.susx.ac.uk/users/andywu/r_map.html

¹² Egbert Jürgens, senior development manager på Native Instruments

efterhånden som vi foretager flere observationer, hvilket gør at kompleksiteten i observationen stiger efterhånden som vi følger musiker-tendenser. I observationen stiger antallet af kriterier i takt med at mikro/makro-niveauer bliver mere komplekse. Tendenser i omverdenen iagttages som f.eks. en fiksering på et makroniveau af enten 'sværm' eller 'sværmens retning'. Den løbende justering af optikker sker gennem iterationer, hvor forholdet mellem interne kriterier og ekstern kompleksitet er under konstant forandring.

Iterationen som grundprincip tjener dermed som illustration af en proces, hvori orden skabes ud af støj gennem en kontingent streng af valg og fravalg. Aktualisering af et valg skaber en forandring i systemet, hvilket bliver et 'udsigtspunkt' for systemet, før næste aktualisering skal foretages. Udsigtspunktet og en efterfølgende mulighed for iteration er præget af såvel informationsmæssig som tidlig kompleksitet. Udover denne fremadskuen i udsigtspunktet opererer systemet også med en tilbageskuen, der er funderet i hukommelse. George Spencer Brown inkorporerer i denne betragtning forløb i tid og tidens betydning. F.eks. i forhold til systemets hukommelse, hvor tid og hukommelse danner en tunnel mellem former, eller illustrerer sondringen mellem nyt og gammelt. Spencer Browns *re-entry*-begreb¹³ bearbejder denne genindføring af former over tid, hvori sondringen af former kan tage udgangspunkt i en forvirring af nogle former med andre.

Forvirringen (støjen) opstår, når en form er statisk og etableret med sider, og en form magen til placeres ovenpå. Dette gør hverken resultatet til en ny form, da der ikke kan observeres ændringer, eller til den samme form som før, da bevidstheden om at en ny form er lagt ovenpå er fastlåst i hukommelsen. Den 'nye' etablerede form ligner derfor, men er ikke 'lig med' den oprindelige form. At afdække *hvad der er forvirret med hvad* er en informationsgenererende proces i kraft af formetablering og logiske eksperimenter med formgrænser og formsubstanser. Eksperimentet i at sondre en grænse beskrives hos Spencer Brown gennem hans begreb *oscillering*¹⁴. Når en form er defineret, findes der altid 'den anden side' af formen. I operationen med at etablere form opstår prototyper på alternative former, og forestillede tilstande for denne form. Spencer Brown betegner den forestillede tilstand som en horisont – et *imaginary state*. Oscillering opstår i bevægelsen (svingning) mellem formen og dens forestillede modpol over tid. Tiden mellem betegnelser af tilstande kaldes *frequency*, og er et begreb for hyppigheden/forekomsten af reelle og forestillede tilstande. Spencer Browns grundsten for at kunne erkende ud fra sondringer/oscilleringer er hukommelsen. Hver sondring mellem former skaber information, som vil gå tabt, hvis informationen ikke bliver registreret i processen. *Re-entry*-begrebet anslår også denne emergente type erkendelse, hvor formforvirring over tid skaber information og er et produkt af hukommelsen.

Iterationsbegrebet beskriver dermed en sondringsproces, der inkorporerer hukommelse, kompleksitetshåndtering og etablering af kriterier i et samlet hele.

Iteration, information og designsystem

En præmis for formuleringen af vores designmetode er en forudgående forståelse af designprocessen som en proces, der er forankret i og foranlediget af et designsystem og dets omverden. Med 'designsystem' mener vi den informationsbaserede praksis eller et system bestående af kommunikation om en designaktivitet. Et system, der begriber og formidler den konkrete designproces; en kommunikation omkring valg og fravalg, muligheder og barrierer, forventning og overraskelse, nutid og fremtid, der iterativt forsøger at stabilisere/destabilisere en kontingent proces. En proces, der gennem formsondringer leder fra stor usikkerhed til større sikkerhed på designets endelige form. I designsystemet foregår denne kommunikation og konstante skelnen med reference til såvel tidligere iterationer som aktuelle forstyrrelser.

Vi kan anvende en veltrådt skovsti som eksempel og se iterationen som (endnu) et gennemløb af den konkrete sti, der derved gøres endnu mere fastdefineret, dvs. fastlægges som 'denne sti og ingen andre'. Meningen med at inddrage system- og formteori er netop at definere 'de andre stier', dvs. fravalg og ikke-aktualiseringer som senere kan genaktualiseres gennem refleksion og hukommelse. Der kobles en tidslighed til aktualisering/genaktualisering i disse processer, hvor en operation er et forløb, der kræver ressourcer. Ressourcer muliggør iagttagelse af forstyrrelser.

For at tage sti-eksemplet op igen, er det muligt at iagttage forstyrrelser fra stien. Eksempelvis spændende steder i skoven, der foranlediger et svinkeærinde, der så i fremtidige iterationer kan forstærkes i retning af en etableret sti. Eller stien kan gro til og forsvinde, hvis dens mål ikke længere er relevant. Vekselvirkningen mellem selvreference (f.eks. i form af reference til tidligere erfaringer og opnåede kompetencer) og fremmedreference (f.eks. i form af brugerundersøgelser og -tests) bliver dermed et væsentligt element i forståelsen af dynamikkerne i designprocessen. Forholdet er også væsentligt i formuleringen af en designmetode, der i tilstrækkelig grad begriber og muliggør realiseringen af disse dynamikker – vekselvirkningen fra analyse til syntese. Disse dynamikker angår designmetoden, formuleret som en abstraktion af designprocessen. Men også designprocessen, forstået som en konkret realisering af designmetodens anvisninger, i forhold til løsning af et specifikt designproblem. Sådanne dynamikker har vi indledningsvist forsøgt at eksemplificere ovenfor med vekselvirkninger som valg og fravalg, muligheder og barrierer, osv.

¹³ Spencer Brown 1969, p. 69

¹⁴ Ibid., p. 61

Pointen er så, at såvel de dynamikker, vi opererer med *indenfor* designproces og designmetode, som de dynamikker, vi anvender til at beskrive designprocessen *udefra* (de dynamikker, der indkredser det metodiske problem i specialet) lader sig beskrive som vekselvirkninger. Disse vekselvirkninger kan beskrives med udgangspunkt i ideen om et selvreferentielt designsystem. Beskrivelsesformen muliggør en kobling af *design og konstruktion* med *refleksion og metodik* indenfor den samme terminologi og samme teoretiske baggrund (systemteori). Relevansen af systemteori og formteori er dermed, at de muliggør denne kobling indenfor samme terminologi og teoretisk basis.

En fundamental og vigtig vekselvirkning er forholdet mellem innovation og adaption, hvilket er en sondring med mange tilhørsforhold. For det første i forhold til en metodisk-teoretisk balancering, og for det andet i forhold til vores konkrete brug af teknologi. Dette er emnet for næste afsnit.

Innovation og adaption

Tankerammen inddrages som et bidrag til konceptet og processen bag prototypen, da sondringen resulterer i ny information om vores design-koncept, såvel som den metodiske udviklingsproces. Udvikling kan teoretisk defineres som enten et innovativt, nyskabende virke eller et adaptivt, tilpassende virke. Afsnittet diskuterer dette spændingsforhold, ikke som *enten/eller*, men som *både/og*. Afsnittet vil tage udgangspunkt i innovation og adaption som vigtige sondringspoler i en udviklingsproces, ikke som modsætninger der udelukker hinanden, men som et informationspotentiale. Informationen opstår i formsondringer af f.eks. adaption af teknologi i forhold til innovation af teknologi.

På det konceptuelle plan kan prototypen kaldes innovativ, da den understøtter et nyt brugsmønster, hvorimod den teknologiske akse af prototypen i højere grad bærer præg af en adaption af eksisterende komponenter. Vi finder bevægelser mellem innovative og adaptive elementer givende i forhold til at allokere prototypens styrker, og som et processuelt beskrivende værktøj, der bidrager til specialets refleksive tråd.

Innovation i forhold til designet er at relatere prototypens brugbarhed til prototypens oplevelsesværdi. Sondringen mellem innovation og adaption ansporer videre, hvor prototypen iht. egne definitioner er innovativ, hvilket også kan ses i specialets disponering af primære arbejdsområder og prioriteter. Innovationen ligger i skabelsen af en interaktionsform, hvor teknologi adapteres for at understøtte innovationen.

Behov for innovation og adaption

Udvikling har altid et udgangspunkt; et punkt hvor der findes behov, som ikke er varetaget af et design. En given udviklingsproces vil derfor etablere sig selv som svar på både udtalte og udtalte behov og tilpasses ad hoc omstændighederne. Innovation beskriver i specialet, hvordan et udtalt behov varetages af et artefakt – som en opfindelse. Adaption beskriver hvordan formulerede problemdomæner kan overkommes med artefakter, der varetager ekspliciterede mangler i forhold til dette – justeringer af opfindelser.

Men hvad er hvad? Det er forholdsvis ligetil at argumentere for at adaption er opfindelse på et andet niveau, eller at innovative designs læner sig op ad andre adaptive problemløsninger i konteksten. Opdelingen er i specialet foretaget i forhold til hvornår behov er ekspliciterede, og i forhold til hvad. Refleksionen, der opstår i sondringen, tilvejebringer konstant nye vurderingskriterier, der over tid kan genindføres i designprocessen som et metodisk værktøj¹⁵. Refleksionen er relateret til behov; hvilke forhold disse behov står i, og hvad de er defineret igennem. Refleksionen blotlægger også, hvordan behov og nyskabelse behandles som information i en proces, og hvilke kriterier informationen underlægges. Dette kan eksempelvis ske i designmæssige udsagn som: ”koncept afvist pga. manglende nyskabelse”, ”ikke fungerende pga. manglende teknologisk adaption”, ”innovativt men upassende interface”. Samtidig kan dette også vendes til positiver: ”godtaget pga. innovativt koncept”, ”passende og adapteret interface”, osv.

Innovation/adaption - tid, mål og metodik

Innovation har en tidlig binding, hvor det innovative eksempelvis behandler relevante problemer i samtiden, som en periode med en start og en slutning. Det innovative aspekt forsvinder, hvis designet ikke modsvarer behov, eller en løsning der ikke længere er kompatibel med omverden – dvs. et design, der kom for tidligt eller for sent i forhold til omstændigheder og omverden. Adaption er i specialet tænkt tidligt fragmenteret, da adaption kan udlægges som iterationer af foregående udviklinger, og som en periode der starter, men sjældent slutter. Adaption er en fortløbende, ikke-lineær proces, der er præget af en justeringstendens. Adaption harmoniserer med andre ord innovationer. I konkrete designprocesser bliver innovation ofte et mål for udvikling. Der indføres nærmest pr. automatik deterministiske metodikker, hvor innovation kommer ud af teknologi, korrekte procedurer, rigtige analyser, det rigtige tidspunkt osv. Innovation er ikke tænkt som et mål, der kan nås med sådanne metodikker i specialet; innovation afhænger af et tilfældigt sammenspil mellem kontekstuelle faktorer som eksempelvis teknologi, brugstypologi og samtiden. Analyser og strategier kan kun i nogen grad afdække potentialet af et design og dets innovative

¹⁵ Her tænker vi specielt på begrebet om *re-entry*

værdi. Eksempelvis vil brugersegmentering eller valg af teori/metodik kun indføre et mindre udsnit af dette potentiale i designprocessen, hvorved kortlægningen af potentialet er underlagt redigering af de undersøgende. Dermed er en undersøgelse ikke det samlede antal muligheder for accept/afvisning af et design, eller på nogen måde vished om at designet vil opfattes innovativt. Analyser er kun retningsangivende fragmenter, og en basisviden, der må behandles som ikke-definitiv viden om innovationsværdi, men som et produkt af fortolkning.

Innovationssfærer

En sfære er (metaforisk) et afgrænset rum, der indeholder elementer og forhold mellem elementerne. Heriblandt modsætningsforhold som skaber en diversitet, hvilket ifølge Kalthoff & Nonaka¹⁶ er en kilde til innovation. At indramme diversiteter er umiddelbart en holistisk tilgang, der binder fænomenet innovation sammen med gentænkning af traditionelle modsætningsforhold, hvilket specialets behandling af designmæssige dialektikker også indebærer. Vi overtager tanken om en sfære som et stort antal sæt af definerede ekstremer, hvor yderlighederne er polariserede punkter, der er koblet i et forhold. Holismen toner vi dog på nogle områder ned, da de enkelte elementer er af stor betydning for både processen og designet. Dog er det holistiske udgangspunkt givende, når elementer skal sammenfattes i eksempelvis et koncept eller en systemfilosofi.

Kalthoff & Nonaka¹⁷ konstruerer en sfære omkring tre punkter: *Management Innovation*, *Technological Progress* og *Aesthetic Appeal*. Organisationen og dens ledelse skal ifølge Kalthoff & Nonaka opfinde objekter i forhold til organisationens teknologi, og give det æstetisk appel. Disse løse og ret ubrugelige definitioner og handlingsanvisninger frasiger vi os. Vi beholder imidlertid sfæren og overskrifterne:

Management innovation definerer vi som designorienteret informationsbehandling i en organisation. Specielt i forhold til hvordan koncepter udvikles og på baggrund af hvilken information, og hvordan denne information sætter formsondringer igang. I designprocessen er vi nødt til at tænke kreativt, og tænke på brugerne, der selv tænker kreativt og har forventninger om innovative/adaptive tiltag. Samtidigt skal informationen registreres, lagres og behandles, og videre må vi vurdere, hvordan den indgår i designet, på hvilket tidspunkt og med hvilke bevæggrunde. Vi opstiller med andre ord kriterier på baggrund af sondringer mellem innovative og adaptive elementer, hvilket også ses i designprocesser uden for universitetets vægge. I en virksomhed kan det være marketingafdelingen, der kommer med et design, hvor det er vurderet innovativt med kriterier som ”salgbarhed”, ”hype” og ”tendenser”. Er det ledelsens forslag, vil et innovativt design

¹⁶ Kalthoff & Nonaka, 1997, p. 155

¹⁷ Kalthoff & Nonaka, 1997, p. 156

ofte være sat i relation til firmaets brand, produktlinjer og være vurderet med kriterier som ”konsistens” og ”firmaøkonomi”. Kommer programmørerne med et design, er det for første vurderet med kriterier som ”muligt”, og for det andet ofte løsrevet fra alle andre kriterier i firmaet. Specialets behandling af designrelevant information er også et forsøg på at opøve management strategier og relatere dem til virkelighedens virksomheder. Et mål er at eksperimentere i så vid udstrækning som muligt, da universitetet som organisation tillader dette. I en markedsorganisation er økonomien en faktor i forhold til dybde og grundighed af metodiske undersøgelser eller udviklingstiltag, der sætter grænser for eksperimenter. *Management innovation* er nye udviklingsmetoder; irritationer af informationsudvekslingen i organisationen, der søger muligheder for innovation, uden at strukturen kollapser under informationsmængden. Her ser vi brugerinddragelse i konceptudviklingen, ikke kun i testfasen, som et innovativt tiltag. Hvordan det konkret implementeres må ske gennem adaption og metodemæssige iterationer.

Technological progress ser vi udvidet til teknologi der findes, ikke kun i organisationen, men også som komponenter, tredjeparter har lavet. Elementer der ligeledes gennem adaption fortløbende kan inkorporeres i koncepter, designs, mv. Et vigtigt indstik er, at det i lige så høj grad er viden om processer, som viden om teknologi, der gør et design passende, godt eller i det mindste bedre. Ofte har teknologien virket som forklaringsmodel for innovation, hvor det ikke kun er et element af en sfære, men sidestillet med innovation. Dette er umuliggjort af krav om at teknologien skal styres i en given struktur, og at det er brugere af teknologien, der fastslår den innovative værdi.

Aesthetic appeal ser vi også som et informationselement, der med fordel kan behandles både internt og eksternt i en organisation. Hvordan vi konkret fastlægger denne æstetik, behandler vi i afsnittet *Æstetik* i kapitlet *Design*. Vi har fastholdt appel som en relativ term, en given bruger kan afvise eller acceptere.

Sfæren indeholder elementer, hvis divergerende forhold via analyse kan blive en basisviden om designet. Vi anvender pragmatisk sfæren som en opsummering af muligheder og modsætninger. Endvidere placerer vi i sfæren et forhold mellem kombinatoriske muligheder og egne ideer som er formuleret internt i designsystemet. Sondring af muligheder og idestyring er i denne proces også rettet mod omverden for designet. Sfæren beskriver et samlet rum for innovation, som vi prøver at orientere os indenfor i overensstemmelse med æstetiske, funktionelle og metodiske beslutninger. Specielt når vi taler om kreative processer bliver navigationen i sfæren relevant, da modsætningsforholdene er flere og mere diffuse, men stadig har en basis i de tre nævnte punkter. Forholdet mellem kreativitet og systematisk fiksering af denne proces tager vi op i næste afsnit.

Priming, kreativitet, inspiration og information

Afsnittet berører emner som kreativitet og fikseringen af kreative processer. Denne fiksering kan benævnes inspiration, der begrebsligt vil behandles som information, der indoptages i den kreative proces. Fikseringen ser vi som en emergent proces, hvori designet gradvist udvikles og specialiseres.

Framework: Kreativitet som kombinations- og forestillingsproces

Kreativitet anses i specialet som en kombinationsevne, hvor elementer sammenkædes i henhold til forskellige kriterier. Elementer som instrument og instrumentbetjening sammenstilles ud fra æstetiske eller funktionelle kriterier som 'flot' eller 'fungerende'. Denne sammenstilling sker ud fra forestillinger om hvordan elementer kan kobles, hvilket vi anser som en kreative sondringer af et design under udvikling. Kreativitet er derfor også en forestillingsevne om f.eks. former eller musikalske strukturer. Dvs. former, der kombineres ud fra forestillinger om at kombinationen er attraktiv og mulig. Forestillinger er en sammenstilling af elementers egenskaber, og hvordan disse egenskaber kan kobles til andre sæt egenskaber. Lyd som materiale har den egenskab at den er modificerbar. Ligeledes har et interface samme modificerbare egenskab. Disse to modificerbare domæner kobles konkret i koden. I denne kobling opstår nye egenskaber, da objektet, værket, mv. ændrer form i takt med at egenskaber kombineres – som sondringer mellem virkelige og forestillede former¹⁸. Kombination af egenskaber er på denne måde tæt forbundet til forestillinger¹⁹, hvor kreativitet er processen, der flytter forestillinger fra ideplanet til et objekt eller et værk – som kontinuerlig fiksering. Kreativitet er som proces en selekterende mekanisme, der for det første stabiliserer en konstruerende praksis, gennem afprøvning eller sondring af etablerede former, og hvordan formen videre kan udvikles fra det nuværende stadie (udsigtspunkt). For det andet skaber kreativitet differentieringer mellem elementer, og bibeholder f.eks. en designer eller en musiker i en selekterende modus. Kreativiteten udfolder sig altid indenfor begrænsninger, for der vil være sondringspoler for objekter, der er umulige eller ikke attraktive. Dette kan i praksis ses som fravalgte former for designet, der ville være 'upraktiske', eller lydlige muligheder, der er fravalgt på grund af 'ubehagelig' klang. Disse begrænsninger for muligheder fastsættes af individet vha. operationelle kriterier som f.eks. æstetisk overbevisning, og referentielle kriterier som eksempelvis

¹⁸ Iht. formteori *re-entry* af former i former. Processen i at sondre mellem disse former kaldes oscillering, jf. afsnittet om *Iterationer*

¹⁹ Spencer Brown udlægger dette som sondringspoler for en given form, hvilket vi opsummerer som forestillingsevner. Spencer Brown nævner også selv formsondringer som sammenstillinger af mentale modeller for forestillede tilstande af formen. På teoriens præmisser: hvad der findes på den anden side af distinktionen. Se evt. afsnittet om *Iterationer*

konceptuel stringens. Konkret konfigureres værktøj og materiel i overensstemmelse med en ide, hvor kreativiteten justerer konfigurationen.

Kreativ fiksering

Kreativiteten er som nævnt en stabilisering af en handlende proces mod en form – hvilket vi benævner fiksering. At sondre en form (og tilhørende forestillede tilstande for denne form) er at fikse en form, hvilket i praksis sker som kombinationer af elementer. Denne praksis kan som anslået i afsnittet *Iterationer* forskydes iterativt gennem information. Michael Hamman²⁰ udlægger processen i fiksering som genaktivering af forestillede tegn (*cues*): ”*creative thinking involves a process of iterative activation of cues which direct the generation of ideas*”.

Her er en fiksering af bestemte *cues* defineret som en negativ retning for en kreativ proces; som et *deadlock*, der hindrer et objekts form. Steven M. Smith deler dette synspunkt i artikler som *Incubation and the persistence of fixation in problemsolving*²¹, hvor der foregår en negativ fiksering af kreativitet. Hamman definerer en styring af kreative processer som *priming* (podning) af den kreative tænkning. Smith er af den kognitive tradition, og har traditionen tro lavet en del laboratorieforsøg. Designere, der udsættes for bestemte former og farver, vil i firs procent af tilfældene have tilbøjelighed til at designe noget lignende i konkretisering af lignende problemstillinger. Dette ses som en negativ retning for den kreative tænkning og designet, og ikke som et potentiale i processen, hvor man kan betjene sig af eksempelvis eksisterende problemløsninger.

På den anden side kan *priming* ses som inspiration, hvor vi ser et potentiale, og ikke et negativt *deadlock*. Potentialet ser vi i at forståelsesrammer af problemdomæner og kreativiteten indenfor disse rammer kan fikses gennem genanvendelse af former, strukturer, koncepter, mv. Iterative genaktiveringer af designrelevant information anser vi som et grundlag for formen, og ikke en trussel mod processen i formgivningen²². Konkret inddrager vi andre designere som en *priming* af vores kreativitet for at betjene os af eksisterende og fungerende problemløsninger, som kan videreudvikles.

To slags kreativitet

I specialet arbejder vi med kreativitet på to akser: For det første vores egen kreativitet som systeminterne kriterier i formgivningen. For det andet musikerens kreativitet som kriterier, der

²⁰ Hamman, 2000

²¹ Se www.ecologylab.cs.tamu.edu/people/steve.html

²² *Re-entry*-begrebet beskriver videre det informationelle potentiale i formforstyrrelser

anvendes i udforskningen af prototypens funktionalitet, og til at vurdere hvorvidt funktionaliteten understøtter musikerens (systemlokale) kriterier for kompositorisk praksis. Vi anvender derfor såvel systemintern som systemekstern information til at forskyde vores fiksering af muligheder i henhold til en bruger, og søge input i den designmæssige forestillings- og kombinationsproces. I forhold til musik og musikalsk praksis har vi søgt empirisk inspiration hos designere og musikere, der præsenteres i afsnittet *Empiri – domæneindkredsning* i kapitlet *Design*.

Vi har også søgt teoretisk inspiration hos forskeren og komponisten Otto Laske²³, der anser komposition som en leddeling mellem interfaces, der udvikler og understøtter musikalsk materiale, og de processer, der genererer musikalske strukturer. Vi udlægger i specialet leddelingen i musikalske strukturer som et forhold mellem struktur og tekstur. *Struktur* ser vi abstrakt som et skelet, der markerer nøglepunkter der løbende defineres, efterhånden som kompositorisk idémateriale omdannes til kriterier. Struktur kan også ses som et framework, der gennem konfiguration bliver en base eller fundament for enten et værk eller en *session*. *Tekstur* ser vi abstrakt som elementer, der kan materialisere skelettet i en konkret auditiv form. Dette ser vi konkret som *samples*, instrumenter og effekter. Teksturer er det eneste, der er muligt at høre af værkets elementer, hvor skelettets struktur varetager en indbyrdes orden mellem disse elementer som f.eks. rytmisk placering.

Vi abonnerer på ideen om at software kan teste (formmæssige) grænser og muligheder i interaktionen med struktur og tekstur. Software rummer begge akser af kreativitet og er som system åbent for informationsforskydninger eller kreative indfald. Dermed er der også et potentiale for at rammerne for musik ændres, efterhånden som informationsmængden i designsystemet stiger. At ændre interaktionsmulighederne for musikalske strukturer og teksturer er vores belæg for at søge en *priming* af vores kreativitet. Vi forsøger i samme bevægelse at *prime* de brugere, der har *primet* os, ved at bygge software til musikere på baggrund af information om kreativitet fra dem.

Opsamling

Grundlæggende handler de foregående afsnit om vilkårene for den iterative proces, der er blevet beskrevet gennem en basis-terminologi. Vilkår, der vedrører informationens placering, dens iterative bearbejdelse, og hvordan informationen bruges i sondringer af innovation, adaptation og kreativitet.

²³ Se <http://www.emf.org/subscribers/laske>

Fra denne begrebsafklaring vil vi nu kæde metoder fra systemudvikling med basis-terminologien, hvilket tjener til at formulere den metode, vi anvender for design.

Formulering af designmetoden

Vi fremsætter eksistensen af et designsystem som forudsætning for formuleringen af en designmetode og realiseringen af en designproces. Vi vil nu se nærmere på det konkrete indhold af designmetoden ud fra det præsenterede form- og systemteoretiske grundlag.

Vi kan i første omgang illustrere en iterativ proces, der er åben for forstyrrelser udefra, med den nedenstående figur. Perturbationen er forstyrrelsen, mens iterationens cirkel illustrerer hvordan systemets aktuelle operation forholder sig til tidligere operationer gennem selvreference:

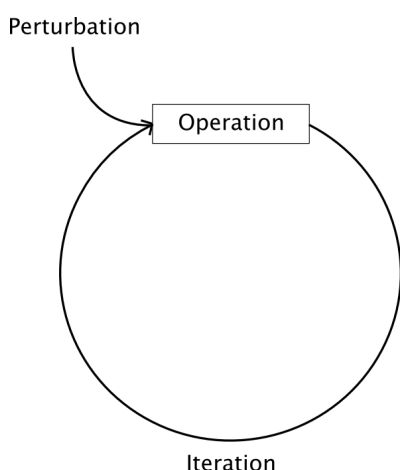


Fig. 1: Iteration som følger perturbation

Cirkelbevægelsen illustrerer relationen mellem input og output, forstået på den måde at output fra iteration 1 gøres til input for iteration 2 og så fremdeles. Endvidere illustrerer pilen 'udefra' perturbationer, der iagttages og muligvis indoptages i systemet. Input og output forstås meget generelt som information; det kan eksempelvis dække over resultatet af en brugerevaluering (en form for output), der gøres til input for en efterfølgende iteration, der reviderer designet bl.a. ud fra denne evaluering. Cirkelbevægelsen illustrerer på denne vis såvel selvreferencen som fremmedreferencen i systemet, og beskriver på denne måde designmetoden som en abstraktion over den vekselvirkning mellem designercentreret og brugercentreret design, vi har skitseret som en del af vores metodiske problem. Formteorien ved Spencer Brown skitserer også forandring som sondringer mellem sider. Her virker teorien som en rig beskrivelse af, hvordan iterationer forandrer former. Formteorien blotlægger ikke kun indersiden og ydersiden af systemet, men stiller også

analytiske spørgsmål til hvordan disse sondringspoler etableres. Hvilket er et vigtigt moment, når iterationernes løbende forandring skal beskrives – et blik på designet før og efter iterationen.

For at komme videre fra denne indledende identificering af de abstrakte egenskaber ved en iterativ proces finder vi inspiration i en konkret begrebsliggørelse af designprocessen, der tager udgangspunkt i en cirkulær/iterativ forståelse fremfor en lineær/sekventiel forståelse, nemlig John M. Carrolls *task-artifact cycle*²⁴:

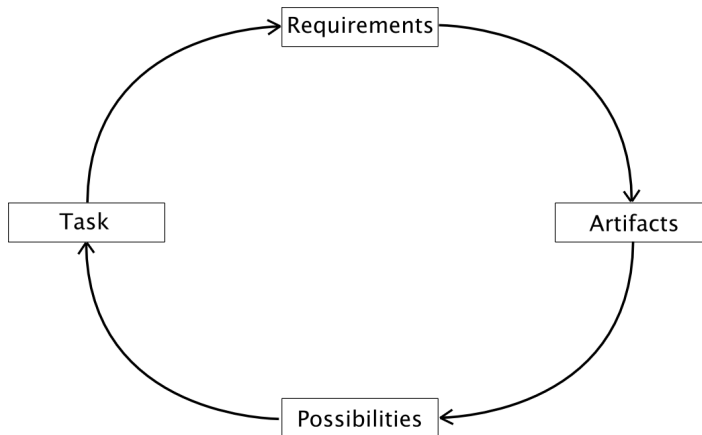


Fig. 2: Task/artifact cycle, efter Carroll, 2000.

I denne cyklus beskrives, analyseres og begribes *tasks* som det første trin, hvad der dernæst udløser specifikke krav (*requirements*) til et design, som efterfølgende implementeres som et konkret artefakt. *Requirements* sidestiller vi med de æstetiske og funktionelle krav til prototypen, som opsummeres af koncepter. Disse krav er emnet for *Design*-afsnittet.

Task (opgave) er et abstrakt begreb, der i vores design dækker over procesorienteret lydgenerering. En diffus opgave, der er svær at beskrive entydigt, men som vi forsøger at tilnærme os gennem empiri og teori, der leder til scenarier, *use cases* og en prototype.

Artefaktet eksisterer eller 'lever' derefter i en række brugskontekster, hvilket igen afføder nye muligheder og forventninger i sammenhæng med udformningen af nye teknologier. Fra designsystemets side kan disse muligheder endeligt iagttages som grundlag for beskrivelsen af nye *tasks*, hvormed cyklussen starter igen. Modellen beskriver på abstrakt vis processens karakter af tilbageføring og viser et *increment*²⁵. Et *increment* er forøgelsen af informationsindhold mellem iterationer, som er et resultat af f.eks. løbende konstruktion og analyse. Iterationer kan udlægges som en mikroenhed i udvikling og hænger sammen med hvad der tilbageføres, i form af

²⁴ Carroll, 2000

(gen)læsningen af designets værdi²⁶ i forhold til en brugskontekst. Med andre ord gælder cyklusmodellen for foreløbige mock-ups, konkrete koncepter, abstrakter og antagelser, der formes i en prototype. Prototypens placering i en kontekst iagttages gennem brugertests, interviews, mv. Cyklusmodellen gælder også for det endelige design, hvis placering i konteksten iagttages gennem bug-databaser, brugerrespons på online-fora, marketing-undersøgelser, økonomi, mv. – alle procesrelaterede vurderingskriterier, der udspringer af organisatorisk virke. Vores udvikling divergerer fra nogle af disse vurderingskriterier, når vi arbejder på universitetet, men vi har øje for dem, da de findes i verden uden for universitetet.

I vores brug af Carrolls abstraktion over designprocessen justerer vi delvist hans terminologi. For det første finder vi det værdifuldt at understrege artefaktets placering i en kontekst udenfor designsystemet. For det andet må vi nødvendigvis redefinere *task* som oplevelse eller aktivitet. Dette gør vi, da vores designproces ikke retter sig kun mod løsning af en bestemt arbejdsopgave (hvad der ville retfærdiggøre ordet *task*), men også mod understøttelse af en bestemt oplevelse. Oplevelsesbegrebet indebærer mere end en foruddefineret opgave hos brugeren; en lyst til at udforske, eksperimentere, lege og blive overrasket. Oplevelsen ser vi som en individuel konstruktion, der er baseret på motivationsfaktorer, som ligeledes er individuelle konstruktioner²⁷.

For så vidt som Carrolls model er en idealisering af det informationelle forløb i designprocessen, illustrerer den også designsystemets indre beskrivelse af dets omverden, i særdeleshed den omverden som fremtidige og nuværende brugere udgør. Dette kommer til udtryk i hvad vi her har kaldt artefaktets placering i en kontekst, men også i form af nye teknologier, der åbner nye muligheder for designet. Konteksten for designet beskrives systeminternt gennem markedsundersøgelser, fora-undersøgelser, interviews og tests af prototypen, der alle bygger på system-definerede kriterier og metoder. Mulighederne for videreudvikling af designet bygger på designsystemets anvendelse af denne kontekstuelle information, men også på de muligheder og begrænsninger, som nye teknologier udgør. En anden aktivitet er derfor en løbende kortlægning af de aktuelle teknologier, der er relevante for designet. Inddrager vi den systeminterne distinktion mellem system og omverden i det kommunikative kredsløb, ser modellen således ud:

²⁵ Booch, Jacobson & Rumbaugh, 1999, p. 101

²⁶ Værditilskrivende handlinger er drevet af motivation og kommer til udtryk i kontingente valg. Jf. afsnittet *Information*

²⁷ Jf. vores behandling af motivation under *Information*

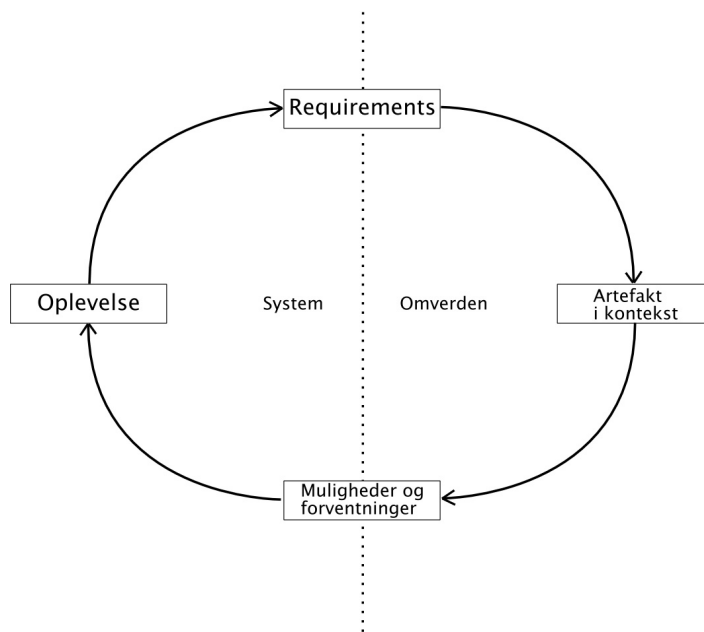


Fig. 3: Adskillelsen mellem system og omverden, samt redefineret terminologi

Det er vigtigt at holde sig for øje, at 'omverden' her refererer til systemets indre konstruktion af dets omverden, dvs. dets selvreferentielle beskrivelse af brugerforventninger, teknologiske muligheder, resultater af brugerevalueringer, osv., der hele tiden sker med reference til det aktuelle designproblem og den igangværende iterationsproces²⁸. Når vi taler om muligheden for at arbejde udenom en metodisk 'glasvæg' mellem designgruppen og dens empiri, må det derfor ske med det forbehold, at vi i alle tilfælde også taler ud fra en selvreference i designsammenhæng. Omverdenen for designet eksisterer i designsystemet i form af indsamlet og fremanalyseret information, hvis konstruktion er formet af de initierende betingelser, designgruppen opstiller for sit arbejde. Fremmedreferencen i designsystemet kortlægger ikke omverdenen på brugernes eller teknologiens præmisser, men på designets egne præmisser.

Ligeledes indikeres en skelnen mellem system og omverden, foranlediget af Borchs, Spencer Browns og Luhmanns tanker, der indebærer en mulighed for forstyrrelse gennem differentiering og *re-entry*. Hermed kombinerer vi Carrolls model for designprocessen med iterationsbegrebets blik for perturbationens rolle i det iterative forløb. Med den modifikation, at perturbationen kan indtræde i hver af de fire faser, som Carroll beskriver. En grundlæggende ide optræder som den initielle forstyrrelse, ud fra hvilket designprocessen (og dermed designsystemet) etableres:

²⁸ Jf. behandlingen af udsigtspunkt og hukommelse under *Iterationer*

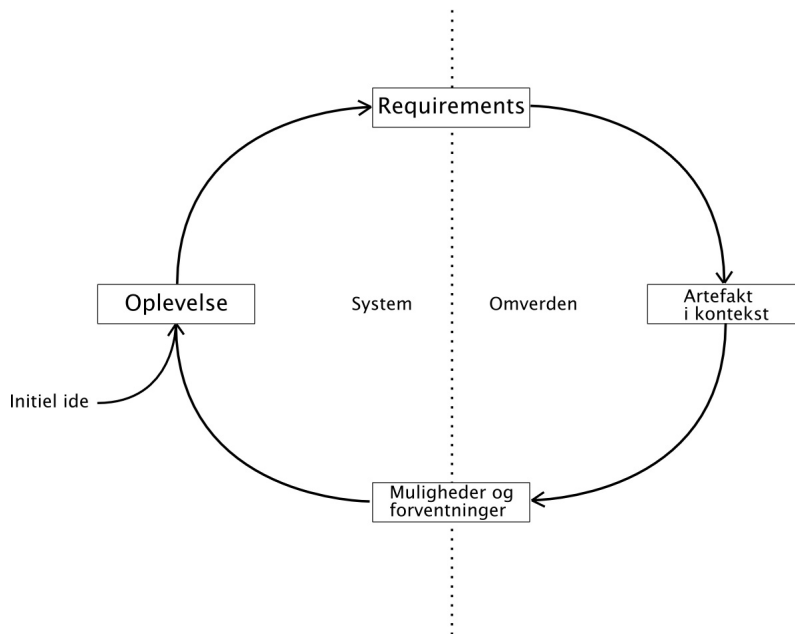


Fig. 4: Den initielle forstyrrelse

Som kilden til etablering af designsystemet iagttager vi altså en indledende forstyrrelse af en kommunikativ praksis, der eksisterer i forvejen, og som f.eks. kan være forankret i en konkret udviklingsafdeling eller designgruppe. Her opererer designsystemet med en høj grad af usikkerhed på såvel målet for en eventuel designproces som de muligheder og forventninger, der gør sig gældende i domænet for designets brug. Egentlig har vi altså at gøre med et svært definerbart begreb om 'den gode ide', den første formulering af et koncept, der kan tjene som en kreativ forstyrrelse af vante mønstre med rod i designgruppen selv. Selvom usikkerheden på designet og designprocessen altså her er stor, fastlægges der med denne initiering af processen alligevel nogle helt overordnede rammer for designprocessen; herunder den generelle retning og vision for designet, der på dette tidspunkt er uafhængig af konkrete teknologiske muligheder og barrierer såvel som brugeres forventninger. Dermed fastlægges også en kurs i retning af reduktion af disse usikkerheder: Spørgsmål om brugerforventninger, konceptuelle og teknologiske muligheder formuleres indledningsvist og tjener til gradvis vidensindsamling og syntese af designs gennem fremtidige iterationer.

En systemintern vurdering af 'den gode ide' og dens værdi kan for så vidt føre til at ideen forkastes, men vælger designgruppen at fastholde ideen, beskriver designmetoden en videre kurs gennem iterationer, der har til formål at raffinere og redefinere designet gennem reduktion af usikkerhed på en række områder. Samtidig må designsystemet forholde sig åbent i forhold til uventede bruger-input, nye teknologier, indskud til anderledes koncepter, og 'gode ideer' internt i designgruppen som en udkrystallisering af information. Når 'brugere' bliver 'brugergrupper'

bevæger vi os også fra et mikro- til et makroniveau i iagttagelsen af brugere som en 'sværm', som bevæger sig iht. nogle tendenser. Tendenserne beskriver nogle præferencer på et makro-niveau, der iagttages som forstyrrelser af designet, og fungerer som det generelle billede af brugergruppen.

Åbningen af processen kan kun muliggøres gennem en åbning af designsystemet vha. informationsindoptagelse af forskellig art. Dette fordrer en balance mellem åbning og lukning. Det er denne balancegang, vi med den næste revision af modellen for designmetoden søger at illustrere – nemlig de forskellige former for forstyrrelse, systemet kan forholde sig til i forbindelse med cyklussens forskellige faser:

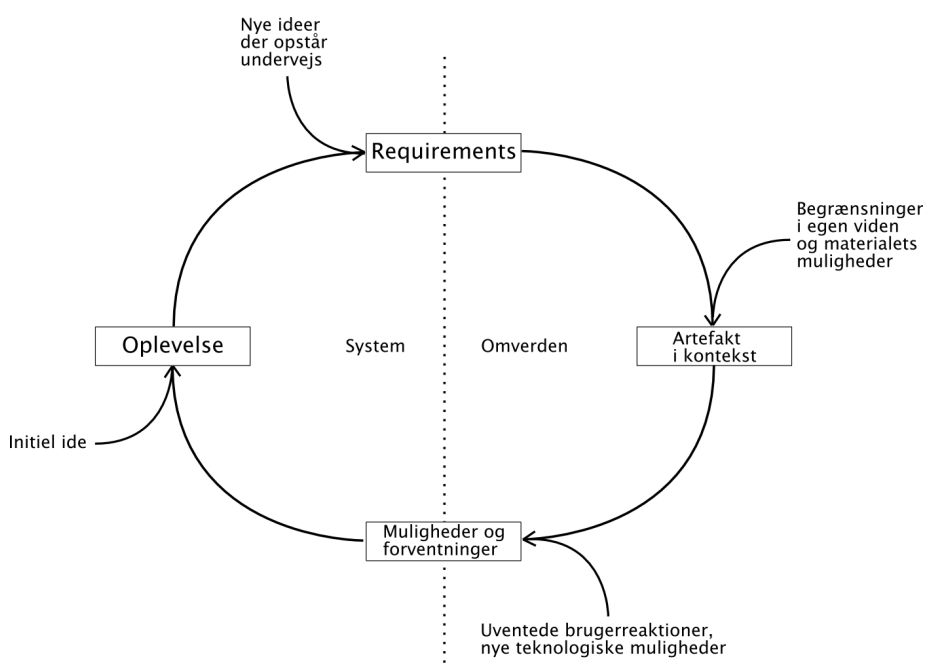


Fig. 5: De forskellige perturbationer/forstyrrelser, systemet kan forholde sig til

Endelig indarbejder vi i metoden fire specifikke faser, der tjener som blotlægning af vejen mellem de fire positioner, som figuren ovenfor illustrerer. I disse fire faser arbejder vi med den forrige 'stations' erfaringer såvel som de forstyrrelser, der bliver indoptaget i denne fase. Faserne tegner et billede af de konkrete arbejdsprocesser, der indgår i metoden:

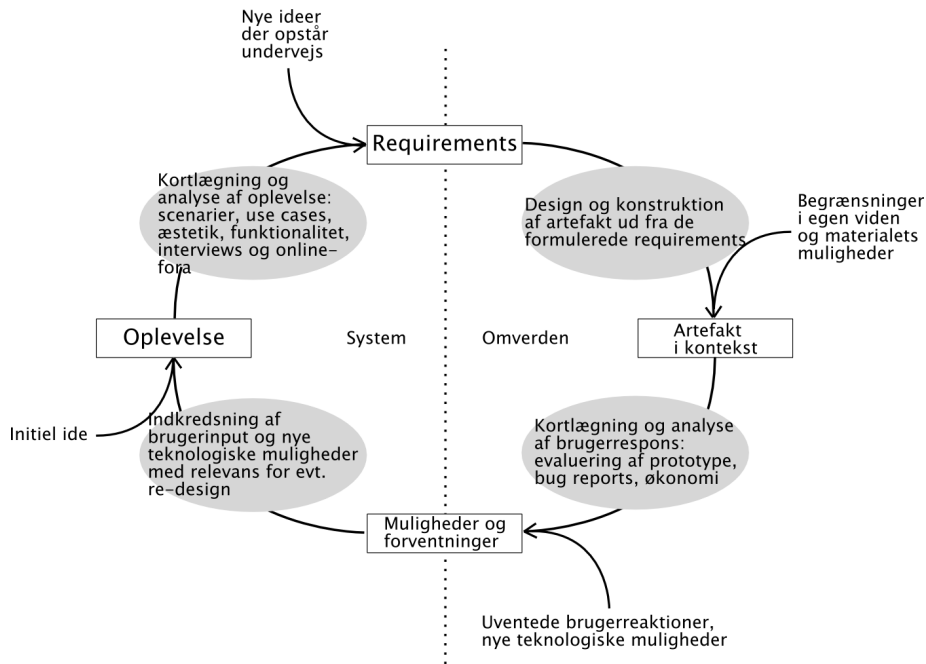


Fig. 6: De forskellige arbejdsfaser i metoden

Den første fase kendetegnes af kortlægning og analyse af *tasks* sammen med en udvælgelse af relevante og interessante *tasks* i form af f.eks. scenarier. Forud for scenarier er der abstraktioner og grundsten til koncepter (begreber som modifikation, rumlighed, osv.) hvorefter teknologiske farbare veje mod disse ideer udredes i en dialektisk proces mellem ide og materiale. Ad hoc opstår æstetiske og kompositoriske problemdomæner i forhold til omverden, der i specialet består af musikere med æstetiske og funktionelle sondringer, hvilket en omskrivning af *task* dækker over. Konkret laver vi et stort framework af påvirkningsfaktorer før vi formulerer *use cases* og scenarier.

Disse leder til formuleringen af *requirements*, der indsnævrer usikkerheden på hvad der skal designes og konstrueres. Her opererer vi med en muligheden for en forstyrrelse, der går forud for formuleringen af krav eller *requirements* til det artefakt, iterationen munder ud i. Denne forstyrrelse vedrører ny information internt i designsystemet – nye ideer, der opstår på baggrund af den forudgående iteration, eller i sammenhæng med den formulering af brugssituationer eller scenarier, som beskrivelsen af *tasks* indikerer. Når vi begriber det som forstyrrelse, er det fordi vi i så høj grad muligt lader informationen lede mod en revidering af de *requirements*, der var indeholdt i forrige iteration – et *increment* i forhold til interne designkriterier.

Design- og konstruktionsfasen følger på formuleringen af *requirements* og leder til et artefakt, der kan indsættes i en kontekst, hvadenten denne består af en konstrueret kontekst (som i

brugerevalueringer) eller den reelle brugssituation som kontekst for det endelige produkt. Denne fase er åben for forstyrrelser fra den anvendte teknologi – der kan gøre knuder og være svær at forstå. Teknologien kan forstyrre konceptet, designsystemet opererer med, der kan vise sig svært at realisere med den valgte teknologi, eller mindre plausibelt end forventet i den foregående analysefase.

Efter indsættelsen af artefaktet i en kontekst følger forskellige reaktioner og inputs fra brugere, hvad enten dette er i form af indsamlede interviews, svar på spørgeskemaer eller rapportering af *bugs* og andre reaktioner på et endeligt produkt. Også disse inputs fordrer en kortlægning og analyse. Og også her må designsystemet være åbent for forstyrrelser. F.eks. kan en forundersøgelse, der følger op på en brugertest af en prototype, være præget af designgruppens forudgående antagelser og dermed spørge efter særlige temaer og problematikker. Dermed er der en chance for at vi tematisk låser undersøgelsen i bestemte diskurser, fremfor at søge tematisk åbenhed og brugerinitiativer.

I så vid udstrækning muligt vil vi selv teste prototypen i felten og give den videre til alle involverede. Vi må i designgruppen være opmærksomme på svar, der peger i retning af uventede og uafdækkede problematikker, da brugere har stor indflydelse på designprocessen på dette punkt i iterationen. Dette fordrer en åben testform, hvori spontane brugerreaktioner registreres og sammen med efterfølgende interviews og samtaler danner grundlag for evaluering af prototypen.

Efter artefaktets indsættelse i en kontekst følger også muligheden for forstyrrelse fra andre steder end brugere og testere af designet. Det drejer sig især om muligheden for inddragelse af nye teknologier, der viser sig relevante og plausible for designprocessen. Valget af teknologi er nært forbundet med problematikker vedrørende kompatibilitet, standardisering, *entrenchment* og *intellectual property*, som vi vil vende tilbage til senere. Her vil vi blot anføre, at kortlægningen og udvælgelsen af teknologiske muligheder pga. disse problematikker opererer indenfor nogle begrænsninger, som designsystemet må medreflektere i sine operationer.

