

6. Technikentwicklung: Konzeption und Implementation von Applikationen

[...] un mariage de la musique avec la machine
la plus puissante au monde.

*Iannis Xenakis*¹

Dieses Kapitel beschreibt die technischen Entwicklungen, die im Rahmen der Studie entstanden sind. Bei diesen Entwicklungen handelt es sich um zwei Applikationen. Die erste Applikation dient der technikgestützten Tempovermittlung und ist damit ein Werkzeug, das bei der Ausführung von Musik verwendet wird. Die zweite Applikation wird im Kompositionsprozess eingesetzt und ist ein Hilfsmittel, um die zeitlichen Strukturen eines Stücks zu entwerfen.

Die Entwicklungszyklen beider Applikationen waren mit der Praxis eng verflochten: Künstlerische Anforderungen, Bedürfnisse der Komponist:innen und Idealvorstellungen von Arbeitsabläufen führten zur Ausarbeitung der Spezifika, die wiederum durch den Einsatz der Applikationen in der Praxis validiert, verbessert und weiterentwickelt werden konnten. Zudem ermöglichten die Applikationen, das künstlerische Potenzial der technikgestützten Tempovermittlung immer besser kennenzulernen und aus diesen Erkenntnissen heraus neue Anforderungen zu formulieren. Solche Zyklen, die zwischen künstlerischer Praxis und Technikentwicklung oszillieren, sind der Musik nicht fremd: So beschreibt z. B. Herbert Heyde den Entwicklungsprozess von (vornehmlich akustischen) Musikinstrumenten als Regelkreis, bei dem die Merkmale der Musikinstrumente verändert werden, um dadurch Anpassungen an musikalische, klangliche oder technische Anforderungen vorzunehmen. Entspricht ein Instrument diesen zeitlich und regional determinierten Anforderungen, wird es weiter in der gleichen Gestalt gebaut. Entspricht es ihnen nicht, wird es geändert oder scheidet allmählich aus der Musikpraxis aus.²

1 Iannis Xenakis: *Musiques formelles*, Paris: Richard-Masse 1963 (La Revue musicale: double numéro spécial 253 et 254), S. 166.

2 Herbert Heyde: *Grundlagen des natürlichen Systems der Musikinstrumente*, Leipzig: Deutscher Verlag für Musik 1975.

Die hier beschriebenen Applikationen hängen auch eng mit meiner eigenen künstlerischen Praxis zusammen, da ich gleichzeitig Entwickler und Anwender bin. Das bedeutet, dass die iterativen Prozesse, bei denen Anforderungen der Praxis in die Technikentwicklung einfließen und die neuen technischen Eigenschaften wiederum in der Praxis validiert werden, besonders schnell ablaufen, weil sie keine intersubjektive Kommunikation benötigen. Dass sich Künstler:innen ihre eigenen Werkzeuge erschaffen, indem sie Hard- oder Software entwickeln, ist keine Besonderheit in der Medienkunst und der elektronischen Musik. Da es sich bei der Technikentwicklung auch um einen kreativen Akt handelt, nicht weniger als beim Komponieren, ist es oft schwierig, eine strikte Trennung zwischen den »technischen« und »künstlerischen« Anteilen der Arbeit vorzunehmen.³ Ich war jedoch nie die einzige involvierte Person. Es waren immer andere Personen beteiligt: Musiker:innen, die mithilfe dieser Applikationen spielen, Komponist:innen, die diese Applikationen für ihre eigene Musik verwenden, und schließlich – als die Applikationen eine gewisse Größe und Komplexität erreicht hatten – auch noch weitere Programmierer, die zur Entwicklung dieser Applikationen beitragen.

6.1 Die Applikation für die Aufführung von Musik

Das in dieser Studie entwickelte Hilfsmittel für die technikgestützte Tempovermittlung ist die Applikation *PolytempoNetwork*. Diese Applikation stellt auf dem Computerbildschirm den Notentext dar und zeigt das Tempo an (Abb. 6.1). Alle Musiker:innen haben je eine Instanz dieser Applikation vor sich und für das Ensemblespiel können mehrere Instanzen in einem lokalen Netzwerk verbunden und so untereinander synchronisiert werden (Abb. 6.2).

2010 wurde ein erster Prototyp dieser Applikation in der Programmierumgebung *Max*⁴ entwickelt. Die Grundidee war, für die Aufführung von Instrumentalmusik mit elektroakustischem Zuspiel eine Alternative zum gängigen Click-Track zu finden. Bald wurde auch die Möglichkeit genutzt, mit verschiedenen Instanzen der Applikation gleichzeitig verschiedene Tempi anzuzeigen. In den Jahren 2010–13 wurde dieser Prototyp für vier künstlerische Projekte verwendet⁵ und damit seine konzeptuelle Tauglichkeit nachgewiesen. Seit 2014 wird eine platt-

3 Für eine interessante Untersuchung der wechselseitigen Beziehung von künstlerischer Praxis und technischer Entwicklung anhand von historischen und zeitgenössischen Fallstudien vgl. Dieter Daniels/Barbara U. Schmidt (Hg.): *Artists as Inventors, Inventors as Artists*, Ostfildern: Hatje Canz 2008.

4 <https://www.cycling74.com>.

5 Eigene Werke: *Solo für Klarinette* (2010) und *Trails I* (2011), Marc Kilchenmann: *Egregoros* (2012), Kilian Deissler: *Fusions-Fuge* (2013).

Abbildung 6.1: Bildschirmfoto der Applikation PolytempoNetwork.



Abbildung 6.2: Probesituation mit PolytempoNetwork.



formübergreifende Stand-Alone-Software entwickelt.⁶ Sie ist in C++ geschrieben und verwendet das Framework JUCE.⁷ Die Applikation kann frei heruntergeladen werden und ihr Quellcode ist offengelegt.⁸ Im Folgenden wird der Entwicklungsstand dieser Applikation im Jahr 2021 beschrieben.

6 Philippe Kocher: *Polytempo Network: A System for Technology-Assisted Conducting*, in: *Proceedings of the International Computer Music Conference*, Athens 2014, S. 532–535.

7 <https://juce.com>.

8 <https://github.com/philippekocher/polytempo>.

