

Zeitschrift:	Dissonanz = Dissonance
Herausgeber:	Schweizerischer Tonkünstlerverein
Band:	- (2011)
Heft:	114
Artikel:	Freiheit und Restriktion in der Computermusik : dargestellt an der Software Max/MSP und dem Scripting Interface Note for Max
Autor:	Gottstein, Björn
DOI:	https://doi.org/10.5169/seals-927548

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 23.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Freiheit und Restriktion in der Computermusik

Dargestellt an der Software Max/MSP und dem Scripting Interface Note~ for Max

Björn Gottstein



«This note's for you.»

Neil Young

1

«Man hatte gemeint, dass das elektronische Studio die Wunderwelt aller nur denkbaren Klänge eröffnen würde. Das hat es vielleicht nicht getan. Das hat auch bisher der Computer nicht getan. Aber der Möglichkeit nach ist es natürlich drin. Ganz sicher.» Dies erklärte Gottfried Michael Koenig einst im Rückblick auf fast 50 Jahre elektronische Musik.¹ Es spricht nicht so sehr Resignation aus dieser Feststellung als vielmehr die Erkenntnis, dass in der elektronischen Musik noch längst nicht alle Probleme gelöst sind. Zwar verheisst die Arbeit mit elektronischen Speichermedien einen grossen Freiheitsgrad, aber die Restriktionen, die von der Technik ausgehen, sind immer noch enorm.

Das Versprechen der Freiheit geht seit jeher mit der Technifizierung der Tonkunst einher. Apparaturen dienten etwa Ferruccio Busoni und Edgar Varèse als Mittel, um sich über historisch sedimentierte Konventionen hinwegsetzen zu können.² Busoni imaginerte die Erweiterung des geläufigen Ton- und Stimmungssystems, Varèse war vor allem an neuen Klangfarben interessiert.

Dass dieses Versprechen bis heute kursiert und ständig erneuert wird, liegt wohl vor allem daran, dass es nie eingelöst wurde. Der Slogan «Imagine a sound! Now make it!»³, mit dem der Synthesizer-Hersteller EMS sein Instrument pries, mag werbestrategisch opportun sein. Der Wirklichkeit entspricht er nicht. Die Geschichte der elektronischen Musik ist eine Geschichte künstlerischer Desiderate und Fehlschläge, die sich in den Werken selbst niederschlagen, angefangen bei den Versuchen im Köln der 1950er Jahre, auch komplexe, vierteilige

Klanggemische zu synthetisieren, bis hin zu den zahlreichen Einschränkungen, mit denen Computerprogramme bis heute behaftet sind.

«Many of the synthesis parameters that determine the sound of a given <patch> on a synthesizer are not manipulable, or are hard-linked to other parameters, or have a restricted range, or are otherwise not completely free for the user to determine. Though digital synthesizer architecture has tended increasingly to provide a more detailed user interface, allowing greater control of low-level synthesis variables, most synthesizers hide much of the architecture from the user, making it difficult or impossible to manipulate many crucial parameters of the generated sounds»⁴,

fasst Aden Evens das Problem im Jahre 2005 zusammen und nennt als mögliche Ursachen Kostengründe, die inverse Abhängigkeit zwischen Leistungsfähigkeit und Flexibilität, den Schutz des Verbrauchers sowie marktorientiertes Kalkül.⁵

Kim Cascone, ein in jeder Hinsicht technikkritischer Musiker, der vehement die kreative Zweckentfremdung elektronischer Apparate propagiert, beschreibt eine in diesem Zusammenhang wichtige ästhetische Rückkoppelung:

«Artists download tools and information, develop ideas based on that information, create works reflecting those ideas with the appropriate tools, and then upload that work to a World Wide Web site where other artists can explore the ideas embedded in the work.»⁶

Auch wenn dieser Informationskreislauf und die damit einhergehende historische Dialektik an sich begrüssenswert sind,

stellt sich die Frage, ob die implizite Abhängigkeit der Ideen von den Apparaten nicht auch eine künstlerische Falle sein kann. Muss man nicht, überspitzt formuliert, schlussfolgern, dass der Komponist sich nur das vorzustellen in der Lage ist, wofür die Maschine ohnehin ausgelegt worden ist? Heisst, mit «appropriate tools» zu arbeiten, nicht auch, dass der Komponist nur noch die Logik des Programmierers bestätigt? Im besten Falle wird die Tatsache, dass ein Werk entsteht, weil man sich mit einer Software auseinander setzt, auch ästhetisch reflektiert und Teil des musikalischen Ergebnisses. Das Instrument «sine chords» [Arbeitstitel] des Berliner Komponisten Stefan Streich entstand 2010 beim Erlernen der Software Pure Data. Aus den ersten Übungsaufgaben mit Sinustonoszillatoren heraus ergab sich die Idee eines komplexen Sinustongeflechts, das schliesslich zu einem elementaren, aber durchaus wirkungsvollen Instrument führte, das im wesentlichen aus 24 Sinustongeneratoren besteht, die entweder live oder per Zufallsalgorithmus steuerbar sind. Die seither mit «side chords» realisierten kurzen Klangstudien tragen einerseits Spuren des Erlernens, sofern Streich auf technische Elaborate verzichtet, während sie andererseits infolge ihres ungeschliffenen, lebendigen und organischen Charakters über die blosse technische Situation hinausweisen.

