Oscilla: The Score as Performable Interface

Rob Canning
rc@kiben.net
Independent Researcher

Abstract

Oscilla is a free and open source, browser-based system in which the musical score functions as an interactive interface and instrument. Rather than serving solely as a representational artefact, the score becomes a performable surface: visual elements operate as controls, spatial regions define sonic and structural conditions, and navigation constitutes a primary performance gesture. Notation, interaction, audio behaviour, and signal routing coexist within a unified environment that runs entirely in the browser and synchronises across networked clients.

This paper presents Oscilla's design, interaction models, and use across a spectrum of practice, from composed networked ensemble scores to improvised solo performance using animated spatial constellations. Spatial traversal, page-based navigation, path-constrained touch interaction, and continuous control lanes function as control structures within the score. A signal routing architecture enables cross-cue modulation, positioning the score as a performed signal topology. A session layer allows performers to restructure, annotate, draw on, and sonically extend the score during performance. Across these modes, performers exercise selective agency, choosing when to engage and withdraw from active control within a shared or solo musical structure. The system is situated within a taxonomy of score types spanning control, interactive, stochastic, generative, and retroactive modes.

Keywords

graphic notation, score-as-instrument, browser-based interface, selective agency, networked performance

1 Introduction

New musical interfaces are often conceived as physical devices, sensor assemblies, or specialised software instruments. Scores, by contrast, are typically treated as representational objects: artefacts that describe music rather than enact it.

Oscilla begins from a different premise. The score itself is the interface. It is not only read, but navigated, activated, and played. Visual elements become interactive controls, spatial regions define sonic and structural conditions, and navigation through the score constitutes a primary performance gesture.

Oscilla is a browser-based system for animated graphic notation developed to support synchronised, distributed performance using SVG scores, networked timing, and OSC routing [4]. Over time, the system has evolved from a synchronised scrolling display toward a more general conception of the score as a playable environment. Rather than separating notation, control, and sound production into distinct tools, Oscilla embeds these concerns directly into the score surface (Figure 1).

This paper presents Oscilla as a musical interface, focusing on its interaction design and on the range of performance practices it supports, from fully composed networked scores to improvised configurations in which the interface layer itself constitutes the score.

NIME '26, June 23–26, 2026, London, UK

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. A taxonomy of score types developed within the IanniX community provides a recurring point of reference.

2.1 UPIC: Drawing as Compositional Material

Xenakis's UPIC system [27] proposed that drawn graphic form could itself be compositional material, rendered directly into sound via synthesis. Oscilla shares this premise (that musical thought may proceed visually) but diverges in that performance semantics remain attached to the drawing: the score continues to operate as a score, read and enacted by performers.

2.2 IanniX: Graphical Sequencing and Score Typology

IanniX [6], a direct descendant of UPIC, is an open source graphical sequencer built around three primitives: *triggers* (discrete events), *curves* (spatial trajectories), and *cursors* (playheads that move along curves, emitting continuous positional data and firing triggers they encounter). The cursor-along-curve model is the closest precedent to Oscilla's object-to-path (o2p) interaction, in which a performer drags an object constrained to a drawn SVG path, emitting high-resolution OSC data based on normalised position. Oscilla's *oscCtrl* continuous control lanes, where the global playhead reads values from a drawn path as it traverses, are closer to IanniX's standard cursor behaviour.

Coduys and Jacquemin [7] identify six score types that frame the systems discussed below: *control* (autonomous, deterministic sequencing of external processes), *reactive* (visualising incoming data without generating output), *interactive* (bidirectional exchange, the score as mapping device), *stochastic* (incorporating randomised elements), *generative* (self-generating structures), and *retroactive* (output fed back as input, producing self-modifying scores).

Oscilla's cue-driven playhead and oscCtrl lanes function as control scores; its signal routing and o2p touch faders operate as interactive scores; its audio pools and pattern functions (`Prand()`, `Pshuf()`) introduce stochastic behaviour; and its constellation practice (Section 6.1) produces generative structures through animated spatial geometry. The retroactive mode, in which a score's output modifies its own structure, is supported through Oscilla's signal routing but has not yet been explored in compositional practice. A rotation angle modulating the rotation speed of its own group would constitute a simple retroactive configuration; more complex feedback topologies are architecturally possible.