Bildschirmdarstellung und visuelle Tempovermittlung

Die Applikation erfüllt die Funktion eines »elektronischen Notenpults« (Abb. 6.3). Dazu lädt sie Graphik-Dateien in den Arbeitsspeicher und stellt beliebige Ausschnitte davon auf dem Bildschirm dar. Beim Inhalt dieser Graphik-Dateien kann es sich neben traditioneller Notenschrift auch um alternative Formate handeln wie z. B. als Bilder oder Texte dargestellte Spiel- oder Improvisationsanweisungen. Der dargestellte Ausschnitt kann programmgesteuert ausgewechselt werden. Damit lässt sich eine Art automatisches »Umblättern« erzielen.⁹

Das Tempo wird visuell vermittelt.¹⁰ Die Taktschläge werden in zwei Balken am linken und oberen Bildschirmrand angezeigt. Während der Dauer eines Taktschlags wird eine Bewegung suggeriert, indem eine schwarze Fläche den Balken allmählich ausfüllt und sich dann wieder zurückzieht. Dadurch entsteht ein Bewegungsmuster, das sich in stark abstrahierter Form an Dirigierfiguren anlehnt: Eine Abwärtsbewegung zeigt die Eins im Takt an, eine Seitwärtsbewegung die anderen Zählzeiten (Abb. 6.4). Damit wird bis zu einem gewissen Grad auch die metrische Hierarchie der Taktschläge wiedergegeben. Diese Form der visuellen, animierten Tempovermittlung besitzt im Gegensatz zur akustischen Tempovermittlung mittels Click-Track in der Praxis einige vorteilhafte Eigenschaften:

- Die visuelle Tempovermittlung ist geräuschlos. Damit wird sie weder bei leisen Passagen unbeabsichtigt für das Publikum hörbar noch bei lauten Passagen übertönt und damit für die Ausführenden unverständlich.
- Der Hörsinn der Ausführenden wird nicht zusätzlich beansprucht und bleibt damit frei für das Wahrnehmen des eigenen Instruments und der Instrumente der Mitmusiker:innen zum Abgleich von Klang, Intonation und Lautstärke.
- Tempobezogene Zeichen wie Gesten der Dirigentin, des Stimmführers oder der Kammermusikpartnerin nehmen Musiker üblicherweise visuell wahr.
- Da die Animation zur Tempodarstellung Dirigierfiguren ähnelt und sie neben dem Notentext erscheint, kann an der vertrauten Praxis angeknüpft werden, ein Dirigat aus den Augenwinkeln zu beobachten und ihm zu folgen.
- Die kontinuierliche Bewegung der Animation, im Gegensatz zum punktuellen Signal eines Click-Tracks oder eines Blinklichts, vereinfacht es, den Zeitpunkt des nächsten Taktschlages zu antizipieren. Diese Vorwegnahme des Taktes ist

⁹ Dazu hat sich Folgendes bewährt: Der Bildschirm wird horizontal in zwei Hälften geteilt und der Inhalt jeder Hälfte wird jeweils dann ausgetauscht, wenn dort nicht gelesen wird.

¹⁰ Die hier beschriebene visuelle Tempovermittlung ist die bevorzugte Form, daneben besteht auch die Möglichkeit, einen Click-Track über den Audio-Ausgang wiederzugeben oder rhythmisierte MIDI- oder OSC-Nachrichten auszusenden.

besonders vorteilhaft in Situationen, in denen das Tempo sich verändert (accelerando oder ritardando).

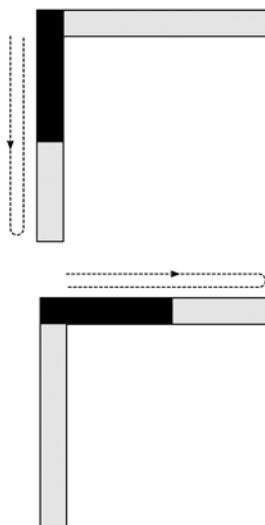
- Die Animation hat zudem den Vorteil, dass der Taktschlag eine geringfügige, aber wirkungsvolle ›Unschärfe‹ bekommt. Ein punktuelles Signal lässt die Ausführenden jede Abweichung vom richtigen Zeitpunkt unmittelbar als Fehler wahrnehmen, während eine kontinuierliche Bewegung weniger pedantisch wirkt und den Musiker:innen viel eher die Freiheit lässt, etwas vor oder hinter dem Schlag zu spielen. Damit wird eine musikalischere, weil entspanntere Darbietung gefördert.

In der Praxis zeigte sich, dass die meisten Musiker:innen der visuellen Tempovermittlung auf Anhieb mühelos folgen konnten und sich schnell daran gewöhnten. Das Spiel mit der visuellen Tempovermittlung wurde als angenehm bezeichnet und die vereinzelt negativen Rückmeldungen waren durch andere Faktoren bedingt, z. B. einen schlecht leserlichen Notentext auf einem zu kleinen Bildschirm oder eine grundsätzliche Ablehnung der Musizierpraxis mit computergestützter Tempoanzeige. Die meisten Musiker:innen begrüßten es, keinen Kopfhörer tragen zu müssen, wenn es auch einige wenige Personen gab, die dennoch einen herkömmlichen Click-Track bevorzugten, entweder weil sie besonders daran gewöhnt waren oder weil es ihnen unmöglich war, beim Spielen einen Bildschirm zu fixieren, wie das z. B. bei Schlagzeugern mit einem größeren Instrumentarium der Fall sein kann.

Abbildung 6.3: Ein Computerbildschirm auf einem Mikrofonstativ als ›elektronisches Notenpult‹.



Abbildung 6.4: Animierte Darstellung der Taktschläge: Zählzeit eins (oben), andere Zählzeiten (unten).



Per Knopfdruck kann das virtuelle Dirigat gestartet und gestoppt werden und es ist möglich, an jeder beliebigen Stelle im Stück einzusetzen. Durch die Eingabe eines Tempofaktors kann das Gesamttempo proportional verändert werden. Dies wird genutzt, um das Tempo einerseits beim Üben und Proben zu verlangsamen, andererseits in Aufführungssituationen, z. B. aufgrund der Raumakustik, anzupassen. Sind mehrere Instanzen dieser Applikation in einem lokalen Netzwerk verbunden, synchronisieren sie sich untereinander. Soll bei einer Ensembleprobe an einer bestimmten Stelle im Stück begonnen werden, wird der gewünschte Zeitpunkt an alle übermittelt, worauf jede Instanz den nächsten Taktanfang sucht – der bei einem Werk, bei dem die Instrumente in verschiedenen Taktarten oder Tempi spielen, durchaus für jede Instanz an einem anderen Ort liegen kann – und den Start des Dirigats so verzögert, dass die Ausführung der Musik synchron bleibt.

Das Event-System

Alle Aktionen, die die Applikation *PolytempoNetwork* ausführen kann, werden von Steuerbefehlen, sogenannten *Events*, kontrolliert. Das bedeutet, dass es für die Anzeige eines Taktschlags oder die Darstellung eines Ausschnitts aus einer Graphik-Datei usw. je einen entsprechenden Steuerbefehl gibt. Die Benutzer:innen haben verschiedene Möglichkeiten, diese Steuerbefehle zu verwenden. Eine erste Möglichkeit besteht darin, die Steuerbefehle mit einer Zeitangabe zu versehen und sie in einer Datei, einer Art ›elektronischen Partitur‹, abzuspeichern.¹¹ In dieser Weise kann der Ablauf eines Stücks festgehalten werden (Abb. 6.5).

