March 25th 2009
Computational Thinking Seminars
School of Informatics
University of Edinburgh

Michael Edwards
Music, ACE, University of Edinburgh
http://www.sumtone.com/michael.php
michael.edwards@ed.ac.uk

Computational Thinking in Music Composition

Despite the still-prevalent but essentially nineteenth century perception of the creative artist, an algorithmic approach to music composition has been in evidence in western classical music for at least one thousand years. The history of algorithmic composition—from both before and after the invention of the digital computer—will be presented along with musical examples of the distant and recent past. The author's own work will then be placed in this context, focussing upon recent compositions for instruments and computer created with custom software developed in Common Lisp.

Algorithmic Composition: Background / History

Models of musical process are arguably natural to human musical activity.

Listening involves both enjoyment of the sensual sonic experience and the setting up of expectations and possibilities of what is to come:

"Retention in short-term memory permits the experience of coherent musical entities, comparison with other events in the musical flow, conscious or subconscious comparison with previous musical experience stored in long-term memory, and the continuous formation of expecations of coming musical events." (Christensen, "The Musical Timespace, a Theory of Music Listening", 1996)

This second, active part of musical listening is what gives rise to the possibility, the development of musical form:

“Because we spontaneously compare any new feature appearing in consciousness with the features already experienced, and from this comparison draw conclusions about coming features, we pass through the musical edifice as if its construction were present in its totality. The interaction of association, abstraction, memory and prediction is the prerequisite for the formation of the web of relations that renders the conception of musical form possible.” (Ligeti, 1966)

For centuries, composers have taken advantage of this property of music cognition to formalise compositional structure.

Around 1026 Guido d’Arezzo (the inventor of modern staff notation) developed a formal technique to set a text to music: a pitch was assigned to each vowel so the melody varied according to the vowels in the text.

The 14th and 15th centuries saw the development of isorhythm, where rhythmic cycles (“talea”) are repeated, often with melodic cycles (“color”) of similar or differing lengths.

Compositions based on number ratios are found throughout musical history.

E.g. Dufay's (1400-74) isorhythmic motet "Nuper Rosarum Flores" was written for the consecration of Florence Cathedral on March 25th 1436.

The rhythmic structure of Nuper Rosarum Flores is based on the ratios 6:4:2:3, these being the proportions of the nave, the crossing, the apse, and the height of the arch of the cathedral.

[show slide]

Musical Example: Dufay (1400-74):

Nuper Rosarum Flores (1436)

Mozart’s Musikalisches Würfelspiel (“Musical Dice”) is another example, where musical fragments are to be combined randomly, according to dice throws.

[show slide]

Mozart’s Musikalisches Würfelspiel (“Musical Dice”)

Mozart
Musikalisches Würfelspiel

Numbers over columns refer to eight parts of walz; numbers to left of rows indicate possible values of two thrown dice; numbers in matrix refer to measure numbers of four pages of musical fragments

ZAHLENTAFEL.

TABLE de CHIFFRES.

Erster Theil.

Premiere Partie.

	A	B	C	D	E	F	G	H
2	96	22	141	41	105	122	11	30
3	32	6	128	63	146	46	134	81
4	69	95	158	13	153	55	110	24
5	40	17	113	85	161	2	159	100
6	148	74	163	45	80	97	36	107
7	104	157	27	167	154	68	118	91
8	152	60	171	53	99	133	21	127
9	119	84	114	50	140	86	169	94
10	98	142	42	156	75	129	62	123
11	3	87	165	61	135	47	147	33
12	54	130	10	103	28	37	106	5

Such formalisation procedures have not been limited to religious or art music.

The Quadrille Melodist, sold by Prof Clinton of the Royal Conservatory of Music, London, in 1865, was marketed as a set of cards which allowed a pianist to generate quadrille music (similar to a square dance); apparently 428 million quadrilles could be made with the system.

The Geniac Electric Brain of 1956 allowed customers to build a computer with which they could generate automatic tunes.

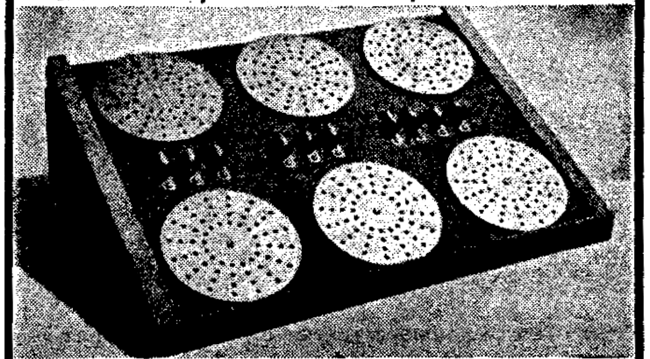
[show slide]

New! A MACHINE THAT COMPOSES MUSIC!

Actual tune composed on Geniac

GENIAC ELECTRIC BRAIN

COMPUTES, PLAYS GAMES, "REASONS"!



BUILD IT YOURSELF in a few hours!

Using a statistical analysis of simple tones plus the special circuitry of GENIAC, the Electric Brain Construction Kit, you can compose original tunes automatically. These new circuits are not available elsewhere!

