

Ästhetik findet sich vor allem im Live-Coding und in der Netzmusik wieder, denn die Kombination aus beiden lässt eine Rekursion wie eine Phrase aussehen, die sukzessiv von den Spielern variiert wird.

### **3 Anwendung der Kriterien**

In diesem Kapitel werden drei konkrete Beispiele hinsichtlich des Kriterienkataloges überprüft. Nach einer Beschreibung des Setups der Beispiele sollen Hauptmerkmale hervorgehoben werden und im späteren Rückblick dann bezüglich der Kriterien beleuchtet und gegenüber gestellt werden.

#### **3.1 Beispiele aus der improvisierten Laptopmusik**

Als Grundlage für das folgende Kapitel 3.1.1. KRK diente ein Interview mit Matthew Ostrowski, das im Anhang D abgedruckt ist. Für PowerBooks-UnPlugged in Kapitel 3.1.2 führte ich ein Interview mit Prof. Dr. Alberto de Campo (siehe Anhang E). Im letzten Fall werde ich aus eigenen Erfahrungen meinen Teil zur letzten Performance von tryek beschreiben.

Alle drei Formationen haben unterschiedliche Ansätze in Interaktion und der in Echtzeit generierten Computermusik.

##### **3.1.1 KRK**

Der gebürtige New Yorker studierte am Oberlin Conservatory in Ohio und am Institut für Sonologie in Den Haag. Matthew Ostrowski<sup>156</sup> spielte schon mit David Behrman, Nicolas Collins und John Zorn zusammen. Mit George Cremaschi<sup>157</sup> formierte er sich zu dem Live-Elektronik Duo KRK<sup>158</sup>, bei dem live-elektronische Stücke mit Computer (Ostrowski) und Kontrabass (Cremaschi) auf der Bühne improvisiert werden.

Computer und Bass sind zwei eigenständige Instrumente, die nicht etwa über ein Mikrofon verbunden sind. Ostrowski wählt während der Performance zwischen

---

156 Vgl. URL <http://www.ostrowski.info> (Zugriff am 08.06.2009)

157 Vgl. URL <http://www.cesta.cz/georgecremaschi.htm> (Zugriff am 08.06.2009)

158 Vgl. URL <http://www.myspace/krkelectric> (Zugriff am 08.06.2009)

vier digitalen Instrumenten und bedient diese mit einem P5 Handschuh, der ursprünglich als Game-Controller entwickelt wurde, und einem Controller namens *Nostramo* mit einem Rad und einigen Knöpfen. Das auf HID basierende Nostramo-Rad wird benutzt, um zwischen den Instrumenten des Computers zu wechseln. Der P5 unterstützt kein HID, aber die Daten, die eine höhere Auflösung als der MIDI-Standard haben, werden über ein Unix-Programm – *P-5osc* – zu OSC-Messages konvertiert und lokal wieder empfangen. Die Programme für die Audio-Synthese wurden ausschließlich in Max/Msp implementiert.

Der Grund für den P5 liegt – nach jahrelanger Erfahrung mit unterschiedlichen Controllern – in seiner Benutzung als Auslöser einer Geste und nicht der Kontrolle von einzelnen Parametern. Diese Geste wiederum steuert ein Bündel von Parametern an, die aber durch die gemeinsame Geste einen Ursprungsauslöser haben und dadurch in Abhängigkeit gebracht werden. Ostrowski sieht sich als Interpret und Musiker, der ein Instrument bedient und seine Programme auch instrumental aufbaut: nur wenn Parameter aktiv durch den Aufführenden geändert werden, erzeugt das digitale Instrument Klänge und dabei werden durch die Geste nicht nur ein, sondern gleich mehrere Parameter beeinflusst.

„If I move the bow closer to the bridge I am changing several parameters. Or if I change the pressure of the bow I am changing several parameters with one gesture. Because an acoustic instrument is an interdependent system and so I try to write code that is interdependent systems rather than simply mapping (...) but my goal is to have everything as interdependent as possible and not have my glove be a series of knobs but have it one more level of abstraction.“<sup>159</sup>

Ostrowski versucht also auch einfache und isolierte Zuordnungen zu vermeiden, weil Abhängigkeiten der Systeme untereinander und innerhalb, die er in Interaktion zwischen Programm und Interface implementiert, das Spielen mit dem Computer als Instrument unterstützen.