Eine wichtige Eigenschaft dieses Systems von Steuerbefehlen ist, dass weitgehend auf Abstraktionen verzichtet wurde und sich jeder Steuerbefehl immer nur auf eine einzelne Aktion bezieht. Dies lässt sich gut an den Steuerbefehlen für die Taktschläge erläutern: Ein Takt muss immer aus einzelnen *beat*-Events zusammengesetzt werden, es gibt keinen Steuerbefehl für ganze Takte. Die Applikation ›weiß‹ nicht, was ein Takt ist, oder, technisch formuliert, es ist in der Applikation nicht festgelegt, wie das Konzept ›Takt‹ umgesetzt werden soll, d. h. wie eine ›Dirigierfigur‹ auszusehen hat und ob alle Taktschläge gleich lang, oder einer Agogik im Taktinnern unterworfen sind. Mit diesem Verzicht auf Abstraktionen wurde angestrebt, eine hohe Anschlussfähigkeit zu verschiedenen Arbeitsweisen zu gewährleisten. Im Beispiel der Taktschläge heißt das: Die Berechnung der absoluten Zeitpunkte der Taktschläge kann auf einem beliebigen, von den Benutzer:innen gewählten Verfahren zur Berechnung von Tempo, Tempoveränderung oder Agogik beruhen. Dazu kann die weiter unten beschriebene Applikation *Polytempo-*

11 Das dafür verwendete Format ist JSON (*JavaScript Object Notation*), das die Daten kompakt, strukturiert und in einer für Menschen lesbaren Textform darstellt.

Abbildung 6.5: Die »elektronische Partitur«, um die oben abgebildeten vier Takte aus Mussorgskys Bilder einer Ausstellung im Tempo ♩ = 60 zu dirigieren.

Moderato comodo assai e con delicatezza



```

{
  "init":{
    {"addRegion":{"regionID": 1, "rect":[0, 0, 1, 0.5]}},
    {"addRegion":{"regionID": 2, "rect":[0, 0.5, 1, 0.5]}},
    {"loadImage":{"imageID": 1, "url": "mussorgsky.png"}},
    {"addSection":{"sectionID": 1, "imageID": 1, "rect":[0, 0, 1, 0.3]}},
    {"addSection":{"sectionID": 2, "imageID": 1, "rect":[0, 0.3, 1, 0.3]}}
  },
  "default":{
    {"image":{"time": -1, "sectionID": 1, "regionID": 1}},
    {"image":{"time": -1, "sectionID": 2, "regionID": 2}},
    {"beat":{"time": -1, "duration": 1, "pattern": 21, "cue": 1}},
    {"marker":{"time": 0, "value": "1"}},
    {"beat":{"time": 0, "duration": 1, "pattern": 12}},
    {"beat":{"time": 1, "duration": 1, "pattern": 22}},
    {"beat":{"time": 2, "duration": 1, "pattern": 22}},
    {"beat":{"time": 3, "duration": 1, "pattern": 22}},
    {"beat":{"time": 4, "duration": 1, "pattern": 21}},
    {"marker":{"time": 5, "value": "2"}},
    {"beat":{"time": 5, "duration": 1, "pattern": 12}},
    {"beat":{"time": 6, "duration": 1, "pattern": 22}},
    {"beat":{"time": 7, "duration": 1, "pattern": 22}},
    {"beat":{"time": 8, "duration": 1, "pattern": 22}},
    {"beat":{"time": 9, "duration": 1, "pattern": 22}},
    {"beat":{"time": 10, "duration": 1, "pattern": 21}},
    {"marker":{"time": 11, "value": "3"}},
    {"beat":{"time": 11, "duration": 1, "pattern": 12}},
    {"beat":{"time": 12, "duration": 1, "pattern": 22}},
    {"beat":{"time": 13, "duration": 1, "pattern": 22}},
    {"beat":{"time": 14, "duration": 1, "pattern": 22}},
    {"beat":{"time": 15, "duration": 1, "pattern": 21}},
    {"marker":{"time": 16, "value": "4"}},
    {"beat":{"time": 16, "duration": 1, "pattern": 12}},
    {"beat":{"time": 17, "duration": 1, "pattern": 22}},
    {"beat":{"time": 18, "duration": 1, "pattern": 22}},
    {"beat":{"time": 19, "duration": 1, "pattern": 22}},
    {"beat":{"time": 20, "duration": 1, "pattern": 22}},
    {"beat":{"time": 21, "duration": 1, "pattern": 21}}
  ]
}

```

Composer verwendet werden, aber ebenso gut eine andere Arbeitsumgebung oder ein Skript in einer beliebigen Programmiersprache.

Eine zweite Möglichkeit für die Benutzer:innen, die Steuerbefehle zu verwenden, besteht darin, sie in Echtzeit über das lokale Netzwerk an die Applikation zu senden.¹² Damit wird es möglich, den zeitlichen Ablauf der Musik von einem externen Programm kontrollieren zu lassen, was sich z. B. für die Realisation von in Echtzeit generierter algorithmischer Musik anbietet. Kompositionen, bei denen

¹² Als Datenformat wird hier OSC (Open Sound Control) verwendet.

der zeitliche Ablauf nicht im Voraus festgelegt ist, werden als Werke mit ›offener‹ oder ›variabler‹ Form bezeichnet.

Offene Form

Die Geschichte der offenen Formmodelle reicht zu den musikalischen Würfelspielen zurück, die im 18. Jahrhundert eine Zeit lang populär waren. Dasjenige von Mozart ist in diesem Zusammenhang das am häufigsten zitierte. Weitere historische Beispiele finden sich erst wieder im 20. Jahrhundert. Zusammen mit anderen Tendenzen, die traditionelle tonale Formensprache aufzulösen, wurde auch die Abgeschlossenheit des musikalischen Werks in Frage gestellt. Umberto Eco lieferte mit seinem Buch *Das offene Kunstwerk* dazu einen wichtigen theoretischen Beitrag¹³ und spezifisch für die Musik wurde das Konzept der offenen Form von Konrad Boehmer eingehend behandelt.¹⁴ Eines der frühen Beispiele aus dem 20. Jahrhundert ist das *Mosaic Quartet* (1935) des amerikanischen Komponisten Henry Cowell, ein Streichquartett, bei dem es den Spielern erlaubt ist, die Reihenfolge der fünf kurzen Sätze nach eigenem Ermessen zu gestalten und sogar einzelne Sätze zu wiederholen. In den 1950er Jahren entstanden Kompositionen, bei denen auf einem Notenblatt mehrere Notationsfragmente oder Lesepfade zur Auswahl stehen, die in einer »vom Augen-Blick abhängigen Weise«¹⁵ bei der Aufführung ausgewählt werden, z. B. Morton Feldmans *Intermission 6* (1953), Karlheinz Stockhausens *Klavierstück XI* (1956) oder *Zyklus für einen Schlagzeuger* (1959). Wenn man davon ausgeht, dass die Komponisten tatsächlich den Zufall und nicht irgendwelche Verhaltensmuster der Ausführenden als generatives Prinzip anstrebten, wäre schon damals ein Bildschirm, auf dem der zu spielende Notentext aufscheint, möglicherweise das angemessene Darstellungsmedium gewesen. Bloß war damals die benötigte Technik weder weit genug entwickelt noch breit verfügbar.