Streichs Sinustonstudien sind eine Ausnahme, denn die Engführung technischer Voraussetzungen und ästhetischer Fragestellungen ist in den vergangenen Jahrzehnten regelrecht tabuisiert worden. Anschauliche Beispiele hierfür liefert der spezifische «Sound» einzelner Studios für elektronische Musik, den zum Beispiel grosse Einrichtungen wie das Pariser Ircam oder das Freiburger Experimentalstudio für sich geltend machen können. Gleichwohl stellen sowohl Komponisten als auch Studioleiter die Existenz eines spezifischen «Sounds» häufig in Abrede, weil seine Existenz offenbar das Gebot der künstlerischen Autonomie gefährden würde. Denn einerseits muss ein Studioleiter seinen Komponisten die grösstmögliche Freiheit gewähren können, die andererseits der Komponist für sich und seine Arbeit wünscht, um sich als autonomes Künstlersubjekt zu positionieren. Pierre Boulez wies die Frage nach dem spezifischen Sound des Ircam mit den Worten «es gibt keinen Studio-Stil»⁷ kategorisch von sich. Nichtsdestotrotz lässt sich ein solcher Studiosound benennen, und selbstkritische und reflektierte Komponisten wie Péter Eötvös («Was noch typischer ist: dieser sogenannte Ircam-Klang. Egal, wer das komponiert hat, nach fünf Sekunden sage ich: Aha, das ist ein Ircam-Komponist.»)⁸ und Michael Obst («Ich erinnere mich daran, dass wir zum Beispiel immer von einem ganz bestimmten Ircam-Sound sprachen, einer charakteristischen Klangfarbe, die sehr, sehr vielen Stücken aus dem Ircam anhaftet.»)⁹ haben denn auch mit Nachdruck auf seine Existenz hingewiesen.

Ähnliches gilt natürlich für musikalische Computeranwendungen. Der Kölner Musiker und Komponist Marcus Schmickler räumt ein, dass «bestimmte Algorithmen auch ihren Sound haben, also dass eine bestimmte Supercollider-Anwendung eben anders klingt als die im Prinzip gleiche Anwendung in einem Max/MSP-Patch.»¹⁰ Der Wiener Laptoptrompeter Christian

Fennesz beschreibt die abstrakten Electronica der späten 1990er Jahre, für die er mit seinem Album *Hotel Parallel* 1997 selbst stilbildend wurde, als ein Genre, bei dem man «nur noch Software gehört» habe: «Ich habe Supercollider gehört und Max/MSP, aber eigentlich keinen Musiker dahinter. Eine Zeit lang war genau das gewollt. Auch ich habe das eine Zeit lang gewollt, dass das Ego völlig verschwindet hinter der Musik.»¹¹ Dabei ist der Wunsch, hinter der Musik und der Maschine zu verschwinden, durchaus als kritischer Reflex des Künstlers zu verstehen, der die Potenz der Maschine qua stilisierter Selbstauslöschung gleichzeitig anerkennt und infrage stellt. Das Verhältnis zwischen dem Menschen und der Maschine ist ambivalent.

Natürlich ist der Befund, dass die Maschinen die Phantasie der Künstler lenken, insofern kein Sonderfall, als fast jeder Lebensbereich von der digitalen Revolution erfasst worden ist. Nicht nur die musikalische Praxis, sondern auch wissenschaftliche Verfahren sind davon betroffen. Naturwissenschaftler sehen einer Arbeit zum Beispiel an, mit welcher Programmiersprache Daten analysiert wurden: Die Logik der Maschine prägt die Erscheinungsformen der Erkenntnis. Gleichermaßen gilt für das wissenschaftliche Schreiben, dessen Duktus sich infolge der Möglichkeiten, die mit Lösch-, Überschreibungs- und Kopievorgängen einhergehen, grundlegend verändert hat. Trotzdem bleibt der Zusammenhang zwischen der Kunst und ihren Maschinen problematisch. Der amerikanische Laptop-Pionier Carl Stone ist keine Ausnahme, wenn er die Hoffnung äussert, «that my music doesn't sound like the software that I use». Der Computer und die industriellen Standards der Maschine gefährden die Autonomie des Künstlers in einem Masse, in dem dies die Musikinstrumente vergangener Jahrhunderte nicht taten.

