

Georg Wötzer

Entstehung einer algorithmischen Komposition am Beispiel von "SYN" für Instrumentalensemble und Live-Elektronik(1993)

I) Voraussetzungen

Algorithmisches Komponieren, also die Entwicklung kompositorischer Gedanken mit Hilfe von Computersoftware, setzt mehrere Dinge voraus, die sich beim konventionellen Kompositionsprozeß nicht vorfinden.

1) Dazu gehört zunächst und unabdingbar eine passende hardwaremäßige Ausrüstung, also Rechner, Drucker, Synthesizer, Sampler, Effektgeräte, Mischpult, Patchbay, und genügend Kabel.

2) Weiterhin benötigt man, um mit einem Rechner kommunizieren zu können, Programme, und zwar einmal kommerzielle Programme aber auch - für mich ganz selbstverständlich - möglichst viel selber hergestellte Kompositionssoftware.

a) An kommerzieller Software verwende ich (Stand: SYN, also 1993) ein sehr umfangreiches und bewährtes Sequenzerprogramm, mit dem sich Notenaufzeichnungen, -Bearbeitung, -Druck und -Wiedergabe ziemlich professionell bewerkstelligen lassen. Für die verschiedenen Synthesizer bzw. Sampleplayer, mit denen ich arbeite, gibt es kommerzielle Editorprogramme, so daß die Einstellungen an den Geräten vom Computer aus vorgenommen werden können.

b) Das Zentrum meines algorithmischen Komponierens bildet indes ein System von vielen aufeinander bezogenen Programmen, die glücklicherweise nicht käuflich sind. Die meisten Programme sind in GFA-BASIC auf einem ATARI-Rechner entwickelt worden. Meine neueren Musikprogramme wurden allerdings in BORLAND-C auf einem PC geschrieben. Dazu gehört ein sich in Arbeit befindliches Generalbaß-Programm sowie ein Prototyp für ein musikalisches Gestaltssystem. Die Verfügbarkeit über ein solches algorithmisches Kompositionssystem ist m.E. indes die Voraussetzung für rechnergestütztes Komponieren.

c) Die Komponenten eines solchen Systems sind in Kürze folgende:

1) **'Klangfarbenlexikon'**, d.h. ein Programm, das für eine jeweilige Komposition spezifische Klangfarbenskalen bereithält. Diese sind nach klanglichen Eigenschaften von Hand sortiert und enthalten

Informationen über die einzelnen verwendeten Spieltechniken bzw. Klangfarben der im Stück vorhandenen Instrumente.

2) '**Generierende**' Programme:

α) Hierbei handelt es sich um eine Gruppe von Programmen, die Gestalten bzw. Teile von Gestalten nach vorgegebenen Kategorien neu generieren können.

β) Ein Spezialfall stellen Programme dar, die mittels Angleichung quasi als entwickelnde Variation transitive Gestalten, also Übergänge produzieren; die Angleichung kann sich sowohl auf die Gestaltdaten selbst als auch nur auf die Methoden beziehen, mit denen eine Gestalt generiert wurde. Die Methoden selbst sind dabei als Methodenskala organisiert.

χ) Im letzten Jahr (1993) hatte ich ein größeres (prototypisches) Programm in Arbeit, das eine bestehende, von Hand komponierte Gestalt gemäß einem Satz von Gestaltkategorien klassifiziert und daraus weitere Gestalten ableitet, wie z.B. ein genau definiertes Gegenteil zur vorhandenen Gestalt, oder eine genau definierte verwandte Gestalt. Die Aufreihung der so gewonnenen neuen Gestalten bilden - mit der Grundgestalt als Zentrum - ein Gestaltssystem, aus dem sich dann durch Modifikationen oder Übergänge neue Gestalten entwickeln lassen als Grundlage eines gestaltkategorienmäßig genau definierten Formablaufs.

3) '**Modifizierende**' Programme:

α) Ungeheuer wichtig für effizientes kompositorisches Arbeiten mit dem Rechner sind ausgefeilte Korrekturprogramme, mit denen sich mühelos und schnell alle möglichen Änderungen an Gestalten, Klangfarbenskalen, Rhythmen, Harmonien etc. vornehmen lassen.

β) Daneben existieren Programme für globale Gestalttransformationen, z. B. für Transposition, Zeit- und Lautstärkeumwandlungen sowie Transformationen von Klangfarbenskalen.

4) Programme zur Kommunikation über die **MIDI - Schnittstelle**:

Um das erzeugte Resultat auf dem Synthesizer anhören oder im Notenbild mittels eines Sequenzerprogrammes betrachten bzw. ausdrucken zu können, müssen MIDI-Routinen zur Verfügung stehen. Diese ermöglichen aber nicht nur die problemlose Übertragung von Gestaltdaten. Geradezu lebensnotwendig für effizientes kompositorisches Arbeiten mit dem Rechner sind hier Filtermechanismen zur Unterdrückung unerwünschter Informationen. So muß es z.B. möglich sein, aus einer Gestalt mit großer Orchesterbesetzung, zu der vielleicht auch vier Schlagzeuger gehören, nur die Partie des 3. Schlagzeugs auszugeben, und zwar so, daß alle Fellinstrumente unbestimmter Tonhöhe auf einem eigenen Notensystem notiert werden, ebenso die Holz - und Metallinstrumente usw.

- 5) Schließlich sollte der besseren Übersicht wegen eine **Shell** vorhanden sein, aus der heraus die benötigten Programme bequem gestartet werden können (Bsp1).¹

II) Algorithmische Darstellung eines musikalischen Problems

1) Computerprogramme sind in bestimmten Sprachen geschrieben, welche die mehr oder bequeme Kommunikation mit einem Rechner ermöglichen. Da es viele ziemlich unterschiedliche Computersprachen gibt, die meist mit erheblichem Aufwand erlernt werden müssen, ist die Wahl der passenden Sprache besonders für den nicht professionellen Programmierer eine wichtige Vorentscheidung für den zukünftigen Erfolg seiner Arbeit. Nach meiner Erfahrung sollten hierbei drei Aspekte berücksichtigt werden: Schwierigkeit, Mächtigkeit und Portabilität einer Sprache.

Ich selber habe bisher mit der sehr einfachen, relativ mächtigen aber nicht übermäßig portablen Sprache GFA-BASIC programmiert. Seit geraumer Zeit entwickle ich meine Musikprogramme in C mit Blickrichtung auf C++ bzw. Smalltalk. Der Hauptgrund für den Umstieg heißt Portabilität. Programme, die in C, C++ oder Smalltalk geschrieben sind und auf Maschinenspezifika verzichten, laufen fast problemlos auf vielen Rechnertypen. Allerdings ist bereits C wesentlich schwerer zu erlernen als BASIC, von C++ nicht zu reden.