Im Rahmen der vorliegenden Studie war es der Komponist André Meier, der sich ausgiebig damit auseinandergesetzt hat, wie mithilfe der Applikation *PolytempoNetwork* offene Formen realisiert werden können. Meier interessiert sich für die Frage, was mit einer Komposition passieren kann, wenn sie kein fixiertes Objekt ist, sondern ihre Form für jede Aufführung von einem Computer neu generiert wird. In einer Reihe von Kompositionen, die in den Jahren 2016–20 entstanden, näherte er sich dieser Frage zunehmend konsequenter an. Meier sieht seine kompositorische Arbeit darin, eine musikalische Umgebung oder Situation zu erzeugen, ein übergeordnetes, aber bloß imaginiertes Ganzes, von dem bei

13 Umberto Eco: *Das offene Kunstwerk*, Frankfurt am Main: Suhrkamp 1973.

14 Konrad Boehmer: *Zur Theorie der offenen Form in der neuen Musik*, Darmstadt: Tonos 1967.

15 Karlheinz Stockhausen: *Texte zu eigenen Werken, zur Kunst Anderer, Aktuelles. Aufsätze 1952–1962 zur musikalischen Praxis*, Köln: DuMont 1988, S. 69.

jeder Aufführung immer wieder eine andere Version ›beobachtet‹ werden kann. Dazu legt er eine formale Grammatik in der Form von Markow-Ketten fest, die die Übergangswahrscheinlichkeiten zwischen den einzelnen Abschnitten angeben. Die Kompositionsalgorithmen implementiert Meier in der Programmiersprache LISP innerhalb der Programmierumgebung *OpusModus*.¹⁶ Beim Ausführen dieser Algorithmen werden Steuerbefehle erzeugt und über eine Netzwerkverbindung an die Instanzen der Applikation *PolytempoNetwork* geschickt.

Zu einer ersten Annäherung an diese Formkonzeption kam es in der Komposition *the same [not] the same* für Flöte, Klarinette, Trompete, Klavier, Schlagzeug und Streichtrio. In dieser Komposition werden einzelne Passagen vom Computer zufällig angeordnet, aber der größte Teil ist in einer linearen Partitur – wenn auch mit einer gewissen Freiheit für die Ausführenden – notiert. In *modularform [#date]* für Ensemble war die Form des gesamten Stücks dem Kompositionsalgorithmus unterworfen. Die Musiker:innen erhielten 50 traditionell notierte Abschnitte im Voraus. Die Abschnitte sind meist 2–6 Takte lang, einige wenige Abschnitte sind mit bis zu 21 Takten deutlich länger. Aus diesem Material wird für die Aufführung eine zufällige Auswahl getroffen und den Musiker:innen auf dem Bildschirm dargestellt. Der Kompositionsalgorithmus erzeugt zufallsgeneriert auch die Tempi für die Abschnitte und die Art und Weise der Überlagerung zwischen den einzelnen Instrumenten des Ensembles. Bei *Machine Behaviour* für 6 Instrumente sind die Abschnitte kürzer und zahlreicher: Für jedes Instrument gibt es ca. 300 2/4-Takte, aus denen der Kompositionsalgorithmus die Musik zusammensetzt.

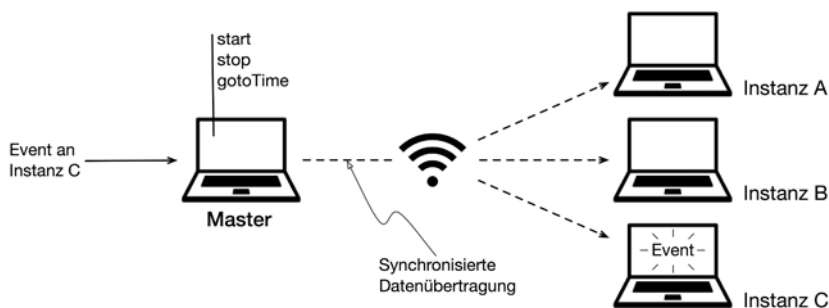
Netzwerk

Um Ensemblesmusik aufzuführen, können sich mehrere Instanzen der Applikation *PolytempoNetwork* über ein lokales Netzwerk untereinander synchronisieren. Dazu wurden die im Folgenden beschriebenen Funktionen implementiert: Einer der im Netzwerk verbundenen Applikationen wird die Funktion der Hauptinstanz (Master) zugewiesen. Dies kann die Applikation desjenigen Ensemblemitglieds sein, das die Probe oder die Aufführung leitet, oder es wird dafür ein zusätzlicher Computer verwendet, der z. B. bei der Tonregie steht. Wird nun die Master-Instanz gestartet oder gestoppt, folgen alle anderen nach. Damit allenfalls auftretende Netzwerklatenzen sich nicht störend bemerkbar machen, was insbesondere bei drahtlosen Netzwerken der Fall sein kann, synchronisieren alle Instanzen ihre internen Uhren und sowohl die von der Master-Instanz lokal ausgeführten wie auch die über das Netzwerk versendeten Steuerdaten werden mit einem Zeitstempel leicht in die Zukunft verschoben. Wie weit die Steuerdaten vorausdatiert werden müssen, hängt von der Übertragungsgeschwindigkeit des Netzwerks ab

¹⁶ <https://opusmodus.com>.

und reicht von einigen wenigen Millisekunden bis zu etwa einer halben Sekunde. Die Konsequenz davon ist, dass die Instanzen zwar nicht mehr unmittelbar, sondern leicht verzögert, aber dafür immer genau gleichzeitig reagieren. Diese Korrektur der Netzwerklatenz kann auch dann genutzt werden, wenn die Events in Echtzeit von einer externen Applikation generiert werden. In diesem Falle wird nicht diejenige Instanz, die den Event ausführen soll, direkt adressiert, sondern der Event wird an den Master geschickt und von diesem mit dem entsprechenden Zeitstempel versehen und weitergeleitet (Abb. 6.6).

Abbildung 6.6: Mehrere Instanzen synchronisieren sich in einem lokalen Netzwerk und folgen dem Starten und Stoppen der Master-Instanz. Zudem können Events von der Master-Instanz synchronisiert weitergeleitet werden.



Diese Applikation zur technikgestützten Tempovermittlung ist nicht an ein hoch spezialisiertes Setting gebunden, sondern nutzt die Tatsache, dass Digitaltechnik heute breit verfügbar ist. Dies bedeutet, dass die Musiker:innen mit dieser Applikation auch zu Hause üben können, was sich in der Praxis als sehr nützlich erwiesen hat. Um dieser breiten Anwendbarkeit noch weiter entgegenzukommen und auch ein Anwendungsszenario zu ermöglichen, bei dem nicht notwendigerweise eine Applikation heruntergeladen werden muss, wurde zusätzlich eine auf die wichtigsten Funktionen beschränkte Web-Applikation erstellt, die auf jedem Endgerät mit einem Webbrowser ausgeführt werden kann.

