Scores That Run: Graphic Notation with Embedded Performance Semantics

Dr Rob Canning
Independent Researcher
rc@kiben.net

ABSTRACT

This paper presents an approach to digital graphic notation in which performance semantics are embedded directly into the visual surface of the score. Working in standard SVG and authored entirely in Inkscape, the score is composed as a graphic–semantic document: visual elements carry lightweight cue structures encoded in their identifiers, and these cues are executed in real time by a browser-based runtime. The score therefore functions simultaneously as image, temporal structure, and performative interface, without reliance on symbolic engraving or external playback systems.
The framework supports hybrid formal topologies, including continuous scrolling trajectories, page-based local environments, and patterned navigation between sectional states. Animated motion fields provide shared gestural resources for ensemble coordination and may optionally drive live electronic processes, enabling a unified grammar of acoustic and electronic gesture. All cue semantics—structural, temporal, gestural, textual, and media-based—are authored within the same executable layer as the notation, so behaviour and interpretation arise from a single surface.
Because the system is based entirely on open web standards, it enables a direct draw-and-perform workflow accessible to composers and performers without specialised technical infrastructure.

1. INTRODUCTION

Graphic notation continues to play a significant role in contemporary composition, offering visual structures that invite interpretation, gesture, spatial metaphor, and temporal fluidity. However, the transition from the drawn score to a performable one often requires additional mediation: rehearsal explanation, auxiliary timing systems, proprietary playback environments, or manual synchronisation cues. In many cases, the notation and its performance logic remain separated, with behaviour residing outside the score in software patches, conductor instruction, or group negotiation.

This paper presents an approach in which the score itself encodes its own performance semantics, allowing the visual domain and the temporal domain to coexist within a single representational layer. The system described here employs SVG as the underlying score format, authored in the WYSIWYG environment of Inkscape, with performance behaviour expressed through lightweight cue microsyntax embedded into standard element identifiers. When loaded in the browser, these cues are parsed and enacted in real time, making the score executable: it animates, signals entrances, triggers durations, and guides the unfolding of form without requiring external software environments or scripting languages.

A browser-based runtime interprets these embedded semantics to support timeline-oriented, page-based, and sectional score geometries, enabling the composer to shape temporal structure according to the demands of the material rather than a predetermined display model. When used in ensemble contexts, device synchronisation is handled through a coordinating process that maintains a shared temporal state while leaving visual rendering and interpretation local to each performer's browser. The result is a direct authoring-to-performance workflow in which visual composition, temporal behaviour, and performance execution remain tightly integrated.

This work extends the author's earlier investigations into web-based score distribution and real-time networked notation. The NodeScore system [1] demonstrated how web standards—HTML5, SVG, and WebSockets—could provide infrastructure for networked score coordination, while subsequent work on parallax scrolling interfaces [2] explored continuous timeline navigation for composed improvisation. Whereas the earlier Oscilla paper [3] introduced the system as a synchronized performance environment, the present paper addresses the score as a graphic–semantic notation system, authored directly in SVG and enacted in performance through embedded cue semantics and animated gesture fields.

2. RELATED WORK

The question of what constitutes a musical score in digital environments has been explored across systems that treat notation variously as visual composition, reactive interface, control structure, or networked performance space. The approach developed here aligns with work that treats the score as a graphic surface endowed with temporal and behavioural semantics, while differing in that these se-

mantics are authored visually and embedded directly into the document itself.

2.1 Graphic Composition: Drawing as Score

A foundational precedent appears in Xenakis's UPIC system [4], where drawn lines, shapes, and trajectories define musical structure and are rendered into sound via synthesis. UPIC proposed that graphic form could itself be compositional material rather than a representation of something composed elsewhere.

Oscilla shares this premise—that musical thought may proceed visually—but diverges in its treatment of performance. In UPIC, drawings are translated into an independent sonic artefact. In Oscilla, performance semantics remain attached to the drawing: the score continues to operate as a score, read, interpreted, and enacted by performers, even when it also animates or influences electronics.