GENIAC gives a wonderful introduction to the logic machines of our modern age! Over 30 machines can be built with the kit including: adding, subtracting, multiplying and dividing machines, in binary and decimal. There is also a binary to decimal converter.

Syllogisms, computing and coding machines, as well as game playing circuits (think you can beat the machine at Tic-tac-toe and Nim), actuarial analysis and intelligence testing are performed by the 34 different machines you can set up with this kit.

The machines are easy to build—all tools come with the kit and safe—requiring only one flashlight battery for power. Simple enough for a child to build (with complete wiring diagrams), they are fascinating for adults.

GENIAC is a comprehensive introduction to basic computer and problem solving circuits.

WHO ARE OUR CUSTOMERS?

International Business Machines • Westinghouse Electric • Remington-Rand • Albert Einstein Medical College • Barnard College • Naval Research Laboratories • Los Angeles Public Schools • Bell Telephone Laboratories • and thousands of other satisfied customers.

WHAT COMES WITH THE KIT?

Each kit comes with a 64 page experimental manual (Simple Electric Brains and How to Make Them), Wiring Diagrams, a Beginners Manual, GENIAC Study Guide listing additional readings in computer fundamentals, and Minds and Machines a 200 page text on computers and automation, plus a display and assembly rack and plastic parts tray. Over 400 parts and components.

$19.95

Computer-based Algorithmic Composition

After WWII, western classical music composition continued to develop the serial procedures developed by Arnold Schnberg (1874-1951).

Several composers, notably Xenakis and Ligeti, offered criticisms and alternatives to serialism but interestingly their music was often also governed by complex, even algorithmic, procedures.

The complexity of new composition systems made their implementation in computer programmes ever more attractive.

The development of software algorithms in other disciplines has made cross-fertilization rife.

Thus some algorithmic composition techniques are inspired by systems outside the realm of music, e.g. Chaos Theory (Ligeti, Désordre), Neural Networks (Gerhard E Winkler, Hybrid II "Networks").

University of Illinois, Urbana-Champaigne

Lejaren Hiller (1924-1994) is widely recognised as the first person—as early as the mid-1950s—to apply computers to algorithmic composition.

The use of specially-designed, unique computer hardware was common at American universities of the day. Hiller used the Illiac computer of the University of Illinois.

His collaboration with Leonard Isaacson resulted in 1956 in the first computer-composed piece of music, "The Illiac Suite for String Quartet" — programmed in binary The algorithms involved 'random walks' to generate notes.

The algorithms involved 'random walks' to generate notes.

Scotland

Scotland was not slow on the uptake of computer-based algorithmic composition.

The Barr and Stroud Solidac composing computer was built at the University of Glasgow in 1959.

This was both an algorithmic composition device and digital sound generator.

Had a clock rate of 30KHz (!) and used paper tape readers.

The developers claimed it could generate about one billion trios in the style of Haydn.

Stochastic versus Deterministic procedures

A basic division in the world of algorithmic composition is between indeterminate and determinate models, i.e. those that use stochastic/random procedures (e.g. Markov chains, which we will implement in PD) and those whose results are fixed by the algorithms and never change no matter how often the algorithms are run.

Iannis Xenakis (1922-2001) was a pioneer of algorithmic composition and computer music.

"With the aid of electronic computers, the composer becomes a sort of pilot: pressing buttons, introducing coordinates, and supervising the controls of a cosmic vessel sailing in the space of sound, across sonic constellations and galaxies that could formerly be glimpsed only in a distant dream" (Xenakis, 1992)

Xenakis's Stochastic Music Programme (SMP) used formulae originally developed by scientists to explain the behaviour of gas particles (Brownian Motion).

Xenakis saw his stochastic compositions as clouds of sound, individual notes being the analogue of gas particles.

The choice and distribution of notes was decided by procedures that involved random choice, probability tables that weight the occurence of specified events against those of others.

Xenakis created several works with SMP, often more than one work with the output of one computer batch process (gaining access to the IBM 7090 was not easy).

"Eonta" (1963-4, 2 Trumpets, 3 Tenor Trombones and Piano) is one of Xenakis's works composed with SMP, in particular the massively complex opening piano solo.

[show slide]

Musical Example: Iannis Xenakis (1922-2001)

Eonta

1963-4

2 Trumpets, 3 Tenor Trombones and Piano

Like yet another algorithmic/computer music pioneer, Gottfried Michael Koenig (1926-), Xenakis had no compunction in adapting the output of his algorithms as he saw fit.

Indeed, Koenig believes that the transcription process (i.e. from computer output to musical score) is essential to the process.

Others, e.g. Hiller, believed that if the output of the algorithm is deemed insufficient, then the programme should be modified and the output regenerated.

Of course several algorithmic composition programmes (especially modern examples) include direct computer sound synthesis, thus obviating the need for transcription.

Ligeti: Désordre

György Ligeti (1923-2006):

His work is known to the general public mainly through its use in several Stanley Kubrick films:

- "2001: A Space Odyssey" ("Lux Aeterna" and "Requiem", used without Ligeti's permission and subjected to a protracted but failed lawsuit)
- "The Shining" ("Lontano")
- "Eyes Wide Shut" ("Musica Ricercata")

Although in the late 1950s he worked in the same studios as Cologne electronic music pioneers Stockhausen and Gottfried Michael Koenig he produced very little electronic music of his own.