2

«Die Architektur von MAX ist derart offen, dass damit nahezu sämtliche interaktiven Aufgabenstellungen programmiert werden können und viele bereits vor der Einführung von MAX existierende Programme durch einen entsprechenden Einsatz von MAX ersetzt werden können», lobt Martin Supper die Flexibilität und Leistungsfähigkeit der Software Max/MSP.¹² Und Ian Willock spricht von einem System «capable of enormous flexibility».«¹³ Die 1986 von Miller Puckette am Ircam entwickelte Programmiersprache gilt als Idealfall einer den Bedürfnissen der Musiker anpassbaren Anwendung. Nicht nur lässt sich fast jede übliche Routine mit Max/MSP realisieren. Aufgrund der graphischen Darstellung in Max/MSP, bei der die einzelnen Objekte wie in einem Logikdiagramm auf einer einem Blatt gleichenden «Leinwand» (Canvas) abgebildet und mit Kabeln gleichenden «Strichen» verbunden werden, ist der Zugang auch dann möglich, wenn man die Syntax einer Programmiersprache nicht beherrscht. Wie vielseitig Max/MSP tatsächlich ist, lässt sich vielleicht daran ablesen, dass nicht nur das Gros der Laptop-Musiker heute – vor allem im Live-Konzert – mit einem Max/MSP-Patch arbeitet, sondern dass sich zum

Beispiel auch die von Luigi Nono ursprünglich am Halaphon realisierten Raumklangbewegungen seiner späten live-elektronischen Werke in Max/MSP darstellen und realisieren lassen. Im Gegensatz zu vielen kommerziellen Applikationen ermöglicht Max/MSP den Zugriff auf «low-level synthesis variables»,¹⁴ sodass sich alle darstellbaren Parameter der Klangentstehung und -bearbeitung steuern lassen.

Um mit anderen Geräten und Instrumenten zu kommunizieren, nutzt Max/MSP den MIDI-Standard (Musical Instrument Digital Interface). Der MIDI-Standard hat sich seit seiner Einführung im Jahr 1983 weithin durchgesetzt. Er verschlüsselt akustische Signale anhand spezifischer Parameter, die den Datenaustausch zwischen elektronischen Musikinstrumenten erleichtern, sodass sich ein Sampler zum Beispiel leicht mit einer Tastatur spielen lässt. Mit anderen Worten: MIDI-Signale transportieren Steuerdaten, die einem elektronischen Klang-erzeuger wie einem Synthesizer, Sampler oder Drumcomputer mitteilen, welchen Ton er wie laut, wie lange und mit welchem seiner verfügbaren Sounds spielen soll. Den einzelnen Parametern (Tonhöhe, Lautstärke, Dauer etc.) werden dabei Werte auf einer Skala von 1 bis 128 zugeordnet.

Entwickelt wurde MIDI von der amerikanischen Audio Engineering Society. Verwaltet wird der Standard von der MIDI Manufacturers Association, einem Industrieverband, dem Mitglieder internationaler Firmen wie Yamaha, Korg und Creative Labs vorstehen.¹⁵ Die industrielle Ausrichtung des Standards ist insofern sinnvoll, als er zunächst die Kompatibilität kommerzieller Geräte sichern soll. Diese Ausrichtung bedeutet aber auch, dass sich der Standard im wesentlichen an den Bedürfnissen der Hersteller und Nutzer von Unterhaltungselektronik orientiert. Die Auflösung der Werte auf einer Skala von 1 bis 128 ist für ein komplexeres und differenzierteres Klangereignis hingegen kaum hinreichend. «Although the introduction of MIDI in the early 1980s resulted in explosive growth for the synthesizer industry», beklagt Nicolas Collins diese Entwicklung, «these new machines were purposefully generic, the musical equivalent of the putty-colored office PC.»¹⁶ MIDI ist vielleicht so etwas wie der Sündenfall der elektronischen Musik, wenn man bedenkt, dass, wie David Behrman rückblickend schreibt, es in den Anfangsjahren keinen Unterschied zwischen «high tech» und «low tech» gegeben habe, sondern alle am Entwicklungsprozess Beteiligten im gemeinsamen technisch-musikalischen Interesse gearbeitet haben:

«In the old days there wasn't any distinction between high tech and low tech. The early analog synths were made by creative individuals like Bob Moog and Don Buchla; even the early microcomputers were mostly made by garage start-ups and there wasn't so much difference between these and the craft shops that had made lutes, guitars, or violins for centuries. [...] Only in recent decades have music instruments and software become corporate, mostly mass-produced and mass-marketed, and only recently are the computers used for music generally the same ones found in tens of millions of business establishments.»¹⁷

Die flexible Struktur von Max/MSP wird bei der Kommunikation mit anderen Anwendungen und Geräten durch den MIDI-Standard empfindlich gestört, wenn nicht gar hinfällig. Es führt vor allem dazu, dass sich viele Parameter nicht differenziert darstellen lassen, was infolge des groben Rasters, das der MIDI-Verschlüsselung zugrunde liegt, vor allem die Tonhöhe betrifft. Natürlich kann man mit einem MIDI-Signal jedes beliebige Klangereignis triggern, aber eine tonhöhenfixierte Struktur in Max lässt sich außerhalb der temperierten Norm nur sehr mühevoll exportieren.

