

„Computational Creativity“ in der Komposition

Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung,
dem Landesprüfungsamt für Erste Staatsprüfungen
für Lehrämter an Schulen (Außenstelle Köln)

vorgelegt von

Nikolaus Rademacher

Köln, Februar 2016

Gutachter: Prof. Dr. Hans Neuhoff

Institut: Hochschule für Musik und Tanz Köln

Fachbereich: 5 (Musikwissenschaft, Musikpädagogik,
Kirchenmusik, Chorleitung)

Inhaltsverzeichnis

„Computational Creativity“ in der Komposition

1 Einleitung	4
2 Kreativitätsforschung	7
2.1 Überblick über qualitative Modelle	10
2.1.1 Vier-Phasen-Modelle	10
2.1.2 Divergent statt konvergent	12
2.1.3 Bisoziation statt Assoziation	12
2.1.4 Im Flow	14
2.2 Computational Creativity als quantitatives Modell	16
2.2.1 Theoretische Grundlagen	19
2.2.2 Information Dynamics of Thinking (IDyOT)	21
3 Methoden der algorithmischen Komposition	23
3.1 Zufall	23
3.2 Wissens- und regelbasierte Methoden	25
3.2.1 Grammars	25
3.2.2 Regel-lernende Programme	26
3.2.3 Constraint-Satisfaction	27
3.2.4 Case-based reasoning	28
3.3 Markov-Ketten	28
3.4 Künstliche neuronale Netze	29
3.5 Evolutionäre Algorithmen	31
3.6 Weitere Methoden	33
3.6.1 Fuzzy Logic	33
3.6.2 Sonification	34
3.7 Bewertung	35

4 Beispiele zu Computational Creativity in der Komposition	37
4.1 Emmy und Emily	
David Cope's Computer Models of Musical Creativity	38
4.1.1 Emmy & die Rekombination von Musik	38
4.1.2 Emily & (induktive) Assoziations-Netze	41
4.2 Spiel es noch einmal, Sam!	
Das Sony Computer Science Laboratory	49
4.2.1 Der Continuator	49
4.2.2 Markov-Constraints	51
4.2.3 Stil als zentrale kreative Eigenschaft	52
4.3 Der Sohn des Apollo	
Melomics	55
4.3.1 Iamus	55
4.3.2 Evo-Devo als Kompositionsalgorithmus	56
4.3.3 Melomics heute	58
5 Einschätzung und Fazit	60
Quellen- und Literaturverzeichnis	63
Erklärung	68

1 Einleitung

Ein Konzert zu Ehren des Computerpioniers Alan Turing sorgte 2012 für großes Aufsehen: Das erste mal in der Geschichte der Menschheit sei eine Komposition gänzlich ohne menschliche Hilfe komponiert worden. Iamus, der Computercluster, der in der Fachpresse als die „Mozart-Maschine“ bezeichnet wird, komponiert auf Knopfdruck Musik.

Die grundlegende Idee automatisch generierter Musik ist nichts Neues: Schon Wolfgang Amadeus Mozart erstellte ein musikalisches Würfelspiel mit dem Titel „Anleitung so viel Walzer oder Schleifer mit zwei Würfeln zu componiren so viel man will ohne musikalisch zu seyn noch etwas von der Composition zu verstehen“. Das Prinzip: Vorbereitete Teilstücke, die nach Belieben miteinander kombiniert werden können. Über eine Tabelle lassen sich die gewürfelten Augen bestimmten Teilstücken zuordnen und man würfelt so lange, bis das Stück zu Ende ist.¹ Natürlich handelt es sich hierbei nicht um gänzlich automatisierte Musik, sondern nur um die Rekombination existierender Teilstücke.

Der ehemalige Musikwissenschaftler und selbst ernannte Computer-Wissenschaftler David Cope entwickelt mehrere Programme, die das Prinzip der Rekombination verfeinern: Cope glaubt, dass jeder Komposition eine Anleitung innewohne, ähnliche Kompositionen in diesem Stil zu schreiben. Seine Software, die er liebevoll Emmy nennt, extrahiert den Stil einer Komposition und erstellt daraus neue Werke. Techniken der künstlichen Intelligenz werden hierbei zur Analyse der existierenden Software verwendet. Cope behauptet, dass dieser künstliche Vorgang mit menschlicher Kreativität vergleichbar sei.

¹ Eine digitale Version dieses Würfelspiels gibt es beispielsweise hier zu finden: <http://www.amaranthpublishing.com/MozartDiceGame.htm> (abgerufen am: 27.12.2015).

Auch ein französisches Forschungsteam beschäftigt sich mit der Replikation von Stil und kommt dabei zu erstaunlichen Ergebnissen: Ein Charlie-Parker-Blues, kombiniert mit Kompositionstechniken von Pierre Boulez, ein Jazz-Standard in einer Wagner-typischen Harmonisierung oder der Liedtext zu Beatles „Yesterday“ in einer Version von Bob Dylan – all das produziert von Computer-Software. Das „Flow Machines“-Projekt, das diese Ergebnisse hervorbrachte, hat das selbstgesteckte Ziel, menschliche Kreativität zu erweitern, um den „Flow“-Zustand nach Csikszentmihalyi² aufrecht zu erhalten.

Emmy, Iamus und die „Flow Machines“ fallen alle in das Gebiet der „Computational Creativity“: einer wissenschaftlichen Disziplin, die sich als Schnittstelle zwischen künstlicher Intelligenz, kognitiver Psychologie, Philosophie und den Künsten versteht. Doch was ist überhaupt Kreativität? Und lässt sich menschliche Kreativität tatsächlich mit Computern simulieren? Der humanoide Roboter „Myon“ feierte im Juni 2015 sein Debüt als Opern-Darsteller an der Komischen Oper Berlin. Sein Erfinder Manfred Hild erklärt in der ZEIT:

„Intelligenz baut sich langsam auf [...] Intelligenz ist die Fähigkeit, der Situation angemessen zu handeln. Geschafft haben wir das bei Myon, wenn er bei anderen Intentionen erkennen und selbstständig darauf reagieren kann.“³

Das Ziel der Kooperation mit der Komischen Oper Berlin ist es, dem Roboter über die Musik beizubringen, was Emotion ist.⁴ Erst dann könne der Roboter wirklich fühlen. Ansonsten handle er nur oberflächlich und seelenlos. Schließlich solle es keine Maschine sein, die etwas in der Verlängerung ausführt, sondern ein Individuum, das „in dem Moment die Entscheidung trifft.“⁵

² Vgl. Kapitel 2.1.4.

³ HILD zit. n. PIRICH, Ein Roboter lernt fühlen.

⁴ Vgl. ebd.

⁵ PIRICH, Können Roboter fühlen?

Strömungen innerhalb der künstlichen Intelligenzforschung glauben an den Zeitpunkt der Singularität: Da sich die Rechenleistung von Prozessoren alle zwei Jahre verdoppelt, gebe es irgendwann den Zeitpunkt, an dem die Maschinen die Rechenleistung des menschlichen Gehirns erreicht haben.⁶ Diese Behauptung ist nicht ganz unumstritten: Die Frage ist, ob die Architektur eines Prozessors tatsächlich mit der Struktur eines menschlichen Gehirns gleichzusetzen ist. In der KI-Forschung werden neuronale Netze imitiert, um Probleme wie die Gesichts- oder Spracherkennung zu lösen. Solche Mechanismen stecken heutzutage in jedem Smartphone und sind mal mehr, mal weniger erfolgreich.

Alan Turing, dem zu Ehren das eingangs erwähnte Konzert stattfand, ist unter anderem für den von ihm vorgeschlagenen Test von künstlicher Intelligenz bekannt: Beim Turing-Test unterhält sich ein Fragesteller mittels Bildschirm und Tastatur sowohl mit einer menschlichen Person als auch mit einer Maschine. Der Turing-Test gilt für die Maschine als bestanden, wenn der menschliche Fragesteller nicht ausmachen kann, welcher der beiden Gesprächspartner die Maschine ist.⁷

Das zweite Kapitel soll einen Überblick über die aktuelle Forschung zu Kreativität liefern. Der Fokus steht dabei auf Modellen, die den Kreativitätsprozess erklären. Es wird anschließend betrachtet, inwieweit die Computational Creativity diesen Modellen entspricht. Das dritte Kapitel erklärt die gängigsten Methoden der künstlichen Intelligenz, die in der algorithmischen Komposition verwendet werden. Im vierten Kapitel werden drei verschiedene Beispiele für die Computational Creativity in der Komposition genannt. Die Auswahl folgt anhand der gefühlten wissenschaftlichen Relevanz, die sich aus der Forschung zu diesem Thema ergab. Das fünfte Kapitel liefert ein abschließendes Fazit.

⁶ Vgl. KURZWEIL, *The Singularity Is Near*, 22f sowie 135f.

⁷ Vgl. TURING, *Computing Machinery and Intelligence*.

2 Kreativitätsforschung

Ähnlich wie die Intelligenz ist auch die Kreativität, obwohl die Fähigkeit zur selbigen gemeinhin als eine der grundlegenden menschlichen Eigenschaften angesehen wird, noch wenig erforscht. Empirisch lassen sich genau wie bei der Intelligenzforschung bisher nur die Resultate, nicht jedoch der Entstehungsprozess festhalten. Über letzteren werden verschiedene Vermutungen angestellt, die sich mal besser, mal schlechter belegen lassen. Existierende Modelle beziehen sich meistens auf wissenschaftliche Teilgebiete: So gibt es neurowissenschaftliche, psychologische oder philosophische sowie neuerdings computergestützte Erklärungsansätze des Phänomens Kreativität.

Im Umfeld von Hobby-Zeitschriften wird Kreativität gerne als Synonym für den englischen Begriff *craftsmanship* gebraucht, der sich ausschließlich auf handwerkliche Fertigkeiten bezieht. Von diesem Kreativitätsbegriff möchte ich in dieser Arbeit absehen. Kreativität wird häufig als etwas Wundersames, Mythisches beschrieben – die Antike brachte gar Schutzgöttinnen hervor, die als Musen den entscheidenden kreativen Impuls geben sollten, weil man sich die Entstehung von etwas Neuem *ex nihilo* nicht erklären konnte. Die Wissenschaft ist da etwas nüchterner und versucht, die wenigen existierenden Anhaltspunkte zu einem Modell zusammenzusetzen. Was macht also Kreativität aus? In der Forschung werden häufig die folgenden drei Voraussetzungen genannt, damit eine Idee oder ein Kunstwerk als kreativ bezeichnet werden kann:

Neuartigkeit: Das lateinische Wort *creare* lässt sich mit „etwas Neues schaffen“ oder „etwas ins Leben holen“ übersetzen. Auch bei Kreativität handelt es sich in der Regel um etwas Neues – ist eine Idee schon einmal da gewesen, würde sie wohl kaum noch als kreativ bezeichnet. So definiert auch Margaret Boden, die sich am Department of Informatics der University of Sussex mit Kognitionswissenschaften beschäftigt, Ideen oder

Kunstwerke als kreativ, die „new, surprising and valuable“⁸ sind. In einem Definitionsversuch des Kreativitätsforschers Siegfried Preiser spielt die Neuartigkeit ebenfalls eine zentrale Rolle:

„Eine Idee wird in einem sozialen System als kreativ akzeptiert, wenn sie in einer bestimmten Situation neu ist oder neuartige Elemente enthält und wenn ein sinnvoller Beitrag zu einer Problemlösung gesehen wird.“⁹

Die Einstufung eines Produktes als kreativ ist nach dieser Definition aber auch abhängig vom Betrachtungswinkel des Rezipienten und Kreativität damit kein objektiver Maßstab. Da, so Preiser, zahlreiche Erfindungen unabhängig voneinander gemacht worden seien, könne der Zeitpunkt einer Erfindung aber nicht ausschlaggebend sein – bei all diesen Erfindungen sei schließlich auf kognitiver Ebene ähnliches geleistet worden, vorausgesetzt der Erfinder hatte keine Kenntnis über die simultan stattfindende Entwicklung seiner Idee. Preiser differenziert daher genauso wie Boden zwischen einer psychologischen und einer historischen Betrachtung von Kreativität^{10,11}: Als *P-creativity* bezeichnet Boden also die Entstehung von Ideen und Kunstwerken im psychologisch-subjektiven Kontext, als *H-creativity* hingegen solche im historisch-objektiven Kontext.

Unvorhersehbarkeit: Boden betont mehrmals, dass kreative Ideen unvorhersehbar und überraschend seien.^{12,13} Daraus ergebe sich allerdings ein wissenschaftliches Paradoxon:

„If – as many people believe – science conveys the ability to predict, a scientific psychology of creativity is a contradiction in terms. Someone who claims that creativity can be scientifically understood must therefore

⁸ BODEN, *The creative mind*, 1.

⁹ PREISER, *Kreativitätsforschung*, 5.

¹⁰ Vgl. ebd., 2.

¹¹ Vgl. BODEN, *The creative mind*, 2.

¹² Vgl. ebd., 1.

¹³ Vgl. ebd. 13.

show in just what sense it is unpredictable, and why this unpredictability does not anchor it firmly in the depths of mystery.“¹⁴

Diese von Boden konstatierte Unvorhersehbarkeit von Kreativität habe allerdings nicht zur Folge, dass sie nicht wissenschaftlich untersucht werden könne.¹⁵ Zunächst allerdings müsse die mystische Komponente von Kreativität aufgelöst werden.¹⁶ Geraint A. Wiggins, seines Zeichens Professor für Computational Creativity an der Queen Mary-Universität London, widerspricht an dieser Stelle und entwickelt sogar ein quantitatives Kreativitätsmodell auf Grundlage der Annahme, dass kognitive Kreativität ein Ergebnis von Vorhersage sei.¹⁷ Der Aspekt der Unvorhersehbarkeit ist interessant in Anbetracht des Themas der vorliegenden Arbeit: Es wird zu klären sein, ob ein Computer so programmiert werden kann, dass seine Ausgabe zwar unvorhersehbar, aber nicht zufällig ist.

Wertbarkeit: Neben der Neuartigkeit und Unvorhersehbarkeit scheint auch die Möglichkeit, einer Idee oder einem Gegenstand einen Wert geben zu können, eine Rolle für die Einstufung als kreatives Produkt zu spielen. So werden Ideen, die rein zufällig zustande kamen, gegenüber solchen, deren Entstehung beabsichtigt war, zuweilen nicht als kreativ bezeichnet. Der Weg zu einer Idee muss einen äußeren Sinn ergeben.¹⁸ Wiggins u.a. beschreiben etwa den höheren Wert eines mathematischen Beweises, wenn er „elegant“ ist.¹⁹ Sie definieren Wert als Beziehung zwischen dem Gegenstand, seinem Schöpfer, dem Beobachter sowie dem Kontext.²⁰ Elisabeth Gutjahr beschreibt diese Eigenschaft nicht als „Wert“, sondern als „Relevanz“.²¹

¹⁴ Ebd., 13.

¹⁵ Vgl. ebd.

¹⁶ Vgl. ebd., 23.

¹⁷ Vgl. WIGGINS u.a., The evolutionary roots of creativity, 4.

¹⁸ Vgl. BODEN, The creative mind, 3.

¹⁹ Vgl. WIGGINS u.a., The evolutionary roots of creativity, 2.

²⁰ Vgl. ebd.

²¹ Vgl. GUTJAHR, Der Mythos Kreativität, 13.

Sie wirft im Übrigen auch die Frage auf, ob sich Kreativität auf ein Teilgebiet beschränken kann, oder ob es sich um eine fächerübergreifende Eigenschaft handelt.²² In der griechischen Antike etwa wurde Kreativität sogar auf die Berufsgruppe der Dichter beschränkt.²³ Sollte eine solche Beschränkung von Kreativität auf ein Teilgebiet zutreffen, hätten Computerwissenschaftler jedes Recht, einen Computer, der zwar nur wenige Dinge, diese dafür aber zuverlässig erledigen kann, als kreativ zu bezeichnen.