His interest in science and mathematics however led to several pieces influenced by e.g. fractal geometry or chaos theory.

“Somewhere underneath, very deeply, there’s a common place in our spirit where the beauty of mathematics and the beauty of music meet. But they don’t meet on the level of algorithms or making music by calculation. It’s much lower, much deeper–or much higher, you could say.” (Ligeti quoted by Steinitz, Musical Times 3/96.)

[show slide]

Musical Example:

György Ligeti (1923-2006)

Désordre

from Études, Book 1 (1985)

Pierre-Laurent Aimard, piano

The main argument of Désordre consists of foreground and background textures:

[show slide]

- Foreground (accented, forte): two simultaneous instances of the same basic process (melodic/rhythmic: see below for details), one in each hand, both doubled at the octave, and using white note (right hand) and black note (pentatonic, left hand) modes.

- Background (piano): continuous, generally rising quaver pulse notes, centred between the foreground octaves, one in each hand, in the same mode as the foreground hand.


ÉTUDE 1: „DÉSORDRE"
Dédiée à Pierre Boulez
György Ligeti
1985
Molto vivace, vigoroso, molto ritmico 𝅝 = 76
Piano
sempre legatissimo possibile
sempre simile
Stets sehr sparsamer Gebrauch des Pedals / Utiliser la pédale très discrètement (pendant toute la pièce)

The basic foreground process consists of a melodic pattern cycle consisting of the following scale-step shape:

[show slide]

Ligeti, Désordre: Foreground melodic pattern

```
Right hand (white notes), 26 notes, 14 bars
Phrase a:    0  0  1  0  2  1 -1
Phrase a’: -1 -1  2  1  3  2 -2
Phrase b:    2  2  4  3  5  4 -1  0  3  2  6  5

Left hand (black notes), 33 notes, 18 bars
Phrase a:    0  0  1  0  2  2  0
Phrase a’:  1  1  2  1 -2 -2 -1
Phrase b:    1  1  2  2  0 -1 -4 -3  0 -1  3  2  1 -1  0 -3 -2 -3 -5
```

This is stated on successively higher (right hand, 14 times, 1 diatonic step transposition) and lower (left hand, 11 times, 2 diatonic steps transposition) degrees.

This creates a movement from the middle of the piano outwards to the high and low extremes.

The foreground rhythmic process consists of slower-moving irregular combinations of quaver-multiples that tend to reduce in duration over the melodic cycle repeats to create an accelerando towards continuous quaver pulses:

[show slide]

Ligeti, Désordre: Foreground rhythmic pattern (quavers)

		right hand:	left hand:
cycle 1:	a:	3 5 3 5 5 3 7	3 5 3 5 5 3 8
	a’:	3 5 3 5 5 3 7	3 5 3 5 5 3 8
	b:	3 5 3 5 5 3 3 4 5 3 3 5	3 5 3 5 5 3 3 5 5 3 3 5 3 5 3 5 5 3 8
cycle 2:		3 5 3 4 5 3 8	3 5 3 5 5 3 8
		3 5 3 4 5 3 8	3 5 3 5 5 3 8
		3 5 3 4 5 3 3 5 5 3 3 4	3 5 3 5 5 3 3 5 5 3 3 5 3 5 3 5 5 3 8
cycle 3:		3 5 3 5 5 3 7	3 5 3 5 5 3 8
		3 5 3 5 5 3 7	3 5 3 5 5 2 7
		3 5 3 5 5 3 3 4 5 3 3 5	3 4 3 4 4 2 2 4 4 2 2 3 2 3 1 3 3 1 4
cycle 4:		3 5 3 4 5 2 7	1 3 1 2 2 1 3
		2 4 2 4 4 2 5	1 2 1 2 2 1 3
		2 3 2 3 3 1 1 3 3 1 1 3	1 2 1 2 2 1 1 2 2 1 1 2 1 2 1 2 2 1 3
cycle 5:		1 2 1 2 2 1 3	1 3 1 2 2 1 3
		1 2 1 2 2 1 3	1 2 1 2 2 1 3
		1 2 1 2 2 1 1 2 2 1 1 2	1 2 1 2 2 1 1 2 2 1 1 2 1 2 1 2 2 1 2
		....	....

The similarity between the two hands' foreground rhythmic structure is obvious but the duration of seven quavers in the right hand at the end of cycle 1a as opposed to eight in the left makes for the clearly audible decoupling of the two parts, i.e. the start of the process of 'disorder' or chaos (something reflected in the unsynchronised barlines of the score starting at this point).

This clearly algorithmic (though not computed) thinking lends itself quite naturally to a software implementation.

[show slide and PD]

Musical Example: Désordre implemented in PD by Michael Edwards (based on Lisp algorithm by Tobias Kunze)

Resistance to Algorithmic Composition

There has been considerable resistance to algorithmic composition from all sides, from musicians to the general public.