2) Zur Veranschaulichung will ich eine leicht gekürzte Sequenz aus einem Programm herausgreifen (Sprache: BORLAND-C). Und zwar geht es um den bereits erwähnten Prototyp für ein musikalisches Gestaltssystem.

Dieses besteht aus einer Grundgestalt, die nach vielen parametrischen Klassifikationen organisiert ist, und aus dieser abgeleiteten Gestalten, die entweder reduzierte oder extendierte Varianten sind der Grundgestalt. Ein wichtiges Teilproblem dabei ist die Programmierung des Reduktionsmechanismus der überlagerten Gestaltschichten (=Ebenen) und der Reduktion der Bestandteile der Schichten, welche ich Module nenne und am einfachsten mit Motiven verglichen werden können. Jede Gestalt besteht mindestens aus einer Schicht, und jede Schicht besteht mindestens aus einem Modul, häufig aber aus viel mehr. (Bsp2).

Dieser Reduktionsmechanismus wird in der Funktion **rahmen_reduktion** (**Parameterliste**) folgendermaßen definiert:

- 1) Setzen der Schichtanzahl der Grundgestalt.
- 2) Ermittlung der Schichtanzahl für jede Reduktionsvariante;
- 3) a) Setzen der Ordnungsnummern der Schichten der Grundgestalt;
b) Setzen der Modulzahl pro Schicht.

¹Die Beispiele folgen am Ende des Textes.

4) Setzen der Ordnungsnummer der Schichten der Reduktionsvarianten über Zufall. Sortierung derselben. Um größtmöglichen Zusammenhang zwischen den Reduktionsvarianten zu gewährleisten, soll der Zufall so gelenkt werden, daß nur auf in vorhergehenden Gestalten vorhandene Schichten zurückgegriffen wird. Außerdem muß sichergestellt werden, daß sich keine Schichtnummer in derselben Reduktionsgestalt wiederholt.²

```
void rahmen_reduktion(int anz_sch,int ***sch_ges, /*Parameterübergabe*/
                    int ***m_p_sch,int **sch_zahl,int *ges_mpsch,int ges_ges){
    /*Variablendeklarationen*/
    int g,h,j,s,t,schicht_nr,alt_modzahl,neu_modzahl,ok,zuf_wert,found;
    int **sges_ptr1,*sges_ptr2,**mps_ptr1,*mps_ptr2,*schz_ptr;

    *sch_zahl=(int *)malloc(ANZ_REDGESTALTEN*sizeof(int)); /* Beschaffung
                                                            von Speicher*/
    schz_ptr=*sch_zahl; /* Setzen der Schichtzahl der Grundgestalt*/
    for(g=0,h=ANZ_REDGESTALTEN;g<ANZ_REDGESTALTEN;g++,h--){
        schz_ptr[g]=anz_sch*((float)h/ANZ_REDGESTALTEN); /* Schichtanzahl für
                                                            jede Reduktionsvariante(Reduktion a)*/
        if(!schz_ptr[g])
            schz_ptr[g]=1;
    }

    /* Kontrollen über genügend vorhandenen Speicher wurden weggelassen*/
    *sch_ges=(int**)malloc(ANZ_REDGESTALTEN*sizeof(int *)); /* Beschaffung
                                                            von Speicher*/

    sges_ptr1=*sch_ges;
    *m_p_sch=(int**)malloc(ANZ_REDGESTALTEN*sizeof(int *));
    mps_ptr1=*m_p_sch;

    for(g=0;g<ANZ_REDGESTALTEN;g++){
        /* Beschaffung von
           Speicher*/
        sges_ptr2=(int *)malloc(schz_ptr[g]*sizeof(int));
        sges_ptr1[g]=sges_ptr2;
        mps_ptr2=(int *)malloc(schz_ptr[g]*sizeof(int));
        mps_ptr1[g]=mps_ptr2;
    }

    for(g=s=0;s<anz_sch;s++){
        sges_ptr1[g][s]=ges_ges[s]; /* Die Schichten der Grundgestalt werden
                                     gesetzt*/
        mps_ptr1[g][s]=ges_mpsch[s]; /* Die Module der Schichten der Grundgestalt
                                     werden gesetzt*/
    }
}
```

²Programm-Kommentare stehen zwischen /* ein Kommentar */ und sind fett gedruckt.

/* Schichtauswahl über Zufall;jede folgende Reduktionsgestalt greift auf vorhandene ausgewählte Schichten zurück;diese sind außerdem sortiert (Reduktion b)*/

```
for(g=1;g<ANZ_REDGESTALTEN;g++){ /*für jede Gestalt(Grd.+Red-Gest) etwas tun */
    sges_ptr2=sges_ptr1[g]; /* zuvor allozierter Speicher wird an sges_ptr2 zugewiesen*/
    ok=0; /* für jede Gestalt wird ok auf 0(==FALSE) gesetzt*/
    while(!ok){ /* solange wie ok ungleich 0 ist*/
        for(s=0;s<schz_ptr[g];s++){ /*für jede Schicht pro Gestalt */
            found=0; /*found wird für jede Schicht auf 0 gesetzt */
            while(!found){ /*solange wie found 0 ist */
                zuf_wert=random(anz_sch); /*Setzen einer Zufallszahl von 0-Zahl der Schichten der Grdgest.-1 */
                for(t=0;t<schz_ptr[g-1];t++){ /*für jede Schicht der vorangegangenen Gestalt */
                    if(zuf_wert==sges_ptr1[g-1][t]){ /*wenn der Zufallswert gleich gewählter SchichtNr der vorausgegangenen Gestalt ist */
                        sges_ptr2[s]=zuf_wert; /*wird der aktuelle Wert zugewiesen */
                        found=1; /*und die Variable found auf TRUE gesetzt */
                        break; /*sowie die innerste Schleife abgebrochen */
                    }
                } /*Da found nun TRUE ist,wird die innerste while-Schleife verlassen */
            } /*sodaß dieser Vorgang für alle Schichten der aktuellen Gestalt wiederholt werden kann*/
            ok=auf_verschieden_pruefen(sges_ptr2,schz_ptr[g]); /*Dann erfolgt Prüfung,ob die verschiedenen Schichtnummern pro Gestalt verschieden sind,und zwar so lange, bis alle Schichtnummern der aktuellen Gestalt verschieden sind.*/
            if(ok){ /*Wenn verschieden, wird von unten nach oben sortiert und die gesetzten Schichtnummern zugewiesen */
                sortiere(sges_ptr2,schz_ptr[g]);
                sges_ptr1[g]=sges_ptr2;
            }
        }
    }
}
for(g=0,j=ANZ_REDGESTALTEN;g<ANZ_REDGESTALTEN;g++,j--){
    for(s=0;s<schz_ptr[g];s++){
        schicht_nr=sges_ptr1[g][s];
        alt_modzahl=mps_ptr1[0][schicht_nr];
        neu_modzahl=alt_modzahl*((float)j/ANZ_REDGESTALTEN); /*lineare Umrechnung der (gegenüber der Grundgestalt reduzierten) Anzahl der Module pro Schicht*/
        if(neu_modzahl==0)
            neu_modzahl=1; /* Jede Schicht hat mindestens 1 Modul, auch wenn 0 errechnet wurde*/
        mps_ptr1[g][s]=neu_modzahl;
    }
}
}
```

```
int auf_verschieden_pruefen(int *fld,int s_z){/*hier werden alle Zahlen,d.h.
    gewählten Schichtnummern miteinander verglichen*/
    int o_k,i,j;
    if(s_z>1) {/* wenn Schichtzahl 1 ist, wird Vergleich umgangen u. TRUE
        zurückgegeben, da dann keine Schichtnummernverdoppelung auftreten kann */
        o_k=1;
        for(i=0;i < (s_z-1);i++){/*Test auf Gleichheit bzw. Wiederholung einer
            SchichtNr*/

            for(j=(i+1);j<s_z;j++){
                if(fld[j]==fld[i]){
                    o_k=0; /* wenn Gleichheit entdeckt, Rückgabe von FALSE*/
                    return o_k;
                }
            }
        }
    }
    return o_k; /*sonst wird TRUE zurückgegeben */
}

void sortiere(int *fld,int groe){
    qsort((void *)fld,groe,sizeof(fld[0]),int_sort); /*Bibliotheksfunktion */
}

int int_sort(const void* a1,const void* a2){ /*vergleicht zwei Zahlen miteinander */
    int *z1,*z2;
    z1=(int *)a1;
    z2=(int *)a2;
    if(*z1==*z2) /*und liefert Vergleichsresultat zurück */
        return 0;
    else if(*z1<*z2)
        return -1;
    return 1;
}
```