Das Problem des MIDI-Standards ist ontologischer und wahrnehmungstheoretischer Natur. Jedes Ereignis wird im Zuge seiner Digitalisierung in diskrete Werte zerlegt. Die Darstellungsmöglichkeiten sind mithin begrenzt. «There is always another note between two notes, another dynamic value between two values», beschreibt Aden Evens das Problem und fährt fort:

«Though many electronic instruments make distinctions more subtle than the MIDI standard, every digital instrument stops subdividing at some point, as that is the nature of the digital. In other words, practical limits of computation power, timing accuracy, instrument sensitivity, data storage, and potentially more restrictive limits imposed by design and material costs require that a digital instrument ultimately assigns each parameter a particular value at a finite level of subtlety.»¹⁸

Ein ontologisches Problem ist der hier geschilderte Sachverhalt also, weil die Reduktionsverfahren des MIDI-Standards in der Natur des Digitalen liegen. (Ontologisch ist er aber auch, weil wir streng genommen nicht davon ausgehen können, dass es Kontinua tatsächlich gibt bzw. auch Zeit und Raum gemäß der heisenbergschen Unschärferelation diskrete Größen sind. «Wir wissen nicht, ob Kontinuität überhaupt existiert, aber in unseren Köpfen spielt sie eine wichtige Rolle», erläutert Erik Oña diesen Sachverhalt.¹⁹) Wahrnehmungstheoretischer Natur ist das Problem, weil schliesslich nicht die tatsächliche Auflösung des Tonhöhenraums entscheidend ist, sondern das, was der Hörer überhaupt wahrzunehmen in der Lage ist. Die mögliche Auflösung eines 64-Bit-Computers ermöglicht da viel feinere Intervalle als der Schwellenwert der Intervallwahrnehmung des menschlichen Ohres, während die Halbtöne selbstverständlich, aber auch Achteltöne die Kontinuitätsanforderung noch verletzen.

EXKURS: STIMMUNGSSYSTEME IN DER ELEKTRONISCHEN MUSIK

Die Frage nach dem Stimmungssystem ist in der elektronischen Musik selten gestellt worden. In der frühen elektronischen Musik Kölner Prägung galt die Tonhöhe noch als zentraler Parameter. Die Elektrifizierung der seriellen Praxis führte zu einer Organisation von Tonhöhen und Spektren, die außerhalb sowohl der von Partialtönen ausgehenden, als auch der

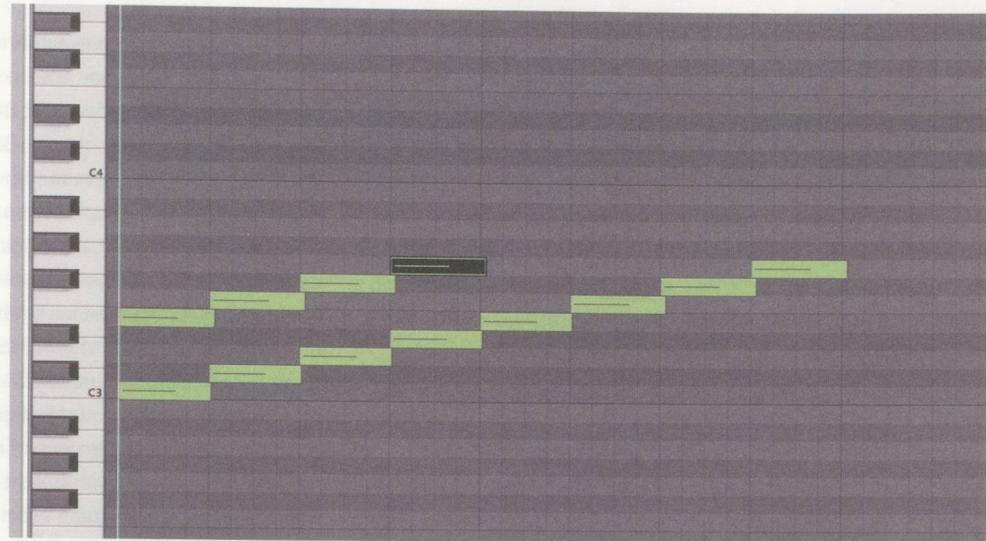


Abbildung 1: Möglichkeiten der Tonerzeugung in einer herkömmlichen Sequenzer-Software. Der MIDI-Standard und die ästhetische Ausrichtung des Programms verhindern die Modifikation der von der Klaviatur und der temperierten Stimmung vorgegebenen Tonhöhen. Mikrotonales Komponieren ist in einer solchen Umgebung nicht möglich.