Scordato [23] documents compositional practice with IanniX across these categories, including spatialisation, audiovisual works, and interactive installations. IanniX generates no audio,

delegating synthesis entirely to external environments via OSC, MIDI, or other protocols. Where IanniX is a pure controller requiring a desktop application, Oscilla embeds audio alongside OSC control and runs entirely in the browser.

IanniX was a direct influence on Oscilla's animation and cursor-driven features. Oscilla's design emerged from combining the animated spatial sequencing that IanniX pioneered with the author's earlier browser-based systems: NodeScore [2], a page-based non-linear score environment, and Parallaxis [3], a networked scrolling score with independently variable part speeds.

2.3 INScore: Scripted Augmented Notation

INScore [12, 13] is an open source environment for interactive, augmented music scores developed at GRAME-CNCM over more than a decade. It composes heterogeneous objects: symbolic notation (Guido Music Notation, MusicXML), images, text, and graphic signal representations. These are controlled through an OSC-based scripting language, with temporal and spatial event triggers and an embedded IRCAM gesture follower. Originally designed to be driven from dataflow environments, its web version (INScoreJS, via WebAssembly) adds Faust-based signal processing.

In terms of the IanniX taxonomy, INScore operates primarily as a control and reactive environment: scripted commands sequence display changes (control), while signal objects visualise incoming performance data as curves (reactive). Its event and gesture-following systems add interactive capability.

A distinctive feature is INScore's score-level composition operators [18]: symbolic scores can be combined in sequence, parallel, transposed, or rhythmically restructured through scripting, with support for both static and dynamic evaluation, enabling real-time, computed transformations of symbolic notation that have no equivalent in Oscilla's graphic-only model.

INScore and Oscilla share common ground: both are open source, treat score elements as interactive objects, and communicate via OSC. The primary divergence is authoring model. INScore scores are programmed through scripts; the score is a display layer orchestrated by textual commands. Oscilla scores are drawn in Inkscape, with behaviour declared locally through identifiers on visual elements. Where INScore supports conventional notation alongside graphic objects, Oscilla works exclusively with SVG graphic notation.

Oscilla's signal routing connects control sources to target parameters within the score's behavioural layer, whereas INScore's signal objects serve a visualisation role. INScore's scripting and composition operators offer considerable power for computed and symbolic score manipulation; Oscilla prioritises visual authoring accessible to non-programmers, at the cost of lacking any symbolic notation support. Systems such as MaxScore [9] and the Bach library [1] share INScore's programmatic orientation, situating notation within dataflow environments.

2.4 Decibel ScorePlayer

The Decibel ScorePlayer [15], developed since 2012 by the Decibel new music ensemble, provides network-synchronised scrolling of graphic notation with embedded audio and OSC communication, running on iPad, Mac, and Android. Score images can be created in any graphics tool and packaged into bundles using a dedicated macOS application; the player source code is available under GPL-3.0. In its core scrolling mode, the ScorePlayer functions as a control score: the playhead moves at a fixed speed, deterministically sequencing the performer's encounter with notation.

The 2019 introduction of a canvas scoring mode [26] extended the system into interactive and generative territory: external applications (Max, Python) can send OSC drawing commands to create, animate, and remove visual objects on the iPad in real time, supporting generative and animated notation at frame rates tested up to 66 FPS. A Python library further lowers the barrier to programmatic score generation.

Subsequent work [25] added an annotation layer for performer markings, per-part audio within a single score file, and QR-based score distribution, each driven by direct user feedback. Audio integration pairs a single file to the score's full duration, allowing waveform visualisation to align with notation. This is well suited to works for graphic notation and fixed media.