6.2 Die Applikation für die Komposition von Musik

Das in dieser Studie entwickelte Hilfsmittel für die Kompositionspraxis ist die Applikation *PolytempoComposer*. Ihr wichtigster Anwendungszweck besteht darin, eine »elektronische Partitur« für die Applikation *PolytempoNetwork* zu erstellen. Dazu werden die Taktarten und Tempi einer bereits bestehenden Komposition eingegeben und anschließend alle Taktschläge als entsprechend formatierte Datei ex-

portiert.¹⁷ Die Applikation *PolytempoComposer* kann aber ebenso dafür verwendet werden, die zeitliche Struktur einer Komposition grundsätzlich zu entwerfen. Die Hilfestellungen, die die Applikation den Komponist:innen dazu anbietet, beziehen sich insbesondere auf tempopolyphone Musik und die dort oft benötigte Konstruktion von Koinzidenzen, d. h. Stellen, an denen musikalische Ereignisse (meist Taktschläge) aus verschiedenen Temposchichten zeitlich zusammentreffen.

Für Komponist:innen ist es gerade bei tempopolyphoner Musik eine schwierige mentale Aufgabe, sich die vollständige, alle Temposchichten kombinierende Aufführung vorzustellen. Falls man beabsichtigt, die zeitliche Entwicklung der Musik genau unter Kontrolle zu behalten, muss man ausrechnen können, an welchen Stellen die musikalischen Ereignisse der verschiedenen Temposchichten zusammentreffen. Bei konstanten Tempi in einfachen Proportionen sind die hierzu benötigten Berechnungen trivial, es handelt sich um einfaches Bruchrechnen und verlangt von den Komponist:innen nicht mehr als die Bereitschaft, ein bisschen Arithmetik in ihre kompositorischen Denk- und Arbeitsabläufe einzubeziehen. Bei komplizierteren Tempoverhältnissen oder bei dynamischen Tempopolyphonien, bei denen die Temposchichten sich voneinander entkoppelt beschleunigen oder verlangsamen, werden die Berechnungen anspruchsvoll. Um eine Tempopolyphonie zu konstruieren, müssen diese Berechnungen auch nach einer Unbekannten aufgelöst werden können, z. B. um die richtigen Tempi zu finden, damit zwei Stellen in zwei verschiedenen Temposchichten zusammentreffen. Den Komponist:innen, die sich mit tempopolyphoner Musik befassen, wird mit der Applikation *PolytempoComposer* dafür ein Werkzeug an die Hand gegeben.

Tempomathematik

Um die Prinzipien einer Musik mit parallelen Tempi zu verstehen und diese Prinzipien für kompositorische Anwendungen zu operationalisieren, müssen zunächst die zugrunde liegenden mathematischen Gesetzmäßigkeiten geklärt werden. Die Frage lautet: Wie kann das zeitliche Voranschreiten der Musik in Bezug auf ihre Notation in rhythmischen Werten formal beschrieben werden? Als Annäherung an diese Frage bietet es sich an, in Entsprechung zur Kinematik die Größen Zeit, Ort und Geschwindigkeit (Tempo) zu verwenden.

17 Daneben existiert noch die Möglichkeit, die Zeitpunkte aller Taktschläge als neutrale Zahlenreihe zu exportieren, um sie auch in Arbeitsabläufen, bei denen andere Programme für Computermusik benutzt werden, weiterzuverwenden.

- *Zeit* bedarf keiner weiteren Erläuterung,¹⁸ sie wird in Sekunden gemessen. Eine Differenz von zwei Zeiten ist eine Dauer.
- *Ort* bezeichnet eine bestimmte Stelle in der symbolischen Repräsentation der Musik. Es ist eine abstrakte Einheit, mit der die zeitliche Abfolge der Ereignisse proportional festgelegt wird. Es liegt nahe, für diese Einheit Notenwerte zu verwenden, die im deutschen Sprachgebrauch ja als Bruchzahlen ausgedrückt werden. So kann z. B. der Ort der zweiten Viertelnote im dritten 4/4-Takt eindeutig und von jeglicher Metrik befreit als $2\frac{1}{4}$ oder 2.25 angegeben werden. Der Abstand zwischen zwei Orten im Notentext ist eine Strecke.¹⁹
- Für die Geschwindigkeit wird der in der Musik besser etablierte Begriff *Tempo* verwendet. Das Tempo entspricht der Strecke im Notentext pro Zeiteinheit.

Ein Mehrwert dieser Analogie zur Physik – noch bevor man zu rechnen beginnt – ist die Klarheit der Begrifflichkeit, die sie schafft. Gerade bei traditionell notierter Musik wird die Trennung zwischen zeit- und ortsbezogenen Bezeichnungen oft nur unscharf vorgenommen.²⁰ Es wäre jedoch wichtig zu beachten, dass ein Notenwert die *Länge* einer Note bezeichnet und nicht seine *Dauer*. Erst wenn ein Tempo dazukommt, z. B. $\text{♩} = 60$, bekommt eine Note mit der *Länge* von $\frac{1}{4}$ eine *Dauer* von einer Sekunde. Diese Unterscheidung mag spitzfindig wirken, aber ein klarer Sprachgebrauch kann eine große Hilfe sein, multitemporale Strukturen zu verstehen und beschreiben.

Die zur Quantisierung des Tempos üblicherweise verwendete Metronomzahl gibt die Anzahl Taktschläge pro Minute – englisch *beats per minute* (bpm) – an. Da viele Musikpraktiken das Konzept eines regelmäßigen Pulses kennen, ist der pragmatische Nutzen dieser Metronomzahlen unbestritten. Metrische Strukturen mit nicht-isochronen Pulsen, wie sie bei z. B. asymmetrischen oder wechselnden Taktarten vorkommen, können jedoch wegen der implizierten Regelmäßigkeit des Pulses weniger gut abgebildet werden. Auf jeden Fall muss zur Präzisierung dieser Metronomzahl noch ein Notenwert angegeben werden: *Viertel* gleich 60. Wird dieser Notenwert nun auch rechnerisch miteinbezogen, erhält man das *absolute Tempo*, eine Maßeinheit, die nicht als Wiederholungsrate eines

18 Um alle Missverständnisse auszuräumen, sollte möglicherweise doch darauf hingewiesen werden, dass Zeit hier als naturwissenschaftliche Größe verstanden wird. Alle im Zusammenhang mit Musik bisweilen herangezogenen philosophischen Konzepte einer ›gefühlten‹ oder ›erlebten‹ Zeit sind für mathematische Zwecke nicht brauchbar.

19 Hier ist *nicht* der Abstand zwischen den graphischen Symbolen der Notation gemeint, der in einem Längenmaß ausgedrückt wird. Für die Erörterung der Fragen zur tempopolyphonen Notation vgl. Kapitel 8.

20 Unglücklich sind auch Wortschöpfungen wie »symbolic score time« z. B. bei Henkjan Honing: *From Time to Time: The Representation of Timing and Tempo*, in: *Computer Music Journal* 25 (2001), S. 50–61.

Pulses definiert ist, sondern – losgelöst von Konzepten wie Puls und Metrum – als Quotient einer Strecke im Notentext und der dafür benötigten Dauer:

$$\frac{\text{Strecke}}{\text{Dauer}} = \text{Tempo} \quad \left| \quad \text{Beispiel } j = 96: \quad \frac{96 \times \frac{1}{4}}{1 \text{ min}} = \frac{24}{60 \text{ s}} = 0.4 \text{ s}^{-1} \right.$$

Diese Maßeinheit hat einen Nachteil: Musiker:innen können sich darunter kaum etwas vorstellen und werden in der Musikpraxis lieber bei der Metronomangabe $j = 96$ bleiben. Für jegliche Berechnungen hingegen, bei denen die verstrichene Zeit oder die zurückgelegte Partiturstrecke zwischen musikalischen Schichten mit verschiedenen Bezugsgrößen für den Taktschlag verglichen werden soll, ist dieses absolute Tempo unentbehrlich.