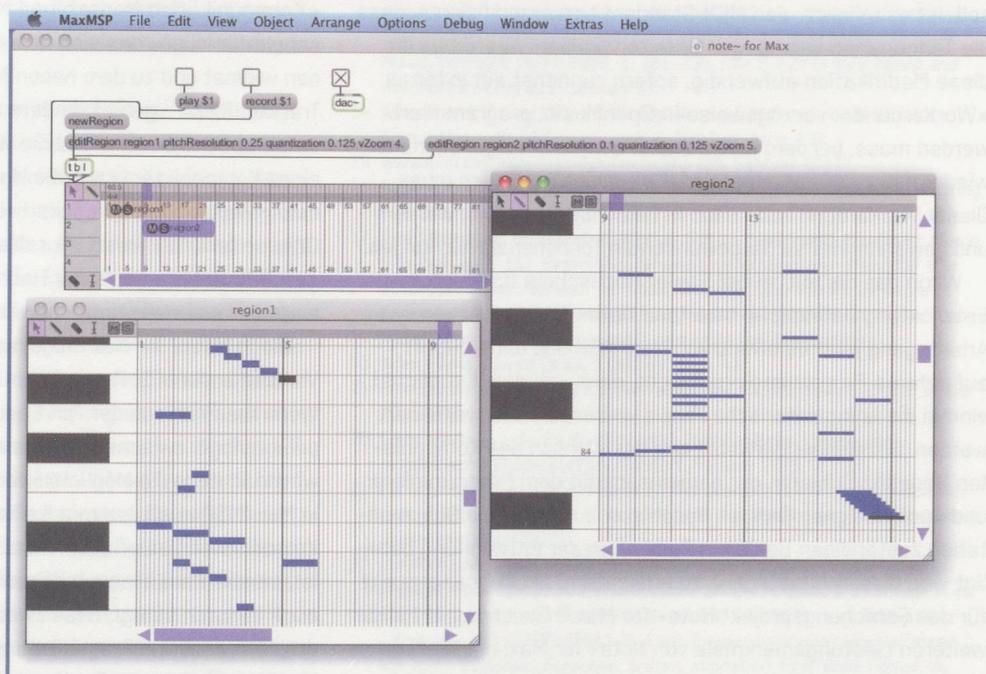


Abbildung 2: Ein ähnliches Projekt in Note~ for Max. Hier lassen sich Tonhöhen auch zwischen den temperierten Werten festlegen. Die Restriktionen des MIDI-Standards werden dabei in der Kommunikation mit externen Geräten qua Fliesskommazahl überwunden.

temperierten Stimmung liegen. Die Prämissen der additiven Synthese, die unterstellen, dass sich in Anlehnung an das Fourier-Theorem jedes beliebige Schallereignis aus einer Reihe von Sinustönen zusammensetzen lässt, zeitigen neue Ordnungsprinzipien der Tonhöhen, die die geläufigen Stimmungstraditionen ausser Kraft setzen.²⁰ Diese Prämissen verloren in den folgenden Jahrzehnten in dem Mass an Bedeutung, in dem die additive Synthese von anderen Syntheseverfahren abgelöst wurde. Die Auflösung des herkömmlichen Tonhöhenrasters wurde stattdessen vor allem in der Instrumentalmusik genutzt, wo die Arbeit mit natürlichen Spektren nicht zuletzt von den Komponisten des Groupe L'Itinéraire unter dem Schlagwort «Musique spectrale» eine neuartige Ausprägung erfuhr. Georg Friedrich Haas arbeitet sowohl mit natürlichen Obertonreihen als auch mit der von Ivan Wyschnegradsky

entwickelten Sechstelton-Systematik.²¹ Aber auch künstliche Spektren sind in der Instrumentalmusik realisiert worden, etwa in Werken von Enno Poppe, der Akkorde auf der Basis der Addition und Subtraktion zweier Frequenzen schuf – in Anlehnung an die Funktionsweise des Ringmodulators.²² Komponisten elektronischer Musik hingegen haben sich zunächst vor allem der Klangfarbe als zentraler Gestaltungsgröße gewidmet, deren Eigenschaften durch Sample- und Modulationsverfahren, aber auch durch das Physical Modeling²³ einschneidende Veränderungen erfuhren. So blieb der Parameter Tonhöhe in der elektronischen Musik verhältnismässig unentwickelt. Leistungsfähige und auch in der avancierten Elektronik häufig verwendete kommerzielle Produkte wie Ableton Live und Logic Audio sehen keine differenziertere Bearbeitung der Tonhöhe vor, sondern stellen dem Anwender eine graphische

Klaviatur bereit, mit der ausschliesslich Töne der temporierten Stimmung erzeugt werden können – ein übler Anachronismus, wenn man bedenkt, mit welchem Aufwand Karlheinz Stockhausen 1954 alle mit einer Klaviatur versehenen elektronischen Apparate, darunter das Trautonium und das Bode-Melochord, aus dem Kölner Studio für elektronische Musik entfernen liess. Dass auch Max/MSP aufgrund der Kodierung der Klangereignisse im MIDI-Standard keine mikrotonalen Modifikationen vorsieht, macht deutlich, dass Stimmungssysteme und Mikrotonalität ein blinder Fleck der Computermusik sind.