2.2 Dynamic and Message-Driven Notation

INScore [5] introduces a model for scores modifiable at runtime. Visual elements can be repositioned, transformed, and updated through OSC messaging, making the score a live visual environment responsive to external control.

Oscilla differs less in capability than in authorship model. INScore relies on script-based construction and continuous message control; Oscilla embeds performance semantics into the SVG document itself. Authoring remains entirely visual: behaviour is declared locally in the drawing rather than orchestrated from an external control layer.

Systems such as MaxScore [6] and the Bach library [7] situate notation within Max/MSP's dataflow environment, enabling close coupling between visual material and algorithmic or DSP processes. These frameworks have shaped much practice in digital notation, but they position notation as an outcome of patch logic rather than as the primary compositional site. Oscilla takes the opposite stance: the visual page remains the principal site of composition.

2.3 Trajectory-Based Control: IanniX

Building on conceptual ground opened by UPIC, IanniX [8] extends drawn form into a real-time control environment. Trajectories, triggers, and cursors are evaluated at runtime, generating OSC and MIDI messages that drive external sound engines.

Like Oscilla, IanniX treats the graphic layer as active and performative. The divergence lies in orientation: IanniX positions the score primarily as a controller for electronic systems, whereas Oscilla treats the SVG document as a performable score in its own right. Animated motion fields, navigation cues, and temporal structures are intended to be read by musicians at the same time as they may produce control data. Notation and execution coincide rather than being divided into controller and instrument.

2.4 Distributed Performance Environments

Hajdu's Quintet.net [9], launched in 2000, established an early model for networked notation under conductor control, connecting performers across local and wide-area networks through a shared symbolic score environment. It demonstrated that distributed coordination could be managed through notation interfaces rather than audio synchronisation alone.

More recently, the Decibel ScorePlayer [10] developed by Hope, Vickery, and Wyatt provides network-synchronised scrolling of graphic notation across iPads. The system supports animated scores, embedded audio, OSC communication with external processes, and a "canvas layer" for real-time drawing.

Oscilla shares Decibel's commitment to graphic notation and distributed synchronisation, but differs in authoring model and platform. In Oscilla, cue semantics are embedded directly into SVG identifiers and authored in a general-purpose vector editor (Inkscape); no dedicated authoring tool or platform-specific app is required. The score is a standard web document interpreted by any modern browser, and behaviour arises from the same surface as the notation rather than from a separate runtime layer.

2.5 Programmatic Notation: neoscore

neoscore [11], developed by Yoon under the DigiScore project, takes a code-first approach to graphic notation. As a Python library, it treats scores as programmatically constructed shapes and text, supporting generative score-making, animation, and live-coding.

Where neoscore requires programming, Oscilla requires only drawing. Both reject the assumptions of traditional engraving software, but they occupy opposite ends of the authoring spectrum: neoscore suits composers comfortable with code; Oscilla suits those who think through visual arrangement and spatial relationship.

2.6 Positioning the Present Work

Across these systems, multiple models emerge for how notation might participate in performance: as graphical composition, as dynamic interface, as controller surface, as networked document.

Vear's taxonomy of the digital score [12] identifies defining features including the "animated score," "technological conductor," and "system as score." Oscilla spans several of these categories. It is an animated score in which motion fields and temporal cues unfold visually; it embeds conductor-like temporal semantics that coordinate ensemble behaviour without external control; and the browser runtime that interprets the SVG is inseparable from the score itself—the system *is* the score.

The contribution of Oscilla is to treat the score as a single executable surface in which visual composition, temporal structure, performer interpretation, and optional electronic control are co-authored inside the same drawing. Cues, navigation, and animation are not layered on afterwards or driven from external scripts; they are embedded into the graphic objects themselves and enacted at performance time.

3. SYSTEM OVERVIEW

A preliminary account of OscillaScore as a synchronized scrolling display environment appears in Canning [3], where the emphasis is on network coordination and distributed performance. The present work addresses the score itself as a composed graphic–semantic document, focusing on authoring workflow, cue structure, hybrid formal topology, and gesture semantics.

