Is text an adequate tool for modelling musical analysis, composition and performance?

Language and Music as Cognitive Systems
Cambridge (UK), 11-13 May 2007

Bernard Bel

Laboratoire Parole et Langage
CNRS - Université de Provence

The Bol Processor open-source project
sourceforge.net/projects/bolprocessor

Background in electronic & software design for music research

1979 - Shruti Harmonium: A digitally-programmable polyphonic (4-octave) keyboard instrument for the empirical study of microtonal scales.
1980 - Bol Processor (BP1): An expert system mimicking the ability of drum (tabla) players to compose variations on a musical theme or assess their acceptability. (With Jim Kippen)
1981 - Melodic Movement Analyser (MMA): A real-time digital melograph with fundamental pitch extractor for the accurate transcription of melodic movements in Hindustani music.
1985 - Automatic transcription of microtona melody in Hindustani ragas. (With Wim van der Meer)
1989 - Question-Answer Validated Analytic Inference Device (QAVAID): A machine-learning environment for inferring grammars from sets of examples provided by drum (tabla) experts.
1990 - BP2: A new implementation of Bol Processor for music composition in the MIDI and Csound environments. (With Kumar Subramanian)
2006 - Bol Processor is open-sourced.
2007 - Bol Processor is ported to MacOS X by Anthony Kozar.

The Bol Processor project originated in 1980 as a word processor facilitating the transcription of quasi-onomatopoeic syllables used as an oral notation system for Indian drumming. It grew up as an ***expert system*** mimicking the ability to compose variations on a musical theme or assess their acceptability. ***Pattern grammars*** (a subset of type-2 formal grammars) proved appropriate for modelling the musical system under study. In 1989 a numeric-symbolic learning device was developed for inferring grammars from examples.

The next implementation (BP2) addressed the issue of music composition in the MIDI and Csound environments of electronic music. The new challenge was to deal with superimposed sequences of events (polyphony) within the framework of text-oriented rewriting systems. This was achieved by means of ***polymetric representation***. Minimal descriptions of polyphonic/polyrhythmic structures may be 'expanded' by the system to produce arbitrarily complex musical scores. This 'sonemic description' is an arrangement of musical events ('sound-objects' and 'time-objects') along symbolic time measured with integer ratios.

Producing the actual performance requires additional information which the Bol Processor encapsulates in metrical/topological properties of 'sound-object prototypes'. The ***time-setting algorithm*** modifies sound-objects taking into account physical timing and their adjacent sound-objects, much in a similar way human speakers modify the articulatory parameters of speech sounds with respect to the speaking rate and influence of adjacent segments (coarticulation).

In a classical electronic music environment, musical works are represented on scores analogous to Western staff notation. Thus, human and machines rely on 'rules of interpretation' for the actual performance. The polymetric representation makes it possible to produce sophisticated time structures from information comprehensively ***imbedded in compositional rules***, thereby maintaining the consistency of interpretation. This is a major discovery for computer music as 'natural' phrasing is no longer achieved by randomness. Similar frameworks could be applied to the alignment of tonal symbols in speech synthesis.

Pattern grammars

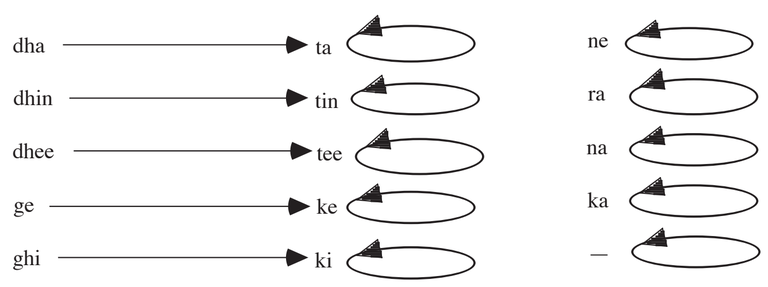
Theme and variations (qa'ida)
Lucknow style of tabla playing

dhatidhage	nadhatrkt	dhatidhage	dheenagena
dhatidhage	nadhatrkt	dhatidhage	teenakena
tatitake	natatrkt	tatitake	teenakena
dhatidhage	nadhatrkt	dhatidhage	dheenagena