In den letzten Jahren gab es gerade in den Neurowissenschaften eine große Zunahme an Publikationen zu Kreativität, die sich meistens auf das Messen von Hirnströmen beschränken. Eine konkrete Lokalisierung von Kreativität ist hierbei nicht erkennbar, vielmehr sind eine Vielzahl von Hirnregionen aktiv.²⁴ Die Vermutung ist, dass es sich bei Kreativität um ein komplexes Zusammenspiel von gewöhnlichen kognitiven Prozessen (etwa Gedächtnis, Aufmerksamkeit, Ausführung und Gefühl) handeln könnte.²⁵

2.1 Überblick über qualitative Modelle

Die folgende Auswahl soll einen Überblick über existierende Kreativitätsmodelle bieten. Die vorgestellten Modelle beschreiben den kreativen Prozess in der Regel als einen Problemlösungsprozess. Es handelt sich um qualitative Beschreibungen.

2.1.1 Vier-Phasen-Modelle

Eines der frühesten Modelle zum kognitiven Kreativitätsprozess beschreibt der Mitbegründer der London School of Economics, Graham Wal-

²² Vgl. ebd., 12.

²³ Vgl. BECKER, Art. Creativity Through History, 304.

²⁴ Vgl. WIGGINS u.a., Bridging computational and neuroscientific approaches, 4 sowie 5.

²⁵ Vgl. ebd.

las, um 1926 in seiner Arbeit „The art of thought“.²⁶ Er teilt den Prozess schematisch in vier Phasen:

1) Preparation: Durch die gründliche Betrachtung eines Problems wird das kreative Ziel definiert und ins Auge gefasst. Hierbei werden Wissen und intellektuelle Techniken erworben. – **2) Incubation:** Das Problem wird nicht bewusst, sondern unterbewusst weitergehend bearbeitet. – **3) Illumination:** Die kreative Idee wird geboren – dieser fast magisch erscheinende Moment der „Erleuchtung“ beschreibt etwa den Zeitpunkt, zu dem Archimedes „Heureka“-rufend die Badewanne verlässt. **4) Verification:** Die Idee wird getestet und hierbei weiterentwickelt und feingeschliffen.²⁷

Die Phasen aus dem Modell Preisers entsprechen weitestgehend denen von Wallas, werden allerdings vorher durch die Phasen „Person-Umwelt-Interaktion“ und „Problemwahrnehmung und -analyse“ sowie nachher um die Phasen „Kommunikation“ und „Realisierung“ ergänzt.²⁸ Bei den laut Preiser empirisch nachweisbaren Phasen handle es sich nicht unbedingt um reale, vorfindbare Sachverhalte, sondern um replizierbare Konstrukte. Die Phaseneinteilung stelle hierzu eine sinnvolle gedankliche Hilfskonstruktion dar. Durch entsprechende Versuchsanordnungen sei auch eine Zuordnung zu objektiven, physiologischen Charakteristika möglich (etwa anhand des EEG), außerdem ließen sich aus dem Phasenmodell konkrete Anweisungen und Hilfsmittel ableiten, um den Problemlösungsprozess zu fördern.²⁹

Auch Gutjahr beschreibt vier Phasen, die inhaltlich den oben zitierten entsprechen: „Anbahnung“, „Odyssee“, „Heureka“ und „Verifikation“.³⁰

²⁶ Vgl. WALLAS, The art of thought.

²⁷ Vgl. WIGGINS u.a., The evolutionary roots of creativity, 4 sowie BECKER, Art. Creativity Through History, 306.

²⁸ Vgl. PREISER, Kreativitätsforschung, 42.

²⁹ Vgl. ebd., 48.

³⁰ Vgl. GUTJAHR, Der Mythos Kreativität, 25f.

2.1.2 Divergent statt konvergent

Ein anderer Ansatz ist der von Joy Paul Guilford. Mit einem Artikel im *American Psychologist* gab der Intelligenzforscher in den 1950er Jahren den Startschuss zu einem Paradigmenwechsel in der Kreativitätsforschung. Der Hauptansatz Guilfords ist die Unterscheidung zwischen konvergentem und divergentem Denken: ersteres beschreibt das Ausrichten des Handelns an konventionelle oder korrekte Problemlösungen.³¹ Letzteres hingegen ermöglicht es, von einem Punkt aus in verschiedene Richtungen auszuschweifen. Dieses *divergent thinking* ist zu einem feststehenden Begriff in der Intelligenzforschung geworden. Einer Phase divergenten Denkens, in der durch Abschweifung vom ursprünglichen Gegenstand eine mögliche Analogie zum behandelnden Problem gesehen wird, folgt in der Regel eine Phase konvergenten Denkens, in der die neue, nun im Denkmuster etablierte Lösung, angewendet wird. Beide Phasen können bewusst und unterbewusst ablaufen und sich stetig wiederholen.³² Dieses Modell widerspricht dem oben genannten Vier-Phasen-Modell nicht, sondern entspricht lediglich einem anderen Betrachtungswinkel.

2.1.3 Bisoziation statt Assoziation

Der österreichisch-ungarische Schriftsteller Arthur Koestler bezieht in seinem umfassenden zweiteiligen Werk „The act of creation“ (1964, deutsch: „Der göttliche Funke“, 1966) jegliche schöpferischen Prozesse auf Bisoziation. Hierbei werden ähnlich wie bei der Assoziation verschiedene Gegenstände auf kognitiver Ebene miteinander verknüpft. Während es sich bei Assoziationen allerdings um Verknüpfungen von Gegenständen auf der gleichen Ebene (Koestler spricht von Matrizen³³) handelt, durchbrechen bisoziative Prozesse die Ebenen und verknüpfen Gegenstände, die

³¹ Vgl. RUNCO, Art. Divergent Thinking, 400.

³² Vgl. WIGGINS u.a., The evolutionary roots of creativity, 4.

³³ Die Matrix steht vermutlich für etwas ähnliches wie die *conceptual spaces* bei Boden.

üblicherweise nicht miteinander in Verbindung gebracht werden.³⁴ In gewisser Weise wird hier das *divergent thinking*, also das Abschweifen vom ursprünglichen Gegenstand, weitergedacht, indem das Gefundene dann wieder mit dem ursprünglichen Gegenstand verknüpft wird. Bisoziative Prozesse sind Koestler zufolge auch die Grundlage von Humor: Die Gegenüberstellung nicht zusammengehörig scheinender Gegenstände wird zuweilen als komisch rezipiert.³⁵ Den Unterschied zwischen psychologischer und historischer Kreativität (vgl. Boden) beschreibt Koestler folgendermaßen:

„Minor, subjective bisociative processes do occur on all levels, and are the main vehicle of untutored learning. But objective novelty comes into being only when subjective originality operates on the highest level of the hierarchies of existing knowledge.“³⁶

Objektive Neuartigkeit (\cong *H-creativity*) ist also nur möglich, wenn der Kreativitätsprozess unter Einfluss größtmöglich existierenden Wissens geschieht. Prozessual gibt es demnach keinen Unterschied. In einer Tabelle fasst Koestler den Unterschied zwischen assoziativen und bisoziativen Prozessen zusammen:³⁷

Gewohnheit	Originalität
Assoziation innerhalb der Grenzen einer gegebenen Matrix	Bisoziation unabhängiger Matrizen
Orientierung an vorbewussten oder außer-bewussten Prozessen	Orientierung an unterbewussten Prozessen, normalerweise unter Zurückhaltung
Dynamisches Gleichgewicht	Aktivierung erneuerbaren Potenzials
Starr gegenüber flexiblen Variationen eines Themas	Super-Flexibilität
Wiederholung	Neuartigkeit

³⁴ Vgl. KOESTLER, The act of creation, 646f und 656.

³⁵ Vgl. ebd., 38-45.

³⁶ Ebd., 658.

³⁷ Vgl. ebd., 659f.

Gewohnheit	Originalität
konservativ	destruktiv-konstruktiv

2.1.4 Im Flow

Mihaly Csikszentmihalyi interessierte sich für das Phänomen intrinsischer Motivation bei dem Ausüben von kreativen Tätigkeiten. Er befragte in qualitativen Interviews etwa Chirurgen oder Extremsportler über die Art der Erfahrung, wenn eine solche Tätigkeit gut läuft. Hieraus entwickelte er seine „Flow“-Theorie, die er erstmalig 1975 veröffentlichte.³⁸

Der „Flow“-Zustand beschreibt das Gleichgewicht zwischen persönlichen Fähigkeiten und gestellten Anforderungen: Wenn die Anforderungen einer bestimmten Aufgabe größer sind als die persönlichen Fähigkeiten, tendiert das persönliche Empfinden in Richtung „wachsam“ bis „ängstlich“. Wenn die persönlichen Fähigkeiten jedoch die gestellten Anforderungen übertreffen, tendiert es in Richtung „entspannt“ bis hin zu „gelangweilt“. Flow findet also statt, wenn sich die wahrgenommenen Herausforderungen einer Aktion sowie die wahrgenommenen persönlichen Fähigkeiten im Gleichgewicht befinden.³⁹

Während des Flow-Zustands arbeitet der Ausführende mit größtmöglicher Leistungsfähigkeit und größtenteils kreativ. Weitergehend beschreibt Csikszentmihalyi folgende subjektiv wahrgenommenen Eigenschaften:

- eine intensive und fokussierte Konzentration auf die ausgeübte Tätigkeit,
- das Verschmelzen von Aktion und Bewusstsein,
- den Verlust des reflektierten Selbstbewusstseins,
- das Gefühl der Kontrolle sowie

³⁸ Vgl. CSIKSZENTMIHALYI, The Concept of Flow, 241–246.

³⁹ Vgl. ebd., 241 sowie 248.

- eine geänderte Zeitwahrnehmung.⁴⁰

Außerdem beschreibt er eine intrinsische Motivation, also das Ausüben der Aktion um ihrer selbst Willen, anstelle einer extrinsischen Motivation etwa durch die Aussicht auf ein durch die ausgeführte Tätigkeit entstehendes Produkt.⁴¹ Diese subjektiv wahrgenommene Belohnung ermutige den Ausübenden auch, an einer Aufgabe dranzubleiben oder sie nach einer Zeit wieder aufzunehmen. Durch das Ausüben der Tätigkeit werden die persönlichen Fähigkeiten mit der Zeit erweitert.⁴²

Die Richtung des sich ausbreitenden Flow-Zustands wird sowohl von der Person als auch von der Umwelt bestimmt: Csikszentmihalyi beschreibt das als „emergent motivation“, also als aufstrebende Motivation, die innerhalb eines offenen Systems stattfindet: Das einer Aktion folgende Ereignis entsteht aus der Interaktion mit dem unmittelbar zuvor stattfindenden Ereignis, wird also nicht von Intentionen seitens der Person oder Umwelt vorbestimmt.⁴³

Obwohl sich die Forschung zunächst nur auf das Phänomen der intrinsischen Motivation als solches beschränkte, gab es schon früh Vermutungen über eine „autotelische Persönlichkeit“ (von auto = selbst, telos = Ziel), die das Aufkommen von Flow begünstigt, auch wenn die Fähigkeit dazu nahezu universal erscheint. So können Unterschiede in der Dauer, der Intensität oder der Häufigkeit von Flow-Zuständen ausgemacht werden. Als autotelische Person beschreibt Csikszentmihalyi eine Person, die das Leben genießt und Dinge um ihrer Selbst willen tut. Charaktereigenschaften seien allgemeine Großzügigkeit, Interesse am Leben, Ausdauer sowie nicht ausgeprägte Egozentrik.⁴⁴

⁴⁰ Vgl. ebd., 241.

⁴¹ Vgl. ebd., 244.

⁴² Vgl. ebd., 249.

⁴³ Vgl. ebd., 242.

⁴⁴ Vgl. ebd., 245.

Die oben beschriebenen Kreativitätsmodelle sind für eine computergestützte Simulation nur bedingt tauglich, da sie die Kreativität qualitativ beschreiben. Für einen rechengestützten Prozess bedarf es allerdings quantitativer Modelle. Hierbei lässt sich die Frage aufwerfen, inwiefern sich quantitative Modelle überhaupt eignen, um ein Phänomen wie Kreativität zu beschreiben.

2.2 Computational Creativity als quantitatives Modell

Die Computational Creativity versteht sich als eine Schnittstellendisziplin der künstlichen Intelligenz, der kognitiven Psychologie, der Philosophie und der Künste. Mit Hilfe des Computers erstellt sie Modelle, Simulationen oder – so beschreibt es The Association for Computational Creativity (ACC) – Kreativitätsreplikationen. Als Ziele nennt die Vereinigung

- das Erstellen von Programmen oder Rechnern, die zu menschlicher Kreativität fähig sind (= künstlicher Reproduktion menschlicher Kreativität),
- einen Erkenntnisgewinn gegenüber menschlicher Kreativität mit Hilfe einer algorithmischen Perspektive auf menschliche Kreativität sowie
- die Konzeption von Programmen, die menschliche Kreativität erweitern, ohne dabei selber kreativ zu sein.

Forschungsfelder seien sowohl theoretischer als auch praktischer Natur, wobei sich ersteres mit einer genauen Definition von Kreativität beschäftige und letzteres mit der Implementierung von Systemen, die Kreativität aufweisen.⁴⁵ Die erwähnte Association for Computational Creativity ist eine „non-profit organisation“, deren Aufgabe unter anderem das Ausrichten der jährlich stattfindenden „International Conference on Computational Creativity (ICCC)“ ist.⁴⁶

⁴⁵ Vgl. AA.VV., Computational Creativity (computationalcreativity.net).

⁴⁶ Vgl. AA.VV., About (computationalcreativity.net).

Interessant an obiger Selbstdarstellung der ACC ist, dass die grundsätzliche Möglichkeit zu künstlicher Kreativität gar nicht in Frage gestellt wird – es wird zumindest davon ausgegangen, dass eine Replikation menschlicher Kreativität möglich ist. Das ist vor allem deswegen erwähnenswert, weil die Forschung, wie wir später sehen werden, noch weit entfernt ist von einem grundlegenden Verständnis kognitiver Kreativität. Der Glaube der künstlichen Replikation von Kreativität entspricht aber dem uneingeschränkten Fortschrittsglauben der künstlichen Intelligenzforschung.

Unabhängig von der grundsätzlichen Möglichkeit ist auch die Rezeption von künstlicher Kreativität ein interessanter Aspekt: Boden spekuliert, dass selbst bahnbrechende Theorien wie solche von Einstein nicht als kreativ bezeichnet würden, sofern sie von einem Computer entwickelt worden sein sollten.⁴⁷ Der Musiktheoretiker und selbsternannte Computer-Wissenschaftler David Cope erzählt, dass das Vorspielen von Stücken, die sein Computerprogramm erstellt hat, solange auf Zustimmung stößt, wie der Zuhörer im Unklaren darüber gelassen wird, dass es sich um eine künstliche Komposition handelt.⁴⁸ Auf einer musikwissenschaftlichen Konferenz in Köln wurde Cope sogar für den Tod der Musik verantwortlich gemacht und der ihn beschuldigende Redner hatte sich offenbar so in Rage geredet, dass er versuchte, Cope zu schlagen.⁴⁹ An diesen Beispielen sieht man, dass die eingangs erwähnte Wertbarkeit kreativer Ideen oder Gegenstände eine nicht zu unterschätzende Wichtigkeit hat. Und ganz offensichtlich – und weitestgehend selbstredend verständlicherweise – wird Computern nicht der gleiche Wert zugeschrieben wie Menschen. Dieses Problem beschreibt auch Simon Colton, der Entwickler der Software *The Painting Fool*, die sich auf der Webseite folgendermaßen vorstellt:

⁴⁷ Vgl. BODEN, *The creative mind*, 7.

⁴⁸ Vgl. BLITSTEIN, *Triumph of the Cyborg Composer*.

⁴⁹ Vgl. STIELER, *Die Mozart-Maschine*, 32.