Hiller's article on his early work for the Scientific American led to much controversy and press attention.

Hostility to his work was such that the Grove Dictionary did not include an article on his work until shortly before his death.

Much of the resistance to algorithmic composition stems from a basic misunderstanding, that somehow the computers compose the music, not the composer.

But it takes a good composer (not necessarily programmer though!) to design musical algorithms that will result in music that captures the imagination.

Main Focus of my work

The integration of electronic and acoustic sound sources and/or instruments

Using electronics as an independent self-sufficient contrapuntal voice instead of a colouring of basically instrumental music

“If you can formalise it, you can programme it”

To further an individual musical and compositional development through computer-programming-enabled “voyages of discovery”

[show slide]

slippery when wet

for Solo Violin, Alto Flute/Piccolo, Clarinet/Bass Clarinet, Horn, Percussion, Violin, Viola, Cello, and Stereo Tape

1999/2000 | 13:00

Translating structural processes for computer music into instrumental music was by no means straightforward.

It took me several years to understand how to do this: need more data for instrumental music than for pure computer music.

Commission from the Österreichisches Ensemble für Neue Musik and the solo violinist Frank Stadler

Whole piece based on the composition method that I first tested out in an earlier piece *pas de poule, pas de pot*

This method then became the basis for my composition program *slippery chicken*

[show 2 slides]

slippery when wet: the first 8 of the 21 rhythmic sequences

slippery when wet: rhythmic mapping structure

1	solo	1	10	19	13	12	5	21	2	solo	2	11	20	15	10	9	8	3	solo	3	12	21	17	14	13	10
8	vln	2	11	20	15	10	9	8	8	vln	3	12	21	17	14	13	10	8	vln	4	13	1	16	19 21	15 17	14
54	vla	3	12	21	17	14	13	10	49	vla	4	13	1	16	19	15	14	43	vla	5	14	3	18	21	11 17	16
	vc	4	13	1	16	19	15	14		vc	5	14	3	18	21	11	16		vc	6	15	7	2	20 21	17	19
	fl	5	14	3	18	21	11	16		fl	6	15	7	2	20	17	19		fl	7	16	5	4	1	2	12
	cl	6	15	7	2	20	17	19		cl	7	16	5	4	1	2	12		cl	8	17	9	6	3	4	18
	hn	7	16	5	4	1	2	12		hn	8	17	9	6	3	4	18		hn	9	18	8	11	7	6	20
	perc	8	17	9	6	3	4	18		perc	9	18 11	8	11	7	6	20		perc	1	10	19	13	12	5	21
4	solo	4	13	1	16	19	15	14	5	solo	5	14	3	18	21	11	16	6	solo	6	15	7	2	20	17	19
7	vln	5	14	3	18	21	11	16	6	vln	6	15	7	2	20	17	19	5	vln	7	16	5	4 13	1	2	12
44	vla	6	15	7	2	20	17	19	34	vla	7 8	16 10	5	4	1	2	12	33	vla	8	17	9	6	3	4	18
	vc	7	16	5	4	1	2	12		vc	8	17 10	9	6	3	4	18		vc	9	18	8	11 13	7 1	6	20
	fl	8	17	4 19	6	3	4	18		fl	9	18	8	11	7	6	20		fl	1	10	19	13	12 1	5	21
	cl	9	18	8 19	11	7	6	20		cl	1	10	19	13	12	5	21		cl	2	11	20	15	10	9	8
	hn	1	10	19	13	12	5	21		hn	2	11	20	15	10	9	8		hn	3	12	21	17	14 1	13	10
	perc	2	11	20	15	10	9	8		perc	3	12	21	17	14	13	10		perc	4	13	1	16	19	15	14
7	solo	7	16	5	4	1	2	12	8	solo	8	17	9	6	3	4	18	9	solo	9	18	8	11	7	6	20
4	vln	8	17	9	6 17	3 14	4	18	3	vln	9	18	8	11	7	6	20	2	vln	5 6	10	19	13	12	5	21
21	vla	9	18	8	11	7	6	20	20	vla	1 3	10	19	13	12	5	21	33	vla	2	11	20	15	10 12	9	8 21
	vc	1	10	19	13 17	12 14	5	21		vc	2 3	11	20 19	15	10	4 5	8		vc	3	12	21	17	14 12	13	10 21
	fl	2	11	20	15	10	9	8		fl	3	12	21	17	14	13	10		fl	4	13	1	16	19	15	14 21
	cl	3	12	21	17	14	13	10		cl	4	13	1 19	16	19	15	14		cl	5 6	14	3	18	21	11	16
	hn	4	13	1	16	19	15	14		hn	5 3	14	3 19	18	21	11	16		hn	6	15	7	2	20	17	19
	perc	5	14	3	18	21	11	16		perc	6	15	7	2	20	17	19		perc	7 6	16	5	4	1	2	12 21
10	solo	1	10	19	13	12	5	21	11	solo	2	11	20	15	10	9	8	12	solo	3	12	21	17	14	13	10
1	vln	2	11	20	15	10	9	8	2	vln	3	12	21	17	14	13	10	3	vln	4	13	1	16 18	19 14	15	14
8	vla	3	12	21	17	14	13	10	23	vla	4 3	13	1	16	19	15	14	31	vla	5	14	3	18	21 14	11	16
	vc	4	13	1	16	19	15	14		vc	5	14	3	18	21	11	16		vc	6	15 14	7 3	2 18	20 14	17	19
	fl	5	14	3	18	21	11	16		fl	6 3	15	7	2	20	17	19		fl	7 6	16	5	4 18	1 14	2	12
	cl	6	15	7	2	20	17	19		cl	7 3	16	5	4	1	2	12		cl	8	17	6 3	6 18	3 14	4	18
	hn	7	16	5	4	1	2	12		hn	8	17	9	6	3	4	18		hn	4 6	18	8	11	7 14	6	20
	perc	8	17	9	6	3	4	18		perc	9 3	18	8	11	7	6	20		perc	1 6	10	19	13	12	5	21

