

Komponieren mit dem Interface

Plädoyer für mehr Computerkompetenz in der Musikwissenschaft

1 Aus stilistischen Gründen ist die weibliche Form hier und im folgenden bei Bezeichnungen wie Komponist, Künstler, Musikwissenschaftler nicht ausgeschrieben, aber immer mit eingeschlossen.

In den ersten Jahrzehnten der Produktion von elektroakustischer und Computermusik war die Arbeit des Komponisten¹ an Studios gebunden, die die technische Ausstattung und das unterstützende Wissen zur Verfügung stellen konnten. Mit der Verwendung des Computers in der elektroakustischen, der Computermusik und der Klangkunst und im Hinblick auf seine Miniaturisierung ist seit Ende der neunziger Jahre vom sogenannten »Laptop-Studio« die Rede. Klangsynthese, aufwendige algorithmische Kompositionsverfahren oder mehrspurige Aufnahmesysteme sind nun »niedrigschwellig« verfügbar und ermöglichen einen Großteil der künstlerischen Arbeit abseits der etablierten Studios und Institutionen.

Wie aber ist der Umgang des Künstlers mit dem Computer charakterisiert? Welche qualitativen Aussagen lassen sich über die Beschaffenheit des Kompositionswerkzeugs »Computer« überhaupt treffen? Wie verhält sich der Arbeitsprozeß am Computer zum Werk in ästhetischer Hinsicht? Wirken die verwendeten Werkzeuge zurück auf den Komponisten und seine Musik?

Analyseverfahren

Versuche, sich diesen Fragen zu nähern, werfen verschiedene Probleme auf. Das größte von ihnen ist nicht neu: Der Prozeß des kompositorischen Schaffens entzieht sich weitgehend dem analytischen Zugriff. Sichtbares und allgemein zugängliches Anzeichen dafür, daß ein solcher Prozeß stattgefunden hat, ist nach wie vor das musikalische Resultat. Bei elektroakustischer Musik ist das häufig nur das klingende Ergebnis, als Datei oder gespeichert auf Tonträger, in der Klangkunst die fertige Klanginstallation. Im letzten Fall steht oft nur eine Dokumentation der Installation zur Verfügung, meist in Form eines inszenierten Videos, das viele künstlerische Aspekte der Installation nur unzureichend wiedergeben kann wie etwa den klanglichen Umgang mit einer vorgefundenen Raumsituation. Live-elektronische

hin einer detaillierten Beschreibung des technischen Aufbaus, dennoch werden verwendete Strategien und Algorithmen meist in der Software verborgen bleiben, zumal sich auch aus ihnen kaum ästhetische Aussagen über die Musik ableiten lassen. Weitere Arbeitsmaterialien des Künstlers sind selten zugänglich. Die Frage, ob nachträglich vom Komponisten angefertigte Partituren für eine ästhetische Einschätzung überhaupt geeignet sind, ist schon oft diskutiert worden.

Der analytische Zugriff auf diese Musik muß sich also weiterhin hauptsächlich auf das klingende Werk, seine Dokumentation und auf jene zusätzlichen Materialien stützen, die der Komponist öffentlich oder auf Nachfrage zugänglich macht. Dabei ist hinzunehmen, daß die Selektion und Reflexion dieser Materialien durch den Komponisten selbst bereits einen starken Einfluß auf die Analyse ausüben können. Um so wichtiger sind daher umfassende Kompetenzen des Musikwissenschaftlers, die Materialien einzuordnen, sie zu verstehen und zu deuten. Im Bereich der elektroakustischen, der Computermusik und der Klangkunst müssen diese Kompetenzen neben den musikalischen Mitteln, dem »Was?«, immer auch das »Wie?« umfassen. Untersuchungen zur Verwendung bearbeiteter oder synthetischer Klangfarben oder die Analyse der Raumkomponente bei mehrkanaliger Lautsprechermusik nutzen solche Materialien erst dann umfassend, wenn der Möglichkeitshorizont und die Beschaffenheit des primären Kompositionswerkzeugs, des Computers, in die Betrachtungen einbezogen werden. Dies umfaßt sowohl die Kenntnis der physikalischen Prinzipien der Synthese, Bearbeitung, Montage oder Projektion von Schall, deren Anwendung durch den Computer ermöglicht werden, als auch ein Verständnis dieses Anwendungsprozesses selbst: der verwendeten Computerprogramme und ihrer Beschaffenheit.