Koinzidenzen in parallelen Tempi

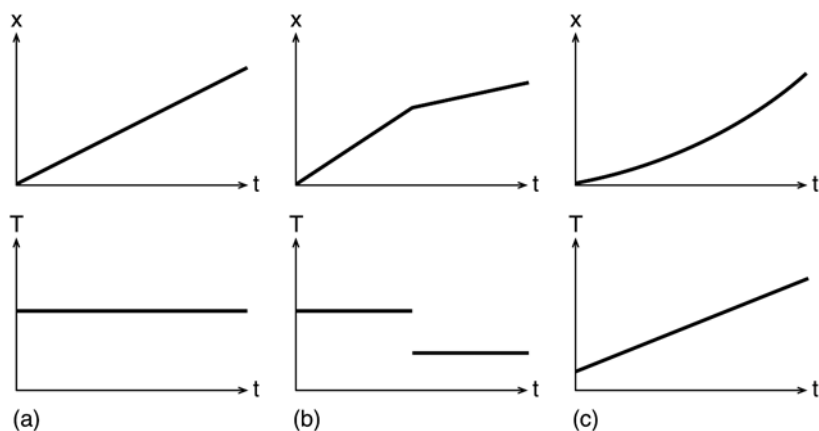
Eine Aufgabe, vor die sich Komponist:innen beim Konstruieren von Tempostrukturen oft gestellt finden, besteht im Ausrechnen von Koinzidenzen, d. h. von Orten in den einzelnen Temposchichten, die zum gleichen Zeitpunkt erreicht werden. Solchen Koinzidenzen können einen musikalischen Zweck haben, indem diesen Stellen gemeinsame Aktionen oder formale Einschnitte zugeordnet werden. Um Koinzidenzen zu *finden*, muss für jeden Ort im Notentext die genaue Zeit berechnet werden können, um Koinzidenzen zu *konstruieren*, müssen wahlweise Zeit, Ort oder Tempo als Unbekannte behandelt werden. Für konstante Tempi ist das trivial. Die oben genannte Formel gibt die Beziehung der drei Größen Zeit, Strecke und Tempo an, sind zwei davon gegeben, kann die dritte berechnet werden. Ungleich anspruchsvoller sind die Berechnungen, wenn auch Tempoverläufe vorkommen, d. h. unabhängig voneinander verlaufende Beschleunigungen oder Verlangsamungen.

Um dies zu erläutern, sei beispielhaft eine Denkaufgabe gestellt: Man stelle sich zwei verschieden eingestellte Metronome vor. Sie ticken unterschiedlich schnell, jedes in seinem eigenen Tempo. Stehen die beiden Tempi zueinander in einem rationalen Verhältnis, fällt immer nach einer bestimmten Periodendauer das Ticken der beiden Metronome zusammen. Nun soll das Tempo eines Metronoms allmählich verändert werden und zwar in einer Weise, dass es zu einem gegebenen Zeitpunkt nicht nur im selben Tempo wie das andere tickt, sondern auch phasengleich. Wie muss diese Tempoänderung vorgenommen werden, damit sich die gewünschte Koinzidenz ergibt? Wie schnell muss man am Metronom drehen, wie viel vor dem gewünschten Zeitpunkt muss man damit beginnen und soll man regelmäßig drehen oder z. B. langsamer beginnen und dann schneller werden? Mit anderen Worten: Es soll ein Tempoverlauf berechnet werden, der die vorgegebenen Bedingungen erfüllt. Ein Tempoverlauf besteht aus einem Start- und Endpunkt im Notentext, einem Start- und Endtempo und einer bestimmten Zeit-

dauer, während der dieser Tempoverlauf ausgeführt werden soll. Alle diese Parameter sind voneinander abhängig. Um diese Abhängigkeit zu verstehen, bietet es sich an, eine graphische Repräsentation zu Hilfe zu nehmen.

Es gibt zwei Möglichkeiten zu visualisieren, wie Musik zeitlich voranschreitet. Die erste Möglichkeit ist eine *tempo map*, ein Tempo-Zeit-Diagramm, das das Tempo als Funktion der Zeit darstellt, die zweite eine *time map*, ein Ort-Zeit-Diagramm, das den Ort im Notentext als Funktion der Zeit darstellt (Abb. 6.7). Beide Darstellungen lassen je eine Größe nur implizit erscheinen: In der *tempo map* ist dies die im Notentext zurückgelegte Strecke, die sich als das Integral unter der Kurve berechnen lässt, in der *time map* ist es das momentane Tempo, das durch die Steigung der Kurve ausgedrückt wird. Die *tempo map* ist die häufiger verwendete Visualisierung. Für viele Zwecke scheint sie naheliegender oder intuitiver zu sein, weil sie das Tempo und dessen Veränderung – das ›Drehen am Metronom‹ – als Wert der y-Achse direkt veranschaulicht. Die *time map* ist mehr aus der Interpretationsforschung bekannt, weil mit ihr die durch Rubato, Agogik usw. erzeugten Abweichungen von einem idealen Tempo gut als Verzerrung einer geraden Linie dargestellt werden können.

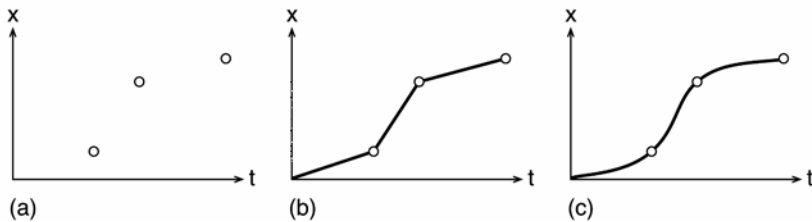
Abbildung 6.7: Eine Gegenüberstellung von drei *time maps* (oben) und den entsprechenden *tempo maps* (unten) für ein konstantes Tempo (a), einen abrupten Tempowechsel (b) und ein *Accelerando* (c).



Für den hier angestrebten Zweck der Festlegung von Koinzidenzen zwischen einzelnen Temposchichten ist die *time map* die zweckmäßigere Darstellung. Auf diesem Diagramm ist nämlich unmittelbar ersichtlich, wie sich der Ort im Notentext zur Zeit verhält. Soll nun ein bestimmter Ort im Notentext zu einem bestimmten Zeitpunkt erreicht werden, um dort mit einem Ereignis in einer anderen Temposchicht oder in einem Medienzuspiel eine Koinzidenz zu bilden, wird die entsprechende Stelle auf der *time map* markiert. Diese Markierungen werden

Kontrollpunkte genannt. Anschließend werden diese Kontrollpunkte mit Linien verbunden. Eine gerade Linie bezeichnet ein konstantes Tempo, eine gekrümmte Linie ein Tempo, das sich zwischen den Kontrollpunkten verändert (Abb. 6.8). Um eine im richtigen Maße gekrümmte Linie zu erhalten, wird eine parametrische Kurve benötigt, um die Kontrollpunkte zu verbinden. Die kubische Bézierkurve hat sich dafür als brauchbar erwiesen.²¹

Abbildung 6.8: Gegebene Punkte auf der time map (a) werden mit einer Linie verbunden, was zu abrupten Tempowechseln führt (b) oder zu einem kontinuierlichen, variablen Tempo (c).



