

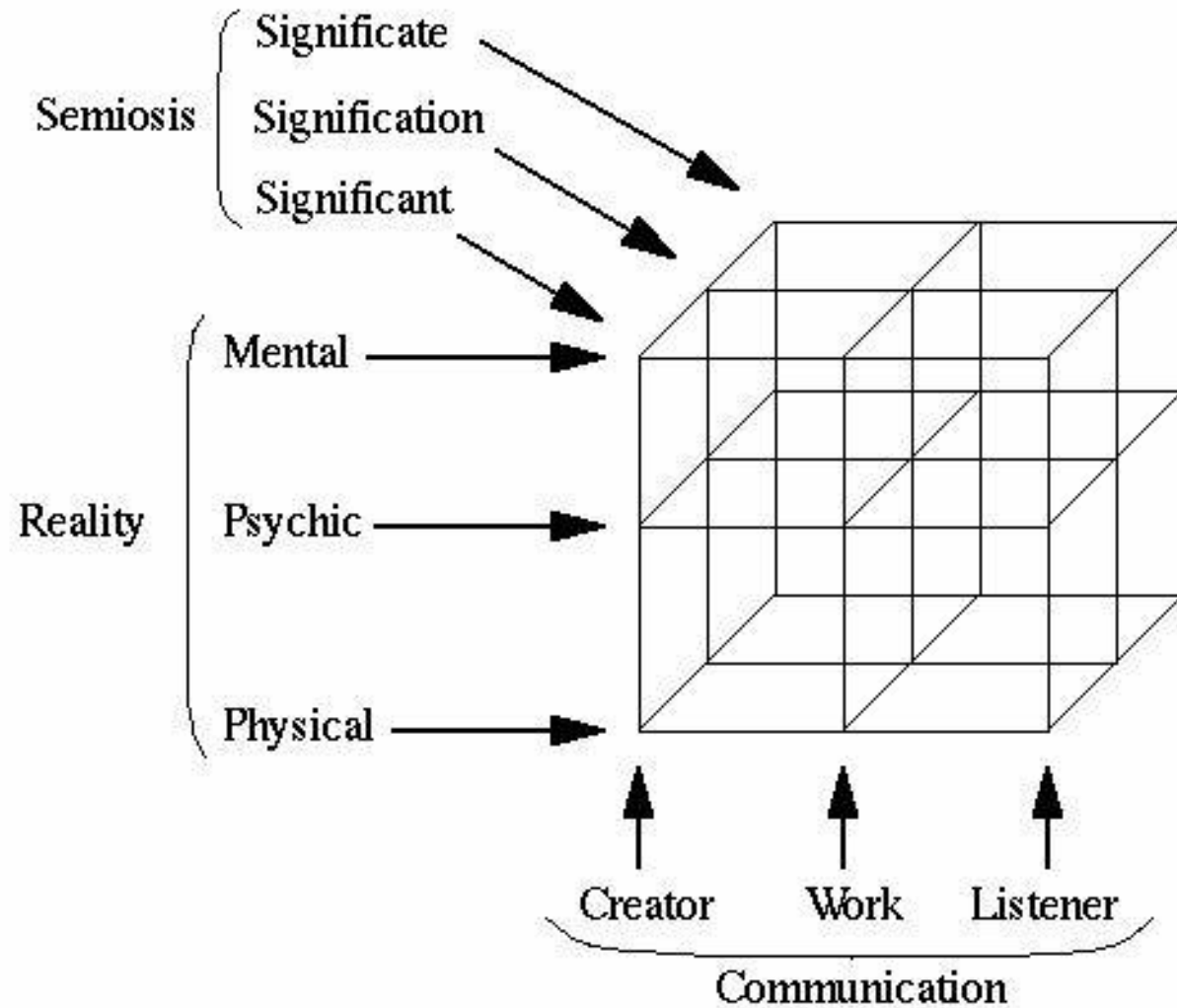
Kodierung musikalischer Objekte und Prozesse

Seminar: Musikinformatik
Florian Grothe

Überblick

- Musiktopographie
- Kodierung auf neutraler Ebene
- Kodierung von Sound-Events
- Kodierung von Note-Events
- Kodierung von Konstrukten
- Eulerraum der Tonhöhen
- Music N
- MIDI
- Andere Ansätze