„I’m The Painting Fool: a computer program, and an aspiring painter. The aim of this project is for me to be taken seriously – one day – as a creative artist in my own right.“⁵⁰

Colton stellt fest, dass die Kreativität eines Computerprogramms häufig seinem Programmierer zugewiesen wird. Er vergleicht dieses Verhalten mit der Absurdität, die Kreativität eines Künstlers seinem Lehrer zuzuschreiben. Schließlich sei das Ziel seiner Arbeit ja, dass das Verhalten der Software nicht vorausgesehen werden könne.⁵¹ Er schlägt daher einen „creative tripod“ vor: Die drei Beine dieses Dreifußes stellen die seiner Meinung nach nötigen Verhaltensweisen eines Systems dar, um als kreativ rezipiert zu werden, angefangen von Fertigkeit über Anerkennung bis hin zu Vorstellungskraft.⁵² Nur bei Vorhandensein all dieser Eigenschaften könne ein System als kreativ wahrgenommen werden:

„Our position is that, if we perceive that the software has been skillful, appreciative and imaginative, then, regardless of the behaviour of the consumer or programmer, the software should be considered creative. Without all three behaviours it should not be considered creative, but the more aspects which extend each leg of the tripod, the more creativity we should project onto the software.“⁵³

Er schlägt daher vor, den Rezipienten der Produkte von künstlich-kreativer Software genaueste Beschreibungen über die Funktionsweise der Software zu geben.⁵⁴ Cope geht sogar noch einen Schritt weiter und antwortet auf das Argument, Computer hätten keine Seele und könnten allein daher nicht kreativ sein, wie üblich provokativ: „The question [...] isn’t whether computers have a soul, but whether humans have a soul.“⁵⁵

⁵⁰ COLTON, SIMON, About The Painting Fool (thepaintingfool.com).

⁵¹ Vgl. COLTON, Creativity Versus the Perception of Creativity in Computational Systems, 18.

⁵² Vgl. ebd.

⁵³ Ebd.

⁵⁴ Vgl. ebd., 20.

⁵⁵ COPE zit. n. BLITSTEIN, Triumph of the Cyborg Composer.

2.2.1 Theoretische Grundlagen

Neben der grundsätzlichen Voraussetzung der Erstellung quantitativer Modelle, die von Computern verarbeitet werden können, gibt es zwei Theorien, die eine aus der kognitiven Psychologie, die andere aus der statistischen Mathematik, auf die sich die Computational Creativity häufig beruft:

Global Workspace Theory

„Almost everything we do, we do better unconsciously than consciously.“⁵⁶

In seiner Publikation „A cognitive theory of consciousness“ (1988) vergleicht Bernard J. Baars zunächst bewusste und unterbewusste Prozesse und kommt hierbei zu obigem Schluss. So laufe etwa die Verarbeitung von textlicher Syntax stets unterbewusst ab und würde als bewusster Prozess viel mehr Zeit in Anspruch nehmen. Baars zufolge verbessert sich das Ausüben einer Aufgabe von Mal zu Mal: je effizienter eine Aufgabe erledigt wird, desto weniger findet sie im Bewusstsein statt.⁵⁷

Laut der Theorie finden viele Prozesse gleichzeitig im Unterbewusstsein statt, die für das alltägliche Leben nötig sind – das reicht von banalen Beispielen wie Atmen oder der Fähigkeit zu Laufen bis hin zu komplexeren Aufgaben wie der zuvor erwähnten Fähigkeit der Syntax-Analyse. Es handelt sich um spezialisierte Prozesse, die meist nur eine bestimmte Aufgabe haben. Während sie sozusagen „hinter der Bühne“ stattfinden (die Theater-Metapher entspricht ebenfalls Baars), geschieht der bewusste Prozess immer im „Global Workspace“ ($\hat{=}$ Bewusstsein). Alle unterbewusst ablaufenden Prozesse kämpfen um einen Zugang zu diesem Global Workspace, wobei sie auch miteinander kooperieren können, um dieses Ziel zu erreichen. Der Global Workspace dient als Informationsaustausch, über den die

⁵⁶ BAARS, Cognitive Theory of Consciousness, 73.

⁵⁷ Vgl. ebd.

Prozesse miteinander interagieren können. Gelangen Prozesse in den Global Workspace, können sie von hier aus Nachrichten an andere spezialisierte Prozesse übermitteln, die ihr Verhalten dementsprechend anpassen können.⁵⁸

Shannons Informationstheorie

Claude Elwood Shannon gilt mit seiner Informationstheorie von 1948 als Mitbegründer der Informatik. Eine Information wird hierbei grundsätzlich als die Antwort auf eine Frage definiert. Die einfachste Information ist die boolesche Variable (ja/nein bzw. true/false bzw. 0/1), gemessen in der Einheit Bit. Da es natürlich auch komplexere Informationen gibt, beschreibt man mit dem Informationsgehalt (= Entropie) eine zusätzliche Größe: Dieser Informationsgehalt ist per Definition die Anzahl von benötigten Ja-/Nein-Fragen zum vollständigen Erfassen einer Information. Um etwa die Augen eines geworfenen Würfels zu erkennen, muss man bis zu sechs Fragen stellen. Damit gilt der Würfel als sogenannte 6-Alternative, seine Entropie ist gleich 6.

Shannon entwickelte die Informationstheorie mit dem Ziel, Informationen über Telefonleitungen zu übertragen, ohne dass sie von Störungen betroffen sind. Gerade in der digitalen Übertragungstechnik haben fehlerhaft übertragene Informationen fatale Auswirkungen auf die empfangene Nachricht. Trotz der ursprünglichen Anwendung im Bereich der Nachrichtentechnik wird seine Theorie inzwischen auf Kommunikationsprobleme im weitesten Sinne angewandt:⁵⁹ Die ursprünglich für Drähte oder elektromagnetische Wellen formulierte Theorie wird also auch auf Nerven, Alphabete oder Sinnesorgane verwendet.⁶⁰ In der Informatik gilt die Informationstheorie als allgemeiner Mechanismus für die Organisation von Informationen und Ereignissen in der Welt.

⁵⁸ Vgl. ebd., 74.

⁵⁹ Vgl. MITTENECKER, Art. Informationstheorie, 780.

⁶⁰ Vgl. RANDOW, Was ist Informationstheorie (zeit.de).

2.2.2 Information Dynamics of Thinking (IDyOT)

Wiggins u.a. beschreiben einen Fortschritt in Richtung einer computer-gestützten kognitiven Architektur mit dem bezeichnenden Namen IDyOT (Information Dynamics of Thinking). Diese Architektur soll gewisse Aspekte menschlicher Kreativität mit einem „pre-conscious predictive loop“, also einer vorbewussten Schleife von Vorhersagen modellieren.⁶¹

Der Ansatz ist, dass sich Shannons Informationstheorie, die in der künstlichen Intelligenz als allgemeiner Mechanismus für die Organisation von Informationen und Ereignissen in der Welt gesehen wird, auch zur Darstellung von nicht-bewusster Kreativität eigne.⁶² Die Informationstheorie wird hierbei mit der Global-Workspace-Theorie Baars' verknüpft:

- Eine große Anzahl von Generatoren sagt Wahrscheinlichkeiten für einzelne Teilbereiche voraus, indem sie ein großes statistisches Sequenzmodell abtastet
- Jeder Generator enthält dabei einen Teil der Wahrnehmung (wobei auch falsche Wahrnehmungen enthalten sein können), aus dem mit Hilfe von Markov-Ketten⁶³ Vorhersagen über den künftigen Zustand dieses Teilbereichs errechnet werden.
- Die Vorhersagen der Generatoren werden immer wieder mit den aktuellen Wahrnehmungen abgeglichen.
- Wenn eine Übereinstimmung im Vergleich zu anderen Übereinstimmungen eine größere Informationsdichte aufweist, wird der Generator in den „Global Workspace“ entleert.
- Die Informationen gelangen dadurch ins Bewusstsein und werden zusätzlich abgespeichert (und Teil des obigen Sequenzmodells).
- Wenn ein Generator keine neue Eingabe (also keinen Teil der Wahrnehmung) bekommt, läuft er frei und wird von zuvor gespeicherten

⁶¹ Vgl. WIGGINS u.a., IDyOT: A Computational Theory of Creativity, 127.

⁶² Vgl. ebd.

⁶³ Vgl. Kapitel 3.3.

Ereignissen aus dem Gedächtnis angetrieben. An dieser Stelle werde Kreativität zugelassen.⁶⁴

In der Theorie von Wiggins u.a. findet Kreativität also an den Stellen statt, wo Teile der aktuellen Wahrnehmung mit Vorhersagen aus dem Gedächtnis kombiniert werden.

Durch dieses Modell lasse sich darstellen, wie Mozart seine eigene Kompositionsweise charakterisiert: Der Komponist erklärt, die Ideen kämen ihm, wenn er alleine sei, nach einem guten Essen spaziere oder des Nachts. Wenn eine Idee ihm besonders gut gefalle, bliebe sie in seinem Gedächtnis. Dort werden die Ideen gegebenenfalls abgewandelt und so lange gesammelt, bis sie sich zu etwas Neuem zusammensetzen ließen. Er höre die Teile dabei nicht nacheinander, sondern alle gleichzeitig.⁶⁵

Das Modell von Wiggins wurde bisher noch nicht umgesetzt. Eine abschließende Bewertung ist daher nicht möglich. Es ist anzunehmen, dass das Modell sich in jedem Fall eignet, um tiefergehende Erkenntnisse über Bewusstsein und Kreativität zu bekommen: Die Global-Workspace-Theorie sowie viele andere theoretische Ansätze, die in dieses Modell einfließen, könnten auf diese Weise in einer Simulation erprobt werden. Das alles bringt die Forschung gewiss weiter, unabhängig davon, ob die Simulation erfolgreiche oder weniger erfolgreiche Ergebnisse zu Tage bringt.

⁶⁴ Vgl. ebd., 134–136.

⁶⁵ Vgl. ebd., 145.

3 Methoden der algorithmischen Komposition

Als algorithmische Komposition wird gemeinhin die (teilweise) automatisierte Komposition durch Software beschrieben. Hierbei gibt es verschiedene Methoden. David Cope liefert einen Überblick, der allerdings nicht ganz trennscharf, teilweise sogar fehlerhaft ist⁶⁶ (es werden Methoden miteinander vermischt, die zunächst einmal nichts miteinander zu tun haben). Die folgende Strukturierung basiert zu großen Teilen auf einem Artikel von Fernández u.a.⁶⁷, in dem Beispiele der algorithmischen Komposition auf ihre eingesetzten Methoden untersucht werden. Diese Darstellung hat keinen Anspruch auf Vollständigkeit, sondern soll nur einen Überblick liefern. In der praktischen Anwendungen werden die Methoden häufig miteinander kombiniert, um zu besseren Ergebnissen zu führen.

3.1 Zufall

„... especially in the arts, random output often competes with creativity, at least in terms of novelty and surprise.“⁶⁸

David Cope führt auf, in welchen Zusammenhängen das Wort „Zufall“ häufig gebraucht wird: So werden zum Beispiel Prozesse als zufällig bezeichnet, die eine höhere *Komplexität* aufweisen. Die Position eines Wasserstoffatoms im Ozean etwa ließe sich schwer über eine gewisse Zeit vorhersagen. Natürlich wäre eine Berechnung rein theoretisch denkbar, doch wäre sie so komplex, dass sie auf den ersten Blick nicht nachvoll-

⁶⁶ Vgl. WIGGINS, CMMC. A Review, 113: „If you find a copy of Computer Models of Musical Creativity in your library, I suggest you advise your undergraduate students not to read this section (pp. 57–80), lest they believe it and lose credit in their exams as a result. It’s worth emphasising here that these errors are not matters of subjective opinion, but of defined fact.“

⁶⁷ FERNÁNDEZ u.a., AI Methods in Algorithmic Composition, 513–582.

⁶⁸ COPE, Computer Models of Musical Creativity, 51.

ziehbar sei. Auch Prozesse, die kein erkennbares Muster aufweisen, werden zuweilen als „zufällig“ bezeichnet: Ein Beispiel hierfür sind die Nachkommastellen der Zahl π , bei denen sich kein wiederholendes Muster erkennen lässt.

In Programmiersprachen gibt es keinen echten Zufall: tatsächlich handelt es sich um deterministische Prozesse, die daher auch als „pseudo randomness“⁶⁹ bezeichnet werden. Stattdessen wird beispielsweise die *Mathematische Modellierung* verwendet, bei der Funktionen mit sehr unregelmäßigen Ausgaben (z.B. $f(x) = 1 / \cos x^2$) etwa mit der aktuellen Zeit auf der x-Achse befüllt werden, um zufallsähnliche Ausgaben zu erzeugen.⁷⁰

„The bottom line for my own research is that randomness is not an engaging mystery, but a simple reflection of ignorance. Aside from the *possible* exception of quantum physics, randomness refers to behaviour that is either too complex, too patternless, or too irrelevant to make prediction possible. None of these features seem to me to be associated in any way with creativity. In fact, while much of what we call creativity is also unpredictable, creativity often turns out in hindsight to be the most rational way to have proceeded. Reverse engineering even the most complex creative processes demonstrates this rationality. Randomness, on the other hand, perpetuates or even complicates problems – and should never be confused with creativity.“⁷¹

Trotz dieses recht überzeugenden Statements gibt es, wie wir später sehen werden, Programme, die natürlich nicht ausschließlich, aber unter anderem zufallsbasiert sind.

⁶⁹ Ebd., 53.

⁷⁰ Vgl. ebd., 76.

⁷¹ Ebd., 57.

3.2 Wissens- und regelbasierte Methoden

Unter diese Kategorie fallen Techniken der „good old-fashioned AI“⁷². Sie umfasst Grammatiken, regelbasierte und Regeln-lernende Programme. Meistens werden dazu symbolische Programmiersprachen wie LISP, Haskell oder Mathematica genutzt.⁷³

3.2.1 Grammars

Grammars können als Regelset beschrieben werden, durch das auf hierarchische Weise übergeordnete Symbole zu detaillierteren Symbolsequenzen erweitert werden. Wie in einer Grammatik werden also bestimmte Regeln über die weitere Verarbeitung von Daten aufgestellt.

Solche regelbasierten Programme funktionieren grundsätzlich nach dem Wenn-Dann-Prinzip, das in der Programmierung über If-Else-Konditionierung herbeigeführt werden kann.⁷⁴ Der Programmierer stellt dadurch eine musikalisch-kompositorische Grammatik auf. Das eignet sich zum Beispiel hervorragend für zweistimmige Kontrapunkte nach festen Regeln oder für Bachsche Kadenzten mit barocktypischen Klauseln. Ein Anwendungsfall regelbasierter Programmierung wäre eine „Kontrapunkt-Hausaufgabenkontrolle“, mit der überprüft werden kann, ob ein zweistimmiger Kontrapunkt den Regeln Palestrinas oder Lassos entspricht. Die Kontrapunkt-Regeln werden in entsprechende If-Else-Konditionen abgewandelt, etwa:

- „wenn in der Melodie eine kleine Sext abwärts vorkommt, gib einen Fehler aus“ oder

⁷² FERNÁNDEZ u.a., *AI Methods in Algorithmic Composition*, 518.

⁷³ Wobei Symbole einfache Wörter darstellen, die für Regeln oder Datensätze stehen. Das erlaubt ein einfaches Lesen und Schreiben des Programmcodes.

⁷⁴ Vgl. COPE, *Computer Models of Musical Creativity*, 60.

- „wenn sich beide Stimmen parallel bewegen, überprüfe, ob die Parallelbewegungen rhythmisch verzögert ist. Wenn nicht, gib einen Fehler aus“.

So sehr sich regelbasierte Programmierung für die Analyse eignet, so wenig ist sie doch als alleinige Methode für die automatische Komposition zu gebrauchen: Trotz der strengen Kontrapunkt-Regeln gibt es natürlich auch hier immer mehrere Optionen. Es bedarf also eines zusätzlichen Mechanismus, der aus diesen Optionen auswählt.

Fälschlicherweise ordnet Cope auch Markov-Ketten unter die regelbasierte Programmierung ein. Auch, wenn aus Markov-Ketten Regeln abgeleitet werden können, sind sie eher unter dem Oberbegriff des „Machine learning“ einzuordnen.