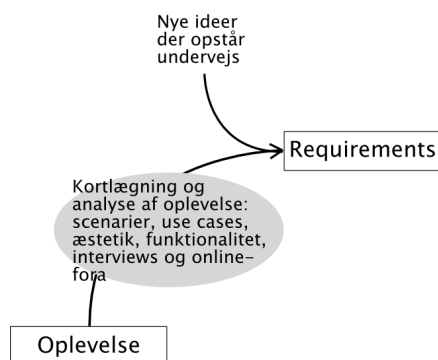
Tilsammen giver brugerreaktioner og teknologiske muligheder – efter en kortlægning og analyse af begge – en række kilder til videre reduktion af usikkerheden i designprocessen, og en konkret formulering af de designmæssige *muligheder* (*opportunities* i Carrolls model), der er for næste iteration. Disse muligheder er dog ofte bredere end nødvendigt for (re)formuleringen af den oplevelse eller aktivitet, designgruppen designer mod, og kræver en selektion af de teknologiske og konceptuelle muligheder, der er relevante for denne reformulering. Dermed er vi tilbage til den første fase i Carrolls model, beskrivelsen af *tasks*, som vi altså gentænker som oplevelse eller aktivitet. Den form for forstyrrelse, der her kan være med til at lede reformuleringen af den

ønskede aktivitet eller oplevelse på et andet spor, er for så vidt magen til den initiale forstyrrelse, der igangsætter designprocessen – nemlig en intern impuls bestående af gode ideer og kreative indfald hos designgruppen. Men derudover er denne forstyrrelse mulig at iagttage for designsystemet ud fra de erfaringer og den reduktion af usikkerhed, som kortlægning af brugeraktiviteter, formulering af *requirements*, konstruktion af en prototype og indsættelse af et artefakt i en kontekst, som den forudgående iteration har tilvejebragt i systemet.

Dette beskriver cyklussen, dens fire faser og muligheder for forstyrrelse i overordnede termer. Mellem Carrolls fire punkter eksisterer der med denne gentænkning fire overordnede aktiviteter, hhv. 1) kortlægning og analyse af en oplevelse eller aktivitet, 2) design og konstruktion af et artefakt, 3) kortlægning og analyse af brugerforventninger og teknologiske muligheder, og 4) selektion af de muligheder, der er relevante for reformuleringen af den oplevelse eller aktivitet, designarbejdet stiler imod. Vi vil nu se nærmere på hvilke konkrete aktiviteter, der er relevante og givende i hver fase og hermed beskrive forskellige arbejdsområder, designsystemet som informationel praksis omkredser.

De konkrete aktiviteter i de fire arbejdsfaser

Første fase



Den første fase er præget af kortlægning og analyse af den oplevelse, vi designer mod, med det formål at formulere en række forventninger eller *requirements* til den konkrete formgivning af softwaren. Dette sker gennem forskellige teoretiske og empiriske aktiviteter. Empirien omfatter interviews såvel som udsagn indsamlet på forskellige online-fora og fremanalysedes af os til designrelevant information. Ligeledes arbejder vi i denne fase med forskellige teoretiske inputs til formuleringen af *requirements*, hvilket vil sige at vi læser en række teoretikers udsagn om emner indenfor software-æstetik og –funktionalitet ud fra en specifik interesse, der går på at indsamle brugbar information til vores designproces. Vores læsning går derfor ikke mod en redegørelse for hver teoris helhed og intention, men er en selektiv læsning rettet mod en syntese af information ud

fra vores særlige designproblem. Kvalificeringen af en sådan læsning sker for det første gennem en refleksion over de forskellige teories samspil gennem opstillingen af modsætninger og kombinatorikker, og for det andet gennem en empirisk eftersøgning af tematikker med betydning for designprocessen. Kortlægningen er baseret på imødekommelsen af brugerforventninger og bygger derfor på empiriske udsagn. Samtidig indarbejder vi en teoretisk forståelse af den oplevelse, vi designer mod, hvilket sker gennem formulering af et begrebsligt framework. Formuleringen af frameworket leder videre til formuleringen af scenarier og *use cases*.

Tilnærmelse fordrer en dybde i såvel argumentation som refleksion. Dybden bygger på blotlægningen af vejen fra 'oplevelse' til 'requirements' – dvs. hvor konkret vi i denne iteration kan omformulere vage udsagn om oplevelsen til mere specifikke udsagn om den form, vi ønsker at give vores design. Nogle specifikke former for denne blotlæggelse er de førnævnte scenarier²⁹ og *use cases*³⁰. Et scenarie forstår Carroll som "*a concrete story about use*"³¹. Et scenarie udgør en narrativ beskrivelse af en (eventuelt fremtidig) brugssituation, der tjener til at gøre designet nærværende for brugere såvel som andre designere. Det enkelte scenarie er ikke tænkt som en udtømmende eller entydig bestemmelse af et design, men kan tjene to hovedformål:

- 1) at beskrive en eksisterende aktivitet eller oplevelse mhp. at understøtte den i software, eller,
- 2) at beskrive et design af et fremtidigt stykke software.

Som designsprog er scenariet altså funderet på designere og brugeres evne til at forstå og udtrykke sig i narrative forløb. Herunder at tænke en kontekstforståelse 'udenom' det specifikke scenarie og udfylde 'huller' i scenariets beskrivelse af en aktivitet eller oplevelse. På den måde gøres scenarier til dagligsproglige modelleringer af fremtidig brug, der er værdifulde som repræsentation af den kompleksitet, som brugeres mulige forventninger udgør.

Som et mere software-nært trin i kortlægningen af oplevelsen har vi *use cases*, der kan forstås som mere snævert og formelt definerede scenarier. En *use case* er nært forbundet med det konkrete software-systems struktur og processer, og vil derfor være formuleret med softwarens objekter og relationer in mente. Relationen til softwarens underliggende arkitektur og *objektsystem*³² er mere eller mindre eksplicit, og *use cases* udgør dermed en mulig vej fra scenarier bygget på empirisk og teoretisk grundlag til en konkret formgivning af softwaren. Men stadig indgår der også en række faktorer udover den konkrete brug som *use cases* ekspliciterer – f.eks. den æstetik, softwaren

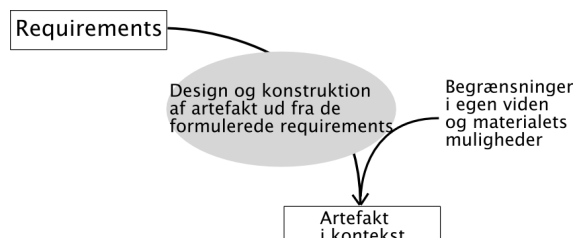
²⁹ Carroll, 2000

³⁰ Booch, Jacobson & Rumbaugh, 1999

³¹ Carroll, 2000, p. 14

bygger på, og de mere u håndgribelige forventninger, musikere vil have til at se softwaren som instrument. Derfor arbejder vi også mod at kortlægge det, der ikke umiddelbart kan beskrives ved scenarier og *use cases*, nemlig oplevelsens og instrumentets æstetik.

Anden fase



Vejen fra *requirements* til et artefakt, der kan indsættes i en kontekst går over design og konstruktion. I tidlige iterationer repræsenteres artefaktet ved en prototype, der udgør en hypotese om vores design og dermed fungerer som det primære artefakt, vi internt i designgruppen kommunikerer omkring og diskuterer æstetik, interaktion, teknologi osv. ud fra. Formålet med prototypen er at afprøve denne hypotese i en kontekst mhp. videre redefinerings af dens indhold og formål. Dette betyder at den oplevelse eller information, vi som designgruppe ønsker at formidle, er implementeret i en mere konkret form mhp. på at lære af test-brugeres konkrete reaktioner og yderligere forventninger. Meningen med prototyping er at skabe en form, der repræsenterer designets endelige form uden at inddrage alle dens elementer eller fulde interaktionspotentiale. Dermed gives vores testere et tilstrækkeligt grundlag at vurdere designets muligheder for interaktion, dets æstetik og dets relevans for deres arbejdsprocesser, uden at vi implementerer et fuldt system, og dermed investerer tid i et design, der ikke er valideret som relevant for brugere i domænet 'musikere, der anvender computere.'

Dermed afprøves konkrete designs i en form, der giver test-brugere bedre muligheder for at forestille sig det endelige designs form, og samtidig komme med uventede indsigelser til designet.

Software-teoretikeren Christiane Floyd skelner mellem tre overordnede prototyping-former, hhv. *eksplorativ*, *eksperimentel* og *evolutionær* prototyping³³. Brugen af eksplorativ prototyping søger at involvere brugerforventninger aktivt i designprocessen, og bygger igen på kommunikation, der understøtter gensidig læring³⁴. Dette indebærer, at designgruppen lærer om brugernes forventninger, men også at brugergruppen lærer om de teknologiske muligheder, og derigennem opnår større mulighed for at formulere nye forventninger, der kan realiseres. Eksplorativ

³² Mathiassen, m.fl., 1997, p. 50

³³ Floyd, 1984, p. 6-12

prototyping har således karakter af samtidig udforskning af muligheder og forventninger. Teoretisk udlægger vi dette som eksperimenterende formforstyrrelser i processen (*re-entry*). *Re-entry* beskriver her hvordan muligheder genindføres i allerede etablerede muligheder, hvorved der opstår differencer eller *increments* i formen gennem eksperimenter.

Derimod lægger eksperimentel prototyping sig nærmere op ad, hvad der indenfor ingeniørvidenskab traditionelt forstås ved en prototype, nemlig den første udgave af et konkret design. Her kendes målet for designet allerede, mens den konkrete udformning afprøves i en eksperimentel proces, der involverer test-brugere³⁵. Dette knytter vi til den modellering af fremtidig brug, som anvendelsen af scenarier indebærer, og som kommer til udtryk i Carrolls *task-artifact cycle*.

Endelig indebærer evolutionær prototyping en kontinuerlig udvikling og versionering af et softwaresystem, hvorved grænsen mellem prototype og endeligt system bliver mere udflydende.

Adskillelsen mellem disse tre former er ikke stringent, og vores prototyping-proces låner også elementer fra såvel eksplorativ som eksperimentel prototyping. Det eksplorative består i at lade selve prototypen repræsentere en hypotese om et fremtidigt design, der afprøves blandt konkrete testbrugere. Prototypens form er i hver iteration allerede skitseret ud fra de *requirements*, der er indsamlet i første fase. Derfor kan vi ikke i testsituationen tilpasse prototypen til brugerforventninger frit, men vi må basere testen på den form, vi har etableret. Dermed bærer vores prototyping-proces også karakter af eksperiment. En måde, hvorpå vi dog kan komme denne problematik imøde, er at arbejde med mere end én prototype, hvilket igen bygger på empirisk iagttagelse af flere forskellige brugsmønstre, der senere formuleres som scenarier og *use cases*. Vi arbejder iterativt med et sæt af prototyper, hvorved vi i testsituationen har mulighed for at præsentere testbrugerne for mere end én konkret formgivning, funktionalitet og æstetik.

Dog kan man ikke karakterisere vores prototyping-proces som evolutionær, på trods af dens indlejring i en iterativ designmetode. Dette fordi en udvikling gennem versionering kræver at software-redskaberne til prototyping (3D-engine og scripting-sprog i vores tilfælde) også anvendes til de versioner, der kommer efter prototypen. Grundet vores valg af software-redskaber forekommer dette ikke realistisk, da de konkret udvalgte redskaber ganske vist er lettere at arbejde med, men også 'tungere' for computeren at køre. En version udover prototyping-stadiet vil således fordrer en videre udvikling i mere produktionsorienterede udviklingsmiljøer. Dette stemmer også

³⁴ Ibid., p. 6-7

³⁵ Ibid., p. 8

godt overens med den eksplorative og eksperimentelle prototypings *quick & dirty* karakter: Det drejer sig om at konstruere en nogenlunde funktionel og æstetisk fungerende skitse af et fremtidigt stykke software, uden at dette har for store tidsomkostninger. Dermed kan vi tillade os kontinuerligt at smide de skitser væk som ikke virker, og i stedet fokusere på prototypen som et middel til erkendelse af udtalte og uudtalte brugerforventninger.

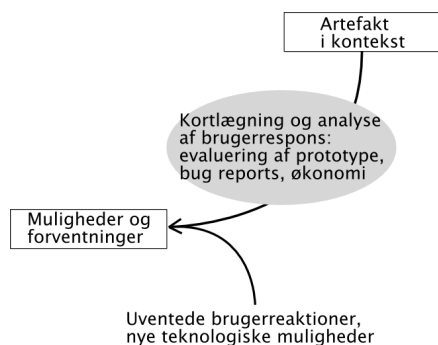
Samtidig medfører konstruktionsprocessen en vekselvirkning mellem det digitale materiale og vores designkompetencer. Begrænsninger i disse fungerer som forstyrrelser, der kan imødekommes gennem indhentning af yderligere viden om f.eks. teknologi, interaktionsformer og æstetik. Det artefakt, som anden fase munder ud i, er altså ikke blot en realisering af de konkrete *requirements*, men også en form designet under de begrænsninger, som vi møder i den konkrete konstruktionsproces. Donald A. Schön, der forsker i design og designviden, karakteriserer denne proces som designerens *refleksive konversation* med de forhåndenværende *materialer*³⁶. Materialet ser vi som 'det digitale materiale', dvs. mulighederne og barriererne for implementering af en given proces på en computer. Disse muligheder og begrænsninger er givet ved de specifikke begrænsninger ved de valgte værktøjer: 3D-engine og scripting-sprog. Dermed afhænger materialets brug ikke blot af computeren som sådan, men også af de valg og fravalg af specifik funktionalitet, som den enkelte grundteknologi er designet ud fra. Dog determinerer valget af grundteknologi ikke entydigt mulighederne for designet, da kvalifikationer til at bruge denne også skal være til stede i designgruppen. Vurderingen af muligheder og begrænsninger for designet indebærer dermed en refleksion over forholdet mellem vores kvalifikationer og relevant teknologi, der kan afføde behov for indhentning af nye kvalifikationer. Schöns konversations-perspektiv illustrerer, at teknologi og kvalifikationer ikke altid følges ad, men at de to gang på gang må mødes i den specifikke designsituation.

I anden fase må vi imidlertid lægge os fast på et konkret sæt prototyping-redskaber eller udviklingsmiljø for overhovedet at have et digitalt materiale for hånden. Dette indebærer et valg og samtidigt et fravalg. Vi vil i delen *Prototype* bl.a. se nærmere på dette konkrete valg af teknologi og dets konsekvenser for designprocessen. Her vil vi starte med at anføre, at vi dermed må udbygge vores kompetencer undervejs i designprocessen, bl.a. ved at søge hjælp og informationer i manualer, *tutorials* og onlinefora. Dette indebærer en tidsmæssig investering i bestemte teknologier, der kan være ganske substantiel. Hvor den enkelte prototype kan være forholdsvis let at smide ud, er det derimod sværere at udskifte hele det teknologiske grundlag for konstruktion af

³⁶ Schön, 1992, p. 9-11

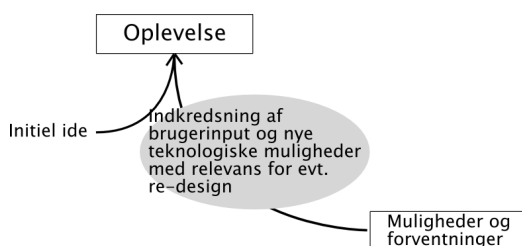
prototyperne. Som sagt bliver dette dog nødvendigt i overgangen fra prototype til endelige versioner af softwaren.

Tredje fase



I den tredje fase kortlægger og analyserer vi den brugerrespons, der kommer ud af afprøvningen af prototypen som en hypotese om et design. Samtidig må vi her være åbne for nye teknologiske muligheder i den valgte grundteknologi, der kommer til syne på baggrund af brugerrespons. Dette leder til formuleringen af muligheder og forventninger til revideringen af designet. I specialet behandler vi afprøvningen af designet i delen *Evaluering*, der også omfatter en refleksion over metodens egen anvendelighed. I første omgang kan prototype-evalueringen struktureres ud fra de opstillede *requirements* og de spørgsmål, vi umiddelbart kan udlede af disse. Men som tidligere nævnt må vi samtidig være åbne for responser, der ikke falder ind under disse, og som f.eks. kan pege på helt andre brugssituationer eller andre forventninger til designet. Kortlægningen og analysen handler dermed om det udtalte såvel som det uudtalte i testen af prototypen. Begge dele kan gøres eksplicitte med reference til de tematikker, som designprocessen opererer omkring, og griber således tilbage til den foregående kortlægning og analyse i første fase. Men samtidig er det også i tredje fase at nye vekselvirkninger i designproblemet kan blive relevante ud fra den nye feedback, som evaluering af prototype-testen (og *bug reports*, etc. i den endelige version) medfører. Det er altså i tredje fase, at designprocessen i væsentlig grad får sin retning ud fra ventede og uventede brugerinputs, mens første og anden fase nærmere er præget af analyse og syntese på designsystemets præmisser.

Fjerde fase



Endelig arbejder vi os i iterationens sidste fase tilbage mod et udgangspunkt for revidering og raffinering af designet: Nemlig gennem selektion i de problematikker, nye forventninger og forslag til forbedringer, der forhåbentligt er resultatet af prototype-evalueringen. Selektionen udgør en vej tilbage mod en afgrænsning af designproblemet, således at vi ved næste *increment* i iterationsprocessen står med et snævrere defineret designproblem, end vi umiddelbart ville gøre ved afslutningen af prototype-evalueringen. Redefineringen af designproblemet bliver dermed til en redefinering af den specifikke oplevelse vi designer mod, fremfor et vagere designproblem bestående af mange divergerende brugerforventninger. Endelig er vi i fjerde fase åbne for forstyrrelser i retning af selve definitionen af denne oplevelse; dvs. indtryk udefra, der peger i retning af en anden kerneoplevelse. På samme vis som den initielle ide danner redefinitioner grundlag for den videre formulering af en oplevelse.

Efter denne meget konkrete gennemgang af metodens arkitektur vil vi nu se på en overordnet sontring i brugen af metoden. Næste afsnit vil derfor være et refleksivt afsnit om systeminterne kriterier for metodens form og perspektiv.

Stringente og ikke-stringente designprocesser

Vi vil her se på den metode, vi anlægger for vores designproces, i et bredere perspektiv, nærmere bestemt som en metode, der er fastlagt indenfor en vekselvirkning mellem stringente og ikke-stringente designformer. Dette leder frem mod en diskussion af metodens rationale og perspektiv – en søgen efter nuancering i tilnærmelsen til en form.

Navigation i gradueringen af stringens

Gradueringen fra stringent til ikke-stringent kan beskrives i henhold til graden og karakteren af den åbenhed, designsystemet har i forhold til sin omverden: I stringente designprocesser forudsættes det i høj grad at det endelige mål for designprocessen er muligt at definere på forhånd, og derfor også mere eller mindre uafhængigt af ekstern påvirkning (undtagen i en evalueringssituation, hvor det på forhånd fastdefinerede design valideres i forhold til internt fastlagte kriterier). Derfor kan en stringent tilgang operere ud fra den utopiske forudsætning, at det er muligt at 'gøre det rigtigt første gang'. En sådan tankegang kommer bl.a. til udtryk i en software-metodik som vandfaldsmodellen, der foreskriver en lineær proces fra ide til endeligt produkt. Et eksplicit rationale for denne tankegang er, at ændringer og backtracking koster mere, jo længere henne i den lineære proces, designsystemet befinder sig. Dette hænger sammen med den forankring i økonomiske og administrative rammer, som de fleste designsystemer er underlagt som forudgående betingelser for

deres eksistens. Et sådant designsystem er primært åbent overfor forstyrrelser og restriktioner i organisationen internt, og kun i mindre grad åbent for eksterne påvirkninger.

En mindre stringent designproces kan f.eks. karakteriseres ved iterative metodikker, og den mulighed for feedback fra tidligere til aktuelle iterationer, som de foreskriver. Hermed anerkendes det internt i designsystemet, at det måske ikke er muligt at 'gøre det rigtigt første gang', men at en gradvis raffinering og redefinering er ønskværdig for at opnå et bedre design. Denne vej er ikke-lineær, da tidslighed tit er opløst i forhold til konkrete arbejdsområder. Det informationsmæssige input til en sådan designproces bliver dermed lige så vigtigt som anerkendelse af de organisatoriske rammer, som designsystemet er forankret i. Og dermed kræver designprocessen en højere grad af åbenhed overfor dets omverden (udover dets organisation) og muligheden for forstyrrelse af forudfattede designkriterier, antagelser og kategorier. Karakteren af denne åbenhed ændres i retning af en vedblivende inddragelse af påvirkninger udefra gennem hele designprocessen, fremfor udelukkende i den indledende og afsluttende kontakt med omverdenen. Dette gør designprocessen mere påvirkelig og 'plastisk', og kan gennem iterationer føre til at større mængder tid og penge bruges på analyse og evaluering, end i mere stringente designprocesser. Dermed bliver designprocessen også sværere at håndtere for en organisation, da dette sker med økonomiske og administrative kriterier. Proceshåndtering er ikke desto mindre nødvendig, hvilket kræver en større refleksiv kapacitet i det mindre stringente designsystem. Det er f.eks. i ikke-stringente processer nødvendigt at kunne afgøre, hvornår der er foretaget tilstrækkelige iterationer, om evalueringer og analyser har været givende nok i forhold til en eventuel næste iteration, osv. På denne vis kan en mindre stringent tilgang få lov at få karakter af eksperiment eller udforskning, uden at hensyn til tidsforbrug, osv. slippes af syne.

Gradueringen mellem stringens og ikke-stringens indebærer dermed en graduering mellem regelbaseret udvikling og udvikling baseret på tilbagevendende refleksion. Dette kan formuleres på den måde, at regelbaseret, stringent design forudsætter at en designmetode er kontekstafhængig, dvs. at den kan genbruges umiddelbart fra ét designproblem eller udviklingsprojekt til det næste. Også på anden vis forekommer det stringente design at indebære en kontekstafhængighed: Designet indsættes ved designprocessens afslutning i dets tiltænkte brugssituation, men muligheden for feedback indtænkes sjældent i denne indsættelse, der snarere får karakter af overdragelse af et artefakt.

Derimod hænger det, vi her har kaldt en mindre stringent designform, sammen med en stigende kontekstafhængighed. Både hvad angår muligheden for at genbruge metoder, såvel som forholdet mellem designsystem og omverden. Metoder kan ikke umiddelbart genbruges, men må gentænkes i forhold til den specifikke designproces. Fordi designproces, designmetode og artefakt

gensidigt påvirker hinanden. Ligeledes muliggør en mere åben, mindre stringent designform en større gensidig afhængighed mellem designsystem og omverden. Forstået på den måde, at designsystemet i højere grad orienterer sig mod uventede forstyrrelser, og ikke blot be- eller afkræftelse af egne hypoteser om designets værdi.

Et rationale for designmetoden

Rationalet bag vores designmetode må forstås som en vekselvirkning mellem stringens og ikke-stringens. Domænet for adaptation og innovation i designet af software-instrumenter er sværere at definere end for f.eks. kontor-software. Dette hænger sammen med at sådant software retter sig mod etablerede arbejdsgange, mens brugen af musiksoftware er underlagt et behov for eksperimenteren og egen determinering af ønskværdige arbejdsgange hos den enkelte musiker. Dermed bliver det sværere for os at fastlægge rammerne for designets koncept, teknologi og æstetik på forhånd. Det er mere plausibelt at definere designets koncept og realisering gennem en stadig vekselvirkning mellem egne ideer og påvirkninger udefra. På den anden side må vi fastholde designprocessen og dens omfang i tid gennem en stadig refleksion. Det er en refleksion over relevansen og værdien af de aktiviteter, vi foranstalter i hver fase af en iteration. Dette står i modsætning til en helt og holden stringent designproces, hvor et sådant behov for refleksion er erstattet af en på forhånd fastlagt skematik for designprocessens aktiviteter og deres indbyrdes værdi.

Nuancering, tilnærmelse og kombinatorik

Et gennemgående element i vores tænkning om såvel design som metode er en sondring og navigation mellem poler eller modparter, f.eks. brugercentreret/designercentreret, æstetik/funktionalitet, forventning/overraskelse, osv. Dette afspejler ikke en lyst til at inddrage alt i sort og hvidt, men tværtimod en tendens mod nødvendige nuanceringer, der bygger på en forståelse af disse poler som yderpunkter i et spektrum, hvori der findes mange frugtbare erfaringer og informationer indenfor en lang række områder. Vores udgangspunkt i Spencer Brown, Borch og Luhmann kan på sin vis ses i sammenhæng med denne tendens. Polariseringerne indikerer umiddelbart differentieringer i såvel designets som metodens form, og nødvendiggør rationale for disse. Et gennemgående rationale er ønsket om nuancering af de valg og fravalg, der foretages, og dette afstedkommer igen vores ønske om at åbne designprocessen for en så høj grad af designrelevant information som muligt. Nuancering tjener til redefinerende af designet og gentænkning af metoden på baggrund af designprocessen som *case*. Dermed tegner der sig også en tendens mod gradvis tilnærmelse mod et design med en mere reflekteret metode. Forandringen i designet kan vi umiddelbart erkende som bedre eller dårligere gennem test af prototypen, mens

forandringen i metoden må erkendes gennem refleksion over de inddragne teoriers anvendelighed som fundament.

Tankeformen bygger på identifikation og kombination af polariseringer, hvorigennem der opbygges et potentiale for refleksion og nytænkning af såvel design som metode. Kombinatorikken præger designet såvel som metoden og tenderer mod en refleksiv eklekticisme. Vores mål med kombinatorikken er imidlertid ikke at tage 'det bedste fra to verdener', men gennem kombinationen at nytænke relationerne mellem de enheder, vi kan identificere i såvel design som metode. Eklekticismen i såvel design som metode begrundes i en løbende eftersøgning af et rationale for værdien af de konkrete kombinatorikker, vi indarbejder.

Undervejs synes det af og til at vi med lysten til kombinationer løber ind i paradokser. F.eks. kan det forekomme paradoksalt at vi forsøger at designe et potentiale for interaktion, der på én gang giver musikerens mulighed for kontrol, men samtidig tillader musikerens at slippe tøjlerne og lade softwaren operere autonomt. Men trænger vi nærmere ind i dette modsætningsforhold ser vi – bl.a. med grundlag i egne erfaringer, teori og empiri – at begge momenter kan samvirke i et design som en vekselvirkning mellem *glitch* og kontrol (dette behandles nærmere i afsnittet *Æstetik* i kapitlet *Design*).

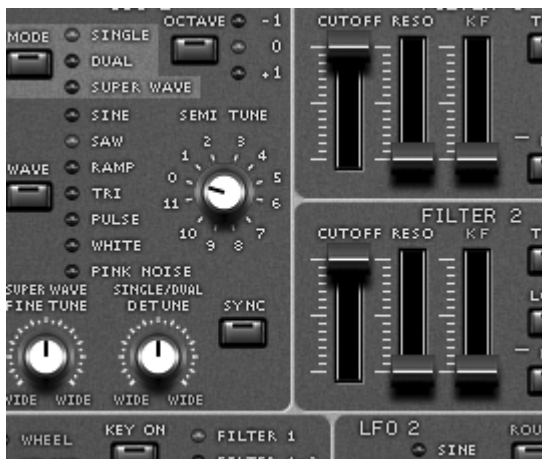
Den løbende kombinatorik af mange polariseringer tjener til opbygningen af en teoretisk og empirisk dybde bag de forskellige kombinatorikker. Dette giver både os selv og andre en mulighed for at se os i kortene og gør, at design og metode opbygges omkring information samlet gennem argumentation. Med dette ser vi tendensen til eklekticisme og rekombination kvalificeret gennem refleksion.

Design

Softwareinstrumenter

Vi starter designprocessens opbygning af en informationsbasis med at se på designet af andre software-instrumenter, og de paradigmer for design, der ligger bag dem. Designet af instrumenter indebærer et forhold mellem interface og lydkilde, hvor interfacet muliggør interaktion med en matematisk proces. Denne bagvedliggende proces vender vi tilbage til under behandlingen af *Funktionalitet og modificering*, der beskriver grundlaget for de radikale lydmuligheder, der er tilgængelige gennem forskellige lyd-algoritmer. Her vil vi kort beskrive de 'indpakninger' af disse algoritmer, der er gængse, og deres rod i akademiske tankegange.

At processen bagved softwarens interface er matematisk kommer umiddelbart til udtryk i 1) kommando- eller programmerings-baserede interfaces, og 2) *knobs and sliders* interfaces, der er implementeret i f.eks. mange VST-plugins. I begge tilfælde opererer vi med en direkte manipulation af talværdier, der igen får soniske konsekvenser. I *knobs and sliders* interfaces er dette implementeret gennem grafiske elementer, der hver især er direkte bundet til separate parametre for lyd-algoritmen:



Interaktionen med det musiske materiale kan dog udvides mod en repræsentation, hvor instrumentet indebærer en mere abstrakt oversættelse fra grafiske til soniske former. Historisk er dette bl.a. forsøgt af komponisten Iannis Xenakis, der med sit UPIC-system kobled linjer og andre basale former til musiske strukturer³⁷. Ideen er at gøre algoritmer og musiske strukturer

³⁷ Chadabe, 1997, p. 213ff

tilgængelige gennem ukonventionelle interfaces. Ifølge Xenakis fordi 'alle kan tegne'³⁸ og dermed interagere med et interface, der ikke baserer sig på matematiske eller musiske konventioner.

Som historik sker der dermed en udvikling i metaforikken. Et tal går fra at være repræsenteret ved et tal (i kommando-baserede interfaces) til en repræsentation ved en *slider*, og endelig en repræsentation som punkt på en linje. Tanken bag at omskrive interfacet fra tal til grafik bygger på et ønske om kompleksitetsreducing. Konsekvensen er imidlertid, at den grafiske kompleksitet stiger samtidig med denne omskrivning, bl.a. fordi der ikke findes noget entydigt grafisk sprog, hvilket åbner op for brugerens fortolkning af repræsentationen. Interaktionen ændrer sig også i takt med at kompleksiteten flyttes fra et matematisk til et æstetisk-grafisk interface.

Alternative repræsentationer er muliggjort gennem ideen om *mapping*, der beskriver en omsættelse fra f.eks. et grafisk input-domæne til et sonisk output-domæne gennem etablering af relationer. *Mapping* bliver en adgang til tekstur og strukturs manipulerbare egenskaber. Dermed skabes også en relation mellem musisk proces og musisk materiale. I et fleksibelt input-domæne er mange former mulige, men nogle undgår måske musikerens opmærksomhed. Komponisten John Cage søgte at overkomme dette ved at involvere tilfældigheder i selve musikkens tilblivelse³⁹, noget vi videre behandler som *glitch*. Dermed repræsenteres det soniske materiales muligheder i et rum for interaktion, der er afgrænset af musikerens udforskning fremfor musiske konventioner. Og ligesom lydmulighederne ændres med valget af algoritme eller plugin, ændres lydmulighederne også gennem valget af interface eller repræsentation. Mulighederne i instrumentets input-domæne kan f.eks. udvides gennem et potentiale for tilfældigheder som Cage foreslår, og gennem en mere umiddelbar grafisk formgivning, som Xenakis opererer med. Indenfor disse paradigmer vil vi forsøge at udvide vante, etablerede arbejdsgange med nye interaktionsformer, der giver muligheder for nye musiske former.

Akademiske traditioner er dog kun ét sted for indkredsning af etablerede praksisser såvel som muligheder for designet. Et andet relevant sted er *lead users* og *lead developers*, hvis udsagn vi vil se på gennem interviews. Dette er emnet for de næste to afsnit, hvor vi først præsenterer fremgangsmåden for denne involvering af empiri, og derefter dens resultater.

Empiri – metode

³⁸ Ibid., p. 215

³⁹ Hamman, 2002, p. 12

Specialets empiriske udgangspunkt, der er en del af metodens første og tredje fase, vil i det følgende afsnit behandles.

Information fra omverden kan gennem analyse indarbejdes i specialets prototype, hvilket er en teoretisk antagelse. Denne antagelse afprøver vi med en empirisk metode der er rettet mod to primære grupper. Grupperne ser vi som informationskilder, hvor der forefindes designrelevant information, der tilgodeser to forskellige 'informationspotentialer'.

Det første potentiale er indarbejdelse af (udenforstående) information i konceptet, hvor andre musikere, designere eller konceptudviklere kan forstyrre processen, hvori en ide konkretiseres og bliver en form. På en tidslinje er denne undersøgelse foretaget før prototypekonstruktionen påbegyndes, men ideen er fikseret tilstrækkeligt til at vi diskutere den med andre som et koncept.

Det andet potentiale er indarbejdelse af information fra brugere i prototypen, hvor testsituationer kan forstyrre prototypens form i kraft af at nye muligheder opdages eller mangler blotlægges. Tidsmæssigt ligger denne aktivitet sideløbende med at prototypen bygges og færdiggøres. Dette er empiriens rolle i metodens tredje fase, hvilket vi vil behandle i kapitlet *Evaluering*.

I metodens første fase er vores empiriske metode rettet mod hhv. en gruppe, der kan betegnes *andre udviklere af musiksoftware*, og en gruppe, der kan afgrænses *andre brugere af musiksoftware*.

For at holde en nogenlunde tematisk stringens i undersøgelsen har vi taget udgangspunkt i temaer der berører begge faser såvel som begge grupper. Dog på forskellige sider af designets fiksering, da designere konstruerer ud fra disse temaer og brugere oplever/anvender temaerne i denne konstruktion.

Overordnede temaer

- *At åbne komposition for muligheder - hvordan sker dette i praksis?*

Vi vil gerne vide hvilke applikationer, der anvendes til at give slip på kompositionen. Herunder hvor hyppigt de bruges, og på hvilke områder inden for musikalsk struktur / tekstur. Spørgsmålet eftersøger musikerens midler til modifikation og vedkommendes tanker omkring potentialet i modifikation. Spørgsmålet er relateret til improvisation som et kernepunkt for musikalsk ekspansion eller kollaps – hvordan musikken udvides i nuet og med hvilke begrundelser og midler.

- *Forhold til generelle arbejdsprocesser med musik – hvilke arbejdsmønstre er tydelige?*

Hvis der findes et ønske om at ændre disse mønstre, må dette ønske også have en retning, og i et eller andet omfang afspejle de nuværende brugsmønstre. Heri ligger også et svar på, hvordan processerne er etableret; f.eks. gennem gentagelse, eksperimenterende afprøvninger, mv. Dette ser vi teoretisk som et møde mellem referentielle og operationelle kriterier. Når der spørges til procesetablering er det konkret på området 'faser og kontinuum'. Her kan den interviewede afgrænse, hvad der for vedkommende er højt-lavt, meget modificeret-lidt modificeret, osv. Ligeledes findes der i musisk praksis et skel mellem at forberede sig på at lave musik og spille koncert – den faktiske udøvelse. Dette skel behandler vi senere som et skel, der vedrører faser som prækomposition⁴⁰, komposition og performance. Vi har oversat forholdet fra engelsk, hvor det også benævnes 'setup and execution'. Her refererer 'setup' til prækomposition, og 'execution' til såvel komposition som performance.

At lave musik anses som en proces, hvor det kan ekspliciteres, hvad der blokerer eller letter vejen fra ide til produkt/lyd, og hvorfor, og i hvilken kontekst.

- *Forhold til arbejdsprocesser indenfor 3D domæner – hvad giver visuelt mening?*

Igen et spørgsmål til brugsmønstre, men her specifikt i relation til 3D-brugerflader i musiksoftware. På dette punkt er det vigtigt for os at vide, hvordan et forhold mellem eksempelvis dybde-cues i billedet og lyden etableres for musikeren. Videre forholdet mellem 'setup og execution', hvori der ligger overvejelser af f.eks. systembelastning i forhold til det musikalske udbytte.

- *Ergonomiske overvejelser og forhold – hvad placeres hvor, for at lave hvad og hvorfor?*

Her spørger vi til 'det gode setup' mhp. ergonomiske præferencer. Viden om hvordan et studie ser ud kan tegne et bredt spektrum af mangfoldige setups, hvor to sjældent er ens. Vi vil med dette spørgsmål kortlægge kontinua mellem 'stort setup', 'minimalt setup', 'eksperimenterende setup', hvilket er individuelle konstruktioner. Dette gør vi for at finde klasser som det *almindelige* eller det *gennemsnitlige* setups form, forstået som det, der ofte observeres. Konkret spørger vi om der arbejdes primært med mus/tastatur, eller andre enheder som f.eks. keyboards, sequencere og controllere. På skærmen findes der også ergonomiske præferencer, hvilket eksempelvis er layouts af programmer, hvordan disse placeres i forhold til hinanden, og i hvilke situationer. I alle nævnte præferencer ligger der implicit den interviewedes input på hvad der skaber gode resultater, og i hvilke sammenhænge.

- *Gamer relationer – spiller du?*

⁴⁰ Jf. <http://en.wikipedia.org/wiki/Precompositional>

Dette er et krydsrefererende spørgsmål i forhold til temaet interaktionsformer. Erfaringer i spil, dvs. interaktion med mus/tastatur i forhold til grafik kan anvendes i forhold til valg af interaktionsform, eller mapping af muligheder. Her spørger vi konkret til disse erfaringer, og om disse har en betydning for præferencer af layout, interface, mv. Videre findes der store grupper af online gamere, hvori viden og community i og omkring gaming flyder frit over større portaler, fora eller nyhedsgrupper, hvilket vi ser som interessante medier. Dette community ligner i høj grad de online fællesskaber, der er omkring musikproduktion og herunder softwareudvikling. Derfor er det ligeledes relevant at spørge til online aktiviteter som gamer.

- *Grad af tilpasning i forhold til forskellige miljøer (hosts/shells: Buzz, Cubase, Logic, PD, MAX, Reaktor) - Hvad gør en bruger/designer for at adaptere forskellige miljøer i enten design eller musikproduktion?*

Kombinatorikken mellem programmer, og den viden kombinatorikken påkalder og resulterer i, ser vi videre som designrelevant information på et fundamentalt plan. Konkret spørger vi til hvad en bruger eller en designer vil gå igennem af f.eks. tillæring af kodeegenskaber for at skabe et 1:1 forhold mellem ide og materiale. Videre om disse processer hæmmer eller gavner musikken eller programmet. På dette punkt spørger vi også til systemspecifikationer mhp. opgraderinger eller afinstallationer, og ud fra hvilke kriterier hvilken handling vælges. Dette vil også være en bruger/designers svar på sværhedsgraden af opsætning, og hvilken komponent-kommunikation, der vælges og sættes op.

- *Forventninger til udvikling – hvordan ser din fremtid ud, hvilke tendenser ser du?*

Vi er interesserede i ambitioner, både på softwareniveau og på et musikalsk niveau. Stort set alle har en ide om, hvad der mangler før perfektion er opnået, hvilket er en stræben vi kan konstruere ud fra. På dette punkt er alle vilde ideer velkomne på områder som: retning for GUI-evolution, modifikation, nye komponenter, æstetik, mv. Tendenser er et nøgleord i forhold til hvilke dele af udviklingen, der enten skal stabiliseres og etableres som standarder, eller destabiliseres og skrottes. Dette hænger sammen med brugermønstre, og hvilke værktøjer, der understøtter disse mønstre. Heri ligger også brugerens/designerens sondringer omkring GUI æstetik og hvordan denne kædes til funktionalitet.

Metoden bag empiriindsamlingen

Mængden af interviews er forholdsvis stor, og de er foretaget i forskellige situationer, hvilket har tilvejebragt store mængder data og en diffus metode. Metoden er dog sammenholdt af tematisk stringens, som er designrelevante diskurser. Der er i specialet ingen foretrukken eller stringent empirisk metode som f.eks. kvalitative fremfor kvantitative undersøgelser, da diskurser og temaer

er vægtet højere end metodisk entydighed, og naturligt berører begge undersøgelsesmetoder. Vi har nærmere søgt at etablere komplementære forhold mellem kvalitative og kvantitative metoder, f.eks. ved at krydstabulere temaer i både kvalitative og kvantitative situationer. Vi mener at et kombinatorisk udgangspunkt tilvejebringer en større mængde designrelevant information end en enkelt metode ville have gjort isoleret. F.eks. kan nye temaer i kvalitative situationer kan udvide fundamentet for kvantitative undersøgelser, hvor et overraskende svar kan blive et nyt tema eller et nyt problemområde, der påkalder en videre kvalitativ analyse. Hvis vi vil undersøge, hvor mange der er berørt af det specifikke problem, eller hvis vi ønsker at reproducere problemet hos flere brugere, er det relevant at undersøge temaet med kvantitative metoder, hvilket er et eksempel på metodisk komplementære forhold. I denne kombinatoriske bevægelse har vi som filosofi at gensidige (metodiske) forstyrrelser skaber et større potentiale i form af mere designrelevant information, der går gennem et højere antal selektionsled. Vi har anvendt hybridmetoder, hvor et lignende komplementært forhold findes i konceptudviklingen. F.eks. kombineres forundersøgelser med kvalitative interviews, der iterativt tilvejebringer nye temaer. Vi har ad hoc fokuseret vores metode, og indpasser den til de forskellige situationer og formål.

Kvalitative metoder

Interviews er et udgangspunkt for at forevige personlige og dybdegående fortolkninger af temaer. Disse tager tid, både planlægningen bag og udførelsen af dem. Det har derfor været et bevidst valg at reducere tiden, hvor data behandles, ved at foretage en kondensering af interviewene. En transskription ville være givende, hvis vi søgte på udsagn og forhold på et ordniveau, hvor ords placering i sætninger var af betydning. Dette er ikke tilfældet, da vi søger meninger, tendenser og dermed generelle udsagn, der kan udvindes af en mindre (tids) krævende kondensering.

I situationerne er der anvendt en mikrofon og en laptop. Vi har vurderet, at selvom vi spørger til ergonomi og dermed fysiske vaner, er video unødvendigt for at gøre betragtningerne til designrelevant information. Vi har i specialet også anvendt interviews, vi ikke selv har foretaget, som inspiration; eksempelvis fra Berklee⁴¹, hvor ressourcepersoner udtaler sig om temaer, der vedrører specialet, så som improvisation, musikalsk struktur, musikalitet, osv. En anden kilde er en række *developer interviews* på online-forummet *Jack Dark*⁴², hvorfra vi eksplicit inddrager en række citater fra udviklere af VST-plugins, der bl.a. handler om designkriterier og forventninger til fremtidige udviklinger.

⁴¹ <http://www.berklee.edu/>

⁴² <http://jackdark.proboards39.com/>

Det har ikke været hverken muligt eller nødvendigt for os at lave disse interviews, da de allerede tematisk er relateret til vores udviklingsproces. Vi har med andre ord også anvendt online kvalitative ressourcer, som vi ikke direkte er i kontakt med. Vi anvender disse kilder som et bidrag til vores empiriske framework, og har valideret kilderne efter ophav (f.eks. Berklee eller andre institutioner) eller designmæssigt output (VST-udviklere).

Kvantitative og hybridmetoder

Vi har anvendt onlinefora som kilde, hvilket er et metodisk svært placerbart fænomen. Mange mennesker er involveret, hvilket giver et forum karakter af kvantitative undersøgelser i den forstand, at der er meget information, der skal fortolkes. Dog er et forum ikke spørgeskema i ren form, da de data, der findes på et fora er kvalitative, individuelle udsagn. Hvilket er svært at behandle statistisk, hvad der heller ikke ville gavne vores brug af fora synderligt. Vi finder det irrelevant at stille spørgsmål til, hvor mange procent, der mener hvad i forhold til temaer, vi ikke har stillet op eller har kontrol over. Videre adskiller et forum sig fra en kvantitativ natur i kraft af at dialogen er fortløbende, hvilket et spørgeskema heller ikke er i stand til at indfange. Af disse grunde kalder vi brug af onlinefora for hybridmetoder, da empirisk brug af et fora tenderer mod kvalitativ såvel som kvantitativ vidensindsamling.

Empiri – domæneindkredsning

I dette afsnit præsenterer vi de interviewsvar og citater, vi gennem kvalitative og hybridmetoder har inddraget i designprocessens første fase. En række af interviewene er foretaget i forbindelse med et ophold i Berlin, hvor kunstnere, såvel som programmører fra firmaet *Native Instruments*⁴³, har deltaget. Andre interviews er foretaget i København. Endelig præsenterer vi de citater, vi har uddraget af Jack Darks *developer interviews*.

Interviews, Berlin og København

- *At åbne komposition for tilfældigheder - hvordan sker dette i praksis?*

Klaus Voltmer, lead programmør hos Native Instruments, Berlin, april 2005: “It is all about enhancing new features that are easy to spot, but hard to figure out in a given interface. For instance I’ve built a new sequencer that works with known metaphors of sequencers that exist today – but mapped to new functionality. Something known, and something surprising at the same time. The data processing and signal routing diverse the known sequencers and you have a new level of modularity – now you can send so called ‘abstract’ data between modules in the program

⁴³ <http://www.native-instruments.de>

and not only OSC or MIDI. You can connect any entity to another and get some pretty surprising results. Right now you can't save you work and it has different bugs I need to work out, but the framework for a new type of computational improvisation is there. I think that is the kick I get from this application – I get new results every time I patch something differently, which is really cool to get away with in a performance situation. It is not visible to the listener what I do or the programs I use, but the experience I have as a musician increase in a positive manner this way”.

Anders Holst Jensen, Jazz guitarist, masteruddannet på Royal Conservatory of Haag, København, marts 2005: [om freejazz og modal improvisation] ”Det handler om at finde et så ærligt udtryk som muligt, for tit når jeg har teoretiseret en ide eller et mønster bliver musikken nerveløs og fortænkt. Og tit begynder jeg at gentage mig selv gennem faste antagelser eller mønstre, hvilket gør det sværere at udvikle sig musikalsk. Hvis jeg åbner min improvisation eller et nyt nummer mod en teoriløs tilstand, hvor tidligere øvede mønstre mere ligger som et latent sprog, finder jeg tit et følelsesmættet udtryk der føles godt. Musikken bliver i den forstand tilfældig, da jeg arbejder enormt struktureret mod at blive strukturløs.. giver det mening? Ikke kun tonalt men også lydæssigt hvor jeg har en del gamle stomboxes [guitarpedaler] jeg bruger til at forme tonen anderledes...[hvordan?] har en eller forkærlighed for en masse delays i serie med filtre imellem, hvis man slår guitaren an med et piskeris giver det sådan en 'tjirp, twing bing, riiiitsch' lyd, hvilket kan forandres radikalt ved at slå effekter til eller fra. Nu har jeg lige anskaffet mig en laptop og glæder mig til at voldtage den for lignende muligheder for at udvide det lydæssige univers yderligere. Du har vel sikkert også et par programmer til en gammel jazzrotte (griner)”.

Nestor Pridun, compiler & installation builder hos Native Instruments, Berlin, april 2005: “What the fuck do you mean Mads? Nothing is coincidence when I do stuff, I just fuck up a lot and make a lot of flaws in my code on purpose. Small surprises for the poor idiot that is going to execute the file (evil laugh). Seriously.. For instance when I build a live setup using Numerology⁴⁴ I always use plugins like Destroy FX [granulær sampler/delay effekt, <http://destroyfx.smartelectronix.com/>] or other non linear signal treatment. Just to get beyond what I thought it would sound like, but still with a certain degree of control, which really vary. Sometimes I want every parameter to do exactly what I want it to do.. sometimes not. Lately I have been using Live⁴⁵ a bit with a wide variety of plugins, it is easy to change the musical structure as you go along with good time warping features, and the bashing of the textures comes through the plugins”.

⁴⁴ <http://www.five12.com/>

⁴⁵ <http://www.ableton.com/>

Pure, Noiz kunstner fra Berlin, professionel musiker: “I find it very easy to break routines; look at the libraries available online. For all shells basically there are huge PD⁴⁶, Max/MSP⁴⁷ and Reaktor⁴⁸ forums and everybody is bridging everything. That is for me the easiest way to change as a musician, or do something conscious like opening a piece up for random outcomes – to go online and get new ideas, machines or whatever you feel you need. I always get surprised checking out new code or machines, and lately it has been mostly random stuff because I had to play at galleries and had the role of accompanying images in installations. Glitch is good for this, because people tend to get abstract when listening to abstract music. In that way random composition is kind of a portal to experience art as something without a certain meaning but as a means of thinking differently. Right now I’m using Plogue⁴⁹, PD and assorted Linux terrorism as hosts for a lot of freeware plugins. I have very little money to spend on software so I settle with freeware.”