Oscilla takes a different path: playhead speed is variable and performed, audio is triggered per-region rather than paired to a fixed timeline, and visual elements are authored directly as interactive components in Inkscape. Even with canvas mode, Decibel's interactive behaviour originates from external applications, whereas in Oscilla behaviour is embedded in the score's SVG elements. Oscilla has not attempted waveform-synchronised display. Both systems now support freehand performer annotation: Decibel via a bitmap ink layer [25], Oscilla via SVG stroke drawing that can be local or shared across the network.

2.5 Networked and Code-First Approaches

Hajdu's Quintet.net [14] established an early model for networked notation under conductor control, sharing with Oscilla an emphasis on opening graphic score performance to wider participation through accessible network technology. Oscilla is agnostic on hierarchical structure, however: scores can be conductor-led, distributed, or anywhere in between. The author's own work tends toward the distributed end, but this is a per-composition decision rather than a feature of the system. More recently, neoscore [28] takes a code-first approach to graphic notation as a Python library. Where neoscore is code-first, Oscilla is drawing-first: behaviour is attached through a compact microsyntax embedded in SVG attributes, with optional DSL expressions available at runtime. Both reject the assumptions of traditional engraving software but occupy different points on the authoring spectrum.

2.6 Theoretical Frameworks

Emmerson [10], addressing the preservation of live electronic works, proposed the "superscore" (a score expanded to include video and sound elements alongside notation) as a documentary format capable of sustaining works beyond the lifetimes of their creators. The concept anticipates the kind of multimedia, executable score that Oscilla implements, though Emmerson's framing is archival rather than performative.

Vear's taxonomy of the digital score [24] identifies features including the "animated score," "technological conductor," and "system as score." Oscilla spans several of these categories. The contribution of the present work is to treat the score as a single executable surface in which visual composition, temporal structure, performer interaction, and electronic control are co-authored inside the same drawing.