3.2.2 Regel-lernende Programme

Cope grenzt die datengestützte Programmierung von der regelbasierten Programmierung ab. Als datengestützt bezeichnet er Programme, bei denen Daten aus der Analyse gewonnen werden (z.B. harmonische Beziehungen, hierarchische Informationen, Stileigenschaften o.ä.) und anschließend entsprechend den Benutzervorgaben nachgebildet und neu zusammengesetzt werden. Copes „Experiments in Musical Intelligence“ (EMI) verfolgt diesen Ansatz. Formal richtiger ist allerdings die Bezeichnung der Regel-lernenden Programme: Letztendlich werden nämlich auch hier aus den Daten Regeln abgeleitet.

„While the knowledge implemented in rule-based systems is usually static, part of the knowledge may be dynamically changed or learned. The natural term for this concept is *machine learning*, but its meaning is unfortunately vague, because it is used as a catch-all for many methods, including neural networks and Markov chains.“⁷⁵

⁷⁵ FERNÁNDEZ u.a., AI Methods in Algorithmic Composition, 528.

Die für den Lernprozess bereitgestellten Daten spielen hierbei eine zentrale Rolle.⁷⁶ Der Ansatz von Regel-lernenden Programmen, die aus großen Datensätzen lernen, ist sehr nachvollziehbar, sind doch auch bei vielen Komponisten klare Einflüsse erkennbar. Auch ein Kompositions- oder Tonsatzstudium fordert die Kenntnis über ein Gros der klassischen Meisterwerke. Eine Datenbank könnte diese Kenntnis widerspiegeln. Es ist denkbar, dass aus einem solchen Wissen bestimmte Regeln erst entstehen. Allerdings ist es eher unwahrscheinlich, dass sich dadurch stilprägende Kompositionsregeln (etwa eine klassische Phrasenbildung) nachbilden lassen.

3.2.3 Constraint-Satisfaction

Das Constraint-Satisfaction-Problem (CSP) sucht für mehrere Variablen Belegungen, die alle aufgestellten Bedingungen (= Constraints) erfüllen. Teilweise beeinflussen sich hierbei die Variablenwerte untereinander. Dadurch können CSP auch unlösbar sein.⁷⁷ Ein typisches CSP ist das sogenannte „Damenproblem“, bei dem acht Damen auf einem Schachbrett so positioniert werden sollen, dass sich keine zwei Damen einander entsprechend der ihnen eigenen Zugmöglichkeiten schlagen können. Das Damenproblem hat für ein Schachbrett normaler Größe 92 verschiedene Lösungen.

In der Komposition von Musik hat man es oftmals mit ähnlichen Problemen zu tun: Die Regeln der Stimmführung etwa könnten als Constraint-Satisfaction-Problem gesehen werden, da bestimmte Bewegungen innerhalb einer Stimme die Bewegungsmöglichkeiten der anderen Stimmungen beeinflussen können.

⁷⁶ Vgl. COPE, *Computer Models of Musical Creativity*, 62f.

⁷⁷ Vgl. POOLE u.a., *Artificial Intelligence*, 117f.

3.2.4 Case-based reasoning

Ein ebenfalls auf Daten aufbauender Prozess ist das Case-based reasoning (CBR, dt. fallbasiertes Schließen). Das sind Systeme, die in einer Falldatenbank nach bekannten Problemen und ihren zugehörigen Lösungen suchen, um sie auf ähnliche Probleme anzuwenden. Die Lösung wird hierbei nicht angepasst, sondern es wird vermutet, dass die Lösung des bekannten Problems sich so auch auf das neue Problem anwenden lässt. Cope gibt zu bedenken, dass Case-based reasoning allein kein kreatives Verhalten aufweist – in Kombination mit „association-based procedures“ werde das kreative Potential allerdings enorm gesteigert.⁷⁸

3.3 Markov-Ketten

Markov-Ketten werden benutzt, um die Wahrscheinlichkeiten für einen künftigen Zustand anhand eines vergangenen Zustands vorherzusagen. Es handelt sich also um einen zeitdiskreten stochastischen Prozess: Die zugehörige Markov-Eigenschaft besagt, dass zukünftige Entwicklungen eines Prozesses ausschließlich vom zuletzt beobachteten Zustand abhängig sind. Markov-Systeme werden daher auch als gedächtnislos beschrieben.⁷⁹

In Matrizen werden die Wahrscheinlichkeiten für das Eintreten bestimmter Ereignisse anhand der Betrachtung der rückliegenden Ereignisse gespeichert. Markov-Ketten können verschiedener Ordnung sein: Eine Markov-Kette 1. Ordnung bezieht sich auf das unmittelbar vorige Ereignis, eine Markov-Kette 2. Ordnung auf die vorigen zwei usw. Je höher die Ordnung der Markov-Kette ist, desto höher ist auch die Wahrscheinlichkeit, dass das Nachfolgende dem vorherigen entspricht – desto weniger neu (und damit vermutlich kreativ) ist aber auch das Nachfolgende.

⁷⁸ Vgl. ebd., 65.

⁷⁹ Vgl. WALDMANN, Stochastische Modelle, 11.

David Cope hat ein Beispiel für Markov-Programme online gestellt, das einigermaßen musikalische Beispiele produziert. Die Matrizen können auch auf der Analyse existierender Musik basieren:

„The program collects note successions from a database and then lists these notes with their possible successors in sublists representing probabilities. Multiple note entries in a sublist indicate a higher probability that they will be chosen. A note’s absence from a sublist indicates that it will never be chosen as following grouping (zero probability). The resulting analysis can then be used by the Markov program to produce new music in a style roughly comparable to the original upon which it is based.“⁸⁰

Die Ergebnisse sind auf den ersten Blick erstaunlich. Tatsächlich verzeiht man dem Programm aber auch vieles. Das Interessante an solchen Markov-Ansätzen ist, dass das Produkt nur aufgrund von Wahrscheinlichkeiten in Abhängigkeit zu vorherigen Ereignissen erstellt wird. Auf eine Harmonielehre oder Syntax wird hierbei gänzlich verzichtet. Das ist plausibler, als es zunächst erscheinen mag: Ein Kind, das Sprechen lernt, lernt schließlich auch keine Grammatik, sondern imitiert Gehörtes.

Ein Beeinflussen von Markov-basierten Ausgaben würde der Markov-Eigenschaft widersprechen. Da eine solche Beeinflussung für die Komposition von Musik aber sinnvoll sein kann (etwa wenn eine generierte Phrase auf einem bestimmten Ton enden soll), entwickelte François Pachet mit seinem Team die sogenannten „Markov Constraints“.⁸¹

3.4 Künstliche neuronale Netze

Künstliche neuronale Netze abstrahieren die Nervenzellvernetzung im Gehirn oder Rückenmark in einem Software-Modell. In einem solchen Netzwerk gibt es Input-Neuronen zur Eingabe, versteckte Neuronen zur Bearbeitung und Output-Neuronen zur Ausgabe. Alle Neuronen werden

⁸⁰ COPE, *Computer Models of Musical Creativity*, 62.

⁸¹ Vgl. Kapitel 4.2.2.

miteinander verbunden, die Verbindungen werden zunächst zufällig gewichtet. Höher gewichtete Verbindungen werden für die Datenverarbeitung bevorzugt. Der „Lernprozess“ eines neuronalen Netzwerkes stellt sich folgendermaßen dar: Die Ausgaben des Programms werden mit gewünschten Ausgaben verglichen. Aus den Differenzen werden neue Gewichtungen für die Neuronen-Verbindungen errechnet. Dieser Prozess wird solange wiederholt, bis eine annähernd optimale Gewichtung erreicht wird. Vor allem in Hinblick auf komplexere Netze fallen entsprechend viele Daten an. In der künstlichen Intelligenz werden neuronale Netze häufig für Gesichtserkennung, Spracherkennung oder verwandte Prozesse genutzt.⁸²

Um einem neuronalen Netzwerk „das Komponieren beizubringen“, werden in der Regel Stücke in ähnlichem Stil an die Input-Neuronen gegeben und beispielhafte Ausgaben als *desired output* definiert. Die hierdurch entstandene Gewichtung wird in einem zweiten Schritt auf andere Eingaben angewandt. Die Ausgaben allerdings sind bisher weniger überzeugend, aber vor allem bei der Benutzung mit sehr unterschiedlichen Eingaben nicht uninteressant.⁸³

Da bei neuronalen Netzen nur die Ergebnisse bewertet werden können, die Entstehung derselbigen über die versteckten Neuronen aber nicht nachvollziehbar ist, hat Google im Rahmen seiner Bilderkennungssoftware eine Ausgabe entwickelt, die die erkannten Gegenstände (die in dem Fall den Output-Neuronen entsprechen) wieder über das eingegebene Bild legt: erkennt die Software auf einem eingegebenen Bild zum Beispiel einen Baum, überdeckt sie diese Stelle mit dem alternativen Bild eines Baumes. Das führte zu Landschaftsbildern, bei denen Büsche, Bäume, Wolken oder ähnliches fast ausschließlich mit Katzen oder Hunden überlagert wurden – wahrscheinlich aufgrund der enormen Vorkommnisse entsprechender

⁸² Vgl. SHIFFMAN, *The Nature of Code*, Chapter 10. Neural Networks.

⁸³ Vgl. COPE, *Computer Models of Musical Creativity*, 71.

Bilder im Internet. Dieses Ergebnis spricht nicht für die Zuverlässigkeit von neuronalen Netzen.⁸⁴

3.5 Evolutionäre Algorithmen

Evolutionäre Algorithmen werden in der Regel benutzt, um die Lösung zu einem komplexen Problem zu finden. Dabei werden Prinzipien der Genetik auf die Programmierung übertragen. Das, was in der Genetik die Population von Individuen ist, ist bei evolutionären Algorithmen eine sich ändernde Menge von möglichen Lösungen. Diese „Population“ wird einem wiederholten Zyklus von Evaluierung, Selektion und variiertes Reproduktion unterzogen:

- Die „Startpopulation“ wird entweder vom Benutzer definiert oder mehr oder weniger zufällig generiert.
- Jeder Lösungskandidat wird entsprechend einer zuvor definierten „Fitness-Funktion“ evaluiert.
- In der „Auswahlphase“ wird eine neue Menge von Lösungskandidaten durch Kopie der alten generiert; dabei werden solche Kandidaten, die zuvor als „fit“ evaluiert wurden, entsprechend häufiger kopiert. Hierdurch wird die Diversität der Population verringert.
- Auf einen Bruchteil der Lösungskandidaten werden Variations-steigernde Operatoren angewandt (im Sinne von Mutation oder Rekombination).
- Durch Wiederholung dieser Schritte wird die beste und durchschnittliche Fitness der Lösungskandidaten verbessert.⁸⁵

Schon sehr einfacher Code kann hierbei komplexe Ausgaben erzeugen.⁸⁶ Eine minimale Änderung hat typischerweise entscheidende Auswirkungen auf die künftige Entwicklung: „The system of the ,universe

⁸⁴ Vgl. HONEY, Wie ein künstliches Gehirn das Träumen lernt (zeit.de).

⁸⁵ Vgl. FERNÁNDEZ u.a., AI Methods in Algorithmic Composition, 546.

⁸⁶ Vgl. COPE, Computer Models of Musical Creativity, 66.

as a whole' is such that quite small errors in the initial conditions can have an overwhelming effect at a later time"⁸⁷, beschreibt Turing. Ein berechenbares Universum ist laut Cope auch die Grundlage von Wolframs Behauptungen, während er sich selber von einer solchen Universalität distanziert.⁸⁸

In gewisser Weise verwandt mit evolutionären Algorithmen ist die zelluläre Automatentheorie des Mathematikers John Conway aus dem Jahre 1970. Es handelt sich nicht um einen evolutionären Algorithmus im eigentlichen Sinne, allerdings demonstriert das Beispiel auf ähnliche Weise, wie schon wenige Regeln zu komplexen Verläufen führen können. Das von Conway entwickelte Brettspiel „Life“, bei dem Spielsteine Populationen darstellen, hat die folgenden drei Regeln:⁸⁹

1. Eine Population überlebt, wenn sie zwei oder drei Nachbarpopulationen hat.
2. Eine Population stirbt, wenn sie vier oder mehr Nachbarpopulationen hat (= Überbevölkerung) oder wenn sie gar keine Nachbarpopulation hat (= Isolation).
3. Eine neue Population entsteht auf einem leeren Spielfeld mit genau drei Nachbarn.

Der Spielverlauf ist zwar theoretisch berechenbar, aber mitnichten vorhersehbar. Im Internet gibt es etliche Java-Applikationen, die das Spielverhalten von Conway's Life nachbilden.

Durch die Imitation DNS-typischer Vererbungen von Charakteristiken sowie durch Kreuzungs- und Mutations-Techniken können evolutionäre Algorithmen zu noch komplexeren Resultaten führen. Der kreative Schein von evolutionären Algorithmen kann laut Cope allerdings trügerisch sein, da die Ergebnisse in ihrer Komplexität meistens denjenigen einfacherer

⁸⁷ TURING, *Computing Machinery and Intelligence*, 440.

⁸⁸ Vgl. COPE, *Computer Models of Musical Creativity*, 69.

⁸⁹ Vgl. GARDNER, *The fantastic combinations of „Life“*, 120f.

Algorithmen überlegen sind. Ein komplexeres Ergebnis muss aber nicht zwingend ein besseres Ergebnis sein.⁹⁰ Ein jüngeres Beispiel für künstliche Kreativität in der Komposition unter Verwendung eines evolutionären Algorithmus ist der Iamus genannte Computer-Cluster⁹¹.

3.6 Weitere Methoden

3.6.1 Fuzzy Logic

Während in der Informatik grundsätzlich nur die Zustände „an“ oder „aus“, dargestellt durch die boolesche Variable, existieren, können mit Fuzzy Logic auch alle Zustände dazwischen beschrieben werden. So gibt es also nicht nur „wahr“ oder „falsch“, sondern stattdessen Abstufungen, die den Grad der Wahrheit und den Grad der Falschheit angeben. Somit können auch relative Angaben, wie sie Menschen häufig verwenden („Ich hätte gerne eine *große* Tasse Kaffee mit ein *wenig* Milch“), vom Computer verarbeitet werden.

Cope beschreibt Fuzzy Logic als sehr hilfreich im Einsatz für künstliche Kompositionen.⁹² Ein möglicher Einsatz wäre der folgende: Im Rahmen von Modulationen werden Akkordfunktionen häufig umgedeutet. Diese Akkorde lassen sich mit Hilfe von Fuzzy Logic z.B. gleichzeitig als Dominantparallele der Ursprungstonart und als Mollsubdominante der Zieltonart beschreiben. Damit wird das „Vorhören“ der Zieltonart, das bei jedem Menschen unterschiedlich ist, in gewisser Weise simuliert. Die Vielfalt der Möglichkeiten wächst hierdurch enorm. Eine weitere Einsatzmöglichkeit ist daher auch die harmonische Analyse durch Fuzzy Logic.⁹³

⁹⁰ Vgl. COPE, Computer Models of Musical Creativity, 69.

⁹¹ Vgl. Kapitel 4.3.

⁹² Vgl. COPE, Computer Models of Musical Creativity, 73.

⁹³ Vgl. ebd.