- *Forhold til generelle arbejdsprocesser med musik – hvilke arbejdsmønstre er tydelige?*

Klaus Voltmer, lead programmør hos Native Instruments, Berlin, april 2005: “Right now I’m really into working with my old hardware again. Recently I have patched a 808⁵⁰ drum machine to change programs in my Juno 60⁵¹. This is done rhythmically so the program changes for every beat, you have to have 8 different programs you preprogrammed for 8 beats on the drum machine for instance, 1 a bass sound, 2 a pad sound, 3 a fx sound, 4 a bass different sound and so forth. You see, the Juno 60 makes a really nice sound when changing programs – like a little ‘chirp’ and flicking through programs sounds really nice and rhythmic correct... glitchy. Also nice melody lines coming out of this setup, both bass, pads and fx on the same machine. This hardware fetish combined with a lot of Pure Data patches makes a nice arsenal of fun scenarios. Right now I’m trying to keep every possibility open and not get locked with certain tools because of principles – and I use tools that keep this window open like PD or Reaktor if I want a GUI”.

Mate Galic, Native Audio evangelist CEO, Berlin, april 2005: “The other day I was blown away by the sound of my old synth – all this computer bullshit just sounds bad compared to the headroom of a real dedicated machine. But then when I actually want to make music I have no patience screwing around with hard-to-operate synths, I just sample the sounds and get on with it. Same goes for applications; I want a nice interface that doesn’t confuse too much and have sufficient possibilities. I like Reaktor in that sense that all the nerdy users build machines and I

⁴⁶ <http://www.pure-data.org>

⁴⁷ <http://www.cycling74.com>

⁴⁸ <http://www.native-instruments.de>

⁴⁹ <http://www.plogue.com/>

⁵⁰ <http://www.synthmuseum.com/roland/roltr80801.html>

don't have to think about that. And I like the fact that we control what possibilities they have, it is actually empowerment, because we have a certain control over what electronic music is sounding like these days. I can spot the sound of a Reaktor filter miles away, I know how the aliasing was build and how it works the sound... [tænker lidt] Responsibility in a way”.

Julian Ringel, Program manager for Reaktor og Kontakt hos Native Instruments, Berlin, 2005: “You know I loved the Micro Modular⁵² and the relation between hardware and software. Building a patch in software and upload into a interface-less device where only the essentials are represented [4 knapper der kan mappes frit til funktioner i patchen] is a solid connection between the vast options you have in the computer and the sound and ease of use of hardware. This machine represents the ideal way – superior possibilities and dedicated DSP's in a dedicated machine. Also the Muse⁵³ [hardware man kan loade VST-plugins i] is one of my favourites – at least the concept of it, because it is crashing like hell and seems really unstable. On work patterns I try to use hybrids like this when I have the time to do something in the studio”.

Nestor Pridun, compiler & installation builder hos Native Instruments, Berlin, april 2005: “Never, never, never ever repeat yourself. Never! When you have done something once – get it perfect because you are never going to be in the same place again. Making music should be fluctuant and in flux. Since you have no need for motoric skills as a nerd to maintain musical skills – why not play your instrument differently every time you use it? Anti repeat would be my pattern”.

Anders Holst Jensen, Jazzguitarist, masteruddannet på Royal Conservatory of Haag, København, marts 2005: ”Jeg skriver noder, dvs. stykker med struktur som melodilinjer, broer osv. Hvad der kommer mellem disse strukturer er op til en given besætning at fortolke gennem improvisation. Det er de facto standarden for hvordan jazzmusikere arbejder sammen. Strukturer som en slags enighed og en form for frihed ovenpå disse strukturer”.

- *Forhold til arbejdsprocesser indenfor 3D domæner – hvad giver visuelt mening?*

Klaus Voltmer, lead programmør hos Native Instruments, Berlin, april 2005: “I try to keep everything flattened and avoid esoteric 3D that brings my setup to its knees. It is good fun working in 3D, but it is rare that this type of interface brings out new forms of modulation, composition or whatever. In a way I see 3D interfaces as overkill because it is so CPU hogging and hard to get into the code I'm writing these days. I definitely see the potential of 3D and work around spatial

⁵¹ <http://www.vintagesynth.com/index2.html>

⁵² <http://www.clavia.se/nordmodular/>

⁵³ http://www.museresearch.com/receptor_overview.php

representations with vector because of this, but OpenGL is still a no go. However I've built video software for doing visuals which includes all standards. It reacts to an audio input, in this use case I see 3D as the missing link between music and listener and a fun translation between sound and image”.

Egbert Jürgens, senior development manager hos Native Instruments, Berlin, april 2005: “In principle 3D is the funniest way to work computers. Just imagine running around in a game-like interface and shooting this thing will make that sound. Have you seen Japanese arcade games? [ja] – most of them are musical games, like get the sequence of notes right, dance like this in front of the machine and so forth. Basically through the concept of the arcade machine you exercise everything there is to making and understanding music – like timing, understood as space between notes or periods. It is a very bodily way of having fun, and gaining musical experience through playing both the game and the music in the game”.

- *Ergonomiske overvejelser og forhold – hvad placeres hvor, for at lave hvad og hvorfor?*

Klaus Voltmer, lead programmør hos Native Instruments, Berlin, april 2005: “Strangely my hands and shoulders aren't fucked up from working with computers yet. So I just pile everything I want to use in the middle of the studio and dive in regardless of ergonomics. I change my setup constantly and have to have a flexible studio in that sense – I use some rack mounts to keep everything in the right height but that is the only consideration I've made”.

Anders Holst Jensen, Jazzguitarist, masteruddannet på Royal Conservatory of Haag, København, marts 2005: ”Guitaren er det primære værktøj når jeg spiller, men komposition foregår ofte foran klaveret. Jeg har let ved oversætte ideer fra guitaren til klaveret, og finder det mindst belastende at sidde ved klaveret mange timer i træk”.

Egbert Jürgens, senior development manager hos Native Instruments, Berlin, april 2005: “Everybody seems to be wanting to do smaller devices, I'm not quite sure I subscribe to this tendency. The human body has average measures that devices need to be adapted to – smaller devices do not always reflect this. Before the components were big and the devices were built as small and handy as possible under the circumstances, and now they are still built as small as possible the components just got really small along the way. It is like old principles in designing devices need to be revised – and foremost developed by asking what device is suited and situated for what type of use. I hate my cell phone because of this – fingers too big for the keys and my mouth should be right beneath the ear. I see instruments with similar flaws”.

Julian Ringel, Program manager for Reaktor og Kontakt hos Native Instruments, Berlin, 2005: “I like to be able to leave my computer; like get a lot of LFO’s to modulate themselves and let the modulation run its own course. I’ve built tons of self-generating machines in Reaktor from the same principle – to let the computer run chaos control on my filters and effects, and make all the notes, beats or whatever on hardware sequencers and synths”.

- *Gamer relationer – spiller du?*

Klaus Voltmer, lead programmør hos Native Instruments, Berlin, april 2005: “The PS2 console in studio is almost always on, snowboard games are cool. But no net based stuff – I don’t have internet in the studio or at home to have some time offline. I play a lot of kicker too [bordfodbold]”.

Egbert Jürgens, senior development manager hos Native Instruments, Berlin, april 2005: “Yes, love it. Mostly shooters and other absurd and violent games”.

- *Grad af tilpasning i forhold til forskellige miljøer (shells: Buzz, Cubase, Logic, PD, MAX, Reaktor) - Hvad gør en bruger/designer for at adaptere forskellige miljøer i enten design eller musikproduktion?*

Klaus Voltmer, lead programmør hos Native Instruments, Berlin, april 2005: “I have all the NI [Native Instruments] components and C programming environments I need to build whatever comes to mind, so no real restrictions in that way. I see myself as a high modular man building highly modular things (griner). It is easy for me to modify products like Reaktor, Kontakt or whatever because I have the source code – that enabled me to start launching and modifying Native releases for Linux and other shells. I find it fun in a really frustrating way to break new boundaries in this way”.

Pure, Noiz kunstner fra Berlin, professionel musiker: “I go all the way – if I find some need that existing applications doesn’t fit into, I build my stuff in C. Now you have more object oriented ways of doing new applications such as Synthmaker⁵⁴, SynthEdit⁵⁵, Energy FX⁵⁶ etc. and often you don’t have to dive into die-hard math, but rather construct your idea from existing modules”.

Julian Ringel, Program manager for Reaktor og Kontakt hos Native Instruments, Berlin, 2005: “We just opened Reaktor up for user integration of home made C modules. Which means you can

⁵⁴ <http://www.synthmaker.com>

⁵⁵ <http://www.synthedit.com>

⁵⁶ <http://www.xt-hq.com>

use your own code within the boundaries of Reaktor. Even the Kontakt sampler has a new ‘scripting’ feature, where you can write commandlines similar to code language to modify samples, layer them, or sequence them. All in all I try to think of our users as pretty hardcore and give them hardcore features. We have other products that benefit the ‘quick-and-dirty-butter-and-bread’ users who want results fast – for instance sample libraries with premade sounds, no coding or screwing around with samples or synths needed to get a production done fast”.

- *Forventninger til udvikling – hvordan ser din fremtid ud, hvilke tendenser ser du?*

Klaus Voltmer, lead programmør hos Native Instruments, Berlin, april 2005: “Right now I see Reaktor as a tool for the ordinary musician, and really well working too. No real obstacles in building whatever you need. For myself I just keep patching and combining what I have and make interesting things that pushes the music in new areas. Used to a lot of art installations and live gigs, but now I’m mostly into studio work doing releases for different labels”.

Julian Ringel, Program manager for Reaktor og Kontakt hos Native Instruments, Berlin, 2005: “Like I said, there are so many sample CD’s floating around which is making most digital productions sound the same basically. A very sad tendency is that the sound of electronic music is becoming homogenised, because musicians chose the easy way out to generate sound fast. Instrument designers like Mike Daliot is fed up with doing presets on his machines because it creates a distance between the capabilities of the machine and the users exploration of these capabilities. If there were no presets, everybody would be forced to sort of ‘dive in’ and find out for themselves. In a way I support this argument, but management dictates that we must do these fucking presets so everybody can feel like a star after flicking through the premade sounds and picking them out for their own productions – quick and dirty”.

Egbert Jürgens, senior development manager hos Native Instruments, Berlin, april 2005: “Tactile interfaces is moving in and the functionality is constantly increasing in all aspects. Therefore the touchable is linked to the *functionable* and interfaces are just better looking now because most people have more than 16MB memory dedicated on their graphics cards – which gives us more headroom to work with as developers. A tendency is also that device communication is increasing, more bandwidth enables peer to peer programs to function [se evt. www.peersynth.de] and musicians are closer now because of it. Sharing both concepts of music, applications, patches and any type of music related material”.

Nestor Pridun, compiler & installation builder hos Native Instruments, Berlin, april 2005: “Right now I feel a bit anti-music, which is making me do evil and fucked up music. I like it – so my

future is a blend of music and anti-music. Along the way I find software that support my current state of mind”.

Interviews, Jack Dark onlineforum

Interview med Russell Borogove (Tiny God), VST-udvikler
(<http://jackdark.proboards39.com/index.cgi?board=Developers&action=display&num=1109306612>)

Where do you see the future of music software/VST creation taking us?

Hard to say. I hope we get higher-level control constructs, and control systems that work well at all levels. The current state of the art in soft synths gives you an incredible ability to craft the details of a single note. That's great as far as it goes, but music is fractal, it has details at all sorts of scale, from the entire catalog of an artist, to one album or one symphony, to one movement or one song, to a section, to a measure, to a note, to the waveform of the LFO modulating the cutoff frequency of the filter and the movement of the keyboardist's hand on the pitch bend wheel, and every part of that is interesting. It's hard to pay attention to all those levels at once, and even though we have tools for all those levels, they aren't well integrated. Take Orion, for instance, it's got a nice fast little piano roll sequencer that's really easy to use, and that bridges the gap between notes and passages. Then it's got a completely different interface, not nearly as intuitive or easy to use, for grouping those passages into songs. From an experienced programmer's point of view, that's incredibly short sighted. It's the same operation, it's recursive chunking. The interface should be as similar between those two levels as you can make it, and more importantly, it shouldn't stop at two levels. You should be able to take a sequence of patterns and say "this is itself a pattern". Then you need to be able to customize each instance, you need to be able to say "repeat pattern foo but transposed up a fourth", "repeat pattern blat but with this effect added," and so on. Honestly, there's probably software out there that does this; there are a lot of audio workstations I haven't even touched, so just gently point it out if I'm overlooking something.

Interview med Oli Larkin, SynthEdit VST-udvikler
(<http://jackdark.proboards39.com/index.cgi?board=Developers&action=display&num=1109459254>)

If you could create the ultimate VST despite coding/time/software limitations, what would it be?

I'd like to make an on-screen controller that used a real physics based environment to control music. So you could make an imaginary room and roll a ball around it and the sound would change

depending on the speed, if the ball bounced, the position of the ball etc. Obviously it'd be best in surround sound. I really like dynamic sounds in music, like the Doppler, Aphex Bouncing Ball or the sound of something bending, snapping or being twanged etc. I'd like to work more on this kind of stuff.

Interview med Paul Chana og Chris Mower (Exponent), VST-udviklere (<http://jackdark.proboards39.com/index.cgi?board=Developers&action=display&num=1109915683>)

You release your plugs under the "commentware" clause. How influential is/are feedback/comments to the future of your plug-in design decisions? Or are you mainly interested in how well an already released plug-in works?

User comments are everything in the development of future plugins. All future plugins get many of their core ideas / features from these feedback messages that we receive. It also lets us judge what people think of the plugins we already have out there (were we along the right route of design / was the quality good enough etc etc). Both Chris and I have been very surprised by some of the comments that we have received for the Delta series, and it made us go back to the design board and redo the core of the effects.

Interview med G. Hagen (H. G. Fortune), SynthEdit VST-udvikler (<http://jackdark.proboards39.com/index.cgi?board=Developers&action=display&num=1108861374>)

Where do you see the future of music software/VST creation taking us?

Just yesterday I thought that maybe in 10 or 20 years we might have a big synthengine where You only need to drag and drop some SysEx-data from e.g. a Prophet 5 and it will show up a Prophet 5 and sound like a Prophet 5. In a further 5 or 10 years this engine might be capable of sounding like a revision 1, 2 or 3 Prophet 5. The same maybe with drag and drop of fpx and fxb-files from VSTi. But this is only the part of preserving the sound of the classics. On the other hand there might be new means and media to control the software, and I think this will be of growing importance. To my opinion most people have more fun using compact and efficient VSTi than big monsters where you rather need to get a diploma to program it. Ok, there are a few people who like to tame big monsters. But in most cases the art of wise restriction is a lot more efficient in the end and the result is still featuring a huge and unique sound but a lot easier to be handled by the user esp. if he wants to make music with the instrument.

Vores konkrete anvendelse af disse citater i udviklingen af prototypen vil diskuteres i kapitlet *Prototype* som empirisk fiksering af hhv. æstetik og interaktion.

Systemfilosofi

Før vi behandler hhv. æstetikken og funktionaliteten mere indgående, vil vi her først se på den kombinatorik, der binder dem sammen i vores konkrete design. Denne kombinatorik forstår vi som *systemfilosofi*⁵⁷, dvs. en filosofi, der angår 'systemet' i betydningen 'softwaresystemet' og den udvikling, der leder frem til dette softwaresystem. Systemfilosofien kan forstås som en konkretisering af de interne præmisser for designet.

Æstetik og funktionalitet – værktøj, oplevelse og forstyrrelse

Vi sonderer i designet mellem æstetik og funktionalitet – ikke som hinandens modsætninger, men som to egenskaber ved et design, der i kombination giver det form og retning. Æstetikken og funktionaliteten griber ind i hinanden. Således rummer en visuel æstetisk taktik muligheder for en alternativ processuel tilgang til funktionalitet. Samtidig vil et andet udbud af funktionalitet afspejle en anden æstetisk oplevelse, både visuelt, auditivt og i det interaktionsniveau, der binder disse domæner. Forstår vi 'æstetik' som 'hvordan det opleves' og 'funktionalitet' som 'hvad det kan', ser vi at kombinationen af disse to er væsentlig i designet af et software-instrument. At spille på et instrument er ikke blot nøgtern betjening; det indebærer også et aspekt af oplevelse af instrumentets artikulation, der rækker ud over det umiddelbare auditive resultat i højttalerne. Oplevelsen driver en motivation for at udforske instrumentet videre⁵⁸. Således må et software-instrument indenfor denne systemfilosofi forstås som mere end et koldt værktøj – det rummer også et potentiale til overraskelse, forstyrrelse og uventede oplevelser hos brugeren. Feedbacken fra disse uforventeligheder repræsenteres visuelt, hvilket er en æstetisk-funktionel strategi. Vores præference i designet går derfor mod en kombination af et værktøjs- og et oplevelses-aspekt, således at softwaren ikke blot har en specifik form som alternativ MIDI-controller, men også som en kilde til selvvalgt forstyrrelse af den kreative proces hos den enkelte musiker.

Gensidige forstyrrelser mellem omverden og designsystem

Som sagt forstår vi også designaktiviteten som rettet mod forstyrrelse af eksisterende vaner og forventninger i musikproduktion. Dermed forekommer vores rolle som softwaredesignere ikke blot som neutrale leverandører af værktøjer, men også som 'forstyrrende elementer' med et anderledes

⁵⁷ Mathiassen, m.fl., 1997

⁵⁸ Jf. behandlingen af motivation under *Information* i kapitlet *Metode*

bud på hvordan modificering og interaktion med musik kan foregå. Dermed træder vi bevidst udenfor den rolle, som metoder til systemudvikling normalt tildeler designere. Vi trækker i stedet på en forståelse af designerens rolle, der traditionelt forbindes med såvel industriel innovation som kunstnerisk aktivitet. Vores ønske er at kombinere to forskellige paradigmer – hhv. et værktøjsparadigme og et forstyrrelses- eller oplevelsesparadigme – idet vi ser vores bud på et softwareinstrument som noget, der har korrelater i begge paradigmer. Med andre ord vil vi ikke påberåbe os en neutralitet som designere af et alment værktøj, men tværtimod hævde at en sådan neutralitet er umulig i designet af artefakter. Selvom softwaren kan være nyttig som værktøj opnår den også en form som forstyrrende element i kraft af dens særlige kombinatorik af æstetik og funktionalitet. Når artefaktet anvendes i en specifik kontekst, tages det i brug med divergerende forventninger hos forskellige brugere og underlægges forskellige tolkninger. Dette medfører at vi ikke designer mod et neutralt eller objektivt defineret anvendelsesområde. Derimod klargør vi på forhånd, at vores design har potentiale til overraskelse eller forstyrrelse, alt afhængig af den motivation og interesse som en fremtidig bruger møder designet med.

Designprocessen mindsker designets usikkerhed i forhold til en konkret anvendelse, idet det fremstår som resultatet af en lang række valg og fravalg med rod i empirisk og teoretisk bekræftet information. Softwarens værdi beror på en sammenvævning med konkrete ideer og behov, men antager samtidig karakter af mulighed for forstyrrelse, idet den gives en særlig identitet gennem designprocessen.

Dynamikken mellem perturbation udefra og designets egen forstyrrelse af den omverden, det indsættes i, kan genlæses i termer af innovation og adaptation. 'Den gyldne mellemvej' mellem tilpasning og forstyrrelse søger vi derfor at nå gennem en reflektiv vekslen mellem disse to momenter og deres indbyrdes dynamik.

Næste afsnit vil behandle æstetikken nøjere som en dynamik mellem tilpasning og forstyrrelse, mens funktionaliteten tages op under *Funktionalitet og modificering*.

Æstetikken i og omkring prototypen

Æstetikens vilkår

Det er i sig selv en umulig opgave at definere æstetik entydigt. Alt efter hvilket medie æstetikken beskrives i forhold til, rykker definitionerne sig. Dette er svært at applicere som tankegrundlag i forhold til prototypen, der simulerer eller anvender flere medier med dertilhørende æstetiske definitioner. I prototypen anvendes der en billedkomposition (traditionel billedkunst), der anvendes taktile former (skulpturkunst), formerne er interaktive (installationskunst), og der arbejdes med lyd

(musik), osv. Dette skaber en del overlap i det teoretiske apparat, som derfor bærer præg af kombinatorik af grenenes æstetiske nedslagspunkter og områder.

Videre har æstetik altid været relateret til organisationsstyrede konsensusformer, hvor relative termer som grimt, skønt, spændende eller ligegyldigt fastslås på baggrund af segmentering af mængder – hvor mange der synes det er grimt, lyder godt, forekommer ligegyldigt, osv. Disse sociale bevægelser kan også ses i kunstens historik, hvor romantikken, modernismen, postmodernismen, mv., alle har haft forkæmpere, der finder argumenter i nævnte konsensusformer – ideen om hvad kunst er og hvad der ikke er kunst. Dette er en form for enighed, der bygger på principper, som vi ser som mikro-former. Principper i mængder danner på et makroniveau en form, der illustrerer individers placering i forhold til hinanden. Dette beskriver vi som tendens, hvorunder også kunsthistoriske perioder også kan udlægges som tendenser.

Omverden for kunsten i dag lader sig i dag imidlertid ikke segmentere eller indlemme i nogen entydige tendenser, da eksempelvis 'beskuergrupper' eller 'kunstnere' er blevet diffuse størrelser. Beskuere, der for det første ikke godtager ekspertmeninger ukritisk, da mængden af tilgængelig kunst er vokset, hvilket har formindsket kunsteksperternes status som 'olympisk altoverskuende'. For det andet fordi information om kunst ligeledes er frit tilgængelig, og ofte er information der flyder 'direkte' mellem kunstner og beskuer i performance-situationer. Dette fører til at æstetiske begreber ikke er universelle, men derimod opbygget gennem beskuerens egen erfaring i observationen af kunst.

Nogle grundlæggende definitioner

Projektets definition af æstetik kan forklares med få, relativt banale, udsagn.

Æstetik defineres som *noget der gør noget*, hvilket er en løs formulering, men indkredser æstetikken som en påvirkningsfaktor, hvilket er det eneste vi kan fastslå med sikkerhed. Æstetik igangsætter en reaktion, hvis udgangspunkt eksempelvis ligger i et kontinuum mellem 'lige glad' og 'opmærksom'. Hvis definitionen udvides til *noget der gør noget ved nogen*, åbnes begrebet for et relativt moment, da ingen har den nøjagtigt samme reaktion. Førnævnte kontinuum er et individuelt sæt af kriterier, og derfor hænger reaktioner også sammen med individuelle konstruktioner.

Nogen der gør noget til noget beskriver hvordan konventioner opstår i kunsten, og indirekte i opfattelsen af æstetik. Dette sker gennem validering af noget frem for andet, gennem forskellige instanser og deres tilknyttede personale.

Kunst er anskuet som indeholdende et paradoks – kunsten består af både kunst og ikke-kunst. Bag dette ligger der også et sæt kriterier for æstetik og ikke-æstetik, der påvirker definitions- eller afgrænsnings-spørgsmål.

Intern og ekstern æstetik

Som kapitlets overskrift antyder, anskues æstetikken overordnet fra to vinkler i forhold til prototypen; hhv. den interne æstetik, programmet betjener sig af, og de eksterne æstetiske betingelser, som en bruger af programmet opererer med. Et udtalt formål med specialet er at gøre en kobling mulig, men ikke sikker, mellem en musikers æstetiske kriterier, og den æstetik, der gennem en brugerflade og auditiv feedback binder funktionalitet til grafik. Vi finder det givende at operere med et begreb om systemintern æstetik i forhold til brugerfladen, da eksempelvis metaforikken i interfacet forskydes gennem iterationer, og påvirker konceptet. Denne æstetik er fastlåst af de designmæssige selektioner, vi aktualiserer i bevægelsen fra analyse til syntese, og ikke et element, som en brugers interaktionsmønstre kan forandre. En bruger kan derimod forandre forståelsen af metaforikken for at følge eksemplet til dørs. En musiker er et kulturelt betinget fænomen, der for interfacet indebærer en systemekstern æstetik, som ligeledes gennem vores analyse forskyder konceptet i forhold til de første udkast gennem iterationer. Dette sker specielt i forhold til funktionaliteten, og i hvor vid udstrækning denne understøtter de kriterier, musikkulturen betjener sig af i komposition eller performance. Specialets æstetiske udgangspunkt er dialektisk, da forklaringer på hhv. systemekstern og systemintern æstetik hverken findes udelukkende i systemet eller i den omverden, systemet er designet til. Æstetikken er såvel individuel som de ydre påvirkningsfaktorer, der skubber individet mod en forståelse af interfacets æstetik.

Systemintern æstetik

I følgende afsnit redegøres for interne æstetiske aspekter af prototypen.

Visuelt design

Prototypens kobling af auditive og visuelle elementer påkalder en uddybelse. Ikke alene har interfacet betydninger for opfattelse af musikbegreber som notation, partitur eller interfacelayout i softwareområdet. Men også æstetiske konsekvenser for musikken, da vi anser teknik som et medie for kompositoriske antagelser. Hvilket står i et gensidighedsforhold, da kompositoriske antagelser også bliver et medie for teknik. Der er et forhold mellem teknologiens specifikke egenskaber og de konceptuelle rammer, disse egenskaber bliver udfoldet i. Computeren bliver i denne tænkning et værktøj, der understøtter en (konceptuel) nytænkning af musik, og prototypen et middel i denne proces. Koblingen mellem det visuelle og auditive er bl.a. inspireret af Iannis Xenakis' grafiske

tilgang til notation, hvis motivation han beskriver som: ”Give birth to the yet unconceived through visual thinking”⁵⁹.

Xenakis’ mål var at etablere nye plastiske kvaliteter ved musikken gennem en anderledes notation, hvilket er visuelle eksperimenter med musik. Igen kan teknikken udlægges som medie for æstetiske antagelser og vice versa. Sven Sterken har også anskuet Xenakis’ kobling af *material spaces* (vægge, vinduer, passager mellem rum) og *energetic spaces* (varme, lugt, farve, lys og lyd) i hans polytope⁶⁰. Med hovedspørgsmål som kropslighed i immaterielle rum som udgangspunkt, når han dog ikke længere end til et beskrivende niveau af hvad der findes i begrebet rum. I specialet udvides dette framework til modale rum, hvor der findes modsætningsforhold som vi anvender som æstetisk designstrategi. Vi betragter immaterielle rum som hybrider, hvor der i software er virkelighedsreferencer som funktion og simulation. Funktion ligger som sagt før ønsket om oplevelse, og skal relateres til en brugskontekst, hvor der er en anvendelsesdimension, vi skal overveje i designet, før designet kan understøtte en oplevelse. Simulation refererer til omverdensrelationer som tyngdekraft, masse, teksturer og former, osv. Når simulationen kobles til funktionen fremstår prototypen som et *material space* med dertilhørende love og forventninger til lovene, men samtidigt også som et *energetic space*, da prototypen spiller på sansemæssige elementer (hovedsageligt lys og lyd). Som Mitch Kapor hævder, er det fordelagtigt at designe mod ”the full range of human experience”⁶¹, hvilket vi i afsnittet her gerne vil udvide til også at være en oplevelse af funktion.

Æstetisk systemfilosofi

Vores design er funderet i funktion og form, der som anslået i afsnittet *Systemfilosofi* er et spændingsfelt i designet og designmetodikken. Kreativitet er en kobling, der gør et design muligt, men samtidigt også det led i udvikling, der er mest problematisk at eksplicite. Kreativitet er sat i gang og opretholdt af en æstetisk dømmekraft, hvor denne ’domfælden’ har udgangspunkt i systemlokale kriterier. Spørgsmål som ’hvorfor blå’ besvares med ’fordi blå er fedt’, der ikke nødvendigvis giver mening eller berettiger noget som helst. Kreativiteten i designets brugerflade og funktionalitet er derfor syntetiseret af os, mens brugeren på den anden side af designet kan bruge designet kreativt. Prototypen er en artikulation af vores æstetiske selektionskriterier, der faciliterer rammer for en brugers æstetiske orientering og kreativitet. Det er brugerens kreativitet og æstetiske dømmekraft, der er prototypens omdrejningspunkt, hvor en given funktion er afledt af systematisk udvikling af et design gennem informationsbehandling. I systemfilosofien er vores og brugerens

⁵⁹ Kanach, 2002, u.s.

⁶⁰ Sterken, 2001, u.s.

⁶¹ Chu, 1999, p. 1

æstetiske orientering indarbejdet som to betydningstunge led, og iht. vores begreb om *gensidige forstyrrelser* er æstetik en variabel, der skaber emergente former og designs.

Interfacet er konkretiseret ud fra denne æstetiske semantik, der har rod i systemfilosofien. I denne proces arbejder vi som nævnt med hybridrum, eller mere udfoldet, modale miljøer. I disse rammer har vi sigtet mod at skabe konsistens, forstået som feedback fra interfacet i forhold til det auditive output. En emotionel konsistens mellem objekter og musikeren, der manipulerer objekterne i real time⁶². Et instrument fungerer på denne måde, når der er en intimitet til stede. Instrumentets mekanismer er bekendte, og instrumentets reaktioner på forskellige handlinger er ligeledes opfattet i f.eks. en øvesituation gennem gentagelser. Det udlægges som sansemæssig redundans, hvor flere forskellige sanser samler information for at processere en oplevelse.

Metaforik, rumlighed og naturlige egenskaber

Som anslået i afsnittet ovenfor anvender vi sansemæssig redundans for at understøtte oplevelsen af et instrument. Denne redundans berører metaforer som et bindeled mellem auditive og visuelle enheder – hvor noget kendt gøres modalt. En metafor har en udsigelseskraft, hvis betydningsreferencer og kontekst kan forskydes eller ændres. I vores tilfælde er disse referencer og forhold også brugt som interaktionsdesign (som næste afsnit vil indeholde). Metaforen binder objekterne til en bestemt type interaktion, og fundamentalt blotlægger metaforen muligheder og begrænsninger i prototypens interface. At have objekter, der kan manipuleres direkte, er en metafor for objekter, der findes i verden, der har samme manipulerbare egenskab – at sparke til en bold har konsekvenser for boldens placering i et rum, og hårdheden af sparket bestemmer boldens fart i rummet. Flexibiliteten og kontrollen i interfacet er på den måde direkte relateret til interfacets omverden gennem metaforer. Metaforikken har vi overført til prototypen, hvor handlingers attributter som eksempelvis *hvor hårdt*, *hvor hurtigt*, osv., bestemmes på samme måde som i den verden, hvor metaforen har sit afsæt. Det kan udlægges som en metafor for naturlige egenskaber, hvor vi anvender erfaringer med tyngdekraft og bevægelsesenergi i designet. Rumligheden i designet er ligeledes baseret på metaforik, da vi har anvendt teksturer og bitmaps for at ”tapetsere” prototypen med kendte elementer. For det første for at skabe en niveau af intimitet i interfacet i forhold til brugeren, og for det andet fordi vi anser det som nødvendigt at give æstetisk funktionalitet som lydmodificering en ligeledes æstetisk funderet form.

For at kunne begribe et sæt x, y, z-koordinater i et interface må koordinaterne have et referencepunkt, hvor der af beskueren konstrueres en forgrund, baggrund, osv. Referencepunktet er i prototypen den interaktive: Brugeren, der i overensstemmelse med x, y, z-orientering konstruerer

og indarbejder metaforen som fortolkningsredskab af rumlig placering. Denne æstetiske strategi er anvendt i 3D-spil, hvor referencepunktet flytter sig i overensstemmelse med mus og keyboard-input. Udgangspunktet har vi overtaget med det belæg at skabe intimitet med kontrol og grad af fleksibilitet.

Interaktionsdesign

Som sagt anvender vi referencer for at designe interaktion. I forhold til en brugers kontingente orientering i prototypens interface kan vi ikke 'sikre' eller determinere en brugers meningskonstruktion og efterfølgende interaktion, da motivation ikke kan sikres. Det er ikke sikkert, at brugeren har de samme referencer som os. Disse referencer hænger sammen med de selektionskriterier, der for brugeren (som udøvende musiker) og os (som designere) tilvejebringer mening. De sæt af selektionskriterier, der findes i hhv. designer- og brugergrupper er som nævnt konstruerede lokalt i systemerne, og divergerer på forskellige områder. Meningen med interfacet må vi derfor stabilisere gennem brug af metaforik, sansemæssig redundans, osv. Dette er alle elementer, som må indordnes under begrænsninger i et interface, der er det eneste vi kan designe i forbindelse med interaktion. Disse begrænsninger berører det æstetiske layout af prototypen som de objekter, der bliver *mappet* til funktionalitet i interfacet. Interaktionen er designet ud fra udsagn, der beskriver prototypens systemfilosofi, og som skal afdækkes af musikeren. Disse udsagn er:

- Objekter til stede i interfacet kan påvirkes; hvilke, hvordan og med hvilke konsekvenser for lyden afdækkes gennem en sondring mellem eksperimenter og lytning. I denne proces indoptages eller afvises designergruppens selektionskriterier. Begrænsningerne går på hvilke objekter, der er manipulerbare.
- Prototypens form som scene er på et interaktionsniveau tilgængelig gennem referencer til computerspil eller andre simulerede 3D interfaces. Filmiske virkemidler som overlappende planer, farvegradueringer og andre billedkompositoriske værktøjer ændres i forhold til den interagerendes 'position' i prototypen. Interaktionen og hermed navigation er mulig i henhold til tidligere erfaringer og altså en kernereference i forhold til brugerens analyse af hhv. form og funktion. Begrænsningerne vedrører 'hvor' brugeren kan befinde sig i prototypens interface, og hvilken interaktionsform, der understøtter denne kontrol.
- Prototypens kobling til det auditive erkendes som en kontrolflade der er spatielt programmerbar. Prototypen vurderes som brugbar i forhold til en anvendelseskontekst og

⁶² Serafin & Young, 2003 beskriver dette som *playability*

kan indgå i en given opsætning, der ikke spidsbelastes. Begrænsningerne går på hvilke kommunikationsstandarder mellem komponenter, vi anvender, hvilket resulterer i en æstetisk begrænsning: Prototypen fordrer en sequencer, der kræver en del computerkraft. Hvis prototypen overskrider et 'tolerance' niveau må æstetikken nedtones – eksempelvis færre lys effekter, lavere opløsning af de visuelle effekter, herunder specielt bitmaps.

Instrumentdesign

Dette afsnit vil forsøge at indarbejde erfaringer og æstetiske perspektiver fra forskere, der har arbejdet intensivt med digitale instrumenter, gestik og musikinterfaces. Her tager vi udgangspunkt i den generelle udvikling indenfor design af musikinstrumenter hvor interface og lydkilde er adskilt, hvilket er relevant for vores design.

Gestik og instrument

At spille musik kræver finmotoriske færdigheder, såvel som øvelse for at tilegne sig evner. Målet er i sidste ende at gøre instrumentet transparent, hvor instrumentet som medie ikke hindrer den udøvende musiker i at flytte sin forestilling om lyd fra ideplanet ud i højttalerne. Dette niveau er virtuosens, hvor der findes en højtudviklet intuition og forståelse af instruments kapacitet forstået som muligheder og begrænsninger.

Dette grundlag kan imidlertid reverseres, idet man lige så godt kunne arbejde med ideen om at tilpasse instrumenterne til den udøvende, i stedet for at den udøvende tilpasser sig instrumentet. Denne fleksibilitet er funderet i de teknologiske principper, der ligger bag instrumentets udformning, da et elektronisk instrument muliggør en stor grad af klangmæssig fleksibilitet gennem muligheden for rekonfiguration af lydmaterialer såvel som interfacet. Eksisterende ikke-elektroniske instrumenter er i musisk praksis klang- og formmæssigt begrænset af de materialer, de er konstrueret af. Derimod kan digitale instrumenter i princippet betjenes gennem alt, hvad der kan generere eller styre data, hvilket ikke fjerner *alle* begrænsninger for instrumentdesign, men udvider området betragteligt. Her er det relevant at skue til Perry Cook⁶³, der blandt andet opridser principper som: '*Existing instruments suggest new controllers*' og '*everyday objects suggest amusing controllers*'⁶⁴. Dette ansporer et design af en *controller* eller et interface, der udnytter muligheder for teknologisk og konceptuel kombinatorik og omverdensreference. Et designkriterie for os bliver dermed *transparens* (der behandles videre i næste afsnit) hvor interfacet gennem kombinatorik udnytter en allerede eksisterende proces til at etablere en ny proces. I denne

⁶³ Cook, 2001, p. 1

⁶⁴ Cook, 2001, p. 1

sammenhæng er det i vores favør at digitale instrumenter kan reagere på symbolsk information, eksempelvis objekter i bevægelse inden for et virtuelt rum. Herved åbnes op for en forandring af det klangmæssige potentiale, der i høj grad afhænger af musikerens evne og lyst til at manipulere og rekonfigurere synthesizere, plugins, samples, osv., via prototypernes interface.

Transparens og konsistens

Som ovenfor beskrevet opererer vi med et aspekt af transparens, enten set som virtuosens omfattende viden om sit instrument, eller som det tilgængelige instruments egen transparens i kraft af, at det er muligt at tilpasse til musikeren. Vi designer mod at gøre instrumentets interface tilgængeligt gennem metaforik, så det virker transparent, og ikke kræver virtuosens kompetenceniveau. Samtidig erkender vi, at interfacet, for også at have brugsværdi for 'ekspert' musikere, skal kunne mere end give *instant gratification* – der skal være et potentiale for at gøre interfacet tilpas ekspressivt.

I den anledning er vi inspireret af begrebet om *control intimacy*⁶⁵ for at beskrive brugerens grad af fortrolighed og nærhed i interaktionen med interfacet, symbolsk såvel som konkret. Begrebet er introduceret af F. Richard Moore: "*Control intimacy determines the match between the variety of musically desirable sounds produced and the psycho-physiological capabilities of a practiced performer*"⁶⁶.

Umiddelbart handler *control intimacy* altså om et gensidighedsforhold mellem musiker og instrumentets muligheder. I denne forbindelse er det værd at nævne Cooks fokus på *player bandwidth*⁶⁷, der beskriver, hvordan instrumenter fordrer en større eller mindre mental 'båndbredde' af musikeren. Dette forhold anvendes i prototypen som begrænsninger for, hvor mange enheder i interfacet, der skal mappes til funktioner. Vi vil i så vid udstrækning muligt forhindre at interfacet *løber løbsk* og overskrider musikerens *bandwidth*, hvilket er et udtalt designkriterie.

Dermed fungerer *mapping* som en mere systemnær beskrivelse af hvad konsistens og transparens indebærer – en grad af forståelse og sammenkædning, der gennem praksis etableres af en musiker.

I prototypen anvendes 'naturlighed' i interfacet, forstået som en sammenhæng mellem "hvad det kan" og "hvordan det ser ud/hvad det ser ud til at kunne". Med andre ord en *konsistens* mellem

⁶⁵ Mulder, 2000, p. 4

⁶⁶ Mulder, 2000, p. 4

⁶⁷ Cook, 2001, p. 1

artefakt og operation. Konsistensen mellem artefakt og operation er ofte udledt af generalisering ud fra musikerens omverdenskendskab.

Kausalitet og ekspressivitet

En gennemgående motivation for at designe disse artefakter er den tilsyneladende mangel på kausalitet i arbejdet med elektronisk musik. I eksempelvis laptop-baserede performances er der en reel risiko for at kausaliteten skjules, når musikeren står bag maskinen og i en tilskuers øjne lige så vel kan læse e-mail som skabe musik.

Mere abstrakt er kausaliteten altså både det, der fordrer og forhindrer en forståelse af, hvordan lyd bliver tilvejebragt, hvilket kan modsvares med en projektor og metaforiske interfaces. En anden tendens, vi kan generalisere ud fra, er at musikere ofte arbejder med virtuelle instrumenter som haptiske og visuelle grænseflader for at højne kausaliteten i performances. Kendskab til denne proces ser vi som essentiel i forhold til at designe et interface, der ikke effektiviserer arbejdet med musik, men understøtter oplevelse i stedet, og altså en anden proces i vejen fra ide til lyd. Dette udgør begrundelsen for at arbejde med transparens og konsistens som designkriterier, der hjælper os med at designe mod *control intimacy* og kausalitet, og mod *playfeel* set som oplevelsesværdien ved et interface, der har disse transparente egenskaber.

Systemekstern æstetik

I følgende afsnit redegøres for æstetiske aspekter af prototypens omverden.

Musikerkultur – Glitch-æstetik og andre operationelle kriterier

Musikere har forventninger til klangmæssige potentialer og til de processer, der realiserer klangen. Et brud på disse forventninger er et *glitch*; noget ikke-forventet eller konkret en overraskende klang eller tekstur. En overraskelse, der i nogen grad er kontrolleret gennem parameterindstillinger, og ofte er en decideret planlagt fejl eller kontrolleret mulighed for overraskelse. Æstetikken i *glitch* handler om at give slip på kontrollen, men alligevel at besidde muligheden for i nogen grad at kunne kontrollere resultatet, og at indoptage dette moment af kaos i et musikalsk virke. Kontrol er en graduering foretaget af musikeren: Hvilke parametre, der skal overlades til tilfældigheder, frem for andre.

Glitch opstår i praksis f.eks. når et MIDI-apparat modtager flere signaler end det kan processere, hvilket skaber et misforhold mellem hvordan apparatet fungerer og hvordan det burde fungere. Ligeledes kan et computerprogram belastes til at lave samme misforhold ved at justere signaler og signalruter. *Glitch* begrænser sig ikke til 'mekaniske' tilfældigheder, men indbefatter også mere raffinerede algoritmer for lydbehandling. Eksempelvis volumen i forhold til et lydligt forløb over

tid, hvor volumenstyrke som en talværdi kontrolleres af ligninger, hvis variabler er relative værdier, der forandres over tid (af andre ligninger). Dette skaber tilfældige udfald af lyden, og et overraskende moment, der i nogen grad, rent kompositorisk, er underlagt kontrol.

Pointen er stadig, at de specifikke elementer i lyden, der skal underligges kaos eller kompleks adfærd, er udvalgt af musikeren ud fra æstetiske kriterier – om så uforudsigeligheden opstår ud fra mekanisk overbelastning, matematisk kaosgenerering i computeren, eller slet og ret en ligning/algorithm, der er mere kompleks end hvad brugeren forventer. Michael Hamman⁶⁸ ser musikeren som rummende en ad-hoc adaptering af sondinger mellem materiale og struktur – hvilket kan ledes tilbage til æstetiske definitioner/konstruktioner som *glitch*: "*The composer must synthesize an emergent comprehension of the behaviour of the system with a similar emergent model of musical material*"⁶⁹.

Udsagnet står i reference til de forventninger til kompositoriske og klangmæssige potentialer. Potentialerne opstår som nævnt i sondringer af muligheder via eksperimenterende handlingsmønstre. Heri relaterer musikeren modificeringsmuligheder til lydligt materiale. Disse viser sig i praksis som en vifte af muligheder, hvor nogle aktualiseres frem for andre i overensstemmelse med musikerens definition af kaos og orden (eller grad af *glitch*) i den musikalske struktur og den klangmæssige tekstur. Disse mekaniske muligheder drøftes i afsnittet *Funktionalitet og modificering*.

Musikerkultur – Serialisation

Det klangmæssige potentiale i materialet kan beskrives som en iboende egenskab ved digital lyd. I næste afsnit præsenteres begrebet om residualberegninger, der illustrerer, hvad der løbende sker med et element (som eksempelvis en tone) når det transformeres i et system. Via en analyse af forskelle på et element over tid, fra før elementet er modificeret, og efter det er modificeret, behandles lyd som residualberegnet. Vi anvender ordet forskel som et udsagn for, hvad der er sket med et element før og efter behandling i et system. Forskellen på video, lyd eller generelt data, der går fra input til output af et digitalt system, er ofte stor. Dette på grund af muligheden for manipulation, hvor begrebet serialisation er en faktor. Begrebet beskriver processering eller manipulation af digitalt materiale, og hvordan denne proces i forbindelse med lyd ikke er én isoleret instans, men ofte mange instanser, der er koblet serielt. Et banalt eksempel er effektmoduler, der sættes i en kæde for at opnå den ønskede klang. Eller mere vidtgående kan

⁶⁸ Hamman, 2002

⁶⁹ Ibid.

begrebet udlægges som en serialisering af algoritmer for signalbehandlingen i systemet. Principper for hvordan signalet skal behandles kan ligeledes serialiseres, hvor volumestyrke kan kontrolleres af flere instanser (som ovennævnte i *glitch*-eksempel). Musikere betjener sig af serialisering som en realiseringsform for æstetiske forventninger – hvordan det klanglige potentiale udvindes. Serialiseringen i projektet går på at binde visuelt og auditivt materiale sammen via komponenter og deres bagvedliggende koncepter.

Musikerkultur – Scriptzation

Tilfældighed og andre operationelle kriterier for musikere underlægges som nævnt et moment af kontrol. *Scriptzation*-begrebet beskriver denne kontrol, hvor scripts gennem algoritmer styrer det auditive output. Konkret er kontrollen med scripts ofte gemt i GUI-baserede interfaces⁷⁰, der er kompileret som VST-plugins eller *host/shell*, hvor scripts og deres funktion er repræsenteret med knapper. Efter en kompilering⁷¹ foreligger der ikke mulighed for at ændre et scripts funktion, medmindre programmet er et miljø, hvor scripting er tilgængelig for brugeren som f.eks. Max/MSP, Pure Data⁷², mv. Disse miljøer er ikke-kompilerede GUI-baserede interfaces, der lader brugeren definere funktioner, og bestemme i hvor vid udstrækning scripts skal gemmes væk i en pæn GUI. Brugeren kan med andre ord skrive scripts ind for signalruter, modifikation osv. I denne familie af ”semi-nørdede” programmer findes et overlap til basale programmeringsudsagn som ”*hvis – så*” (hvis volumen er X, så skal applikationen gøre sådan). Imellem disse paradigmer for GUI og kompilering findes der hybridmiljøer⁷³, hvor brugeren for det første kan definere scripts, og for det andet repræsenterer disse scripts i en GUI. Hybridmiljøerne kan relateres til objektorienteret programmering, hvor moduler indeholder scripts, en bruger kan inddrage og udforske alt efter evne, lyst eller behov. Der findes med andre ord en etableret æstetik for konstruktion af lyd. Brugeren besidder også mulighed for selv at kompilere eksempelvis en VST-plugin⁷⁴ eller en eksekverbar fil, og bestemmer derfor også graden af selvstændighed for applikationen, og hvor/hvordan designet skal eksekveres.

Funktion og form – en sang om struktur

Enhver repræsentation må have en funktionel binding for at gøre brugerflader interaktive. Metoden, hvormed denne kobling etableres, er et æstetisk kernepunkt, hvor funktion ofte kommer i første

⁷⁰ Som f.eks VST-plugins, Cubase, Logic.

⁷¹ En proces, hvor programkomponenter (kildekode) sammenfattes til en installationsfil, andre brugere kan installere

⁷² Se evt. ordlisten for et screenshot af Pure Data

⁷³ Som f.eks. Reaktor eller SynthEdit. Se evt. ordlisten for screenshots

⁷⁴ SynthEdit har denne funktion, hvor et objektorienteret miljø kan generere en selvstændig VST-fil (Konkret som eksport af en DLL-fil)

række. Ud fra denne balancering kan et design placeres i forhold til et formål – ud fra, hvordan funktion er relateret til æstetik i brugerfladen, i kraft af at designet har en form og i vores tilfælde en brugsdimension. Sondringen mellem hvad der vises, og hvad det kan, bliver i prototypen anvendt som et primært kriterie. Form og funktion er begge potentialer, der kun tilvejebringes gennem processuelle reduktioner, hvor vi har reduceret funktionaliteten i forhold til et æstetisk koncept. Funktion ser vi som et sæt kontrolinstanser; eksempelvis som scripts, der konkret indgår i en *mapping* eller et sæt af *constraints*. En struktur opstår gennem designmæssige reduktioner af muligheder, der bliver til principper som *glitch*, *scriptzation* og *serialisation*. Reduktioner af funktionalitet resulterer i færre muligheder, hvilket interaktionsformen også justeres i forhold til via kontrolinstanser og deres bagvedliggende principper. Formen er det auditive resultat af disse instanser, med andre ord hvad interfacet gør ved lyden. Dette moment er opjusteret i forhold til mængden af kontrolinstanser vi inkorporerer i designet – hvor det æstetiske udbytte af interfacet og dets modificeringsegenskaber er valgt frem for et tal-og-bogstavs interface med større funktionalitet. Dette valg er ikke kun foretaget med afsæt i empirisk data⁷⁵ og teori⁷⁶, men også fordi forholdet mellem form og funktion er muligt at reversere eller i det mindste irritere – udelukkende fordi man kan. Vi har i den bevægelse stræbt efter at konkretisere en mulighed for sonisk *re-entry* som en auditiv formforstyrrelse, der også kan spores og manipuleres visuelt. Denne mulighed raffineres gennem sondringer mellem formmæssige overvejelser, empiri, og de processer, vi er underlagt i udarbejdelsen af prototypen. Vi vurderer på trods af prototypens 'irritations' kriterier konceptet brugbart, da der findes et udtalt behov for konstant nytænkning af domænet musik og musikere.