Die physikalischen und technischen Zusammenhänge der musikalischen Arbeit mit dem Computer werden in der Literatur umfangreich behandelt. Derartige Abhandlungen sind meist historisch orientiert oder sie folgen einer technischen Systematisierung, oft in arbeitsanleitender Form (*tutorial*). Seltener erfolgen Versuche einer Systematisierung von Klangsyntheseverfahren nach ästhetischen Gesichtspunkten. Von Martin Supper stammt die Gegenüberstellung von vertikaler und horizontaler Klangsynthese.² Mit vertikaler Klangsynthese ist hier die »direkte« spektrale Komposition der Klangfarbe gemeint, während die horizontale Synthese die »Komposition der Klangfarbe in der Zeit« bezeichnet. Diesen bei-

2 Martin Supper, *Elektroakustische Musik und Computermusik. Geschichte, Ästhetik, Methoden, Systeme*, Darmstadt 1997, S. 3762

den Klassen lassen sich Supper zufolge sämtliche wesentliche Syntheseverfahren zuordnen, die additive Synthese etwa der vertikalen, die Granularsynthese der horizontalen. Deutlich wird dies auch an der annähernd deckungsgleichen, prinzipiellen Gegenüberstellung von Fourier und Gábor, mit der Supper die beiden Extrema seiner ästhetischen Einordnung von Klangsyntheseverfahren begründet. Ästhetisch führe die Anwendung der Fourieranalyse vorhandener Klänge zu einem »instrumentellen Denken«, während das klangliche Resultat horizontaler Synthese »meist fern jeder Ähnlichkeit zu akustischen Instrumenten« sei, »wirklich innovative Klangkomposition« mithin erst durch die Loslösung »von der analytischen Orientierung an akustischen Instrumenten« entstehe.

Die ästhetischen Implikationen vertikalen (spektralen) und horizontalen (zeitlichen) kompositorischen Denkens sind wichtige Aspekte bei der Analyse elektroakustischer Musik, die oftmals auf komplexe Weise miteinander verschränkt sind. Der Wunsch, zusätzliches Wissen über die vom Komponisten angewendeten Verfahren in eine Analyse einzubeziehen, entspringt der Einsicht, daß eine nur phänomenologische, klangliche Betrachtung selten umfassend über derartige Bezüge Aufschluß gibt. Der Versuch, die beiden ästhetischen Aspekte des Vertikalen und des Horizontalen mit einer rein physikalischen Klassifikation von Klangsyntheseverfahren in Deckung zu bringen, stellt jedoch einen problematischen Ausweg dar. Die einzelnen Syntheseverfahren sind natürlich besonders im Hinblick auf ihr klangliches Ergebnis nicht ästhetisch neutral. Sie sind aber nicht nur über ihre physikalischen Eigenschaften in den Kompositionsprozeß eingebunden, sondern vielmehr durch die Computerprogramme, mit denen der Komponist diese Verfahren anwendet. Die Art des Zugriffs auf die verschiedenen Parameter eines Syntheseverfahrens kann weit stärker auf das kompositorische Denken wirken als das physikalische Verfahren selbst. Darüber hinaus enthält die konzeptuelle Gestaltung der Benutzungsoberfläche von Kompositionsprogrammen oft musikalische Annahmen über die mit ihnen produzierten klanglichen Resultate, die der Komponist bei der Arbeit mit dem Programm nur schwer umgehen kann.