Die Entscheidung den Kontrabass nicht in Echtzeit über ein Mikrofon an den Computer zu übertragen beruht auf der Aussicht von Klangräumen, die sich auf einer Achse zwischen kontrastreich und ähnlich befinden. Diese Achse von Möglichkeiten ist für Ostrowski durch eine Collection von Kontrabass-Samples eher zu erreichen, als seinen Mitspieler direkt aufzunehmen. Außerdem gehen für

---

159 Interview Anhang D, S. 90

ihn die meisten Live-Elektronik-Techniken, die allgemein eingesetzt werden, nicht über Looping und Transposition hinaus, wovon er sich distanziert.

Streng algorithmische Elemente in den Stücken gibt es nicht, weil Ostrowski zum einen durch die Beschäftigung mit algorithmischen Prozessen nicht zu Ergebnissen gekommen ist, die ihn klanglich überzeugten. Der andere Grund besteht darin, dass ihn auch Klangresultate von zum Beispiel George Lewis, der Programme für Live-Elektronik schrieb und eigentlich einen guten Ruf auf diesem Gebiet genießt, nicht motivierten in dieser Richtung weiter zu forschen. Einzig die Real Time Composition – Toolbox (RTC) für Max/Msp liefern 12-Ton- und Strukturgeneratoren, die für Echtzeit-Disposition von ihm genutzt werden.

Unabhängig von der Strukturgenese ist die Klangsynthese oft an ein physikalische Prinzip angelehnt, das als Physical Modelling implementiert wird. Zum Beispiel ist eines der digitalen Instrumente ein Granular-Synthesizer, der verschiedene Positionen des Samples über ein Masse-Sprungfeder-Modell ausliest. Der Audio-Buffer wird in mehrere Segmente unterteilt und jede Segment-Schnittstelle ist ein Lesekopf in dem Sound-File. Wie in dem physikalischen Prozess auch, werden die Leseköpfe durch die Bewegung mit dem Handschuh angeregt und schwingen, abhängig von der Stärke der Handbewegung und der Distanz und Stärke der Feder, hin und her, bis die Energie gleich Null ist. Die Segmente beeinflussen sich bei diesem Prozess gegenseitig, da sie nicht alle gleichzeitig in dieselbe Richtung ausschlagen, sondern auch gegeneinander schwingen und sich dadurch in der Amplitude ausbremsen.

„Things like the velocity, the acceleration and those kinds of parameters are also controlling things like pitch and timbre and pan position and stuff like that. So basically the more active I am the higher the pitch, the crinklier the timbre, the more dispersed it is in space etc. etc. So that's the kind of idea where I'm doing a single gesture and there's this whole physical model layer between me and that set of parameters.“<sup>160</sup>

Das virtuelle Instrument setzt dasselbe Paradigma voraus, wie bei mechanischen Instrumenten auch. Für den Rezipienten ist Klangsynthese eindeutig nachvollziehbar, denn nur wenn sich der Spieler bewegt, wird der Klang generiert. Ein weiterer Aspekt, der dem Publikum entgegen kommt, ist die Körperlichkeit des

---

160 Interview Anhang D, S. 95

Performers. Das Setup und das Interface ist so aufgebaut, dass der Spieler immer sichtbar ist und die Interaktionen einsehbar sind. Diese konsequente Analogie im Vergleich zu mechanischen Instrumenten macht die Verhältnisse zwischen Mensch und Computer deutlich.

Ausgehend von einer Basis-Struktur der Stücke von KRK wird über Klangmaterial improvisiert, das vor dem Konzert schon festgelegt ist. Klangdimension und grober Plan des Stückes ist somit abgesprochen und während der Stücke geht es vor allem um Detailarbeit am Klang selbst. Ostrowski sieht sich nicht als Erfinder von neuen Formen, weil es für ihn musikalische Elemente gibt, die nicht ignoriert werden können und benutzt deshalb Themen, Variationen und Scherzos als Grundlage für seine Arbeit und stellt kontrastierende Momente von kleinen und großen Gesten, Stabilität und Zerbrechlichkeit oder Vorhersehbarkeit und Spontanität gegenüber. Gute Improvisation zeichnet sich für ihn durch das spontane Entwerfen von weitreichenden Struktur und Formen, die in einem eigenen Stil münden.