Med belæg i mængden af strukturerende applikationer, hvor æstetikken er nedtonet i forhold til funktion⁷⁷, argumenterer vi for det ønskværdige i at give slip på dele eller helheder af musikalske strukturer og teksturer. Omverden for vores design er præget af inkluderende applikationer, hvor man kan læse udsagn som '*alt hvad den moderne producer har for*' eller '*complete suite*' på retail-emballeringer og andet PR-materiale, der omgiver produkterne. Der forefindes tit flere niveauer af software i et release, hvor tredjepartsapplikationer varetager det primære programs mangler – f.eks. demo-udgaver, der vedlægges retail-produkter mhp. reklameværdi. Endnu oftere er disse niveauer ikke harmoniserede. Demo-udgaver varetager, på grund af versionering, ikke funktionerne alligevel, eller situationen hvor et operativsystem blokerer alle funktioner, pga. en demo-udgaves udløb. Formålet med en inkluderende applikation er at etablere en garant for funktion, der

⁷⁵ Jf. interviews med VST-udviklere under *Empiri - domæneindkredsning*

⁷⁶ Jf. John Cages fokus på forholdet mellem musikalske strukturer og musikalske processer, der er behandlet under *Softwareinstrumenter*

tilvejebringer en form karakteriseret som *'nemt og smertefrit'* eller *'easy to use'*. Vores applikation er ekskluderende, da den er højt specialiseret og ikke (nødvendigvis) rettet mod købestærke målgrupper som moderne producere – måske endda ikke-fungerende hos nogle brugere af uforudsete årsager. Vi benytter ligeledes flere niveauer software for at etablere en basal komponentkommunikation, men prøver i så udstrækning muligt at reducere de niveauer, brugeren skal betjene, for at tilvejebringe funktion. Både gennem *kompilering* til en enkelt installationsfil, og gennem brug af komponenter, der allerede er installeret og i brug hos vores testergruppe (MIDIYoke⁷⁸). Den professionelle gruppe er ikke medtænkt i konceptet, da vi ikke arbejder med økonomiske faktorer, og ikke finder det hensigtsmæssigt at rette et æstetisk funderet design mod en professionel brugergruppe. Professionelle vægter i høj grad funktion, som gerne skal vises og bindes til grafik på en konventionel og logisk måde. Gruppen opererer med kriterier som *'tid er penge'*, *'DAW må ikke overbelastes af grafiktunge applikationer'*, *'det skal gøre, hvad jeg forventer det gør'*⁷⁹. Specialiseringen af vores prototype rammer derimod en brugergruppe, der forventer differens som en anden form for eksistensberettigelse. Et design, der berettiger sig ved kontroversielt at binde kendt og ukendt, og tages i brug på trods af prototypemæssige ”barnesygdomme” som tung grafik, svær opsætning eller svært gennemskuelig metaforik. Der findes en forventning om ukonventionelle tiltag, og en anderledes balancering af hvad der vises og hvordan. Ulogiske interfaces bliver taget op som en udfordring, der både overrasker og forvirrer. I brugergruppen er tid ikke penge, men leg der, der kan ende med inspiration til ny musik, eller et *crash*, hvilket dog ikke er katastrofalt for den kreative proces.

Kunst – industri eller manifest

Vores interface ser vi placeret i en brugergruppe af kunstnere, der arbejder med musik, eller mere generelt lyd. For at kunne imødegå og forstå denne gruppe, og den æstetik, der er omdrejningspunktet for alt virke indenfor denne gruppe, anvender vi den samme sociologiske forståelsesramme, som motiverer os til at indsamle empiri. Rammen fokuserer på mennesket i en kontekst, hvor horisonten for kunstneren også fungerer som en æstetisk påvirkningsfaktor. Radikale udsagn (som f.eks *glitch*, serialisering, mv.), der bliver til kendetegnende principper for denne brugergruppe, finder vi ligeledes tydelige i kunstnerens relationer til vedkommendes horisont. Dette afsnit vil derfor se lidt nærmere på kunstnerens tilhørsforhold. Herunder de horisonter kunstneren har, i kraft af at omverden i nogen grad etablerer æstetik som f.eks. retningslinjer mod økonomisk succes. Vi anskuer æstetik som en påvirkningsfaktor, hvor politik i

⁷⁷ Eksempelvis Buzz (<http://www.buzzmachines.com>), Kontakt (<http://www.native-instruments.de>)

⁷⁸ Denne komponent præsenteres nærmere under *Prototype*-afsnittet

⁷⁹ Under et praktikophold hos Native Instruments var dette ledende udsagn for en professionel, kapitalstærk brugergruppe af programmer som Kontakt, Battery og Absynth

systemteoretiske termer ligeledes kan ses som en påvirkningsfaktor. Politik stabiliserer/destabiliserer et socialt system via og vha. information, og ofte ender denne information i en konkret form, f.eks. som en lov, der er baseret på relevant information. Beslutningen påvirker kunstnerens muligheder og er en horisont for kunstneren, og ofte en sondringspol i udformningen af værker og objekter – når kunstneren begynder at placere sin kunst i en kontekst. Her er det relevant at nævne hele fænomenet 'mainstream' som en sondringspol for kunstneren – mainstream er etableret som en modtagergruppe, der utvivlsomt kan nås i kraft af politisk-æstetiske beslutninger. Politik er konkretiseret som f.eks støtteordninger⁸⁰ for kunstneren, eller mediepolitiske beslutninger, der i sidste ende bestemmer, hvad der høres/ses og af hvem, hvilket også påvirker kunsten i hele grupper af kunstnere. Mainstream-kunst er ofte det, der både har støtten og den mediemæssige opmærksomhed. Politik dirigerer i høj grad økonomien i domænet kunst, hvilket der også er tradition for, i kraft af at der findes et ministerium for kunst. Politik og æstetik er i dette afsnit en perspektivering af væsentlige påvirkningsfaktorer indenfor domænet 'kunstnere', da der findes sondringer, der er udslagsgivende for musikeren, og de betingelser, vedkommende arbejder under.

TV⁸¹ giver et indtryk af musik som et produkt via massiv markedsføring, hvor succes sidestilles med salgstal, men det kan næppe overraske, taget i betragtning af hvor desperat industrien er. Desperationen er tydeligst i det faktum, at musikkanaler sjældent sender musik, men fungerer som udstillingsvindue for diverse producenter og markedsstrategier. I takt med at der kommer nye medier at lancere kampagner inden for, bliver musikindustriens udgifter større. Herved sker en gradvis overtagelse af tv- og radios frekvenser og tid gennem aggressiv markedsføring. Udskiftningen af kunstnere i industrien sker i overensstemmelse med nye markedstiltag for at holde pengemaskinen i gang – hvilket efterlader mainstream-musikken som et brug-og-smid-væk-produkt. Mikkel Bolt Rasmussen⁸² konstaterer at (hårdt markedsført) konsumkunst minimerer afstanden mellem kulturindustrien og kunsten, og at denne udvikling eskalerer. Videre at kunstens funktion flytter sig i takt med formålet at tjene penge: "*Avanceret kunst bliver en forpost i erobring af nye subkulturelle markeder*"⁸³.

Uden om de etablerede medier findes der Web tv og Web radio, hvor der ikke findes playlister og rotationer, der er dikteret af selskaber - endnu. Webcasts ser vi som et smuthul uden om industrien,

⁸⁰ KODA (<http://www.koda.dk>), Dansk Musiker Forbund (<http://www.dmf.dk>). En stor liste uafhængige støttemuligheder og diverse legater kan også findes på KODA's hjemmeside

⁸¹ Lokale udsendelser som Musikprogrammet, Boogie, Rundfunk, mv. Og diverse internationale dedikerede kanaler som MTV, VH1, mv.

⁸² Bolt-Rasmussen, 2003, p. 31

⁸³ Ibid., p. 31

der gradvist lukker sig i takt med at industrien ser potentialer og nye markeder. Æstetikken og opfattelsen af musik er her ikke demokratiseret og lyttere vil også flytte sig efter behov. Men muligheden for at lyttere opsøger nye kulturportaler bliver mindre, i kraft af portalernes sponsorater og bidragsydere lukker muligheder i form af PR, opkøb af konkurrenter, playlister, reklameindslag, lobbyisme osv. Kunstens funktion er i denne tænkning det samme som indtægt, i kraft af et økonomisk potentiale i objekterne, der produceres.

Ejvind Hansen⁸⁴ mener at kunstens funktion er abstraktion, og at kunsten bruges til at abstrahere og frigøre sig fra de *daglige basale behov*. Imellem linjerne mener han, at kunsten har et kritisk potentiale, hvilket i høj grad kan diskuteres. Det er relevant at stille spørgsmål til, hvordan det abstrakte skal frigøre kunsten fra de rammer og instanser, der adskiller kunst og ikke-kunst – mere konkret stiller han ikke spørgsmål ved, hvordan værker skal adskille sig fra en institutions (som f.eks. DR) æstetiske selektioner. En institution, der netop er etableret for at varetage daglige basale behov, og varetager disse behov gennem udvalg af noget frem for andet, i henhold til principielle styringer som *public service*. Kunstens funktion kan af denne grund ikke være abstraktion fra de rammer, der eksponerer kunsten, men nærmere adaptation til rammerne/institutionerne. Hvilket i sidste ende påvirker den kunst, der produceres. Kunsten har altid haft en kommentar til den dominerende orden, og blevet brugt som en modreaktion på konventioner – på områder som politik, æstetik, begrebet om det menneskelige osv. Denne egenskab, som et element der søger irritation og modreaktion, er nu anvendt af en decideret kulturindustri. En industri med en økonomisk tyngde, der gør industrien politisk interessant. Kunstens funktion har flyttet sig, men kunstens renommé sikrer stadig industriens kulturkanaler opmærksomhed og eksponering (gennem udstillinger, showcases, medier, sceneplads, mv.), hvilket politikken indordnes efter. Kritiske røster hævder, at kunsten i denne politiske og institutionelle bevægelse er blevet hvermandseje, og det kritiske potentiale af kunsten umuliggjort: ”*Det usynlige er blevet synligt, det uudsigelige og autentiske er forvandlet til kulturelt kendt (...) billedet er trængt ind i alle samfundets kroge og mættet det samfundsmæssige rum*”⁸⁵.

Denne lidt opgivende mentalitet indskrives vi os dog ikke i, da der findes kritisk kunst der stadig provokerer, og der findes kunstnere, der varetager en samfundsmæssig irritation. Der findes stadig innovative veje mellem en ide og hvordan den formes i materiale, og ligeledes kulturelle kanaler at eksponere både objekt og proces igennem. Det er matematisk/kombinatorisk usandsynligt at alt er set og hørt – og at der nogensinde kommer et punkt, hvor det sker. Det bærende punkt er dog at lyttere og beskuere opsøger kunsten, hvilket virker som et slalomløb mellem produkter. En anden

⁸⁴ Hansen, 2003, p. 87

faktor er, at instanser og kulturportaler i højere grad kan anvende kunsten til at abstrahere fra den etablerede industris bud på kunst – og tilbyde et alternativ til en 'samfundsmæssig mætning'. Et valg, som beskueren af kunsten selv er i stand til at foretage i kraft af emergent medieudvikling. Selvom f.eks. webcasts gradvist kommerialiseres, eksisterer der stadig et potentiale for anderledes distribution og dermed et anderledes udvalg for beskuerne. Dette potentiale kan realiseres gennem andre instanser uden kobling til politiske instanser eller økonomiske kriterier. F.eks. græsrodsorganisationer som *netlabels*, uafhængige netmedier, *podcasts*, osv. Et meget relevant eksempel er *netlabels*, dvs. Web-baserede 'pladeselskaber', der udgiver musik som MP3-filer og mere tenderer mod interessefællesskaber om f.eks. bestemte musikgenrer. Deres eksistens er dog under pres, da de opererer som *non-profit* organisationer uden decideret indkomst. Lovgivning om *intellectual property* som f.eks. patentering af MP3-formatet bliver dermed en juridisk og økonomisk faktor, hvis *netlabels* skal købe en licensaftale for at udgive musik. Ideologien bag mange *netlabels* er derfor, at intellektuel frihed er vigtigere end intellektuel ejendomsret⁸⁵. Dette angår såvel udgivelser som den teknologi, som udgivelserne er medieret igennem. Momentet i intellektuel frihed opererer vi ud fra i designet af prototypen. For det første fordi vi ikke kan realisere prototypen uden brug af andre *open source* teknologier (Python, Blender). For det andet ved at anvende viden gennem brug af åbne onlinefora omkring teknologier. Og endelig ved at videregive prototypen i en form, der muliggør modifikation fra brugerens side, og er *open source*.

Hermed er vi tilbage ved den konkrete prototype og de muligheder, den giver musikere. Vi vil i næste afsnit behandle prototypens funktionalitet og matematiske basis. Dette forklarer den teknologiske basis for at tale om modificeringens æstetik.

Funktionalitet og modificering

I afsnittet om systemekstern æstetik behandlede vi begreber om *serialisation*, *scriptzation* og *glitch* som operationelle kriterier for musikeren. Dette afsnit vil behandle den teknologi, der har tilvejebragt disse kriterier. Prototypen er et kontrolinterface, hvori der ligger at der er noget der skal kontrolleres. Afsnittet her vil være med fokus på det prototypen kontrollerer, og altså den lydkilde, interfacet er *mappet* til. Afsnittet tematiserer 'den anden side af designet', da prototypen ikke genererer lyd i sig selv, men skal *mappes* til en lydkilde. Derfor vil afsnittet være tænkt som en

⁸⁵ Bolt-Rasmussen, p. 40

⁸⁶ En væsentlig fortaler for dette synspunkts samfundsmæssige fordele er Lawrence Lessig. Se f.eks. Lessig, 2004

perspektivering, men alligevel beskrive lyd-kilder som et særdeles bevidst moment i designet af prototypen.

Som videoerne viser⁸⁷, ændres klang og tekstur alt efter hvilke MIDI-data, prototypen udsender. Dette er ændringer i lyd-kilden (Reaktor på videoerne), som i afsnittet her vil behandles som modifikation. Modifikationen har i vores udlægning et afsæt i teorier om residualberegninger, der er hentet fra Ph.D. afhandlingen *Adaptive Signal Models: Theory, Algorithms, and Audio Applications* af Michael Mark Goodwin⁸⁸. Afsnittet er en forholdsvis teknisk forklaring af, hvad vi mener med modifikation af lyd.

Residualberegninger

Et lyd-signal kan analyseres matematisk som f.eks. bølgesvingninger over tid, hvilket tanken om residualberegninger også har basis i. For at forklare koblingen mellem matematik og modifikation er der en basisterminologi, der kræver et par linjers forklaring.

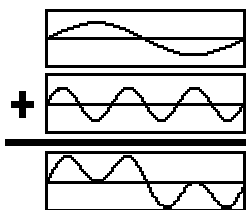
Et signal kaldes $x[n]$. $[n]$ angiver positionen i lyd-signalet. Signalets svingninger kan nedbrydes eller manipuleres med algoritmer, hvor der opstår et nyt signal som følge af transformationen. Det oprindelige signal hedder som sagt $x[n]$, og det modificerede signal kaldes for $\hat{x}[n]$. Forskellen på det transformerede signal og det oprindelige ikke-modificerede signal kaldes et *residual*, hvilket kaldes $r[n]$. Formlen for residualet bliver som følger:

$$r[n] = x[n] - \hat{x}[n]$$

Formlen eksekveres, hver gang et objekt i interfacet bevæges, og lyd-kilden modificerer lyden. Eksempelvis når filtre, der filtrerer noget af lyden fra, skaber et modificeret signal ud af et oprindeligt signal (den ufiltrerede lyd). En anden form for residualberegninger findes i kategorien 'oscilleringsprincipper' som f.eks. additiv syntese. Addere betyder at noget lægges sammen, hvilket egentligt er lettere at illustrere end forklare. To lydsvingninger kombineres, hvorpå der opstår en ny svingning. I fagtermer: en oscillators⁸⁹ svingninger adderes til en sekundær oscillators svingninger. Denne svingningshastighed kan reguleres som nedenstående eksempel på additiv syntese blandt andet illustrerer.

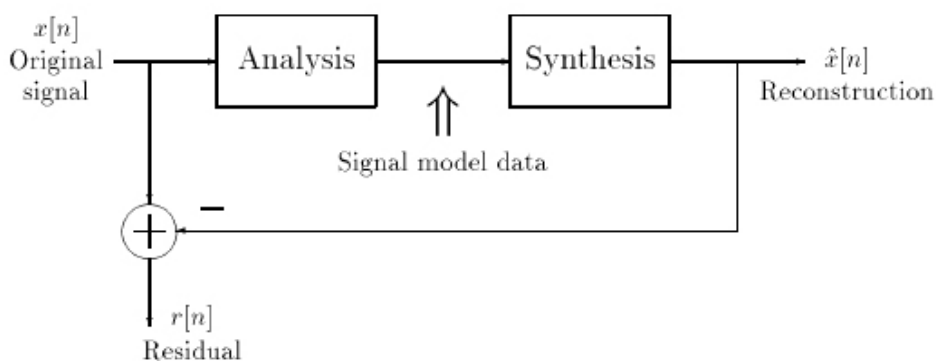
⁸⁷ Vedlagt på bilags-CD'en i folderen *Videoer af prototypen* og findes på http://www.hum.aau.dk/~mlin00/kilog_prototype/

⁸⁸ Goodwin, 1992



Den første oscillators svingning er langsommere, hvilket begrebet frekvens beskriver⁹⁰. Frekvens angiver hvor mange svingninger der forekommer indenfor en hvis tidsramme (440 Hz = tonen A). Den næste oscillators frekvens er hurtigere, hvorefter det residualberegnete signal (facit) bliver et resultat af de to foregående signaler.

$r[n]$ er i eksemplet med additiv syntese et produkt af analyse. Det oprindelige signal analyseres med henblik på at modificere signalet – syntese på baggrund af analysen. Dvs. en proces der går fra analyse mod syntese, hvor syntesen resulterer i at et originalt signal modelleres om, hvilket resulterer i en auditiv forskel.



(Model fra Goodwin, 1992, p. 2.)

Additiv syntese, filtrering af signaler mv., er alle modeller, der beskriver hvad der skal ske med det originale signal, når en analyse af det er komplet. Modeller for hvordan lyden moduleres findes i leddet *signal model data*. Her findes med andre ord information (som matematiske formler) om hvilken proces, det oprindelige signal skal gennemgå. I signal model data findes også et afsæt for

⁸⁹ En oscillator er en lydkilde, der genererer et periodisk signal, hvor f.eks en svingningshastighed på 440 svingninger per sekund er tonen A

⁹⁰ Uden for emnet matematik skal det som en note nævnes at oscillering indgår også i Spencer Browns begrebsunivers, hvor en sondring mellem to former ses som en svingning mellem to tilstande. Denne oscillering foregår også med en frekvens, hvilket han udlægger som hyppighed. Disse svingninger registreres i hukommelsen over tid og genererer information om hvilke former, der er sondret, og med hvilke resultater

glitch æstetikken⁹¹, hvor et modulationsprincip som granulær syntese⁹² findes. Granulær syntese springer ud af ønsket om *cross synthesis*⁹³, hvor principper som additiv og granulær syntese kombineres og hybrider opstår. Granulær syntese foregår ved at foretage 'udtræk' eller 'samples' af en lydkilde, og genindføre disse samples i den originale lyd, som en art *re-entry*. Eksempelvis en 10 sekunders wave fil, hvor der iht. granulære principper samples 1 sekunders bidder, der så videre afspilles med en anden tonehøjde ovenpå den originale lydfil. *Glitch* er relateret til denne proces, da granulær syntese ofte lyder som kontrollerede uheld, hvor bestemte musikalske strukturer ødelægges med en struktureret metode. Leddet *signal model data* rummer altså beskrivelser af de soniske muligheder, der foreligger i arbejdet med signaler. Hvilket har betydelige æstetiske konsekvenser, da modeller for modifikation af lyd i sidste led bestemmer muligheder for musikeren. Algoritmer bliver dermed essensen af hvad det vil sige at kontrollere en lydproces gennem analyse og forståelse af *signal model data*. Dette baner vej for en diskussion af lyd/signaler som et plastisk element, der via *analyse - signal model data - syntese* kan gennemgå radikale forandringer. Brugeren kan også realisere denne treleddede proces i miljøer som Reaktor, Pure Data, Max/MSP, mv. – der alle lader brugeren konstruere *cross synthesis* ud fra egne evner og behov. Udviklingsmiljøerne er på nuværende tidspunkt blevet udviklede og kraftfulde nok til f.eks at lave additiv syntese og køre det igennem et granulært effektmodul. Miljøerne kan afvikles på forholdsvis små computere. Lyden bliver gradvist mere fleksibel sideløbende med denne udvikling, hvor regnekraften stiger og øger mulighederne for større og flere modeller i *signal model data*.

Med denne kortlægning af kontrolmulighederne i lydbehandling vil vi nu flytte blikket fra konkret matematik til abstrakt matematik. Med andre ord: Fra et matematisk-semiotisk domæne til et grafisk-semiotisk domæne. I *fuzzy logic* vil vi anvende formler som abstraktioner over kontrol, og formulere designkriterier ud fra en matematisk relativisering. Relativisering har været et generelt tema for musikerens kreative proces, der hermed gives et formelt grundlag.

Fuzzy Logic

Afsnittet er baseret på udvalgte artikler af Peter Elsea⁹⁴ og vil behandle relativ logik i forhold til vores design. Afsnittet kan betragtes som et videnskabsteoretisk indstik til specialet, da

⁹¹ I denne sammenhæng er der et grundlag for at tale om matematiske principper, der bliver bestemmende for den æstetik, applikationer betjener sig af. Specielt i forbindelsen: givne muligheder der skaber bestemte udtryksformer

⁹² Ophavsmand til granulær syntese er fysikeren Dennis Gabor, der foreslog *grain* som et matematisk princip i 1947. Senere videreført af Xenakis som lydmodificeringsprincip i 1971

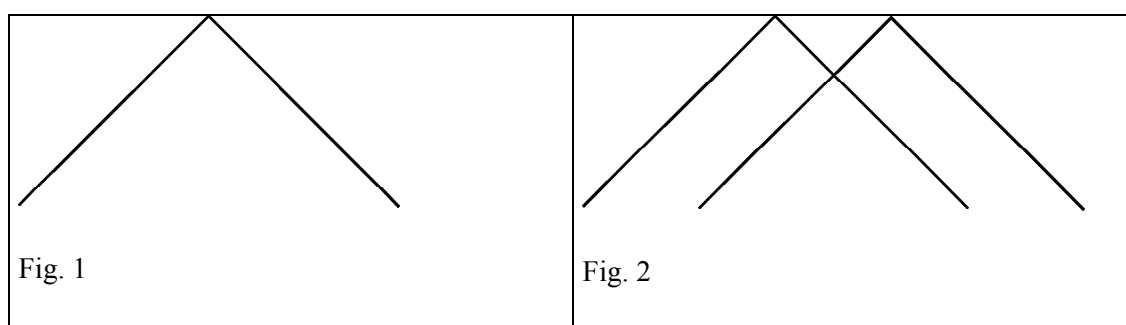
⁹³ Goodwin, 1992, p.8

⁹⁴ Elsea, 1995 og Elsea, u.å.

eksempelvis specialets systemfilosofi afspejler tankerammen relativisering af matematiske kontrolmuligheder.

Logikken vedrører, hvordan talværdier gennem fortolkning kobles til elementer, der ellers forekommer ikke-numeriske, og hvordan definitionsspørgsmål i dette domæne kan repræsenteres og behandles logisk. Arbejdsformen med logikken foregår med matematiske funktioner og formler, der kan omsættes til kode og videre konkretiseres i prototypen. Vi bruger med andre ord også denne logik som en slags kodefilosofi, hvor vi står overfor at konstruere relative gradueringer. Her benytter vi logikken som et værktøj til at nedfælde processen, der definerer eksempelvis gradueringer og de faser der implicit ligger i en graduering. Logikken har sit udgangspunkt i hvordan elementer er forbundet af et kontinuum og behandler i matematisk form sprogligt vage beskrivelser af dette kontinuum, som f.eks forskellen mellem: høj - lille, meget – lidt, hurtigt - langsomt osv.

En samling af elementer med fælles karakteristika kaldes en *mængde*, hvori der videre findes delmængder, alt efter hvor definitionerne foretages. Rummet mellem to mængder skaber en mulighed for et overlap (fællesmængde eller *intersection*), eller en krydsning af to elementer. Denne krydsning fra et element til et andet kan både være et overlap eller en sammentrækning, og skaber via denne kobling en semantisk forbindelse mellem enhederne. Forbindelsen opstår i kraft af medlemskaber som vi definerer og indarbejder i modellen. Mellem elementerne opstår der faser for medlemskab, der sammenfattet kan defineres som en graduering. Faserne fungerer som flydende overgange fra et element til et andet. Flydende forstås som stigende og faldende tilhørsforhold til den foregående og fremtidige fase (Fig. 1):



En serie af faser med stigende og faldende medlemskab kan illustreres som Fig. 2.

Ved at spørge til overlap og sammentrækninger kan enhederne i højere grad afgrænses i forhold til hinanden, og et kontinuum mellem enhederne kan leddes. Her går arbejdsindsatsen på at eksplicite faser ud af førnævnte vage beskrivelser, og fasernes indbyrdes placering i forhold til

hinanden. Leddelingen er en proces mod at etablere definitioner for faser, og hvordan et sæt af faser for det første er et kontinuum, og for det andet en semantisk klasse med yderpunkter, der er graderet gennem stigende/faldende medlemskab. Alle definerede delmængder står derfor i et komplementært forhold til hinanden.

Et projektrelevant eksempel på komplekse prædikater er oversættelsen af tonerækken til tal, hvor et diffust udsagn som 'lige under tonen C' omsættes til talværdier:

C	C#	D	D#	E	F	F#	G	G#	A	A#	H
0	0	0	0	0	0	0	0.3	0.6	0.7	0.9	1

Skemaet er en matematisk repræsentation af en bedømmelse. En bedømmelse af, i hvor høj grad hver tone passer under kriteriet 'under C', og vedrører medlemskaber af de faser, der er defineret mellem tonen C og H.

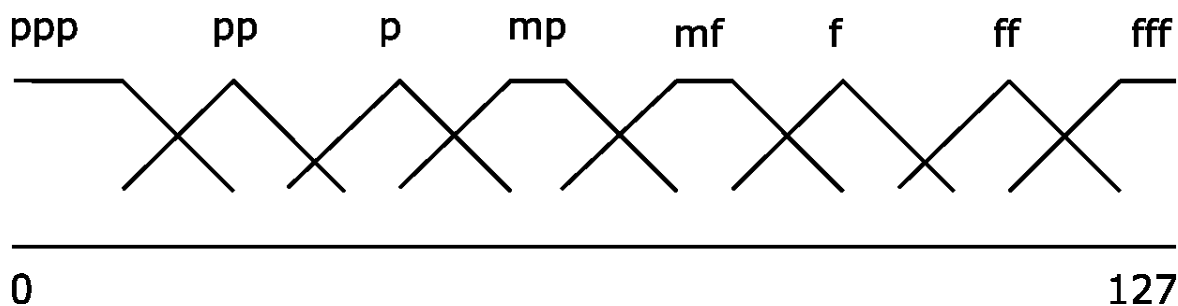
Dette er en graduering, hvor medlemskabet af 'under C' stiger, jo tættere skalaen er på C. Adskillelsen af C og C# er ligeledes en flydende (fuzzy) overgang. Imellem C og C# findes 100

cent, som en flydende overgang mellem semitonerne

(som eksempelvis fig. 2). En instrumentals på en kontrabas har ikke afgrænsede klasser som C, C#, mv., da tonerne ikke er mekanisk definerede. En guitar har f.eks. denne mekaniske begrænsning i

kraft af *frets* (små metalskinner). Bassen betjener sig af flydende, relative medlemskaber mellem tonerne og gør det muligt at spille cent, der ligger mellem tonerne. På guitaren er det dog ikke muligt at spille cent uden at bøje strengen eller stemme den anderledes. Der findes i fuzzy logics termer et stigende og faldende medlemskab mellem tonerne på bassen, hvor det er op til bassistens intonering (definition af klasser som cent og semitoner) at etablere mængder som 'over C', 'under C'. Et andet projektrelevant eksempel er forholdet mellem pianissimo og forte, som er et lydstyrkeforhold, der betjener sig af fortolkning. Lydstyrkeforholdet er baseret på relative termer og individuelle fortolkninger af disse termer, der ligeledes kan opstilles som mængder og delmængder. Her illustreret med stigende og faldende medlemskaber af ppp, pp, p, osv., der markerer flydende overgange:





Eksemplerne er en omsætning af lingvistiske udsagn til en gruppe regler, der foretager differentieringer. Der kan konstrueres en model ud fra disse differentieringer: 'hvis' mere end værdi 0 under C, 'så' er tonen tættere på C, eller 'hvis' mere end ppp 'så' er volumen tættere på fff. Med andre ord formaliseres relative udsagn, som eksempelvis kan omsættes til konkret kode; parametre, regler og klasser. Ud fra et designmæssigt perspektiv benytter vi en model, der behandler værdier som produkter af fortolkning. Hvilket bliver et grundlag for at konstruere en prototype, der læser en brugers interaktion; dvs. en brugerflade, der læner sig op ad en brugers fortolkning i højere grad end en matematisk repræsentation, der begrænser relativiseringen af udsagn.

Relativiseringen af talværdier ser vi i forbindelse med prototypens interface. Interfacets hovedfunktionalitet er repræsenteret uden numeriske værdier og forbinder rumlig placering og klang som to mængder af mulige værdier. Dette bliver til informationen, som den udøvende interagerer ud fra. 'Højt oppe' og 'langt nede' i interfacet er forbundet af et kontinuum, hvor placeringen mellem sdringspolerne eller enhederne ikke kan angives i tal, eller erkendes som stringente talværdier - for brugeren vel og mærke. Derimod er placering i rummet også x,y og z koordinater, hvilket er talværdier i koden, hvilket er en syntese, vi som designere må konstruere. 'Højere end' og 'lavere end' er ud fra et designmæssigt synspunkt klasser, som en enhed kan have stigende eller faldende medlemskab af. Hvordan denne talværdi bruges, er op til brugerens *mapping*. Hvadenten en x, y, z placering skal angive tonehøjde eller toneklang, skubber dette ikke ved kravet om at betjene sig af relative klasser i designfasen. Modificeringen af lyd på en visuel, ikke-numerisk måde, er et designkrav på et interaktionsplan, hvorimod der i samme bevægelse skabes krav om klasser af talværdier på et kodeplan. Overgangene er ikke hårdt opdelt i prototypen, hvor enheder som C og C# ikke bliver mekanisk afgrænsede (som på guitaren), men med mulighed for glidende overgange (som på bassen). Overgangene er videre et nedslagspunkt i vores afgrænsning og definition af *intersections* i forhold til interfacet – de er ikke mekanisk begrænsede. Dette bliver et krav til brugerens operationer og forventning, f.eks intonering, fortolkning af lydstyrke, rytme eller andre musikalske kriterier. Kriterier, der er relateret til brugerens *mapping* af muligheder til interfacet. Et decideret succeskriterium binder sig til forholdet

mellem hvilke funktioner, der er mappet til interfacet, hvordan og med hvilken lethed disse kan betjenes, og det auditive æstetiske resultat, interaktionen tilvejebringer. I specialet anvender vi som beskrevet logikken til at spørge til *intersections* – overlap af medlemskaber eller sammentræk, der markerer en midte i et kontinuum. Dette er for at fremhæve hvordan vi på et metaplan anvender begreber til at bearbejde definitionsspørgsmål ud fra sproglige udsagn, og herudfra etablerer en semantik. En *intersection* optræder i specialet som et konceptuelt kontinuum mellem eksempelvis *modificeret* og *ikke modificeret*. Vi definerer i den bevægelse yderpunkter for eksempelvis modificering af lyd, der skal indarbejdes i designet som et opnåeligt element. Denne bevægelse fra et yderpunkt til et andet skal for brugeren designes som flydende overgange mellem faser med stigende og faldende medlemskab, hvor interfacet understøtter denne fortolkning af *intersections*.

Intersections er som anslået produkter af vores fortolkning, og anvendelsen af semantik i den forbindelse er videre vores konstruktion. For dette kritiseres *fuzzy logic* ofte, men vi har fundet det konstruktivt at kæde lingvistiske udsagn og kode for at eksplicite eller materialisere processen i denne kobling.

Phrasetrækker som eksempel på brugen af fuzzy logic

Mere konkret i forhold til prototypen har vi inkorporeret tankegangen i designet *Phrasetrækker*. I prototypen er relative afstande mellem rytmiske punkter faser, der tilvejebringer en auditiv forandring. Brugers interaktion angiver brugers fortolkning af fænomenet 'rytme', 'taktfasthed' eller 'time' – dette er i både interfacet og i brugeren foran interfacet fortolkninger af input. I *Phrasetrækker* er placeringer i x, y, z-positioner videre faste punkter med stigende og faldende medlemskaber indbyrdes. Strækningerne mellem fem punkter angiver hver især en glidende overgang mellem parameterindstillinger, hvilket vi ser som realisering af *intersections*.

Fuzzy logic er det sidste teoretiske udgangspunkt for fastlæggelsen af en æstetik og funktionalitet. Emnet for næste afsnit er en formulering af et framework, der indfanger kernepunkter i musikeres arbejds mønstre, som igen leder til formuleringen af scenarier.

Frameworks og dialektikker – indkredsning af arbejdsprocesser

Vi studerer arbejdsprocesser og vaner hos musikere for at kortlægge og analysere den aktivitet og oplevelse, vi designer mod. Ikke kun i form af empiriske undersøgelser, men også gennem en teoretisk kortlægning, der søger afgrænsningerne og nogle væsentlige generelle karakteristikker for den musikalske arbejdsproces. Formålet med dette afsnit er at præsentere denne kortlægning, der sammen med empiriske udsagn danner basis for formuleringen af scenarier og *use cases*.

Først vil vi afklare forudsætningerne for den teoretiske kortlægning af musikerens arbejdsproces. Vi forudsætter for det første, at musikerens *motivation*, der driver den kreative proces fremad, er individuel og svær at inddrage direkte i designprocessen pga. dens forankring i de enkelte psykiske systemer. F.eks. kan denne motivation være forankret i et kritisk samfundsperspektiv, en nysgerrighed, en lyst til det ærlige udtryk, trangen til at skabe det originale og uhørte, eller et ønske om berømmelse og karriere indenfor musikken.

Men vores design angår nærmere mødet mellem *ide* og *materiale* end mødet mellem motivation og ide. Ide forstår vi som den musiske ide, som musikeren gerne vil udtrykke, mens materialet består af de tilgængelige instrumenter, host-software, samples, osv. I dette møde kommer kreativiteten hos musikeren i spil som en form for kompleksitetshåndtering, hvori bestemte elementer i materialet udvælges til fordel for andre, hvorigennem ideen langsomt gives en bestemt form fremfor andre. Vores design ser vi dermed som en udvidelse af det forhåndenværende materiale, der kan anspore en anderledes arbejdsproces og dermed give kreativitetens kompleksitetshåndtering en anden form.

Uanset den enkelte musikers individuelle motivation og æstetiske præferencer indkredser vi derfor en vekselvirkning mellem ide og materiale, der går igen på tværs af forskellige kunstneriske perioder, genrer og skoler. Ideen kan være bundet til f.eks. etablerede former (som koralharmonik eller jazz-standards), koncept- eller kritisk kunst forankret i værdisætninger, eller en nysgerrighed efter 'hvad denne plugin kan gøre'. Uanset grundlaget må ideen gives en konkret form i et bestemt materiale, før den er hørbar for andre. Uden at basere os på bestemte æstetikker eller motivationer kan vi dermed identificere nogle dialektikker eller vekselvirkninger i vejen fra ide til musik. Disse vekselvirkninger udgør tilsammen et begrebsmæssigt framework, der sammen med empirisk information kan omsættes til konkrete scenarier og *use cases*, ud fra hvilke vi kan designe mod bestemte interaktionsformer og aktiviteter. Vi vil her præsentere disse vekselvirkninger og ud af dem fremanalysere et grundlag for vores scenarier og *use cases*. De næste tre dele præsenterer disse dialektikker.

Prækomposition – komposition – performance

En vigtig vekselvirkning i den kreative proces er skiftene mellem forskellige faser af musikkens tilblivelse. Prækomposition beskriver struktureringen af et *setup* og basal udvælgelse af soniske materialer. I komposition bringes dette *setup* i spil (*execution*) og soniske materialer indordnes i komponistens prædefinerede struktur. Endelig kan *setup*'et blive en improvisatorisk platform i performance, hvor strukturer genoptages og udvides *live* foran et publikum.

I performance-situationen bliver tidligere komponerede temaer, osv. realiseret af musikeren efter mere eller mindre stringente æstetiske præferencer. Udover prototypiske eksempler som opførelsen af et klassisk værk eller en jam-session kan vi her også pege på aktiviteter knyttet til den elektroniske musik som *live mixing*⁹⁵, *tweaking*⁹⁶ og arrangering. Disse aktiviteter indebærer et krav om *realtime* kontrol; en umiddelbar auditiv konsekvens i interaktionen med instrumentet, der fastlægger at musikken forandres og opleves lige nu og her. Denne samtidighed er som oftest givet i brugen af akustiske instrumenter, men ikke determineret, når computeren anvendes som instrument. For det første pga. tekniske begrænsninger (*latency*⁹⁷) og for det andet fordi musikken kan bestå af allerede indspillet materiale såvel som lyd, der skabes i nuet.

Dette sidste hænger sammen med de aktiviteter, der foregår i såvel kompositions- som prækompositionsfasen. I kompositionsfasen indgår indspilning af musiske elementer, der kan manipuleres, genanvendes og kombineres i performance-situationen. *Sequenceren*, der er et typisk værktøj i kompositionsfasen, fungerer ikke blot som notepapir eller båndoptager, men antager karakter af et særegent værktøj med muligheder for at skrive musik såvel som at mixe, manipulere og kombinere i performance-situationen. Og i prækompositionsfasen skabes de materialemæssige rammer for komposition og performance, bl.a. gennem konfiguration af sequencer og plugins, design af nye lyde, *samples* og *patches*⁹⁸, og konfiguration af computeren som værktøj for komposition og performance. I disse faser er kravet om realtime-kontrol oftest mindre udtalt, mens kravet om kontrol med lyd-mæssige detaljer såvel som det overordnede musikalske forløb er større. *Mappingen* fra interface til lyd er en del af prækompositionens fokus på konfiguration, og foregår ofte i studiet som en foregribelse af de muligheder, musikeren ønsker at give sig selv i kompositions- og performance-øjemed.

Således griber disse faser ind i hinanden, og de behøver ikke at følge hinanden lineært kronologisk. Kompositionsfasen kan f.eks. gribe tilbage til prækompositionen, når vekselvirkningen mellem ide og materiale fordrer det, og performances kan danne erfaringsgrundlag for en ny prækomposition. Faserne identificerer imidlertid forskellige arbejdsmodi på tværs af genrer og æstetikker, og peger

⁹⁵ Justering af volumen, balance og equalizer for hvert element i kompositionen, oftest på en dedikeret hardware-mixer eller i software. Mixing bringer musikkens elementer i sammenhæng med hinanden og kan varieres henover et musik forløb

⁹⁶ Justering af instrumenter og effekter, således at klangfarver ændres undervejs i musikken. Et klassisk eksempel er techno-musikkens *tweaking* af Rolands TB303-synthesizer

⁹⁷ Latency er tiden fra en tangent anslås eller en parameter ændres, og til ændringen rent faktisk kan høres. Ofte udgør denne tid få millisekunder, men alligevel kan en for stor latency virke hæmmende i live-sammenhæng, da sammenhængen mellem anslag og resultat brydes

⁹⁸ Ordet *patch* dækker over en konfiguration af lydmateriale, f.eks. en lyd i en sampler, der er opbygget af flere samples, filtre, envelopes, osv. En *patch* kan også være et instrument eller en effekt opbygget i et modulært lydprogram som f.eks. Reaktor eller Pure Data

på væsentlige behov i den kreative proces' forskellige faser. Mødet mellem ide og materiale udspiller sig på forskellig vis i disse tre faser, og antyder som beskrevet en dynamik, hvori musikeren bevæger sig på tværs af faserne.

Makroniveau og mikroniveau

En anden væsentlig vekselvirkning i musikerens arbejde med materialet er skiftene mellem et makro- og et mikroniveau. I modsætning til prækomposition/komposition/performance er disse niveauer bundet til strukturer i materialet, fremfor en tidslig akse i den kreative proces. F.eks. finder vi en vekselvirkning mellem makro- og mikroniveau i skiftet mellem redigering af enkelte lyde og sammensætningen af disse lyde til en større helhed. Eller i skiftet mellem fine justeringer i sekvenser, fraser, etc. og sammenkædningen af disse sekvenser til hele værker.

I prækomposition såvel som komposition er denne vekselvirkning tilgængelig som en kontinuerlig tilbagevenden med mulighed for justering af tidligere kreative valg. I performance-situationen er den derimod underlagt kravet om *realtime* performance. Konsekvensen kan blive, at mange valg på såvel mikro- som makroniveau fastlåses og at kun en delmængde af disse valg gøres tilgængelige for ændringer i opførslen af et værk. F.eks. kan et værks sammenkædning af sekvenser fastlægges på forhånd, således at musikeren i performances i stedet kan koncentrere sig om live *mixing* og *tweaking*. Eller værkets sammenhæng kan brydes op og gøres tilgængelig for ændringer, eksempelvis i henhold til musikerens læsning af et publikum eller egne præferencer.

Vekselvirkningen beskriver en skiftende fokusering på forskellige elementer i det musiske materiale, og er således tæt knyttet til de parametre eller egenskaber, som den enkelte musiker finder det interessant at arbejde med i forskellige situationer.

Forventning – overraskelse

Endelig opererer vi med en veksel mellem forventning og overraskelse, der egentlig beskriver kontingensen i mødet mellem ide og materiale. På trods af mål og kriterier kan der i mødet med det digitale materiale opstå *glitches*, osv., der vækker genklang hos musikeren, selvom de opstår uventet. Forventninger forskydes dermed, og andre musiske former opstår. Vi ser en glædelig overraskelse som et element, der sætter en iterationsproces i gang, drevet af motivation.

Eksempelvis er software-instrumenter og –effekter ofte komplekse og består af mange parametre, der virker sammen i skabelsen eller ændringen af en bestemt klangfarve. I modulære miljøer er det endda muligt kontinuerligt at ændre signalvejene og antallet af parametre. Derfor kan det være svært at forudsige konsekvenserne af specifikke parameterændringer, og endnu sværere at modellere en bestemt klangfarve med et softwareinstrument. I udforskningen af parametre mødes

musikerens ide med det digitale materiale, men forventninger forløses ikke altid umiddelbart i softwaren, hvilket kan afstedkomme en anden forventning om at ville overraskes. Denne nye forventning kan forløses gennem *randomization*⁹⁹ eller andre former for autonome parameterændringer, der dog kan være underlagt en form for kontrol med det uforventelige (f.eks. begrænsninger på graden af tilfældighed).

Vekselvirkningen mellem forventning og overraskelse muliggør kort sagt 'held i uheld', hvor andre musiske former med relevans for den enkelte musiker opstår som brud på umiddelbare forventninger. Denne vekselvirkning kan understøttes i software f.eks. i form af autonomi, hvor den eksplicite kontrol med hver enkelt lydegenskab overlades til mere eller mindre forudsigelige processer. Processerne er helt eller delvist uafhængige af musikerens input. På denne måde åbnes vejen fra ide til materiale for selvvalgte forstyrrelser, og materialet får mulighed for at tilbagevirke på ideen, der kan forskydes i takt med at 'forventningen om overraskelser' indfris.

Frameworket og dets grundlag

Med dette framework af forskellige vekselvirkninger beskriver vi nogle udvalgte dynamikker i musikers bevægelse fra ide til materiale. Formuleringen af frameworket tjener som basis for formuleringen af use cases og scenarier. Dialektikkerne kan ses som et forstudie til disse, da de indkredser spændvidden i de anvendte scenarier, og dermed indirekte selekterer visse scenarier fremfor andre. Med andre ord er frameworket en eksplicitering af vores afgrænsninger for designets anvendelsesområde med basis i vores egne erfaringer som musikere. Beskrivelsen af vekselvirkningerne bygger på disse erfaringer, og kan læses som systematiseringer af begreber, vi finder anvendelse for i vores virke som musikere. F.eks. er det væsentligt at kunne kommunikere musikere imellem om hvilken 'fase', man netop nu arbejder i, om det forhåndenværende materiale lever op til ens forventninger, osv. Begreberne er altså knyttet til en sproglig praksis, og i særdeleshed det semantiske indhold i begreberne. Vekselvirkningerne fungerer som kortlægninger af denne semantik. Gennem distinktionen mellem forskellige yderpunkter fremdrager vi ny information, der danner grundlag for et anvendeligt framework, der igen fungerer som afklarende informationsgrundlag for formuleringen af scenarier og use cases.

Disse tjener som udlægning af den oplevelse, vi designer mod, og er således en del af vores *requirements*. Den mere konkrete forankring af denne oplevelse i konkrete software-objekter og interaktionsmuligheder beskriver vi gennem *use cases*, og da dette angår et så konkret konstruktionsniveau, har vi henlagt opstillingen af use cases til kapitlet *Prototype*. Vejen fra

⁹⁹ Generering af tilfældige parameterværdier eller inputs til et software-instrument eller -effekt.

framework til scenarier indikerer en understøttelse af særlige interaktionsformer med det musiske materiale, hvorved vi vælger nogle former og fravælger andre. Vejen fra scenarier til *use cases* udspecificerer de konkrete *requirements* på et niveau, hvor deres implementering i en prototype bliver det naturlige næste trin.

Scenarier – skitsering af interaktionsformer

Tweaking/editering af parametre

For at udvide potentialet for tweaking vil vi give musikeren mulighederne for

1. at sætte en tweaking-proces i gang og vende tilbage til den senere i et musisk forløb, hvormed computerens muligheder for brugerinput via mus og tastatur frigøres til andre aktiviteter i f.eks. sequencer eller plugins, og
2. at operere med flere eller mange parametre ad gangen.

Når en proces sættes i gang, betyder det at yderpunkterne for et bestemt 'tweak' fastlægges af musikeren, hvorefter softwaren tilvejebringer den faktiske ændring af parametrene. Med andre ord fastlægges en art 'frase' af musikeren, som computeren så reproducerer. Justeringen af parametre over tid er tit forbundet til det basale tempo i musikken, hvorfor det er relevant at give brugeren muligheden for at justere hastigheden på en given frase.

Endelig giver håndteringen af flere parametre samtidigt musikeren mulighed for at relatere flere lydparametre til hinanden i den samme frase. Dette skal ses i sammenhæng med at bestemte elementer i lydsoftware ofte kræver mere end et parameter i deres repræsentation. F.eks. vil et filter i en software-synthesizer være karakteriseret ved såvel dets *cutoff*, dets resonans og evt. også flere egenskaber. Set som instrument vil det i tweaking-situationen derfor ofte give mening at kunne relatere disse egenskaber visuelt og interaktionsmæssigt.