Jim Kippen & Bernard Bel (1992). Modelling Music with Grammars: Formal Language Representation in the Bol Processor. In A. Marsden & A. Pople (eds.): Computer Representations and Models in Music. London, Academic Press: 207-38.
http://halshs.archives-ouvertes.fr/halshs-00004506

dhatidhage	nadhatrkt	dhatidhage	dheenagena
dhatrktdha	*tidhagena*	**dhatidhage**	**teenakena**
tatitake	natatrkt	tatitake	teenakena
dhatrktdha	*tidhagena*	**dhatidhage**	**dheenagena**
dhatidhatr	*ktdhatidha*	*trktdhage*	*nadhagena*
dhatidhage	nadhatrkt	**dhatidhage**	**teenakena**
tatitatr	*kttatita*	*trkttake*	*natakena*
dhatidhage	nadhatrkt	**dhatidhage**	**dheenagena**
dhagenadha	*trktdhage*	*nadhatrkt*	*dhatrktdha*
tidha-dha	*tidhagena*	**dhatidhage**	**teenakena**
takenata	*trkttake*	*natatrkt*	*tatrktta*
tidha-dha	*tidhagena*	**dhatidhage**	**dheenagena**

Alphabet and 'khuli/band' homomorphism



Pattern syntax

(=dhatidhage	nadhatrkt	dhatidhage	dheenagena)
(=dhatrktdha	tidhagena)	(=dhatidhage	teenakena)
*(:tatitake	natatrkt	tatitake	teenakena)
(:dhatrktdha	tidhagena)	(=dhatidhage	dheenagena)

Pattern (transformational) grammar

GRAM#1 (subgrammar)

```
<10>  S     —> (=A16) (=V8) (=A'8) *(:A16) (:V8) (=A8)
<60>  S     —> (=V16) (=A'16) *(:V16) (=A16)
<30>  S     —> (=V24) (=A'8) *(:V24) (=A8)
      A16   —> dhatidhagenadhatrktdhatidhagedheenagena
      A'16  —> dhatidhagenadhatrktdhatidhageteenakena
      A8    —> dhatidhagedheenagena
      A'8   —> dhatidhageteenakena
```

weights

GRAM#2

```
[1]  V8   —> V V V V V V V V
[2]  V16  —> V V V V V V V V V V V V V V V V
[3]  V24  —> V V V V V V V V V V V V V V V V V V V V V V V V
```

GRAM#3

```
[1]  V     —> V1
[2]  V V   —> trkt
[3]  V V V —> V3
```

declarative

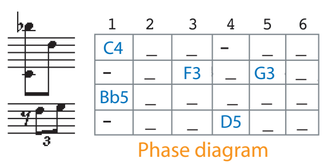
GRAM#4 (context-sensitive)

```
[1]          V3      —> dhagena
[2]  LEFT    #dha V1 —> #dha dha
[3]  LEFT    #- V1   —> #- -
[4]  LEFT    dha V1 #tr —> dha ti #tr
[5]  LEFT    na V1 #tr  —> na ti #tr
[6]  LEFT    - V1 #tr   —> - ti #tr
[7]          (= V1 #tr  —> (= ti #tr
```

procedural

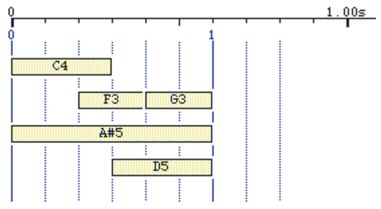
Polymetric structures

The 'comma' notation



1	2	3	4	5	6
C4	_	_	-	_	_
-	_	F3	_	G3	_
Bb5	_	_	_	_	_
-	_	_	D5	_	_

Phase diagram



{ 1 , C4 - , - F3 G3 , Bb5 , - D5 }

relative duration — 4 fields

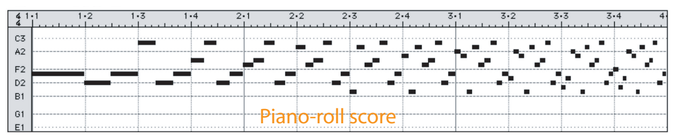
Sound-object score (*mm = 100*)