### **3.1.2 PowerBooks\_UnPlugged**

Die Mitglieder von PowerBooks\_UnPlugged sind Alberto de Campo, Julian Rohrhuber, Echo Ho, Hannes Hoelzl, Jan-Kees van Kampen und Renate Wieser. Die Spieler sitzen entweder unverstärkt als Teil des Publikums mit im Raum und nutzen nur die Lautsprecher der Laptops. Oder sie werden, dies aber in eher seltenen Fällen, durch einen Subwoofer mit verstärkt. Alle Spieler benutzen SuperCollider als verbindende Software, die von den Mitgliedern auf die Ensemble-Struktur hin erweitert wurde (vgl. dazu Kapitel 2.3.4).

Jedes Mitglied kann während der Performance auf einen gemeinsamen Pool von Struktur- und Audio-Generatoren zugreifen, als Library zur Verfügung stehen. Weiterhin werden zur Laufzeit evaluierte Prozesse in einer gemeinsamen History abgelegt, die jeder einsehen und für Weiterverarbeitung verwenden kann. Besonders gelungene Algorithmen von früheren Performances dienen auch als Ideen-Geber. Somit wird nicht live ein neues Programm geschrieben, sondern vorhandene Programme werden verändert. Der ursprüngliche Code wird durch

den Dialog der Mitspieler manipuliert und weitergereicht und erfährt eine Endlos-Variation, die nur groben Form-Richtlinien unterliegt.

Die Absprache der Formteile ist nicht das Hauptziel einer Aufführung und sind meist auch nur grob skizziert (Struktur-Flächen fallen im Mittelteil auseinander, Rausch-Rampe zum Ende hin). Dem Ensemble ist viel wichtiger, dass sich durch den Dialog über den Code neue und spontane musikalische Wendungen ergeben, die nicht abgesprochen sind. Die Dialoge werden über drei Kommunikationsebenen geführt:

- Sound-Events können lokal oder auf den Rechnern der Mitspieler ausgeführt werden, der im gleichen Netzwerk ist.
- Der ausgeführte Text wird über die History an alle Mitspieler gesendet.
- Text-Messages können als Broadcast an alle Spieler gesendet werden und erscheinen groß über den ganzen Bildschirm

Ein weiteres Merkmal von PowerBooks\_UnPlugged ist der Gebrauch des Laptops als Instrument nicht nur hinsichtlich der Klangsynthese, sondern auch als Schallwandler und Resonanzkörper. Dieser konsequente Schritt den Laptop, mit dem Klänge generiert werden, auch als Instrument zu benutzen, hängt auch damit zusammen, dass die somit erreichten Lautstärken auch eindeutig mit den physikalischen Eigenschaften und Grenzen des Instrumentes selbst verbunden sind. Die Folge ist, dass minimale Bewegungen mit Maus und Tastatur auch eher mit den Klängen korrelieren und sich keine Diskrepanz von Aktion und Nachvollziehbarkeit einschleicht. Die akustischen Einschränkungen stellen auch gleichzeitig eine Herausforderung dar, fordern den Spieler auf Klänge zu produzieren, die meist über 250 Hz liegen und sogar den Lautsprecher-eigenen Frequenzgang optimal ausnutzen (zwischen drei bis fünf kHz sind Frequenzen präsenter angelegt).