Technologische Mittel

Deutlich sichtbar ist dies etwa anhand der Trennung von Instrumenten und Partitur in der weit verbreiteten Software *Csound*, deren historische Entwicklung auf die ersten Pro-

gramme zur digitalen Klangsynthese von Max Mathews zurückgeht. Um ein klangliches Ergebnis mit *Csound* zu realisieren, müssen zunächst Klangsynthesearchgorithmen beschrieben werden, die Instrumente (orchestra). Diese werden flexibel aus atomaren signalverarbeitenden Einheiten, sogenannten unit generators, zusammengesetzt und erhalten Platzhalter für freie Parameter wie Tonhöhe oder Wellenform. Die Instrumente können dann in einer Partiturdatei (score) aktiviert werden, indem der Einsatzzeitpunkt und weitere Parameter spezifiziert werden. Im eigentlichen Syntheseprozess verarbeitet das Programm diese Informationen und berechnet für jedes Partiturergebnis die digitale Repräsentation des Klangs, den das jeweils aktivierte Instrument synthetisiert. Es gibt zahlreiche Kompositionen, etwa *Boundless Space* von John Fitch, deren Formgestaltung derart stark von diesem Konzept beherrscht ist, daß dem Zuhörer die *Csound*-Bildschirm Ausgaben der einzelnen Partiturergebnisse förmlich vor dem inneren Auge ablaufen, wenn er das Programm selbst schon einmal benutzt hat.

Dieses Charakteristikum wirkt allein über die Benutzungsoberfläche, das Interface des verwendeten Programms, und ist nicht über die technischen Möglichkeiten der Software erfaßbar. Prinzipiell läßt sich mit *Csound* nahezu jede Form der Klangsynthese umsetzen, es gilt sogar als das Programm mit der umfangreichsten Bibliothek an unit generators. Bei der Analyse läßt das Wissen um die Beschaffenheit des Interface der verwendeten Software aber Rückschlüsse darauf zu, welche möglichen »Umwege« und »Kniffe« der Künstler für die Umsetzung seiner kompositorischen Idee gehen bzw. anwenden mußte und kann damit auch bei spärlichen zusätzlichen Informationen wertvolle Einblicke in den Entstehungsprozeß einer Komposition geben.

Rückt neben dem klingenden Resultat auch der *Prozeß* des musikalischen Schaffens mit dem Computer in den Fokus der Analyse, werden Gesichtspunkte des Interface noch weit wichtiger. Dies ist unmittelbar einleuchtend für die Betrachtung von live-elektronischer Musik oder Improvisationen, weil die Arbeit am Computer zum Zeitpunkt der Aufführung und zumeist sichtbar stattfindet. Hinsichtlich der Interface-Problematik gibt es hier aber keinen qualitativen Unterschied zur eher verborgenen Arbeit des Komponisten von Tonbandmusik, der nicht erst bei der Umsetzung, sondern schon bei der Findung des Kompositionsvorhabens seinen Werkzeugen ausgesetzt ist: »Häufig entsteht sogar erst die künstlerische Intention durch eine Exploration des Materials, durch die unmittelbare Ausein-

3 Barbara Becker, Gerhard Eckel, *Künstlerische Imagination und Neue Medien. Zur Nutzung von Computersystemen in der zeitgenössischen Musik*, Arbeitspapiere der GMD, Nr. 960, St. Augustin, Dezember 1995, zitiert nach: <http://iem.at/~eckel/publications/beck95a/beck95a1.html>, Abschnitt III.5

5 Aus der Korrespondenz des Autors mit dem Komponisten.

6 Aus der Korrespondenz des Autors mit dem Komponisten.

4 Gerhard Eckel, Ramón González-Arroyo, Martin Rumori, *foo sound synthesis*, <http://foo.sourceforge.net>

foo_interface: Arbeitssituation mit dem textuellen Kompositionsprogramm »foo« in Kombination mit dem Soundfile-Editor »snd«.

andersetzung mit den technologischen und ideellen Mitteln, die aktuell verfügbar sind.«³ Die Annäherung an einen derartigen Begriff des kompositorischen Prozesses setzt eine systematische Analyse der technologischen Mittel voraus, wie sie sich dem Komponisten bei der praktischen Arbeit darstellen. Dazu ist eine sehr konkrete Auseinandersetzung mit den tatsächlich verwendeten Geräten und ihrer Ergonomie erforderlich, insbesondere solchen zur Ein- und Ausgabe von Daten und mit der Gestaltung von Computerprogrammen für musikalische Anwendungen. Für die Analyse des Kompositionsprozesses ist ein Katalog von Kriterien hilfreich, mit denen sich die Benutzungsoberfläche solcher Programme charakterisieren läßt.