Kontrolleret kaos

For at imødekomme en 'forventning om overraskelse' vil vi gerne skabe et interface, der rummer såvel uforudsigelighed som muligheden for delvis kontrol. En metafor for dette kan være simuleringen af den fysiske verdens egenskaber som f.eks. tyngdekraft, kollisioner og bevægelse. Selvom sådanne fænomener virker meget forudsigelige på forhånd, ser vi nemlig muligheden for at instrumentet bliver mere komplekst, når flere simulerede objekter eksisterer i samme miljø. At objekter falder ned er ganske forudsigeligt; at deres bevægelse vokser i kompleksitet med antallet af kollisioner, osv. ligeså – men de specifikke bevægelser bliver mindre forudsigelige i takt med at

kompleksiteten stiger. Disse bevægelser kan vi omsætte til lydændringer, hvormed musikeren gives muligheden for at operere visuelt med et kontrolleret kaos, der kan bindes (*mappes*) til selvvalgte lydparametre.

Musikeren kan indvirke på simuleringen af objekter ved at 'slå' til dem, altså gennem en simuleret interaktion mellem et musiker-styret objekt og objekter, der reagerer på tyngdekraft og kollisioner. Grundet kompleksiteten i den fysiske simulering er kontrollen kun delvis og sker på simulationens præmisser. Dette kan opleves som en vis autonomi i den kreative proces, hvorved computerens potentiale til 'held i uheld' frigøres, uden at musikeren nødvendigvis føler sig udenforstående i materialets forandringer. Tværtimod kan denne mulighed for overraskelse imødekomme en skriveblokade hos musikeren, idet nye klangfarver eller fraser opstår, eller mødet med et anderledes instrument giver anledning til nye ideer.

Konfiguration og sammenkædning

Endelig indgår der i såvel prækomposition, komposition og performance en mulighed for at tilpasse softwaren de arbejdsmønstre og det setup, den enkelte musiker arbejder med. Konkret kommer dette til udtryk som muligheden for 1) at vælge MIDI-output port og kanal for softwaren, og 2) at vælge MIDI-controller-numre for de forskellige objekter, brugeren kan manipulere med i vores interfaces. På makro-niveauet for prækomposition kan musikeren dermed vælge den *host/shell* eller den hardware, han/hun ønsker at sammenkæde med interfacet. Og på mikroniveau for prækompositionen findes der en mulighed for at skifte mellem forskellige lydparametre ved at justere MIDI-controller-numrene, hvorved konfigurationen af interfacet kan ændres af musikeren. Interaktionen med det musiske materiale beror også på denne udvælgelse hos musikeren, som vi gerne vil gøre 1) tilpas let og 2) mulig at gemme fra gang til gang, således at interfacet ikke skal konfigureres på ny for hver opstart.

Use cases – interaktionsformernes forankring i funktionalitet

Den endelige skitsering af funktionalitet og oplevelse sker gennem *use cases*, der er en række 'mikro-scenarier' med udgangspunkt i vores framework og scenarier. Hver enkelt *use case* udtrykker betingelser eller egenskaber ved vores konstruktion, der skal være tilstede for at opfylde vores krav til funktionaliteten, og dermed understøtte en oplevelse. Udover de umiddelbare *use cases*, der kan siges at udtrykke 'positive' brugssituationer, formulerer vi her også en række *abuse cases* eller negative *use cases*, der skitserer situationer, som ikke må forekomme i brugen af

interfacet, fordi de bryder oplevelsen eller funktionaliteten. Med andre ord udtrykker en *use case* 'what a user should be able to do', mens *abuse cases* udtrykker 'what should not be possible'¹⁰⁰.

Som antydnet under behandlingen af vores begrebsmæssige framework er de forskellige vekselvirkninger forhold, der understøtter og overlapper hinanden. Når vi her ser på de forskellige yderpunkter i isolation er det for at samle en række krav til designet og samtidig forholde os til designet på et konkret mikro niveau. Hensigten er, at de forskellige aspekter af oplevelsen understøttes som en helhed i den endelige prototype. Samtidig forholder opstillingen af *use cases* sig til de eksplicitte scenarier, vi har formuleret som hypoteser om den musiske proces.

Positive use cases

Mikroniveau-editering: Musikeren kan kontrollere enkelte parametre ved enkelte enheder med interfacet. Konkret kan f.eks. et filters parametre eller tonehøjde ændres. Den eksplicitte kontrol med lydmaterialer er forankret i *tweaking*-scenariet og kræver muligheden for direkte manipulation med interfacet. Dette betyder i praksis at muse-input omsættes umiddelbart til positioner og værdier i interfacet ud fra en interaktion, der er modelleret ud fra eksisterende interfaces brug af mus.

Makroniveau-editering: Samtlige parametre i en komposition kan editeres med interfacet. Konkret som tempo i en sang, samtlige lydniveauer for kompositionen, kompositionens forløb eller struktur. Dette har vi tidligere benævnt skelet.

Prækomposition/udforskning: Primær aktivitet; Strukturering af processer og opbygning af rammer for komposition og performance. Konkret mapping af parametre til visuelle elementer, design af deres gensidige relationer. F.eks. tyngdekraftens påvirkning på udvalgte parametre. Afdækning af relative værdier hvor musikeren udvælger poler, centre og kontinua mellem 'højt' og 'lavt'. Denne aktivitet ser vi foregå i studiet. I denne kontekst er kriterier som stabilitet, interface og usability nedjusteret i forhold til soniske muligheder. Aktiviteten foregår som et selvvalgt overlap mellem mikroeditering og mapping.

Performance/oplevelse: Realisering af processer mhp. at nå et mål og sætte påvirkninger i gang mellem elementer (parametre og interface-elementer). Konkret: Kaste, ryste elementer eller starte bevægelser; et eller flere kontinua tages i brug. Egenskaber som 'slid', 'glitch', 'tyngde' anvendes i forhold til decideret klang og klangens placering i en musikalsk struktur. Dialektikken mellem tekstur og struktur høres i kraft af at interfacet med sine egenskaber (slid, glitch, tyngde) forskyder

¹⁰⁰ Booch, Jacobson, Rumbaugh, 1999

forholdet mellem de to, i henhold til brugerens valg. Gestik, som eksempelvis musebevægelser, aflæses af systemet og danner basis for informationen mellem bruger og interface.

Tekstur: Lydes iboende potentiale for modificering kan afdækkes med forskellige metoder. Interfacet faciliterer enten en kontrolleret metode eller en semi-kaotisk tilgang til dette potentiale. Ligeledes åbnes klangmuligheder der er svære at programmere. Interfacets output er ikke lineært i kraft af tyngdekraftalgoritmer, gestik aflæsninger, der skaber eksempelvis stigende og faldende (tal) kurver, eller matematisk kaotiske figurer. Alligevel er interfacet baseret på gentagelse, der skaber en overordnet figur, eller forventelighed – en tendens.

Struktur: Kompositioners overordnede potentiale for modificering. Brudstykker og andre elementer kan organiseres gennem interfacets ”puls” – i kraft af metaforikken omkring interfacets klasser som: tyngdekraft, bruger aflæsning, taktile objekter. Og metaforikkens sammenhæng med en grad af forventelighed; mønstre gentages selvom de er kaotiske.

Forventning og overraskelse. Designidealet er at etablere et rum hvor mødet mellem det kaotiske og det strukturerede er muligt, som tilgængelig metaforik for brugeren. Metaforen binder visuel information og auditiv information, som en sansemæssig redundans. En redundans der alt afhængigt af brugerens læsning etablerer information mellem bruger og interface. Brugeren kontrollerer givne mønstre for modifikation, men giver gennem brug af interfacet slip på kontrollen. Hvor og hvad dette kaos skal modificere er igen under brugerens kontrol. Hvilket ligeledes er et kompositorisk element brugeren kan give slip på. Et scenarie er en skriveblokade, hvor overraskelser kan forstyrre den kompositoriske proces. Dette forudsættes som hensigtsmæssigt hvis forstyrrelsen i nogen grad er under kontrol.

Abuse cases – negative use cases

For det første opererer vi her med de tekniske forudsætninger for prototypen, hvor en række betingelser skal være opfyldt før softwaren virker. Prototypen baserer sig på en række komponenter, der arbejder sammen, hvilket forudsætter at denne interne kommunikation rent faktisk finder sted. Ofte må disse *abuse cases* imødegås gennem constraints i softwaren eller manualer, da softwaren ikke er programmeret til at give feedback om fejl. De umiddelbare tekniske faldgruber er:

- Protokolproblemer:

Hvis ikke enten fysiske eller virtuelle MIDI-porte er installeret på computeren, kan vores prototype ikke generere sit output, og vil fejle med en fejlmeddelelse om dette.

Afviklingen af 3D-interfacet kræver en Blender *player*, der fungerer analogt til f.eks. Shockwave/Flash-playeren. Denne player kræver igen en form for 3D-understøttelse på computeren, der er faciliteret gennem OpenGL-standarden. Findes dette ikke, vil playeren fejle under opstart.

- Installationsproblemer:

Vi konstruerer en standard Windows installer (setup.exe), der indeholder de fornødne filer samt instruktioner om hvor disse skal kopieres hen på computeren, etc. Denne installer kan dog ikke af sig selv installere MIDIYoke (virtuelle MIDI-porte). Dette skal brugeren desværre selv gøre, hvilket udløser en længere installation med diverse advarsler og spørgsmål fra Windows undervejs. Hvis dette går galt, er kun brugen af hardware-MIDI-porte tilgængelig, hvilket svækker anvendeligheden af softwaren i et rent software-setup.

Endvidere kan installeren være behæftet med fejl, f.eks. mangle nødvendige filer eller installere filer til forkerte lokationer. Sådanne fejl vil dog oftest ikke være synlige for brugeren i form af andet end et ufungerende program-ikon, og kræver en robust installer fra vores side.

- Kompatibilitet:

Endelig kan der efter installationen og opstarten af softwaren indtræde kompatibilitetsproblemer. Softwaren kan principielt kobles til mange forskellige hosts og hardware-komponenter via MIDI-protokollen, men ikke alle modtagere af MIDI-signaler behandler deres input ens. Dette kan vise sig som overbelastninger af host eller hardware, eller ved at MIDI-kommunikationen slet ikke etableres.

Endvidere kan vi løbe ind i versionskonflikter mellem allerede installeret software og de komponenter, der indgår i vores software. MIDI-komponenten

- Uforståelig metaforik. Sammenhængen mellem auditive og visuelle informationer misforstås.
- Uforståeligt systemkoncept. Brugeren konstruerer ikke et praktisk bindeled til systemets tiltænkte funktion.
- Brugeren misforstår arkitekturen, dvs. den rent tekniske kontekst programmet indgår i. Eksempelvis forbindelsen mellem tredjeparts komponenter og interfacet. Misforståelsen kan også være et uafdækket potentiale, hvor muligheden for at anvende flere computere ikke opdages.

- Latency mellem interface og host. Rytmask anvendelighed i forhold til at *sequence* interfacet forsvinder.

Disse use cases lukker vores empirisk-metodisk-teoretiske behandling i *Design* kapitlet. Den hidtil akkumulerede mængde designrelevant information har på dette punkt en tilstrækkelig tyngde til at kunne omsættes til en konkret form. Denne transformation beskrives i næste kapitel *Prototype*.

Prototype

Arkitektur

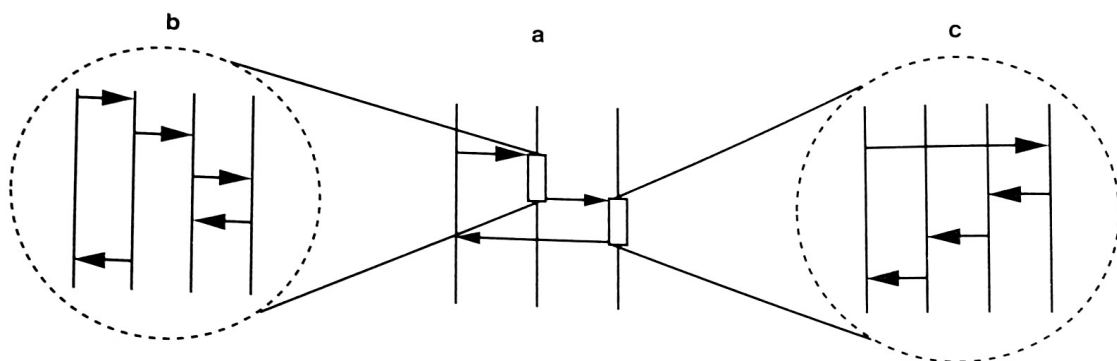
Prototype – iterativ udvikling af arkitektur

Konstruktionsfasen har været en iterativ proces, hvilket afsnittet *Iterationer* var et teoretisk grundlag for. Følgende afsnit er en videre pragmatisk af tankerammen, og udlægning af hvad der processuelt sker i et designende system når relevant information behandles. Afsnittet vil se på hvad der sker mellem to iterationer, dvs. hvilken 'forøgelse', 'formindskning' eller ændring, hver iteration medfører. Ændringer som kan spores i både design, metode og teori, hvilket alle er iterativt udviklede fundament for specialet. Afsnittet er en behandling af prototypens arkitektur som en beskrivelse af denne proces, og som resultat af i forvejen udviklede elementer.

Arkitekturen

Vi har i vores udvikling taget udgangspunkt i use cases, empiri, sondringer af designets form mv. De designmæssige kernepunkter, der iterativt udvindes af denne information, er ikke statiske i kraft af at et givent udviklingsdomæne er under konstant forandring (tendenser). Eksempelvis når brugsmønstre eller andre behov ændres i musikergruppen i takt med at ny teknologi introduceres. Forandringer skaber et behov for iterativt at redefinere grundlæggende elementer som eksempelvis koncepter i takt med at miljøet uden om designet transformeres. For at eksplicite forholdene mellem elementer og elementernes forandring anvender vi modeller for eksempelvis konceptdefinition, analyse, syntese mv. Modeller, der er abstraktioner af elementer, og som indgår i en overordnet abstraktion af et design. En designabstraktion indbefatter også det forhold, der findes mellem modeller som eksempelvis analyse og syntese. Disse forhold benævnes i *United software development proces*¹⁰¹ som *trace dependencies*, hvilket kan oversættes som en sporing af de afhængighedsforhold, der findes mellem modeller.

¹⁰¹ Ibid., p. 23



Dette er en abstraktion af arkitekturen, hvor der i designsystemet [a] findes subsystemer som [b] og [c]. I subsystemer findes trace dependencies mellem eksempelvis metodiske/teoretiske/empiriske lag af designet eller mellem forskellige lag i softwaren, som konkret komponent kommunikation mellem *Blender* og *MIDIYoke*.

Forhold mellem modeller virker specialiserende for udviklingsprocessen som helhed, og er retningsgivende for metodiske og designmæssige prioriteringer på alle trin. Givet at to udviklingsprocesser ikke er ens, må den fortløbende specialisering af processer være justerbar. Denne løbende rekonfigurering af modeller er videre et argument for at arbejde iterativt, da f.eks. tekniske, empiriske, æstetiske eller andre domænefaktorer som nævnt forskyder det faktiske designproblem.

Modeller leder mod arkitekturen, for det første fordi både modeller og arkitektur opstår igennem designrelevant information, for det andet fordi en arkitektur rummer et stort antal abstraktioner af systemet. Abstraktioner, der for det første er indbyrdes funktionelt koblet (som forhold mellem analyse og syntese), og for det andet hænger semantisk og tematisk sammen. Arkitekturen ser vi som en sammenfatning af disse abstraktioner og en konkretisering af dem, da vi i kodeskrivningsprocessen omsætter sproglige udsagn til eksekverbar kode. En arkitektur producerer ifølge *United software development process* use cases, der er afledt af tidligere erfaringer med software, hvilket vi i specialet udvider en smule. Arkitekturen i prototypen er påvirket af mange informationskilder, interne som eksterne, og ikke kun af egne erfaringer med software. Use cases er også formuleret ud fra konceptet som noget af det første i udviklingen af prototypen. Koncepter ser vi som forløberen til arkitekturen, da koncepter opsummerer informationen fra de kilder, der påvirker designet. Vi ser arkitektur som broen mellem koncept og konstruktion. For at blive i metaforen, en bro der krydses mange gange i kraft af vores iterative arbejdsform. Tidsligt arbejder vi også flydende i forhold til udvikling af f.eks. metodens relation til designet. Arbejdsopgaverne har på områderne ikke en decideret start og slutfase, men foregår som udvikling,

der sker sideløbende med at prototypen tager form. Den manglende stringens på dette område ser vi som en 'åbenhed' overfor den information en prototype tilvejebringer. Vi ser denne differentiering som den primære forskel på design af en prototype og en endelig version af softwaren. Information er prototypens styrke og primære formål i forhold til udviklingsprocessen – at understøtte en formulering af arkitekturen, som en sammenfatning af ad hoc udviklede modeller.

Konstruktion

Teknologi

Udvælgelsen og kombinationen af forskellige software-komponenter er kendetegnende for den form, vores prototype antager. Udvælgelsen er i første omgang sket ud fra de muligheder, vi har som studerende med begrænsede budgetter til indkøb af ny software, bøger, osv. og tid til at lære nye teknologier at kende. På grund af prototyping-processens karakter af design-og-smid-væk-proces (rapid prototyping) er større investeringer af såvel tid som penge heller ikke retfærdiggjort i forhold til prototypens endelige form. Hovedformålet med prototypen er at generere mere designrelevant information ud fra kontrollerede eksperimenter med teknologi, hvor informationen gendrives med opfølgende evalueringer – empiriske såvel som interne evalueringer i designgruppen, ud fra konceptuelle eller æstetiske kriterier.

Udvælgelsen sker endvidere med reference til de forskellige domæner, vi søger at kombinere og få til at kommunikere i vores prototype, dvs. hhv. et visuelt, et interaktionsmæssigt og et auditivt domæne. Dette er benævnt trace dependencies i afsnittet om arkitektur, hvor forhold mellem givne modeller kun kan fungere via gensidig informationsudveksling. På et softwareniveau vil det sige, at processen i at modificere lyd er distribueret over flere programmer eller komponenter, der varetager forskellige domæner (hhv. visualisering, interaktion og lydbehandling). Denne fragmenterede distribution over flere eksekveringsplatforme¹⁰² kan betyde synkroniseringsproblemer. F.eks. kan *Blender* som en eksekveringsplatform være ude af synkronisering med platformen *Python* på grund af kodefejl, hvorved processen fejlesekveres og stopper. Når et givent problem opstår, påkalder det en kodemæssig iteration for at rette synkroniseringsproblemet. Det sker som en justering af platforme for processer, og en form for designmetodisk mekanisme, der udløses når problemer verificeres¹⁰³. Problemer, der har med flere platforme at gøre, kan anskues som generiske problemer, da de inkluderer mange sæt forhold i prototypen. Den samlede mængde af forhold kalder vi arkitektur, og anser generiske problemer for

¹⁰² Vi præsenterer disse platforme eller kerneteknologier nedenfor

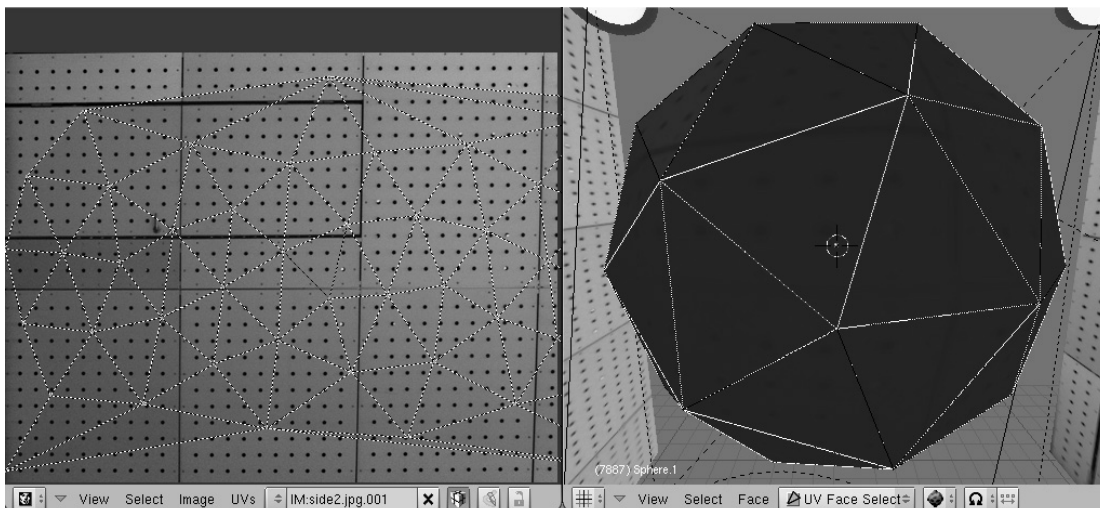
¹⁰³ F.eks. ved at problemet reproduceres på en anden computer, et fora gør opmærksom på en fejl, eller en bruger oplever noget lignende

at være relaterede til arkitekturen og dens mange facetter. Generiske problemer ser vi løst ved at iterativt konstruere fungerende interfaces mellem platformene. Disse designinterne interfaces mellem platforme vil i det følgende diskuteres i forbindelse med vores primære teknologier: *Blender, Python og MIDI*.

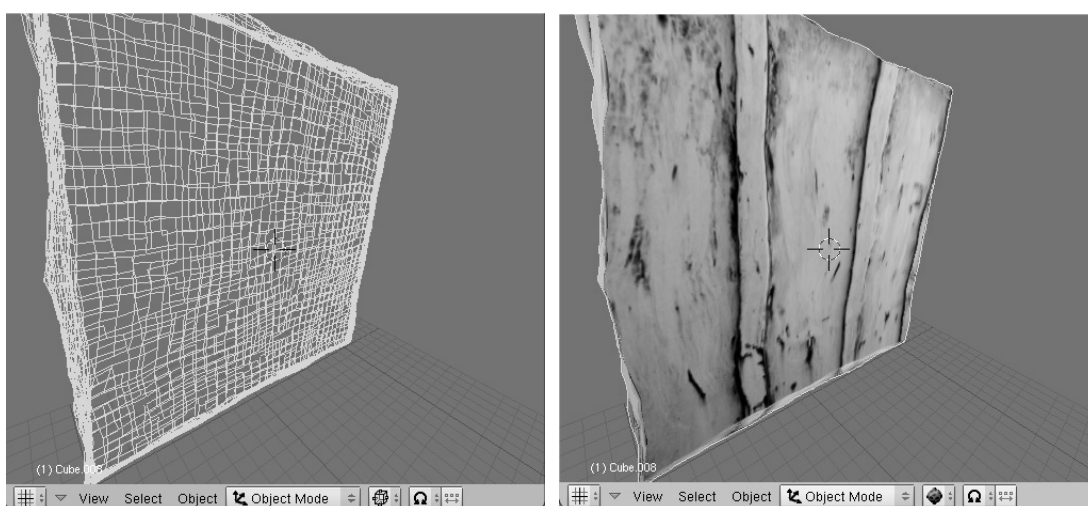
Blender

Det visuelle og det interaktionsmæssige domæne repræsenteres i vores prototype vha. programmet *Blender* (www.blender3d.org), der indeholder en 3D *game engine*, hvori 3D-objekter kan opbygges (modelleres) og gives egenskaber som vægt, hastighed og rotation. Endvidere kan disse objekter underlægges styring fra brugerens side via mus og tastatur. Blender tjener som miljø for udformningen af de konkrete visuelle elementer, såvel som de interaktionslogikker, der kendetegner vores design. Blender er et miljø, hvor alle inputs (keyboard/mus) resulterer i forandringer i det grafiske layout, og er det eneste element i prototypen, der 'forstår en bruger'. En tidligere prototype¹⁰⁴ var opbygget i python, hvor det grafiske element blev konstrueret i en udvidelse til Python (VPython, <http://vpython.org/>), som producerer 3D-vektorgrafik. Interaktionen var også baseret på Python-scripts, og var forholdsvist begrænset. Det var eksempelvis ikke muligt at nå musens scrollhjul med Python-kode, hvilket er muligt i Blender. Endvidere er Blender et integreret miljø, hvor 3D-modellering og programmering kan foregå i samme brugerflade. Det er dog primært på grund af den mere ekspressive omgang med visuelle elementer, at Blender er valgt ud. Det er muligt at 'fastklæbe' teksturer i VPython, men vanvittigt besværligt. Blender er derfor det modelleringsværktøj, der gør det muligt at udvide den visuelle æstetik, fra vektorgrafik til teksturerede elementer. Begreber som *tiling* (tile=flise) og *bump mapping* beskriver i spiludvikling, hvordan billeder 'klistres' på et skelet af x, y og z koordinater. Videre behandler *tiling* hvordan et billede, der er firkantet, brydes op i mindre polygoner for at kunne tilpasses en sfære, hvilket plejer at være en kompliceret procedure, men forholdsvis nemt i Blender. Her et screenshot fra *Phrasetrækker*, hvor vi ser hvordan et firkantet billede til venstre passer til sfæren til højre:

¹⁰⁴ MUSK controllerbox. Kan ses/hentes på http://www.bitreactor.dk/_private/science.htm



Et eksempel på kompleks tiling med et firkantet billede, der er brudt i mindre firkanter, hvilket påklister et komplekst skelet af x, y, z koordinater. Her en bulet væg fra den stenramte container:



Python

Der findes som sagt interaktionsmæssige overlap til Blenders måde at behandle input, i forhold til den tidligere prototype (MUSK). I denne prototype varetages formgivningen af interaktion gennem Blender. Python's primære formål er at binde data fra Blender-interfaceset til MIDIYoke. Python varetager oversættelse af al scripting af den grafik, vi har bygget i Blender, og udsender MIDI-signaler. Dette gøres gennem scripts som *PositionToMidi* og scriptet *Initialize*, der åbner MIDI-porte:

```

import GameLogic
import Util
from Singleton import state

# finder XYZ-position og konverterer
# til midi cc paa kanal 1
# ID = 0 -> 20, 21, 22
# ID = 1 -> 23, 24, 25
# osv.
co = GameLogic.getCurrentController()
cc = (co.getOwner().ID * 3) + 20

vx = co.getOwner().getPosition()[0]
vx = int(Util.clip(vx * 8.0, 0, 127))
vy = co.getOwner().getPosition()[1]
vy = int(Util.clip(vy * 8.0, 0, 127))
vz = co.getOwner().getPosition()[2]
vz = int(Util.clip(vz * 8.0, 0, 127))

if state.initialised:
    state.output.sendMsg(state.messages.controller(0, cc, vx))
    state.output.sendMsg(state.messages.controller(0, cc + 1, vy))
    state.output.sendMsg(state.messages.controller(0, cc + 2, vz))

```

```

import Rasterizer

from Singleton import state
from Prefs import prefs

if state.initialised == False:
    # vis musepil afhængigt af scenetypen
    state.scene_name = GameLogic.getCurrentScene().getName()
    print 'Starting', state.scene_name
    if state.scene_name == 'Pong':
        Rasterizer.showMouse(False)
    else:
        Rasterizer.showMouse(True)

# juster framerate
GameLogic.setLogicTicRate(60)
GameLogic.setPhysicsTicRate(60)

# find midi devices
import MIDI
devices = MIDI.DeviceList()
state.outputDevices = devices.outputDevices()
state.inputDevices = devices.inputDevices()

# aabn midi output device valgt i Prefs
if prefs.outputDevice in state.outputDevices:
    state.output = MIDI.OutputDevice(prefs.outputDevice)

# lav en MIDI message factory
state.messages = MIDI.MessageFactory()

# initialiser activepoint (for Tracker)
state.active_point = 1

# initialiser timer-variabler (for Tracker)

```

Disse scripts sørger for at MIDI-porte er aktive, og at positionen og anden aktivitet i de konstruerede scener omsættes til MIDI-data, ud fra hvordan brugeren har konfigureret interfacet. Blenders *Game Logic*-udvidelse omdanner brugerinput til konsekvenser i scripting-miljøet. Bindningen sker konkret mellem denne Game Logic og Pythons mulighed for at transformere enhver datatype til MIDI-data. Game Logic varetager f.eks. navigation i miljøet via scripts som f.eks *CameraNavigation* eller *TrackerCamera*:

```

# Kamera zoom og rotation

import GameLogic
import Rasterizer
from math import sin, cos, pi

# Hent input fra mus og tastatur
co = GameLogic.getCurrentController()
cam_rightclicked = co.getSensor('cam_rightclicked').isPositive()
cam_mousedrag = co.getSensor('cam_mousedrag').isPositive()

# Træk med højre musetast:
# op/ned = zoom ind/ud
# højre/venstre = roter vandret
if cam_rightclicked and cam_mousedrag:
    # Find det aktive kamera
    cam = GameLogic.getCurrentScene().active_camera

    # Find musepilens flytning i x- og y-retningerne
    rmb = co.getSensor('cam_rightclicked')
    dx = ((Rasterizer.getWindowWidth() / 2) - rmb.getXPosition()) /
    dy = ((Rasterizer.getWindowHeight() / 2) - rmb.getYPosition()) /

    # Zoom ind/ud, indenfor lovlige værdier
    pos = cam.getPosition()
    if (-50 < (pos[1] + dy) < 8.5):
        cam.setPosition([pos[0], pos[1] + dy, pos[2]])

```

```

import GameLogic
import Rasterizer
import Util
from Singleton import state

co = GameLogic.getCurrentController()
cam = co.getOwner()

# Get mouse inputs from sensors attached to camera
cam_mouseover = co.getSensor('cam_mouseover')
cam_leftclicked = co.getSensor('cam_leftclicked').isPositive()
cam_wheelup = co.getSensor('cam_wheelup').isPositive()
cam_wheeldown = co.getSensor('cam_wheeldown').isPositive()
cam_mousedrag = co.getSensor('cam_mousedrag').isPositive()

# Get keyboard inputs from sensors attached to camera
cam_arrowup = co.getSensor('cam_arrowup').isPositive()
cam_arrowdown = co.getSensor('cam_arrowdown').isPositive()

# Get timer attached as property sensor
cam_timer = co.getSensor('cam_timer')

# Get objects in scene
objs = GameLogic.getCurrentScene().getObjsList()
if cam_leftclicked:
    # Mål tiden fra sidste klik (i sekunder)
    state.then = state.now
    state.now = co.getOwner().cam_time_mouse
    state.last_interval = state.now - state.then

# Hvis der er gået længere tid end et 'træk-interval'
# saa skift til næste punkt
if state.last_interval > 0.1:
    if state.active_point < 5:
        state.active_point += 1

```

MIDI

Repræsentationen af det auditive domæne er variabel i forhold til de specifikke arbejdsredskaber og miljøer, den enkelte musiker foretrækker. Med andre ord baserer vi os her ikke på én udvalgt komponent, men derimod på brugerens eget valg af komponent(er). Vi orienterer os mod fleksibilitet. Dog må dette valg indskrænkes af en nødvendig 'mindste fællesnævner' – en fælles digital repræsentation – som såvel vores prototype som brugerens egne arbejdsredskaber kan håndtere. Når det drejer sig om digital repræsentation af musikalsk information er standardløsningen MIDI-protokollen, der metaforiserer 'musik' som et forløb af begivenheder

(toner slås an, et tempo defineres, klangparametre ændres, instrumenter omkonfigureres, osv.). Vi anvender derfor MIDI-standarden mhp. modifikation af hhv. musikalsk tekstur og musikalsk struktur, hvor begge elementer kun kan kontrolleres i *realtime* via MIDI. Et andet relevant valg, som understøtter mulighed for modifikation i *realtime*, er OSC (Open Sound Control), der er en netværksbaseret protokol. OSC understøtter ændring af parametre og kan sågar sende decideret lyd gennem protokollen, og ikke kun information om lyd. Men da OSC kræver et *IP-netværk* og dertilhørende netværksport på hardwaren, udelukker vi med dette valg at kunne kontrollere hardware som synthesizere, samplere, mv., der ikke har netværk- og kun MIDI-porte. Videre er OSC protokollen indtil videre sjælden i software, da implementeringen er fravalgt på grund af manglende udbredelse og accept blandt hovedudviklerne af musiksoftware¹⁰⁵. En protokol indebærer at forskellige producenter af musikhardware og -software er blevet enige om en teknologisk standard for hvordan musiske begivenheder repræsenteres digitalt. Det er dermed op til os at overholde denne protokol i vores eget design, således at vores prototype rent faktisk kan kommunikere med det eller de arbejdsredskaber, den enkelte musiker ønsker at anvende.

MIDI er oprindeligt defineret i sammenhæng med musikhardware som synthesizere, samplere, keyboards, osv. Når kommunikationen flyttes til et software-domæne må den fysiske basis for kommunikationen (MIDI-kabler og porte) gentænkes i et virtuelt domæne. Dette er gjort gennem anvendelsen af såkaldte virtuelle MIDI-porte, hvoraf vi anvender en implementation ved navn MIDIYoke, der er Windows-specifik software.

Formmæssig fiksering af prototypen

Identifikationen af komponenter fungerer som *trace dependencies* mellem forskellige lag af softwaren, der sikrer at en konceptuel model understøttes. Interaktionsformen og de designmæssige selektioner, der tilvejebringer den, er videre æstetiske og funktionelle konstruktioner, som komponenterne skal understøtte. I det følgende vil vi diskutere nogle af disse konstruktioner, hvilke kilder de har, og hvilke aspekter af prototypen de påvirker.

Empirisk fiksering af interaktionen

Interaktionsdesignet hænger sammen med vores funktionelt og æstetisk funderede systemfilosofi og konceptdefinitioner. Interaktionen er vigtig at fastslå på et tidligt tidspunkt for at kunne justere den gennem et større antal iterationer. Empirien påvirker i denne sammenhæng koncepter i prototypen.

¹⁰⁵ Alle større plugin- og sequencer producenter som Steinberg, Emagic, Digidesign, mv.

Tiny God – “*I hope we get a higher-level control constructs, and control systems that work well at all levels [...]It's hard to pay attention to all those levels at once.*

Tiny God taler om kontrol over modificering af musikalsk tekstur såvel som struktur, hvor denne modificeringsproces har mange lag og mange muligheder. I henhold til *control intimacy* og *player bandwidth* må interaktionen i forhold til brugerens egen konfiguration ikke overbelaste det musikalske overblik. Fikseringen af interaktionen leder mod et højt funktionelt niveau, hvor kontrollen af musik kan foregå ubesværet på alle niveauer. Dette har bl.a. medført kriteriet ’forenkling af kompleksitet’, der resulterede i ideen om en *Controllahtwistah* som et overblik over komplekse data fra 12 kilder. Designet startede med 128 kilder, hvilket ville omfatte et komplet overblik over alle MIDI-controllere på en given kanal. Tekniske begrænsninger i Blender reducerede dette til 12 pga. for lav ydelse. Interaktionen i *Controllahtwistah* er designet mod at skabe en funktionel kobling til to faktorer: Hvilket controller-nummer der arbejdes med (1-12) og hvilken værdi, controlleren har (hvor hurtigt grafikken roterer).

Oli Larkin - “*I'd like to make an on-screen controller that used a real physics based environment to control music. So you could make an imaginary room and roll a ball around it and the sound would change*”

Dette udsagn må ses besvaret i *Containerturm*: Et miljø, hvor fysiske love som tyngdekraft og andre naturlige egenskaber påvirker objekter, der sender MIDI-data. Et miljø med bolde i en scene, hvor boldenes bevægelse og retning er bestemt af ujævne overflader såvel som brugerens input. Interaktionen skal ifølge udsagnet betjene sig af kendte objekter som bolde, der i sin natur kalder på berøring, hvilket vi har indført. Videre har vi indarbejdet ’naturlige’ constraints i så vid udstrækning muligt, f.eks som væggen i containeren, hvor kun batten og ikke boldene kan gennemtrænge overfladen. Den funktionelle kobling består af boldenes x, y, z koordinater, der for hver bold omdannes til output på tre MIDI controller-numre (seks controllere i alt).

Mate Galic – “*I want a nice interface that doesn't confuse too much and have sufficient possibilities.*”

Vi forenkler interfacet, og i samme bevægelse også interaktionen. Dette gør vi for at formindske forvirring i forhold til funktionalitet. Derimod kan forvirring opstå, når interfacets begrænsede funktionalitet skal forbindes til en musikers eksisterende arsenal af instrumenter. Interaktionen har derfor også brugerens *mental bandwidth* og *control intimacy* som udgangspunkter – i hvor stor en grad, musikeren kan relatere egen musik til et givent interface, dvs. overskue dets muligheder for musikalsk modificering i forhold til grundliggende kompositoriske ideer. *Phrasetrækker* er designet mod ikke at overskride en musikers overblik gennem rumlig hukommelse, hvor rumlig placering

bliver ensbetydende med en grad af modificering. I *Phrasetrækker*-scenen bliver objektet, der bevæger sig, ensbetydende med hvor 'lyden' befinder sig i forhold til det ikke-modificerede udgangspunkt. Med andre ord viser objektet hvor mappede parametre befinder sig i rummet. *Sufficient possibilities* er i udgangspunktet op til brugeren, da brugeren bestemmer hvilke musikalske aspekter, der skal modificeres af interfacet. Dette er muligt da *shells*, *sequencere*, *plugins* og andre afviklingsmiljøer er højt modulare og lader brugeren konfigurere hvad der modificeres af hvad. I denne tænkning bliver interfacet et samlingspunkt for konfiguration og interaktion, hvor brugerens individuelle opsætning og binding til interfacet udforskes gennem brug.

Egbert Jürgens– ” *you exercise everything there is to making and understanding music – like timing, understood as space between notes or periods*”

Exercise oversætter vi som udøvelse, dvs. en kropslighed i musikken, hvor udfoldelse og eksperimenter sker gennem leg. Musikken gøres relevant gennem en engageret form for indføling, for at kunne følge et bestemt *gameplay* eller plot. Vores plot er leg med lyd gennem manipulation af objekter, og interaktionsmomentet vil henføre til dette plot. F.eks. gennem simulering af tidligere oplevelser med spil og navigation i spil, hvilket metaforisk anvendes i vores musik-interface til leg og navigation i lyd. Interaktionen er herigennem bundet til metaforen 'spil', hvilket rent funktionelt er understøttet gennem en *gameplay*-agtig anvendelse af grafik. Samtidigt er interaktionen et produkt af et komplekst netværk af konfigurationsmuligheder, hvor 'at lave musik' er den primære aktivitet med prototypen, og en forudsætning for et udbytte af prototypen. 'At *game*' er en sekundær aktivitet, da prototypens funktionelle værdi prioriteres i vores design højere end oplevelsesværdien i interfacet. Med andre ord er grafikken understøttende for en oplevelse for en musiker, men musikerens udbytte af kontrolløren som et MIDI-værktøj vejer tungere i vurderingen end pæn grafik. En (funktionelt baseret) leg med musikalske grundsten som *timing*, og som anslået en metaforisk forståelse af lyd.

Sammenfattende har empirien været designmæssige sunde forstyrrelser, der får os til at overveje temaer som:

- Hvor og hvordan vi skal implementere muligheden for interaktion med informationen mellem de mange lag af musikalsk software, hvor kontrollen af materialet skal være intim og relevant.

- Hvilke constraints og potentialer, der skal implementeres for at understøtte interaktionsdesignet. Specielt sondringen af prototypernes relation til designkriterier som *naturlige egenskaber* og *oplevelsesmæssig adgang til musikalske strukturer/teksturer*.
- Fleksibel konfiguration af interfacet i forhold til brugerens allerede installerede musiksoftware. Kompatibilitet er kriteriet for hvordan interfaces mellem disse applikationer skal designes.

Teoretisk fiksering af interaktionen

Innovation - Adaption

Vi har opstillet en antagelse om at innovation kan ske gennem adaption af eksisterende teknologier. Antagelsen forfølger vi i designet af interaktionen som en adaption af *game* metaforer, og gennem brug af naturlige egenskaber som forhold mellem handling og auditive/visuelle resultater. Indoptagelsen af spil-lignende interaktionsmønstre er videre en adaption i forhold til lydmodificering, hvor omverdenskendte fænomener som tyngdekraft (overraskende) medfører innovative soniske muligheder. Adaptionen af spilteknologi har medført, at vi på en musikalsk akse har konstrueret nye modificeringsmuligheder, heriblandt automatisering. Det er med alle tre interfaces muligt at modificere musikalsk tekstur eller struktur uden at røre enten mus eller tastatur, efter at have sat et forløb i gang. Disse forløb over tid, der modificerer lyd, er sædvanligvis enheder som LFO-moduler. En LFO er en lavfrekvent oscillator, der med en stabil bølgesvingning modificerer lyd – et eksempel er en *tremolo* effekt, hvor volumen i henhold til regelmæssige bølgesvingninger skrues op og ned af LFO'en. Denne kontrol er med en LFO en lineær og forventelig svingning. En lignende, lineær kontrol har vi indbygget i *Phrasetrækker* og *Controllahtwistah*. Kontrollementerne i *Containerturm* er ikke-lineære – de udsender ikke-lineære MIDI-data, da de for det første er styret af en ikke-lineær tyngdekraft-algoritme. For det andet fordi ujævne overflader i miljøet i nogen grad nedbryder forventninger til boldenes bane. For ikke at foregribe afsnittet om den teknologiske fiksering af interaktionen, må det her siges at antagelsen om adaption og innovation kan observeres i vores design af de tre interfaces. Spilteknologier som *tiling*, *Game Logic* og den interaktion, vi herudfra designer, er gennem adaptive processer blevet til instrumentmæssig innovation, udelukkende fordi de kendte teknologier er sat ind i en anden kontekst/diskurs. En krydsning af områderne spil og musik giver overordnet overraskende udfald og interaktionsmuligheder i arbejdet med musik.

Fuzzy Logic

Fuzzy logic formidler teoretiske pointer i forhold til ikke-numeriske interfaces med glidende overgange mellem forskellige medlemskaber, hvilket blev udlagt som *intersections* (overlap)

mellem mængder og delmængder af egenskaber. I de tre interfaces konstruerer vi interaktionen ud fra flydende medlemskaber af forskellige egenskaber som f.eks. 'meget modificeret' og 'lidt modificeret'. Vi arbejder henimod en grafisk repræsentation af glidende overgange, eller mere præcist mod at etablere visuelle gradueringer af et sonisk kontinuum. Et transit mellem yderpunkterne sker gennem en rumlig erfaring, og ikke som en erfaring, der kræver læsning af numeriske værdier i en brugerflade. En given MIDI-controller kan have værdi 0-128, hvilket er et forhold, vi har omsat til afstanden mellem væggene i *Containerturm* og *Phrasetrækker* (Fig. 1). Med andre ord, en mængde talværdier, der omskrives til 'vejen fra punkt a til b'. *Controllahtwistah* (Fig. 2) indeholder en lignende omskrivning af et 'MIDI-kontinuum', hvor værdi 0-128 afhænger, af hvor hurtigt grafikken roterer.

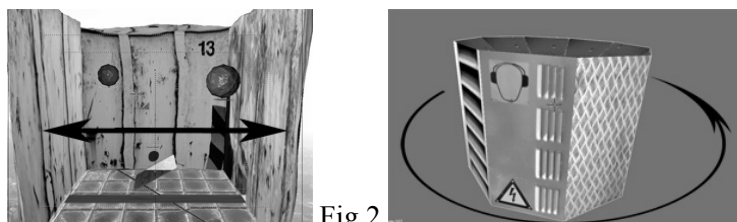


Fig.1

Fig.2

Interaktionen er på grund af det visuelt repræsenterede kontinuum en afgrænsning af bevægelsesmuligheder mellem yderpunkter i kontinuaene. Rummelige fysiske grænser bliver også grænser for lydlig modificering. Interaktionen med musikalske elementer som cent og semitoner (tonehøjde) kan også relateres til det rumlige, ikke-numeriske forhold til kontrolelementer i de tre interfaces. F.eks. ved at binde tonehøjde til et af interfacets parametre, og indarbejde trinløse og glidende overgange mellem toner ved at koble denne instans til interfacet.

Intersections er nedslagspunkter for interaktion i interfacet, hvor der for hvert kontinuum er skabt en mulighed for at interagere med dette kontinuum, f.eks. den lydige placering i forhold til 'en høj tone' eller 'en lav tone'. Med andre ord har vi som designere analyseret et modsætningsforhold, som vi vil give brugeren muligheden for at navigere i. Valget har vi truffet fordi tal på et feedback-interaktionsmæssigt plan giver mening som rumlig placering i forhold til det auditive resultat. 'Til venstre i miljøet' kunne lige såvel være tre forskellige parametre, der står på hhv. 20, 45, 12. I interfacet har brugeren muligheden for at konstruere en *mapping*, hvor f.eks. justering af tre parametre giver det samme auditive resultat som 'til venstre i miljøet'. Her er en interaktionsmæssig vigtig pointe i, at der ikke er mulighed for at betjene tre mus på samme tid, eller tre sæt piletaster. Men der findes derimod en mulighed for at konfigurere interfacet så tre parametre automatisk justeres i forhold til hinanden, hvilket kan betjenes med én mus eller ét sæt piletaster. På interaktionsniveauet har vi her konstrueret interfacet med reference til *player bandwidth* og *control intimacy*, hvor overblikket og overskuddet i arbejdet med musikalsk

materiale understøttes via forenkling af controller-mapping. Vi har videre fjernet numeriske værdier fra interfacet og har designet interaktion ud fra sansemæssig redundans – som auditive og visuelle overlap i processen med at skabe musik. Sansemæssig redundans leder mod en form for transparens i forbindelse med interaktionen. Funktionaliteten og dennes iboende kompleksitet harmonerer igennem redundansen med funktionalitetens repræsentation. Med andre ord at instrumentet bliver gennemsigtigt i vejen fra ide til lyd, hvor hindringer som (for høj) kompleksitet ellers kan stoppe processen. For høj kompleksitet er motivationen for at omskrive komplekse interface-data til figurer og andre repræsentative former. Konsistens mellem interface og funktionalitet er i denne bevægelse vigtig, da unødvendig kompleksitet opstår ud fra misforhold mellem hvordan funktioner er mappet, og hvordan de er gjort tilgængelige. Transparens og konsistens er forudsætninger for interaktionen, hvor funktioner må være tilgængelige og forståelige, før interaktion i det hele taget er en mulighed.

Opsamling

Emergens er et begreb, vi anvender for at afrunde diskussionen af teoretisk og empirisk fiksering af interaktionen. Emergensen i designprocessen har mange kilder og beskriver en samlet udvikling for flere elementer på samme tid, hvilket vi forstår som mange påvirkningsfaktorer af et design. Når emergens relateres til interaktive konstruktioner er det muligt at anskue prototypens finite interaktionsdesign ud fra hvilke muligheder, vi har haft, og hvilke, der ud fra kriterier er aktualiseret. Emergens er en gradvis fremtoning af aspekter af prototypen, der iterativt er udviklet for at modsvare interaktionsmønstre, der eksisterer i musikergruppen. Disse kriterier er udviklet teoretisk såvel som empirisk udvundet. F.eks. findes der et empirisk behov for at kunne give slip på interfacet, som derefter arbejder videre ud fra mønstre, brugeren har sat i gang (*Phrasetrækker* og *Containerturm*). Empiriske udsagn relateres i denne proces til teoretiske antagelser om, hvordan funktioner skal repræsenteres, og under hvilke kriterier, interaktionen med interfacet foregår. Dette er grobunden for gensidige forstyrrelser (trace dependencies) mellem modeller for teori og empiri. Dependencies udviklet undervejs afspejler specialets iterative og emergente konstruktion af interaktionen. Empiriske udsagn forstyrrer teoretiske antagelser og vice versa, og jo mere fokuserede, vi er på dette forhold i udviklingen, des flere elementer og variabler vil der indgå i designet som muligheder for brugeren. Denne proces anser vi som en emergent fremtoning af prototypens form og herunder iboende interaktionsmuligheder.