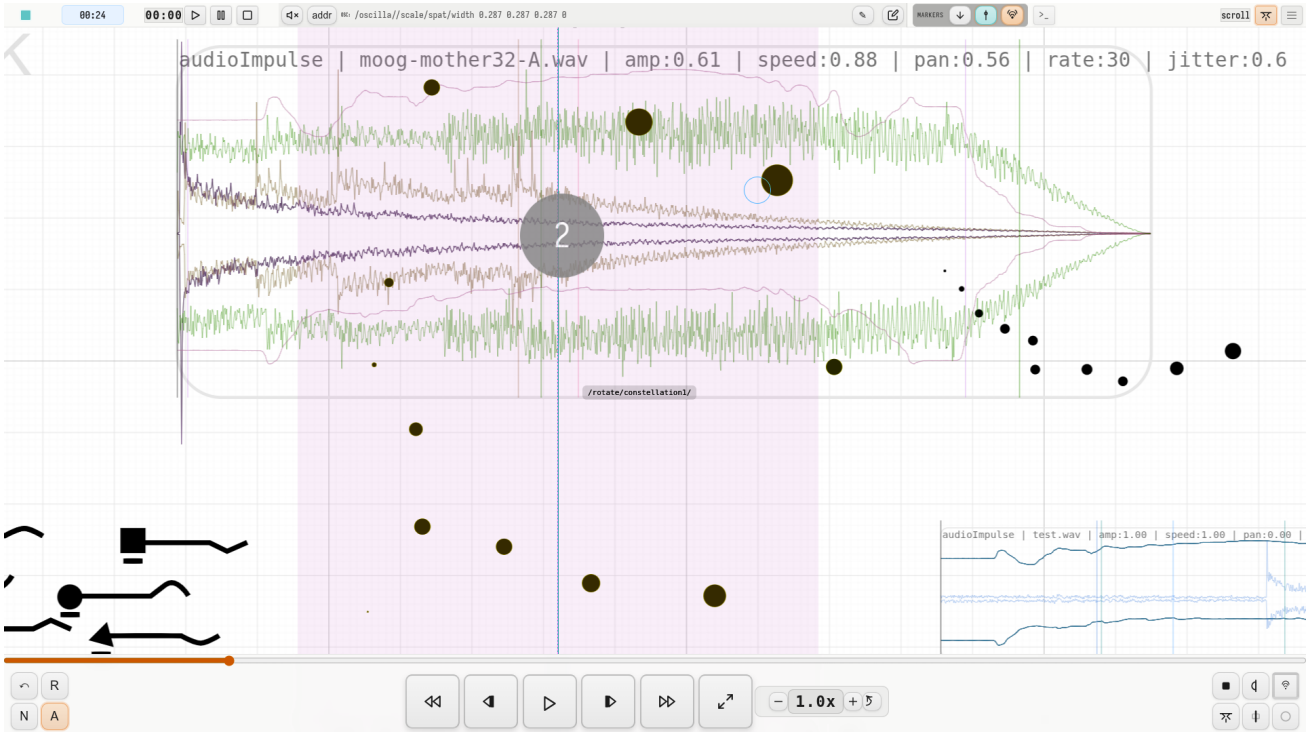

Figure 1: An Oscilla score in scroll mode. Visual elements function simultaneously as notation, interactive controls, and sound triggers within a single browser-based environment.

3 System Design

This section describes the architectural and design decisions that shape Oscilla as a platform. The interaction models that constitute the score-as-interface are presented in Section 4.

3.1 Design Motivation

The environments in which scores are authored and performed shape the music that emerges from them, an instance of the broader principle, traced by Kittler [17], that writing tools work on the thoughts they record. Conventional staff notation, for all its power, embeds assumptions about linearity, discrete pitch, and sequential time that constrain certain kinds of musical thought. The graphic notation practices surveyed by Karkoschka [16] (scores built from spatial form, motion, colour, and non-linear structure) demonstrated that notation could think differently, but remained bound to the printed page.

Oscilla asks what happens when those graphic forms become interactive and executable: when drawn shapes trigger sound, when spatial arrangement generates structure, and when the score can be navigated rather than only read.

Oscilla developed from the conviction that building the instrument and composing the music are not separable activities, and that designing how a score behaves is itself compositional work. A long involvement with free software communities reinforced this perspective: in cultures where tools are collectively built, modified, and redistributed, the boundary between user and maker dissolves. Oscilla applies the same logic to the score, treating it as a shared, modifiable instrument rather than a fixed artefact delivered from composer to performer.

In practice, this means choosing tools that lower barriers rather than raise them. Many existing systems for interactive or electronic performance require significant setup, specialised software, or hardware dependencies, limiting participation in workshops, educational settings, or ad-hoc ensembles. Oscilla leverages the modern web browser (capable of graphics, animation, audio, and network communication) to create a system that requires no installation beyond opening a URL. Scores can be shared instantly, viewed on multiple devices, and updated without redistributing files or software. Complete projects can also be exported as `.oscilla` files (zip archives bundling SVG scores, configurations, and audio assets) for offline distribution, archiving, or transfer between Oscilla servers. External OSC targets (synthesis patches, spatialisation rigs, hardware controllers) remain composition-specific and travel separately from the score itself.

The scope of Oscilla is defined less by sonic ambition than by interaction design. Its built-in audio and synthesis capabilities are intentionally modest; where greater complexity is required, OSC routing [22] allows the score to control external synthesis, spatialisation, or media systems.

3.2 Free and Open Source Development

Oscilla is developed as free and open source software. Ensembles can adapt scores to local contexts, extend cue behaviours, and fork projects without licensing constraints or proprietary formats. The system can be deployed locally or on private networks without reliance on external services or subscription models.

3.3 Visual Authoring in Inkscape

A defining aspect of Oscilla is its use of Inkscape as the primary score authoring environment. Rather than introducing a bespoke editor, Oscilla treats a widely available vector graphics tool as an instrument-building workspace. Composers and performers design the visual and interactive structure of the score using

familiar graphical operations: drawing, grouping, layering, alignment, and colour.