3

Natürlich ist es möglich, auch in Max/MSP mikrotonal zu arbeiten. Zum einen lassen sich die Frequenzwerte eines Oszillators innerhalb von Max/MSP im Dezimalbereich modifizieren. Und wenn ein mikrotonal gearbeitetes Objekt exportiert werden soll, ist es möglich, den MIDI-Standard so zu modifizieren, dass die Tonhöhenabweichungen erhalten bleiben. Allerdings ist diese Modifikation aufwendig, sofern zunächst ein externer «Workaround», vorzugsweise in OpenMusic, programmiert werden muss, bei dem jedem einzelnen Ton ein eigener, immer wieder neu zu stimmender Kanal zugewiesen werden muss. Dieser Workaround kann dann in Max/MSP importiert werden und gewährleistet so die gewünschten Tonhöhenabweichungen.

Wenn das derzeit an der Basler Hochschule für Musik in Entwicklung befindliche Interface Note~ for Max diesen Arbeitsgang jetzt innerhalb von Max/MSP und ohne Zugriff auf externe Programme ermöglicht, dann soll damit zunächst einmal die kompositorische Praxis verbessert bzw. verfeinert werden. Bezeichnenderweise war der Wunsch des Komponisten Benedikt Schiefer, der gemeinsam mit dem Programmierer und Audiodesigner Thomas Resch sowie dem für die Dokumentation zuständigen Benjamin Federer an der Entwicklung beteiligt war, diesen Arbeitsgang zu erleichtern, ein Ausgangspunkt für das Forschungsprojekt Note~ for Max.²⁴ Gleichermaßen gilt für die weiteren Leistungsmerkmale von Note~ for Max, die in erster Linie das Sequencing und die Notation betreffen. Max/MSP arbeitet mit dem Konzept eines Logikdiagramms; die Idee des «Patches» hängt eng mit der Idee des Steckfeldes und der Verknüpfung der Module am Synthesizer zusammen. Das legt dem Benutzer nicht nur eine gewisse Struktur nahe, sondern es erschwert auch die Lesbarkeit zeitlicher Verläufe. Die Darstellung der Prozesse in einem x/y-Koordinatensystem mit den Dimensionen Zeit (x-Achse) und Spur (y-Achse), mit der die meisten Sequenzer arbeiten und die Note~ jetzt auch in Max/MSP ermöglicht, verbessert den Nachvollzug zeitlicher Verläufe erheblich. Ähnliches gilt für die Notationsverfahren, die in Anlehnung an das geläufige MusicXML-Format entwickelt werden und zum Beispiel Tonhöhenabweichungen qua Cent-Nennung kennzeichnen. Note~, ein Akronym für «Notation of Timed Events», ist derzeit im Stadium einer Beta-Version und ist nur im Bereich der Notation noch nicht vollständig entwickelt.²⁵ Es lassen sich mit Note~ alle denkbaren Klangobjekte in Max/MSP wie mit einem Sequenzer verwalten und bearbeiten,

was auch in Max/MSP sonst nur schwer zu realisierende Aufnahme- und Wiedergabeverfahren erleichtert. Da die Bearbeitungsoptionen sich aber eben nicht auf den MIDI-Standard beschränken, geht Note~ weit über die Möglichkeiten gängiger Sequenzerprogramme hinaus. Gleichzeitig lassen sich in Note~ auch Klangereignisse herstellen. So ist es zum Beispiel möglich, mithilfe einer graphischen Klaviatur einzelne Töne zu erzeugen, die dann mit beliebigen Frequenzwerten und Dauern versehen werden können. Note~ ist tatsächlich so etwas wie eine Universalmaschine innerhalb von Max/MSP (siehe Abb. 1 und 2). Erik Oña spricht vom Konzept der «ästhetischen Neutralität», das man der Software zugrunde gelegt habe.²⁶ Und auch wenn sich vollständige Neutralität angesichts nicht ganz auszuschaltender persönlicher Vorlieben der am Projekt beteiligten Personen wohl nicht realisieren lässt, muss man Note~ zugute halten, dass es in einem weitgehend von wirtschaftlichen und privaten Interessen befreiten Umfeld realisiert wurde. Note~ for Max ist Teil des Forschungsfeldes «Kommunikation Mensch und Maschine» der Basler Hochschule für Musik, das sich dem intuitiven Umgang mit Maschinen widmet und zu dem neben Note~ auch der Multi Action Tracker IRMAT²⁷ gehört. Anders als die profitorientierte Zielsetzung der Industrie und die werkorientierte Zielsetzung eines Komponisten, kann die Hochschulforschung tatsächlich eine gewisse Unabhängigkeit für sich in Anspruch nehmen. Oder anders formuliert: Nur die von äusseren Interessen geschützte Situation einer Hochschule kann die Universalität einer Software wie Note~ gewährleisten.