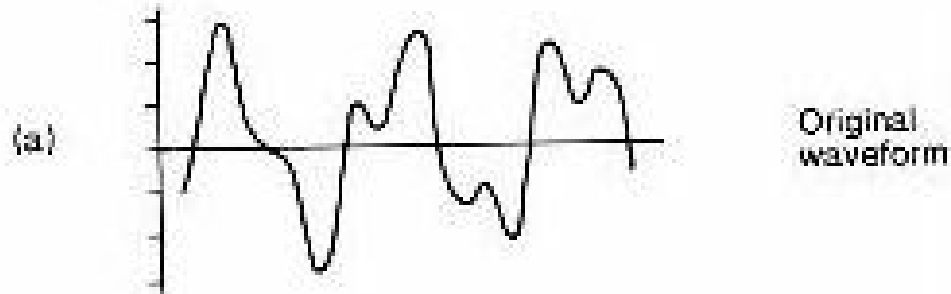
Musiktopographie



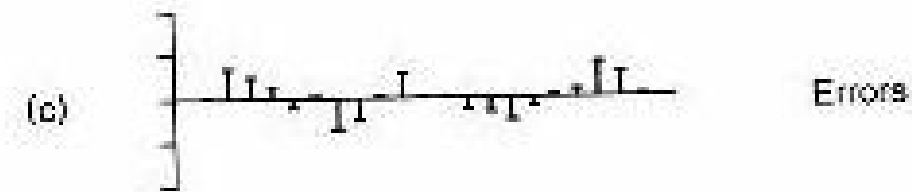
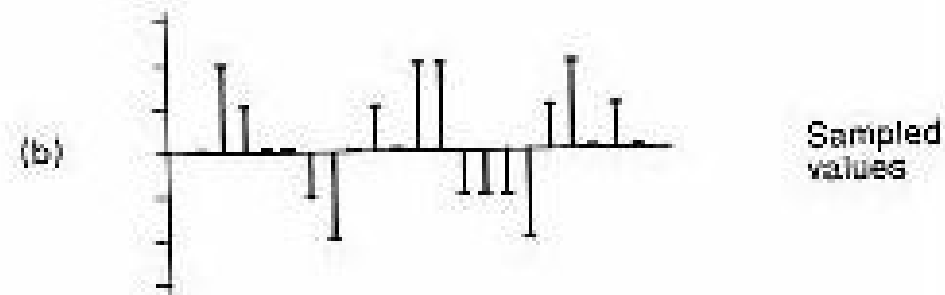
Kodierung auf neutraler Ebene (1)

- Neutrale Ebene ist im Koordinatenraum der Musik das Werk
- Musikalisches Werk ist akustisch gesehen eine einzige Luftschwingung
- Datenträger, wie CD oder LP sind analoge bzw. digitale Spur dieser Schwingung

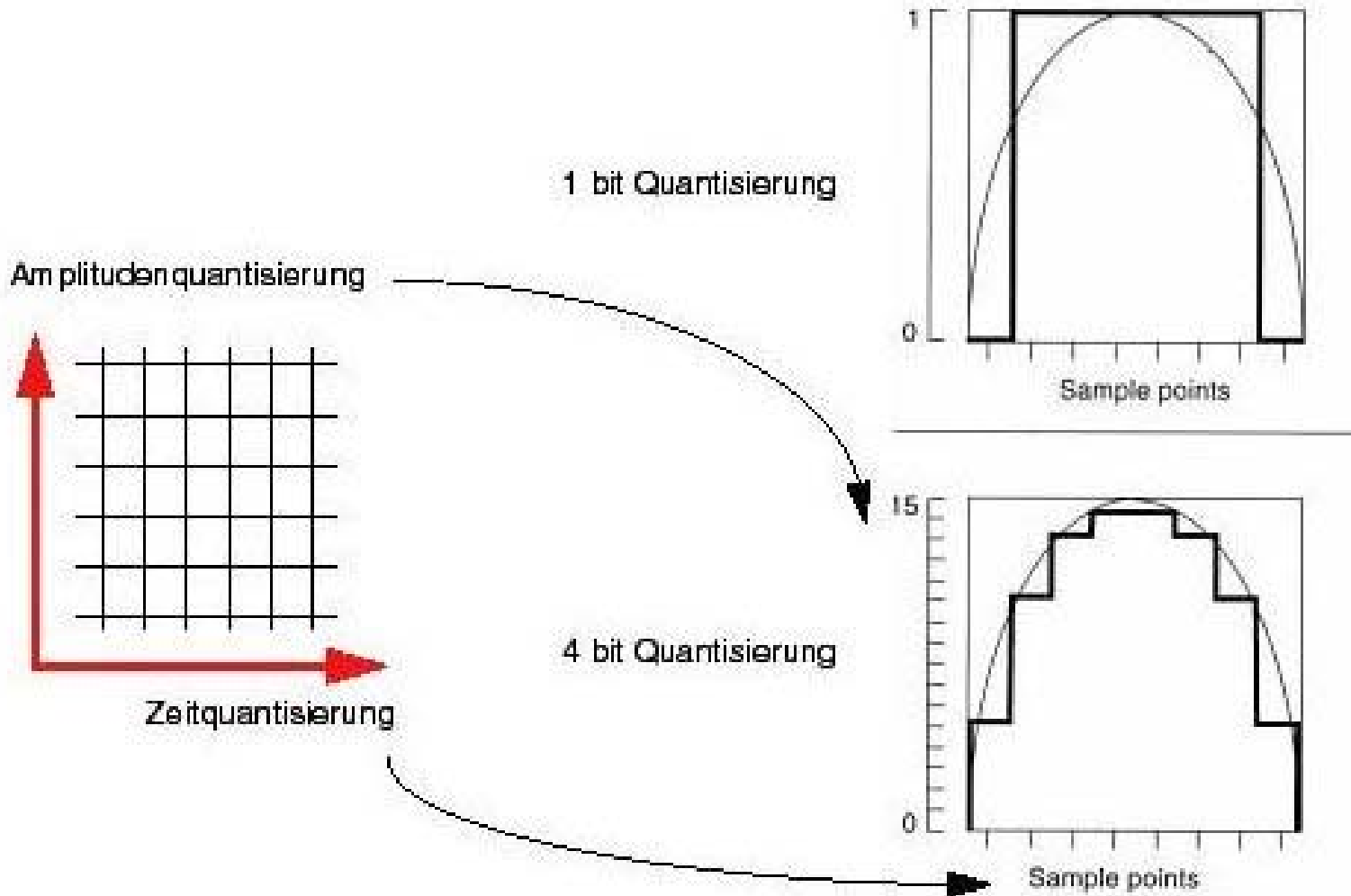
Kodierung auf neutraler Ebene (2)