Kompositionsprogramme

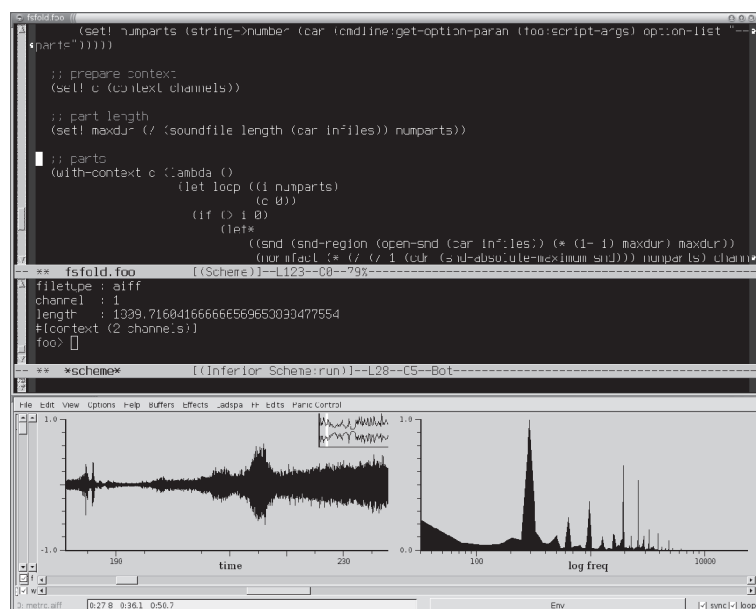
Ein zentrales Unterscheidungsmerkmal von Kompositionsprogrammen in dieser Hinsicht ist, ob die Klänge in Echtzeit, also unmittelbar, erzeugt und verändert werden können, oder ob ein iterativer Prozeß der Programmierung, Berechnung und des Hörens des klanglichen Resultats erforderlich ist. Zur Zeit herrscht die Meinung vor, Echtzeitsysteme seien allgemein benutzungsfreundlicher und stünden dem Komponisten bei der Arbeit weit weniger »im Weg«. Die Zeit jedoch, die ein Nicht-Echtzeitprogramm zur Berechnung der Klangdaten benötigt, kann integraler Bestandteil des Kompositionsprozesses sein: Nachdem die Kompositionssoftware *foo*⁴ vom Autor zur Verwendung auf modernen Computersystemen übertragen worden war, »beschwerte« sich der spanische Komponist Ramón González-Arroyo, das Programm sei nun so schnell, daß er während der nur noch kurzen Berechnungs-

phasen keine Zeit mehr habe, über das weitere kompositorische Vorgehen nachzudenken.⁵

Ein weiteres wesentliches Merkmal ist die Unterscheidung zwischen einer grafischen und einer textuellen Benutzungsoberfläche. Programme mit grafischer Oberfläche erscheinen grundsätzlich »freundlicher« und einfacher erlernbar, auch wenn sie dies bei näherer Betrachtung nicht unbedingt sind. Meist läßt sich aus einer Palette an Funktionen die gewünschte einfach auswählen, während sie bei textuell gesteuerten Systemen in einer bestimmten Programmiersprache formuliert werden muß. Die Untersuchung der Beschaffenheit solcher Sprachen, ihrer Paradigmen, Möglichkeiten und Einschränkungen, reicht weit in das Gebiet der Informatik, genauer des Compilerbaus und der Softwaretechnik hinein. Sie erscheint jedoch lohnend, denn sie kann Aufschluß darüber geben, welche Transformations- und Abstraktionsleistungen bei der computergerechten Formulierung musikalischer Ideen erbracht werden müssen.