Die in Note~ for Max möglichen Verfahren integrieren die Vorteile anderer Softwaretypen in die Max/MSP-Umgebung. Wenn man nun von der von Cascone beschriebenen Rückkopplungslogik zwischen Instrument und Komponist ausgeht, wird klar, dass die Möglichkeiten mikrotonalen Komponierens in Max/MSP auch den musikalischen Vorstellungshorizont des Komponisten beeinflussen. Dass Note~ for Max keine Spielerei ist, sondern aus einem triftigen Bedürfnis hervorgeht, wird auch dadurch belegt, dass zwei andere aktuelle Forschungsprojekte ähnliche Fragestellungen aufgreifen: Sowohl MaxScore²⁸ als auch Bachproject²⁹ befassen sich primär mit Fragen der Notation und der Ausführbarkeit von Partituren, die für Note~ for Max ebenfalls von zentraler Bedeutung sind.

4

Der amerikanische Komponist Bob Ostertag ist kein Freund der Computermusik: «It appears that the more technology is thrown at the problem, the more boring the result.»³⁰ Und Kim Cascone pflichtet ihm bei: «Sometimes, not knowing the theoretical operation of a tool can result in more interesting results by <thinking outside of the box>.»³¹ Ostertag befürchtet, dass die Technokratie die spielerische Freiheit des Komponisten bedroht; Cascone fordert, dass man als Komponist ausserhalb des technischen Zusammenhangs zu denken habe. Wenn Note~ for Max Vorteile externer Programme in Max/MSP implementiert, wird das Ausmass an Technik «thrown at the

problem» nicht geringer, wodurch die Musik zwangsläufig langweiliger werden müsste. Gleichzeitig importiert Note- for Max Strategien und Denkmöglichkeiten von «outside of the box», wodurch Resultate interessanter werden können.

Natürlich lässt sich nicht vorhersagen, ob die elektronische Musik infolge von Note- for Max insgesamt besser oder schlechter wird. Es ist bisweilen der Verdacht geäussert worden, dass technische Erleichterungen der Kunst nicht zugute kommen, sondern dass im Gegenteil (es ist eine Binsenweisheit) Not erforderlich mache. Erik Oña, der das Note- for Max Projekt als Leiter des Studios für Elektronische Musik an der Hochschule für Musik Basel betreut, erwidert, dass die einzige Grenze, die zählt, «unsere Vorstellung» ist.³² Wenn Oña darauf hinweist, dass eine der primären Vorgaben für Note- die «ästhetische Neutralität» gewesen ist, dann stellt er damit in Aussicht, dass sich die Software der Phantasie des Komponisten andient und sie nicht in eine bestimmte Richtung lenkt. Man könnte im Umkehrschluss natürlich auch argumentieren, dass es jedem Künstler frei steht, auf technische Errungenschaften zu verzichten, um seiner Kreativität möglichst enge Grenzen zu setzen.

Der entscheidende Punkt ist, dass Technik nie unschuldig ist. In seinem Rückblick auf die Pionierzeit der 1970er Jahre in der kalifornischen Bay Area bezichtigt sich Tim Perkis der Naivität, weil er glaubte, «dass wir die Technik für einen anderen Zweck nutzen könnten, dass das die bevorstehende grundlegende Aufgabe wäre, dass wir die Bedeutung der Technik selbst schärfen könnten», wobei man «die ungeheure Macht der Technik» schlachtweg unterschätzt habe: «Künstler, insbesondere technisch orientierte Künstler, laufen Gefahr, Diener einer Macht jenseits menschlicher Führung zu werden, eine Situation, die ich <kulturelle Technokratie> nennen möchte.»³³ Note-, so könnte man es vielleicht formulieren, versucht eben jene Dienerschaft aufzuheben, indem sie dem Komponisten die Verwirklichung einer musikalischen Idee ermöglicht. Wenn Perkis davon spricht, dass «Technologien Verstärker unserer schon existierenden Visionen und inneren Orientierungen»³⁴ sein sollten, dann ist Note- genau solch ein Verstärker, dessen tatsächliche Wirksamkeit sich dann allerdings in den mit ihm realisierten Werken zeigen wird.

use those sounds. Sometimes the design limitations exist to protect the user from useless or bad sounding patches by limiting parameter manipulation to a range deemed broadly useful. [...] Sometimes the limitations of a synthesizer design are deliberate and extraneous, to cripple the synth in order to generate demand for more powerful synthesizers at higher price points.» (ebd.).