This polymetric expression is expanded as:

tempo(6) {C4 _ - _ _ , - _ F3_ G3_ , Bb5_ _ _ _ _ , - _ _ D5_ _}

The 'period' notation

S —> A B C D E F G H I J K L
A —> E2 •
B —> D2 A
C —> B2 B
D —> G2 C
E —> F#2 D
F —> A#2 E
G —> C2 F
H —> G#2 G
I —> A2 H
J —> D#2 I
K —> C#2 J
L —> F2 K

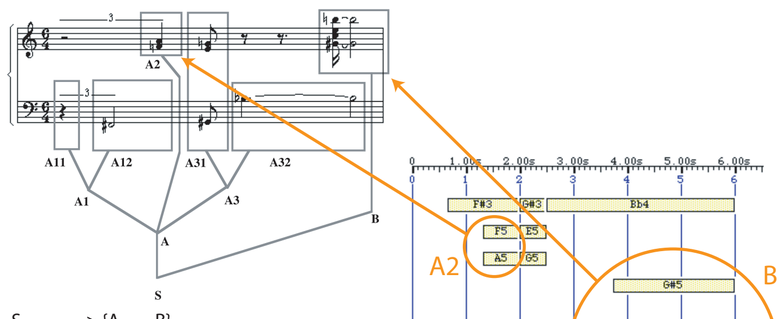


BP2 score:

E2 • D2 E2 • B2 D2 E2 • G2 B2 D2 E2 • F#2 G2 B2 D2 E2 • A#2 F#2 G2 B2 D2 E2 • C2 A#2 F#2 G2 B2 D2 E2 • G#2 C2 A#2 F#2 G2 B2 D2 E2 • A2 G#2 C2 A#2 F#2 G2 B2 D2 E2 • D#2 A2 G#2 C2 A#2 F#2 G2 B2 D2 E2 • C#2 D#2 A2 G#2 C2 A#2 F#2 G2 B2 D2 E2 • F2 C#2 D#2 A2 G#2 C2 A#2 F#2 G2 B2 D2 E2 •

No tempo() tool is required for speeding up!

Undetermined rests, concatenation



S —> {A , … B}
A —> {2, A1, … A2} {4, A3}
A1 —> A11 A12
A3 —> A31 A32
A11 —> -
A12 —> {2, F#3}
A2 —> {F5, A5}
A31 —> {1/2, G#3, E5, G5}
A32 —> {3/2, Bb4&} {2, &Bb4}
B —> {1/4, G#5&, C6, E6, B6&} {2, &G#5, &B6}

Undetermined rest

Concatenation

Sound-object score (*mm = 60*)

Serial tools, recursive grammars

S —> _volume(90) _vel(70) _rndvel(10) M1 M1 - M1 M1 {_rotate(1) M1 M2 M3 M4 M5 M5 M5} - {_transpose(11) M1 M2 M3 M4 M5} - {{12, M1a} {M2a M5a}} _transpose(2) {_rotate(1) M1 M2 M3 M3 M3 M4 M5} - {5, _keyxpand(C4,-1) M1} {5, _transpose(11) _keyxpand(C4,-1) M3} - {_transpose(-1) M1 M2 M3 M4 M5}
M1 —> {C5, - B3, F3, _chan(9) {1/2 C4 B3 - F4 -}}
M2 —> {_transpose(11) M1}
M3 —> {_transpose(5) M1 • M2}
M4 —> {_transpose(-11) M1 • M2 M3}
M5 —> {_transpose(11) M1 - • M2 - M3 M4}

Recursive rule

<1-1> M1a —> {1, C2, - B3, - F3, - C4, - - B4, {_chan(9) - - - B3 F3 -}} _transpose(5) {_retro M1a} **_repeat(6)**
<1-1> M2a —> _transpose(-1) {1, C2, - B3, - F3, - C4, - - B4, {_chan(9) - - - B3 F3 -}} _transpose(7) {_rotate(1) M2a} **_repeat(5)**
M5a —> _retro {_transpose(11) M1 - - • M2 - M3 - M4}