Die Software-Erweiterungen zu jitLib und SuperCollider beziehen sich auf die schon erwähnte History für den Austausch von Code und dem Aufruf der Mitspieler als Teil des Synthese-Prozesses in den Algorithmen. Dazu müssen alle Mitspieler über ein Kürzel und einer IP-Adresse in der Ensemble-Konfigurationsdatei registriert sein, welche dann innerhalb des Codes in

Verbindung mit zum Beispiel der *where*: Klausel aufgerufen werden können.

```
(  
// birdsOfCanada  
Pdef(\birdsOfCan, Pbind(  
  \i, Pseries(0, 1, 100 * q.ipNums.size),  
  \where, Pkey(\i) % q.ipNums.size,  
  \instrument, "glisson",  
  \freq, Pkey(\i) % 10 * 100 + 1000,  
  \freq2, Pkey(\i) % 13 * -100 + 3000,  
  \sustain, 0.03,  
  \amp, 0.1,  
  \dur, Pfunc({ levl (4 / (ev.i + 10)) })  
)).play;  
)
```

Das hier abgebildete Event-Pattern *Pdef* erzeugt das Instrument *glisson* und ruft abhängig von der Variable *i* und dem Modulo der Länge der Liste der eingetragenen IP-Adressen immer auf einem anderen Rechner über die *\where* – Klausel auf.

Die Besonderheit von *jitLib* ist der Proxy-Space, der in Kapitel 2.3.4 und im Anhang C erläutert wird. Der Wechsel von einem Synthese-Prozess zum nächsten wird über Crossfades ohne digitale Störgeräusche vollzogen, was beim Erzeugen von DSP-Objekten im Max/Msp immer zu hören ist.

Der hierfür entwickelte Proxy-Mixer zeigt zum einen übersichtlich alle Proxy-Busse und kann diese auch über virtuelle Slider wieder dem Mix-Bus hinzufügen. Da bei einer Vielzahl von aufgerufenen Prozessen die Kontrolle von dem letzten evaluierten Zustand des Busses verloren gehen kann – SuperCollider bietet keine Syntax-Markierung, die auf ein benutzte Prozedur verweist – kann der zuletzt aufgerufene Code über den *doc*-Button angezeigt werden. Diese letzte Kontroll-Instanz schafft neben dem Live-Coding selbst und den Kommunikationsebenen eine Regie, mit der globale Entwicklungen geführt werden können oder bei Fehlern die betreffenden Kanäle schnell auf stumm geschaltet werden können.

### 3.1.3 Kurbelrichter von tryek<sup>161</sup>

Kurbelrichter bezieht sich auf ein Prinzip, das wir als immer wiederkehrendes

---

<sup>161</sup> Das Stück 'Kurbelrichter' wurde am 18.05.2009 im Rahmen des Konzerts der Kompositionsklassen der Folkwang Hochschule in Essen uraufgeführt.

Formelement konzipiert haben und bei dem ein Takt in drei Vierteltriole unterteilt wird. Jede Triole wird einem Spieler zugeordnet und dieser darf nur in dieser Dauer seine Generatoren Klänge synthetisieren lassen. Als Überbau griff eine globale Geste (siehe Abbildung 6), die über eine Länge von 10 min gezeichnet war. Diese Geste tauchte auch in kleineren Strukturen immer wieder auf. Die Partitur war in vier Hauptabschnitte unterteilt, deren Klangräume grob skizziert waren. Abhängig von den Hauptstationen änderten sich bei jedem Spieler die globalen Einstellungen der Generatoren und auch die Bildprozesse.

Bei dem Stück Kurbelrichter geht es auch darum Kontrolldaten, die die Audiounits steuern und entweder von Automaten oder Interfaces kommen, auf Video-Objekte zu mappen. Das Video-Environment wurde von Christof Schnelle in Openframeworks<sup>162</sup> programmiert und über OSC angesteuert. Die Idee hierbei war das Zeichnen von drei unterschiedlichen Farbklecksen, die die Bewegungsverläufe von Pollocks Action-Painting imitieren sollten.

Die Performer befanden sich in der Mitte des Parketts des Saales vor dem Mischpult, es war ihnen also nicht daran gelegen während der Aufführung gesehen zu werden. Zum einen weil das Video visueller Schwerpunkt sein sollte und zum anderen auch, weil die Körperlichkeit nicht in einem eindeutigen Verhältnis zum musikalischen Ausdruck stand.