CD: 1/44'100 sec
16 bit
pro Kanal



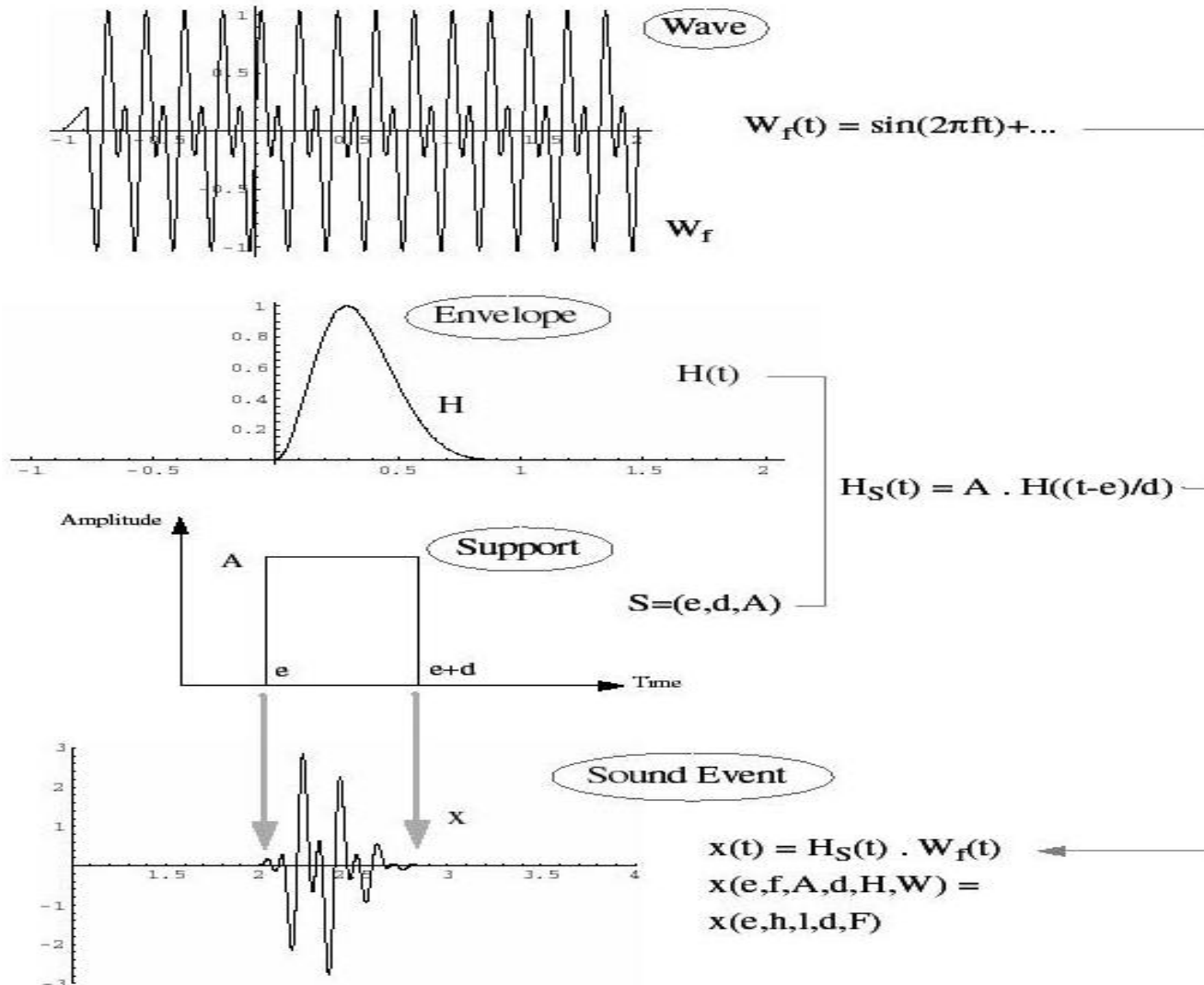
Kodierung auf neutraler Ebene (3)



Kodierung von Sound-Events (1)

- Aus Sicht des Schöpfers / Hörers wird das Werk akustisch betrachtet aus einer Liste von Sound-Events aufgebaut / zerlegt
- Zerlegung ist nicht eindeutig realisierbar!!!
- Annahme: Wir haben ein einzelnes Sound-Event isoliert (z.B. Klavier)
- Diese Auslenkung des Luftdrucks ist noch eine neutrale Beschreibung

Konstruktion von Sound-Events (1)



Konstruktion von Sound-Events (2)

- $X(\text{Einsatz, Frequenz, Amplitude, Dauer; Hüllkurve, Wave})$
- $X(e, f, A, d; H, W)$
- Farbparameter $F = (H, W)$
- Support $S = (e, d, A)$
- Anpassung der Parameter f und A an die menschliche Hörphysiologie

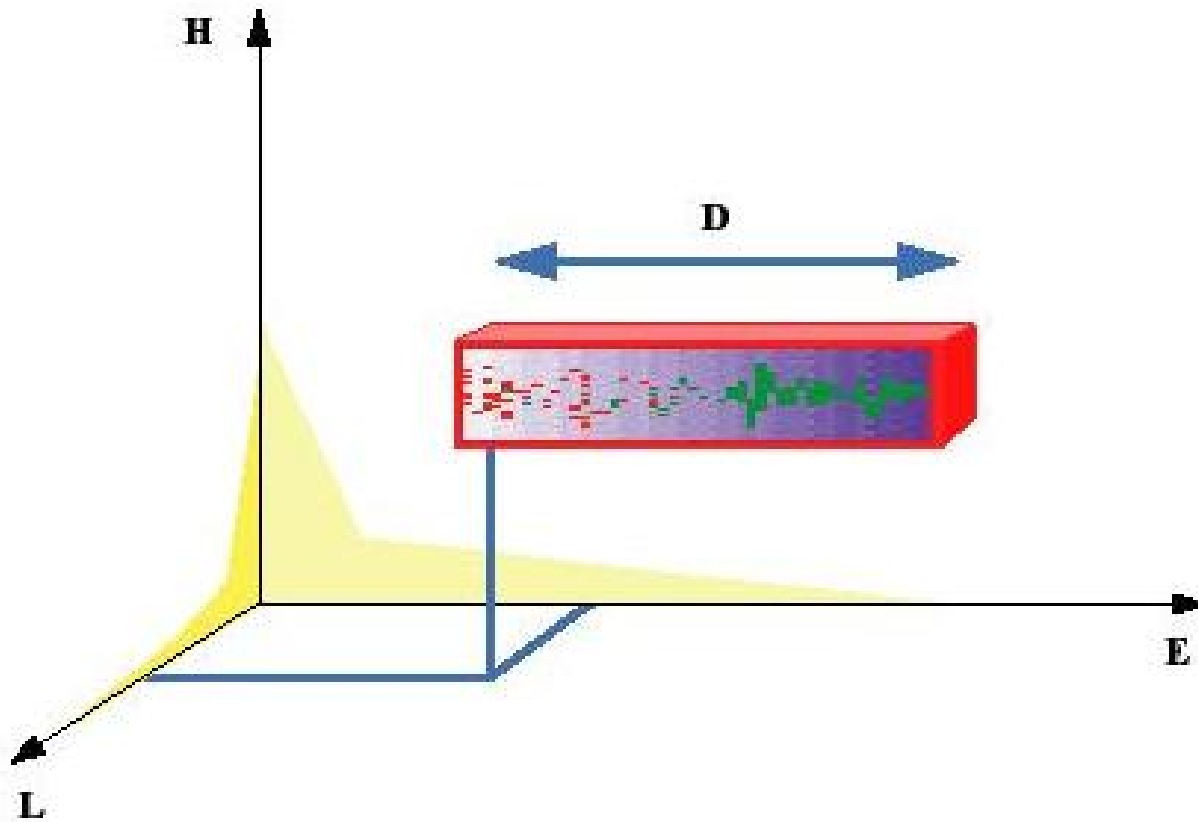
- $X(e, h, l, d; F)$
- $X(e, \text{Tonhöhe, Lautstärke, } d; F)$