Teknologisk fiksering af interaktionen

Vi anser det som sidespild at diskutere mus og tastaturs begrænsninger i forbindelse med interaktion. F.eks. at musen skulle have flere knapper og måske svæve frit i luften, hvilket er

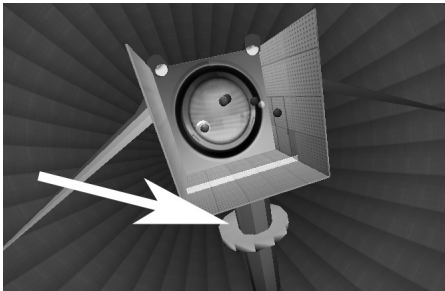
teknologiske perspektiveringer, der ikke er givende at diskutere i forbindelse med de begrænsninger, vi har indordnet os under. Vi vil hellere opridsede de kodemæssige problemer, vi har haft med at indkredse en interaktionsform, som hardwaren selvfølgelig er en del af, men ikke som 'roden til problemet'. På denne måde beholder vi fokus på eksisterende teknologi, fremfor fremtidige interfaces, der lige skal opfindes først. Den største udfordring i fikseringen har været at få softwaren til at understøtte input i henhold til f.eks. teoretiske antagelser om repræsentation af funktioner eller empirisk data om ergonomi. Som nævnt er der trace dependencies mellem alle modeller i specialet, hvor teknologien ofte er en stopper for hvilke koncepter eller empiriske udsagn, der er mulige at konstruere. Dette er et softwareproblem, der må løses gennem konkrete overvejelser på baggrund af empirisk og teoretisk fiksering af interaktionen. I det følgende beskrives hvordan teknologien fikserer interaktionen med udgangspunkt i den hardware og software, vi har anvendt.

Venstreklik

Venstre museknap er en aktualiseringsknap; taktile overflader vælges og manipuleres med denne knap, hvilket er konsistent med knappens funktion i alt andet software. Vi har i *Containerturm* og *Controllahtwistah* fravalgt at implementere en mere avanceret timing, hvor knappen har én funktion, når der klikkes, og en anden funktion når knappen holdes inde – en mere kompleks modalisering. I *Phrasetrækker* har vi derimod eksperimenteret med timing for at afprøve Blender-teknologien i forhold til interaktive funktioner og gøre musen mere spændende. Dette finder vi dog ikke nødvendigt i *Containerturm* og *Controllahtwistah*, da vi har været omhyggelige med at *mappe* få, men essentielle funktioner til venstreknappen. Med andre ord har vi overvejende haft som målsætning at undgå en forvandling af elementære funktioner som venstreklik til mere avancerede features i interaktionen. Dette har vi gjort ud fra et designkriterium, der behandler eksisterende brugsmønstre i forbindelse med spil og anden musiksoftware, for at reducere de forhindringer for interaktionen som mus, tastatur og software kan producere.

- *Containerturm*: Knappen bruges til at holde fast i battet, og flytte det.
- *Controllahtwistah*: Knappen bruges til at 'spinne' objekterne. Når knappen er trykket ned, måler prototypen, hvor meget musen bevæges. En stor afstand fortolkes i koden som 'hurtigt' og grafikken roterer hurtigt. En lille bevægelse er modsat fortolket langsomt, hvilket får grafikken til at rotere langsomt.
- *Phrasetrækker*: Knappen bruges til at placere objekter i miljøet; knappen kan holdes inde for at placere et objekt. Jo hurtigere der klikkes, des hurtigere vil den røde bold bevæge sig imellem objekterne. Interfacet læser derfor i nogen grad 'rytme' og indpasser timingen i koden efter

brugerens interaktion. Rytmen kan videre ændres ved at spinne hjulet lige under scenen, hvilket også gøres med venstre museknap.

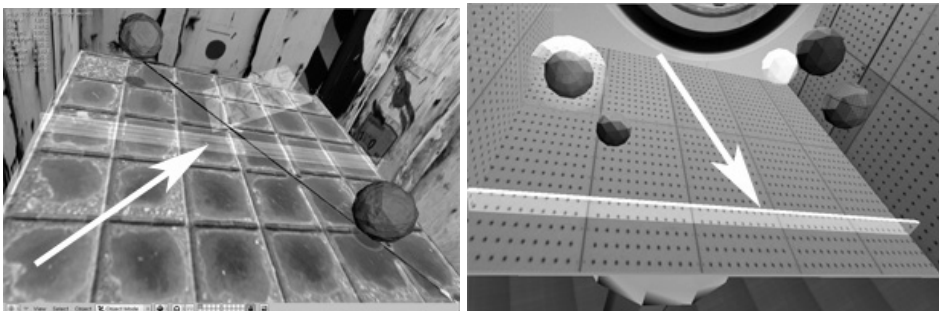


Højreklik

Højre museknap er valgt til at zoome ind og ud af miljøet. I *Containerturm*-interfacet er mappingen kun foretaget for at kunne opleve grafikken anderledes, altså en kosmetisk implementering. I *Phrasetrækker* er zoom funktionen derimod nødvendig for at kunne nå hjulet, der justerer tempo (billedet ovenfor). I *Phrasetrækker* er højreklik derfor nødvendig for at kunne 'nå' en funktion i interfacet. Dog er funktionen duplikeret, da tidsmæssig afstand mellem venstreklik ligeledes justerer tempo. I *Controllahtwistah* er højreklik ikke mappet, og er derfor en inaktiv funktion.

Scrollhjul

Hjulet justerer dybde cues i *Containerturm* og *Phrasetrækker*, hvilket er repræsenteret med bjælken nederst i scenen. Skygger fra stenene i containerturm er også i nogen grad med til at angive, hvor dybt i scenen de befinder sig. Dette viste sig at være kompliceret at lave, hvilket også kan ses, da skyggerne ikke altid opfører sig som de skal. Derfor valgte vi ikke at lave skygger i *Phrasetrækker*.



Scrollhjulet anser vi stadig som en luksus, ikke alle har – eksempelvis Mac-mus, der er kendt for én knap. Af denne grund kan man også justere dybde cues med piletast op og ned, for at give brugere uden scrollhjul muligheden for at få adgang til vitale funktioner i prototypen.

Tastatur

Piletast op og ned justerer som sagt dybdecues i to scener, og Enter-tasten genstarter scenen. Genstartsfunktionen er nødvendig, da stenene nogle gange ryger ud af *Containerturm*-scenen. Vi

har valgt ikke at gøre tastaturet essentielt i forhold til prototyperne, da vi gerne vil konstruere interfaces, der kan betjenes udelukkende med musen. Igen i henhold til simplificering eller kompleksitetsreducing af interaktionen, hvor det, der kan interageres med, er essentielt i forhold til funktioner, der binder sig til modifikation af lydligt materiale. Vi kunne have kontrolleret hele prototypen med tastaturet, hvilket kunne resultere i flere funktioner, der kunne mappes til flere inputs. Men dette er ikke det primære formål med interfacet, da vi sigter mod en større mængde kontrol med mindre mængde hardware; musen skal isoleret kunne styre alt i interfacet. Videre ville en mapping til tastaturet bevirke, at bevægelsen i rummene ville hakke. Et tastatur er hardwarekodet til at lave gentagelser af f.eks. en piletast med ret store mellemrum. Raten hvormed disse gentagelser sker er i forhold til en mus' baudrate betydeligt lavere. En mus sender med andre ord en mere fintmasket informationsstrøm til interfacet, hvormed interfacet kan sondres med flydende bevægelser, og ikke med hakkende bevægelser som med et keyboard. Prøv evt. forskellen på at justere dybde-cues med hhv. scrollhjul og piletaster. Det hardwaremæssige designvalg er begrundet i fuzzy logics tanke om intersections gennem flydende overgange. Overgangen mellem medlemskaber foregår ikke i abrupte trin, men via stigende og faldende gradueringer.

Software

Blender er knudepunktet, der forbinder brugerflade med underliggende kode – det er dermed den eneste variabel, der kan påvirke interaktioner mellem bruger og kode, da Blender er den grafiske tilgang til at forstå hvad koden kan. Forståelsen er et forhold, der opstår når en brugers forventninger møder interfacets muligheder. Herunder konkret feedback fra interfacet, der er etableret som et svar på brugerforventninger – forventninger om hvad interfacets funktioner er og hvordan disse funktioner opdages. Såfremt forventninger ikke modsvares i interfacet, standser interaktionen, hvilket vil sige at interaktionen stopper på grund af teknologien i interfacet. Mappingen, der ligger i interfacet, bliver derfor et knudepunkt, hvor mange faktorer som f.eks. funktionalitet, æstetik, og inputs skal forenes. Faktorerne må overvejes i forbindelse med et interface, der giver feedback om hvad der er muligt, og hvordan brugeren interagerer med disse muligheder. Ved at simplificere en MIDI-kontrolflade har vi fjernet en del potentielle blokader for interaktionen, da mængden af inputs, der kan forstyrre et interaktivt forløb er reduceret. Der er dog en reel mulighed for at nye blokader opstår, når vi implementerer avancerede algoritmer som tyngdekraft. Specielt når algoritmerne 'arbejder i kulissen' og ikke kan nås med en brugers interaktion. Mange algoritmer i Blender er designet til at etablere simuleringer af kendte fænomener som f.eks. tyngdekraft. Simuleringer anser vi som referencer til fundamentale forståelser, hvilket bruges som en guide i interaktionsdesignet. Gennem vores designvalg af simuleringer (algoritmer) ekspliciterer vi i interfacet en forståelse af interaktionsformen. En brugers forståelse af interaktionsdesignet opstår sideløbende med en forståelse af hvad interfacet refererer

til. Med andre ord opdager en bruger, hvordan Blender udtrykker denne kobling af metaforik med interaktionen med musikalsk materiale.

Interfacet er i denne sammenhæng som nævnt et knudepunkt for interaktionsdesignet, hvor blokerende faktorer for musikalsk virke imødegås gennem empiriske sondringer og teoretiske perspektiveringer. Vi arbejder imod at teknologien bliver gennemsyret af empirisk/teoretiske overvejelser, og i forhold til vores brugergruppe ikke kan spores tilbage som blokaden af interaktionen. Tværtimod understøtter Blender som teknologi en fiksering af interaktionen som et avanceret instrument forklædt som et spil.

Empirisk fiksering af æstetikken

Æstetikken og æstetiske kriterier er fundamentale dele af specialet, hvor æstetikken er anskuet som en påvirkning af designets finitte form. Æstetik er et diffust fænomen, vi inkorporerer i designet gennem analyse. I denne proces gendriver vi så meget designrelevant information som muligt for at imødegå æstetikkenes diffuse natur, og placere æstetiske kriterier som enten empiriske, teoretiske eller teknologiske funderede. Påvirkningen af designet har med andre ord hhv. empiriske, teoretiske og teknologiske kilder, hvor det er metodisk vigtigt at placere kildernes ophav, hvilket sker i de følgende afsnit. Æstetikken er i specialet udlagt som både et systeminternt fænomen og som et systemeksternt fænomen. I dette afsnit vil være fokus på musikere som en ekstern æstetik for designet, der hver især har systeminterne æstetiske kriterier. Disse kriterier vil her diskuteres som en empirisk indkredsning af æstetikken som horisonter for prototypen, og hvordan kriterierne påvirker designets form.

Klaus Voltmer: *“It is all about enhancing new features that are easy to spot, but hard to figure out in a given interface”*. Æstetikken bliver i Klaus’ udlægning et bindeled mellem funktionalitet og dennes repræsentation i et interface. *Easy to spot* handler om funktioners synlighed i interfacet, hvor interaktive elementer som f.eks. bolde og cylindere er mere fremtrædende end passive, ikke-interaktive elementer som f.eks. en baggrund. Dette er såvel æstetiske som interaktive designkriterier, hvilket er svært at adskille, da æstetikken i interfacet (f.eks. metaforik) er *cues* til hvad der kan interageres med, og interaktionen på den anden side forløber i henhold til et æstetisk formsprog. *Hard to figure out* handler om det auditive output, hvor interfacets feedback giver information til brugeren om hvad koden gør ved lyden. Det kan være svært at gennemskue, hvad der rent faktisk sker, hvilket ifølge Klaus er en givende mystifikation af et program. Æstetikken indrammer med andre ord prototypens mysterie, da visuel og auditiv æstetik bliver et overlap, brugeren kan interagere med. *“It is good fun working in 3D, but it is rare that this type of interface brings out new forms of modulation, composition”*. Æstetikken i interfacets metaforer om

rumlighed, tyngdekraft, mv. har været vigtige designkriterier, som der er lagt en del arbejde i at repræsentere. I udviklingen af disse kriterier er der sideløbende opstået nye funktionelle aspekter af prototypen, hvor æstetisk funderede elementer som hoppende bolde medfører en ny måde at modulere sin lyd på. Dette har været en emergent formgivning af prototypen, hvor æstetikken i interfacet også er empirisk funderet, da udvikleres udsagn om f.eks. 'interfacemysterier' og visuel modifikation af lyd er blevet en del af prototypen. Vi har fikseret prototypens æstetik som et mystisk samspil mellem billede og lyd, der netop giver nye muligheder for modifikation. Mulighederne må i henhold til æstetik indpasses efter hvordan disse fremtræder for brugeren, da der er et reelt faremoment i at anvende ny teknologi for at skabe nye interaktionsmønstre (prototypen kan blive *for* mystisk). Vi sonderer derfor disse muligheder gennem analyse og med forbehold, da der findes et uudnyttet potentiale i hvordan muligheder og interface kobles, hvilket Klaus som en ekstern kilde på designet også er enig i: "*I definitely see the potential of 3D and work around spatial representations*".

Anders Holst Jensen: "*Hvis jeg åbner min improvisation eller et nyt nummer mod en teoriløs tilstand, hvor tidligere øvede mønstre mere ligger som et latent sprog, finder jeg tit et følelsesmættet udtryk der føles godt. Musikken bliver i den forstand tilfældig, da jeg arbejder enormt struktureret mod at blive strukturløs*". Vi har konstrueret interfacet som en måde at undgå at intellektualisere processen i at lave musik. Æstetikken bliver et middel i dette henseende, hvor vi forsøger at understøtte en mulighed for at abstrahere fra strukturer, skalaer, mv. Vi har med andre ord omsat udsagnet *arbejde strukturelt mod at blive strukturløs* til et interface, der i sin opsætning (konfiguration og *mapping*) kræver struktureret arbejde. På den anden side af dette arbejde findes muligheden for at nyde anstrengelserne, i den forstand at instrumentet bliver *latent*, og en form for strukturløs leg med lyd kan begynde. Laptopmusikere arbejder på to lignende akser – *prækomposition* og *performance/komposition* – hvor der i prækomposition gennem struktureret arbejde etableres et grundlag for at lave lyd i performance og komposition. I strukturel prækomposition opbygges et miljø¹⁰⁶, hvor lydmateriale og lydkilder ordnes og sættes indbyrdes i forhold. F.eks. justeres volumenniveauer i forhold til hinanden, eller effekter kobles på flere forskellige instrumenter. Dette er dybest set en form for 'design', da det springer ud af en brugers æstetiske kriterier, og sættes i gang, hvorved kompositions- eller performance-processen er i gang. Brugeren kan vælge at konfigurere sit miljø mod strukturløshed ved at binde lydligt materiale til tilfældigheds-instanser, f.eks. til *Containerturm*, der udsender kaotiske og ikke-strukturerede MIDI-data. Denne søgen mod spontan og improviseret musik er understøttet i så vid udstrækning muligt ud fra disse tanker om struktur og ikke-struktur. Når en musiker når et stadie, hvor

sondringer af musikalske strukturer foregår med øre for overordnede forløb i et værk, er instrumentet ikke længere et bevidst moment i denne proces. Instrumentets æstetik, klang, muligheder, funktion, mv., er en integreret del af musikerens udtryk og ligger på fingerspidserne. Vi har diskuteret dette som transparens i forbindelse med instrumentet, hvor det 'at øve' skaber en intimitet med instrumentet, hvorigennem instrumentet bliver gennemsigtigt i processen med at omdanne ide til lyd. Operationelle kriterier, hvilket er viden om instrumentet, bliver her forenet med referentielle kriterier, der kan defineres som 'hvad jeg vil med instrumentet'. Når dette overlap fungerer for den enkelte, er der et følelsesmæssigt udbytte at hente – en oplevelse. En oplevelse af at sammenhængen mellem strukturerede og ustrukturerede forløb er kohærent. Med andre ord at forholdet mellem prækomposition og komposition fungerer, og at der er et udtryk at hente for den enkelte musiker. Når flere musikere på dette stadie mødes, skabes der ikke alene for musikeren en intimitet med instrumentet gennem brug, men også en intimitet med den sammenhæng, instrumentet indgår i. I forbindelse med jazz, men lige så vel for laptopmusikere, gælder det om at have motoriske eller intellektuelle færdigheder til at indpasse sit instrument og sit spil i en større sammenhæng, f.eks. blandt andre musikere. *"Hvad der kommer mellem disse strukturer er op til en given besætning at fortolke gennem improvisation. Det er de facto standarden for hvordan jazzmusikere arbejder sammen. Strukturer som en slags enighed og en form for frihed ovenpå disse strukturer"*. Æstetik er her det eneste, der strukturerer den enkelte musikers valg i forhold til de andre musikere, hvor klang, indføling, intonation, mv., bliver underbegreber til æstetik. Møder kunne betegnes som en fællesæstetik, hvor alle involverede opnår en æstetisk enighed om et udtryk. Hvilket også giver et frirum for eksperimenter med denne struktur. I prototyperne er der ligeledes en mulighed for både at arbejde struktureret (*Controllahtwistah, Phrasetrækker*) og ustruktureret (*Containerturm*). I den sammenhæng er det muligt at anvende et interface, der passer bedst ind i det individuelle æstetiske framework, og et interface der tager hensyn til den musikalske sammenhæng det indgår i.

Nestor Pridun: *"Making music should be fluctuant and in flux"*. Processen med at lave musik tilsigtes at blive så transparent som muligt, hvor der ikke reflekteres over værktøjer, metoder og teorier. Dette deler vi også som en form for æstetisk sondring, da æstetikken kan fungere enten som en blokade eller et potentiale for (flydende) musikprocesser. Flydende overgange mellem forskellige stykker af et værk bliver også en gradvis transformation af det lydige materiale, hvor den lydige æstetik ændrer sig. Et flow betyder en konstant forandring, hvor værket konstant forandrer sig og gentagelse nedprioriteres. På den ene side for at skabe et lydligt billede af et musikalsk forløb, der også er hørbart for en lytter på den anden side af værket. Men absolut også et

¹⁰⁶ F.eks. i Buzz, Reaktor, Max/MSP, Pure Data, mv.

forløb, der for den enkelte musiker (Anders og Nestor i hvert fald) har et følelsesmæssigt større udbytte end statiske gentagelser i musikalsk forløb.

Pure: *“Glitch is good for this, because people tend to get abstract when listening to abstract music. In that way random composition is kind of a portal to experience [visual] art as something without a certain meaning but as a means of thinking differently.”* Æstetikken er her nærmest deterministisk udlagt, hvor bestemte teknikker, der frembringer abstrakt musik, afføder bestemte måder at tænke på. Der er uden tvivl en psykisk reaktion på lyd, men reaktionen anser vi som baseret på dybt individuelle konstruktioner, der ikke nødvendigvis forårsager, at en lytter ændrer sin måde at opleve billede og lyd. Men vi tilskriver os tænkningen om at musik kan være med til at modalisere eller fremhæve det visuelle output, og at overlappet skaber en ny medbetydning. Billedet og lyden isoleret har hver især et unikt udtryk, som i kombination ændrer udsigelseskraft, hvilket er en æstetisk grundsten i specialet. Pure taler om at fremprovokere nye måder at tænke på ved at kombinere udtryk, men ikke hvilke som helst udtryk. Der må ifølge ham være en konsistens mellem indholdet og formsproget i hhv. billede og lyd – hvilket lidt langhåret kunne udlægges som et semantisk korrelat. En bruger af prototyperne vil givetvis også finde et auditivt miljø, der æstetisk-semantisk hænger sammen med interfacets fremtoning – som f.eks. et miljø hvor *glitch* er muligt.

Julian Ringel: *“I like to be able to leave my computer; like get a lot of LFO’s to modulate themselves and let the modulation run its own course”.* Vi vil ikke gå så langt som at postulere, at vi har bygget AI ind i interfacet. Mere moderat har vi lavet brugerflæsende kode, der fortolker en brugers input og gentager mønstre, som brugeren har sat i gang. Denne aflæsning antager vi vil skabe et grundlag for *control intimacy*, når input gentages, hvilket også er æstetisk bærende punkt – hvordan intimiteten understøttes ved at give brugeren tilstrækkelig auditiv og visuel feedback om hvilke mønstre, der er ’læst’. Ligeledes virker de førnævnte tyngdekraftsalgoritmer som kontrolementer, der kan styre interfacet uden at brugeren interagerer. *Modulation* er en form for æstetisk design fra musikerens side, hvor udvalgt lydmateriale moduleres i forhold til definerede kriterier – hvor disse kriterier både kan være ’kontrol’ eller som Julians ’anti-kontrol’.

Egbert Jürgens: *“Tactile interfaces is moving in and the functionality is constantly increasing in all aspects. Therefore the touchable is linked to the functionable and interfaces are just better looking now because most people have more than 16MB memory dedicated on their graphics cards – which gives us more headroom to work with as developers”.* Interfacet kan som sagt ’berøres’, hvilket har konsekvenser for det auditive og visuelle indhold. Vi har fastslået denne sondring mellem *touchable* og *functionable* som et medie for vores designvalg. Egbert angiver videre teknologisk fremskridt som en platform for en opjustering af software-kriterier som *touchable* og

functionable. Når man begynder at designe ud fra *touchable*-kriteriet, lægges hovedvægten på en æstetisk radikaliserings af forholdet mellem hvad der visuelt kan repræsenteres, og hvilke funktioner, der ligger bag repræsentationen. Denne akse kan ifølge Egbert indføres som et potentiale i designprocesser, hvor æstetisk fremskridt er et designmål og som et *headroom* man metaforisk burde putte hovedet i.

Teoretisk fiksering af æstetikken

Vi har løbende udbygget teoretiske antagelser, hvilket ses i forrige afsnit om empirisk fiksering af æstetikken. Antagelserne er også basis for næste afsnit om teknologisk fiksering af æstetikken, hvor grundsten fra kapitlet *Design* tages op og diskuteres. Processen med at raffinere det teoretiske apparat vurderer vi tilstrækkeligt beskrevet i *Design*-kapitlet, og vi vil derfor henvise til dette afsnit som en fiksering. Fikseringen har som anslået været en del af et emergent forløb, hvor vi ser de teoretisk-æstetiske afsnit som der hvor 'emergensen starter' i specialet. Æstetikken og de teoretiske bevæggrunde har allerede før konceptudviklingen været et grundlag, som vi både bevidst og ubevidst har arbejdet ud fra, hvilket tidsligt placerer etablering af æstetikken som et af de første skridt.

Teknologisk fiksering af æstetikken

Vi har gennem mange iterationer udviklet interfaces, hvor Blender-teknologien er udnyttet mod en oplevelsesværdi. Blender er på den måde også et medie for den idemasse, vi har bygget interfacet med. Som nævnt har Blender understøttet de diskurser, vi har lagt i udviklingen, når f.eks. teknologien kunne rumme konceptet om et instrument som et gameplay. Interfacet er også et resultat af mange generationers editor-software (Blender-editoren), der iterativt er blevet udviklet mod at betjene sig af sidste æstetiske tiltag. Her tænker vi specielt på stilistik, hvor det bliver mindre kompliceret at lave avancerede lyseffekter, algoritmer for interaktion, *tiling*, osv. Som udviklingsværktøj mod at lave et meningsfuldt overlap mellem instrument og spil, med den enorme kompleksitet der ligger i denne sondring, har teknologien tjent formålet. Med andre ord er ideerne sjældent blevet for vilde til at Blenders grafik, interaktionsmuligheder og mulighed for afvikling er kommet til kort.

Historisk set har det været en udfordring at få kode og interface til at hænge sammen, hvilket er relativt ligetil i et udviklingsværktøj som Blender, og derfor også en relevant mulighed at aktualisere.

Procesbeskrivelse

Prototypeproces: Improvisation inden for et framework

Som en opsummering af prototypeafsnittet beskriver afsnittet her designprocessen som et tidsligt forløb, og hverken som et teoretisk eller metodisk fænomen. Forud for hele specialet går der som sagt sondringer i feltet musiksoftware generelt, hvorefter vi fikserer ideer til nye interaktionsformer. Herigennem bliver softwaren, vi har lyst til at lave, gradvist mere specifik igennem iterativ metode/teori udvikling. Metode og teori fungerer som en tankebase, vi løbende relaterer til konstruktion af prototyperne som en formmæssig sondring og en horisont for designet. På et tidligt tidspunkt har vi udvalgt Python på grund af tidligere erfaringer med sproget, og fordi muligheden for MIDI-implementering eksisterer. Herefter udvælges Blender hurtigt som kontrolflade, og så begynder udviklingen af de første modeller og koncepter. Teknologien har givet os et framework, hvor der findes muligheder og begrænsninger, hvilket giver os et grundlag at 'improvisere hen over'. De vilde ideer og drømme begrænses af teknologi, men teknologien giver samtidigt et udgangspunkt for hvad der kan kombineres, og hvad der kan indgå i improvisationen. Denne sondring vil metodeevalueringen behandle nærmere.

Tidslighed i udviklingen

Udviklingen af prototypen er som det eneste element i specialet karakteriseret ved ikke have en slutfase, da en prototype ikke er et færdigt *release*, og fordi vi vil holde designet åbent for forstyrrelser så lang tid som muligt. Mængden af input til designet er enorm, hvilket hele vejen igennem har været et potentiale for os, som vi ikke ville lukke på grund af tidslig stringens – ved f.eks at lukke prototypen for udvikling tidligere i specialeskrivningen. Ikke fordi vi er idéløse i forhold til designet og decideret har brug for input. Tværtimod har vi brug for en form for empirisk og teoretisk kontrol af vores ideer som en fortløbende selektion i idémateriale. Mulighederne for input har vi holdt åbne indtil sidste uge af specialeskrivningen. Installationsfilen er færdig en uge før skriveprocessen stopper, hvorefter specialet er lukket og sendt til print. Processen har været åben til allersidste øjeblik, og sidste iteration er foretaget i ellefte time. Den iterative udviklingsmetode med en åbenhed for forstyrrelser er lykkedes for os, da tidsplanen ikke er forskudt på grund af kode-problemer eller lignende. Samtidig har vi fået inputs, der kunne integreres i designet en uge før deadline. For at tage tråden op om *rapid prototyping*, kan det siges at værktøjerne har tjent formålet; det går stærkt, og komponentintegrationen virker, hvilket behandles i næste afsnit.

Fravalg og konsekvenser

Entrenchment og standardisering er begreber hentet fra teorier om *Social Shaping of Information and Communication Technologies*¹⁰⁷. I udviklingen af prototypen har vi overholdt designmæssige mål og tidsmæssige vurderinger i forhold til udviklingen. Dette ser vi som et resultat af at vælge teknologi ud fra hvad der er fastslået som en standard. Eksempelvis standarder som MIDI til at afvikle lyden/mappingen og OpenGL til at afvikle grafikken. Dette er standarder, der er *entrenched* (trench = grøft) og etablerede som de facto standarder. Vi ser valget af disse standardteknologier som 'skyld' i at prototyperne var færdige inden for tidsrammen. Dette skal ses i forhold til alternative designprocesser målrettet mod at skulle udvikle prototypen iht. eksperimenterende protokoller og standarder som eksempelvis OSC, eller en mere avanceret *game engine*. Gennem valg af standardiserede komponenter har vi sparet tid, og fundet et leje for vores evner som udviklere.

Dette afrunder beskrivelsen af designprocessen som en lang række formsomdringer og gradvise konkretiseringer ud fra teoretisk og empirisk funderede kriterier for æstetik og funktionalitet. Vi vil nu vende blikket mod metodens tredje fase, nemlig evalueringen af designet gennem test af prototypen. Kapitlet *Evaluering* vil endvidere behandle en evaluering og revidering af den metode, vi har anlagt for designprocessen.

¹⁰⁷ Williams, 1997

Evaluering

Test af prototypen

Mod afslutningen af prototyping-fasen arrangerer vi en test med tre musikere fra miljøet omkring elektronisk musik i Aalborg. Det drejer sig om Rasmus Allin, Bo Nicolaisen og Sune Petersen, som vi har tilknytning til bl.a. gennem musikerkollektivet *Søvnig Mandag*. I løbet af testen får de tre mulighed for at prøve *Containerturm* og *Phrasetrækker*, mens det bliver til et hurtigt kig på *Controllahtwistah* i en meget tidlig udgave. Samtidig spørger vi også til de tre musikeres vaner og ønsker ud fra samme tematikker, som vi anvendte i de interviews, vi udførte før prototypingen. Dette sker for at beskrive testpersonernes forventninger og arbejdsmønstre i sammenhæng med de konkrete reaktioner, vi får fra dem i testsituationen, og tjener samtidig til en nærmere præsentation af hver testperson.

Undervejs udvikler testen sig til en diskussion af bl.a. muligheder for revidering af designet, andre relevante teknologier og forskellige målgrupper for vores design. På den måde bliver testen ikke blot til en verificering eller afvisning af vores designhypoteser, men også en rig kilde til nye ideer og indskud.

Testen er dokumenteret ved lydoptagelser og noter gjort undervejs i testforløbet, samt testpersonernes besvarelser af de førnævnte spørgsmål. Noterne og besvarelserne findes som bilag, mens lydoptagelsen er tilgængelig på den vedlagte CD. Citaterne i dette afsnit stammer fra disse besvarelser og lydoptagelsen.

Testpersonerne

Alle tre testpersoner har en tilknytning til computerbaseret musik, og samtidig har alle tre også erfaring med design, enten af egen musiksoftware eller software generelt. Dette giver os gode muligheder for en diskussion af perspektiver og fremtidige potentialer til udvidelse af designet. Samtidig betyder dette dog også at vi har at gøre med musikere, der er vant til konfiguration og teknisk problemløsning, og måske har en højere tolerance overfor sådanne aktiviteter end andre musikere. Men gevinsten er at vi arbejder med testere, der både besidder viden indenfor domænet 'elektronisk musik', og som også har et erfaringsgrundlag med software som foranderligt materiale.

Rasmus er konservatorieuddannet bassist og arbejder med computeren i såvel produktion som performance. Sammen med tre andre spiller han i dub-bandet *Junkyard Production*, hvor han også komponerer numre. For at udvide computerens live-potentiale har Rasmus udviklet sit eget

software kaldet *Artillery* i *Pure Data*¹⁰⁸. Derudover arbejder han med *Logic*¹⁰⁹ som sequencer og produktionsredskab. Meget tidligt i testen beskriver Rasmus hvordan han savner en bedre kontrol end den, hardware-MIDI-controllere kan give ham. For det første giver de ham kun muligheden for at manipulere med en eller to parametre ad gangen, og for det andet holder manipulationen op i det øjeblik, drejeknapperne på en MIDI-controller slippes. For Rasmus er det vigtigt at kunne anvende sit udstyr og sin computer live; dette indebærer at det musiske materiale er åbent for manipulation og tilgængeligt via umiddelbare interfaces. Videre nævner han muligheden for inspiration og tilfældighed i såvel performance som komposition: *"I Logic får man jo ikke rigtig andet igen end det man selv putter ind, og det er jo faktisk ikke særlig inspirerende. Det kunne klart være fedt med nogle tilfældigheds-inspirations-maskiner. I mit live PD-setup¹¹⁰ har jeg jo lavet PflumScramplern, der jo mere eller mindre tilfældigt kan kaste rundt med et beat eller sequence. Dette er jo en stor hjælp live for at undgå ensformighed, men også som inspiration til komposition"*. Rasmus anvender en MIDI-controller med drejeknapper og tangenter som interface til *Artillery*, der for ham er væsentligt hurtigere end musen, som han finder *"langsom og upræcis, plus at den kan alt for få ting ad gangen"*.

Sune er uddannet dataingeniør og arbejder med video, musik og fotografi, bl.a. som idemand til musikerkollektivet *Søvnig Mandag*. Han har også arbejdet som pilot på Eksperimentariet i København. Sune anvender primært *Ableton Live*¹¹¹ og *Sonar*¹¹² på en laptop til sin musik. I sine svar på vores spørgsmål fortæller han, at han savner muligheden for mere avancerede routing-muligheder i Live, evt. som en kombination af et modulært miljø som Native Instruments' *Reaktor*¹¹³ med live sequencing. Noget, der bl.a. ville åbne op for styringen af flere lydparametre med tilfældigt generede værdier, som Sune også er interesseret i: *"Jeg bruger Live's random-funktion, men den kan snildt blive mere avanceret, det ville være fedt at kunne styre flere parametre med tilfældigheder"*. Sune fremhæver hardware-MIDI-controllere som et godt interface, men er mere skeptisk overfor at arbejde i et 3D-miljø med musik.

Bo arbejder til dagligt med bl.a. Flash-programmering og grafisk design, og laver musik i *Propellerheads' Reason*¹¹⁴ og *Ableton Live* på Mac. Bo er tilknyttet spillestedet 1000fryd som lydmand og koncertarrangør. Ligesom Sune nævner Bo muligheden for tilfældighed eller kaos

¹⁰⁸ <http://www.pure-data.org>

¹⁰⁹ <http://www.emagic.de>

¹¹⁰ Refererer til det førnævnte *Artillery*

¹¹¹ <http://www.ableton.de>

¹¹² <http://www.cakewalk.com>

¹¹³ <http://www.native-instruments.de>

¹¹⁴ <http://www.propellerheads.se>

under besvarelsen af vores spørgsmål, men vel at mærke kaos under kontrol: *"Jeg savner muligheden for at skabe kaos og så vende tilbage til kompositionen, jeg er lidt kontrolfreak, og måske også lidt for usikker endnu til at gøre det live uden sikkerhedsnet"*. Et godt interface er for Bo intuitivt – det skal være *"et interface, hvor input hænger sammen med output på en simpel og gennemskuelig måde, så man umiddelbart kan fatte hvad man hiver i vha. af auditiv og visuel feedback"*. Endelig understreger Bo det vigtige i at kunne bruge interfaces ekspressivt i performance-situationer: *"Jeg er blevet stadig mere opmærksom på publikums visuelle oplevelse og opfattelse af ens sceneoptræden, det er for mig vigtigt at komme ud over 'de spiller bare cd'ere og læser email deroppe'; at publikum kan se at man spiller"*. *"Min største forventning er til udviklingen indenfor man-machine interaktion, så man kan flytte sin performance lidt væk fra den rundryggede stance bag 'låget'"*.

Testforløbet

Testen og den efterfølgende diskussion forløber over ca. tre timer, hvoraf ca. en halv time er optaget som lydfiler indeholdende samtale såvel som auditivt output undervejs i testen. Den finder sted i et lokalt lydstudie i Aalborg, hvor vi har mulighed for at koble vores computere, andet musikhardware og ikke mindst højttalere sammen.

Testen falder i to halvdele: Den første del, hvor vi tester med vores egen udvikler-laptop, og den anden del, hvor Rasmus tester med udvikler-laptoppen som interface og sin egen Mac som MIDI-styret lydkilde. Hele forløbet er præget af en kontinuerlig vekslen mellem afprøvning og diskussion, og mange ideer opstår undervejs, som vi indfanger vha. lydoptagelser og noter. Afsluttende diskuterer vi fremtidige perspektiver og generelle ideer til udvidelse af designet. Testforløbet er altså her beskrevet i overensstemmelse med det faktiske kronologiske forløb.

Første halvdel af forløbet: Containerturm

I den første halvdel af testen foregår afvikling af såvel prototype som lyden, der modificeres, på samme computer, nemlig en af vores egne udviklingsmaskiner. Det auditive materiale består af forskellige synth-loops, der udsættes for et filter, der igen er kontrolleret af først *Containerturm* og derefter *Phrasetrækker*. Denne *mapping* er sat op ved testens begyndelse, men ændres undervejs ud fra deltagernes forslag.

På skift prøver Bo, Sune og Rasmus kræfter med de to interfaces. Reaktionen på *Containerturm* er umiddelbart at det er en interessant form for lydmanipulation, hvor de auditive konsekvenser er ret klare. Desværre er kontrollen dog også svær (Bo), og han foreslår at det kunne f.eks. være givende at kunne tage direkte fat i de to bolde, fremfor at styre dem indirekte med et bat.

Containerturm ligger for Bo på grænsen mellem værktøj og spil, da interaktionen med interfacet mere minder om det, man finder i computerspil. Alle tre eftersøger mulighederne for ekstremere (høje hastigheder, mange kollisioner, etc.), mens interfacet levner mindre plads til subtile parameterændringer. Også Sune og Rasmus finder graden af kontrol med lyden lille – Rasmus ser frustreret på, mens boldene gentagelsesvis flyver ud af kassen i *Containerturm*, når han giver dem slag med battet. Samtidig er Rasmus dog også positiv over for ideen med at *mappe* flere parametre (i dette tilfælde tre pr. bold) på én gang, da det udvider potentialet for modificering fra én eller to parametre ad gangen med en hardware-MIDI-controller. Ligeledes opfylder ideen bag *Containerturm* hans forventning om at kunne 'slippe' et parameter og stadig lade det forandre sig.

Første halvdel af forløbet: Phrasetrækker

I testen af *Phrasetrækker* bemærker alle tre testere at denne form for interface har en større grad af kontrol, der fungerer godt. Ligesom for *Containerturm* giver denne scene mulighed for samtidig kontrol af flere parametre (i alt 3), og kontrollen udøves stadig, selvom fokus flyttes til aktiviteter udenfor scenens interface (f.eks. i et sequencerprogram eller en plugin). Faktisk er dette potentiale for simultan kontrol med flere parametre en meget væsentlig faktor for vores testere. Bo foreslår, at ikke blot *Phrasetrækkerens* røde kugle, der bevæger sig mellem kontrolpunkterne, udsender MIDI-controller-værdier, men at også de fem kontrolpunkter udsender sådanne værdier, når de flyttes rundt af musikeren. På denne måde udvides kontrolmulighederne fra tre MIDI-controllere til atten ($5 * 3 + 3$) – og den umiddelbare grad af kontrol forstærkes yderligere, da den røde kugle kun indirekte styres af brugeren gennem kontrolpunkternes position, mens de fem kontrolpunkter flyttes direkte (dog kun ét punkt af gangen).

Manipulationen med kontrolpunkterne er dog også de tre testeres største anke ved designet. I den form som de tester, foregår interaktionen på den måde at et kontrolpunkt kan flyttes indtil det slippes, hvorefter det *næste* kontrolpunkt er tilgængeligt for flytning. Da begrebet 'næste punkt' ikke vises med en grafisk repræsentation i interfacet, er det meget forvirrende at skulle forudsige hvilket af de fem punkter, der ved næste museklik gøres til det aktive. Hvis man i stedet kunne flytte det punkt, der ligger *nærmest* ved musemarkøren, ville interaktionen være mere intuitiv og ukompliceret. Ergonomisk er interaktionen i denne form besværlig, mens konceptet bag *Phrasetrækkeren* er god, og har potentiale til videre udvikling.

Endelig diskuterer vi mulighederne for at gøre *Phrasetrækkeren* mere anvendelig som instrument i sammenhæng med anden software. Hvis f.eks. designet kunne implementeres som VST-plugin, ville det give en række muligheder som synkronisering med *hostens* tempo, muligheden for at

gemme og hente *presets* (færdige opsætninger af kontrolpunkter), *morph*¹¹⁵ mellem *presets* og *triggering*¹¹⁶ af *presets* fra *hostens* sequencer eller hardware-MIDI-controllere. Alt sammen egenskaber, der højner integrationen med det musiske materiale, og som giver flere muligheder for både kontrol og autonomi.

Bo bemærker også den grafiske udformning af *Phrasetrakkeren*, som han finder æstetisk 'lækker' og på den måde også anvendelig som en visualisering, et eventuelt publikum kan nyde synet af under en koncert. Rasmus og Sune bifalder ideen, der kan give et mere direkte link mellem laptop-musikere og de *visuals*, der ofte anvendes i performances. Når disse *visuals* ændrer sig synkront med musikken etableres en æstetisk redundans, ikke kun for kunstneren, der anvender interfacet, men også for publikum, der kan få et indblik i noget af det der foregår i performancen. Dette er et perspektiv, som vi ikke umiddelbart har indtænkt i vores oprindelige designs anvendelsesområde, men set i sammenhæng med et generelt ønske om mere interessante performances for publikum er det et væsentligt indskud til gentænkning af designet.

Anden halvdel af forløbet: Test med Artillery

Sune og Bo har ikke deres egne computere med, mens Rasmus som medindehaver af lydstudiet har sin Mac, som til sidst i testen kobles på testcomputerens MIDI-output fra prototypen, og dermed bliver underlagt andre kontrolmuligheder. På Mac'en kører Rasmus sit eget *Artillery*, hvori han bl.a. *mapper* et sample-baseret glitch-instrument (PflumScrampleren), delay- og rumklang-effekter til *Phrasetrakkerens* MIDI-output. Rasmus afprøver en række forskellige samples, og støder undervejs på et tilfældigt kort sample fra en studie-session, der indeholder starten af et beat og en smule tale. I *Artillery* bliver dette udsat for 'fraser' fra *Phrasetrakkeren*, og tilsammen dannes et loopet beat bestående af tromme-slag og vokalbidder, der aldrig helt gentager sig selv. Rasmus bemærker at det næsten er et nyt nummer, han sidder med, måske grundlaget for et nyt stykke musik.

Muligheden for positiv overraskelse såvel som relativ autonomi fra softwarens side virker umiddelbart tilstedeværende for Rasmus, og vurderes positivt i forhold til det materiale og den lydkilde, som han har valgt at arbejde med. Ligeledes modsvarer kontrollen med tre simultane parametre hans forventning om at kunne kontrollere flere lydparametre ad gangen. Selvom "*musen generelt bare er irriterende*" ifølge Rasmus, tyder det altså på at vi gennem softwaren kan udvide musens potentiale som redskab for lydmanipulation.

¹¹⁵ En gradvis overgang fra én tilstand (f.eks. ét preset) til en anden

¹¹⁶ Beskriver muligheden for at skifte preset eller tilstand i interfacet 'udefra', f.eks. vha. MIDI-notes eller *Program Change*-MIDI-data

Rasmus foreslår et par konkrete tiltag til yderligere kontrol i interfacet: For det første at kunne ændre de tildelte MIDI-controller-numre direkte i brugerfladen (hvad der modsvarer en tilsvarende mulighed på det hardware-interface, Rasmus bruger). Og for det andet at kunne slå selve genereringen af MIDI-output til og fra i interfacet, evt. for hver parameter ad gangen. På denne måde ser Rasmus en mulighed for at operere med pludselige, bratte dynamikker i sin musik – f.eks. hvis en række parametre er bundet til forskellige instrumenters volumen, der så med ét klik bringes fra et nulpunkt til løbende modulering.

Modsvaret til behov og ideer til formændringer

Ud fra testforløbet som helhed kan vi identificere en række centrale områder, som forekommer væsentlige for vores testere, og som på forskellig vis er modsvaret i vores design. Det drejer sig om:

- Kontrol og ergonomi
- Performance, ekspressivitet og visualisering
- Tilfældighed som kilde til inspiration og overkommelse af ensformige forløb i musikken
- Integration med det musiske materiale såvel som anden software

Softwaren som instrument skal være intuitiv og ligefrem - transparent. Men ligeledes er der et behov for kontrol med mange simultane parametre, og løbende kontrol med disse parametres *mapping* til lydegenskaber. Så vidt som vi kan tale om et stykke softwares ergonomi, drejer det sig altså om at finde en balance mellem kompleksitet og enkelhed. Når de specifikke problemer med f.eks. kontrolpunkterne i *Phrasetrakkeren* løses, lader det til at vi med dette interface har ramt en sådan balance. Med de foreslåede udvidelser af *mappingen* fra 3 til 18 MIDI-controllere stiger kompleksiteten, men i en formgivning der virker brugbar. Ligeledes kan *Containerturm* forbedres i retning af bedre kontrol med de to bolde, hvorved kontrollen og transparensen kan højnes. Uden at øge dens kompleksitet kan *Containerturm*-scenen med disse forbedringer blive til et værktøj, der tilvejebringer et kontrolleret kaos i en anvendelig brugerflade.

Derudover er ergonomien også givet ved musen som *input device*, der ikke i sig selv er optimal til kontrol med mange parametre, eller kan operere uden konstante input fra musikeren. Men gennem vores design kan musens potentiale for dette udvides. Blot bør vi være opmærksomme på at musens potentiale ikke kan strækkes i det uendelige, hvad der også kan ses af at testpersonerne

efterspørger integration med de hardware-MIDI-controllere, de allerede har integreret i deres setup og arbejdsmønstre.

Behovet for visualisering og ekspressivitet i performances kan forsøges modsvaret ved anvendelsen af vores interface som visualisering, hvad der som sagt kan give publikum en form for abstrakt indblik i den musiske proces, der i laptop-regi ellers kan være svær at gennemskue. Selve ideen med at lade vores software tjene såvel et instrument-formål som et visualiserings-formål skyldes vi vores testpersoner, og den giver anledning til mere fokus på dette behov for udtryksfuldhed i performances, som den udspringer fra.

Balancen mellem tilfældighed og kontrol kan afføde inspiration og modvirke ensformige musiske forløb. I Rasmus' arbejde med *Phrasetrakkeren* opstår der en grobund for ny musik gennem mødet mellem et anderledes interface og Rasmus' eksisterende software-instrumenter og lydmateriale. I forhold til en testsituation er det også en fordel, at Rasmus kan se såvel sin egen som vores software, og dermed se prototypen iht. egne arbejdsmønstre, altså en form for selvreference. I kompositionsfasen kan designet tjene som kilde til selvvalgt forstyrrelse. Ligeledes ser vores testpersoner et potentiale for modvirkning af repetitive forløb i live-performances. Sammen med den simultane kontrol af mange parametre giver dette muligheden for mere vidtgående og direkte manipulation med det musiske materiale i såvel komposition som performance. Men samtidig betoner bl.a. Bo relevansen af at kunne lade softwaren assistere i højere grad, hvorved behovet for input og overblik mindskes: "*Et interface med controllere, hvor man kan gemme et sæt presets, man kan klikke på, ville være kanon cool. Så kan man flippe ud som man har lyst eller bare køre på rutinen ;)*".

Endelig betyder integration med hver enkelt musikers software-setup meget. Den helt basale integration er faciliteret gennem MIDI-standarden, mens mere avancerede former for integration vil være mulig gennem f.eks. VST-implementering, der bl.a. ville muliggøre tempo-synkronisering, *presets* og at gemme og hente indstillinger for vores software sammen med det specifikke stykke musik, som musikeren arbejder med. Det spektrum af software, som vores testpersoner anvender som *hosts*, centrerer sig om VST-standarden (med undtagelse af *Reason*), der derfor kan fungere som teknologisk fiksering af et videre potentiale til udvidelse i retning af større integration med såvel arbejdsmønstre som det konkrete musiske materiale.

Udvidelse af designet – nye perspektiver

Udover de umiddelbare modsvar til behov, som prototypen illustrerer vores hypotese om, og som gennem testen er vurderet af vores testpersoner, tjener prototypen også som kilde til nye ideer. Vi

har allerede beskrevet en række konkrete ideer fra vores testpersoner til udvidelse af de to scener, *Phrasetrækker* og *Containerturm*, herunder ideer til udvidelse af interaktionen, integration med andre miljøer, forbedring af ekspressiviteten og visualisering.

Men derudover formulerer vores testere også en række ideer til designet som helhed – ideer, der ikke omhandler detaljerne i vores konkrete formgivning, men nærmere perspektiverne i de koncepter, vi har baseret vores formgivning på. I forlængelse af ideen om softwaren som visualisering af musikken beskriver Sune og Bo et perspektiv, der flytter målgruppen for designet til f.eks. eksperimenterier, hvor interaktionen med interfacet kan tjene et formidlende eller underholdende formål. Selvom designet beholder sin eksisterende form, kan den altså få et potentiale i andre brugskontekster end den, den er tiltænkt.

Ligeledes bider alle tre testere mærke i, at vores design nok primært henvender sig til computer-musikere med lyst til at eksperimentere med deres software-setup, herunder den relativt frie konfiguration og *mapping* af komponenter gennem MIDI-protokollen. Skulle vi tiltænke designet en anden musiker-målgruppe med mindre lyst til dette, kunne det derfor være relevant at gøre *mappingen* fastdefineret og binde interaktionen til en bestemt effekt eller lydkilde. Dermed frigøres opmærksomheden fra arbejdet med konfiguration til en umiddelbar anvendelse af interfacet – det bliver lettere at anvende, men på bekostning af fleksibilitet.