3.6.2 Sonification

Bei der Sonification werden normalerweise nicht hörbare Daten in Klänge umgewandelt. Sie wird zum Beispiel eingesetzt, um elektromagnetische Strahlung hörbar zu machen (auch ein Geigerzähler könnte als primitiver Sonifikator bezeichnet werden). Da die Daten in der Regel keine inhärente musikalische Logik aufweisen, sind sie für künstliche musikalische Kompositionen nur bedingt brauchbar. Cope erwähnt allerdings, dass Komponisten Sonification teilweise trotzdem für kurze Passagen verwenden.⁹⁴

Lily Asquith, eine Physikerin vom CERN, hat die Messdaten des Large Hadron Colliders, die vermutlich das Higgs-Boson nachweisen, per Sonification in hörbare Samples umgewandelt. Interessant ist, dass hierbei durchaus eine übergeordnete Struktur erkennbar ist.⁹⁵ Sinnvoll ist das Vorgehen daher, da man mit dem Gehör besser Veränderungen wahrnehmen kann als mit dem Auge. Ein an diesem „LHCsound“ genannten Projekt teilnehmender Flötist und Komponist sagt zu den Hörproben:

„Wir können klare Strukturen aus dem Sound heraushören, es ist fast so, als wäre es eine Komposition. Die Partikel scheinen sich gegenseitig eine kleine Geschichte zu erzählen. Sie sind so dynamisch und wechseln die ganze Zeit, es klingt größtenteils wie Musik, die wir aus zeitgenössischen Kompositionen kennen.“⁹⁶

Die Sonification des Large Hadron Colliders weckt Assoziationen zur Sphärenmusik Pythagoras', der Vorstellung nicht hörbarer harmonischer Zusammenklänge, die aus der Bewegung der Himmelskörper und ihren Sphären entstehen.

⁹⁴ Vgl. ebd., 76.

⁹⁵ Vgl. ASQUITH, Listening to data from the Large Hadron Collider (YouTube).

⁹⁶ Vgl. DOBSON, RICHARD, zit. n. ZAUN, Quantisierte Sphärenmusik.

3.7 Bewertung

Die Frage, ob diese Methoden sich zur Simulation von Kreativität eignen, bleibt bestehen. Cope merkt hierzu an:

„Computer programs must be sufficiently independent of their programmers and users in order to qualify as truly creative. Most apparently creative algorithmic composing programs either produce enormous output from which users make preferential choices or invoke so many programmer-defined rules that the software only proves the creativity of the programmer. Therefore, none of the programs I have discussed here models creativity effectively.“⁹⁷

Er zitiert weiterhin Stravinsky, der beim Komponieren das Ähnliche dem Kontrastierenden vorzieht, wobei ersteres für Stravinsky durchaus schwerer zu finden, ihm dafür aber umso mehr wert sei.⁹⁸ Die Modellierung musikalischer Kreativität müsse, so Cope, also nach Ähnlichem suchen. Und um Ähnliches zu finden, müssen für den kreativen Prozess Beschränkungen (= *constraints*) auferlegt werden, die dazu führen, dass das Produkt dem Bestehenden ähnelt. Was sich zunächst zu widersprechen scheint, bestätigt Stravinsky, ebenfalls zitiert von Cope:

„My freedom will be so much the greater and more meaningful the more narrowly I limit my field of action and the more I surround myself with obstacles.“⁹⁹

Cope folgert daraus, dass solche Beschränkungen für die Modellierung künstlicher Kreativität von enormer Bedeutung sind. Ohne Beschränkungen hätten nämlich alle Entscheidungen den gleichen Wert, gleichwertige Entscheidungen allerdings setzen keine Kreativität voraus und seien damit genauso gut wie Pseudo-Zufallsgeneratoren.¹⁰⁰

⁹⁷ COPE, *Computer Models of Musical Creativity*, 79.

⁹⁸ Vgl. ebd., 80.

⁹⁹ STRAVINSKY, zit. n. COPE, *Computer Models of Musical Creativity*, 80.

¹⁰⁰ Vgl. COPE, *Computer Models of Musical Creativity*, 80.

Anhand einer Betrachtung der Skizzen zu einem Klavierkonzert Beethovens kann man einigermaßen nachvollziehen, wie ein kreativer Prozess bei dem Komponisten vonstatten ging: Auf der Suche nach einer unerwarteten Wendung in seiner Melodie, gibt Beethoven die Stabilität seiner ursprünglichen Idee zunächst auf. Nach einigen offenbar fehlgeführten Versuchen findet er sie schließlich in einer chromatischen Variante. Ein solcher Prozess lässt sich wohl kaum mit den oben beschriebenen Methoden modellieren.¹⁰¹ Cope bestätigt das zwar, behauptet aber dennoch, mit seiner induktiven Assoziation eine neue Methode entwickelt zu haben, die Beethovens Kreativitätsprozess ähnlich sei.¹⁰²

¹⁰¹ Vgl. ebd., 84.

¹⁰² Vgl. Kapitel 4.1.

4 Beispiele zu Computational Creativity in der Komposition

Neben dem bereits beschriebenen Überblick zu Methoden der algorithmischen Komposition folgen nun Beispiele, die diese Methoden nutzen, von denen einige beanspruchen, menschliche Kreativität zu modellieren.

David Cope war einer der ersten, die Computer zur automatischen Komposition von Musik benutzten. Seine Ansätze sind faszinierend und umstritten zugleich. Im Rahmen seiner Arbeit hat er umfangreiche Ergebnisse zur Erforschung eines musikalischen Stils geleistet, deren Einführung in den musikwissenschaftlichen Diskurs sicher von großer Bedeutung sein kann. Eine zentrale Technik seiner Software, die die menschliche Kreativität nachahmen soll, ist die sogenannte induktive Assoziation.

François Pachet leitet das Sony Computer Science Laboratory in Paris, das unter anderem als Mitglied der Vereinigung für Computational Creativity Beeindruckendes geschaffen hat. Das Konzept eines Continuator, einer Software, die eine musikalische Eingabe im selben Stil fortführt, wurde inzwischen auf die umfassenderen „Flow Machines“ übertragen, deren Anwendungsgebiet sich nicht auf die Musik allein beschränkt. Ein technisches Mittel hierbei sind sogenannte „Markov Constraints“, eine Weiterentwicklung der bis dato schlecht kontrollierbaren Markov-Ketten.

2012 sorgte ein Forschungsteam aus Malaga für Aufsehen: Ein Computer-Cluster namens Iamus sollte das erste Mal wirklich kreativ sein, ohne auf die Hilfe von Menschen angewiesen zu sein. Das London Symphony Orchestra ist mit der Aufführung Iamus' „Hello world“ das erste Orchester, das ein von Software komponiertes Stück aufführt. Die zugrunde liegende Technik namens „Melomics“ basiert auf evolutionären Algorithmen.

4.1 Emmy und Emily

David Cope's Computer Models of Musical Creativity

David Cope gilt als der Vorreiter der Computational Creativity in der Musik. Eine Schreibblockade bei einer Auftragskomposition bewegte den US-amerikanischen Theorie- und Kompositionsprofessor dazu, ein Computerprogramm zu schreiben, das ihn in der Komposition einer Oper unterstützt. Umfassende Forschung auf dem Gebiet musikalischer Kreativität führten zu verschiedenen Versionen dieses Programms: Emmy basiert hierbei fast ausschließlich auf der Rekombination von Musik, Emily, der Nachfolger, funktioniert nach dem Prinzip induktiver Assoziations-Netze. Letztere, so behauptet Cope, seien ein geeigneter Prozess, um menschliche Kreativität zu simulieren.

4.1.1 Emmy & die Rekombination von Musik

Da in klassischer westlicher tonaler Musik Regeln zu Stimmung, Melodie, Harmonie, Stimmführung, übergeordneter Form u.ä. existieren, die allgemeinbekannt seien, sei es ein naheliegender Schritt, ein Computerprogramm zu schreiben, das all diese Regeln kenne und anwende. Dieser Ansatz produziere zwar statistisch korrekte, aber fade Imitationen.¹⁰³ Emmy, eine Software, die eigentlich „Experiments in Musical Intelligence“ heißt und 1981 von David Cope entwickelt wurde, verfolgt daher nicht (ausschließlich) den Ansatz einer regelbasierten Programmierung: die Software ist datengestützt und basiert auf der Rekombination dieser Daten.

„I often describe the program as a large database of human-composed music upon which sits a smaller, but robust, analytical system that then feeds its information to a small composing program. Data is therefore most critical, analysis second, and composing third.“¹⁰⁴

¹⁰³ Vgl. COPE, Computer Models of Musical Creativity, 88.

¹⁰⁴ COPE, The well-programmed clavier, 20.

Cope erklärt sein Vorgehen damit, dass die Rekombination ein universaler Prozess sei: Die großen Werke der englischsprachigen Literatur basierten schließlich auf der Rekombination von Wörtern (und die Wörter ihrerseits auf der Rekombination von Buchstaben des Alphabets). Genauso verhalte es sich auch mit den großen Werken der westlichen Musik, bei der an Stelle des Alphabets eben die 12 Töne der wohltemperierten Stimmung stünden.¹⁰⁵ Darüber, ob Komposition tatsächlich nur eine neue Zusammensetzung existierender Teilstücke ist, lässt sich trefflich streiten.

Der Rekombinations-Prozess von Cope ist allerdings nicht mit einer modernen Version des eingangs erwähnten musikalischen Würfelspiels Mozarts gleichzusetzen: So handle es sich nicht um die Fragmentierung und willkürliche Neuordnung existierender Werke (das nämlich produziere „almost certainly [...] gibberish“¹⁰⁶), sondern um eine aufwändige musikalische Analyse und vorsichtige Fragmentierung auf der elementarsten Ebene.¹⁰⁷ Cope glaubt, auf dieser elementaren Ebene Anweisungen zu finden, wie man andere, aber eng verwandte Nachbildungen eines Werkes erschaffen kann.¹⁰⁸ Mit diesen Anweisungen ist wahrscheinlich der Stil eines Werkes gemeint. Das Auslesen dieses Stils ist auch der Ansatz, den François Pachet im „Flow Machines“-Projekt verfolgt (s. nächstes Kapitel).

Die Vorgehensweise seiner Software beschreibt Cope anhand von drei sehr einfachen Melodien, wengleich sich dieses Vorgehen auch auf höherer Ebene eigne: Aus der Analyse der Melodien

C-D-E-D-C , C-D-E-F-G-A-H-C sowie C-D-E-C-F-E-D-C

etwa leite die Software Regeln über die erlaubten Tonfolgen ab. So ergeben sich als mögliche Folgetöne für das C ausschließlich das D, für das

¹⁰⁵ Vgl. COPE, *Computer Models of Musical Creativity*, 88f.

¹⁰⁶ Ebd., 89.

¹⁰⁷ Vgl. ebd.

¹⁰⁸ Vgl. ebd., 88.

D die Töne E und C, für das E die Töne D, F oder C und so weiter.¹⁰⁹ Der einzige Unterschied bei Harmonien gegenüber Melodien sei, dass sich Gruppen von Tonleiterstufen zu anderen Gruppen von Tonleiterstufen bewegen. Melodie und Harmonik müsse daher bei diesem Ansatz gar nicht getrennt werden. Neue Musik könne auf diese Weise aus einer Datenbank musikalischer Werke kreiert werden und dabei die diesen Werke innewohnenden Regeln beibehalten, ohne die Regeln zu kennen.¹¹⁰ Das Programm erschafft also sozusagen aus der Analyse der Daten eine eigene Harmonielehre, die diesen Daten entspricht. Auf diese Weise generiere Emmy Melodien, Harmonien, Rhythmen sowie übergeordnete Strukturen wie Phrasen, Sektionen und schließlich ganze Werke.¹¹¹

Da bei diesem Prozess teilweise auch stiltypische Elemente verloren gehen, sah Cope es als notwendig an, dass Emmy bestimmte Teile vor der Fragmentierung und Rekombination schütze. Diese von ihm als musikalische Signaturen bezeichneten Teile zeichnen sich dadurch aus, dass sie in mindestens zwei Werken eines Komponisten vorkommen, typischerweise ein- bis dreitaktig sind und aus einer Kombination von Melodie, Harmonie und Rhythmus bestehen.¹¹² Passende musikalische Signaturen werden von der Software en bloc in die neue Komposition integriert, und das, sollten sie kontextabhängig sein, an der passenden Stelle. Ein Beispiel für eine musikalische Signatur wäre z.B. eine Schlusskadenz, wie sie in J.S. Bachs Chorälen vorkommt.¹¹³

Eine neben der musikalischen Signatur weitere Eigenschaft, die wichtig sei, damit die Software nicht nur „uninteresting vanilla replications“¹¹⁴ erstelle, sei der Charakter bestehend aus einer Kombination von Rhythmus

¹⁰⁹ Vgl. COPE, *The well-programmed clavier*, 18.

¹¹⁰ Vgl. ebd.

¹¹¹ Vgl. ebd., 19.

¹¹² Vgl. COPE, *Computer Models of Musical Creativity*, 95.

¹¹³ Durch *pattern matching* hat Cope etwa eine Schlusskadenz gefunden, die in über 200 Bach-Chorälen vorkommt, vgl. ebd., 97.

¹¹⁴ COPE, *The well-programmed clavier*, 19.

und Textur. Als Textur bezeichnet er hierbei die Anzahl der gleichzeitig erklingenden Töne.¹¹⁵ Das Programm analysiere zunächst Textur und Charakter anhand der originalen Musik, bevor es diese Musik fragmentiere – so könne sichergestellt werden, dass Kontinuität sowie Änderungen von Textur und Charakter innerhalb der Rekombination an sinnvollen Positionen stattfinden. So wird zum Beispiel verhindert, dass sich Achtel- und Sechzehntel-Bewegungen innerhalb eines Mozart-ähnlichen Stücks nicht zu häufig abwechseln.¹¹⁶

Die Ausgabe von Emmy ist, wie bereits erwähnt, stark von der Auswahl der Stücke in der Datenbank abhängig. Je größer diese Datenbank sei, desto eher klinge die Ausgabe nach dem Original. Stücke mit ähnlichen Tonarten, Modi und Metren beeinflussen die Ausgabe ähnlich positiv wie Stücke mit ähnlichen Texturen und Formen.¹¹⁷

„Compositional recombance often produces convincing new music in the style of the music in its database. Whether these new works result from *creative* processes, however, remains to be seen. According to the definition of creativity offered in chapter 1, one could imagine that the selection processes described here are creative. However, because this program represents the coding of my own particular take on voice leading and structure, the Experiments in Musical Intelligence just described represents *my* creativity rather than my *program's* creativity.“¹¹⁸

4.1.2 Emily & (induktive) Assoziations-Netze

Wie schon zuvor beschrieben, behauptet Cope, dass existierende Computermodelle kein zufriedenstellendes kreatives Potenzial aufweisen.¹¹⁹ Zu abstrakt sind die theoretischen Methoden, zu wenig menschlich der be-

¹¹⁵ Vgl. ebd.

¹¹⁶ Vgl. COPE, *Computer Models of Musical Creativity*, 101.

¹¹⁷ Vgl. ebd., 103.

¹¹⁸ Ebd.

¹¹⁹ Vgl. Kapitel 3.7.

schriebene Prozess. Während die Ergebnisse lernender neuronaler Netze zwar in der Sprach- oder Bilderkennung von großem Nutzen sind, taugen sie doch nur bedingt für die Simulation von Kreativität. Dennoch fühlt man sich bei der Betrachtung einer solchen lernenden Simulation stark an das Lernverhalten eines Kindes erinnert: Das liegt gewiss unter anderem an dem häufigen Scheitern und den unbeholfenen ersten Gehversuchen einer solchen Software. Cope mahnt schon in seinem Methoden-Überblick, dass man sich nicht von der Komplexität und dem versteckten Lernprozess neuronaler Netze täuschen lassen solle.¹²⁰

Er bezeichnet Kreativität als

„the initialization of connections between two or more multifaceted things, ideas, or phenomena hitherto not otherwise considered actively connected.“¹²¹

Damit liegt er in der Definition relativ nahe an Koestlers Konzept der Bisoziation¹²². Um diese Kreativität zu simulieren, bemüht er unter anderem Netzwerke, die er selber „association networks“ nennt und dabei von „semantic networks“ abgrenzt, da semantische Netze keine Gewichtung aufweisen.¹²³ Dieser Abgrenzung widerspricht Wiggins: tatsächlich seien Gewichtungen in semantischen Netzwerken existent. Das, was Cope Assoziations-Netze nennt, falle demnach unter den Oberbegriff der semantischen Netze.¹²⁴

Ziel der Assoziations-Netze von Cope sei das Lernen durch positive und negative Rückmeldungen ohne bewusste Anleitung.¹²⁵ In einer zunächst leeren Datenbank werden Benutzereingaben eingefügt und jegliche eigenständige Teile dieser Eingabe mit anderen eigenständigen Teilen verbun-

¹²⁰ Vgl. COPE, *Computer Models of Musical Creativity*, 69.