Kodierung von Note-Events (1)

- Note-Event entspricht einem Objekt auf der mentalen Realitätsebene (Note der Partitur)
- Eine in der Partitur gesetzte Note X, wird beim Performen auf ein Sound-Event abgebildet.
- Die Parameter von X müssen auf der mentalen Ebene kodiert werden

Kodierung von Note-Events (2)

$X = \text{PianoNote}(E, H, L, D)$



Kodierung von Note-Events (3)

- Sonderfälle: Pause(E, D); Taktstrich(E)

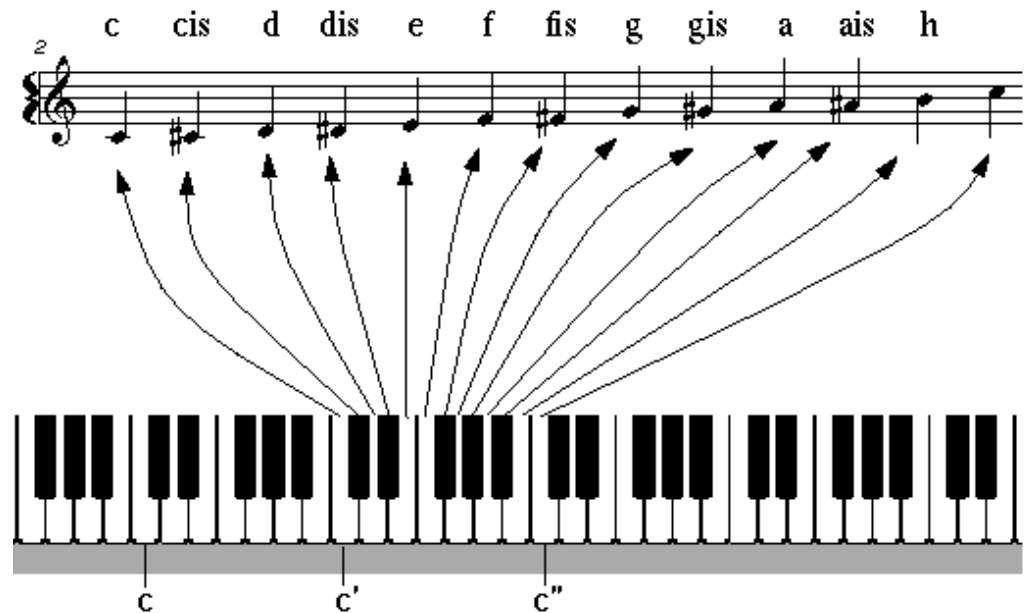
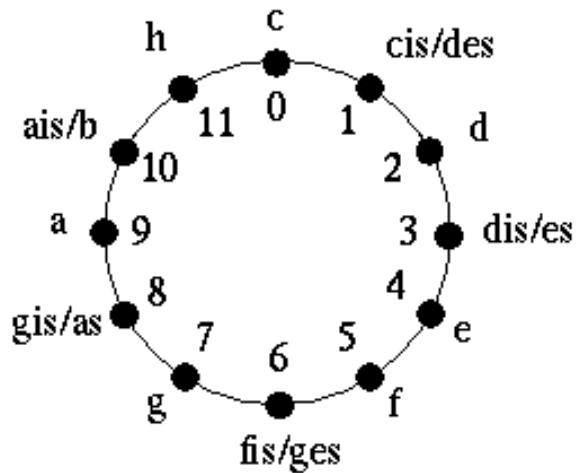
$X = \text{PianoNote}(E, H, L, D)$

- PianoNote verweist auf das Instrument (Klangfarbe)
- E, H, L, D sind reelle Zahlen
- E = (mentale) Einsatzzeit
- D = (mentale) Dauer

Kodierung von Note-Events (3)

X = PianoNote(E, H, L, D)