Potentialet i integration med *host*-software og musens iboende begrænsninger som musisk interface leder også mod en række ideer til udvidelse af designet gennem integration med anden hardware. Testpersonerne nævner *input devices* som *touchscreens*, joypads/joysticks, 3D-mus og *motion tracking*. Rasmus understreger at det fysiske i interaktionen er vigtigt: ”*Det ideelle ville klart være et fysisk 3D-interface*” – ”*I fremtiden vil der blive lavet en fysisk 3D-controller af modellervoks, så man med hænderne fysisk former musikken. Når man så afspiller musikken vil modellervoks-klumpen bevæge sig (måske). Ellers kunne man vel uden særligt besvær lave en 3D-theremin¹¹⁷, der sender MIDI-tal i stedet for lyd*”. Selvom nogle af disse interfaces enten er dyre eller helt ikke-eksisterende, er det alligevel relevant at se på dem som inspiration til udvidelse af designet. Måske kan nogle af disse fysiske interfaces’ egenskaber inkorporeres i et redesign af vores virtuelle interface. Og måske kan nogle af dem tjene til en bedre kontrol med aktiviteterne i 3D-rummet, noget som testerne efterspørger. F.eks. ville joysticks eller 3D-mus udgøre sådanne muligheder, mens vi kan hente inspiration til redesign af den virtuelle verden hos f.eks. *motion tracking*-apparaturl og Rasmus’ modellervoks-interface.

¹¹⁷ En theremin er et tidligt elektronisk musikinstrument, opfundet af Lev Sergeivitch Termen i 1917. Se f.eks. http://www.obsolete.com/120_years/machines/theremin/

Understregelsen af det fysiske interfaces rolle i musikken hænger som sagt sammen med behovet for ekspressivitet og tilstedeværelse. Ser vi igen dette ønske i sammenhæng med perspektivet i at indtænke en målgruppe som f.eks. gæster på et eksperimentarium e.l., flytter vi absolut den indledende ide og motivation for designprocessen til noget ganske andet. Alligevel er denne rekontekstualisering eller gentænkning af designet væsentlig, da den for det første kan være med til at udpege uventede anvendelsesområder, og for det andet kan gøre os opmærksom på de helt grundlæggende designvalg, der allerede er indtruffet i designets tidligste faser. Hvor vores opmærksomhed har været rettet mod kontrol, ergonomi og generering af tilfældighed for den computer-musiker, der ønsker at anvende tid på rekonfiguration og *mapping*, flytter perspektiverne i testernes udsagn vores opmærksomhed over på disse emner:

- Ekspressivitet forstået som visualisering for et publikum, hvorned der etableres en meningsmæssig sammenhæng med musikken gennem æstetisk redundans.
- Designets potentiale for formidling eller underholdning som interaktivt redskab til leg med musik eller lyd, f.eks. i eksperimentarie-sammenhæng.
- Designets potentiale for tættere integration med specifikke lyd-algoritmer, f.eks. som et interface bundet til en specifik VST-plugin.
- Forbedring af *control intimacy* gennem (anderledes) hardware-interfaces og redesign af den virtuelle verden.

Evaluerings af prototypen og testen

Testen leder os frem til nogle umiddelbare eksterne vurderinger af prototypens form, og samtidig til vigtige ideer til redesign og gentænkning af perspektiverne i designet. Dermed er vejen fra 'artefakt i kontekst' til 'muligheder og forventninger' for designet tilvejebragt gennem såvel vurdering af det *eksisterende* design som en indikation af det *ikke-eksisterende* design (*imaginary states*), set ud fra de 'forstyrrelser', som uventede reaktioner udgør.

Denne intention med prototypen – at fungere som kilde til både vurdering og inspiration – er blotlagt i vores metode og i høj grad opfyldt i den faktiske test. I særdeleshed er vores testere med til at udpege specifikke teknologiske muligheder for redesign (konkrete forbedringer af interaktionen, alternative hardware-interfaces) såvel som andre anvendelsesområder, der indebærer nye kontekster for designet. Grunden til dette idemæssigt frugtbare testforløb kan bl.a. være, at vi har fået mulighed for at arbejde sammen med testere, der som vi selv har viden om såvel musisk som softwaremæssig formgivning. Krydsningen af disse to vidensdomæner frembyder en kilde til

nye ideer, ligesom prototypens møde med 'fremmede øjne' udgør en anden frugtbar krydsning af grænsen mellem det systeminterne og det systemeksterne. Den umiddelbare ulempe ved at spørge folk med forudsætninger, der ligner vores egne, er at vi risikerer at få svar som vi spørger. Men med påvisningen af nye ideer, vi end ikke havde tænkt på i processens tidligste faser, synes det ikke som om vi reelt er gået i denne fælde. En anden umiddelbar faldgrube er en blindhed for andre kontekster eller anvendelsesområder, der kan indtræffe når vi tester med personer med arbejdsmønstre, der ligner vores egne. Men når disse testpersoner netop formulerer helt uventede, anderledes målgrupper for vores design som en del af testforløbet, må også denne faldgrube siges at være undgået. Kort sagt er de videnskabsmæssige og kulturelle overlap mellem os selv og vores testpersoner ikke en ulempe, vi i praksis har oplevet. Tværtimod bliver der mulighed for de nævnte krydsninger, hvilket som sagt afføder reaktioner, ideer og perspektiver.

Koblingen fra de muligheder og forventninger, som testen indkredser, og til gentænkningen af den oplevelse, vi gerne vil understøtte med designet, er imidlertid ikke klarlagt umiddelbart ved testens afslutning. Og det er heller ikke vores intention. De udsagn og perspektiver, vi indsamler gennem testen, peger i divergerende retninger og repræsenterer anslag til forskellige designprocesser. Udvalgelsen af én retning fremfor andre må bero på såvel testen som vores egne ideer – der sammen kan blive den initiale ide for videre iterationer af designprocessen. Dermed må vi umiddelbart konkludere at testen har fungeret efter denne hensigt – at skabe såvel vurderinger som nye ideer, og at tjene til videre, systemintern selektion af en retning for designet. Tilbage står at diskutere:

- Hvordan designet som helhed er relevant for testpersonerne.
- Hvordan testen af prototypen formidler designet for testpersonerne – fungerer testen som repræsentation af et 'artefakt i kontekst'?
- Hvordan vi på baggrund af testen kan skabe ideer til videre iterationer af designet.

Designets relevans som helhed

Som helhed har designet en relevans som ekspressivt, interessant og anderledes interface. Testen peger i retning af en umiddelbar genkendelse af de scenarier, der bl.a. ligger bag designet, hvor især *tweaking*-scenariet og konfigurations-scenariet i forvejen er vante arbejdsituationer, og hvor scenariet omkring det kontrollerede kaos vækker testernes interesse for anderledes måder at arbejde med lyd på. Det eksisterende design, vi præsenterer i testen, giver mening på trods af dets skønhedsfejl og indimellem mindre intuitive interaktionsformer.

Potentialet for *realtime*-kontrol med lydparametre modsvarer et lignende potentiale i andre, mere etablerede interfaces som f.eks. hardware-controllerboxe, og forbindes af vores testere tydeligt til performance-situationer. Muligheden for introduktion af tilfældighed modsvarer dog også et behov, der er udtalt i kompositionssammenhæng, hvor det kan tjene til inspiration og direkte forme det musiske materiale under musikerens udvælgelse af ét element for manipulation fremfor et andet.

Udover designets direkte modsvar til disse formulerede behov for kontrol, ekspressivitet og introduktion af tilfældighed, ser vi dog gennem testen dog også et andet behov, der ikke blot involverer computeren som interface eller lydkilde, men hele den kontekst, vores testpersoner som musikere opererer i. Det drejer sig om behovet for visualisering og udtryksfuldhed i performances. Hvor vores interface i første omgang er tænkt ud fra interaktionen mellem musiker og instrument, rummer den videre interaktion mellem musiker og publikum – bl.a. gennem instrumentet som 'sted' for ekspressive former – en række andre behov for indblik i den musiske proces, en visuel og auditiv sammenhæng, og integration med eksisterende udtryksformer som *visuals*¹¹⁸.

Prototypen som repræsentation af et design

Testen tjener umiddelbart et basalt formidlings-formål, nemlig at gøre den oplevelse, vi designer imod, tilgængelig for andre, hvilket kræver en fiksering af designet i konkret teknologi, æstetik og funktionalitet.

Men derudover kræver formidlingen også rammer om selve testen, der illustrerer designets mulige relevans for vores testpersoner, og som taler til deres evne til at forestille sig et anderledes design. De fysiske rammer om testen er et musikstudie, hvor alle tre testpersoner har været eller arbejdet før. Det er vante rammer, og selvom det er os, der udpeger lokaliteten for testen, må det alligevel være muligt at formidle designet i forhold til eksisterende arbejdsmønstre, da vi forankrer testen i sådanne gennem udvælgelsen af et sted, der netop ikke er en laboratorie-*setting*, men ligner eller er testernes vante omgivelser for musisk aktivitet.

Videre forankrer vi testforløbet i de scenarier, der indgår i designets informationsbasis. Testforløbet første halvdel er baseret på *tweaking*-scenariet, hvor et repetitivt musisk forløb (et loop) underlægges manipulationer af klangfarven gennem modulering af et filters parametre. Med testen af *Containerturm* underlægges klangfarven også en relativt autonom behandling gennem simuleringen af tyngdekraft og dennes *mapping* til klangparametre. Dermed inddrages det scenarie, der illustrerer den navigation mellem kontrol og kaos, som vi ønsker at tilvejebringe i designet.

¹¹⁸ Videokunst, der ofte projiceres på en væg bag musikeren/musikerne under en koncert

Endelig illustreres konfigurations-scenariet gennem Rasmus' anvendelse af selvvalgt musik materiale og instrumenter gennem mapping over MIDI-protokollen.

Vores formgivning af testforløbet betyder altså at vi præsenterer prototypen i lyset af de arbejdsmønstre, vi gennem empirisk og teoretisk informationsindsamling har gjort til en del af den informationsmæssige basis for designet. Ser vi testforløbet som et kommunikativt forløb mellem designere og testere, betyder dette at vi præsenterer nogle hypoteser om arbejdsmønstre og hvordan disse kan understøttes. Et af resultaterne af testen bliver en vurdering af, i hvor høj grad disse scenarier rent faktisk er understøttet, og hvilke forbedringer, der kan gøres efterfølgende for at forbedre denne understøttelse. Som kommunikativt forløb og repræsentation af en designhypotese er testen først og fremmest sårbar overfor *breakdowns* i denne kommunikation, som kan opstå pga. *bugs* eller dårligt designede aktiviteter i testforløbet.

Sådanne *breakdowns* er vi tæt på flere gange undervejs i testen. Først og tydeligst ved testernes forvirring over interaktionen med *Phrasetrækker*, hvor metoden til flytning af kontrolpunkter ikke er intuitiv. Her kan testforløbet bryde sammen og prototypen virke uforståelig og irrelevant. Men i praksis oplever vi at testpersonerne selv foreslår den adfærd, vi havde håbet på at implementere, nemlig en mere direkte interaktion med kontrolpunkterne, og dermed viser os, at konceptet bag *Phrasetrækker* giver mening.

Et andet illustrativt eksempel er Rasmus' opsætning af *Artillery* som lydkilde. Her arbejder vi sammen om at løse mappingen fra prototypens MIDI-output til *Artillerys* MIDI-input, og ad den vej etablere en sammenhæng mellem lyd og interface. I stedet for at vurdere vores interface som umiddelbart irrelevant, beder Rasmus os ændre mappingen af MIDI-controllere¹¹⁹, så denne stemmer overens med *Artillery*. Situationen betyder ikke et brud på kommunikationen, men leder nærmere til at Rasmus foreslår en designændring; nemlig at mappingen af controller-numre gøres direkte tilgængelig i interfacet.

Selvom vi i princippet derfor bevæger os tæt på en dårligere test, fordi prototypen på detaljeniveau ikke altid opfylder hverken vores egne eller testernes forventninger, realiseres de nævnte *breakdowns* ikke i praksis. Bl.a. fordi vores testpersoner har en forventning om at designet er fleksibelt nok til at kunne ændres på detaljeniveau undervejs i testen (hvilket kan ske i Rasmus' tilfælde), og fordi de er med til at forestille sig andre formgivninger og præsentere disse som ideer for os. Igen beror testforløbets værdi for redesign lige så meget på testernes evne til 'designer med',

¹¹⁹ Testforløbet foregår før vores *Preferences Editor* er implementeret. Var den tilgængelig, ville det have været muligt at ændre mappingen gennem denne

som på vores formgivning af testforløbet – måske endda mere. Testen fungerer som repræsentation af et 'artefakt i kontekst', men det er ikke kun vores egen fortjeneste.

Muligheder med relevans for redesign

De specifikke forbedringer af funktionalitet og æstetik, som testen opridser, tager vi ind i designprocessens informationsflow som umiddelbare designrevideringer, der bygger på blotlagte kriterier og arbejdsmønstre hos de musikere, vi har testet med. De bygger på observationer gjort af testpersonerne, af f.eks. ergonomiske forhold eller potentialet for kontrol, som giver mening indenfor den tankeramme, som det nuværende design repræsenterer. Som information betragtet er de dermed indarbejdet i designet og tilgængelige for videre iterationer.

De mere vidtgående muligheder for redesign, der ligger i testernes nye perspektiver for designet, kræver en mere reflektiv behandling for at vi kan navigere mellem de divergenser, som de repræsenterer. Hver især bryder de med den eksisterende tankeramme for designet, hvilket ansporer en analyse af de forskellige perspektivers potentialer for videre iterationer.

Såvel visualiserings-perspektivet som eksperimentarie-perspektivet tenderer mod en opfattelse af interfacet som en form for kunst. Enten i kraft af en binding til koncerten eller performancen, der ses som en kunstnerisk udfoldelse for musikeren såvel som grafikeren bag visualiseringen; eller i kraft af eksperimentariets overlap til museet, der institutionaliserer kunstoplevelsen og gør den tilgængelig for et publikum. Modsat har vi perspektiver, der peger i retning af integration med anden software og hardware, og som antyder en tendens mod *brug* fremfor *kunst*, hvor interfacet i højere grad gøres til et værktøj for musikeren.

På den anden side kan vi i vores testpersoners udsagn skelne mellem tendenser, der fokuserer på musikeren som bruger af softwaren, og andre tendenser, der lægger vægt på softwarens potentiale for et publikum. Altså hhv. et *udøver-* og et *publikums-*perspektiv.

Tilsammen kan disse distinktioner tjene til kortlægning af mulighederne for redesign, og derigennem skitsere nogle mulige finite former. Sætter vi distinktionerne overfor hinanden, som det er gjort i skemaet nedenfor, danner vi et grundlag for kombinationer, der fører til nye, konkrete ideer ud fra mødet mellem perspektiver. Disse ideer omfatter bl.a. interaktiv kunst, implementering af et *gameplay* og didaktisk formidling, såvel som udvidelser af softwaren som instrument, og muligheden for at instrumentet også muliggør kontrol med de visualiseringer, som publikum kan opleve. Skemaet illustrerer for hver kombination af distinktioner én eller flere ideer til gentænkning af designet:

	<i>Målgruppe: Udøver</i>	<i>Målgruppe: Publikum</i>
<i>Fokus: Kunst</i>	Et design, hvor musik og grafik kobles og gøres tilgængelig for manipulation af udøveren; et instrument, der genererer grafik såvel som musik. Designet er rettet mod en integration af kunstnerens manipulationsmuligheder i en audiovisuel performance.	Interaktiv kunst; specifikt udformet 'interface' ud fra kunstneriske intentioner og evt. med delvist fastlagt musisk indhold. Designet bevæger sig helt eller delvist fra 'instrument' til 'værk'.
<i>Fokus: Brug</i>	Et design rettet mod mere kontrol, hvor rækkevidden og konsekvensen af interaktioner udvides, og der gives flere direkte manipulationsmuligheder. Eller modsat, et andet design, hvor kontrollen med kaos frigives, og interfacets autonomi bliver større. Evt. som et helt autonomt interface, hvor mere eller mindre uforudsigeligt musisk indhold genereres uafhængigt af musikerens input.	Et design rettet mod formidling eller underholdning; f.eks. kan musiske principper formidles didaktisk, eller designet kan rettes mod skabelse af et <i>gameplay</i> mhp. underholdning. F.eks. ved at integrere flere scener i en samlet verden, som brugeren kan bevæge sig rundt i.

En anden distinktion kan indlægges på tværs af de ovenstående; nemlig en skelnen mellem designs rettet mod *stand-alone* brug overfor designs rettet mod kollaborativ brug gennem anvendelsen af netværksteknologi. Hvor de ovenstående ideer umiddelbart kan læses som *stand-alone* designs, giver kollaborationsperspektivet anledning til ideer som:

- Et kollaborativt virtuelt instrument, hvor flere musikere kan interagere sammen, f.eks. over et IP-netværk gennem OSC-protokollen.
- Et lignende virtuelt instrument, hvor musikere og grafikere kan kollaborere omkring skabelsen af sammenhængende musik og grafik. F.eks. i et web-baseret interface.
- Interaktiv kunst, hvor publikum kan interagere med hinanden.

- En forstærkning af et musisk interface med *gameplay* gennem muligheden for at flere brugere kan spille sammen i den samme virtuelle verden.

Alt i alt afføder mødet mellem perspektiver en række væsentlige ideer, hvor det næste trin vil være at udvælge én eller flere af disse ideer og formulere en basis for videre design, der hænger meningsmæssigt sammen, og hvor de empiriske, teknologiske og æstetiske erfaringer, vi har gjort os gennem designprocessen, kan involveres som vidensbasis.

Vi står imidlertid ved designprocessens umiddelbare afslutning: Disse ideer til videre design kan vi vende tilbage til udenfor speciale-sammenhæng, men indenfor specialets sammenhæng er ringen så at sige sluttet, og vi vil i stedet vende blikket mod en evaluering af metoden, der netop omhandler ringen som helhed.

Evaluering af metoden

Vi etablerer ikke kun designprocessen for at arbejde mod et interface-design med relevans for os selv og andre musikere, men også for at bruge den som *case* for en metodisk udvikling. Dette afsnit er møntet på en refleksion over den anvendte metode og munder ud i en række perspektiver til revidering af metoden som arbejdsredskab – videre iterationer af metoden.

Metode versus proces

Metoden er ikke lig med den proces, den søger at beskrive og anvise. Processen divergerer, fordi den etableres i en tidlig og kompleks virkelighed, der ikke altid lader sig forenkle sådan som metoden implikerer. Med andre ord tjener metoden som et deskriptivt og preskriptivt kort over en processuel virkelighed, der hverken i sine detaljer eller helhed altid lader sig indfange så stringent, som vi gerne ville.

Dette er ikke en undvigelsesmanøvre i retning af et forsvar for den faktiske metode, men et anslag til en refleksion over metodens relevans i den langstrakte helhed, som vores proces udgør. Tidsligt set spænder vi over et forløb, hvor metoden som udgangspunkt kan tjene som planlægningsværktøj, men senere hen nærmere virker som et kort over de handlinger, vi allerede har foretaget os, og beskriver deres relevans for de trin (udsigtspunkter), der endnu er forestående. Dermed etablerer anvendelsen af metoden en kausal sammenhæng mellem de aktiviteter, vi sætter i værk – forstået på den måde, at metoden blotlægger et rationale og en vej for videre trin i processen. Eksempelvis klargør metoden meningen med at arbejde mod en prototype som repræsentation af designet – nemlig for at indsætte et håndgribeligt stykke software i en kontekst for afprøvning og kommentarer.

Skabelsen af en kausal sammenhæng fungerer således både som redskab til planlægning og begrundelse for processens form. Men i den faktiske designproces træder denne kausalsammenhæng i baggrunden for en sammenhæng bundet til en række *tematikker*. Disse tematikker beskriver de foki, vi opererer med som designere, og er baseret på de æstetiske, teoretiske og empiriske grundlag, vi har udvalgt. Nogle begrebsmæssige kerner opstår, og omkring disse udspiller de fire faser i metoden sig. Forskydninger af disse tematikker kommer først og fremmest på tale gennem forstyrrelser udefra. I den første kortlægningsfase orienterer vi os mod kilder til fokusering af vores design, i form af såvel empiri som teori. Og i afprøvningens kortlægning af brugerrespons orienterer vi os mod en vurdering af det eksisterende design, samtidig med at vi søger at åbne designet for andre perspektiver, og dermed flytte de tematikker, der udgør vores tankeramme om designet.

Der tegner sig et billede af to diskurser. På den ene side metodens kausale diskurs: ”Først gør vi A, så B, fordi...”, og på den anden side processens tematiske diskurs, der søger indholdet i de begreber, der efterhånden bliver en kerne for designprocessen. Jo længere væk vi bevæger os fra den indledende formulering af metoden, jo længere bevæger vi os over i den tematiske diskurs og dermed den begrebsverden, som designets faser centrerer sig om. Tematikkerne binder designprocessen indholdsmæssigt sammen over dens tidsspænd og kommer først og sidst til syne i de ideer, konstruktioner og perspektiver, der præger processen. Metodens relevans er en tidlig forankring af den fremgangsmåde, der leder os til at arbejde indenfor de udvalgte tematikker – men det er tematikkerne, der fører til en retning for designet, egenskaber for konstruktionen og kriterier for afprøvningen.

Forholdet mellem metode og proces er samtidig et forhold mellem skitsering og realisering. Vores erfaring er at processens realiseringer orienterer sig mod langt flere sondringer end de, der er indeholdt i metodens skitsering (som f.eks. sondringen mellem brugercentreret og designercentreret design, eller mellem åbenhed og lukkethed i kommunikationen med omverden). Disse andre sondringer, som er de designspecifikke tematikker, bliver meget mere betydningsfulde, jo længere vi arbejder os ind mod en kerne i designet, og væk fra et behov for en generel skitsering af processen. Når vi bevæger os tæt på den konkrete konstruktion, betyder det en bevægelse mod en kompleks udvælgelse af nogle egenskaber fremfor andre. Efterhånden sker denne udvælgelse på konstruktionens detaljeniveau, og designsystemet lukker sig om processen. Derefter åbnes processen gradvist igen gennem åbningen mod testpersonernes ideer til videre design. Men stadig opererer vi i henhold til de etablerede tematikker i udvælgelsen af punkter med relevans for videre iterationer. Endelig kan vi på et refleksionsniveau iagttage, hvordan vi anvender tematikkerne, og hvordan de binder processen sammen. Dette medfører en opmærksomhed på andre relevante

tematikker. Og i sidste ende et blik for selve det at orientere designprocessen mod den diskursive form, som brugen af tematikker udgør.

Bevægelsen fra 'abstrakt' eller 'åben' til 'konkret' eller 'lukket' og tilbage igen er en realisering af vores ide om at veksle mellem refleksion og konstruktion. Samtidig gør den os dog også opmærksom på det forhold, at designprocessen på et tidspunkt i praksis orienterer sig mere indad end udad, nemlig i konstruktionens vidtstrakte arbejde fra koncept til detaljeret prototype. Da vi i den faktiske proces veksler mellem konstruktion og refleksion meget ofte, betyder dette et overlap mellem faser. I denne transit skifter vi mellem faser, der orienterer os mod omverden i form af empiri og teori, og faser, der er præget af internt udviklingsarbejde. Den mindre stringente adskillelse af de to momenter kan medføre, at relevante åbninger overses i arbejdet med konstruktion, eller at åbninger udadtil sinker konstruktionsprocessen. Vi arbejder på et meget konkret konstruktionsniveau, hvorved muligheden for refleksion forsvinder. Sammenhængen i processen opstår i langt højere grad gennem tænkning i tematikker, end i tænkning i kausaliteter.

Informationsbehandling

Behandlingen af information er et vigtigt moment i hele opbygningen af metoden såvel som realiseringen af processen. Involveringen af omverdenen for designet, såvel som opbygning af en informationsbasis for designet, sker gennem observation og behandling af information i form af empirisk eller teoretisk materiale. En grundlæggende aktivitet er dermed observation i bred forstand; både observation af empiriske forventninger, arbejdsmønstre og setups, såvel som observation af teoretiske pointer og sondringer.

Ligesom tematikkerne gennemsyrrer observation derfor designprocessen og dens fremdrift. Observationerne af teoretiske sondringer er kendetegnet ved en læsning af teorier, der er rettet mod design, og i den forstand læser vi teorierne mere som inspiration og muligheder, end som anvisninger for design. Samtidig er vores teoretiske grundlag for designet præget af en stor grad af kombinatorik, hvor vi læser en del forskellige teorier sammen til en helhed, der rummer information med relevans for netop vores design. Begge disse perspektiver på observation gennem læsning betyder, at vi inddrager teori selektivt og reflektivt. Udvælgelsen sker ud fra et relevanskriterie, der igen formuleres som en refleksion i forhold til de problematikker, vi har valgt at arbejde mod. Vi kan ikke direkte læse os fra *oplevelse* til *requirements*, men må i stedet læse konstruktivt og dermed etablere en sammenhæng mellem adskilte teoriområder, der har relevans for os. Den fremanalyserede vekselvirkning mellem f.eks. kontrol og *glitch* er ikke en anvisning på et design, og heller ikke en beskrivelse af en oplevelse, men derimod en tematik med rod i teori om bl.a. æstetik og instrumentdesign. Vekselvirkningen beskriver yderpunkter for en eksisterende praksis og samtidig muligheder for et design. Men vejen fra beskrivelse af praksis og muligheder til

konkret formgivning beror på en sammentænkning af flere tematikker, der tilsammen tegner et rum for designet. Anvendelsen af teori, der hjælper os til at beskrive denne og andre vekselvirkninger, sker således med reference til såvel udgangspunktet i den oplevelse, vi gerne vil understøtte, som den indsnævring af designet, som *requirements* udgør. Indsnævringen er ikke vilkårlig eller retningsløs, men båret af informationsindsamlingen, der ligger bag.

Observationerne af empiri indebærer ligeledes en selektiv og reflektiv behandling. Vi søger i lige så høj grad inspiration som vurderinger og beskrivelser, når vi går til musikere i omverdenen for vores design og spørger til arbejds mønstre, forventninger, osv., og giver dem mulighed for at vurdere designet i en test. Dermed selekterer vi i empiriske udsagn ud fra en eftersøgning af potentialer for videre design. Men samtidig bærer vores empiriske læsninger også præg af en refleksion, hvori vi må erkende og eksplicite kriterierne for vores kortlægning og test. Vi tester ikke for testens skyld, men for at lede designprocessen videre. Derfor er testen præget af eftersøgningen af information, der kan give os en indikation af værdien af vores design og samtidig forskyde vores tematikker i retning af alternative designs. I praksis er åbningen mod forskydning af tematikker lykkedes meget godt, idet vi er blevet beriget med en del divergerende perspektiver og uventede ideer til videre designs. Dermed har vi også nået et punkt for videre iterationer, hvor selektionen af nye områder for videreudvikling – og dermed orientering mod andre tematikker – udgør en egen kompleksitet. Aktualisering af en bestemt mulighed kræver et rationale eller et kriterie, der kan give videre iterationer retning, fremfor en vaklen mellem flere muligheder. Her savner vi i vores metode et udvælgelses kriterie, der kan håndtere den store divergens i mulighederne for redesign, og dermed mulighederne for videre innovation. En grund kan være, at dette udvælgelses kriterie ikke blot skal bygge på designvalg, men også inkorporere de omgivelser, der er for designsystemet. F.eks. kan økonomiske eller tidsmæssige hensyn indgå i formuleringen af et sådant kriterie, noget vi vil vende tilbage til under behandlingen af metodens muligheder for forankring i en organisation.

Metoden i et organisatorisk perspektiv

Dette afsnit er en videre refleksion over de punkter, der er indeholdt i Kalthoff og Nonakas innovationssfære. Vi har som designere og metode-byggere opereret relativt autonomt i forhold til den institution, processen også er forankret i, nemlig universitetet. Selve metodens opbygning hviler på et idemæssigt fundament, som er opbygget gennem tidligere projekters erfaringer, hvor også disse projekter er bundet til universitetets relative frihed fra økonomiske og markeds mæssige optikker på det design og den metode, vi anvender.

Men det er yderst relevant at forsøge at se metoden i et andet lys, nemlig i forhold til dens muligheder for anvendelse i en markedsorienteret udviklingsorganisation. Hvor designprocessen

kan operere autonomt i specialets regi, må vi i stedet indarbejde en række andre kriterier og forhold, når det gælder sådanne organisationer.

En basal orientering er her en skelnen mellem information og konstruktion. Information er ofte tilgængelig, men ikke altid mulig at inkorporere i designet pga. et organisatorisk fokus på konstruktion fremfor information. Vores pointe er for det første, at designet må basere sig på information for at sikre sig en relevans for omverdenen, og for det andet at denne informationsindsamling ikke nødvendigvis udgør den økonomiske og tidsmæssige udgift, som den ofte takseres til i organisatorisk regi. En empirisk basis for design behøver ikke være dyr eller tidskrævende, hvis den realiseres ud fra relevanskriterier og med fokus på inspiration såvel som faktuel information. Og en teoretisk basis behøver ikke genopdages for hvert nyt design eller hver iteration, men kan realiseres som organisationsinterne designmanualer, guidelines og erfaringer.

I praksis har vores designproces været præget af et mønster, hvor indledende åbenhed efterfølges af lukning omkring konstruktionsprocessen, efterfulgt af endnu en åbenhed i form af test og revidering af potentialet for videre udvikling. Alt dette varetages indenfor det samme designsystem, hvilket fastholder informationen i vejen fra kortlægning over konstruktion til test. Samtidig gør den tilbagevendende refleksion over informationen det muligt at fastholde designprocessens kontingens – at den kan udfolde sig anderledes, og f.eks. basere sig på helt andre tematikker.

I dette perspektiv kan den adskillelse mellem omverdensorientering og konstruktion, der ofte institutionaliseres i f.eks. en arbejdsdeling mellem marketing og udvikling, virke u hensigtsmæssig. Hvad vi har kaldt tematikker må kunne operationaliseres som orienteringspunkter i forhold til begge parter, hvilket kan medføre et potentiale for større sammenhæng mellem omverden og design.

Forankringen i en organisation betyder dog ikke kun en række begrænsninger ud fra økonomiske og markedsmæssige optikker. Som vi var inde på i det ovenstående, indebærer forankringen også en mulighed for at designsystemet kan orientere sig efter flere kriterier end de interne designkriterier. F.eks. kan den organisatoriske omverden for designsystemet bidrage med kriterier, der kan hjælpe til at udpege en retning for radikale gentænkninger af designet, når de designinterne kriterier ikke i sig selv kan udpege én retning fremfor andre. Men revideringen af designet må ske ud fra en tilstrækkelig informationsbasis og ikke blot organisationens metodiske eller vanebestemte hensyn til 'at sådan plejer vi at gøre'. Denne informationsbasis må derfor være delt og relevant for såvel designsystemet som andre delsystemer i den samlede organisation. Dette kræver en formgivning af såvel design-information som metode, der indtænker de mange modtagere og afsendere af information, der opererer i en videre organisatorisk kontekst.

Perspektivet i vores behandling af organisationer bunder i observationer fra 8. semesters praktikophold. Metode-nytænkning finder vi relevant, og er et essentielt kriterium for virksomheder, der f.eks. orienterer produkter efter tendens. Tendenser beskriver aktører i omverden, der har individuelle og diffuse behov, hvilket en organisations struktur og metodik må indpasses efter. Metodeperspektivet er at diffuse aftagermarkeder kalder på en anderledes fortolkning fra virksomhedens side, hvilket i henhold til specialet sker gennem iterativ implementering af information. Dette er ikke sensationelt nytænkende og banalt i mange virksomheder, men få virksomheder drøfter hvornår informationen skal implementeres. Virksomheder har en kvalitetssikring i form af alfa- eller beta-testere, f.eks. interne medarbejdere, der foretager tekniske/funktionelle test. Dette tilvejebringer information om softwaren fra brugerne gennem interviews, *benchmarks* (ydelsestest) eller designinterne rapporter. Alfa-, beta- og funktionelle tests foregår på den anden side af designet så at sige, hvor primære udviklingsfaser som konceptudvikling, systemfilosofi, arkitektur, osv. er overstået. Vi anslår derimod implementering af brugerinformation i konceptudviklingen på et tidligt stadie af udviklingen. Der behøver ikke foreligge en funktionel prototype for at diskutere koncepter eller grundsten til en teoretisk arkitektur med potentielle brugere. Informationen sætter automatisk iterationer i gang på et tidligt stadie, hvor de primære designkoncepter stadig kan forstyrres. Netop fordi tendenser flytter sig, må de primære designkoncepter flytte sig i henhold til aftagere, og ikke til organisatoriske rutiner. Med andre ord en tidligere inddragelse af en større mængde designrelevant information, hvilket kun tjener som et eksempel på hvordan organisationer etablerer et forhold til den reelle omverdenskompleksitet tidligt i designprocessen.

Ser vi vores metode i et organisatorisk perspektiv, kræver den derfor en åbning mod andre kriterier end de, der umiddelbart understøtter en navigation mellem forskellige designvalg. Denne åbning kommer ikke blot i stand som observation, men også 'forstyrrelse' af den omkringliggende organisation gennem formidling af tematikker, perspektiver og ideer med relevans for alle de organisatorisk involverede parter.

Revidering af metoden

Ud fra de ovenstående refleksioner kan vi umiddelbart pege på nogle punkter for revidering af metoden. For det første en pragmatisk revidering ud fra vores skelnen mellem åbning og lukning af designprocessen, hvor vi i konstruktionsfasen observerer en intern lukning, der åbnes og forløses i testen. Lukningen bliver dermed en nødvendighed for at føre *requirements* over i en konkret form, der ikke konstant ændrer informationsmæssig basis. Men lukningen medfører en iagttaget risiko for lukning på de forkerte tidspunkter – og metoden giver ikke et bud på, hvordan vi kan skelne mellem de rigtige og de forkerte tidspunkter for lukning. Derfor må vi i videre iterationer i af

metoden indarbejde kriterier for en sådan skelnen, der kan fastholde den informationsmæssige basis for design, og samtidig give konstruktionsfasen et holdepunkt for konkret design.

Konklusion

Vi har forsøgt at blottlægge en vej fra viden til relevant konstruktion indenfor software, der retter sig mod den diffuse proces, det er at lave musik. Vores konstruktion er én vej af mange mulige, hvilket umiddelbart kræver et svar på, hvorfor vi har valgt denne vej og ikke andre. Videre beskriver konklusionen værdien af hhv. den prototype, vi har designet, og den metode, vi har designet ud fra.

Struktur og proces

Vores design får en retning og finit form gennem tematisk specificering med afsæt i en mere generel metodisk baggrund. Vi har behandlet denne proces som fiksering, hvor information implementeres i et designsystem, der iterativt arbejder mod en form. Vi ser denne bevægelse mod fiksering som en vej fra et generelt til et specifikt niveau, hvilket er en leddeling, der i specialet ses som forskellen på *Metode-* og *Design-*kapitlerne. *Metode* ser vi som et vidensgrundlag, der fokuseres i *Design*, formgives i *Prototype*-kapitlet og evalueres i *Evaluering*.

Information

Bevægelsen i specialet går over informationsimplementering, der er relateret til alle specialets processer. Vi arbejder mod at indsamle så meget designrelevant information som muligt, hvilket begrundes i ønsket om en mætning. Et mætningspunkt, hvorfra vi iht. interne designkriterier udvælger information mhp. en form. Dette medfører en orientering mod såvel omverden som egne kriterier, der løbende forstyrres af information, der implementeres. Informationen er implementeret i alle specialets områder, herunder æstetik og funktionalitet, empiri og teori, refleksion og konstruktion.

Form

På et metodisk såvel som et konstruerende niveau har vi identificeret nøglebegreber og mekanismer i forhold til en læsning af George Spencer Browns formteori. Iterativ udvikling af en finit form ser vi som *re-entry* i konkret materiale. Videre resulterer denne formforstyrrelse i information. Vi anvender denne information i teoretisk og metodisk katalysering af en ny form. På et mere filosofisk plan arbejder vi med formsondringer, oscilleringer og *imaginary states* i vores egen kreative proces, der kan udlægges som kontinuerlig *re-entry*. Formteorien indfanger på dette punkt en kaotisk proces i præcise begreber, der tillader os at tænke generelt om en specifik proces. Differentiering mellem f.eks. specifikke og generelle iagttagelser skaber konstruktive forskelle, hvis optikken (metodisk og teoretisk) er i stand til at observere kaotisk mangfoldighed.

Observationen af tendenser i omverdenen fordrer en evne til at sondre mellem et mikro- og et makroniveau og registrere sondringen i hukommelsen. En orden blandt musikere observeres ud fra deres individuelle, kontingente mønstre, hvilket skaber *order from noise*.

Regulativer

Ud fra information, der opstår ud fra formsondringer på alle niveauer i specialet, konstruerer vi metodiske såvel som konkrete regulativer.

De *metodiske regulativer* styrer kriterierne for videre informationsindoptag i designsystemet. Et vigtigt metodisk regulativ knytter sig til åbningen og lukningen af designprocessen. Vi har ikke diskuteret et kontekstafhængigt kriterie for lukning, men ser argumentet for lukning som processpecifikt. Med base i information kan det internt i systemet vurderes om mængden af designrelevant information er tilstede for at syntetisere en form. Dette er vores operationelle lukningskriterie for prototypen; der findes derimod ikke noget lukningskriterie for designprocessen, der fortsætter iterativt iht. indoptaget information. Vi har etableret et åbningskriterie for prototypen, som indtræder, når viden og teknologi understøtter konceptuelle perspektiver, der vurderes som tilstrækkelig information til at igangsætte næste iteration.

Vi syntetiserer en form ud fra de *konkrete regulativer*, hvilket vi har diskuteret som en fiksering og en stigende specialisering af processer. Et regulativ er en argumentation for aktualiseringer, hvilket i praksis harmonerer processer, beslutninger, former, mv., med hinanden. De konkrete regulativer falder i æstetiske og funktionelle kategorier, og optræder som udvidelser og begrænsninger af designvalg. Summen af udvidelser og begrænsninger ser vi som en finit form. Summen kan samtidig ses som en sfære, der knytter sondringspoler til et samlet hele. Indenfor denne sfære står specialets elementer i strukturer i overensstemmelse med regulativer.

Der knytter sig designinterne vurderinger til hvert element iht. kriterier som innovation og adaption, hvilke er kriterier som gendrives empirisk. Den innovative værdi ser vi i forhold til interaktion, mens den adaptive værdi ligger i høj kompatibilitet mellem komponenter. Gendrivelse af innovativ og adaptiv værdi resulterer i information om muligheder for fremtidig udvikling ifølge empirien. Disse muligheder handler både om forbedringer af det eksisterende og muligheder for det ikke-eksisterende. Disse udsagn om muligheder er det informationspotentiale, der er i en prototype.

Prototypens værdi

Prototypen har som nævnt levet op til vores forventninger om, at den ville fungere som kilde til videre inspiration til innovative gentænkninger af designet. Dens nuværende innovative/adaptive værdi udtrykker hele den sfære af elementer, der ligger bag dens form. I henhold til testen og

respons på onlinefora¹²⁰ er designet vurderet som relevant og interessant. Koblingen af et anderledes visuelt og taktilt interface med mulighederne for *glitch*, *scriptzation*, *serialisation*, simulering af naturlige egenskaber, delvis autonomi, og kontrol med flere simultane parametre vurderes positivt. Prototypens form afspejler imidlertid også et blik for dens kontekst, der er udtrykt gennem design mod arbejdsmønstre og æstetisk værdi, visuelt såvel som interaktionsmæssigt. Formen repræsenterer et instrument med potentiale for æstetisk relevant modificering, og med en tilpas grad af transparens og konsistens. Dette vækker ligeledes interesse, der understreger at prototypen har relevans i andre setups og arbejdsgange, hvilket også er bekræftet gennem *Evaluering og involveringen af musikere på onlinefora*.

Metodens værdi

Et perspektiv i vores udviklingsmetode er en varig åbenhed for forstyrrelser udefra, der kobles med interne kriterier for styring af processen. Koblingen af formteori med tanker fra teorier om systemudvikling leder til en metodeformulering og designproces, der har vist sig at understøtte dette perspektiv med positive resultater. Resultaterne er såvel prototypen i dens nuværende form, som de ideer til fremtidig innovation, den har ført med sig. Da designprocessen imidlertid kun repræsenterer én *case* og ikke et grundlag for generalisering af metodiske pointer, vil vi i stedet beskrive metodens værdi som udgangspunkt for videre diskussion. Bl.a. vil det være relevant at videreføre diskussionen om metodens mulige relevans i markedsorganisationer, såvel som yderligere koblinger i dens teoretiske grundlag.

¹²⁰ Se bilag

Litteraturliste

Alle online-ressourcer er sidst tjekket for tilgængelighed ved specialets afslutning.

[Booch, Jacobson & Rumbaugh, 1999]

Booch, Grady; Jacobson, Ivar; Rumbaugh, James: *The Unified Software Development Process*. Addison-Wesley, 1999

[Bolt-Rasmussen, 2003]

Bolt-Rasmussen, Mikkel: Kunst, æstetik og politik. In: *Fra verden til navlen – kunsten i funktion*. Forlaget Philosophia, 2003

[Borch, 2000]

Borch, Christian: *Former der kommer i form*. Tidsskriftet Distinktion, årgang 2000, hæfte 1

[Bødker, Kensing, Simonsen, 2000]

Bødker, Keld; Kensing, Finn; Simonsen, Jesper: *Professionel IT-forundersøgelse*. Samfundslitteratur, 2000

[Carroll, 2000]

Carroll, John M.: *Making Use. Scenario-Based Design of Human-Computer Interactions*. Massachusetts Institute of Technology, 2000

[Chadabe, 1997]

Chadabe, Joel: *Electric Sound. The Past and Promise of Electronic Music*. Prentice-Hall, 1997

[Chu, 1999]

Chu, Lonny L.: *MusiCloth: A Design Methodology for the Development of a Performance Interface*. CCRMA, Department of Music, Stanford University, 1999.

Artiklen findes på cd-rom'en som filen chu99.pdf

[Cook, 2001]

Cook, Perry: Principles for Designing Computer Music Controllers. In: *Proceedings of the 2001 Conference on New Interfaces for Musical Expression (NIME-01)*. Seattle, USA, 2001.

Tilgængelig online på: <http://www.suac.net/NIME/NIME01/cook.pdf>

[Elsea, u.å.]

Elsea, Peter: *Fuzzy Primer*

Artiklen findes på cd-rom'en som Fuzzy-primer.pdf

[Elsea, 1995]

Elsea, Peter: *Fuzzy Logic and Musical Decisions*. University of California, Santa Cruz, 1995

Artiklen findes på cd-rom'en som Fuzzy_Logic_And_Music.pdf

[Floyd, 1984]

Floyd, Christiane: A Systematic Look at Prototyping. In: Budde, R.; Kuhlenkamp, K.; Mathiassen, L.; Zullighoven, H. (eds.): *Approaches to Prototyping*. Springer-Verlag, 1984.

[Gaver, m.fl., 1999]

Gaver, Bill; Dunne, Tony; Pacenti, Elena: Cultural Probes. In: *Interactions: January + February 1999*. ACM, 1999.

[Goodwin, 1992]

Goodwin, Michael Mark: *Adaptive Signal Models: Theory, Algorithms, and Audio Applications*. Ph.D-afhandling, University of California, Berkeley.

Afhandlingen findes på cd-rom'en som filen Adaptive.Signal.Models.Theory,.Algorithms.And.Audio.Applications.pdf og online på <http://ptolemy.eecs.berkeley.edu/papers/97/mgoodwinThesis/>

[Hamman, 2000]

Hamman, Michael: *Priming computer assisted music composition through design of human/computer interaction*. AMTA, 2000.

Artiklen findes på cd-rom'en som filen amta_2000.pdf

[Hamman, 2002]

Hamman, Michael: From technical to technological: The imperative of technology in experimental music composition. In: *Perspectives of New Music; Winter 2002*; 40, 1 (pg. 92). Academic Research Library, 2002.

Artiklen findes på cd-rom'en som filen technical-technological.pdf

[Hansen, 2003]

Hansen, Ejvind: Kunstens abstraherende karakter. Om at kvalificere og genindføre diskussionen af kunst. In: *Fra verden til navlen – kunsten i funktion*. Forlaget Philosophia, 2003

[Lessig, 2004]

Lessig, Lawrence: *Free Culture. How Big Media Uses Technology And The Law To Lock Down Culture And Control Creativity*. Penguin Press, 2004

Bogen findes på cd-rom'en som filen freeculture.pdf

[Luhmann, 1984]

Luhmann, Niklas: *Sociale systemer. Grundrids til en almen teori*. Dansk udgave ved Hans Reitzels Forlag, A/S, København, 2000

[Mathiassen, m.fl., 1997]

Mathiassen, Lars; Munk-Madsen, Andreas; Nielsen, Peter Axel; Stage, Jan: *Objektorienteret analyse og design*. Forlaget Marko, 1997

[Mulder, 2000]

Mulder, Axel G.E.: Towards a choice of gestural constraints for instrumental performers. In: Marcelo Wanderley & Marc Battier (eds.): *Trends in Gestural Control of Music*. IRCAM, Paris, France, 2000

[Kalthoff & Nonaka, 1997]

Kalthoff, Otto; Nonaka, Ikujiro: *The light and the shadow*. Roland Berger Foundation, 1997

[Kanach, 2002]

Kanach, Sharon: Xenakis's hand, or the visualization of the creative process. In: *Perspectives of New Music; Winter 2002*; 40, 1 (pg. 190). Academic Research Library, 2002

Artiklen findes på cd-rom'en som visualization-creative-process.pdf.

[Schön, 1992]

Schön, Donald A.: "Designing as Reflective Conversation with the Materials of a Design Situation". *Knowledge-Based Systems, vol. 5*, 1992 (pp. 3-14)

[Serafin & Young, 2003]

Serafin, Stefania; Young, Diana: Playability Evaluation of a Virtual Bowed String Instrument. In: *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*. Montreal, Canada, 2003

Tilgængelig online på:
http://www.music.mcgill.ca/musictech/nime/onlineproceedings/Papers/NIME03_YoungSerafin.pdf

[Spencer Brown, 1969]

Spencer Brown, George: *Laws of Form*. Juliann Press, 1969

[Sterken, 2001]

Sterken, Sven: Toward a space-time art: Iannis Xenakis's polytopes. In: *Perspectives of New Music; Summer 2001*; 39, 2 (pg. 262). Academic Research Library, 2001

Artiklen findes på cd-rom'en som filen space-time-art.pdf

[Williams, 1997]

Williams, Robin: *The Social Shaping of Information and Communication Technologies*. Research Centre for Social Sciences, University of Edinburgh, 1997

Ansvarsliste

Mads Lindgren

Resumé - Det designmæssige problem – Information – Iterationer - Iteration, information og designsystem - Innovation og adaption - Priming, kreativitet og inspiration - Empiri – metode - Empiri – domæneindkredsning – Systemfilosofi - Æstetikken i og omkring prototypen - Funktionalitet og modificering - Fuzzy Logic – Arkitektur – Konstruktion - Formmæssig fiksering af prototypen - Empirisk fiksering af interaktionen - Teoretisk fiksering af interaktionen - Teknologisk fiksering af interaktionen - Empirisk fiksering af æstetikken - Teoretisk fiksering af æstetikken - Teknologisk fiksering af æstetikken - Procesbeskrivelse

Mads Weitling

Indledning - Formulering af designmetoden - Konkrete aktiviteter i de fire arbejdsfaser - Stringente og ikke-stringente designprocesser - Softwareinstrumenter - Frameworks og dialektikker - Scenarier – skitsering af interaktionsformer - Use cases – interaktionsformernes forankring i funktionalitet - Test af prototypen – Testpersonerne – Testforløbet - Modsvær til behov og ideer til formændringer - Udvidelse af designet – nye perspektiver - Evalueringen af prototypen og testen - Evaluering af metoden

Fælles

Det metodiske problem - Metoden i organisatorisk perspektiv - Konklusion - Alle opsamlinger - Ansvarsliste - Litteraturliste

Bilag

Ordliste

Constraints: Designere begrænser muligheder i forhold til f.eks. hvad der er muligt for en bruger i interfacet, eller i koden bag interfacet. En *constraint* kan imødegå problemer, f.eks. er det u hensigtsmæssigt at en bruger kan slette sin harddisk med vores musikinterface – hvilket kan forhindres med en begrænsning. Grafiske *constraints* har vi indbygget som f.eks. begrænsede kameravinkler, grænser for hvilke objekter, der kan manipuleres, mv.