III) Die Entstehungsweise von SYN

1) SYN ist ein Auftragswerk der Stadt Stuttgart und war ursprünglich als Ballettmusik für die 'Stuttgarter Hofkonzerte' geplant. Wegen organisatorischer Probleme mußte das Ballett bisher ausgespart werden. Die Besetzung des Stückes ist folgende:

Solistisch besetztes Streichquintett (elektronisch verstärkt) - 2 Klarinetten, 1 Baßklarinette, 1 Sopransaxophon - 1 Trompete, 2 Hörner, 2 Posaunen - 3 Schlagzeuger - Live-Elektronik. Letztere verwendet zwei Sample-Player (Proteus 1 und Proteus 2), sowie einen Synthesizer mit Obertonsynthese (Kawai K5).

Die Funktion der Sample-Player besteht in der Verdoppelung des gesamten Live - Instrumentariums, sodaß sehr oft die gespielten von den elek-

tronischen Klängen kaum unterscheidbar sind. Der Synthesizer wiederum hat explizit die Funktion des Lieferanten für 'künstliche' Klänge.

2)a) Da sich die 'Hofkonzerte' hauptsächlich an ein Laienpublikum wenden, war eine der Vorgaben an die Musik, daß sie tonale Elemente enthalten sollte, um einmal das musiksprachlich Ungewohnte zu entschärfen und zum andern eine kompositorische Handlung hörbar zu machen, die sich auf mehreren sprachlichen Ebenen bewegt, und zwar auf den Ebenen Tonalität, angereicherte Tonalität und Nicht-Tonalität.

b) Das kompositorische Vorgehen war so, daß 'von Hand' 23 Gestalten kreiert wurden, die in ihrer Abfolge eine Art Handlung repräsentieren. Die tonalen Gestalten bilden aufgrund ihrer leichteren Identifizierbarkeit sowie ihrer starken 'auratischen' Vorbesetztheit das Formgerüst für das ganze Stück.

So wird ein Bogen geschlagen von Brahms-Ähnlichem als 'vergangeschöner, intakter Musik', von einem Mahler-Zitat aus der 3. Sinfonie (dunkelste Seelenlage) sowie einem Chopin-Walzer in cis-moll (romantische Trauer) über Meyerbeers Schwerterweihe vor dem Untergang aus den 'Hugenotten' und einer nicht-tonalen Gestalt, deren Gestus und Material der Hinrichtungsszene aus Strauß' 'Eulenspiegel' entlehnt ist bis hin zu Isaaks Fassung des Innsbruck-Lieds ('da ich im Elend bin'), bewußt pathetisiert als Ausdruck ernstgemeinter Emphase. Der Schluß des Stücks dagegen setzt Tonalität als leere, schönklingende Fassade ein, Versöhnung im Schmerz vorgaukelnd, wo nur der Schmerz bleibt.

Dem entgegen steht eine Handlungsebene, deren Inhalt eine Klangfarbe ist, ein synthetischer 'heller Klang' im Gestus des Leidenden, symbolisierend 'beschädigte Natur, durch ein Kunstmittel zum Ausdruck gebracht' (und erzeugt mittels digitaler Obertonsynthese). Beide Handlungsebenen beeinflussen sich. Im ersten Drittel bewirkt das Auftreten des 'hellen Klangs' Resignation in der Gegenschicht, die dann umschlägt in eine wilde (Flucht-) Bewegung, aus der dann jedoch ein bedrohlicher Impuls entsteht. Auf dem Höhepunkt, der Hinrichtungsszene, ist aus dem geheimnisvollen, sanft leidenden 'hellen Klang' eine Folge geräuschhafter, verzerrter, nachklingender Clusterschläge geworden, die angstvoll auf die Akkordschläge der Gegenschicht (Hinrichtungsszene) reagieren. Die Clusterschläge verwandeln sich dann wieder zurück in den ursprünglichen Zustand als 'heller Klang', der verschwindet und die Gegenschicht alleine ihr Lied zu Ende singen läßt.

IV) Analysen

α) Analyse einer von Hand komponierten Partie

Die Kompositionstechnik von SYN ist hybrid: einerseits von Hand komponiert (1/5 des Stücks), andererseits durch den Rechner 'erfunden' (4/5 des Stücks). Zur Veranschaulichung folgt die kurze Analyse einer von Hand komponierten Partie; es handelt sich um das drittletzte Pattern (Bsp.3).