3.1 Authoring in Inkscape

Scores are created directly in Inkscape, using line, shape, spatial organisation, typographic emphasis, and contour as primary compositional material. Conventional notation can be included when useful, but the system does not assume symbolic engraving as the default. Visual form and spatial relationships are treated as musically meaningful structures, not illustrations of another underlying representation.

Performance semantics are introduced by assigning identifiers to SVG elements through Inkscape's XML editor (Ctrl+Shift+X), inserting the cue microsyntax directly into each element's id attribute.[1] The composer remains inside a visual workflow: notation remains graphic, while semantics are introduced through small, local adjustments rather than scripting or external configuration. The score remains a readable, editable drawing that directly encodes its own performance behaviour.

1 A dedicated Inkscape extension for authoring cue semantics is in beta testing.

3.2 Cue Semantics Embedded in SVG Identifiers

Performance actions and structural transitions are defined through a compact microsyntax embedded in SVG id attributes. For example:

```
nav(mode(scroll@F), speed:0.8)
pause(10, count:true)
```

These identifiers are parsed at performance time to form an internal representation of temporal flow, navigation, and interpretive events. The semantics remain co-located with the visual elements they govern: the score's behaviour is part of its surface, not an auxiliary control patch.

3.3 Runtime Interpretation

The performance runtime is built on standard web technologies: JavaScript, the SVG DOM, and requestAnimationFrame timing. When a score is loaded, the runtime reads the SVG structure, extracts cues, and constructs either a continuously traversed timeline or a network of navigable score regions, depending on layout mode.

In ensemble settings, navigation and temporal state are synchronised across devices, allowing the score to be performed collaboratively without a dedicated playback master or conductor. The SVG file itself remains unchanged; the runtime enacts what the document declares.

3.4 Score Topology, Animated Gesture, and Media

Oscilla supports a hybrid approach to form in which scroll-based continuity and page-based locality coexist. Scroll mode emphasises large-scale directional movement, allowing the score to unfold horizontally through time. Page mode defines spatially discrete environments in which performers may remain, explore, or develop material. Transitions between these states are authored directly in the score, making the articulation of form visible, repeatable, and performable.

Animated transformations within the score—motion along paths, rotation, scaling sequences, orbital trajectories—are authored visually and evaluated at runtime. These motion fields may be read interpretively by performers, or they may expose continuous control data to live electronics. For example:

```
o2p(path:p03, dur:60, osc:1, oscaddr:"/
o2p/pvoc/freq")
```

When osc:1, the motion produces streams of OSC data suitable for environments such as SuperCollider. Acoustic and electronic layers can therefore share a single, visually authored grammar of gesture.

Audio and video may also be embedded directly within the score and treated as composable performance material rather than external accompaniment:

```
video(file:"reflection-III.mp4")
audio(file:"breath.wav", loop:2)
```

Media may be activated automatically at structural points or triggered by performers. In both cases, it participates in the same runtime semantics as navigation and gesture, entering the performance environment from within the document rather than from an external playback system.

These mechanisms position the score as a place where behaviour is composed. The following section considers the notational consequences of treating cues as part of the score's surface.

4. CUE SEMANTICS AS TEMPORAL AND PERFORMATIVE NOTATION

In Oscilla, cue semantics are treated as notation rather than metadata or auxiliary control logic. A cue contributes simultaneously to how a passage is read and to how it unfolds in real time. The same visual marks that shape interpretation also activate behaviour in the runtime, so that temporal structure, orientation, and execution coexist on the surface of the score.

4.1 Cues as Encoded Temporal Structure

Certain cues articulate form, duration, or transition as explicit temporal notation. A cue such as:

```
metro(bpm:90, beats:3,hold:10)
```

brings up a visual metronome for ten seconds—a performer-facing temporal aid authored directly into the score. Such cues occupy a role analogous to rehearsal letters, fermatas, or tempo markings in traditional notation, but differ in that the temporal structure is not merely indicated—it is executed.

Temporal structure is therefore authored directly in the drawing: the score does not merely suggest time; it performs it.