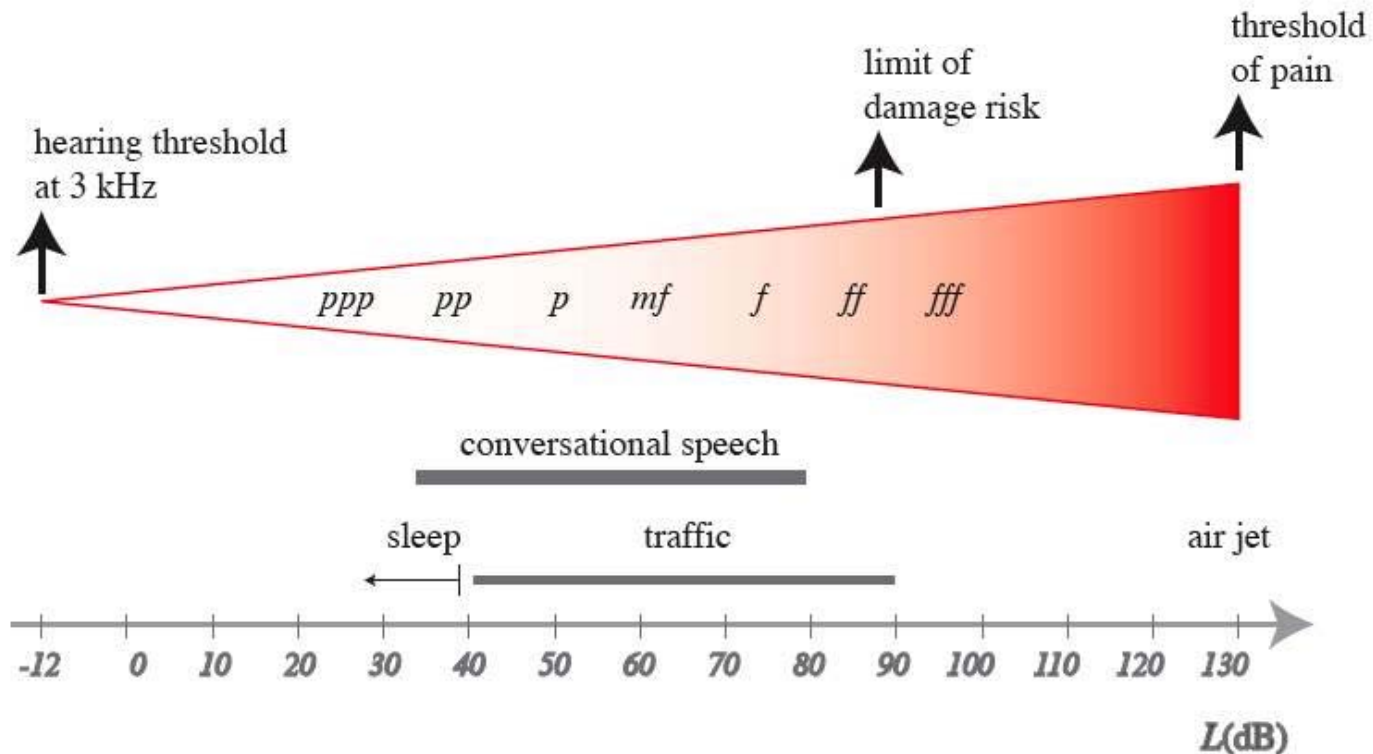
- H = (mentale) Tonhöhe



Kodierung von Note-Events (4)

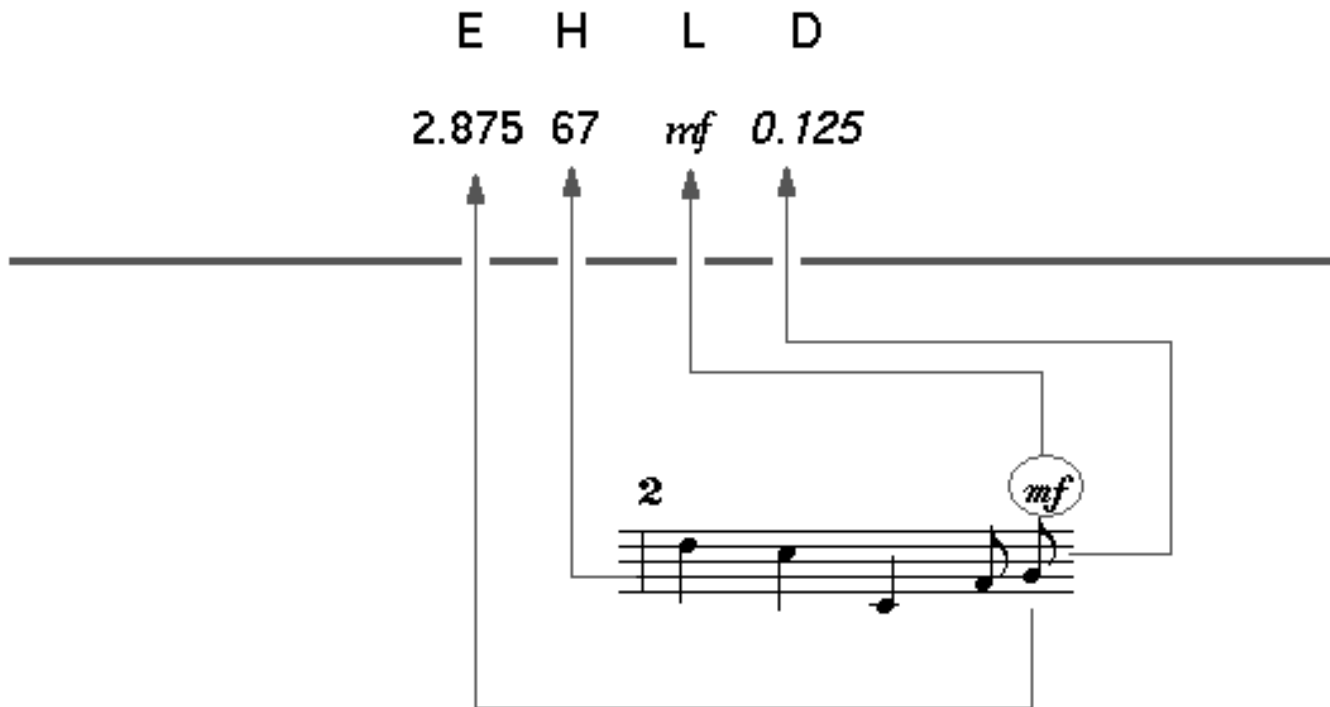
$X = \text{PianoNote}(E, H, L, D)$

- $L =$ (mentale) Lautstärke (nicht absolut, Frage der Kalibrierung)