Musical Example: the beginning of slippery when wet

Recording of the first performance, Salzburg, April 2000

Österreichisches Ensemble für Neue Musik Frank Stadler, Solo Violin, Alberto Caprioli, Conductor

At the beginning of the piece, the whole ensemble plays in equal weight with the tape; at the end of the example only solo violin, vibraphone and tape are playing but the presence of the rest of the ensemble is still felt due to the presence of algorithmically processed instrumental samples.

Cheat Sheet

for Electric Guitar, Ensemble und Live Electronics

2007

13-25 Minutes

This commission from the Bregenz Festival and OENM profits from a (finally!) fully-functioning *slippery chicken* programme:

Algorithmically selected pitches are selected and combined with rhythmic sequences to prepare a 'finished' score

This was perhaps the crux of the whole project: successfully combining rhythmic sequences with pitches is by no means trivial

Procedure (not necessarily in order):

[show slide]

slippery chicken: procedure

- Define the instruments: ranges; chord selection function; microtones; any missing notes;
- Define instrument changes for individual players (e.g. flute to piccolo)
- Define the set palette
- Define (possibly algorithmically) the rhythm sequence palette
- Define the rhythm sequence map: sequence onto instruments
- Define the set map
- Define the tempo maps
- Define the set limits: for whole piece and/or instruments

Pitch selection: Each sequence has an arbitrary number of pitch curves

These are numbers representing pitch height; they're selected in rotation

Combined with the given set, instrument, and any set limits, we can map the numbers onto the notes of the set

The instruments successively choose unused notes from the set
we can define a hierarchy to specify who gets the pick of the notes first

Gordon Brown's confession; the BBC website

The piece is inspired by the idea of the score as (musical) censor, but also the self-censoring website of the BBC.

On Tuesday the 3rd of May 2005, two days before the British public voted to re-elect the Blair Labour government, I was browsing the pages of the BBC News website and was amazed to read the following statement by the Labour Minister Gordon Brown; he was referring to the government's 2003 decision to go to war in Iraq:

> "We believed we were making the right decision in the British national economic interest... at the end of the day we wanted the security of Britain and the British national interest to be advanced."

The idea of censorship and especially self-censorship stimulated my musical imagination.

Musical structure; loops

The whole piece is based on looping through 5 bars of four-part counterpoint.

[show slide]

cheat sheet: rhythmic/contrapuntal loop material

Audio loops created in CLM use an arbitrary number of user-designated markers in a sound file to create loops.

The number of repeats of any segment, and its progression to the next segment is determined by a ‘folding in’ structure based on the fibonacci series

[show slide]

cheat sheet: fibonacci transitions for ‘rhythmic fold in’

(fibonacci-transition 70) $\rightarrow$
(0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1
0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1)

Musical Example: DSP loops of saxophone sample

In my efforts to unify the structures of processed sound and instrumental music, I've applied this essentially DSP technique to conventionally notated musical material by dividing 5 bars of 4-part counterpoint into 400 segments (100 per voice: 10 per crotchet, 10 crotchets in 5 2/4 bars).

Each crotchet is divided into 10 loop points with the semiquaver as the shortest unit.

[show slide]

cheat sheet: semiquaver loop points within a single crotchet

```
'((1 4) (1 3) (1 2) (2 4) (2 3) (3 4) (1 1) (2 2) (3 3) (4 4)))
```

Taking the flute part, and comparing the score and the original 4-part counterpoint with the following structure, we can see how this develops.

[show slide]

cheat sheet: rhythmic loop slice mapping for flute

(1 A 1) → (1 4), (1 A 2) → (1 3) etc.

```
((1 A 1) (1 A 1) (1 A 1) (1 A 1) (1 A 2) (1 A 1) (1 A 1) (1 A 2)
 (1 A 1) (1 A 2) (1 A 1) (1 A 2) (1 A 2) (1 A 1) (1 A 2) (1 A 1)
 (1 A 2) (1 A 1) (1 A 2) (1 A 2) (1 A 2) (1 A 1) (1 A 2) (1 A 2)
 (1 A 3) (1 A 2) (1 A 3) (1 A 2) (1 A 3) (1 A 1) (1 A 3) (1 A 3)
 (1 A 2) (1 A 1) (1 A 3) (1 A 3) (1 A 1) (1 A 3) (1 A 2) (1 A 1)
 (1 A 3) (1 A 2) (1 A 3) (1 A 4) (1 A 1) (1 A 3) (1 A 4) (1 A 1)
 (1 A 3) (1 A 4) (1 A 2) (1 A 4) (1 A 1) (1 A 3) (1 A 2) (1 A 4)
 ....
```

It's perhaps easier to understand this process more intuitively by listening to the beginning of another even more recent piece.

for Magda Cordell, if she'll have it (piano and computer) was written in 2007 for Sarah Nicolls and the Huddersfield Festival.