Die Ausgangsbelegung der acht Kanäle war so geregelt, dass zwei Vierkanal-Ringe übereinander von den drei Spielern belegt wurden. Die Vorderseite des unteren Ringes wurde von Spieler 1 belegt, die Rückseite von Spieler 2. Der obere Ring wurde zunächst nur von Spieler 3 bedient. Spieler 3 bediente auch einen Modulator (Allpass), der später in Teil 3 und 4 von den ersten beiden Spielern mit versorgt wurde. Abbildung 7 zeigt die Lautsprecherkonfiguration.

Als Interfaces wurden vor allem Fader-Boxen und Endlos-Potis benutzt, genauso wie Tastatur, Maus und ein iPod. Die Master-Geste wurde auf allen Rechnern synchron gestartet und durch einen roten Cursor war immer angezeigt, wo die Partitur gerade ausgelesen wurde. Der aktuelle Wert der Geste wurde von jedem Spieler unterschiedlich interpretiert. Der Server konvertierte diesen Wert der

---

162 Vgl. URL: <http://www.openframeworks.cc/> (Zugriff am 26.05.2009).

Geste und ermittelte das Tempo, das sich im Wertebereich von 60 – 240 abhängig von der Geste das ganze Stück über änderte.

Die Beschreibung der Umsetzung der Partitur bezieht sich im Folgenden nur auf die Konfiguration meines eigenen Setups. In Abbildung 8 ist die Schaltung schematisch dargestellt. Diese lässt sich gut in automatisierte und manuell gesteuerte Parameter unterteilen, denn alles, was von den Strukturgeneratoren zu den Audiogeneratoren fließt sind automatisierte Daten – von der Tastatur, iPod und Fader-Box kommen nur manuell gesetzte Parameter. Durch den Wechsel von Presets wird lediglich die Verschaltung geändert und somit der Datenfluss, wodurch Automation und Improvisation im starken Kontrast stehen. Improvisiert wird alles, was manuell gesetzt ist und in diesem Fall ist die Master-Geste nur ein Richtungsweiser. Die Automatismen leben also überwiegend von Sequenzen, die über einen oder mehrere Takte laufen. Es sind Pattern die einzeln oder kombiniert auf die Parameter zugreifen und durch Algorithmen moduliert werden. Die Pattern werden als Listen mit anderen Listen kombiniert oder ihre Stützwerte werden ausgetauscht, wodurch sich neue Rhythmen oder Parameter-Sequenzen ergeben. Einziger Sonderfall ist die Geste als Makro- und Mini-Struktur, die mal als Hüllkurve, mal als diskreter Switch oder Gate funktioniert oder aber globale Parameter wie das Tempo setzt. Über die komponierte Struktur ist der direkte Zugriff über die Interfaces auf die Parameter der Aspekt der Improvisation, der versucht wurde mit der Geste zu koppeln. Die improvisierte Interaktionen bestand vor allem in der Detailarbeit des Klanges selbst. Neben den musikalischen Aspekten galt es auch die Übergänge der Hauptabschnitte vorzubereiten und die Presets zeitlich umzustellen. Das ist eher ein administrativer, denn ein musikalischer Akt, der aber musikalische Konsequenzen hat (ähnlich wie das Blättern der Partitur).

## **3.2 Rückblick**

Bei allen Performance-Ideen, die in Kapitel 3.1 besprochen wurden, stelle ich fest, dass das Thema Nachvollziehbarkeit, wenn auch sehr unterschiedlich, bedacht wurde. Der eher klassische Fall des KRK-Duos setzt das Instrumenten-Paradigma

um und überzeugt durch Eindeutigkeit in Gestik und Körperlichkeit, die vor Algorithmen für die Strukturgenese gestellt werden. Der Computer ist als Physical Modelling-Instanz reaktiv und fällt in die Kategorie des Service. Es werden keine autonomen Entscheidungen getroffen und jede Klangsynthese beruht auf den Aktionen des Spielers. Zwar wird im Fall von PowerBooks\_UnPlugged die Generierung nicht über Gesten formuliert, allerdings bietet die Nähe des Publikums zum Spieler und die eindeutige Ortbarkeit des Klanges einen schnellen Zugang zum Ursprung der Klänge. Die Herausforderung für den Hörer besteht vornehmlich darin, die Entwicklung der Prozeduren als Dialog zu verstehen und wieder zu erkennen. tryek sieht die Übertragung der Strukturen in ein Bildformat als eine Lösung, die zusätzlich noch dadurch unterstützt wird, dass Spielergesten sich nicht nur in Klang und Bild auswirken, sondern auch beim Performer selbst zu sehen sind, da nicht nur Maus und Tastatur als Controller benutzt werden.