Kodierung von Note-Events (5)

- $X = \text{PianoNote}(2.875, 67, \text{mf}, 0.125)$



Kodierung von Konstrukten (1)

Akkorde:

- Akkorde als Menge von Note-Events
- $A = \text{Akkord} \{X_1, X_2, \dots, X_n\}$
- Gilt nur, wenn alle Einsatzzeiten gleich sind

- Alternativ:
Akkord($\{E, \text{Einsatzgruppe}\{\text{Note}(H, L, D)\}\}$)

Kodierung von Konstrukten (2)

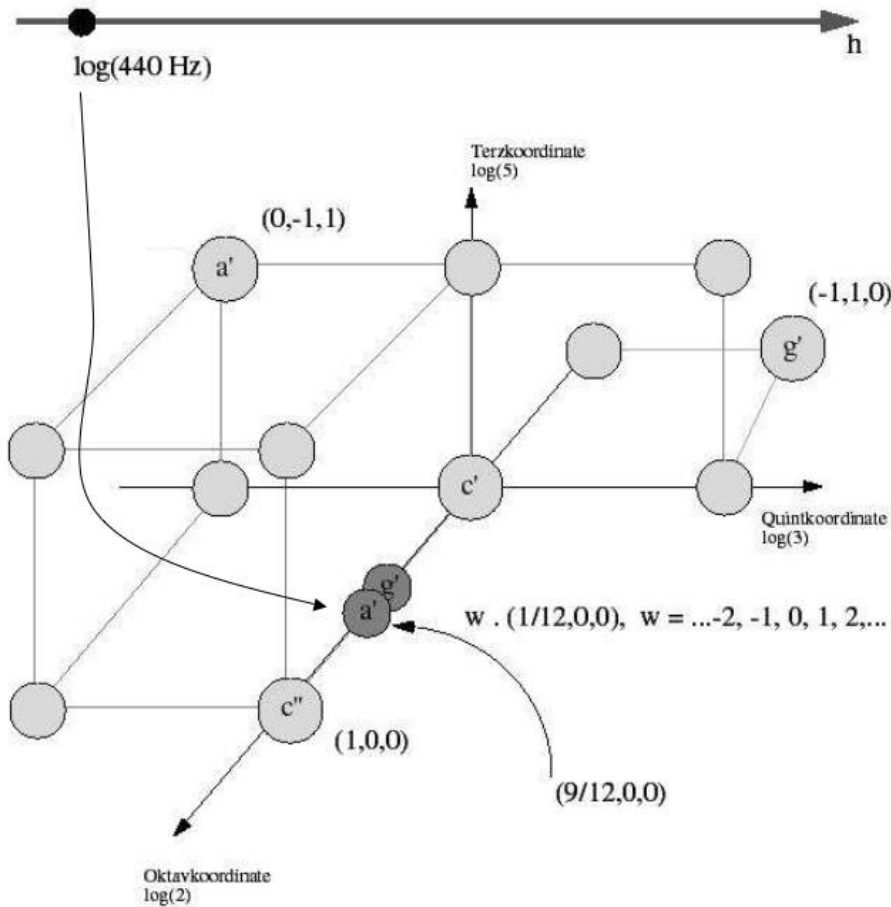
Gruppen:

- PianoGruppe{PianoNote}
- Part(PianoGruppe), Stimme(PianoGruppe)

Sequenzen:

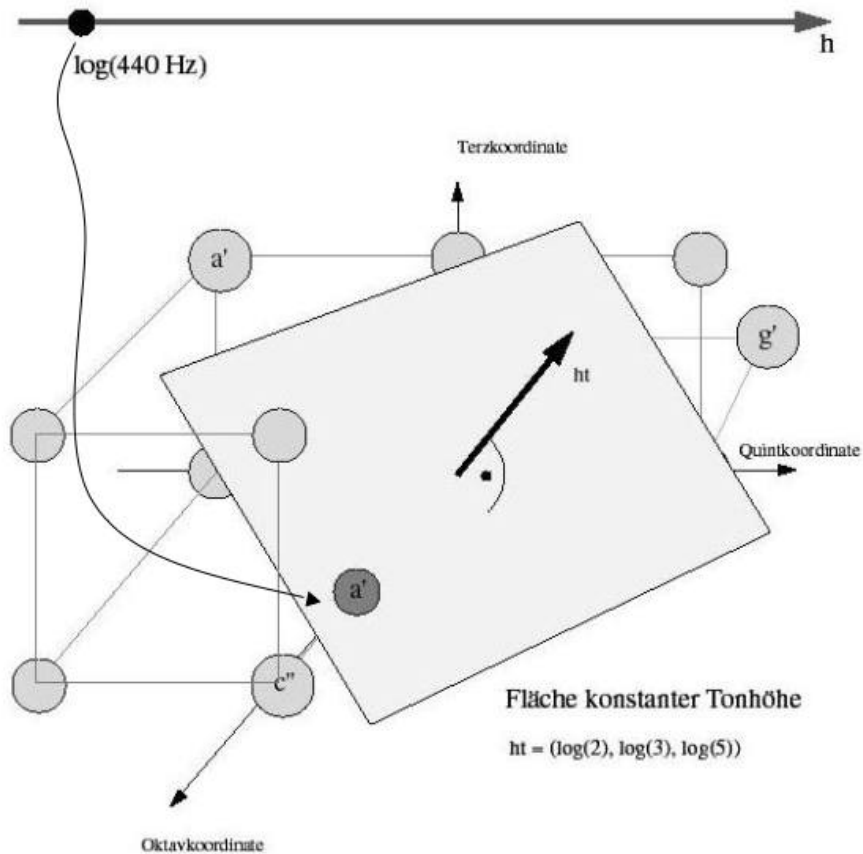
- Sequenz als Liste von Akkorden
- AS = Akkordsequenz[Akkord]
- Akkordsequenz[Akk1, Akk2, ..., Akkn]

Eulerraum der Tonhöhen (1)



- Betrachtungsraum
- Läßt alle Freq. der kl. Stimmung darstellen
- Oktav, Quint, Terz

Eulerraum der Tonhöhen (2)



- Fläche konstanter Tonhöhe
- Unendlich viele gleiche Tonhöhen
- Welche war im mentalen Model gemeint???