¹²¹ Ebd., 272f.

¹²² Vgl. Kapitel 2.1.3.

¹²³ Vgl. ebd., 274.

¹²⁴ Vgl. WIGGINS, *CMMC. A Review*, 113.

¹²⁵ Vgl. COPE, *Computer Models of Musical Creativity*, 272.

den. Die Eingaben seien hierbei Knotenpunkte (= *nodes*), die denen neuronaler Netze ähneln. Auch in Copes Assoziations-Netzen werden die Verbindungen zwischen diesen Knotenpunkten gewichtet.¹²⁶ Dennoch unterscheiden sich Assoziations-Netze laut Cope in ihrer Struktur stark von neuronalen Netzen: So gebe es bei ersteren unbeschränkte Knotenpunkte, bei letzteren hingegen seien diese zuvor festgelegt. Außerdem gebe es pro Verbindung in Assoziations-Netzen jeweils zwei Gewichtungen, für jede Richtung eine. Jeder Knotenpunkt könne zu jeder Zeit abgerufen werden, um eine Rückmeldung über die unterschiedlichen Gewichtungen zu geben. Es gibt also keine „hidden nodes“, wie sie in neuronalen Netzen vorkommen.¹²⁷ Cope erklärt einfache Assoziations-Netze anhand von Sprache. Interessant ist, dass aus den entstehenden Netzwerken entsprechend der Gewichtungen ihrer Verbindungen kausale Zusammenhänge abgelesen werden können. Demonstriert wird das anhand eines Dialoges mit einem solchen Assoziations-Netzwerk (U = Benutzer, A = Assoziations-Netzwerk):

„U: hello! / U: what is your name? / A: your what is hello! / U: my name is david / A: name? / U: your name is computer! / A: name? / U: what is your name? / A: computer! / U: what is my name? / A: david!“¹²⁸

Die Beziehung zwischen „name“, „my“ und „david“ sowie die Beziehung zwischen „name“, „your“ und „computer“ lässt sich in dieser Form auch im entsprechenden Assoziations-Netzwerk wiederfinden.¹²⁹ Wiggins kritisiert, dass die Abbildungen, die diese Netze illustrieren sollen, mangelhaft seien:

¹²⁶ Vgl. ebd., 274.

¹²⁷ Vgl. ebd.

¹²⁸ Ebd., 276.

¹²⁹ Vgl. ebd., 280.

„There are no labels on the network, and there are no weights. In short, these diagrams are utterly meaningless and can only be there to create an impression of technical content.“¹³⁰

Um die Gewichtung des Netzwerkes zu beeinflussen, führt Cope verschiedene Zeichen ein, die man in der Interaktion mit demselbigen benutzen kann: Ein Sternchen (*) hinter einem Wort beeinflusst die letzte gebildete Assoziation der Software negativ in dem Sinne, dass die Gewichtung stärker abgeschwächt wird, ein Pfeil (^) hingegen hat genau die gegenteilige, positive Auswirkung auf eine Gewichtung.¹³¹ So beeinflusst Cope den obigen Dialog folgendermaßen:

„U: what is your name? / A: computer! / U: what is my name? / A: david! / U: no* / A: * / U: what is my name? / A: is computer!“¹³²

Die zuvor richtige Antwort „david“ wird durch die Benutzereingabe negiert und die dadurch ausgelöste neue Gewichtung führt letztendlich zu der eigentlich falschen Antwort „is computer!“. Diese Methode kann dazu genutzt werden, den angeblichen Lernprozess des Netzwerkes zu beschleunigen. Die Pädagogik würde vermuten, dass das Hinzufügen einer solch autoritären Methode eher das Gegenteil von Kreativität zur Folge hat. Auch Cope bestätigt, dass keiner der bisher aufgezeigten Prozesse Kreativität darstelle.¹³³ Um dies zu erreichen, erweitert er das Assoziations-Netzwerk um die sogenannte induktive Assoziation (*inductive association*):

Das bisherige Netzwerk funktioniert rein deduktiv und antwortet jeweils mit den Assoziationen der höchsten Gewichtung. Für Kreativität allerdings sei freie Assoziation nötig. Um diese nachzubilden, erweitert er sein Assoziations-Netz um die sogenannte induktive Assoziation¹³⁴, die er folgendermaßen erklärt: Ein &-Symbol, das einer Benutzereingabe voran-

¹³⁰ WIGGINS, CMMC. A Review, 114.

¹³¹ Vgl. COPE, Computer Models of Musical Creativity, 287.

¹³² Ebd., 286.

¹³³ Vgl. ebd., 287.

¹³⁴ Vgl. ebd., 287f.

gestellt wird, fordert das Netzwerk dazu auf, nicht die naheliegende Assoziation auszugeben, sondern stattdessen einen angemessenen Ersatz zu suchen. Dieses „reasonable substitute“¹³⁵ wird gefunden, indem alle assoziierten Wörter auf die höchste umgekehrte Gewichtung durchsucht werden. Oder anders gesagt: Während bisher nur die einseitige Gewichtung zwischen zwei Verbindungen eine Rolle spielte, werden bei der induktiven Assoziation die Gewichtungen beider Richtungen addiert. Die höchste addierte Gewichtung, die nicht der ursprünglichen deduktiven Assoziation entspricht, gewinnt. Das Netzwerk denkt sozusagen nach Aufforderung einmal um die Ecke. Ob hiermit tatsächlich schon das erreicht ist, was Koestler mit Bisoziation meint (und was der Definition von Kreativität, die Cope anführt, entspricht), ist zu bezweifeln.

Das beschriebene (induktive) Assoziations-Netzwerk benutzt Cope im Folgenden mit Notennamen an Stelle von Wörtern. Statt Sätzen werden also nun Melodien eingegeben, das Netzwerk passt die Gewichtungen verschiedener Notennamen untereinander entsprechend dieser Eingabe an und gibt entweder die naheliegende Assoziation oder nach Aufforderung die induktive Assoziation aus. Das sieht dann beispielsweise so aus (für die bessere Lesbarkeit sind im Folgenden die Benutzereingaben (U) und die Computerausgaben (A) versetzt dargestellt):

„U: c d gb g a gb g?	
U: d c b e gb a g c c!	A: gb c d g a e b g?
U: &?	A: d gb c a g c!
U: &?	A: c gb a g d c!
U: &?	A: d gb c a g c!
U: &?	A: a gb c d g c!
U: &?	A: b gb c d g a c!“ ¹³⁶

Gut erkennbar ist hier die Aufforderung zur induktiven Assoziation durch die Benutzereingabe „&?“. Das Netzwerk antwortet wie gewünscht

¹³⁵ Ebd., 288.

¹³⁶ COPE, Computer Models of Musical Creativity, 306.

immer mit leichten Variationen nach dem zuvor erklärten, induktiven Prinzip. Cope vergleicht diese Computerausgaben mit dem kreativen Prozess Beethovens bei der Skizzierung seines (unvollendeten) Klavierkonzertes in D-Dur: Die Skizzen offenbaren eine stetigen Abwandlung der initialen Melodie bis hin zu dem Punkt, an dem der Komponist offenbar zufrieden ist.¹³⁷ Allein in der Betrachtung dieses Beispiels lässt sich aber erkennen, dass der Kreativitätsprozess Beethovens bei weitem umfassender ist: Es handelt sich nicht um das einfache Austauschen von Tönen, die auf ähnliche Weise passend sind, sondern um völlig andere Ansätze, mit denen die ursprüngliche Skizze erweitert wird.

Durch Rekursion kann das beschriebene Netzwerk Copes weiterhin automatisiert werden: Hierbei wird zunächst über die Benutzereingabe nach und nach eine Datenbank von beispielsweise kurzen Melodien eingegeben, ohne das Netzwerk antworten zu lassen. In einem zweiten Schritt wird die Ausgabe des Netzwerks rekursiv wieder zur Eingabe verwendet (indem „!“ am Ende der Ausgabe automatisch zu „?“ umgewandelt werden):

„The program then composes new music automatically by feeding its output into its input (with reversed punctuation) until a new work – consisting of all or most of the inputs and outputs – is complete.“¹³⁸

Den Prozess der induktiven Assoziation adaptiert Cope auch für Akkorde (mit beispielsweise einer Eingabe „ace“ für einen a-Moll-Akkord), mit zusätzlichen Zeichen werden außerdem Stimmtausch, Stimmführung und Transpositionen ermöglicht.¹³⁹ Die einfache lokale Induktion („&“) wird ergänzt um die regionale Induktion („&&“) und die globale Induktion („&&&“). Diese Erweiterungen ermöglichen die Induktion auch für Phrasen, Abschnitte oder ganze Teile.¹⁴⁰ Das grundsätzliche Konzept des Asso-

¹³⁷ Vgl. ebd., 303.

¹³⁸ Ebd., 307.

¹³⁹ Vgl. ebd., 309ff.

¹⁴⁰ Vgl. ebd., 313ff.

ziations-Netzes, das zur Ausgabe von Musik benutzt wird, bleibt aber bei all diesen Erweiterungen bestehen.

Ein solches Netzwerk nutzte Cope unter anderem für die automatische Anfertigung einer „10.“ Sinfonie Beethovens: Dafür befüllte er das Netzwerk mit den vorhandenen Skizzen Beethovens, einer musikwissenschaftlichen Komplettierung des ersten Satzes nach Barry Cooper, sowie einer großen Fülle von Chor- und Orchesterwerken Beethovens. Über Rekursion (Ausgabe = Eingabe) wurde die Datenbank an Chorwerken zuvor künstlich vergrößert.¹⁴¹ Das Ergebnis ist überraschend, durchaus ist eine Nähe zu den Werken Beethovens erkennbar (was ja aufgrund des Setups nicht unbedingt verwundert). Dennoch entspricht die Musik rein objektiv nicht der Genialität Beethovens. Wiggins kritisiert zu recht:

„[...] the claim that a piece of music based on the structure of Beethoven's fifth would be a good candidate for Beethoven's tenth is absurd, as listening to Beethoven's ninth immediately shows: Beethoven's style developed and changed considerably during his lifetime, so this notion just doesn't hold water. Either Cope is joking (but it's not very funny), or he's being musically naïve (which seems unlikely given his experience as a composer) or he's assuming his readers are musically naïve (which seems insulting). I don't know which.“¹⁴²

Emily, der Nachfolger von Emmy, basiert zu großen Teilen auf der Technik des oben beschriebenen Assoziations-Netzwerks. Andere in seinem Buch „Computer Models of Musical Creativity“ beschriebene Teilprogramme werden dabei mit diesem Netzwerk verbunden und führen letztendlich zu einer Software, die zu großen Teilen auf einer Benutzereingabe beruht: Über eine Schnittstelle kann sowohl über Wörter als auch über musikalische Eingaben mit dem Programm kommuniziert werden: „By encouraging and discouraging the program, Cope attempts to ‚teach‘ it to

¹⁴¹ Vgl. ebd., 318.

¹⁴² WIGGINS, CMMC. A Review, 114.

compose music more to his liking“¹⁴³. Die Aufgabe dieser neuen Programmversion ist also in gewisser Weise nur noch das Anbieten von Teilstücken einer Komposition. Die Entscheidung, welche Teile für die endgültige Komposition benutzt werden, trifft derjenige, der den Computer bedient. Dadurch, dass das Assoziations-Netzwerk „lernt“, werden zudem menschliche kreative Produkte in das System eingegeben.

Es ist bemerkenswert, dass Cope nach über 20 Jahren Forschung und Entwicklung im Bereich der computergestützten musikalischen Kreativität wieder dazu übergeht, eine Software auf einer solchen menschlichen Interaktion aufzubauen. Es ist zu vermuten, dass sich zumindest die Bewertung eines algorithmischen Produktes nicht vollständig automatisieren lässt.

¹⁴³ COPE, Emily Howell, Description (artsites.ucsc.edu).

4.2 Spiel es noch einmal, Sam!

Das Sony Computer Science Laboratory

Das Sony Computer Science Laboratory unter Leitung von François Pachet beschäftigt sich vor allem mit der Replikation von Stil. Der Stil wird hierbei in der Regel mit Markov-Ketten abgebildet. Da sich die Ausgabe von Markov-Ketten nicht beeinflussen lässt, entwickelte das Forschungsteam sogenannte „Markov Constraints“, bei denen sich auch zusätzliche Bedingungen aufstellen lassen.

4.2.1 Der Continuator

Markov-Sequenzen werden häufig in der künstlichen Intelligenz eingesetzt und haben dadurch auch ihren Weg in die künstliche Modellierung von Kreativität gefunden. Auch David Cope benutzt sie an vielen Stellen in seiner Software. Sie sind vor allem dann sinnvoll, wenn man Grund zu der Annahme hat, dass eine zeitlich begrenzte zurückliegende Sequenz Aufschluss über eine mögliche zukünftige Entwicklung gibt. So sagt die Markov-Eigenschaft aus, dass der künftige Zustand einer Sequenz ausschließlich vom letzten Zustand abhängig ist.¹⁴⁴

Der Continuator wurde 2002 von François Pachet und seinem Team vorgeschlagen, wenig später auch umgesetzt. Mit einem Markov-Modell werden musikalische Stileigenschaften wie Rhythmus, Beat, Harmonie und Ungenauigkeit dargestellt.¹⁴⁵ Über ein MIDI-Instrument (Keyboard o.ä.) spielt der Benutzer typischerweise eine Phrase. Diese Phrase wird in eine Sequenz von Symbolen abgewandelt, die „dimension or viewpoint of music“¹⁴⁶ ausdrücken, also etwa Tonhöhe, Dauer oder Tempo. Das System erlernt diese Sequenz, indem ein Modell der Übergangswahrscheinlichkeiten aufeinanderfolgender Symbole errechnet werde. Nachdem die Benut-

¹⁴⁴ Vgl. Kapitel 3.4.

¹⁴⁵ Vgl. PACHET, *The Continuator: Musical Interaction With Style*, 211.

¹⁴⁶ PACHET, *Markov constraints*, 149.

zereingabe beendet ist, generiert das System eine neue Phrase auf Basis des bis dahin errechneten Markov-Modells. Bei einer neuen Benutzer-Eingabe (z.B. durch Unterbrechung) wird das Markov-Modell aktualisiert. Hierbei zeichnete sich offenbar ein schrittweiser „Lernprozess“ ab, der zu „engaging dialogs“¹⁴⁷ sowohl mit professionellen Musikern als auch mit Kindern führte.¹⁴⁸

Die grundlegende Idee, mit Hilfe von Markov-Ketten aus vergangenen Wahrscheinlichkeiten Muster abzuleiten, ist nicht neu. Bei existierenden Modellen beobachtet Pachet die folgenden Dinge:

„1) Markov chain models (and their extensions, notably for variable-length) are able to represent efficiently musical patterns, but 2) there [sic!] generative power is limited due to the absence of long-term information. In another [sic!] words, these models can fool the listener on a short scale, but not for complete pieces.“¹⁴⁹

Das sei allerdings für den konkreten Anwendungsfall des Continuator weniger problematisch, da es sich um ein Interaktions-Modell handle, das ohnehin nur kurze Eingaben verarbeite. Ziel des Systems sei es lediglich, die Lücken zwischen den Benutzereingaben zu füllen.¹⁵⁰ Demnach ist es nicht bzw. nur bedingt möglich, dass das System am Ende einer Improvisationseinheit eine musikalische Phrase vom Anfang derselben zitiert. In Improvisationen wird eine solche Reminiszenz des Anfangs allerdings häufig als Stilmittel eingesetzt. Der Anwendungsfall des Continuator ist damit tatsächlich sehr beschränkt.

¹⁴⁷ Ebd.