In Abb. 6.9 ist zu sehen, dass eine kubische Bézierkurve durch vier Punkte P_0 , P_1 , P_2 und P_3 bestimmt wird. Die Kurve beginnt bei P_0 und endet bei P_3 , die zusätzlichen Punkte P_1 und P_2 dienen dazu, die Form der Kurve zu bestimmen, indem durch sie die Steigung am Anfang und am Ende festgelegt wird. Da auf einer *time map* die Steigung der Kurve dem Tempo entspricht, ist so die Möglichkeit gegeben, das momentane Tempo an den Kontrollpunkten zu bestimmen und damit das Anfangs- und Endtempo dieses Segments festzulegen. Die Form der Kurve und damit der Tempoverlauf zwischen den Kontrollpunkten ergibt sich automatisch durch die Bézierkurve, wobei es die folgenden Möglichkeiten gibt, die Form der Kurve zu beeinflussen: Der Abstand zwischen P_0 und P_1 bzw. zwischen P_2 und P_3 bestimmt, wie sehr das Tempo an den Kontrollpunkten gewichtet wird, d. h. wie schnell oder allmählich das Tempo verlassen bzw. erreicht wird. Dadurch kann, im Rahmen des Möglichen, sogar die gestische Qualität der Tempoveränderung angepasst werden, z. B. kann bei einem *Accelerando* bewirkt werden, dass es »erst später schneller wird«. Um Koinzidenzen zu bilden, kann man in ver-

21 Bézierkurven sind parametrisch modellierte Kurven, die in den frühen 1960er Jahren in der französischen Autoindustrie zur computerunterstützten Konstruktion aerodynamischer Formen unabhängig voneinander von Pierre Bézier bei Renault und Paul de Casteljau bei Citroën entwickelt wurden. Die heutigen Anwendungen sind vielfältig: Computergraphik, computerunterstütztes Design, Beschreibung von Schrifttypen usw. Eine kubische Bézierkurve ist für die vorgegebenen Punkte P_0 , P_1 , P_2 und P_3 im Einheitsintervall (n geht von 0 bis 1) folgendermaßen definiert:

$$P(n) = (1-n)^3 P_0 + 3(1-n)^2 n P_1 + 3(1-n) n^2 P_2 + n^3 P_3$$

schiedenen Temposchichten Kontrollpunkte setzen, die denselben Wert auf der Zeitachse haben. Werden die verschiedenen Temposchichten auf derselben *time map* abgebildet, erkennt man Koinzidenzen daran, dass die entsprechenden Kontrollpunkte vertikal übereinander ausgerichtet sind (Abb. 6.10).

Abbildung 6.9: Eine kubische Bézierkurve wird durch vier Punkte bestimmt.

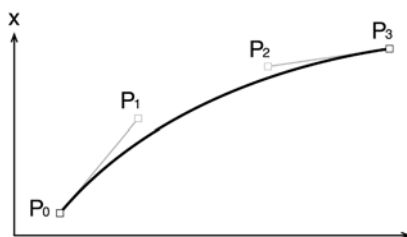


Abbildung 6.10: Koinzidenzen zwischen einem Accelerando (Kurve mit zunehmender Steigung) und einem Ritardando (Kurve mit abnehmender Steigung).

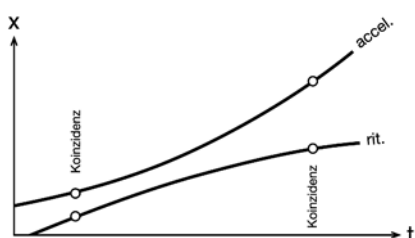
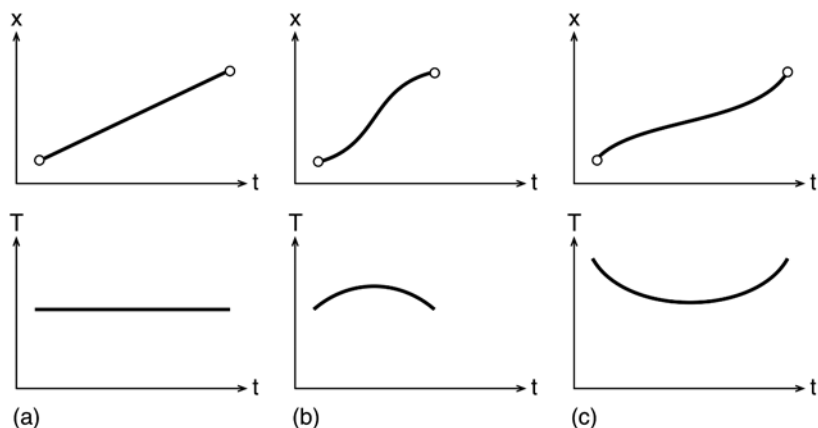


Abbildung 6.11: Anpassung von »unmöglichen« Parameterkonstellationen durch die Bézierkurve auf der *time map* (oben) und die dazugehörige Tempofunktion (unten): konstantes Tempo zwischen zwei Kontrollpunkten (a), dasselbe Tempo, aber zu wenig Zeit (b), gleich viel Zeit aber zu schnelles Tempo (c).



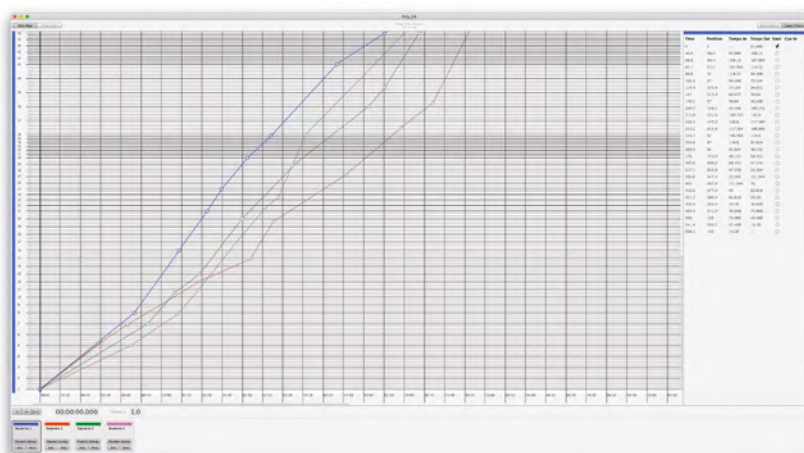
Bei aufeinanderfolgenden Kontrollpunkten auf einer *time map* müssen sich sowohl Zeit als auch Ort erhöhen, denn weder kann die Zeit umgedreht werden noch ist es vorgesehen, im Notentext rückwärts zu lesen. Die resultierende Kurve ist demnach monoton steigend. Durch die Visualisierung lässt sich auch die gegenseitige