Dieses weist vier Schichten auf:

- 1) Streicher + Klarinetten 1,2 (Tonfeld in höherer Lage);
- 2) Blechbläser + B-Klarinette (tiefe Akkordcluster);
- 3) Schlagzeug (Impulse);
- 4) Synthesizer (gehaltene, verzerrte, hohe Cluster).

a) Analyse von Schicht 1 (Streicher+Klarinetten 1,2)

- 1) Rhythmische Entwicklung als kontinuierliche Verdichtung;
- 2) Tonhöhenfelder aus Skalen gewonnen, die ihrerseits Komplemente sind zu den tiefen Bläserakkorden der Schicht 2; Zusammenklänge auf 'schwerer' Zeit meiden Oktaven;
- 3) permanente Zunahme der Lautstärke: mp - mf - poco f - f - molto f.

b) Analyse von Schicht 2 (Blechbläser, B-Klarinette)

- 1) Diese Schicht bildet den Kern der ganzen Gestalt. Der 5-stimmige Cluster ist symmetrisch um den Ton As (Intervalle 2-1-1-2) und steuert die Schichten 1 und 4. Die Akkorde sind so angeordnet, daß der letzte, 7. Akkord die Transposition des 1. Akkords ist um eine kl. Obersekunde. Jedoch darf sich jede Stimme nur um eine kl./gr. Sekunde und nur so weiterbewegen, daß keine Tonverdoppelung im resultierenden Cluster auftritt.
- 2) Die Zeitabstände sind konstant, die Lautstärke nimmt kontinuierlich zu.

c) Analyse von Schicht 4 (Synthesizer)

- 1) Für die Tonhöhenorganisation dient der Blechbläser-Cluster als Modell. Aus dessen Tonvorrat werden so die Obertöne abgeleitet, daß der resultierende Synthesizer-Akkord in sich aus lauter unterschiedlichen Tönen besteht, die aber identisch sind mit den Clustertönen, nur oktavversetzt.
- 2) Die rhythmische Organisation ist so beschaffen, daß zunächst die beiden ersten Akkorde zueinander im Abstand stehen von 12 Sechzehntel - Sextolen. Dann verschiebt sich der erste Akkord 'nach links' um 1 Sextolen - Sechzehntel, der 2. Akkord ebenso nach rechts, sodaß beide Akkorde sich gleichzeitig voneinander entfernen und annähern.

d) Analyse von Schicht 3 (Schlagzeug)

Hier bewegt sich das strukturelle Geschehen auf den zwei Ebenen Fell - und Metallinstrumente.

α) Die Fellinstrumente spielen einen von Hand komponierten Übergang von einer 1. zur 7. Gruppe.

Die 1. (Ausgangs-) Gruppe weist wiederum zwei Ebenen auf: Ebene 1 mit 3 Einzelimpulsen, in größerem Abstand; Ebene 2 mit 3 Impulsgruppen abnehmender Größe in kleinem Abstand.

Die 7. (Ziel-) Gruppe splittet sich in vier Ebenen auf:

Ebene 1 mit Einzelimpuls am Anfang (Conga 2);

Ebene 2 mit 3-er-Impulsgruppe, starkes accelerando (Conga 1);

Ebene 3 mit 3-er-Impulsgruppe, schwaches rallentando (Bongo 2);

Ebene 4 mit 3-er-Impulsgruppe, schwaches rallentando, augmentiert (Bongo 1).

β) Auch die Metallinstrumente bewegen sich auf zwei Ebenen:

1) sich allmählich beschleunigender Impuls im tiefen Becken, dann in den Ambossen;

2) kurzes Wegfiltern einer in den Vierteln 1-4 exponierten, 'frei' komponierten Teilgestalt.

β) Analyse einer mittels gestaltgenerierendem Programm komponierten Partie

Da mein algorithmisches System auch über gestaltgenerierende Software verfügt, kann man je nach Bedarf den Rechner auch zum Erfinden von Grundgestalten oder Teilen davon heranziehen. Bei der 7. Gestalt wurde alles von Hand komponiert außer der Synthesizer-Partie (Bsp4). Die Partiturvorgaben an das Programm waren folgende:

1) errechne einen Abschnitt tonhöhenmäßig zufällig aber überwiegend steigend von Dis nach g³, und zeitlichem Positionsabstand der einzelnen Ereignisse von 1/32;

2) dann dasselbe tonhöhenmäßig zufällig fallend, von g³ nach c²;

3) dann weiter zufällig fallend, von a¹ nach e; Abstandsdichte leicht reduziert, etwas langsamer als 1/32;

4) Dann schließlich ebenfalls zufällig von e nach Dis; Abstandsdichte noch etwas langsamer (1 Sechzehntel-Septole).

Somit ist die elektronische Schicht eine Art Gegenschicht zu den Live-Instrumenten: während jene überwiegend insgesamt langsam steigen, ist die elektronische Schicht überwiegend fallend. Im Zeitverhalten verstärkt sie aber eher die Gestaltintention eines dichten Arpeggiefelds.

Exkurs: Übergänge, vom Rechner komponiert

Die Formkonzeption von SYN beruht auf der Vorstellung von Form als Ansammlung von Gestalten, welche durch vermittelnde Teile miteinander verbunden werden können. Sowohl können die Gestalten als Ensemble untereinander verwandt sein, als auch ihre Sukzession eine übergeordnete 'Formgestalt' und Symbolik ergeben. Andererseits müssen gerade in der direkten Aufeinanderfolge von solchen Gestalten Kontraste oder zumindest deutlich wahrnehmbare Unterschiede auftreten, um möglicher Langeweile durch Häufung des immer Ähnlichen auszuweichen. Auf jeden Fall spielen die Begriffe Vermittlung, Übergang, Verknüpfung eine zentrale Rolle in meinem algorithmischen Kompositionssystem. Erfahrungsgemäß sind es nicht alleine die fehlenden Kontraste, die manche Stücke langatmig geraten lassen, es ist vielmehr die Geringschätzung der kompositionstechnisch erstrangigen Kategorie der direkten Verknüpfung von musikalischen Gedanken.

a) Daher wird die Kategorie Übergang algorithmisch als eigener Parameter behandelt, fast beliebig skalierbar: von keinem Übergang = Kontrast bis zu maximal vielen Übergängen = kaum wahrnehmbare Annäherung an die Zielgestalt.

b) Das Übergangsprogramm, mit dessen Hilfe fast 4/5 von SYN komponiert wurde, kennt mehrere prinzipielle Optionen:

1)α) Übergang als Substitution;

β) Übergang als Annäherung;

2)α) Verarbeitung von ganzen Komplexen (von Stimmen);

β) transitorische Verarbeitung nur einer einzelnen Stimme.

SYN ist ausschließlich bearbeitet worden mittels komplexbezogener Übergänge. Der große Vorteil dabei, vor allem bei größerer Besetzung, ist die enorme Verarbeitungsgeschwindigkeit. Nachteilig ist, daß die Annäherungen häufig 'sprunghaft' und - von der Wahrnehmung her - willkürlich verlaufen (und zwar programmtechnisch infolge indexbezogenem, quantitativem Vergleichsmechanismus).³ Andererseits können solche kontextuellen 'Sprünge' wiederum Abwechslung erzeugen und der Mechanik der variativen Annäherungen entgegenwirken.