Alle drei Konzepte unterscheiden sich auch in der Tiefe der Veränderung des Materials über einen Algorithmus. Das Material bei KRK wird direkt und nur während der Performance generiert, was viel Übung voraussetzt, aber Interaktion und Klanglichkeit über den Algorithmus und damit über den Computer als Mit- oder Gegenspieler stellt. Bei tryek werden Pattern und Verläufe vorformuliert, die während der Performance leicht verändert und kombiniert werden. Zum einen ist das als autonomes Element des Rechners zu werten und zum anderen als Strukturrahmen für das Stück. Das Ensemble um Alberto De Campo ist vor allem daran interessiert Algorithmen live zu entwickeln und diese durch den Dialog zu verändern. Der Grad der Autonomie ist hier also am höchsten.

Ausgehend von der Idee, dass, je komplexer die Parameter werden, je geringer die Spielerflexibilität wird, ist eine Kombination aus digitalem Handschuh und einer Physical-Modelling-Schnittstelle am wirkungsvollsten. Mit einer Geste können viele Parameter, die in Abhängigkeit zueinander stehen, verändert werden und den Klang beeinflussen, der durch sein Modell nicht nur im Moment der Geste klingt, sondern auch noch eigene Ausschwingprozesse hat. Das Live-Coding ist bezogen auf die Semantik des Algorithmus sehr flexibel und steuert viele Parameter zugleich. Jedoch sind dies Prozesse, die solange im Proxy-Bereich zu hören sind,

bis sie entweder stumm geschaltet oder ersetzt werden, als Klänge die ganze Zeit präsent und erfahren keine Änderung. Bei dem Stück 'Kurbelrichter' habe ich es oft als Gratwanderung empfunden, weil viele Parameter auf struktureller und klanglicher Ebene zu kontrollieren waren und mich dadurch manchmal bei der Improvisation behindert haben.

Alle drei Projekte halten sich von Learn-Algorithmen und Gestenerkennungen fern und nur im Fall von KRK wird Gestik als konkretes Element auf der Bühne eingesetzt. tryek bezieht seine Formabschnitte zwar auf eine Geste und dieselbe auch auf Mikrostrukturen, zeigt diese aber nicht als spielerischen Teil für die Bühnenpräsenz. Auch bei diesen beiden Fällen sind die Interaktionsmechanismen deutlich reaktiver als im Falle von PowerBooks\_UnPlugged. Es werden in den meisten Fällen nur Filter- und Generator-Parameter gesetzt, wohingegen im letzten Fall ganze Strukturen neu entwickelt werden und Teil der Interaktion sind. In keinem der drei Beispiele reagieren die Spieler auf Aktionen des Rechners oder müssen sich gar passiv verhalten. Das bedeutet, dass die Rolle des Computers als autonomes Mitglied in der Interaktionskette nicht in Betracht gezogen wurde. Dadurch scheidet auch die Möglichkeit aus den Grad der Interaktivität durch intelligente Prozesse zu erweitern und den Spielern mehr Affordanzen zu ermöglichen.

#### **4 Schlussfolgerung**

Die anfängliche Übersicht von Hörarten und Hörerwartungen bei Laptopmusik hat gezeigt, dass die kritische Haltung auch dadurch stärker wird, dass die Anzahl an Laptop-Performances zunimmt. Das ist gut so, denn so wird die spontane Begeisterung von Künstlern, die mit dem Laptop auf der Bühne Musik machen wollen, schnell gebremst und die Fragen bezüglich Körperlichkeit, Interaktion und Nachvollziehbarkeit aufgeworfen. Damit einher geht auch eine Hör-Entwicklung der Rezipienten, die immer häufiger mit Laptop-Musik konfrontiert werden und sich von älteren Hörmustern lösen. Das wird nicht nur dadurch unterstützt, dass Künstler sich auf der Bühne oder im Konzertsaal von der typischen