This choice has a significant aesthetic implication. Dedicated graphic score editors, whether oriented toward education (such as Hyperscore [11]) or toward compositional research, necessarily define a visual vocabulary: the shapes, colours, and gestures available to the composer are those the tool provides. While this constraint supports accessibility and can produce engaging results, it also means that scores authored in such systems tend to look like the system rather than like the work of a particular composer. The visual language converges on the editor's defaults.

Inkscape imposes no such convergence at the level of visual vocabulary. Because the authoring environment is a general-purpose vector graphics tool, composers define their own shape language from scratch, though authoring hardware (mouse, trackpad, stylus) still inflects line quality and gestural feel. The score's visual identity, including its colours, shapes, line qualities, spatial organisation, and typographic choices, belongs entirely to the author. This openness is both a strength and a demand: it requires compositional decisions about visual form that a more constrained editor would make on the composer's behalf. Oscilla embraces this trade-off, treating the freedom to define a unique graphic language as integral to the score-as-instrument concept.

Behaviour is attached to visual elements through identifiers in SVG attributes, using a compact microsyntax. Shapes and regions acquire interactive meaning without requiring programming, while technical users can extend behaviour through custom cues or OSC mappings. The score remains a readable, editable drawing that encodes its own performance behaviour.

4 The Score as Performable Interface

In Oscilla, the score is an SVG canvas populated by visual elements (shapes, text, paths, and regions) that function as interactive objects associated with behaviours through *cues*. At a high level, the system distinguishes between cue triggering (discrete events bound to score elements), continuous control signals (streamed from faders, paths, and animations), and navigation state (playhead position, page transitions), all authored visually and connected through a shared routing layer.

A cue is a declarative instruction attached to a visual element that defines what should occur when specific conditions are met. Cues are performative rather than representational: a cue may initiate sound, animate a visual element, alter navigation behaviour, transmit OSC data to an external system, or any combination of these. Because cues are declarative, composers specify relationships between score elements and behaviours rather than programming signal flows or timelines.

This section presents the interaction models through which the score operates as a performable interface.

4.1 Spatial Traversal and Page Navigation

Oscilla supports two primary temporal models. In continuous spatial traversal, a global playhead moves across the score surface, and its position determines which cues are active. Traversal speed defines tempo, while changes in direction, looping, or branching alter formal structure. From an instrumental perspective, playhead traversal functions as a gestural input: adjusting speed, reversing direction, or jumping between regions produces immediate musical results.

In page-based navigation, the score operates as a set of linked views. Cues trigger jumps between pages rather than continuous movement across a surface. Pages may represent sections, states, or alternative pathways through the material. This enables a hypertextual approach to musical form in which performers choose pathways, activate conditional transitions, or revisit material non-sequentially. Navigation itself becomes a performance gesture, and form emerges through choice rather than linear progression.

Both modes may coexist within a single score. Scrolling sections articulate directional time, while page environments operate as discrete nodes: places the ensemble may enter, inhabit for a period, and leave under different conditions.

4.2 Embedded Audio and Synthesis

Audio in Oscilla is governed by the same cue system that controls navigation and animation. Sound is therefore conditioned by traversal, spatial presence, and interaction rather than an independent timeline. Sample playback supports parameters such as amplitude, stereo position, playback rate, looping, and fades, which may be fixed or bounded to allow controlled variability. Audio pools define collections of sound files selected according to strategies such as shuffle or random, and when combined with region-bound activation, enable stochastic sonic fields that persist while performers inhabit a region.

Lightweight synthesis supports tones, drones, and textures controlled through pattern functions (`Pseq()`, `Prand()`, `Pshuf()`) whose naming follows SuperCollider conventions [20] but which are implemented natively in Oscilla's browser-based audio engine, governing frequency, amplitude, duration, and filtering. These sounds provide immediate, score-native sonic material without replacing external synthesis environments.

4.3 Drawn Paths as Controllers

Oscilla provides two mechanisms through which hand-drawn SVG paths function directly as control interfaces.