It also uses 4-part note-loops which generate a driving rhythmic character.

The interesting characteristic of note-loops as opposed to audio loops is that you can disconnect rhythm from pitch: a basic repeating rhythmic cell can loop through several harmonic fields, this changing each time.

[show slide]

Musical Example: for Magda Cordell, if she'll have it

for piano and computer

Michael Edwards, 2007

Sarah Nicolls, piano

courtesy BBC

So, finally, some output from slippery chicken using the cheat sheet algorithms

[show slide]

cheat sheet: CMN→EPS output from slippery chicken

cheat sheet: exclamations

There will be no conductor; to help keep everyone together, the players will call out (more or less loudly) the rehearsal letters A-Z as the piece progresses.

Starting halfway through, they also start calling out Brown's statement, cut up and reassembled according to the same looping algorithm that generated the musical sequences.

[show slide]

cheat sheet: exclamations

```
count 71, word-count 0, bar 584, PNO-RH, ‘we’
count 78, word-count 1, bar 584, GTR, ‘we’
count 117, word-count 2, bar 587, PNO-RH, ‘we’
count 135, word-count 3, bar 588, GTR, ‘we’
count 167, word-count 4, bar 590, GTR, ‘believed’
count 179, word-count 5, bar 591, PNO-RH, ‘we’
count 243, word-count 6, bar 595, PNO-RH, ‘we’
count 267, word-count 7, bar 596, GTR, ‘believed’
count 284, word-count 8, bar 598, PNO-RH, ‘we’
count 299, word-count 9, bar 599, VLN, ‘believed’
count 307, word-count 10, bar 600, PNO-RH, ‘we’
count 319, word-count 11, bar 600, GTR, ‘believed’
count 388, word-count 12, bar 605, PNO-RH, ‘believed’
...
count 6114, word-count 555, bar 1162, VLN, ‘believed’
count 6115, word-count 556, bar 1162, VLA, ‘british’
count 6116, word-count 557, bar 1162, VC, ‘we’
count 6120, word-count 558, bar 1163, GTR, ‘believed’
count 6123, word-count 559, bar 1164, PERC, ‘we’
count 6125, word-count 560, bar 1164, PNO-RH, ‘national’
count 6126, word-count 561, bar 1164, PNO-RH, ‘we’
count 6142, word-count 562, bar 1166, PNO-RH, ‘we’
count 6145, word-count 563, bar 1166, VLA, ‘national’
count 6149, word-count 564, bar 1167, PERC, ‘believed’
count 6151, word-count 565, bar 1167, PNO-RH, ‘we’
```

[show slide]

Musical Example: End (from 16:00) of *cheet sheet*

scei = slippery chicken egg

The engagement with *slippery chicken* has become a self-sufficient project.

Contrary to traditional studio work (which can be thought of as sound sculpting) my approach is to generate perhaps hundreds of sound files automatically—file selection becomes the main activity then.

The best results of the algorithms (no editing!) are to be found in the internet as short but complete pieces (http://www.sumtone.com/).

[show slide]

Musical Example: scei V (snow shoes)

Musical Example: scei XII (skin)

Musical Example: scei XVI (charlie)

24/7: freedom fried

for viola d'amore and computer

2004-6 | 14:30

The viola d'amore is a 7-string 7-sympathetic string member of the viol family that had its heyday in the baroque period and subsequently fell out of mainstream use due to the limitations of its tuning system.

Garth Knox is developing a new repertoire for the instrument, in particular in combination with electronics.

Being released soon on a Wergo DVD, in Surround Sound and with a video from Brian O'Reilly (San Francisco).

In addition to the *slippery chicken* structuring methods as described, I use further algorithmic processes in the generation of this piece.

Permutations of the four fingers of the left hand are used as a background to the whole piece.

There are 24 possible permutations of the four fingers:

[show slide]

24/7: freedom fried: 24 possible permutations of the four fingers

```
((1 2 3 4) (2 1 3 4) (1 3 2 4) (3 1 2 4) (2 3 1 4) (3 2 1 4) (1 2 4 3)
 (2 1 4 3) (1 4 2 3) (4 1 2 3) (2 4 1 3) (4 2 1 3) (1 3 4 2) (3 1 4 2)
 (1 4 3 2) (4 1 3 2) (3 4 1 2) (4 3 1 2) (2 3 4 1) (3 2 4 1) (2 4 3 1)
 (4 2 3 1) (3 4 2 1) (4 3 2 1))
```

These 24 permutations are to be played through in any of the many billions (620448401733239439360000) of their possible permutations as fast as possible (unless otherwise notated in the score).