¹⁴⁸ Vgl. ebd.

¹⁴⁹ PACHET, *The Continuator: Musical Interaction With Style*, 212.

¹⁵⁰ Vgl. ebd.

4.2.2 Markov-Constraints

Die Markov-Eigenschaft führt unweigerlich zu dem Problem, dass sich der künftige Zustand in keiner Weise beeinflussen lässt. Sucht man etwa eine musikalische Wendung, die zwar stilistisch auf dem letzten Zustand basiert, aber bestimmte zusätzliche Eigenschaften aufweist, kann man im Einsatz mit Markov-Ketten nur auf den Zufall warten, dass genau diese Eigenschaften erfüllt sind. Gewünschte Eigenschaften könnten beispielsweise globale Aufwärtsbewegungen zur Erfüllung einer bestimmten Melodie-Kontur oder das Enden einer Phrase auf einem bestimmten Ton sein. In der Komposition von Musik sind manche dieser Bedingungen abhängig vom Geschmack des Komponisten, andere sind wissensbasiert (etwa Regeln zur Stimmführung oder das Erfüllen einer Blues-Harmonie-Folge etc.).¹⁵¹ Es ist also anzunehmen, dass in der Komposition mehr als eine solche Bedingung aufgestellt wird – je mehr Bedingungen aufgestellt werden, desto unwahrscheinlicher ist es allerdings, mit normalen Markov-Ketten eine passende Sequenz zu finden.

Die von Pachet und seinem Team vorgeschlagene Lösung sind sogenannte Markov-Constraints, deren grundsätzliche Idee die Formulierung von Markov-Ketten als ein Constraint-Satisfaction-Problem (s. auch Kapitel 3.2.3) ist. Während bei Markov-Ketten der Zustand als Variable einer Funktion definiert wird, wird bei Markov-Constraints die Markov-Kette als solche als eine Variable definiert. Dadurch lässt sich eine Gleichung mit Bedingungen aufstellen und unter Beibehaltung der Markov-Eigenschaft lösen. Für eine ausführliche technische Beschreibung sei auf den entsprechenden Artikel Pachets verwiesen, da die Ausführungen hierzu sehr mathematisch sind und den Rahmen dieser Arbeit sprengen würden.¹⁵²

¹⁵¹ Vgl. PACHET, Markov constraints, 149f.

¹⁵² Vgl. ebd., 159ff.

Markov-Constraints führen zu interessanten neuen Möglichkeiten: Pachet hat auf diese Weise beispielsweise den „Boulez-Blues“ erstellen lassen: Ein Blues im Stile Charlie Parkers, der den Ansatz Boulez', dass jede Note einen anderen Akkord haben muss, aufgreift.¹⁵³ Eine andere Anwendung ist „Virtuoso“, ein Programm, das mit Informationen über typische Soli aus Bebop-Jazz- oder Hard-Rock-Musik befüllt wird und im Stile dieser Informationen in Echtzeit ein Solo improvisieren kann, das den Vorgaben des Benutzers entspricht.¹⁵⁴ Ein anderes Ergebnis dieser Forschung ist der „Flow Harmoniser“, der den Benutzer Leadsheets im Stil verschiedener Jazz-Musiker erstellen lässt. In einem Beispiel wird der Jazz-Standard „Giant Steps“ von John Coltrane im Stile Wagners harmonisiert.¹⁵⁵

4.2.3 Stil als zentrale kreative Eigenschaft

Die oben aufgeführten Forschungsergebnisse Pachets und seines Teams werden weitergehend im „Flow Machines“-Projekt erforscht. Im Sinne Csikszentmihalyis¹⁵⁶ soll die Computational Creativity dazu beitragen, den Flow-Zustand eines Komponisten, Malers oder Texters aufrecht zu erhalten, indem ständige Anreize zur Weiterentwicklung gegeben werden. Der Anspruch ist also weniger die künstliche Nachbildung menschlicher Kreativität, sondern die Erweiterung derselbigen.¹⁵⁷ Es wird mehrfach betont, dass Flow Machines keine kreativen Systeme per se sind, sondern interaktive Computerprogramme, die den Benutzer mit seinem eigenen Stil spielen lassen.¹⁵⁸

Eine dem Projekt zugrunde liegende Annahme ist die, dass nicht einzelne Werke oder Gegenstände im Mittelpunkt der Betrachtung von menschlicher Kreativität liegen sollen, sondern der Stil:

¹⁵³ Vgl. ebd., 155f.

¹⁵⁴ Vgl. ebd., 169.

¹⁵⁵ Vgl. GHEDINI u.a., *Creating Music and Texts with Flow Machines*, 338.

¹⁵⁶ Vgl. Kapitel 2.1.4.

¹⁵⁷ Vgl. PACHET u.a., *Creativity through Style Manipulation*, 1.

¹⁵⁸ Vgl. GHEDINI u.a., *Creating Music and Texts with Flow Machines*, 334.

„We argue that the core outputs of creative behavior lies in the creation of unique *series* rather than unique works of arts. What all these series have in common is commonly called *style*.“¹⁵⁹

Häufig verweist Pachet auf Skizzen, die die Entwicklung Picassos vom Realismus hin zu seinem linienbetonten Stil aufzeigen. Stil entstehe nach dem Erlernen der grundlegenden handwerklichen Fähigkeiten in Relation zum Zeitgeist und stelle damit eine Verbindung zwischen Persönlichkeit, kreativen Ausgaben und externen Einflüssen dar.¹⁶⁰ Aus diesem Grund haben Ghedini u.a. das Flow-Diagramm Csikszentmihalyis modifiziert und „style development“ als eine Erweiterung von „skills acquisition“ definiert.¹⁶¹

Das „Flow Machines“-Projekt wirft viele Fragen zur Verdinglichung von Stil beispielsweise als mathematisches Objekt auf. So müsse eine hohe Wahrscheinlichkeit innerhalb eines statistischen Modells (im Sinne von Markov-Ketten oder Markov-Constraints) nicht zwingend einen erkennbaren Stil darstellen. Pachet, Ghedini u.a. glauben daran, dass Stil-Manipulation im Wesentlichen zur Entwicklung von Stil und damit zur Entwicklung von Kreativität beiträgt.

Forschungsergebnisse im Bereich der musikalischen Kreativität offenbaren einen positiven Effekt in der Entwicklung von Kreativität bei un- ausgebildeten Kindern und jungen Pianisten unter Einsatz von computer- gestützten Systemen, wie sie oben beschrieben wurden.¹⁶² So wurden beispielsweise die Eingaben der Kinder allein durch die automatische Spiegelung des eigenen Stils immer komplexer: während sie anfangs mit einem Keyboard noch unbeholfener Cluster eingaben, entwickelten sich diese Eingaben im Laufe der Zeit zu immer komplexeren musikalischen Stücken.

¹⁵⁹ PACHET u.a., *Creativity through Style Manipulation*, 1.

¹⁶⁰ Vgl. GHEDINI u.a., *Creating Music and Texts with Flow Machines*, 330.

¹⁶¹ Vgl. ebd., 331.

¹⁶² Vgl. ebd., 338.

Das „Flow Machines“-Projekt konzentriert sich momentan auf diese computergestützte Untersuchung von Stil. Ghedini, Pachet u.a. nehmen an, dass die daraus folgenden Ergebnisse nicht nur einzelne künstlerische Produktionen verbessern können, sondern weitergehend zu einem besseren allgemeinen Verständnis des kreativen Prozesses beitragen können.¹⁶³

¹⁶³ Vgl. ebd.

4.3 Der Sohn des Apollo

Melomics

„Melomics“ startete als Forschungsprojekt, das teilweise vom „Spanish Research Programme“ finanziert und an der Universität von Málaga, namentlich in der „Grupo de Estudios en Biomimética“ entwickelt wurde. Seit 2012 existiert das Projekt als *University spin-off* unter gleichem Namen weiter. Der von den Forschern entwickelte evolutionäre Algorithmus trägt ebenfalls den Namen „Melomics“.¹⁶⁴

4.3.1 Iamus

704 Gigabyte Arbeitsspeicher, eine 70 Terrabyte große Festplatte sowie 352 AMD-Prozessoren – das ist die Antwort der künstlichen Intelligenzforschung auf Wolfgang Amadeus Mozart. Die „Mozart-Maschine“¹⁶⁵ soll die erste Software überhaupt sein, die gänzlich ohne menschliche Hilfe dazu in der Lage ist, umfassende Stücke zu komponieren. Die Mozart-Maschine heißt eigentlich Iamus. Der Name geht zurück auf den Sohn des griechischen Gottes Apollo, der in der Lage war, die Stimme der Vögel zu verstehen. Am 2. Juli 2012, anlässlich des hundertsten Geburtstags von Computerpionier Alan Turing, wurde die Musik von Iamus in einem Konzert mit international bekannten klassischen Musikern uraufgeführt.¹⁶⁶

Während die meisten Ansätze der künstlichen Komposition sich auf die Imitation existierender menschlicher Stile beschränken, sei Melomics durch Eingabe von Wissen über Kompositionen in der Lage, in einem eigenen Stil zu komponieren.¹⁶⁷ Anspruch des spanischen Forschungsteams ist es, den gesamten Prozess professioneller Musik-Komposition unter Einsatz von hochentwickelten Algorithmen und spezialisierter Hardware

¹⁶⁴ Vgl. Kapitel 4.3.2.

¹⁶⁵ STIELER, Die Mozart-Maschine, 26.

¹⁶⁶ Vgl. ebd.

¹⁶⁷ Vgl. QUINTANA u.a., Melomics: A Case-Study of AI in Spain, 100.

zu modellieren.¹⁶⁸ 2010 startete Iamus noch als Assistent, bei dem musikalische Bausteine Komponisten als Inspirationsquelle dienten:

„The Iamus currently is able not only to generate musical material of variable duration and instrumentation but also to notate it automatically using the MusicXML specification. This allows composers to work with professional scores, not just raw material, thus making the creative process of transforming, recombining and developing the source material more intuitive and musically satisfying.“¹⁶⁹

Über 10⁹ Kompositionen existieren bereits 2011, ein Jahr nach dem Start des Supercomputers, im „largest repository of music content in the world“¹⁷⁰. Diaz-Jerez erwähnt „Yshqur“, ein Konzert für Klavier und Orchester, in das von Iamus erstelltes Material einfließt. Der Melomics-Ansatz konstituiert eine neue Strategie zur Komposition von Musik: eine Strategie, die der Kreativität des Komponisten zur Verfügung gestellt werde.¹⁷¹

4.3.2 Evo-Devo als Kompositionsalgorithmus

Der zugrundeliegende genetische Algorithmus nennt sich „Melomics“, ein Kunstwort aus „melody“ und „genomics“. Diaz-Jerez beschreibt diesen Algorithmus als eine simulierte Evolution musikalischer Kompositionen. Natürliche Selektion formt die musikalische Struktur der Partituren zu höher entwickelten und komplexeren Kompositionen.¹⁷²

Evolutionäre Algorithmen¹⁷³ funktionieren nach dem Muster, dass eine Reihe von möglichen Lösungen ($\hat{=}$ eine Population von Individuen) wiederholt evaluiert werde ($\hat{=}$ Fitness im Sinne Darwins), ausgewählt und in

¹⁶⁸ Vgl. DIAZ-JEREZ, Composing with Melomics, 13.

¹⁶⁹ Ebd., 14.

¹⁷⁰ Ebd.

¹⁷¹ Vgl. ebd.

¹⁷² Vgl. ebd.

¹⁷³ Vgl. Kapitel 3.5.

neuer Variation ($\hat{=}$ Mutation und Kreuzung) reproduziert werde.¹⁷⁴ Ein typisches Problem solcher Algorithmen sei die Skalierbarkeit und die Struktur der Lösungen. So seien die meisten Lösungen unstrukturiert, schwer anpassbar und zerbrechlich.¹⁷⁵

Klassischerweise stellt bei evolutionären Algorithmen der Genotyp den Suchraum dar und der Phänotyp den Lösungsraum. Im Suchraum finden sich die kodierten Lösungen, im Lösungsraum die konkreten Lösungen. Das Problem ist, dass aufgrund mangelnder Rechenleistung häufig die Genotypen direkt auf die Phänotypen angewandt werden (*direct encoding*). Dabei wird die evolutionäre Entwicklung in gewisser Weise außer Acht gelassen.¹⁷⁶ Bei Melomics allerdings soll der Entwicklungsprozess mit berücksichtigt werden: In der evolutionären Entwicklungsbiologie („evo-devo“) werden evolutionäre Änderungen als kleine Mutationen im Genom interpretiert, die den Entwicklungsprozess auf komplexe Weise modulieren können. Unter Berücksichtigung dieses Entwicklungsprozesses können also auch bei evolutionären Algorithmen die Genotypen potenziell zu komplexen Phänotypen umgewandelt werden (*indirect encoding*). Auch zu komplexen Problemen können so Lösungen und komplexe Variationen dieser Lösungen entstehen.¹⁷⁷

Dieser Evo-Devo-Ansatz stellt den Kern von Iamus dar. Der Computer implementiere damit die Entwicklung komplexer musikalischer Strukturen, dargestellt durch künstliche Genome. Aus jedem dieser Genome könne ein Musikstück nach den Regeln der Entwicklungsbiologie entstehen. Wenn die Population von Kompositionen wachse, tragen Mutationen der Genome zu Änderungen der musikalischen Strukturen bei. Eine Fitness-Funktion evaluiere die Ergebnisse und bestimme, ob diese neuen Strukturen besser oder schlechter geeignet sind, musikalischen formalen Bedin-

¹⁷⁴ Vgl. QUINTANA u.a., Melomics: A Case-Study of AI in Spain, 100.

¹⁷⁵ Vgl. ebd.

¹⁷⁶ Vgl. ebd.

¹⁷⁷ Vgl. ebd.

gungen und ästhetischen Prinzipien zu entsprechen. Im Laufe der Evolution werde die korrespondierende Musik immer komplexer und passe sich den Selektionsbedingungen besser an.¹⁷⁸ Die Fitness-Funktionen seien unter Mitarbeit professioneller Musiker entwickelt worden. Außerdem seien grundlegende Bedingungen zur Instrumentierung (etwa das Komponieren von Klavier-Akkorden mit maximalen fünf Tönen pro Hand) aufgestellt worden. Mit insgesamt 1000 solcher Regeln werde die Umgebung definiert, in der korrekte Musik („correct music“) stattfinden kann.¹⁷⁹

„What is remarkable in Melomics’ music is that it allows the definition of new aesthetics, not imitating the style that any particular musician has previously developed. This distinguishes Melomics from previous attempts at algorithmic composition, making it a powerful tool for composers in any musical style.“¹⁸⁰

Mit Melomics¹⁰⁹ wurde der Melomics-Algorithmus auch auf die Komposition und Produktion von populärer Musik angewandt, indem die Vorgaben auf dafür typische Regeln abgeändert wurden. Entsprechende Beispiele lassen sich unter dem Stichwort „Melomics“ bei YouTube finden.

Da abgesehen von den beiden zitierten wissenschaftlichen Texten bislang nur journalistische Beiträge zu Iamus und Melomics vorliegen, ist eine genaue Bewertung des Algorithmus für Außenstehende derzeit nur sehr eingeschränkt möglich.

4.3.3 Melomics heute

Das University-Spin-Off kümmert sich heutzutage vor allem um die Vermarktung der Technik: Die Kommerzialisierung der aktuellen Generation von Computer-Komponisten ist das selbst gesteckte Ziel von Melomics Media. Hierzu verfolgen sie verschiedene Strategien: Möglich wäre

¹⁷⁸ Vgl. ebd., 101.

¹⁷⁹ Vgl. ebd., 102.