Allerdings sind die technischen Möglichkeiten der Benutzungsoberfläche eines Kompositionsprogramms wohl in den seltensten Fällen allein ausschlaggebend für die Wahl eines Kompositionsprogramms. Vielmehr sind seine Komplexität und das nötige programmertechnische Wissen entscheidend bzw. der Aufwand, mit dem sich dieses Wissen erlernen läßt. Auch bei der Anwendung von Software gilt, daß gelernte Fertigkeiten, aber auch Gewohnheiten, viele Unzulänglichkeiten des Werkzeugs in hohem Maß kompensieren können. Das Stück *Para modulador de frecuencia* des portugiesischen Künstlers Tiago Cutileiro ist eine algorithmische Komposition, deren Klanglichkeit vermuten läßt, daß sie mittels einer kompakten programmiersprachlichen Formulierung in einer geeigneten Kompositionssoftware erstellt wurde. Auf Nachfrage stellte sich jedoch heraus, daß Cutileiro die algorithmischen Parameterverläufe auf Papier von Hand berechnet hatte, die zugrundeliegenden synthetischen Klänge mit dem Programm *Max/MSP* erzeugt und diese anschließend anhand der Ergebnisse seiner Berechnungen mit *Pro-Tools*, einer Software für Harddisk-Recording, in mühevoller Kleinarbeit, Parameter für Parameter, montiert hatte. Er habe gewußt, daß es Kompositionssysteme gibt, in denen sich derartige algorithmische Zusammenhänge recht einfach formulieren lassen, jedoch habe er für die Realisation seines Stücks keine neue Programmiersprache erlernen wollen.⁶

Auch die Verankerung musikalischer Gestaltungskategorien im Benutzungskonzept einer Kompositionssoftware spielt eine große Rolle, wie etwa der Zugriff auf die verschiede-



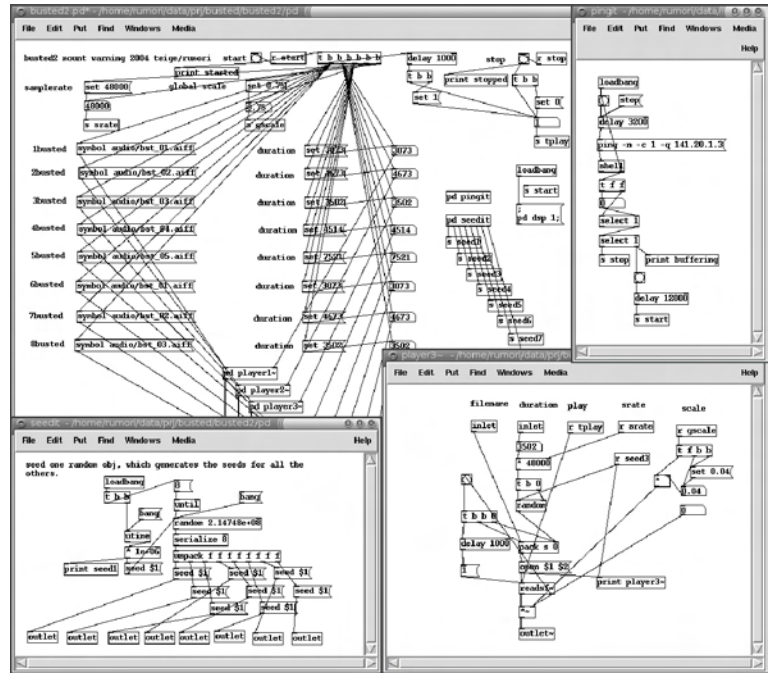
nen Zeitebenen. Ein besonders gutes Beispiel ist die oben erwähnte, konzeptuelle Trennung zwischen Instrumenten und Partitur, die sich häufig auch syntaktisch auf programmiersprachlicher Ebene niederschlägt und das Kontinuum der Zeitebenen künstlich bricht.⁷

Nicht zuletzt spielen soziale Faktoren eine entscheidende Rolle bei der Wahl eines bestimmten Programms für die musikalische Arbeit, etwa die überwiegende Nutzung einer bestimmten Software im Umfeld des Musikers, der »Ruf« eines Programms, oder verbreitete Annahmen über musikalische und computertechnische Kompetenzen des Anwenders einer bestimmten Software, seien diese nun begründet oder vorurteilsbehaftet.