The notes used for the four fingers range over a perfect fourth, reflecting both the natural stretch of fingers 1–4 and the tuning system of the viol.

Though notes (fingers) 1 and 4 are fixed, notes 2 and 3 microtonally interpolate between tetrachords of the phrygian, dorian, and ionian modes:

[show slide]

24/7: freedom fried: tetrachordal interpolation

In choosing the note groups to permutate, the player should wander forwards and backwards along this line.

In the graphic, each of the seven four-note groups are given a number.

The numbers in parentheses represent the groups that may follow the current group, hence after group 1 only 2 can follow; after group 2 may come 3 or 1, depending on whether we are reading forwards or backwards.

The basic pattern is 1 2 3 4 6 4 3 2 1 2 3 4 etc.

Groups 5 and 7 are given in square brackets and represent alternative progressions that should be used to vary the basic pattern.

Thus a constantly varying but basically static microtonal meandering through the various tetrachords is possible, for example:

[show slide]

24/7: freedom fried: tetrachordal meandering

Rules:

```
'((1 ((2)))
  (2 ((3 1)))
  (3 ((4 2)))
  (4 ((6 3 5)))
  (5 ((2 4)))
  (6 ((4 7)))
  (7 ((3))))
```

Result from simply rotating through progressions:

1 2 3 4 6 4 3 2 1 2 3 4 5 2 1 2 3 2 1 2 3 4 6 7 3 2 1 2 3 4 3 2 1 2 3 4 5 4 6 4 3 2 1 2 3 4 5 2 1 2 3 2 1 2 3 4 6 7 3 2 1 2 3 ...

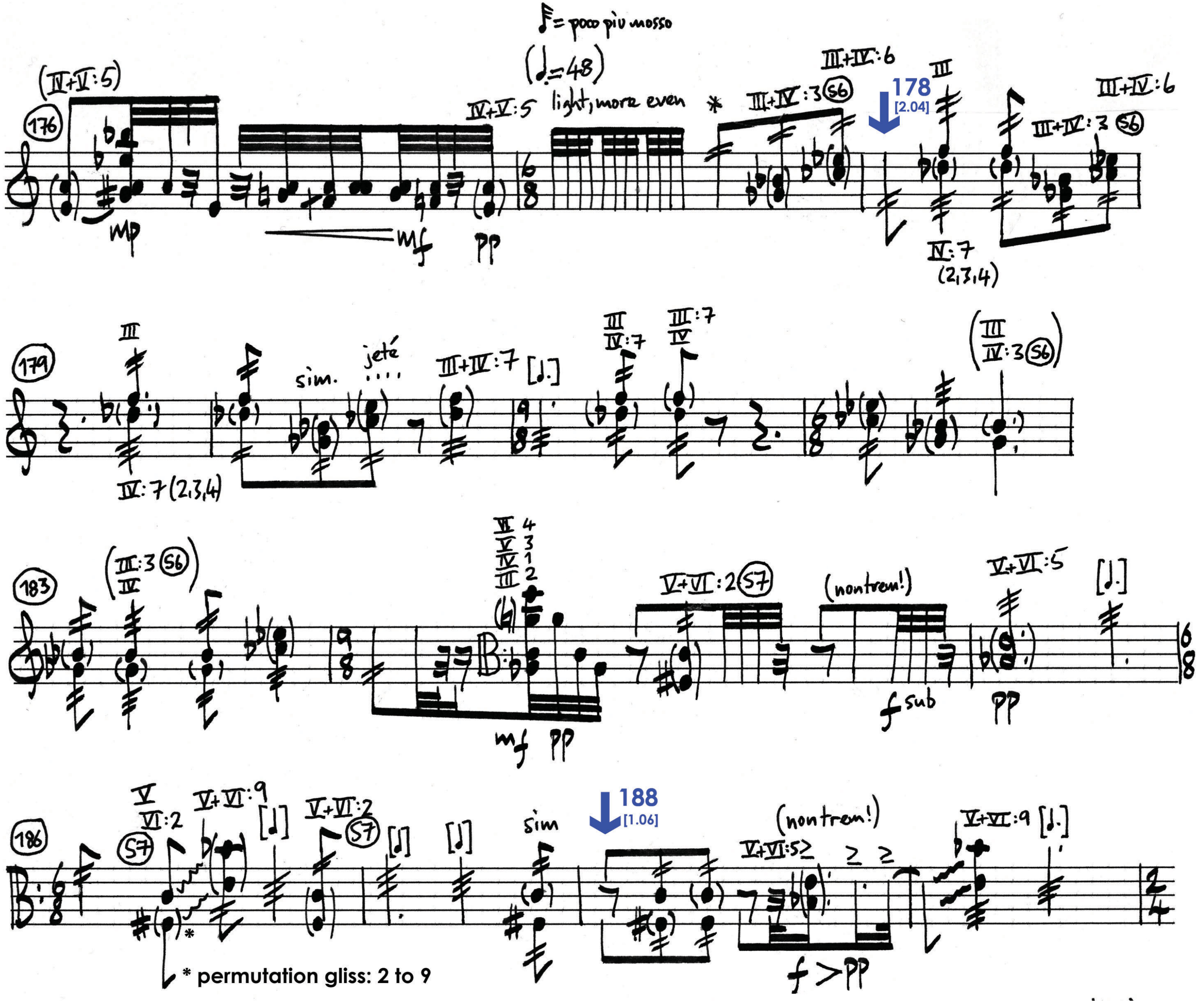
Instead of writing out complex fast passages in the score, a fast permutation is simply indicated by the lowest note(s) in parentheses, and the string(s) and position the note(s) represent.