¹⁸⁰ Ebd.

eine kundenbasierte Musik-Generierung, die allerdings aus Gründen der begrenzten Rechenleistung noch eher in ferner Zukunft zu sehen ist. „Infinitunes“ ist ein Archiv computergenerierter Musik, das die meisten Genres und Formate abdeckt, durchsuchbar ist und per API¹⁸¹ auch von anderer Software benutzt werden kann. Ein mögliches Anwendungsbeispiel wäre die vom Melomics-Team vorgeschlagene „Melomics@life“-Applikation, die für jedes Szenario die passende Musik liefert: im Alltag, zur Musiktherapie oder zur Beschallung von Ladengeschäften.¹⁸²

„The royalty-free character of these songs makes them available worldwide, allowing for a wide range of freemium options (in contrast to the limited options that constrained profit margins allow to companies like Spotify and Pandora Media). However, it also opens possibilities to implement sound trademarking by matching catchy tunes to products or brands, and including them (by inserting their genes) into bigger music pieces.“¹⁸³

Die Entwicklung des komplexen evolutionsbiologischen Algorithmus endet also voraussichtlich als kostengünstigere Version von „Muzak“: als Gebrauchsmusik im Fahrstuhl.

¹⁸¹ „Application Programming Interface“, dt. Programmierschnittstelle.

¹⁸² Vgl. ebd.

¹⁸³ Ebd.

5 Einschätzung und Fazit

In dieser Arbeit wurden zunächst verschiedene Modelle zum Kreativitätsprozess beschrieben. Die Illuminations-Phase aus dem Vier-Phasen-Modell von Wallas, Guilfords „divergent thinking“ sowie Koestlers Konzept der Bisoziation widersprechen sich grundsätzlich nicht. Koestlers Beschreibung der Verbindung per se nicht zusammengehöriger Ebenen ist auch in Anbetracht einer möglichen Computersimulation dabei für diese Arbeit am interessantesten.

Wiggins u.a. entwickelten ein quantitatives Modell, die „Information Dynamics of Thinking“ (IDyOT): Tausende Generatoren treffen Vorhersagen über Teilgebiete der Realität und diese Vorhersagen werden, sollten sie zusammenpassen, an das Bewusstsein weitergegeben. Durch leerlaufende Generatoren können auch Vorhersagen über Teile aus dem Gedächtnis getroffen werden und zum „Global Workspace“, dem Bewusstsein, beitragen. Ist die Kombination von Wahrscheinlichkeitsvorhersagen mit dem Konzept der Bisoziation vereinbar? Es handelt sich zumindest auf den ersten Blick um ein plausibel erscheinendes Modell. Allerdings ist es eine ungeklärte – und für die Möglichkeit einer Computersimulation zentrale – Frage, wie viele Generatoren gleichzeitig tätig sein müssten, um menschliche Kreativität zu simulieren. So hilfreich dieses Modell für ein grundlegendes theoretisches Verständnis sein könnte, so wenig eignet es sich aber für die tatsächliche Imitation von Kreativität.

Vergleichbar ist ein solches Modell vielleicht mit der Entwicklung der virtuellen Realität: In Computerspielen werden Regeln der Physik auf immer komplexere Weise implementiert, dennoch würde niemand behaupten, dass dadurch eine zweite Realität geschaffen wird.

Trotzdem behauptet David Cope, seine Software könne als kreativ bezeichnet werden. Seine induktiven Assoziations-Netze sollen sogar ver-

gleichbar sein mit dem Kreativitätsprozess Beethovens. Durch die Induktion wird nicht die naheliegende Assoziation, sondern sozusagen die Assoziation der Assoziation für die Lösung eines Problems bemüht. Der Ansatz zeigt erstaunliche Ergebnisse und ist sicher geeignet, um etwa naheliegende Alternativen für bestimmte Akkorde zu finden. Diese Alternativen können sicherlich auch inspirierend sein. Dennoch reicht diese Assoziation „um die Ecke“ nicht aus, um einen tatsächlich bisoziativen Prozess im Sinne Koestlers darzustellen. Letztendlich bleibt es doch bei der Rekombination von kleinsten Bestandteilen, die zu etwas neuem zusammengesetzt werden.

Der evolutionsbiologische Ansatz von Iamus ist durchaus interessant. Nicht zuletzt, weil die künstliche Intelligenz mit solchen Ansätzen sehr komplexe Probleme zu lösen in der Lage ist. Aber wie wahrscheinlich ist es, dass sich die Regeln der Entwicklungsbiologie 1:1 auf die Musik-Komposition übertragen lassen – zumal die Evolutionsbiologie ihrerseits ja auch nur ein Modell darstellt. Das Forschungsteam ist stolz, dass Iamus funktioniert, ohne dass man der Software einen bestimmten Stil vorgegeben hat. Offenbar wird sie aber durch die aufgestellten „Fitness-Funktionen“ doch so sehr gelenkt, dass sie ausschließlich zeitgenössische Musik komponiert. Allein dieses Faktum zeigt, dass der Iamus-Computer nicht eigenständig kreativ sein kann: Zwar lassen sich die Kompositionen des „Melomics“-Algorithmus keineswegs vorhersagen. Doch gibt es einen entscheidenden Unterschied zu menschlich kreativen Prozessen: der Iamus-Computer konzentriert sich ausschließlich auf das musikalische Material. Eine Inspiration durch die Umwelt beispielsweise ist technisch überhaupt nicht möglich. Es ist auch nicht davon auszugehen, dass Komponisten im Laufe ihres Lebens immer wieder kleinste musikalische Bausteine generieren, die sie unterbewusst auf ihre „Fitness“ untersuchen, um sie irgendwann, nach Jahren der unterbewussten Weiterentwicklung zu einer Komposition zusammensetzen.

Computer sind mathematische Maschinen, die durch Software gelenkt werden können, um bestimmte Aufgaben zu erledigen. Wer sich mit der Programmierung von Software beschäftigt hat, weiß, dass Programmierung ausschließlich assoziativ funktioniert: Variablen, Funktionen und Objekte werden untereinander verknüpft und schließlich durch eine Befehls-eingabe aktiviert. Auch wenn ein Programm weitestgehend automatisiert abläuft, läuft es dabei doch immer wieder durch die gleichen Schleifen, Funktionen und speichert die Daten in den selben Variablen. Dieses starke assoziative Verhalten widerspricht per se dem kreativen Prozess, wie er von Koestler vorgeschlagen wird.

Der Ansatz der „Flow Machines“ ist da schon realistischer: Ziel ist die Erstellung von Software, die einen „Flow“-Zustand aufrecht erhält. Das tut sie, indem sie den (musikalischen) Stil aus einer kurzen Eingabe extrahiert und Variationen dieses Stils anbietet. François Pachet betont immer wieder, dass die Maschinen selbst nicht kreativ sind, sondern dass es deren einzige Aufgabe sei, die menschliche Kreativität zu spiegeln. Die Ergebnisse sind sehr überzeugend und tragen sicherlich dazu bei, menschliche Kreativität zu erweitern – allein dadurch, dass sich der eigene Stil spielerisch immer wieder auf neue Fälle anwenden lässt. Eine Bisoziation kann dadurch sicherlich ausgelöst werden: Nicht auf der Maschinenseite, sondern beim menschlichen Gegenüber.

Ein Forschungsdesiderat, das über diese Arbeit hinausgeht, ist die Frage, inwiefern die maschinelle Rezeption von Musik eine Rolle für die Komposition derselbigen spielt. Zwar hat auch Beethoven am Ende seines Lebens nicht mehr hören können, doch hat er mit seinen Kompositionen ja gewisse Emotionen ausdrücken wollen, die auf seinen eigenen Hörerfahrungen beruhten. Von einer Maschine, die nicht hören kann, kann man nach dieser Argumentation eigentlich nicht behaupten, dass sie komponieren kann. Ob es ausreicht, einen humanoiden Roboter in die Komische Oper zu schicken und ihm auf diese Weise beizubringen, was Emotion ist, wage ich zu bezweifeln.

Quellen- und Literaturverzeichnis

- AA.VV., Association for Computational Creativity: Computational Creativity. Zu finden unter: <http://computationalcreativity.net/home/about/computational-creativity/> (abgerufen am: 23.01.2016).
- AA.VV., Association for Computational Creativity: About. Zu finden unter: <http://computationalcreativity.net/home/about/computational-creativity/> (abgerufen am: 23.01.2016).
- ASQUITH, LILY, Listening to data from the Large Hadron Collider. Zu finden unter: <https://www.youtube.com/watch?v=iQiPytKHEwY> (abgerufen am 30.01.2016).
- BAARS, BERNARD J., A cognitive theory of consciousness, Cambridge 1988.
- BECKER, MADELE, Art. Creativity through history, in: RUNCO, MARK A. / PRITZKER, STEVEN R. (Hg.), Encyclopedia of Creativity, London ²2011.
- BLITSTEIN, RYAN, Triumph of the Cyborg Composer. Zu finden unter: <http://www.psmag.com/books-and-culture/triumph-of-the-cyborg-composer-8507> (abgerufen am 23.11.2016).
- BODEN, MARGARET A., The creative mind. Myths and mechanisms, London ²2004.
- COLTON, SIMON, The Painting Fool: About me. Zu finden unter: <http://www.thepaintingfool.com/index.html> (abgerufen am 24.01.2016).
- COLTON, SIMON, Creativity Versus the Perception of Creativity in Computational Systems, in: VENTURA, DAN / MAHER, MARY LOU / COLTON, SIMON (Hg.), Papers from the AAAI Spring Symposium, Menlo Park 2008, 14–20.
- COPE, DAVID, Computer Models of Musical Creativity, Cambridge 2005.

- COPE, DAVID, The well-programmed clavier: style in computer music composition, in: XRDS: Crossroads 19 (2013) Nr. 4, 16–20 (DOI: 10.1145/2460436.2460444).
- COPE, DAVID, Emily Howell, Description. Zu finden unter: <http://artsites.ucsc.edu/faculty/cope/Emily-howell.htm> (abgerufen am 02.02.2016).
- CSIKSZENTMIHALYI, MIHALY, The Concept of Flow, in: DERS., Flow and the Foundations of Positive Psychology, Dordrecht 2014, 239–263.
- DIAZ-JERES, GUSTAVO, Composing with Melomics: Delving into the Computational World for Musical Inspiration, in: Leonardo Music Journal 21 (2011), 13–14.
- FERNÁNDEZ, JOSE D. / VICO, FRANCISCO, AI Methods in Algorithmic Composition: A Comprehensive Survey, in: Journal of Artificial Intelligence Research 48 (2013), 513–582.
- GARDNER, MARTIN, The fantastic combinations of John Conway's new solitaire game „life“, in: Scientific American 223 (1970) Nr. 10 v. Oktober 1970, 120–123.
- GHEDINI, FIAMMETTA / PACHET, FRANÇOIS / ROY, PIERRE, Creating Music and Texts with Flow Machines, in: COROZZA, GIOVANNI E. / AGNOLI, SERGIO (Hg.), Multidisciplinary Contributions to the Science of Creative Thinking, Singapur 2016, 325–343.
- GUTJAHR, ELISABETH, Der Mythos Kreativität oder die Erfindung des Selbstverständlichen, Berlin 1996.
- HONEY, CHRISTIAN, Wie ein künstliches Gehirn das Träumen lernt. Zu finden unter: <http://www.zeit.de/digital/internet/2015-07/neuronale-netzwerke-google-inception> (abgerufen am: 30.01.2016).
- KOESTLER, ARTHUR, The act of creation, London 1964.

- KURZWEILL, RAY, *The Singularity Is Near. When Humans Transcend Biology*, New York 2005.
- MITTENECKER, ERICH. Art. Informationstheorie, in: WIRTZ, MARKUS A. (Hg.), *Dorsch – Lexikon der Psychologie*, Bern ¹⁷2014, 780.
- PACHET, FRANÇOIS, *The Continuator: Musical Interaction With Style*, in: NORDAHL, MATS (Hg.), *Proc. of International Computer Music Conference*, Göteborg, 2002, 211–218.
- PACHET, FRANÇOIS / ROY, PIERRE, *Markov constraints: steerable generation of Markov sequences*, in: *Constraints* 16 (2011) Nr. 2, 148–172 (DOI: 10.1007/s10601-010-9101-4).
- PACHET, FRANÇOIS / ROY, PIERRE / GHEDINI, FIAMMETTA, *Creativity through Style Manipulation: the Flow Machines project* (Vortrag auf der Marconi Institute for Creativity Conference in Bologna, 2013).
- PIRICH, CAROLIN, *Ein Roboter lernt fühlen*, in: *Die Zeit* 70 (2015), Nr. 4 v. 22. Januar 2015. Zu finden unter: <http://www.zeit.de/2015/04/kuenstliche-intelligenz-roboter-gefuehle-musik> (abgerufen am: 27.12.2015).
- PIRICH, CAROLIN, *Können Roboter fühlen?* Zu finden unter: http://www.deutschlandradiokultur.de/kuenstliche-intelligenz-koennen-roboter-fuehlen.947.de.html?dram:article_id=276803 (abgerufen am: 27.12.2015).
- POOLE, DAVID L. / MACKWORTH, ALAN K., *Artificial Intelligence: Foundations of Computational Agents*, New York 2010.
- PREISER, SIEGFRIED, *Kreativitätsforschung* (= *Erträge der Forschung* Bd. 61), Darmstadt 1976.
- QUINTANA, CARLOS S. / ARCAS, FRANCISCO M. / MOLINA, DAVID A. u.a., *Melomics: A Case-Study of AI in Spain*, in: *AI Magazine* 34 (2013) Nr. 3, 99–103.

- RUNCO, MARK A., Art. Divergent Thinking, in: RUNCO, MARK A. / PRITZKER, STEVEN R. (Hg.), Encyclopedia of Creativity, London 2011.
- SHIFFMAN, DANIEL, The Nature of Code. Zu finden unter: <http://www.natureofcode.com> (abgerufen am 30.01.2016).
- STIELER, WOLFGANG, Die Mozart-Maschine, in: Technology Review 10 (2012), Nr. 12, 26–34.
- TURING, ALAN M., Computing Machinery and Intelligence, in: Mind, New Series 59 (1950), Nr. 236 v. Oktober 1950. 433–460.
- v. RANDOW, THOMAS, Was ist Informationstheorie? Zu finden unter: <http://www.zeit.de/1965/46/was-ist-informationstheorie> (abgerufen am: 05.02.2016).
- WALDMANN, KARL-HEINZ / STOCKER, ULRIKE M., Stochastische Modelle. Eine anwendungsorientierte Einführung, Berlin 2004.
- WALLAS, GRAHAM, The art of thought, New York 1926.
- WIGGINS, GERAINT A., Computer Models of Musical Creativity: A Review of Computer Models of Musical Creativity by David Cope, in: Literary and Linguistic Computing 23 (2008) Nr. 1 (DOI: 10.1093/lc/fqm025), 109–116.
- WIGGINS, GERAINT A. / BHATTACHARYA, JOYDEEP, Mind the gap: an attempt to bridge computational and neuroscientific approaches to study creativity, in: Frontiers in Human Neuroscience 8 (2014), 201400540 (DOI: 10.3389/fnhum.2014.00540).
- WIGGINS, GERAINT A. / FORTH, JAMIE, IDyOT: A Computational Theory of Creativity as Everyday Reasoning from Learned Information, in: BESOLD, TAREK R. / SCHORLEMMER, MARCO / SMAILL, ALAN, Computational Creativity Research: Towards Creative Machines, Paris 2015.

WIGGINS, GERAINT A. / TYACK, PETER / SCHARFF, CONSTANCE / ROHRMEIER, MARTIN, The evolutionary roots of creativity: mechanisms and motivations, in: Philosophical Transactions of the Royal Society of London B: Biological Sciences 370 (2015), 20140099 (DOI: 10.1098/rstb.2014.0099).

ZAUN, HARALD, Quantisierte Sphärenmusik, Telepolis am 24.02.2011. Zu finden unter: <http://www.heise.de/tp/artikel/34/34244/> (abgerufen am 30.01.2016).

Erklärung

Ich versichere, dass ich die Schriftliche Hausarbeit selbstständig verfasst habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Alle Stellen und Formulierungen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle als Entlehnung kenntlich gemacht. Die Erklärung ist mit Ort, Datum und Unterschrift zu versehen.

Ort, Datum und Unterschrift