MUSICAL BEHAVIORS:

LAYERED COMPOSITIONAL ALGORITHMS AS PLUGINS FOR THE
TRANSFORMATION ENGINE
ALGORITHMIC COMPOSITION IN THE CONTEXT OF “PRACTICAL CREATIVITY”

• “Practical Creativity” - the hands-on creation of any musical work, but especially for:
  - accompaniment of a film, video or play;
  - music that follows a narrative structure;
  - live instrumental performance.

• there are many situations where it is desirable to use algorithmic processes ranging in complexity from “raise pitch logarithmically one octave over 8 measures”, to an astro-physics simulation (e.g. planetary motions) or fractal structure, mathematical process (Euclidean rhythms).

• BUT these algorithmic processes must be further modified, custom shaped, to fit the narrative, or accommodate the limitations of acoustic instruments.
SOFTWARE SUPPORT FOR ALGORITHMIC COMPOSITION

- TWO PROBLEMS:

- TOO LITTLE: Commercial composition software (Cubase, Logic, etc.) dominated by simulation of multi-track recorder for > 30 years. Only timeline-based modification is supported. Little or no support for algorithmic composition.

- TOO MUCH: Experimental composition software (e.g. Max-MSP, Processing, PD, etc.) sometimes supports algorithmic approach, but only with global scope. Algorithm ‘takes over’ all musical processes, prohibiting custom shaping, a requirement for “practical creativity”.
VISUAL ANIMATION SOFTWARE (ADOBE AFTER EFFECTS)

- e.g. Adobe After Effects, Autodesk Maya, Apple Motion

- incorporate BOTH modes of control:
  - algorithmic processes as plugins - e.g. particle system
  - timeline-based automation (for “hand shaping”)

image: https://www.youtube.com/watch?v=1z9ti_dGpQE
VISUAL ANIMATION SOFTWARE
(QUARTZ COMPOSER)

algorithmic processes can control other processes (e.g. frequency modulated LFO controls particle count)

realtime interactive input; realtime display of output
PRIOR WORK

- Apple Logic Pro X - includes a MIDI Scripting language
  - seems to be limited to echo effects and arpeggiators (?)
- Cakewalk - Cakewalk Application Language
  - non-realtime only (?)
SOFTWARE REQUIREMENTS FOR MUSICAL BEHAVIORS (NAME FROM APPLE MOTION SOFTWARE)

**DESIDERATA:**

**MUSICAL BEHAVIORS MUST:**

1. co-exist with timeline-based automation
2. be selectable (i.e. plugin format)
3. combine correctly with one another (i.e. be layerable)
4. have clearly defined scope (i.e. limited to a specific time-segment and instrument)
5. be interactive in realtime, with realtime audio output and graphic display
6. provide full-featured programming language structures (IF-THEN, LOOPs, etc) and access to sequencer data

**BEHAVIORS SHOULD:**

7. allow programmable interconnection between one another
THE TRANSFORMATION ENGINE

- personal composition software
- oriented to traditional music composition (i.e. themes, motivic development, harmonic structure)
- MIDI-based, with extensions for Open Sound Control (OSC), OpenGL & MusicXML
BEHAVIORS IN THE TRANSFORMATION ENGINE

1. Timeline automation combines with algorithmic process
2. Selectable (2), layered (3) “algorithmic” processes
3. (4) Scope of algorithm is limited to theme & track
4. Realtime display of automation + algorithmic output
FULL PROGRAMMING LANGUAGE SUPPORT

- (6) plugins are programmed in the host language, VFXForth, compiled from text source-code
DEMONSTRATION: LFO BEHAVIOR

‘theme’ consists of two eighth notes

NO BEHAVIORS - theme repeats verbatim, with harmonic changes (doubled speed is due to timeline automation settings)
DEMONSTRATION: LFO BEHAVIOR

add LFO BEHAVIOR - Shape: Sine Wave 100%;
Range - +- 12 semitones;
Wavelength: 960 ticks = one measure
Phase: 0 degrees

adjust Phase: 110 degrees
DEMONSTRATION: LFO BEHAVIOR

adjust Damping: -2.5%
DEMONSTRATION: WIND SIMULATION BEHAVIOR

- algorithm derived from Andy Farnell, *Designing Sound* (MIT Press), pp.475 ff
- algorithm originally written in *PureData*
- three components of wind are: *Howl*, *Gust* and *Squall*. Each component has separate amplitude control.
‘theme’ consists of four sixteenth notes

NO BEHAVIORS - theme repeats verbatim, with harmonic changes

LINE SEGMENT BEHAVIOR - adds a one octave drop over phrase
WIND SIMULATION - gives new contour, with gust and squall detail

SIMPLE PATTERN BEHAVIOR - gives variety of rhythmic pace
DEMONSTRATION: WIND SIMULATION BEHAVIOR

graphic display, including MusicXML notation

MusicXML output converted to CMN via Sibelius
CONCLUSION

“Musical Behaviors” in The Transformation Engine provide a software composition environment suitable for “practical creativity” by fulfilling the desired characteristics:

- ✓ co-exist with timeline-based automation
- ✓ individually selectable
- ✓ layer-able
- ✓ have clearly defined scope (i.e. limited to a specific time-segment and instrument)
- ✓ be interactive in realtime, with realtime audio output and graphic display
- ✓ provide full-featured programming language structures (IF-THEN, LOOPs, etc) and access to sequencer data
- ✓ allow programmable interconnection between one another (not demonstrated)