See, for example, page 9 of the score

[show slide]

By freeing up the notation of the fast permutations we can more easily impose external structures upon them, e.g. jeté bowing, tremolo, glissandi, and microtonal compression/stretches of the basic tetrachord.

24/7: freedom fried score page 9



Musical Example: 24/7: freedom fried: c 5:42 (bar 176)

tramontana

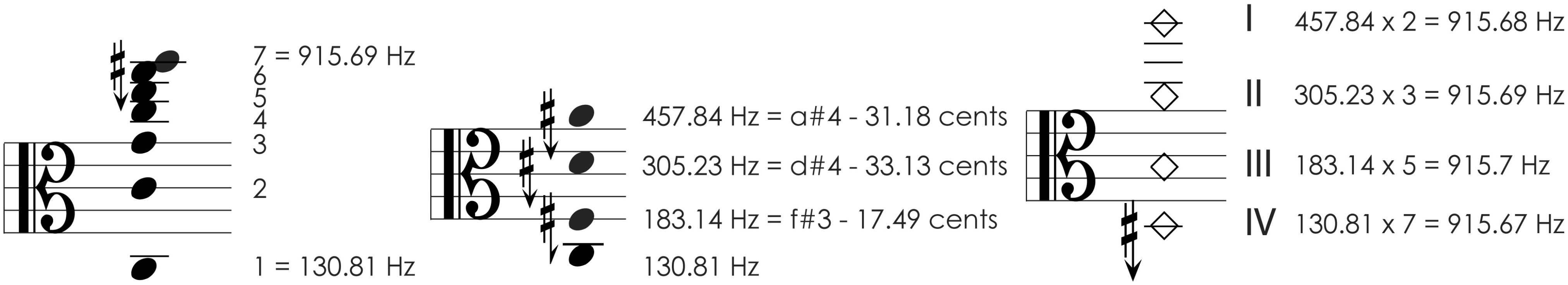
for Viola and Computer

2002-4

scordatura: Strings I, II, and III are tuned to be in-tune with the 7th partial of the C string.

With the following flageolets we get the frequency of this partial (IV=7 III=5 II=3 I=2):

tramontana: scordatura



The score was generated with *slippery chicken* and CMN

The last part of the the piece was generated with L-Systems (Lindenmayer)

tramontana: Lindenmayer Systems

$1 \rightarrow 2\ 3$
$2 \rightarrow 1\ 3$
$3 \rightarrow 2\ 1$

Seed: 2
1 3
2 3 | 2 1
1 3 | 2 1 | 1 3 | 2 3

Self-similarity becomes clear when large result sets are produced:

```
(2 3 2 1 1 3 2 3 2 3 2 1 1 3 2 1 1 3 2 1 1 3 2 3 2 3 2 1 1 3 2 3 2 3 2
 1 1 3 2 3 2 3 2 1 1 3 2 1 1 3 2 1 1 3 2 3 2 3 2 1 1 3 2 1 1 3 2 1 1 3
 2 3 2 3 2 1 1 3 2 1 1 3 2 1 1 3 2 3 2 3 2 1 1 3 2 3 2 3 2 1 1 3 2 3 2
 3 2 1 1 3 2 1 1 3 2 1 1 3 2 3 2 3 2 1 1 3 2 3 2 3 2 1 1 3 2 3 2 3 2 1
 1 3 2 1 1 3 2 1 1 3)
```

Unlike normal L-Systems however I use what I call "Transitioning L-Systems" (where the number returned by the L-Sys is used as lookup into a table whose result depends on a curve)

There is a slow development from:

- More and more fast flageolets on the C and G strings, in comparison to the tremolo chords of previously
- When we arrive at the point where only normal and half flageolets are played, then there is a tendency to have more and more normal and less half flageolets.
- There is also the tendency to have more and more flageolets on the D string.

- Then, more and more of the half flageolets become “real notes”, but these also die out...

The important point here is, that as an instrumental and computer music composer, I have always thought structurally in this way.

Now, with *slippery chicken*, I can programme these ideas, let the music be generated, try things out, change them etc., instead of doing everything on paper and giving up halfway through because it takes too long.

The computer part consists of real-time and non real-time techniques.

The real-time techniqes are:

- Granular-Synthesis (with my Max/MSP External Object)
- Diffusion (with the Strobl-Stiftung's Halophone)
- Live-Recording/Playback/loops

The non real-time techniques are the triggering of pre-prepared sound files.

Musical Example: Transition to the last part of *tramontana* (Live recording of the first performance with Barabara Maurer, Darmstadt, August 2004) (from c. 7:50 c. bar 175)