4.2 Cues as Orientation and Perceptual Condition

Not all cues impose structural change; some shape attention, environment, and perceptual stance without prescribing specific sonic outcomes.

Text and media cues work this way. A text cue does not simply place static words on the page—it references external material and reveals it as a temporally evolving object: lines may appear sequentially, characters may surface gradually, fragments may emerge in shifting order. Similarly, embedded audio or video is treated as score-native material arising from the same semantic layer as notation. A video may function as a moving field of attention; an audio cue may establish resonance, operate as a slowly evolving texture, or be delivered privately to an earpiece to enable close imitation.

Temporal orientation can also be composed without specifying musical content. A countdown pause holds the score in place while sound continues. Alternatively, timing may be shaped by altering the traversal rate rather than stopping it; a cue such as

```
speed(value:1.4, dur:6, ease:inOutQuad)
```

functions as an embedded accelerando, changing how quickly material arrives and how coordination unfolds, without dictating the material itself. These cues treat orientation—textual, medial, temporal—as compositional material in its own right.

4.3 Interpretation & Synchronisation in Performance

During performance, cues are evaluated relative to the current scrolling or page-based state of the score. When a cue is reached, its semantics are executed locally in each performer's browser, while shared temporal position remains synchronised across devices.

Structural coherence is maintained without eliminating interpretive agency. The score functions simultaneously as a graphic notation shaping space and gesture, a temporal specification articulating when form unfolds, an interpretive environment orienting perception, and an executable surface enacted directly in performance.

No distinction is made between the score that is read and the system that is performed: they are the same object.

5. SYSTEM IN PRACTICE

Oscilla supports both fixed-form composition and situations where material unfolds responsively in performance. Rather than prescribing a particular notational style, the framework enables structural, interpretive, and gestural layers to intersect within a single score.

5.1 Structure and Attention

Oscilla shapes performer attention through two complementary modes: navigable structure and animated ges-

ture. Both are authored visually and enacted at runtime, directing where the ensemble is within the work and what they attend to while there.

Scrolling sections articulate directional time, while page environments operate as discrete nodes: places the ensemble may enter, inhabit for a period, and return to under different conditions. A page is not simply turned; it is activated: `page(form-G3)`. From there, the score may remain in that state or move on. A navigation cue acts as a hinge, returning the ensemble to the scrolling trajectory or redirecting to another page:

```
nav(mode(scroll@F), speed:0.8).
```

Page mode functions less as a linear succession of frames and more as a network of connected locations within the score.

Pattern structures can specify traversal through these locations:

```
Pxrand([C1:30,C2:120,C3:45],inf)
```

Here, the work moves among several nodes according to probabilistic choice, and each node carries a proposed duration. Form arises through navigation rather than playback: the same environments may appear in different orders and under different temporal conditions, while remaining fully contained within the notational layer. Both deterministic and indeterminate approaches are possible, and complexity can be increased through nesting. This pattern syntax draws on SuperCollider's pattern library [13], adapted for score traversal rather than event generation.

Pauses, timers, and speed changes articulate how long the ensemble remains in these states, but it is the nodal organisation—sometimes fixed, sometimes open—that determines where they are within the piece. The degree of indeterminacy is a compositional decision rather than a property of the system.

Hypertextual traversal may also be explicitly chosen rather than generated. A set of pages can function as a navigable menu, allowing a conductor or performer to move intentionally between environments in performance. The composer authors the available states and their connections, but not the exact route through them. Form emerges through situated decision-making within a network the score itself defines.

Within these structural states, animated transformations —motion along paths, rotation, scaling, and orbital trajectories—act as shared gestural attention fields. A gesture may remain purely interpretive, or it may generate continuous OSC control for live electronics:

```
rotate(dir:-1, dur:6, osc:1, oscaddr:"/
spat/4chPan/src3", uid:ar03)
```

When osc:0, motion is read. When osc:1, motion is also transmitted, allowing acoustic shaping and electronic processing to share a unified grammar of gesture authored in the score.