Als Beispiel für einen stimmenbezogenen Übergang verweise ich auf die kurze Analyse einer stimmenbezogenen Übergangspartie aus 'ECHOS' (1990).⁴

³siehe meine diesbezüglichen Ausführungen zum Übergangsprogramm in verschiedenen Nummern der STUDIO-BLÄTTER

⁴STUDIO-BLÄTTER Nr 7, S.83 ff.

χ) Analyse eines Übergangs als Substitution

Beim Übergang als Substitution werden vom Rechner sukzessive linear oder per Zufall die Elemente des Ausgangspatterns mit denen des Zielpatterns vertauscht. In SYN wurde diese Vermittlungsart hauptsächlich bei Ein- und Ausblendungen von Gestalten verwendet.

Die Einblendung geschieht so, daß eines oder sehr wenige Elemente einer Zielgestalt als Ausgangsgestalt gesetzt werden und immer mehr Elemente in die folgenden, daher immer extensiveren Wiederholungen eingebaut werden, bis die Zielgestalt erreicht ist.

Das Ausblenden geschieht in umgekehrter Reihenfolge. Selbstredend lassen sich Überlagerungen beider Prozesse gleichzeitig realisieren und/oder andere Überlagerungen. Ein derartiger Übergang ereignet sich z.B. zwischen S.110 und S. 126 der Partitur.⁵

δ) Analyse eines Übergangs als Annäherung

Übergänge als Annäherung sind zielgerichtete Variationen, bei denen die Ausgangselemente in allen Parametern allmählich an einen Zielzustand angeglichen werden. In SYN fand diese Art von Vermittlung überwiegend Verwendung. Da Zahl und Ausdehnung der Übergänge weitgehend beliebig sein können und (fast) beliebig viele Übergänge gleichzeitig stattfinden können, entstehen zwangsläufig durch Überlagerungen ungleich langer Übergangsperioden mehr oder weniger offensichtliche formale Verschränkungen, die Stillstand durch permanent 'kadenzierende' Syntax verhindern hilft.

Ein Beispiel hierfür ist der Übergang vom vorletzten zum letzten Pattern. Die vorletzte Gestalt mündet in einer tonalen Vorhaltswendung, das Zielpattern bringt hauptsächlich über gehaltenen Blechbläserakkorden in den Streichern rasche gebrochene Tremoli. Hörbar wird beim Übergang die vielschichtige Umwandlung der Vorhaltswendung in den Streichern und Klarinetten, und das allmähliche Hervortreten von Haltetönen besonders in den Hörnern. Kurz vor Erreichen des Ziels fällt die Belebung (als Antizipation der raschen gebrochenen Tremoli des Ziels) in den Streichern auf.⁶

V) Gesamtform

a) Zeitstruktur

Die algorithmische Verfügbarkeit bzw. Generierung aller Teile des Stücks, also der 23 Kerngestalten sowie deren Vermittlungen, ermöglicht ein

⁵Aus Raumgründen sei auf die Abbildung des Notentextes verzichtet. Die Partitur kann aber beim Komponisten eingesehen werden.

⁶Partitur S.227-239.

kompositionstechnisches Faszinosum: die totale Strukturierbarkeit der realen Zeit 'in Sekunden' (Bsp5).

1) So wurden die realen Ausdehnungen der Übergänge an den Rändern des Stückes gedehnt, quasi als weiches Ein - und Ausschwingen der Form, während in der Mitte eher kürzere Vermittlungen bzw. Kontraste gewählt wurden, zur Erhöhung der Ereignisdichte, besonders gegen Ende des 2. Drittels.

2)a) Die reale zeitliche Ausdehnung der Übergänge läßt sich auf einer Skala von Sekundenwerten abbilden, die zueinander in einfachen Relationen stehen. Die Werte der Skala sind 0 - 15 - 20 - 30 - 60 - 90 - 120 - 180. Dabei dominiert die Relation 1:2, gefolgt von 1:3, 1:6 usw. bis zu 1:12 oder 4:5.

b) Die von Hand komponierten Gestalten sind in ihrer Zeitausdehnung dagegen frei organisiert und bilden somit eine Art Zeitkontrapunkt zur Schicht der Übergänge.

b) Tonalität und Form

Der Einbau von tonalen Zitaten bzw. zitähnlichen Gestalten ermöglichte eine zusätzliche Ebene formaler Organisation, zusätzlich zur psychologisch-symbolischen und zeitmäßig-transitiven Disposition der Gesamtform. An den Rändern sowie in der Mitte der Form befinden sich tonale Gestalten, die einerseits das Stück 'architektonisch' equilibrieren, andererseits die Ausbrüche des Nichttonalen im Zaume halten - und gerade dadurch wieder, und zwar besonders am Schluß mit seinen tonalen Leerformeln - die Hohlheit, ja Brüchigkeit solcher Konstruktion entlarven (Bsp 6).

c) Live versus elektronisch

Die Verwendung von Live-Elektronik und gespielter Musik zusammen mit Tonalität innerhalb eines Stückes 'Neuer Musik' schafft interessante Konstellationen, die sich wiederum als kompositorische Kategorie planvoll in einem Stück verwenden lassen.

1) So entstanden in SYN elektronische Strukturen aus Übergangsberechnungen, die zum gespielten Klang hinzugemischt wurden.

2) Ebenso entstanden aus Übergangsberechnungen zu spielende Strukturen, die die Spieler überfordert hätten. Solche Partien wurden den Sample-Playern zugeschlagen und somit der Live-Elektronik eine gewissermaßen subsidiäre Funktion zugewiesen. Dadurch konnte übrigens der errechnete strukturelle Gedanke nahezu unverändert 'herübergerettet' werden.

3) Außerdem enthielten bereits viele Rahmengestalten elektronische Bestandteile, einmal als Ausdifferenzierung einer instrumentalen Schicht; des weiteren als Gegenschicht zu einem Instrumentalpart, und zwar dies meist als *tonale* Schicht. Der gesampelte Klang als gespeicherter, abgestorbener wurde somit in bewußten Gegensatz gebracht zum lebendigen Musizieren. Und lebendiges Musizieren verwirklicht sich im zu Ende gehenden 20.

Jahrhundert nicht hauptsächlich im Spielen von älterer 'klassischer' Musik und ihrer uns in Wahrheit völlig fremden Klang- und Ideenwelt, sondern insbesondere durch die Einbeziehung des Geräuschhaften und der Besonderheiten von Klangregistern und Spieltechniken der einzelnen Instrumente. Solche geräuschhaften Klänge und subtilen Spieltechniken sind übrigens im Klangrepertoire gängiger Sampleplayer gar nicht oder nur sehr sparsam vorhanden.