- 6 Kim Cascone, *The Aesthetics of Failure: 'Post-Digital' Tendencies in Contemporary Computer Music*, in: Christoph Cox, Daniel Warner (Hrsg.), *Audio Culture. Readings in Modern Music*, New York: Continuum 2004, S. 392-398, hier S. 397 (Erstpublikation in: *Computer Music Journal* 24/4, 2000, S. 12-18).
- 7 Pierre Boulez im Gespräch mit Kornelia Bittmann am 8. Februar 2000 in Köln.
- 8 Péter Eötvös im Gespräch mit Kornelia Bittmann am 30. Januar 2001 in Köln.
- 9 Michael Obst im Gespräch mit dem Verfasser am 5. Dezember 2000 in Weimar.
- 10 Marcus Schmickler im Gespräch mit dem Verfasser am 16. Juli 2002 in Köln.
- 11 Christian Fennesz im Gespräch mit dem Verfasser in Wien am 10. März 2003 in Köln.
- 12 Martin Supper, *Elektroakustische Musik und Computermusik. Geschichte - Ästhetik - Methoden - Systeme*, Hofheim: Wolke 1997, S. 90f.
- 13 Ian Willock, *Composing without Composers? Creation, Control, and Individuality in Computer-Based Algorithmic Composition*, in: Frank Cox, Claus-Steffen Mahnkopf, Wolfram Schurig (Hrsg.), *Electronics in New Music*, Hofheim: Wolke 2006, S. 221-235, hier S. 229 (= *New Music and Aesthetics in the 21st Century* 4).
- 14 Aden Evans, *Sound Ideas*, S. 88 (vgl. Anm. 4). Siehe auch Ian Willock, *Composing without Composers?*, S. 229 (vgl. Anm. 13).
- 15 www.midi.org/aboutus/aboutmma.php (25. Februar 2011).
- 16 Nicolas Collins, *Handmade Electronic Music. The Art of Hardware Hacking*, New York: Routledge 2006, S. 192.
- 17 David Behrman, *Foreword*, in: Nicolas Collins, *Handmade Electronic Music*, S. ix (vgl. Anm. 16). Bezeichnenderweise nimmt Behrman Max/MSP hier ausdrücklich vom Vorwurf der Kommerzialität aus: «Some software packages like Max/MSP are not really corporate products in the bad sense and they are infinitely personable and endlessly fascinating.»
- 18 Aden Evans, *Sound Ideas*, S. 88 (vgl. Anm. 4).
- 19 Erik Oña im Gespräch mit dem Verfasser und Thomas Resch am 20. Januar 2011 in Basel.
- 20 Siehe hierzu u.a. André Ruschkowski, *Elektronische Klänge und musikalische Entdeckungen*, Stuttgart: Reclam 1998, S. 232f.
- 21 Mit Obertonreihen u.a. im *Ersten Streichquartett* (1997) und in *Nach-Ruf ... ent-gleitend* (1999), mit Sechstel- und Zwölfteltönen zuletzt in *limited approximations* (2010).
- 22 Zur Funktionsweise des Ringmodulators vgl. u. a. André Ruschkowski, *Elektronische Klänge und musikalische Entdeckungen*, S. 171f (vgl. Anm. 20). Massgeblich wurde das Verfahren u.a. in Enno Poppes Zyklus *Holz - Knochen - Öl* (1999-2004). Zu Enno Poppes harmonischen Verfahren vgl. Björn Gottstein, *Forschen, brüten, schreiben. Über Enno Poppe*, in: Katalog Wien Modern 2008.
- 23 Zum Physical Modeling vgl. nochmals André Ruschkowski, *Elektronische Klänge und musikalische Entdeckungen*, S. 332-336 (vgl. Anm. 20).
- 24 Das Forschungsprojekt Note- for Max wird finanziert durch die Maja Sacher Stiftung, Basel.
- 25 Die Beta-Version soll demnächst unter <http://noteformax.net/index.php?/download/> zum Download bereitgestellt werden und steht dann allen Max-Nutzern zur Erkundung zur Verfügung (Stand 2. März 2011).
- 26 Vgl. Anm. 19.
- 27 www.musikforschungbasel.ch/fe/index.php/neuemusik/infrared-multi-action-tracker-irmat (25. Februar 2011).
- 28 www.algomusic.com/maxscore (25. Februar 2011).
- 29 www.bachproject.net (25. Februar 2011).
- 30 Bob Ostertag, *Why Computer Music Sucks*, <http://bobostertag.com/writings-articles-computer-music-sucks.htm> (25. Februar 2011). Zitiert nach: Kim Cascone, *The Aesthetics of Failure*, S. 397 (vgl. Anm. 6).
- 31 Kim Cascone, *The Aesthetics of Failure*, S. 397 (vgl. Anm. 6).
- 32 Erik Oña im Gespräch mit dem Verfasser und Thomas Resch am 20. Januar 2011 in Basel.
- 33 Tim Perkis, *Volksmusiker im Silicon Valley. Computertechnologien als Verstärker von Visionen*, in: *Neue Zeitschrift für Musik* 2008/1, S. 42-45, hier S. 45.
- 34 Ebd.