Across these modes—structural and gestural—the score composes conditions for listening and attention, not merely directives for sound production.

5.2 Distributed Authoring: Propagation and Reuse

Beyond individual cues, Oscilla supports large-scale authoring techniques that remain entirely within the visual workflow.

The propagate(...) pre-parser expands a cue or animation template across all children of a group, assigning each element its own evaluated parameters. This enables variation across many related objects—for instance, clusters of OSC-triggering shapes whose pitch, envelope, or density are derived from their geometry. A collection of freely drawn objects may be assigned a single propagated cue, after which each behaves as if it had been individually composed:

```
propagate(

  osc(
    addr:"/synth/pontalist",
    pitch:deg(${1}, ${2}),
    env:size,
    uid:rndoct1),

  irand(0,11),
  rnd([1,3,4])

  )
```

Here, irand generates new random values continuously, while rnd randomises only at initial parse—a choice between static variation and ongoing mutation.

Complementing propagation, reusable blocks allow structural modules to be defined once and redeployed elsewhere in the score. Groups marked for reuse can be inserted via use(...), preserving internal cues and animations while automatically avoiding identifier collisions. This supports repeatable interface elements, menu structures, and recurring formal modules without leaving the graphic domain.

Together, propagation and reuse introduce modularity and scalability into the notation: visual structures may be repeated, distributed, and differentiated, while remaining authored and legible as part of a single SVG document.