Music N (1)

- Familie von Repräsentationssprachen
- Es wird in 2 Dateitypen unterschieden
 - Orchestra File: Instrumentendefinitionen
 - Score File: Notenlisten, Wave- und Envelope-Tabellen
- Ein Sound-Syntheseprogramm kombiniert die beiden Dateien und generiert daraus Souns-Events und schreibt diese auf ein Sound File

Music N (2)

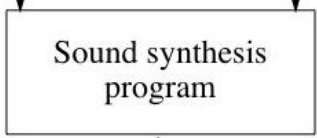
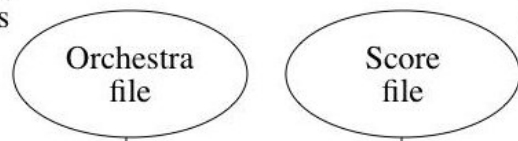
- Orchestra Language nutzt die Beschreibung von Sound-Events (Envelope und Wave aus Score File)
- Score Language enthält neben Envelope und Wave auch die Liste der Sound-Events:
 - p1 = Instrument
 - p2 = Einsatzzeit (physikalisch)
 - p3 = Dauer
 - p4 = Frequenz
 - p5 = Amplitude
 - P6, p7 = Filterdaten

Instrument definitions

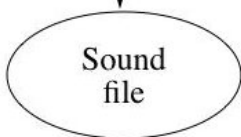
Note list and wavetable functions

Music N

- >Csound
- >Common Lisp Music
- >NeXT MusicKit
- >IRCAM Max
- > etc.

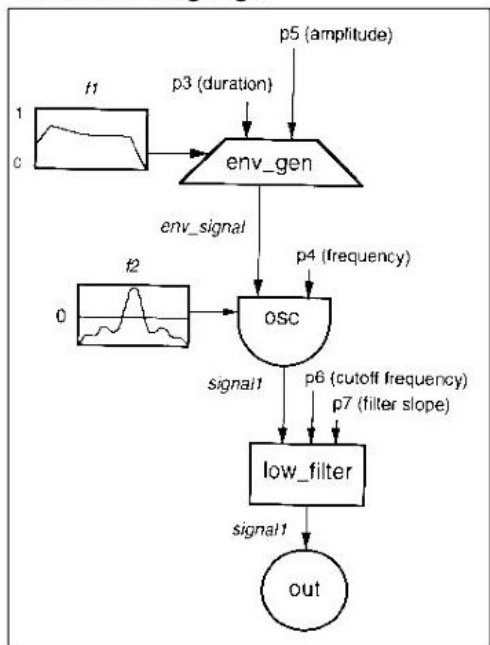


Samples



Score language

Orchestra language



Unit generator

```

/*
Function definitions in the form:
Frame, start time, table size, function generator type, arguments
*/
f1 0 1024 line_segment (0 0) (256 1) (512 .5) (768 .5) (1024 0);
f2 0 1024 fourier 11 (1 .4) (2 .3) (3 .05) (4 .06) (5 .04)
(6 .04) (7 .03) (8 .04) (9 .02) (10 .03)
(11 .01);

/* Instr. Start Dur. Freq. Amp. Filt.cutoff Filter
ID time frequency slope
p1 p2 p3 p4 p5 p6 p7
*/
i1 0 1.0 440.0 2000 4100 6;
i1 1.0 1.0 560.0 2000 3000 5;
i1 1.0 2.0 440.0 2000 2050 4;
i1 2.0 2.0 880.0 10000 9000; 3;

```

MIDI (1)

- Musical Instrument Digital Interface
- Kommunikation erfolgt über 3 unterschiedliche Ports:
 - MIDI-In
 - MIDI-Out
 - MIDI-Thru (zum durchschleifen)
- Serielle Übertragung

MIDI (2) Messages

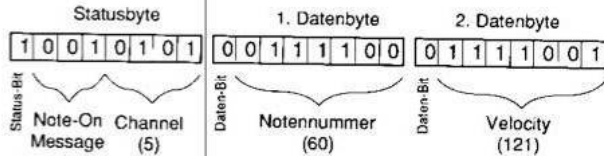
- 10 Bit-Wörter
- Botschaft besteht aus 1 Statusbyte und 2-3 Datenbytes
- Statusbyte: beginnt mit 1
 - Gibt Befehlstyp und Adresse an
- Datenbytes: beginnen mit 0
 - Geben die Werte an, die den Befehl spezifizieren

MIDI (3) Messages

- Channel Voice:
 - Nachrichten die einzelne Kanäle betreffen
- Channel Mode:
 - Betreffen das Zusammenspiel der Kanäle
- SystemExclusive:
 - Herstellerspezifische Nachrichten
- System Common:
 - Systemeinstellungen, Position im akt. Musikstück
- System Realtime:
 - Zeitinformationen, Timing Clock