Die Überlagerung von live und elektronisch tritt in SYN auf zwei Arten auf:

a) einmal als strikt voneinander getrennte Überlagerung von tonal und nichttonal;

sowie b) live und elektronisch tonal, dabei aber live als Begleitung.

4) Des weiteren gibt es im Stück auch Partien ganz ohne Elektronik; nicht vorhanden indes sind rein elektronische Teile ohne menschlichen Spieler.

VI) Die Einrichtung der Live-Elektronik

a) Einrichtung der PROTEUS-Klänge

Die elektronischen Klänge in SYN bestehen sowohl aus gesampelten Instrumentalklängen (PROTEUS 1,2), die das vorhandene Live-Instrumentarium verdoppeln, als auch aus synthetischen Klängen, die mittels Obertonsynthese selber erzeugt wurden (KAWAI K5).

Die benötigten Instrumentalklänge der beiden Sample-Player wurden zunächst für den In- und Output kanalmäßig festgelegt (Bsp7). Bei Schlagzeugklängen ergab sich allerdings eine Notations-Diskrepanz zwischen live und gesampeltem Schlagzeug. Diese mußte durch nachträgliche Umwandlung am Sequenzer beseitigt werden. Gerade in solchen Situationen haben sich meine Filterprogramme als lebensnotwendig erwiesen, da sie die Herausfilterung jedes Instruments bzw. jeder Klangfarbe sowie deren Festlegung auf bestimmte MIDI-Kanäle erlauben.

b) Einrichtung der KAWAI-Klänge

Das Klangmaterial des KAWAI-K5 dagegen besteht aus 11 verschiedenen Klangfarben, die als 'zusammenhängende' Klangfarbenskala synthetisiert wurde, und zwar von ganz hart und kurz allmählich zu ganz weich einschwingend und längerem Ausschwingen. Jede Klangfarbe muß in ihrem Spektrum (Obertöne, deren Lautstärke, Filter, Ein- und Ausschwingverhalten, LFO-Modifikationen usw.) genau festgelegt werden (Bsp 8).

Danach erfolgt ebenfalls - wie bei den Sample-Playern - die Festlegung der In-Out-Kanalbelegung.

VII) Erstellung des Notentextes

Nach der Eingabe der von Hand komponierten Gestalten mittels ausgefeilter Programme aus meinem Musiksystem bzw. der algorithmischen Generierung von Gestaltteilen sowie der Berechnung der Übergänge wurde die fertige Gestalt zum Sequenzer (NOTATOR) übertragen. Hier erfolgte dann, mit den genauen Daten zu jeder Note auf dem Bildschirm des sendenden Rechners und daher im Blickfeld, die stimmenweise Editierung der gesendeten Gestalt auf dem zweiten Rechner bezüglich Spieltechnik, Lautstärke, Artikulation sowie eventuellen Zeitkorrekturen. Danach wurden alle Stimmen einer Gestalt oder eines Übergangs zusammengemischt, sowie die Druckerfonts für Partitur - oder Einzelstimmendruck eingerichtet. Die Partitur wurde schließlich zusammengesetzt als Aufeinanderfolge von 23 Songs, aber je für sich, mit durchlaufender Seiten- und Taktnumerierung und partitur - sowie stimmenmäßig ausgedruckt.

Schließlich wurde aus der Gesamtpartitur die elektronische Schicht durch Eliminieren der anderen Schichten extrahiert. Die 'falsch' notierten Schlaginstrumente (s.o.) wurden transformiert und so die korrekten PROTEUS-Schlagzeugklänge ausgewählt. Auch die elektronische Partie wurde dann als eigene Stimme ausgedruckt.

VIII) Die Ausarbeitung der elektronischen Schicht

Zwar lassen sich im Arrange-Modus des Notator im Songformat abgespeicherte Gestalten als Patterns sukzessive in einer Patternliste anordnen und dann als ganzes Stück abspielen. Jedoch gehen dabei die bisherigen Taktinformationen verloren, sodaß in diese Arrangeliste nach Bedarf ein eigenes Takt - und Tempopattern eingefügt werden muß.

a) Das Tempopattern hat die Funktion, die verschiedenen Abspieltempi im Stück vorzuprogrammieren. Dieses vorprogrammierte Tempo ist aber mittels Tastendruck durch einen Spieler beeinflussbar, d.h., daß im zeitlichen Bereich die Elektronik dem Musiker folgt!

b) Die tatsächliche Taktart für jeden Takt wurde mittels eines Taktpatterns in Noten umgesetzt: eine tiefe Note zeigt die '1' an, eine hohe alle andern Zählzeiten.

c) Während der Aufführung wurde dieses Taktpattern über den Export-Ausgang des NOTATOR einem zweiten ATARI-Rechner via MIDI-IN zugeführt und dort über ein Programm so verarbeitet, daß die '1' an einem Bildschirm seitlich vor dem Dirigenten groß dargestellt wurde, die andern Schläge durch einen großen weißen Kreis sichtbar gemacht wurden. Dadurch hatte der Dirigent jederzeit die optische Kontrolle über Tempo und Zeitposition sowie die Möglichkeit der Beeinflussung des Wiedergabetempos der Live-Elektronik.

d) Immerhin wurden so durch einen einzigen Sequenzer, der im elektronischen Teil von SYN tausende von Noten und Anweisungen zu verwalten hat, drei Synthesizer und ein zusätzlicher Monitor eine halbe Stunde lang problemlos - auch in den Proben - gesteuert. Gerade auch in den Proben war die mühelose Verfügbarkeit, Flexibilität und Geschwindigkeit des live-elektronischen Mediums für Computerlaien, und nicht nur für diese, ein Grund zum Staunen.

IX) Ästhetische Wertungsversuche

a) Erfahrungsbericht

Nach diesen ausführlichen und überwiegend technischen Beschreibungen meines Vorgehens bei der Realisierung von SYN seien in Kürze noch einige eher ästhetisierende Punkte geäußert. Sie bringen - teils schlagwortartig - Reflexionen und Gefühle über mein Verhältnis zum musikalischen Produkt zum Ausdruck auf dem Hintergrund der beschriebenen Herstellungsweise.

1) Mein größtes musikalisches Problem während der Ausarbeitung von SYN bestand darin, daß der Rechner nach meinen Anweisungen mir fremdes Material 'komponierte'.