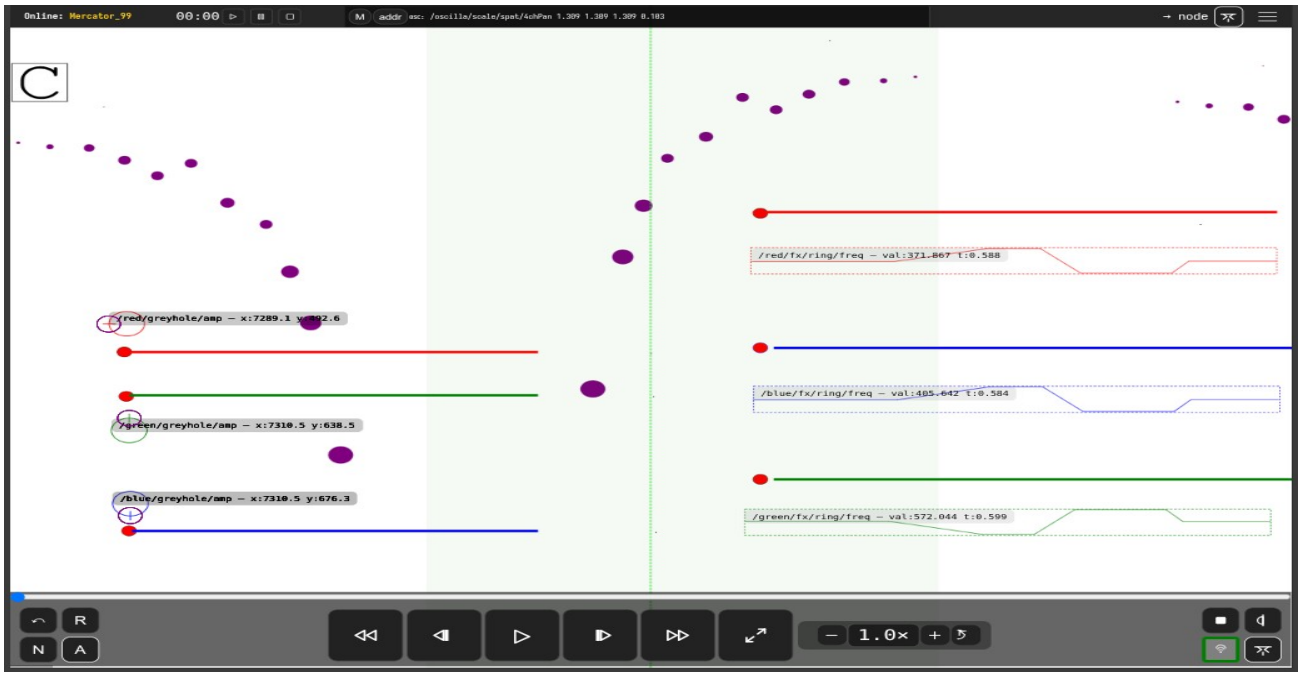


Figure 1. *Section C from Ways Through, Not To.: This excerpt shows the scrolling environment with play-head-triggered nodes (Y=pitch, size=amplitude), indeterminate instrumental lines, o2p gestures for spatial control, and breakpoint curves driving ring modulation. In this screenshot the toggle-able OSC preview overlays are active and visible showing real-time message data directly attached to their source objects..*