Controller (1): Betegner en *mapping* fra interface til parametre via MIDI-standarden. F.eks. kan MIDI-controller 34 være knyttet til rumklangs-volumen af brugeren.

Controller (2): Hardware, der muliggør interaktion med lydparametre gennem f.eks. drejeknapper, der *mappes* til en lydkilde. Eksempelvis en såkaldt *knob box* eller knapper på en synthesizer, der kan *mappes* til software.

Crash: Computeren låses i kraft af at processer og hukommelse er overbelastede. Et crash kommer ofte i kølvandet på eksekvering af kode med semantikfejl.

Cutoff: Afskæringsfrekvens. Betegner den frekvens, hvor et filter udelukker enten højere eller lavere frekvenser i et signal.

DAW: Digital Audio Workstation. – Computer; PC eller MAC. *DAW* indbefatter også nyere sequencer-hardware, der kan afvikle VST-plugins, eller andre moduler.

Game Engine: En softwarekomponent, der varetager sammenkoblingen af grafik og interaktion, og genererer et interaktivt forløb. *Gameplay* beskriver hvordan et narratologisk forløb bliver indbygget i denne brugerflade.

GUI: Graphical User Interface. Dækker over brugerflader, der betjener sig af grafiske elementer.

Host eller *shell:* Et afviklingsmiljø, hvor komponenter som VST-plugins eksekveres og MIDI-data redigeres.

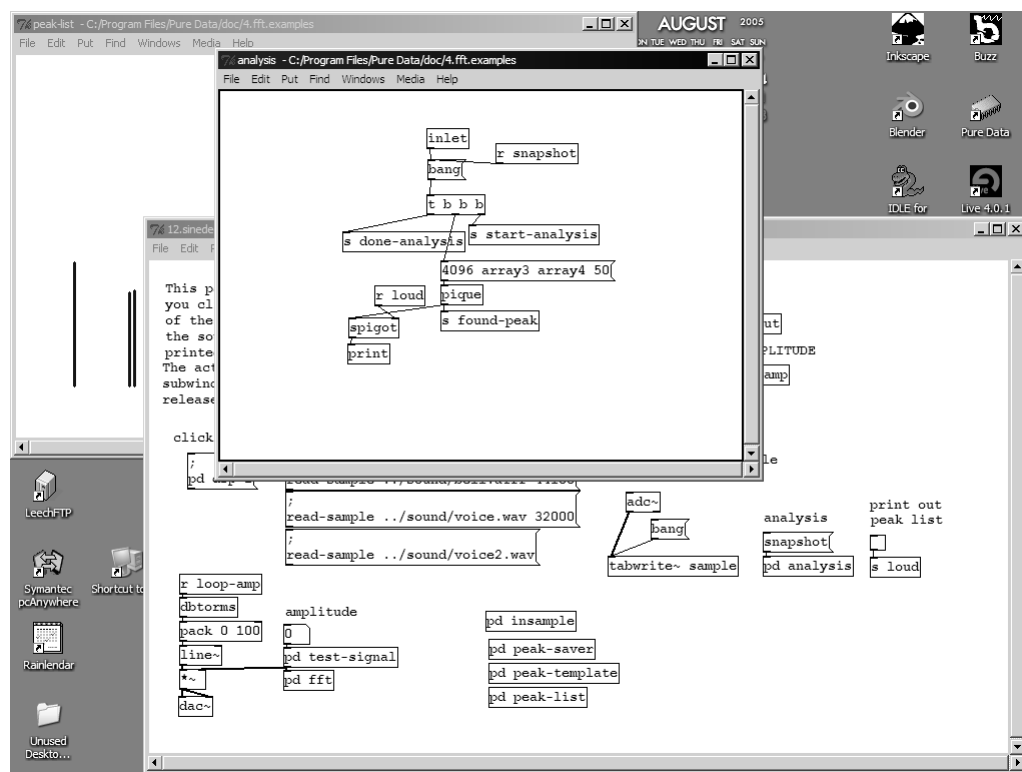
IP-netværk: Et netværk baseret på IP-protokollen, f.eks. Internet eller lokalnetværk.

Mapping: Dækker over relationen mellem input og output, f.eks. fra tangentanslag eller andre brugerinputs til specifikke lyde eller lyd-ændringer. Det er altså en relation mellem to domæner, her et interaktionsdomæne og et lyddomæne.

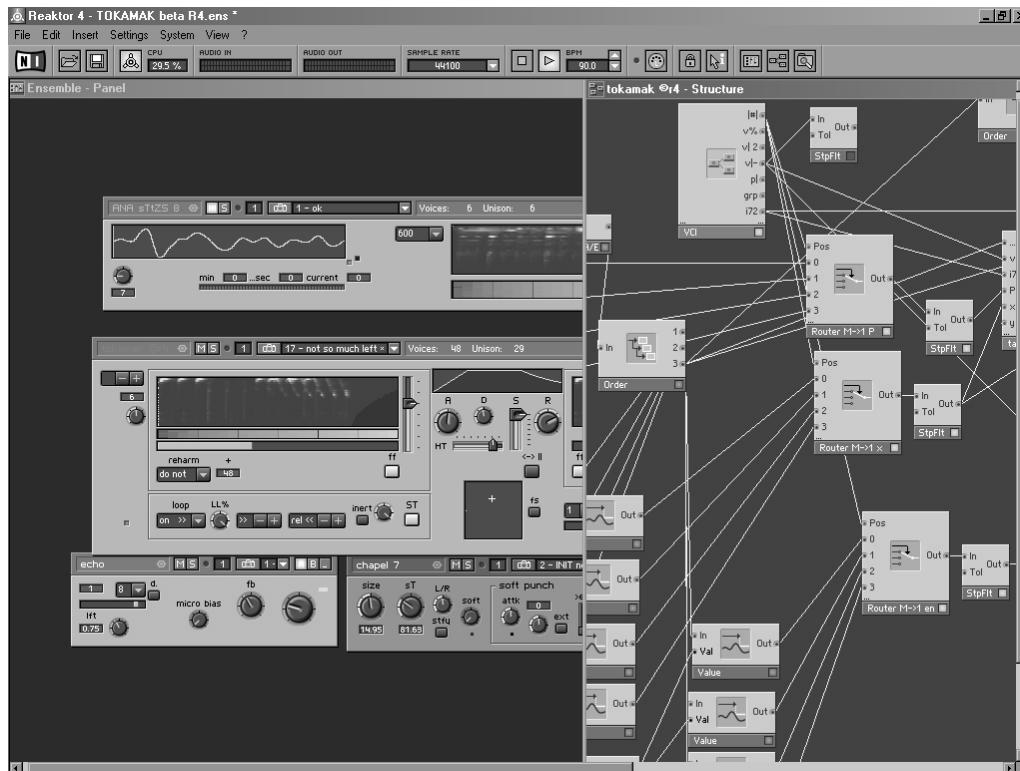
OpenGL: En åben standard for implementering af 3D-grafik på computere. *Blender* benytter OpenGL.

OSC: Open Sound Control. En ny standard for overførsel af kontrol-data mellem digitale lyd-kilder.

Pure Data eller *PD*: Et modulært *shell*-miljø, der er videreudviklet fra *Max/MSP*.



Reaktor: Et modulært miljø for lydgenerering og –behandling, udviklet af *Native Instruments*.

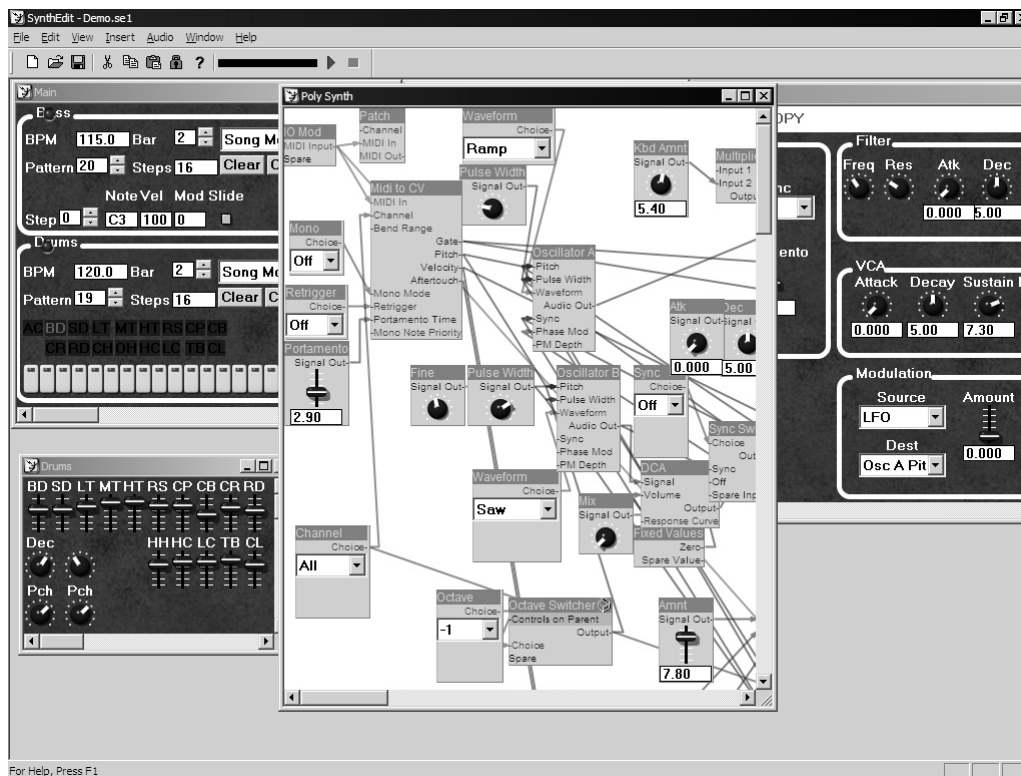


Resonans: En grad af forstærkning af et lydsignals volumen omkring *cutoff*-frekvensen i et filter.

Samples: Lydoptagelser af varierende længde, fra få millisekunder til hele fraser eller mere. Kan f.eks. bruges som fragmenter, der samles til en helhed.

Session: Et kortere ustruktureret forløb. Instrumenter prøves af eller der improviseres.

SynthEdit: Et modulært miljø for lydgenerering og –behandling. Programmet er gratis og findes på <http://www.synthedit.com>



VST-plugin: En standard for *plugins*, dvs. software-komponenter, der kan generere lyd eller MIDI-data, og som kan integreres i en *host*.

Test - interviewsvar

Bo Nicolaisen

Kan du beskrive nogen mønstre i dit arbejde med musik?

Jeg har en ret fast rutine, jeg bruger når jeg komponerer, starter med et simpelt loop med et instrument eller rytmespor, og bygger så dette loop op lidt lissom en lagkage, med flere instrumenter, melodi linjer og efx... derefter lader jeg loopet ligge et stykke tid, og vender så tilbage, og griner lidt og arbejder måske videre med det og kombinerer det evt. med andre loopz.

Savner du muligheden for at åbne din komposition/performance for tilfældigheder - hvordan sker dette i praksis?

Jeg savner muligheden for at skabe kaos og så vende tilbage til kompositionen, jeg er lidt kontrolfreak, og måske også lidt for usikker endnu til at gøre det live uden sikkerhedsnet. Så et interface med controllere hvor man kan gemme et sæt presets man kan klikke på ville være kanon cool. Så kan man flippe ud som man har lyst eller bare køre på rutinen ;)

Hvad er dine forventninger til arbejdsprocesser indenfor 3D domæner – hvad giver visuelt mening?

Et interface hvor input hænger sammen med output på en simpel og gennemskuelig måde, så man umiddelbart kan fatte hvad man hører i, vha. af auditiv og visuelt feedback.

Hvad er dine forventninger til det gode setup - hvad er ergonomisk og brugbart software for dig? Bruger du hardware MIDI-controllere af forskellig slags, og i så fald, til hvad?

Det gode setup er for mig er fleksibelt og "customizable", de kontroller man bruger skal kunne "lægges øverst i input stakken" og gerne kunne sætte op til en hardware controller. Jeg er blevet stadig mere opmærksom på publikums visuelle oplevelse og opfattelse af ens sceneoptræden, det er for mig vigtigt at komme ud over "de spiller bare cd'ere og læser email deroppe", at publikum kan se at man spiller.

Spiller du 3D-spil? Giver dette dig nogle præferencer for 'det gode interface'?

Har spillet en del og udvikler selv interfaces, hvilket gør mig ret kritisk i forhold til "skæve" eller komplicerede interfaces, intuitivt informations flow er meget vigtigt for mig, også fordi jeg, sjovt nok, er ret "teknologi blind" og har svært ved at finde rundt i crappy interfaces.. og derfor hurtigt taber interessen... attentionspan som en guppy :)

Hvordan indretter du dit software-studie efter dine egne krav og forventninger? Hvad gør du for at tilpasse eksisterende miljøer som f.eks. Cubase, Live eller Reason?

Mit hoved setup består af live med reason i rewired, derudover har jeg en midi controller med en fader og 15 drejknapper og et keyboard.. desuden bruger jeg mikrofoner, plastik trompeter, melodica og alt muligt crap til enten at spille på eller generere samples med. Desuden kan jeg godt lide at fiske med forskellige plugins og rode lidt med lidt lydprogrammering.

Hvad er dine forventninger til fremtidige udviklinger indenfor musiksoftware? Hvilke tendenser ser du, og hvordan ser din egen fremtid ud?

Mine forventninger er at instrumenter og software smelter mere sammen, og at der efterhånden vil opstå en platforms uafhængig standard med mere modulariseret software hvor man kan "købe" de funktioner man har brug for.. osv.. min største forventning er til udviklingen indenfor man-machine interaktion, så man kan flytte sin performance lidt væk fra den rundryggede stance bag "låget".

Min egen fremtid ser jeg halse lidt efter pga økonomi :)... men også gerne at udvikle mig som musiker i retning af mere samspil med andre og mere styrke i improvisation ☺

Sune Petersen

Kan du beskrive nogen mønstre i dit arbejde med musik?

Tja, jeg ved ikke rigtigt hvilke mønstre der er, men der er sikkert nogle.

Savner du muligheden for at åbne din komposition/performance for tilfældigheder - hvordan sker dette i praksis?

Ja, jeg vil gerne have flere muligheder for tilfældigheder i min musik. Jeg bruger lives random funktion, men den kan snildt blive mere avanceret, det ville være fedt at kunne styre flere parametre med tilfældigheder.

Hvad er dine forventninger til arbejdsprocesser indenfor 3D domæner – hvad giver visuelt mening?

Jeg har nok ikke nogen. Jeg er lidt skeptisk, jeg har svært ved at se hvordan det skal give bedre muligheder.

Hvad er dine forventninger til det gode setup - hvad er ergonomisk og brugbart software for dig? Bruger du hardware MIDI-controllere af forskellig slags, og i så fald, til hvad?

Jeg bruger ikke hardware midicontrollere, jeg har en, men jeg har ikke noget midi interface. Det er en rigtig god grænseflade, da man for det første kan arbejde med flere parametre ad gangen.

Spiller du 3D-spil? Giver dette dig nogle præferencer for 'det gode interface'?
Nej, jeg spiller ikke 3D spil.

Hvordan indretter du dit software-studie efter dine egne krav og forventninger? Hvad gør du for at tilpasse eksisterende miljøer som f.eks. Cubase, Live eller Reason?
Det gør jeg vel egentlig ikke.

Hvad er dine forventninger til fremtidige udviklinger indenfor musiksoftware? Hvilke tendenser ser du, og hvordan ser din egen fremtid ud?

Mere live sequencing, mere avancerede routingmuligheder, evt. en kombination af reaktor og live. min egen fremtid... tja hvem der bare vidste det.

Rasmus Allin

Kan du beskrive nogen mønstre i dit arbejde med musik?

I studiet arbejder jeg egentlig meget traditionelt, dvs får en musikalsk eller lydmæssig ide og prøver den af. Jeg bruger Logic.

Savner du muligheden for at åbne din komposition/performance for tilfældigheder - hvordan sker dette i praksis?

I Logic får man jo ikke rigtig andet igen end det man selv putter ind, og det er jo faktisk ikke særlig inspirerende. Det kunne klart være fedt med nogle tilfældigheds-inspirations-maskiner. I mit live PD-setup har jeg jo lavet PflumScrampleren der jo mere eller mindre tilfældigt kan kaste rundt med et beat eller sequence. Dette er jo en stor hjælp live for at undgå ensformighed, men også som inspiration til komposition.

Hvad er dine forventninger til arbejdsprocesser indenfor 3D domæner – hvad giver visuelt mening?

Ideen med 3D er vel først og fremmest at gøre det hele mere intuitivt, samt selvfølgelig at kunne tweeke flere (3) parameter på en gang. Der er jo et problem deri at musen jo er 2-dimensionel. Det ideelle ville klart være et fysisk 3D-interface.

Hvad er dine forventninger til det gode setup - hvad er ergonomisk og brugbart software for dig? Bruger du hardware MIDI-controllere af forskellig slags, og i så fald, til hvad?

Live hader jeg musen, der er langsom og upræcis+at den kan alt for få ting ad gangen. I mit PD-setup bruger jeg et midikeyboard med 8 contollere. Med keyboardets tangenter vælger man så hvilke grupper af parameter der tweekes med controllerne. Således er man helt uafhængig af musen, og med et tryk er man videre til næste gruppe parameter. Det er væsentlig hurtigere. Controllerne styrer filterparametre, effektsends, effektparametre og diverse parameter for glitches i PflumScrampleren.

Spiller du 3D-spil? Giver dette dig nogle præferencer for 'det gode interface'?
Nej

Hvordan indretter du dit software-studie efter dine egne krav og forventninger? Hvad gør du for at tilpasse eksisterende miljøer som f.eks. Cubase, Live eller Reason?

Jeg ville gerne kunne bruge VST-instrumenter til PD-setuppet [Artillery], hvilket pt. ikke er muligt. Derudover ville jeg gerne kunne bruge setuppet som "inspirations-maskine" i Logic i studiesammenhæng, men det går jo heller ikke, fuck fuck fuck fuckpislort.

Hvad er dine forventninger til fremtidige udviklinger indenfor musiksoftware? Hvilke tendenser ser du, og hvordan ser din egen fremtid ud?

I fremtiden vil der blive lavet en fysisk 3D controller af modellervoks, så man med hænderne fysisk former musikken. Når man så afspiller musikken vil modellervoks klumpen bevæge sig (måske). Ellers kunne man vel uden særligt besvær lave en 3D-theremin, der sender midital i stedet for lyd. Det fede ville være at fremtidens software var meget let tilgængeligt, men at alle presets og opsætninger skal kunne ændres, så kunne totalt specialiseres. Selv regner jeg med at lave et stort hit og så købe stoffer for alle pengene.

Test – noter taget undervejs

Bo Nicolaisen

splines i tracker (udglatning af overgangen mellem pkt'er)

trackeren er god, især som performance og visualisering af laptopkoncerter.

trackeren ser godt ud og vil fungere rigtig godt m 'vælg-nærmeste' fremfor 'vælg-næste' – stort irritationsmoment for alle 3 deltagere.

egentlig ok at den er løsrevet fra musikkens tempo, men temposync eller tapping ville også være cool.

presets, morph mellem presets.

evt skins eller alternative scener m samme funktionalitet

pong scenen er mindre brugbar i sin nuværende form, ikke for svært at se en sammenhæng ml lyd og billede, men for svær at styre/for ukontrollerbar

midi breakout, pong eller noget ala infinitewheel.com .. en mellemting mellem instrument og et spil, dog mere værktøj end spil i vores tilfælde

hvert pkt i trackeren kan også styre et separat sæt controllere, dvs. $5 \times 3 + 3 = 18$ ctrl 'gemt' i interfacet

Sune Petersen

random walk i tracker (mellem hvert pkt), mindre glatte kurver (modsat splines)

spinneren som lfo og med træghed/inerti

mere kontrol over tilfældigheder, fx justering af rummet for midicontrollere (fra 0-127 til 85-115), gemme sådanne indstillinger

mulighed for respons på midiinput: styring udefra af objekter, retrigs af sekvenser/presets (program change).

flere kasser/miljøer i samme scene eller flere views på samme scene

klart et performance/visualiseringsværktøj

vst integration, integration m modulære miljøer.

Rasmus Allin

godt at have kontrol m mere end 2 parametre, godt at kunne slippe interfacet og stadig få forandringer ud af det.

kunne godt tænke sig muligheden for at slå mange parametre til på én gang (dynamik, 'bang') .. og en rytmisk kobling, fx kun events på 16-dele eller 8-dele (en midigate).

basale sequencerfunktioner, presets i trackeren, morph mellem presets, direkte opsætning af midicontrollernumre i interfacet (on the fly, evt direkte valg af midikanal.

hellere mere kontrol med flere samtidige parametre end indbygning af tilfældigheder.

controller mute/unmute vil være godt, evt som keyboard commands

Diskussion af fremtidige perspektiver

integration m hardwareinterfaces som midicontrollere, keyboards vil være cool .. musen er et dårligt instrument (rasmus)

integration m fx touchscreens (polyfonisk), 3d mus, joypads, multiple skærme (mange laptops kan drive to skærme m separat indh. – sequencer på den ene og 3d interface på den anden), motion tracking

behovet for andre inputdevices hænger osse sammen m behovet for en mere ekspressiv performance – sune nævner monolog og håkan lidbo m hhv guitar og hardware-sequencere, vocodere, etc.

klart laptopnørder som målgruppe, men også et potentiale for mere tæt integration m specifikke lydværktøjer, fx en bestemt filterkode eller synth kan være givende for musikere m mindre lyst til at eksperimentere

en anden målgrp kan være publikum for visuals/koncerter m mere ekspressiv performance
experimentarium setups

Respons på onlinefora og mail

Vi har sendt prototypen ud til en lang liste testere pr. mail. Responsen har været overvejende god. Tematisk kan testernes svar placeres i samme rubrikker som interviewpersonerne i afsnittet *Evaluering*. På grund af overlap inddrager vi ikke mails fra vores testere. Dog skal en mail fra Joel Brindefalk lige indsættes som et eksempel på et optimistisk input, der er rigt på perspektiver (og svensk).

```
Hallå! Tack för senast. Det var fan fedt. Kool skit ni håller på med/laever aller vafan ni säger. I like. Ska testa senare. Är fedt intresserad av just sån skit. Mand. Har försökt få folk/dev's intresserade av en vst eller liknande som hanterar många controllers på en och samma gång, med möjlighet att ta "snapshots" av alla controllers värden/values och sedan "morpha" mellan dessa snapshots. S.c. "preset morphing", liknande Audiomulches metasurface om ni är bekanta med det eller GRM Tools, Pvst loader i buzz m.fl...vore fedt att ha som vst dock, med möjlighet att kunna assigna vilken cc-nrp/nrpn-control som helst i vilken vst-compatible host som helst, so to speak... Ok, love your work, keep it up...later.../Joel :D
```

Vi har anvendt fora med mange agendaer i dette speciale. Både som en empirisk cementering af indledende koncepter og som et samlingspunkt, hvor det er givende at evaluere højt specialiseret software. Brugere, der er involveret i onlinefora er ligeledes specifikke i deres interesser, og bevidste om hvilke fora, der er tematisk egnede for hvilke interesser. Vi mener det er vigtigt at have denne evaluering med i specialet for at vise, hvordan den lidt bredere omverden for designet reagerer. Vi har ikke fundet det nødvendigt at analysere og formidle vores indlæg/respons i disse fora, da det er ret klart hvad der tales om. Vi vurderer at de forskellige tråde kan forklare sig selv.

JackDark – Happy little hellhole [<http://jackdark.proboards39.com/>]

Forummet er trods titlen fokuseret på udvikling af musiksoftware. Forummet har været vores 'designcenter', da emnerne falder indenfor specialets og vores primære interesse. Her er et screenshot fra emneindekset, der viser hvilke overordnede temaer, der diskuteres.





Music Software:				
	News & Links - 2 Viewing The latest and greatest in the realm of freeware VSTs. <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	441	1,320	on Today at 10:57 by jiplan in New: Dual Trashers
	Software Discussion Let's talk about what we love and hate in the realm of music software. <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	34	288	on Yesterday at 10:05 by toits in Which host are you using?
	File Sharing Share your patches, MIDI's, ensembles, etc. <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	2	6	on 6/27/05 at 1:01 by Jack Dark in UNO FM Bank
	Developer's Spotlight Exclusive interviews with the best indie music software producers. <i>Moderator: Jack Dark</i>	11	35	on 7/20/05 at 0:19 by Jack Dark in Lime Light: Insert Plz...
	Best Of The Best The Very Best Freeware VSTs - Annually <i>Moderator: Jack Dark</i>	9	38	on 7/23/05 at 12:04 by natan in Top 5 Vocoders (2004)
	Member Download Section Tons of lost VSTs resurrected. Password required. See "Forum Concerns". <i>Moderator: Jack Dark</i>	25	209	
Music Software Programming:				
	Tips & Tricks & Techniques I'll show you mine if you show me yours. <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	5	53	on Yesterday at 18:23 by Juxta in a little tip for fl st...
	General Q & A Programming woes? Need a VST? MIDI not jiving? <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	20	163	on Yesterday at 14:20 by melinko in loops
	VST / Plug Creation - Programming All aspects of programming virtual instruments and effects. <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	2	15	on 7/28/05 at 17:59 by jaff in a good linky for sourc...
	VST / Plug Creation - GUI Production Discussions concerning all aspects of GUI production. <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	4	22	on 7/26/05 at 18:37 by jaff in Gui aspects of the pro...
	VST / Plug Creation - Brainstorm Got a great idea / theory for a VST? <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	14	75	on 7/28/05 at 21:30 by billfusion in Reverse reverb
Music Music Music:				
	Music By Others Let's talk about music by other people. <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	14	64	on Yesterday at 11:03 by Tourian in Your last musical purc...
	Music By Us Let's talk about our music! <i>Moderators: Jack Dark, DaveOjo, billfusion, ezeboogie, Scotobiotic, melinko</i>	21	116	on Yesterday at 14:18 by melinko in New Album Up For FREE
<i>Music By JD (http://www.jackdark.net/discography.html)</i>				


Vi har lavet to poster på JackDark forummet i hhv. *News & Links* og *GUI production*.

News & Links:




Happy Little Hellhole - Freeware VST Oasis :: Music Software :: News & Links :: 3 new 3D midi controller prototypes



search bookmark reply send topic to friend print



<p>Author</p> <p>jaff Hell Holer: Part-Time ★★★★ member is offline</p>  <p>Joined: Mar 2005 Gender: Male ♂ Posts: 47 Location: denmark Karma: 1 [Exalt Smite]</p>	<p>Topic: 3 new 3D midi controller prototypes (Read 108 times)</p> <p>3 new 3D midi controller prototypes « Thread Started on 7/23/05 at 21:20 » quote</p> <p>the reason why I haven't been around the board for a while: Containerturm</p>  <p>Phrasetrakker</p>  <p>and controllahtwistah</p> 
---	--

	<p>This is.. well. Playtime, fuck around with your midi arsenal in a playful way. Gravity algorithms are implemented to make some nice non linear controls - bouncing balls that send out midi. Phrasing your controllers is also possible. Maybe a bit of overview of your 12 favorites, pounding a bit of metal and tweaking at the same time, tempted? - download http://www.hum.aau.dk/~mlin00/kilog_prototype/setup.exe. Remember to install midi Yoke before the prototype http://www.hum.aau.dk/~mlin00/kilog_prototype/Midi%20yoke%20nt-2k-xp.exe All standalone prototypes are developed with Blender and Python for the midi scripting. All source code and scenes are included in the installer for your development pleassure. Keep the code open! Check out http://www.bitreactor.dk or my development buddy kilotons page http://www.kiloton.tk for more info.</p> <p>and please.. PLEASE post your comments, ideas or whatever.</p> <p style="text-align: right;">Logged</p>
<p>ezeeboogie Hell Holer: Vice CEO ★★★★★ member is offline</p>  <p>Joined: Feb 2005 Gender: Male ♂ Posts: 234 Karma: 6 [Exalt Smite]</p>	<p>Re: 3 new 3D midi controller prototypes « Reply #1 on 7/23/05 at 21:37 » quote</p> <p>Looks amazing! I feel sorry for anyone on dial up though... 😊</p> <p style="text-align: right;">Logged</p> <p>ezee like sunday morning... www.ezeeworld.com</p> <p>World on pause new world coming soon...</p>
<p>billfusion Hell Holer: Vice CEO ★★★★★ member is offline</p>	<p>Re: 3 new 3D midi controller prototypes « Reply #2 on 7/24/05 at 2:55 » quote</p> <p>7/23/05 at 21:37, ezeeboogie wrote:</p>

<p>Joined: Feb 2005 Gender: Male ♂ Posts: 263 Location: san diego Karma: 6 [Exalt Smite]</p>	<p>7/23/05 at 21:37, ezeeboogie wrote:</p> <div style="border: 1px solid black; padding: 5px;">  Looks amazing! I feel sorry for anyone on dial up though... 😊 </div> <h1 style="text-align: center;">NO SHIT</h1> <p style="text-align: center;">« Last Edit: 7/24/05 at 2:56 by billfusion »</p> <p style="text-align: right;">Logged</p>
<p>jaff Hell Holer: Part-Time ★★★★ member is offline</p>  <p>Joined: Mar 2005 Gender: Male ♂ Posts: 47 Location: denmark Karma: 1 [Exalt Smite]</p>	<p>Re: 3 new 3D midi controller prototypes « Reply #3 on 7/24/05 at 8:29 »</p> <p>thanks ezeeboogie, yeah well.. hard to come by any games less than 100 megs these days anyway. I'm sorry if this is such a problem - I would like to send you all CDR's if I had the money. But still - I worked my ass off on this project and I would hate to think network bandwidth is keeping ppl from using and testing it. Would like some comments on the game-midi interface thought.</p> <p style="text-align: right;">Logged</p>
<p>jaff Hell Holer: Part-Time ★★★★ member is offline</p> 	<p>Re: 3 new 3D midi controller prototypes « Reply #4 on 7/24/05 at 8:42 »</p> <p>working on some video demos - going online later. (approx 5MB each!!!)</p>

<p>ezeeboogie Hell Holer: Vice CEO ★★★★★ member is offline</p>  <p>Joined: Feb 2005 Gender: Male ♂ Posts: 234 Karma: 6 [Exalt Smite]</p>	<p>Re: 3 new 3D midi controller prototypes « Reply #5 on 7/24/05 at 9:38 »</p> <p>Cool. I don't really give a monkeys about download sizes, I'm on a 2Mbit DSL line... 🐒</p> <p>I still always think "I'm glad I'm not on dial up!" when I see files sizes over 20 megs though! 😊</p> <p style="text-align: right;">Logged</p> <hr/> <p>ezee like sunday morning... www.ezeeworld.com</p> <p>World on pause new world coming soon...</p>
<p>Dave0jo Hell Holer: Vice CEO ★★★★★ member is offline</p> <p>Linking bastard</p>  <p>Joined: Jan 2005 Gender: Male ♂ Posts: 318 Location: Ire Karma: 7 [Exalt Smite]</p>	<p>Re: 3 new 3D midi controller prototypes « Reply #6 on 7/24/05 at 11:01 »</p> <p>Wow, I can't wait to try these out!</p> <p>Good work. Very original. I can see these being awesome in a live performance with your monitor being digitally projected onto a big wall or screen.</p> <p style="text-align: right;">Logged</p> <hr/> <p>Fuck them if they can't take a joke: www.myfirstairport.com</p>
<p>jaff Hell Holer: Part-Time ★★★★ member is offline</p> 	<p>Re: 3 new 3D midi controller prototypes « Reply #7 on 7/24/05 at 11:23 »</p> <p>Hey thanks dave 🐒 especially for the originality compliment - those are hard to earn these days.</p> <p>The performance scenario was exactly what I had in mind.. ppl. have a tendency to think you are checking</p>

<p>member is online</p>  <p>Joined: Mar 2005 Gender: Male ♂ Posts: 47 Location: denmark Karma: 1 [Exalt Smite]</p>	<p>Hey thanks dave especially for the originality compliment - those are hard to earn these days.</p> <p>The performance scenario was exactly what I had in mind.. ppl. have a tendency to think you are checking your mail when your actually making music. A friend wrote a track about it; puzzleweasel - "do i look like a fucking jukebox". However - this is a prototype without compressed textures and super-duper optimized code, so it is quite memory hungry and maybe a bit dangerous live. Still I got my 1.7 ghz, 128MB mem - running fullscreen interface and reaktor 4-5 in the background without dropouts. So something can be done with it. 😊</p> <p>Uploaded some demo videos with reaktor presets for the still skeptic and not downloading crowd.</p> <p>http://www.hum.aau.dk/~mlin00/kilog_prototype/containerturn.reaktor.avi http://www.hum.aau.dk/~mlin00/kilog_prototype/phrasetrakker.reaktor.avi http://www.hum.aau.dk/~mlin00/kilog_prototype/controltahtwistah.reaktor.avi</p> <p style="text-align: right;">Logged</p>
<p>Jack Dark Hell Holer: CEO ★★★★★ member is offline</p> <p>The King of "EDIT".</p>  <p>Joined: Jan 2005 Gender: Male ♂ Posts: 1,463 Location: Within the tyrant's oily grip. Karma: 38 [Exalt Smite]</p>	<p>Re: 3 new 3D midi controller prototypes < Reply #8 on 7/24/05 at 19:33 ></p> <p>Very cool stuff in the design department... man I hate installing a VST though. Must we molest the registry?</p> <p style="text-align: right;">Logged</p> <p>All the king's horses, and all the king's men, couldn't keep a simple piece of foam from falling again.</p>
<p>jaff Hell Holer: Part-Time ★★★★ member is offline</p>	<p>Re: 3 new 3D midi controller prototypes < Reply #9 on 7/24/05 at 19:39 ></p> <p>thanks jack, really!.. And thanks for the pretty impressive stream of SE stuff lately.</p>


<p>[Exalt Smite]</p> <p>jaff Hell Holer: Part-Time ★★★★ member is offline</p>  <p>Joined: Mar 2005 Gender: Male ♂ Posts: 47 Location: denmark Karma: 1 [Exalt Smite]</p>	<p>All the king's horses, and all the king's men, couldn't keep a simple piece of foam from falling again.</p> <p>Re: 3 new 3D midi controller prototypes < Reply #9 on 7/24/05 at 19:39 ></p> <p>thanks jack, really!.. And thanks for the pretty impressive stream of SE stuff lately.</p> <p>There's no 'VST inside' 😊 just controllers for vst's or hosts.</p> <p>The installer doesn't edit your registry at all, it just extracts blender and python files to a designated dir along with a player. You can delete them without causing registry probs.</p> <p style="text-align: right;">Logged</p>
<p>jaff Hell Holer: Part-Time ★★★★ member is offline</p>  <p>Joined: Mar 2005 Gender: Male ♂ Posts: 47 Location: denmark Karma: 1 [Exalt Smite]</p>	<p>Re: 3 new 3D midi controller prototypes < Reply #10 on 7/26/05 at 8:42 ></p> <p>We became aware of a bug Posted on the KVR yesterday. The preferences GUI, where you assign MIDI ch, controller numers etc, doesn't call up your saved settings when you open it again. This is a graphical problem, because the channel and controllernumbers are still mapped - the sliders just don't display it.</p> <p style="text-align: right;">Logged</p>

GUI Production:

search / bookmark reply send topic to friend print

Author Topic: **Gui aspects of the prototypes (Read 28 times)**

jaff
Hell Holer: Part-Time
★★★★
member is offline



Joined: Mar 2005
Gender: Male ♂
Posts: 47
Location: denmark
Karma: 1
[Exalt | Smite]


Gui aspects of the prototypes
« Thread Started on 7/23/05 at 21:34 » quote

Just added a link to a couple of prototypes elsewhere...but maybe there is more room to get dirty with the gui aspects of the stuff. tjekkitout http://www.bitreactor.dk/_private/science.htm All done in blender (freeware proggy). Its nice for modelling, and with the midi gates open gui's are limited by yur imagination only. I know its no beauty.. but i could imagine Midi shooters, or just more data mining. Let me know what you think of the instrument/controller-game relation.

Logged


Jack Dark
Hell Holer: CEO
★★★★★
member is online

The King of "EDIT".



Joined: Jan 2005
Gender: Male ♂
Posts: 1,465
Location: Within the tyrant's oily grip.
Karma: 38
[Exalt | Smite]


Re: Gui aspects of the prototypes
« Reply #1 on 7/24/05 at 19:28 » quote



= Genius.


Jack Dark
Hell Holer: CEO
★★★★★
member is online

The King of "EDIT".



Joined: Jan 2005
Gender: Male ♂
Posts: 1,465
Location: Within the tyrant's oily grip.
Karma: 38
[Exalt | Smite]

Re: Gui aspects of the prototypes
« Reply #1 on 7/24/05 at 19:28 » quote




= Genius.

I like the way you think! Damn good GUIs and ideas.

Logged

All the king's horses, and all the king's men, couldn't keep a simple piece of foam from falling again.


jaff
Hell Holer: Part-Time
★★★★
member is offline



Joined: Mar 2005
Gender: Male ♂
Posts: 47

Re: Gui aspects of the prototypes
« Reply #2 on 7/24/05 at 20:03 » quote

wow, hey.. don't know what to say - flattered. Just tried out tiling some photos and mask them in photoshop. I seriously have no clue what I am doing, just ideas breeding new ideas with new tech in the hands. Blender (<http://www.blender.org/>) is a pretty scary program initially- steep learning curve, but nice when you get the hang of it. It could be good for gui details like logos, small video clips in the edges of af gui or nicely rendered knobs. The possibilities seems pretty vast, and it could be cool to do some more flattened gui's with it.
Its very easy to get good results fast - which would be proper for your development pace.

Joined: Mar 2005 Gender: Male ♂ Posts: 47 Location: denmark Karma: 1 [Exalt Smite]	Logged
natan Hell Holer: Part-Time ★★★ member is offline Joined: May 2005 Posts: 42 Location: Europa Karma: 0 [Exalt Smite]	Re: Gui aspects of the prototypes « Reply #3 on 7/26/05 at 14:48 » Nice program, this blender thing -def gonna have to digg deeper o this one... thanks! 😊 Logged
jaff Hell Holer: Part-Time ★★★ member is offline  Joined: Mar 2005 Gender: Male ♂ Posts: 47 Location: denmark Karma: 1 [Exalt Smite]	Re: Gui aspects of the prototypes « Reply #4 on 7/26/05 at 18:37 » you should natan. Its free and useful, I only played with the game engine but the renderer for stillshots or objects is totally killer. Good galleries on the blender.org site. Eyecandy majn. Logged

search / bookmark reply send topic to friend print

BuzzChurch [<http://www.buzzchurch.com>]

BuzzChurch er et forum for brugere af programmet Buzz. Forummet tiltrækker også mange græsrods-udviklere med tilknytning til miljøet omkring Buzz.

View previous topic :: View next topic	
Author	Message
kiloton Believer Joined: 17 May 2005 Posts: 10 Location: in the basement	Posted: Sun Jul 24, 2005 12:39 am Post subject: alternative software midi controllers in 3d <input type="button" value="Quote"/> <input type="button" value="Edit"/> hi everyone, for my last university project, I have been working on some software-based midi controllers that are basically 3d environments for sound modification. you can hook these up to buzz using the midiyoke virtual midi port, or hook one computer to another using any midi ports your sound card has. with buzz, I recommend using joachim's buzz knob assigner (http://voyager.adsl.dk/tools/buzz-knob-assigner.phtml) to make a map of all 128 midi controllers before using the software, as it spits out multiple midi cc's simultaneously, making it impossible to auto learn using buzz' midi preferences window. nevertheless, there's some fun to be had here, and i hope you want to test it. please tell us about your experience and any suggestions for further development you may have. there's a setup.exe plus pdf manuals and such files at: http://www.hum.aau.dk/~mlin00/kilog_prototype/ best wishes :=) kiloton & monolog
Back to top	<input type="button" value="Profile"/> <input type="button" value="Pers. Msg"/> <input type="button" value="Website"/> <input type="button" value="MSN"/>
kiloton Believer	Posted: Sun Jul 24, 2005 6:43 pm Post subject: We've added some small AVI films that show the software in action, just to give you an idea of what that slightly large

<p>kiloton Believer</p> <p>Joined: 17 May 2005 Posts: 10 Location: in the basement</p>	<p>Posted: Sun Jul 24, 2005 6:43 pm Post subject: Quote Edit</p> <p>We've added some small AVI films that show the software in action, just to give you an idea of what that slightly large setup.exe carries inside ☺</p> <p>The films are at:</p> <p>http://www.hum.aau.dk/~mlin00/kilog_prototype/</p> <p>(30 sec clips, DivX format)</p> <p>Any comments or suggestions are welcome ☺</p>
<p>Back to top</p>	<p>Profile Pers. Msg Website MSN</p>
<p>mute Monk</p>  <p>Joined: 13 Aug 2004 Posts: 371</p>	<p>Posted: Mon Jul 25, 2005 12:52 am Post subject: Quote</p> <p>i'll have to check this out when i get home, sounds interesting.</p> 
<p>Back to top</p>	<p>Profile Pers. Msg Website</p>
<p>usr Bishop</p> 	<p>Posted: Mon Jul 25, 2005 10:30 pm Post subject: Quote</p> <p>3d software and user input just don't match up.</p> <p>this is what i think. (<- sounds like an elekt sig)</p> <p>WHOEVER I LENT MY BLOK MODULAR TO /MSG ME ASAP</p>

 <p>Joined: 12 Aug 2004 Posts: 309 Location: erlangen, germany</p>	<p>this is what i think. (<- sounds like an elekt sig)</p> <p>WHOEVER I LENT MY BLOK MODULAR TO /MSG ME ASAP</p>
<p>Back to top</p>	<p>Profile Pers. Msg Website</p>
<p>elekt Choir Girl</p>  <p>Joined: 13 Aug 2004 Posts: 55</p>	<p>Posted: Tue Jul 26, 2005 12:05 am Post subject: Quote</p> 
<p>Back to top</p>	<p>Profile Pers. Msg Website</p>
<p>btd Deacon</p> <p>Joined: 13 Aug 2004 Posts: 135</p>	<p>Posted: Tue Jul 26, 2005 12:15 am Post subject: Quote</p> <p>usr wrote:</p> <p>3d software and user input just don't match up.</p>

btd Deacon	Posted: Tue Jul 26, 2005 12:15 am Post subject:	<input type="button" value="Quote"/>
Joined: 13 Aug 2004 Posts: 135 Location: York, UK	<p>usr wrote: 3d software and user input just don't match up.</p>	
<p>They <i>can</i>, but they seldom <i>do</i>. In fact I can't think of any examples of user-friendly 3D software which don't involve driving and/or shooting things (one possible exception: SketchUp, a surprisingly intuitive drawing/modelling package, but that is literally the only non-game I can think of which doesn't feel entirely like wading through treacle).</p>		
<p>Anyway, these controllers. They're very cool ideas, but they don't quite feel responsive and fluid enough (the exception being Controllahtwistah, which isn't true 3D anyway, but it is great fun and works well).</p>		
<p>For example, with Containerturm, I spend most of the time trying to figure out when the ball and the bat are in the same plane (doesn't help that the balls' shadows seem to be a bit glitchy). So I keep swinging for the damn thing and missing. I'd really like to rotate the camera around to get a better look, but I can't (or if I can, I missed how).</p>		
<p>Similar story with Phrasetrakker, better camera control is pretty much essential. Also, a drag and drop system for positioning the markers would be better, and hopefully also allow spinning of the wheel at the bottom without inadvertently planting markers down there.</p>		
<p>So yeah... really nice work, but 3D input needs to be near-perfect to be usable, and this isn't there yet.</p>		
Back to top	<input type="button" value="Profile"/> <input type="button" value="Pers. Msg"/> <input type="button" value="Website"/>	
usr Bishop	Posted: Tue Jul 26, 2005 12:45 am Post subject:	<input type="button" value="Quote"/>
	<p>good point, anyone up for quakesequencer? :mrgreen: WHOEVER I LENT MY BLOK MODULAR TO /MSG ME ASAP</p>	
Joined: 12 Aug 2004		

d9 Choir Girl	Posted: Tue Jul 26, 2005 1:50 pm Post subject:	<input type="button" value="Quote"/>
	<p>usr wrote: quakesequencer</p>	
<p>The stuff of dreams.</p>		
Back to top	<input type="button" value="Profile"/> <input type="button" value="Pers. Msg"/>	

usr Bishop	Posted: Tue Jul 26, 2005 4:35 pm Post subject:	<input type="button" value="Quote"/>
	<p>a few years (?) ago i had the idea of something like a crossover between dance dance revolution and a shooter, that is, the weapons would be synchronized to the music, like, the closer you would match pressing the fire button with the base drum the bigger the blast of the rocket launcher, gun spread controlled by the hihats and stuff like that.</p>	
<p>WHOEVER I LENT MY BLOK MODULAR TO /MSG ME ASAP</p>		
Joined: 12 Aug 2004 Posts: 309 Location: erlangen, germany		
Back to top	<input type="button" value="Profile"/> <input type="button" value="Pers. Msg"/> <input type="button" value="Website"/>	

kiloton Believer	Posted: Wed Jul 27, 2005 12:04 am Post subject:	<input type="button" value="Quote"/>	<input type="button" value="Edit"/>	<input type="button" value="X"/>
Joined: 17 May 2005 Posts: 10 Location: in the basement	<p>Thanks a lot for taking a look and commenting on our designs. I don't want to make excuses, but what we have released is only an initial design, and actually we are grateful for your suggestions and criticisms. Even if you suggest we scrap it - that's useful info as well ☺</p>			
<p>BTD, rest assured that we aren't too happy about the camera navigation as well. That's the part where I was forced to</p>				

<p>usr Bishop</p>  <p>Joined: 12 Aug 2004 Posts: 309 Location: erlangen, germany</p>	<p>Posted: Tue Jul 26, 2005 4:35 pm Post subject: Quote</p> <p>a few years (?) ago i had the idea of something like a crossover between dance dance revolution and a shooter, that is, the weapons would be synchronized to the music, like, the closer you would match pressing the fire button with the base drum the bigger the blast of the rocket launcher, gun spread controlled by the hihats and stuff like that.</p> <hr/> <p>WHOEVER I LENT MY BLOK MODULAR TO /MSG ME ASAP</p>
<p>Back to top</p>	<p>Profile Pers. Msg Website</p>
<p>kiloton Believer</p> <p>Joined: 17 May 2005 Posts: 10 Location: in the basement</p>	<p>Posted: Wed Jul 27, 2005 12:04 am Post subject: Quote Edit X</p> <p>Thanks a lot for taking a look and commenting on our designs. I don't want to make excuses, but what we have released is only an initial design, and actually we are grateful for your suggestions and criticisms. Even if you suggest we scrap it - that's useful info as well ☺</p> <p>BTD, rest assured that we aren't too happy about the camera navigation as well. That's the part where I was forced to learn about projection matrices and such, and I really don't have the mathematical background for that (yet). So we scrapped the parts of the navigation that didn't work for this release. A drag-n-drop interaction scheme for Phrasetrakker was/is half-developed as well, but scrapped since it didn't always work.</p> <p>And yes, I agree that the most important part of an interactive 3D app is the interaction/input handling, which needs much attention. Actually, we might try and see how far we can push the FPS paradigm, since it seems easier from a control point-of-view. usr, your quakesequencer idea sounds interesting ☺</p> <p>Oh and btw, I'll look into Sketchup for future modeling chores, looks very cool!</p>
<p>Back to top</p>	<p>Profile Pers. Msg Website MSN</p>
<p>Display posts from previous: All Posts Oldest First Go</p>	