M1a —> nil
M2a —> nil

'Shape' grammar

S —> Frase1 1 Frase1 Frase2 1/2 {_retro _transpose(12) Frase1} {_rotate(2) _transpose(1) Frase2} {_transpose(-13) Frase3} 1/4 {_transpose(-1) Frase1} Frase4 {_retro _transpose(-1) Frase1} {_rotate(3) _transpose(-11) Frase4} 1 Frase1 {_keyxpand(C4,-1) Frase1} 1/2 {2, _keyxpand(B3,-1) _vel(40) M19, M24} 1/2 {2, _keyxpand(A#3,-1) _vel(40) M19, M24} 1/4 Frase5 {_keyxpand(B3,-1) _transpose(-1) Frase5} {_keyxpand(C4,-1) Frase1} 1/2 {2, _keyxpand(B3,-1) _vel(40) M19, M24} 1/2 {2, _keyxpand(A#3,-1) _vel(40) M19, M24} - Frase6 - Frase6 F1 B3 - {_vel(50) F1 B3} - - {6, _vel(40) F1}

Frase1 —> _legato(100) {_velcont _vel(50) M1 M2 M3 M4 _vel(60) M5 M6 _vel(50)}
Frase2 —> _legato(100) {_velcont _vel(50) _transpose(2) M7 M8 _vel(60) M9 M10 M11 M12 _vel(50)}
Frase3 —> _legato(100) {_velcont _vel(50) _transpose(2) M13 M14 M15 _vel(60) M16 M17 M18 _vel(50)}
Frase4 —> _legato(100) {_velcont _vel(50) _transpose(2) M19 M20 M21 M22 _vel(60) M23 M24 _vel(50)}
Frase5 —> _legato(100) {_velcont _vel(50) _transpose(2) M25 _vel(60) M26 M27 M28 M29 M30 _vel(50)}
Frase6 —> _legato(100) {_velcont _vel(50) _transpose(2) M31 _vel(60) M32 M33 M34 M35 M36 M37 M38 _vel(50)}
M1 —> {5, C3 F#3
M2 —> {3, _transpose(13) C3 F#3
M3 —> {3, _transpose(1) C3 F#3
M4 —> F2 B3}
M5 —> F2 B3}
M6 —> F1 B3}
M7 —> {5, _vel(60) C3 F#3
M8 —> {3, _transpose(13) C3 F#3
M9 —> {3, _transpose(1) C3 {M8 M10} F#3
M10 —> F2 B3}
M11 —> F2 B3}
M12 —> F1 B3}
M13 —> {5, _vel(60) C3 F#3
M14 —> {5, _transpose(13) C3 {_transpose(1) M13 M17} F#3
M15 —> {5, _transpose(1) C3 {M14 M16} F#3
M16 —> F2 B3}
M17 —> F2 B3}
M18 —> F4 B3} … *etc. (49 rules)*

This piece is constructed on the idea of self-imbedding
{5, a b {3, a b {3, a b c d} c d} c d}

(Harm Visser "Waves", 9 Sept. 1998)

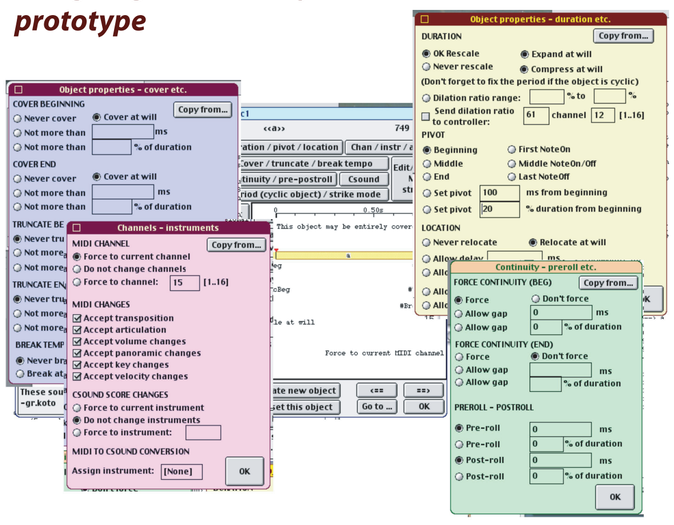
'Natural' phrasing is not achieved by using tempo() or rand() tools, but by the complexity of the expanded polymetric structure. The relative durations of musical material on any scale from individual notes to phrases, sections, or an entire work may be specified by integer ratios allowing for the careful shaping of the entire time structure and its performance by a computer. In this way, 'rules of interpretation' are imbedded in the compositional structure.