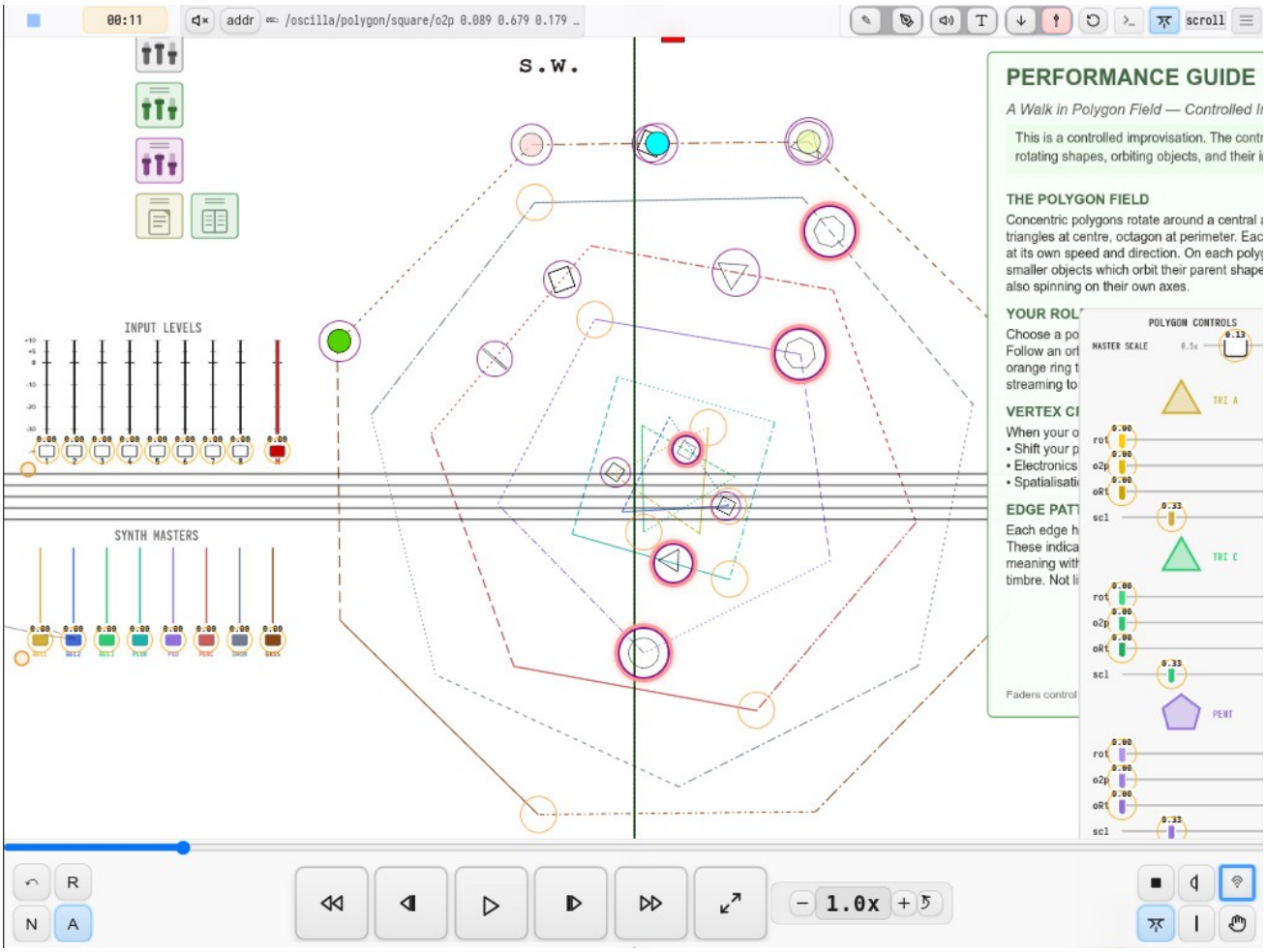


Figure 2 Polygonal paths and an outer spatialisation ring structure performer and electronic behaviour within a constrained yet open environment.

5.3 From Framework to Repertoire

These structures enable scores in which form is composed yet remains flexible, interpretation is guided without being prescribed, and animated gesture functions simultaneously as notation and behaviour. The following case study from Ways Through, Not To. demonstrates how these principles operate in a concrete musical context.

6. CASE STUDY: "A WALK IN POLYGON FIELD"

The preceding sections addressed the system in general terms. The following example presents a section from the developing work *Ways Through, Not To.*, showing how structural, animated, and electronic elements operate when embedded in a real score. This section—titled *A Walk in Polygon Field*—occupies a page-based environment where geometry, motion, and control data become formative agents of form.

6.1 Polygon Fields and Distributed Behaviour

A Walk in Polygon Field takes place within a page-based environment built from concentric polygonal forms. At the centre, a pair of triangles is enclosed by a square, which is in turn enclosed by a pentagon, and so outward to an octagon at the perimeter. Each polygon rotates around the central axis at its own speed and direction, with default states adjustable via a control panel authored directly within the score.

On each polygon sits a pair of smaller "mini polygons," which both orbit their parent shape and rotate on their own axes. These nested motions—orbit and spin—are independently controllable, producing a layered field of visual polyrhythm.

Control faders for all animations are constructed within the score itself, using o2p objects moving along straight paths. These faders communicate via Oscilla's internal signal bus—a subscription system analogous to Pure Data's send and receive objects—allowing any visual parameter to be adjusted in performance. Alongside the animation controls, the score includes mixer faders for instrument input and output levels, effects sends, and synthesiser levels. These mixer controls communicate directly with the SuperCollider patch via OSC, which handles all live electronics and spatialisation.

Any animated element may also expose its motion as OSC data: double-clicking its handle enables continuous streaming of rotation or orbit position to the electronics performer, who may map these streams to synthesis parameters, spectral processing, or spatialisation. In addition, discrete OSC triggers are sent each time an orbiting mini polygon crosses a vertex of its parent shape, providing a rhythmic pulse of attack events synchronised to the visual structure.

Performers may interact with the score directly, pausing and resuming their follower objects via touch control on their tablets. All animations and score state are synchronised over the network; each performer's selection is indicated by a distinct colour assigned at score load time, making individual agency visible within the shared field.

Each side of each polygon carries a distinct stroke pattern —solid, dashed, or dotted—indicating different performance modes: attack type, articulation, or timbral approach, to be negotiated among performers. Performers follow an orbiting mini polygon along these paths, transitioning from one mode to another as their object crosses a vertex. These transitions coincide with the vertex triggers sent to electronics, binding acoustic gesture and electronic response.

Spatialisation is similarly coupled to the score. Each performer's microphone output is mapped to the 8.1 speaker ring according to the rotational position of their parent polygon; vertex crossings trigger shifts in ring modulation frequency applied to that instrument's signal. The result is a tightly intertwined sonic space in which instrumental sound, spatial position, and electronic transformation move together.

Additional structural layers are available to performers who choose to engage with them. The polygon group sits atop a 2x2 quadrat; as an object moves from one quadrant to another, this provides a structural marker that may indicate a mapping change. Collisions or proximity with another performer's object may suggest interaction—whether dynamic, mimetic, or otherwise. There are as many interpretive layers as performers can hold in mind; the score offers depth without mandating it.