In der eigenen klangkünstlerischen Praxis hat der Autor festgestellt, daß für viele technisch orientierte Betrachter die Realisation einer Klanginstallation bereits abschließend geklärt ist, sobald die Verwendung des weit verbreiteten Programms *Max/MSP* »eingerräumt« wird. Demgegenüber sorgt der Hinweis auf die Software *SuperCollider*,⁸ das als komplexer gilt und textuell programmiert wird, oft für direkte Anerkennung und weiteres Interesse an konzeptuellen Einzelheiten und deren Umsetzung.

Ist die verwendete Software nach außen sichtbar, etwa bei improvisierenden Laptop-Musikern, scheinen sich Aspekte des »persönlichen Stils« und der Selbstinszenierung nicht nur an der Musik, sondern auch sehr stark an den verwendeten Programmen zu manifestieren. Manche Musiker werden gar direkt mit einer bestimmten Software assoziiert. So hat sich etwa der amerikanische Programmierer und Musiker Ge Wang innerhalb von ein bis zwei Jahren durch wirksames Auftreten international zum Inbegriff seiner Live-Coding-Software *ChucK*⁹ stilisiert, unter anderem durch einen inszenierten Improvisations-Wettkampf gegen Nick Collins von der konkurrierenden *SuperCollider-Fraktion* auf der ICMC 2005 in Barcelona. Der Eindruck drängte sich auf, daß klangliche Aspekte hier in den Hintergrund traten, ebenso wie konzeptuelle Fragen der Softwaregestaltung, die gleichwohl bei beiden Programmen interessant sind. Die analytische Einordnung dieses musikalischen Ereignisses wäre ohne die Betrachtung der sozialen Funktion der Softwareanwendung und -entwicklung nicht mehr zu erbringen.

Die *community* der Anwender und Entwickler eines Programms spielt aber auch in weniger spektakulären Bereichen eine Rolle. Insbesondere bei Freier Software existiert oft eine über das Internet gut organisierte Gemeinschaft, die zum einen als wertvolle Anlaufstelle bei Fragen und Problemen dienen



kann, zum anderen aber auch ein Zusammengehörigkeitsgefühl und Anerkennung für eigene Beiträge gibt.

Diese Beispiele mögen genügen, um zu zeigen, wie wichtig es wäre, solche Aspekte der Benutzungsoberfläche von Computerprogrammen in die Analyse mit einzubeziehen. Das gilt sowohl für Komposition im herkömmlichen Sinn, als auch für Klangkunst und für die Verwendung des Computers als Performance- und Improvisationsinstrument. Die Arbeit mit diesen Kriterien setzt allerdings die genannten Kompetenzen voraus, die selbst in der systematischen Musikwissenschaft gegenwärtig kaum ausgebildet werden. Vielmehr konnte der Autor in seinem eigenen Studium immer wieder Vorbehalte gegen die umfassende Anwendung computertechnischen Wissens beobachten. Die Angst, sich bei intensiver Beschäftigung in technischen Einzelheiten zu verlieren und darüber die musikalischen Fragen zu vernachlässigen, ist durchaus berechtigt. Allerdings teilt der Analysierende diese Angst mit vielen Komponisten – die Erfahrung ebenso mit ihnen zu teilen, könnte einer angemessenen Analyse sehr zuträglich sein.

Vereinfachend ausgedrückt: das »Auto-graph« einer gegenwärtigen, digitalen Tonbandkomposition besteht zum wesentlichen Teil aus ihren Ausgangsmaterialien, der jeweiligen Hard- und Softwareumgebung und vor allem den Programmtexten, die der Komponist erstellt und verwendet oder verworfen hat. Ob diese nun zugänglich sind oder nicht: die Musikwissenschaft beraubt sich selbst eines ihrer zentralen Gegenstände, wenn sie nicht den Anspruch an sich selbst stellt, diese Texte lesen zu können.

pd_interface: Die graphische Programmiersprache »Pd«.

7 Vgl. die Klassifikation der Zeitebenen von Curtis Roads *Microsound*, Cambridge, Mass., London 2001, S. 3.

8 James McCartney, *SuperCollider: a new real time synthesis language*, in: *Proceedings of the International Computer Music Conference*, Hong Kong 1996; Homepage: supercollider.sourceforge.net

9 Perry Cook, Ge Wang, *Concurrent, On-the-fly Audio Programming Language*, <http://chuck.cs.princeton.edu>