Befehlstyp und seine Adresse im Gerät

Werte zum Befehl
2-3 Datenbytes üblich, aber mehr möglich



Message-Hierarchie

Message				
Channel		System: Statusbyte = 1111...		
Channel Voice	Channel Mode	System Exclusive	System Common	System Real Time

hex	Statusbyte	Datenbyte	Message
\$Bn	1011nnnn	0ccccccc = 121 0vvvvvvv = 0	Reset all Controllers
\$Bn	1011nnnn	0ccccccc = 122 0vvvvvvv = 0/127	Local Control Off/On
\$Bn	1011nnnn	0ccccccc = 123 0vvvvvvv = 0	All Notes Off
\$Bn	1011nnnn	0ccccccc = 124 0vvvvvvv = 0	Omni Mode Off
\$Bn	1011nnnn	0ccccccc = 125 0vvvvvvv = #Kanäle	Omni Mode On
\$Bn	1011nnnn	0ccccccc = 126 0vvvvvvv = 0	Mono Mode On Poly Mode Off
\$Bn	1011nnnn	0ccccccc = 127 0vvvvvvv = 0	Poly Mode On Mono Mode Off

hex	Statusbyte	Datenbyte	Message	Beschreibung
\$8n	1000nnnn	0kkkkkkk 0vvvvvvv	Note Off	kkkkkkk: Key # vvvvvvv: Velocity
\$9n	1001nnnn	0kkkkkkk 0vvvvvvv	Note On	kkkkkkk: Key # vvvvvvv: Velocity
\$An	1010nnnn	0kkkkkkk 0vvvvvvv	Polyphonic Key Pressure	kkkkkkk: Key # vvvvvvv: Velocity
\$Bn	1011nnnn	0ccccccc 0vvvvvvv	Control Change	ccccccc: Controller # vvvvvvv: Value
\$Cn	1100nnnn	0ppppppp	Program Change	ppppppp: Program #
\$Dn	1101nnnn	0vvvvvvv	Channel Pressure	vvvvvvv: Pressure
\$En	1110nnnn	01111111 0mmmmmmmm	Pitch Bend	1111111: LSB Wert mmmmmmmm: MSB Wert

Delta Time in Ticks = 480 pro Viertelnote



Off der ersten Note und On der zweiten Note

Delta Time (hex)	Status (hex)	Num (hex)	Vel (hex)	Interpretation	Musical Description
0	90	34	35	Note On, channel 1, note=52, vel=53	E, octave 3, medium loud
120	34	00		(Running Status) note=52, vel=0	release E3 after 16th note
0	37	26		(Running Status) note=55, vel=38	G3, medium soft
60	37	00		(Running Status) note=55, vel=0	release G3 after 32nd note
0	3B	28		(Running Status) note=59, vel=40	B3, start crescendo
60	3B	00		(Running Status) note=59, vel=0	release B3 after 32nd note
0	40	2B		(Running Status) note=64, vel=43	E4, continue crescendo
60	40	00		(Running Status) note=64, vel=0	release E4 after 32nd note
0	43	2D		(Running Status) note=67, vel=45	G4, continue crescendo
60	43	00		(Running Status) note=67, vel=0	release G4 after 32nd note
0	47	2F		(Running Status) note=71, vel=47	B4, continue crescendo
60	47	00		(Running Status) note=71, vel=0	release B4 after 32nd note
0	4C	32		(Running Status) note=76, vel=50	E5, continue crescendo
60	4C	00		(Running Status) note=76, vel=0	release E5 after 32nd note
0	4F	3A		(Running Status) note=79, vel=58	G5, medium loud with metrical accent
360	4F	00		(Running Status) note=79, vel=0	release G5 after dotted 8th note
0	4F	2A		(Running Status) note=79, vel=42	G5, softer
120	4F	00		(Running Status) note=79, vel=0	release G5 after 16th note
0	4F	42		(Running Status) note=79, vel=66	chord: G5, medium loud with accent
0	48	37		(Running Status) note=72, vel=55	C5, medium loud
0	45	37		(Running Status) note=69, vel=55	A4, medium loud
0	3C	37		(Running Status) note=60, vel=55	C4, medium loud
0	39	37		(Running Status) note=57, vel=55	A3, medium loud
0	34	37		(Running Status) note=52, vel=55	E3, medium loud
480	4F	00		(Running Status) note=79, vel=0	release G5 after quarter note
0	4E	23		(Running Status) note=78, vel=35	F#5, medium soft
480	4E	00		(Running Status) note=78, vel=0	chord: release F#5 after quarter note
0	48	00		(Running Status) note=72, vel=0	release C5 (after half note)
0	45	00		(Running Status) note=69, vel=0	release A4 (after half note)
0	3C	00		(Running Status) note=60, vel=0	release C4 (after half note)
0	39	00		(Running Status) note=57, vel=0	release A3 (after half note)
0	34	00		(Running Status) note=52, vel=0	release E3 (after half note)

Andere Ansätze

- Idee der Klassifizierung von Musikinstrumenten von Hornbostel und Sachs von 1914
- Object Coding Of Harmonic Sounds Using Sparse And Structured Representations
- Object Coding Of Musical Audio Using Bayesian Harmonic Models

Hornbostel-Sachs

4 - an aerophone

42 - the vibrating air is enclosed within the instrument

423 - the player's lips cause the air to vibrate directly (as opposed to an instrument with a reed like a [clarinet](#), or an edge-blown instrument, like a [flute](#))

423.1 - the player's lips are the only means of changing the instrument's pitch (that is, there are no valves as on a trumpet)

423.12 - the instrument is tubular, rather than being a [conch](#)-type instrument

423.121 - the player blows into the end of the tube, as opposed to the side of the tube

423.121.2 - the tube is bent or folded, as opposed to straight

423.121.22 - the instrument has a [mouthpiece](#)

Object Coding Of Harmonic Sounds Using Sparse And Structured Representations

- Zerlegung der Objekte anhand eines Wörterbuches, welches Töne bestimmter Musikinstrumente speichert.
- Größte Übereinstimmung wird vom Ursignal abgezogen und auf das Bild gerechnet
- Resultate: 2kbs; für 1- und 2-Stimmige Musikstücke
- Verbesserung: mehr Instrumente; Zerlegung mit Wahrscheinlichkeiten; Quantisierungen; Percussions