If I belong to a tradition, it is a tradition that makes the masterpiece tell the performer what to do, and not the performer telling the piece what it should be like, or the composer what he ought to have composed.

Alfred Brendel

The time-setting of sound-objects

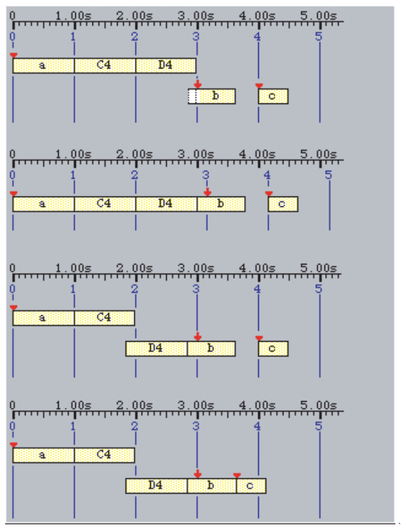
Designing a sound-object prototype



Every sound-object is assigned metrical and topological properties.

The time-setting algorithm

This algorithm is used to solve the set of constraints imposed by metrical-topological properties of sound-objects.



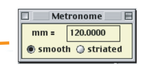
1) Truncate the beginning of <b>

2) 'Break tempo'

3) Move back D4

4) Move back D4, then apply continuity constraint between the end of <b> and the beginning of <c>

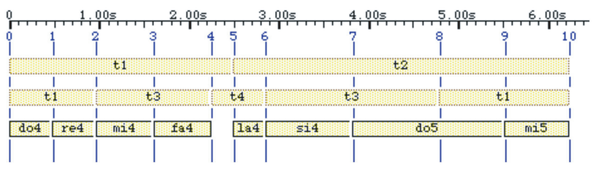
Time-objects, time patterns



_mm(120.0000) **_smooth**

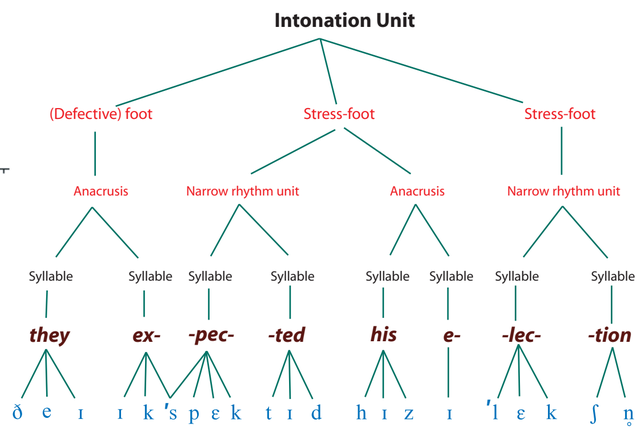
S —> {10 , **t1 t2** , Part1 Part2}
Part1 —> {**t1 t3 t4** , **do4 re4 mi4 fa4 - la4**}
Part2 —> {**t3 t1** , **si4 do5 _ mi5**}

TIMEPATTERNS:
t1 = 1/1 **t2** = 3/2 **t3** = 4/3 **t4** = 1/2



Bel, B. (2005). Two algorithms for the instantiation of structures of musical objects.
http://halshs.archives-ouvertes.fr/halshs-00004504

Speech prosody and computational musicology: related concepts



stress-feet <—> time-objects
rhythm units (anacrusis etc.) <—> time-objects
other suprasegmental units (syllables etc.) <—> variables
phonemes <—> sound-object prototypes
phones <—> sound-objects