Abhängigkeit von Zeit, Ort und Tempo veranschaulichen und begreifbar machen, was für den Kompositionsprozess und die Disposition von polyphonen Tempoverläufen von großem Wert sein kann. Mithilfe der Formel für die kubische Bézierkurve lässt sich beinahe immer eine Lösung finden, zwei Kontrollpunkte zu verbinden, selbst für gewisse »unmögliche« Kombinationen, bei denen z. B. das gewählte Tempo so langsam ist, dass die Zeit zwischen zwei Kontrollpunkten nicht ausreicht, um die Strecke im Notentext zurückzulegen. In solchen Fällen wird die Bézierkurve automatisch s-förmig, was dafür sorgt, dass das Tempo im Abschnitt zwischen den beiden Kontrollpunkten wie gewünscht beginnt, sich beschleunigt und zuletzt wieder verlangsamt, sodass der zweite Kontrollpunkt rechtzeitig und im gewünschten Tempo erreicht werden kann (Abb. 6.11).

Implementation

Die hier dargelegte Tempomathematik wurde in der Applikation *PolytempoComposer* implementiert. Dank ihrer graphischen Benutzerschnittstelle macht diese Applikation komplexe Tempostrukturen sichtbar und manipulierbar und ist damit ein Hilfsmittel zum Entwerfen von tempopolyphonen Kompositionen (Abb. 6.12). Kontrollpunkte werden graphisch auf einer *time map* oder numerisch in einer Tabelle dargestellt und können dort auch bearbeitet werden. Für jeden Kontrollpunkt kann die Zeit, der Ort, das momentane Tempo bei Erreichen des Kontroll-

Abbildung 6.12: Bildschirmfoto der Applikation PolytempoComposer. Die Kontrollpunkte werden auf einer time-map und rechts davon in einer Liste abgebildet. Die verschiedenen Tempoströme (hier Sequence genannt) sind farblich unterschieden und können unten zur Bearbeitung angewählt werden.



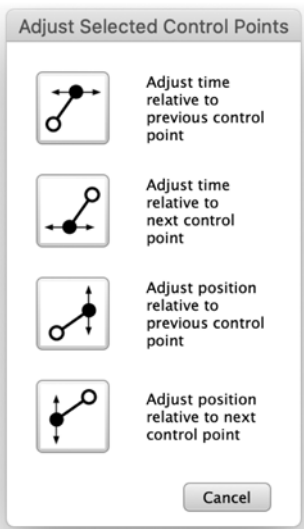
punkts (*Tempo In*) und das momentane Tempo beim Verlassen des Kontrollpunkts (*Tempo Out*) angegeben werden. Dank der Aufteilung des Tempos in zwei Komponenten ist es möglich, an einem Kontrollpunkt auch eine unvermittelte Tempoänderung stattfinden zu lassen. In diesem Fall müsste man *Tempo In* und *Tempo Out* entsprechend auf zwei verschiedene Werte setzen.

Da mit dieser Applikation in der Regel rhythmisierte Signale erzeugt werden, die Taktschlägen entsprechen, liegt es auf der Hand, die rhythmische Struktur als eine Abfolge von Taktarten anzugeben. Dazu werden die Taktarten in eine Tabelle eingetragen, wobei auch alle erdenklichen Formen von zusammengesetzten Taktarten und Taktarten mit *valeurs ajoutées* möglich sind (Abb. 6.13). Die rhythmischen Orte der Taktschläge werden als horizontale Linien auf der *time map* abgebildet und dort, wo diese Linien die Kurve schneiden, lassen sich nun die genauen Zeitpunkte dieser Taktschläge ablesen. Schließlich können die Zeitpunkte exportiert werden, entweder in dem Format, das von der Applikation *PolytempoNetwork* wiedergegeben werden kann, oder als neutrale Zahlenreihe, um sie in einem anderen Programm für Computermusik weiterzuverwenden. Falls eine Rhythmik gewünscht wird, die sich nicht in Taktarten ausdrücken lässt, ist es stattdessen auch möglich, einzelne Events manuell und mit einer absoluten Ortsangabe einzugeben.

Abbildung 6.13: Liste von Taktarten, die die metrische Struktur einer Komposition darstellen. Es sind auch zusammengesetzte Taktarten möglich.

Pattern	Repeats
5/4	5
3+2+2/8	4
4/4	5
1/4+1/4+1/16	6
3/4	5
3+3/8	1
6/4	5
2/4	4
7/4	4

Abbildung 6.14: Die Applikation bietet einige Hilfsfunktionen zur Manipulation von Kontrollpunkten auf einer time map.



Für einige grundlegende Manipulationen von Kontrollpunkten, die beim Konstruieren von Tempostrukturen immer wieder benötigt werden, existieren Hilfsfunktionen. Es können zwei oder mehrere aufeinanderfolgende Kontrollpunkte angewählt und deren Tempo so verändert werden, dass sich immer ein konstantes Tempo zwischen zwei benachbarten Kontrollpunkten ergibt. Es können Kontrollpunkte auch zeitlich oder örtlich so verschoben werden, dass das Tempo zum jeweils vorhergehenden oder nachfolgenden Kontrollpunkt konstant bleibt, bzw. sich linear zwischen den beiden gegebenen Tempi verändert (Abb. 6.14). Alle weiterführenden oder komplexeren Manipulationen von Kontrollpunkten wären so spezifisch auf einzelne Werke oder die Arbeitsweisen einzelner Personen bezogen, dass es nicht als zielführend erachtet wurde, sie zu implementieren. Hier müssen die Komponist:innen individuelle Lösungen finden.

Die Applikation *PolytempoComposer* wird seit 2016 als plattformübergreifende Stand-Alone-Software entwickelt.²² Sie ist ebenfalls in C++ mithilfe des Framework *JUCE* programmiert und quelloffen. Zuerst wurde sie nur dazu verwendet, das Aufführungsmaterial für die technikgestützte Tempovermittlung herzustellen. Dazu wurden die Zeitstrukturen, die die Komponist:innen mit anderen Mitteln konstruiert hatten, in die entsprechenden Tabellen der Applikation kopiert. Seit 2018 wird die Applikation auch immer mehr als Hilfsmittel für den Kompositionsprozess in Anspruch genommen. Weil sich gezeigt hat, dass etliche Komponist:innen zur Ausarbeitung der Details trotzdem gerne auf herkömmlichem Notenpapier arbeiten (mehr zur Operativität der Schrift in Kapitel 8), wurde eine Funktion hinzugefügt, mit der ein leeres Notenpapier erstellt werden kann, auf dem für alle Instrumente die Taktstriche und Zählzeiten zeitmaßstäblich abgebildet sind (Abb. 6.15).

22 Philippe Kocher: *Polytempo Composer: A Tool for the Computation of Synchronisable Tempo Progressions*, in: *Proceedings of the Sound and Music Computing Conference*, Hamburg 2016, S. 238–242.

Abbildung 6.15: Bildschirmfoto: Für eine konstruierte Tempostruktur kann leeres Notenpapier erstellt werden.