2) Da die Verarbeitung des musikalischen Materials immer schichtenweise erfolgte - meist die Streicher separat, dann die Holzbläser, die Blechbläser, das Schlagzeug und die Elektronik je getrennt voneinander, wurde zwar einerseits innerhalb der Schichten nichts 'vergessen', andererseits entstanden dann bei der Überlagerung aller Schichten öfters klangfarbliche und dynamische Inkompatibilitäten. Diese wurden allerdings durch die gemeinsame Verarbeitungsrichtung teilweise neutralisiert, sodaß dann doch wieder ein übergeordneter einheitlicher Eindruck entstehen konnte. Außerdem verfügt mein Musiksystem über einige effiziente harmonische und rhythmische Bearbeitungsprogramme, die ich aber in SYN aus Zeitgründen nicht einsetzen konnte.

3)a) Der Rechner tendiert zur Addition von Einzelem, nicht zur Synthese von sich aus. Daher muß in einer Nachbearbeitungsphase jede 'zufällig' erzeugte Beziehung zwischen Klangelementen von Hand verstärkt werden.

b) Dem Rechner sind Spielschwierigkeiten und spieltechnische Befindlichkeiten des Spielers, die ja beim Komponieren mitgedacht werden müssen, nur sehr umständlich - wenn überhaupt - zu vermitteln. Auch deshalb mußte jede Note des Live-Parts im ganzen Stück von Hand mehrfach nachbearbeitet werden.

4) Ebenso mußten im Editor des Sequenzers alle spieltechnischen, artikulatorischen und dynamischen Anweisungen von Hand ins Notenbild eingetragen werden. Solche Prozesse sind stark kontextbezogen. Daher wäre

diesbezügliche Software zwangsläufig kompliziert zu programmieren und schwer objektivierbar.

Andererseits gibt dieser 'Mangel' aber dem Komponisten auch Gelegenheit, jedes Detail seiner Komposition, gerade weil vieles vom Rechner erzeugt wurde, genau kennen zu lernen.

5) Eine große spieltechnische Entlastung bedeutete die Verwendung der Sample-Player. Viele erzeugte und nur unter größten Mühen spielbare Schlagzeugschichten (aber auch anderer Instrumente) konnten problemlos 'ausgelagert' und dadurch die Musiker entlastet werden.

b) Rechtfertigung für algorithmisches Komponieren

Zum Schluß dieses Arbeitsberichtes will ich noch vier Thesen vorbringen, die allen Schwierigkeiten und Unzulänglichkeiten dieser Arbeitsweise sowie ihrer Resultate zum Trotz die Notwendigkeit algorithmischer Komposition untermauern soll.

1) Alles, was gedacht werden kann, ist auf den Computer übertragbar. Algorithmische Komposition bedient sich daher nur der gewaltigen Kapazitäten dieser Maschine, wenn musikalisch konsequent Durchdachtes technisch korrekt dem Rechner formuliert wird.

2) Ich selber bin neugierig auf die von meinen eigenen Algorithmen errechneten 'fremden' musikalischen Materialien, bin aber zugegebenermaßen auch fasziniert vom algorithmischen Darstellen musikalischer Probleme, als Vorspeise zum musikalischen Hauptgang.

3) Bei Verwendung von Algorithmen während aller Kompositionsphasen zwingt der Rechner zur Systematisierung auch der Kompositionsvorgänge selber. Die Ausprägung eines Kompositionssystems, mein erklärtes Ziel, läßt sich mit einem Computer viel stringenter erreichen als 'zu Fuß'.

4) Schließlich sind Computersprachen als Vermittler zwischen Benutzer und Maschine zum einheitsstiftenden Werkzeug in den Wissenschaften geworden, ja, wie meine Arbeit beweist, auch zwischen Wissenschaft und Kunst. Hierin offenbart sich mir die Unteilbarkeit des Geistes, hierin sehe ich aber auch ein Mittel der Solidarität zwischen meiner Kunst und der realen Welt industrieller Arbeit. Vielleicht wird letztere sogar ein klein wenig beflügelt durch innovative Unterfangen wie SYN, obwohl ich mir über die Entfernungen zwischen beiden Welten keine Illusionen mache, ebenso wenig über die Richtung, von welcher die intellektuellen Innovationsschübe herkommen.

Georg Wötzer, geboren 1946 in Stuttgart; Schulmusik - und Kompositionsstudium (Komposition bei Martin Gumbel). Seit 1994 hauptamtlicher Dozent für Musiktheorie und Computermusik an der Musikhochschule Stuttgart.

Bsp. 1

MUSIC-MENÜ	Starte	Erzeuge	Transform	Korrektur	Ausgabe	GFA-Progs
Gestalt(Harmonik->Akkordauswahl) Gestalt(Harmonik->Akkordprogression) Gestalt(Transpositionen) Gestalt(Zeitumwandlungen) Gestalt(Accelerando-Rallentando) Gestalt(Anpassung an neue KFSkala) Listen (Anpassung an neue KFSkala) Gestalt(Skalenfilter) Gestalt(Horizontal-Spiegelung) Gestalt(Vertikal-Spiegelung) Gestalt(Metrum aus Ls-u.Rh-Modifikation) Gestalt(Crescendo-Decrescendo) Gestalt(Repetition von Fragmenten) Gestalt(Mergen von 2 Gestalten) Gestalt(Pressen/Dehnen:Kf,Th,Ls) Gestalt(nach iDim transformieren) Gestalt(automatisch einz.Para transform.)						

Bsp. 2

Reduktionsmechanismus

	<u>Grundgestalt</u>	<u>1. Reduktion</u>	<u>2. Reduktion</u>	<u>3. Reduktion</u>
gesetzte Schicht Nummer	Modulzahl pro Schicht	Modulzahl pro Schicht	Modulzahl pro Schicht	Modulzahl pro Schicht
I	3			
II	8	6	4	
III	2			
IV	6	4	2	1
V	1	1		
Schichtzahl	5	3	2	1

V

sehr langsam ♩ = 42

arco.5seg
V1_live mp mp mf sempre cresc

arco.5seg
V2_live mp mp mf

arco.5seg
Vla_live mp mp mf

Vc_live

Kb_live

inner ord
Kl1_Live mf

Kl2_live

ord
Bkl_live f

Trp_live

Hrn1_live

arco
Hrn2_live

f
Pos1 (ohne Dpfr)

f
Pos2 (ohne Dpfr)

Met_sz1 nimmt Eisenschl
Eisenschl
Fell_sz1 und Trst f sempre cresc

(hat Trst)
Tri.R
Kuppe
pp. R
Kuppe
Fell_sz2 mp p ff f

Holz_sz2

Mar_sz2

Fell_sz3 F.A.

sempre poco cresc
mf sempre poco cresc

Pkeh_sz3

Synthis

Bsp. 4

SYN *rit. (s=128) (Santapala)* - 37 - (G34) *acc. b)* *Kritisch & unreg. fallen*

Instrumentation:
V1, V2 (Violins)
MG (Viola)
Vlc (Violoncello)
Vc (Violone)
Kl (Klarinetten)
Kl1, Kl2 (Klarinetten I & II)
Dkl (Fagott)
Trp (Trompeten)
Tm1, Tm2 (Trommeten I & II)
KlAL (Klarinette Alt)
Pos1, Pos2 (Posaunen I & II)
Vibr (Vibraphon)
Mars (Marschpauke)
Phe (Pauken)
HG (Horn)
HG (Horn)
HG (Horn)
Z-kl (Zymbel)
HC (Cymbel)
Z-Vg (Zymbel)
Kb (Kontrabaß)

Dynamic Markings: *mp*, *mf*, *f*, *p*, *acc.*, *rit.*, *cresc.*, *decc.*, *unreg. fallen*, *wolte*, *mf*.