The live electronics provide a harmonic and textural field —a space the musicians inhabit as they listen and interpret. Performers are not obliged to follow the score unquestioningly; they should listen, find space, respond, pause, using the animating notation as a guide rather than a dictate.

A performance note explaining how to interpret the score is embedded within the score itself.

7. CONCLUSION

The approach presented here treats the score as a graphic–semantic surface in which visual form, temporal

structure, and performance behaviour are composed together. By embedding cue semantics directly into standard SVG notation authored in Inkscape, the score functions simultaneously as image, structural framework, and executable performance environment. Temporal articulation, sectional navigation, interpretive orientation, media integration, and gestural motion fields are composed at the level of the drawing itself, without requiring symbolic engraving, custom playback interfaces, or auxiliary software patches.

Because authoring occurs in a general-purpose vector editor, the workflow remains accessible to composers and performers without programming expertise or specialised technical infrastructure. The score may be created, revised, and expressed entirely in visual terms. This lowers the barrier to working with animated and temporally active notation, supporting practices where compositional thinking proceeds through drawing, spatial reasoning, and embodied interpretive knowledge rather than formal symbolic encoding.

The resulting performance practice emphasises coordinated attention, shared orientation, and collectively navigated form. Future work will focus on refining authoring ergonomics, expanding documentation and educational resources, and developing further repertoire that explores the musical implications of executable notation.

More broadly, this approach suggests that digital graphic scoring need not replicate symbolic idioms nor treat the score as a passive display, but may instead operate as a composed environment for performance—one in which notation and behaviour coincide on the same surface.

The Oscilla system, documentation, and example scores are available at [14].

8. REFERENCES

[1] R. Canning, "Real-time web technologies in the networked performance environment," in *Proc. Int. Computer Music Conf. (ICMC)*, Ann Arbor, MI, 2012.

[2] R. Canning, "Interactive parallax scrolling score interface for composed networked improvisation," in *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*, 2014, pp. 144–146.

[3] R. Canning, "OscillaScore: A Modular Platform for Graphic Notation in Networked Music Performance," in *Proc. Int. Conf. Technologies for Music Notation and Representation (TENOR)*, Beijing, 2025.

[4] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition*, Pendragon Press, 1992.

[5] D. Fober, Y. Orlarey, and S. Letz, "INScore – An Environment for the Design of Live Music Scores," in *Proc. Linux Audio Conf. (LAC)*, Stanford, 2012, pp. 47–54.

[6] N. Didkovsky and G. Hajdu, "MaxScore: Music Notation in Max/MSP," in *Proc. Int. Computer Music Conf. (ICMC)*, Belfast, 2008, pp. 483–486.

[7] A. Agostini and D. Ghisi, "Bach: An Environment for Computer-Aided Composition in Max," in *Proc. Int. Computer Music Conf. (ICMC),* Ljubljana, 2012, pp. 373–378.

[8] T. Coduys and G. Ferry, "IanniX: Aesthetical/Symbolic Visualisations for Hypermedia Composition," in *Proc. Sound and Music Computing Conf. (SMC)*, 2004.

[9] G. Hajdu, "Quintet.net: An Environment for Composing and Performing Music on the Internet," *Leonardo*, vol. 38, no. 1, pp. 23–30, 2005.

[10] C. Hope, A. Wyatt, and L. Vickery, "The Decibel ScorePlayer: New Developments and Improved Functionality," in *Proc. Int. Computer Music Conf. (ICMC)*, Denton, 2015.

[11] A. Yoon, "neoscore: A Python Library for Notating Music in a Graphics-First Paradigm," 2022. Available: https://neoscore.org/

[12] C. Vear, *The Digital Score: Musicianship, Creativity and Innovation*, Routledge, 2019.

[13] J. McCartney, "SuperCollider: A New Real-Time Sound Synthesis Language," in *Proc. Int. Computer Music Conf. (ICMC)*, 1996.

[14] R. Canning, "Oscilla" 2025. Available: https://oscilla.cc