(Bsp. 4)

JYN

- 38 -

(C74)
G276)

(Handwritten note)

V1

V2

Vi2

Vc

Kl2
(Kl. 2nd)

Kl1

Kl2

DKl

Trop

Hrn. 1

Hrn. 2

Pos. 1

Pos. 2

I. Vibr

II. Accor

Perk. Perk

Conga (+)

Schlagwerk

Hr. Holz, unreg. füllend

Hr. Holz

molto p

I
rasch ♩ = 138

Score for page 102, featuring various instruments and dynamic markings. The score includes:

- V1_live
- v2_live
- Vla_live
- Vc_live
- Kb_live
- Kl1_live
- Kl2_live
- Bkl_live
- Tr_live
- Hrn1_live
- Hrn2_live
- Pos1_live
- Pos2_live
- Met_sz1
- Fell_sz1 hat Bürsten, Flzschl
- Met_sz2
- Fell_sz2
- Holz_sz2
- Mar_sz2
- Pke_sz2
- Synthies

Dynamic markings include *mp*, *mf*, *f*, *ff*, *p*, and *pp*. Performance instructions include "nimmt Spitzdämpfer" and "Dämpfer aufsetzen".

Bsp. 5

Zeitstruktur SYN

Ausggest.Nr	Sec-Ausd	Übergang:Sec-Ausd	Zielgest.Nr	Sec-Ausd
1	2.22	120	2	12.5
2	12.5	180	3	10.5
3	10.5	90	4	9.1
4	9.1	60	5	22.5
5	22.5	120	6	31.4
6	31.4	30	7	7.8
7	7.8	20	8	2.8
8	2.8	60	9	4.3
9	4.3	20	10	3.8
10	3.8	60	11	32.7
11	32.7	90	12	21.4
12	21.4	0	13	15
13	15	120	14	11.7
14	11.7	60	15	15
15	15	30	16	6.5
16	6.5	30	17	16.3
17	16.3	15	18	10
18	10	30	19	16.3
19	16.3	0	20	2.1
20	2.1	120	21	20
21	20	120	22	31.6
22	31.6	180	23	27.3

Zeitstruktur der Teile

Teil I : Gestalt 1 - 5 ; Ausdehnung: 656"

Teil II : Gestalt 6 - 12 ; Ausdehnung: 367.8"

Teil III : Gestalt 13 - 20 ; Ausdehnung: 502.9"

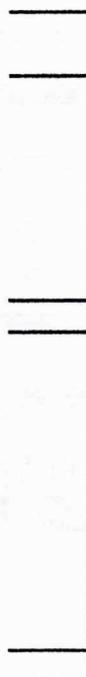
Teil IV: Gestalt 21 - 23 ; Ausdehnung: 358.9"

Gestalt-Nr

Art

Bsp. 6

1	nicht-tonal
2	tonal
3	nicht-tonal
4	tonal
5	nicht-tonal
6	nicht-tonal
7	nicht-tonal
8	nicht-tonal
9	nicht-tonal
10	nicht-tonal
11	tonal
12	tonal
13	nicht-tonal
14	nicht-tonal
15	nicht-tonal
16	nicht-tonal
17	nicht-tonal
18	nicht-tonal
19	nicht-tonal
20	nicht-tonal
21	nicht-tonal
22	tonal
23	tonal



d-moll

d-moll

'Tonale Disposition' in SYN

gis-moll → F-Dur

cis-moll → gis-moll

B-Dur → F-Dur
F / D - Tonalität

Bsp. 7

File Library Edition Creation Utility **Steinberg** **PROTEUS**
 * einst_prol * -SynthWorks-

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ALTO SAX	VIBRA	MARIMBA	CYMBALS	STEEL DRUMS	METAL TOYS	TOM MS	CONG TRANS	MEDICI NEDRUM	SNARE DRUMS	WOOD_B LOCKS	ROCK DRUMS	LATIN PERC.	PERCUS SIVES	ZISCHE SCHE	REP_ZI
64	68	72	76	80	84	88	92	96	100	104	108	112	116	120	124
Prg	Prg	Prg	Prg	Prg	Prg	Prg	Prg								
MAIN OUTPUT	MAIN OUTPUT	MAIN OUTPUT	MAIN OUTPUT	MAIN OUTPUT	MAIN OUTPUT	MAIN OUTPUT	MAIN OUTPUT								
0	-4	0	0	-5	-5	0	0	0	+3	0	0	+3	0	0	0
127	112	127	127	127	127	127	127	127	127	127	127	127	127	127	127

Bsp. 8

DESK EDITOR DISK SOUNDS MIDI OPTIONS DIVERSES The K5-Soundmachine V3.3

HARMONIC: 86 57

SOUND: DEMO_KF **COPY**

CLEAR

S1 > S2
S1 < S2

UNDO

MODE : FULL

RANGE :

LIVE **DIE** **ALL**

S1 S2 BOTH

DDF : OFF
DDF MOD: OFF
CUTOFF : 5
CUT.MOD : 12
SLOPE : 23
SL.MOD : 6
FLAT LV: 12
VELO : 2
PRES : 1
KS : 3
ENV : 4
VEL ENV: 4
LFO : 2
MAX LV.: 5

S1 S2 BOTH

COARSE : 2
FINE : 0
KEY TRK: C-2
ENVELOP: 1
PRS : 1
BEND : 1
VEL. ENV: 1

8-12 12 -9 11 -4
19 18 31 16 23 19

DDA : OFF
ATK. VEL: 10
PRS. DEP: 6
KS. DEP: 3
LFO DEP: 2
ATACK : 7
RELEASE: 1
KS RATE: 3
MAX LV.: 2

MOD. : OFF	VELO DEP: 0	KS DEP : 0	SHADOW: ON	PRS DEP : 0	LFO DEP : 0	CURVE 1: OFF	CURVE 2: OFF	CURVE 3: OFF	CURVE 4: OFF
EFFECT : 7	EFFECT : 2	EFFECT : 0	EFFECT : 2	EFFECT : 0	EFFECT : 2	MAX LV.: 4	MAX LV.: 2	MAX LV.: 1	MAX LV.: 4