Object-to-path (o2p) interaction in touch mode allows performers to drag an object constrained to an SVG path, emitting high-resolution OSC data based on normalised position along the path. The path itself is the controller; its shape, drawn by the composer, constrains the performer's traversal and so produces a different gestural vocabulary: a spiral, a zigzag, and a smooth curve each afford distinct movements of the dragging hand. Touch-mode faders can be collected into named groups, enabling saving and recalling of all fader positions as named presets, tweening between presets with configurable easing, and sequencing preset recalls over time. Rotation handles add a second control dimension to any fader, supporting compound parameters such as position-plus-angle on a single drawn path.

oscCtrl provides continuous control lanes: an SVG path is drawn alongside or within the notation, and as the playhead traverses it, the vertical position is mapped to a value range and streamed as OSC data. The path functions as a breakpoint automation curve tightly aligned to the score's temporal structure. These lanes are particularly suited to parameters such as ring modulation frequency, filter cutoff, or spatialisation coordinates, allowing the composer to draw control gestures directly into the notation rather than programming them in a separate environment.

Together, o2p and oscCtrl position hand-drawn curves as score-native gestural interfaces whose output can drive external synthesis, spatialisation, or media systems via OSC. Interface design and score authoring become the same activity.

4.4 Signal Routing and Cross-Cue Modulation

Oscilla treats control as a first-class signal layer, separate from cue triggering and animation scheduling. A shared control plane consists of a global parameter bus (ParamBus) that stores named control signals, and a control router that connects signals to targets. All control sources (external OSC input, o2p traversal values, rotation angles, scale factors, XY pad positions) converge through the same mechanism and are addressable by path:

```
o2p:sliderA.t      - fader position
rotate:orb1.angle  - rotation angle
osc:fader1         - external controller
```

Any signal can modulate any running cue parameter. A fader's position may control synthesis amplitude; a rotation angle may drive spatialisation; an external controller may reshape animation speed. Modulation connections support scaling, offset, smoothing, and clamping, and can be composed into signal networks within the score.

This architecture enables cross-cue modulation as an intentional compositional resource. A drawn fader controlling a rotation speed that in turn modulates synthesis frequency is not an accidental feedback loop but a designed signal chain authored within the score. The score thereby operates not only as a trigger environment but as a signal-processing topology whose connections are visually authored and performatively controlled.

4.5 Selective Agency

O2p interactions in touch mode can function as selectively animated parameters that performers may choose to activate or suspend. This supports what we describe as *selective agency*: the ability to intentionally enter and withdraw from active control, opening temporary windows of influence within a shared musical structure.

From a musical perspective, selective agency allows performers to shape parameters for a period of time and then step back, making space for others without relinquishing the shared structure of the score. Control is not treated as an always-on resource, but as a negotiated, temporal affordance. Croft's conditions of instrumentality [8] establish when live electronics are perceived as agentive rather than environmental; selective agency extends this by treating the temporal exercise of that instrumentality (the choice of *when* to enter and withdraw from active control) as itself a performed parameter.

The concept extends beyond touch control. Any score element that a performer can activate, suspend, move, or bypass (including playback controls, page navigation, and contribution of new material) participates in the same logic of selective engagement. The score defines a field of possibilities; performers decide which possibilities to realise and when.

5 The Session Layer

The interaction models described above operate on the authored SVG score, but Oscilla also provides a performer-editable stratum, the *Session Layer*, that sits above the composed score and accumulates material during rehearsal and performance. The Session Layer is not a separate tool; it is an integrated set of features through which performers annotate, draw, restructure, extend, and sonically populate the score without editing the underlying SVG.

5.1 Spatial Manipulation

The most immediate Session Layer feature is `drag()`: any SVG group can be made draggable with a single identifier, and drag combines freely with other cues: a group may rotate, scale, and emit OSC data while remaining repositionable by the performer. Drag positions persist across sessions and synchronise across all connected clients, so that spatial arrangement becomes a shared, evolving aspect of the score. This supports practices ranging from performer-led arrangement of aleatoric score fragments, producing distinct versions of a work through spatial ordering, to using spatial position as a live control parameter during performance.

5.2 Temporal Scaffolding

Oscilla's timer system provides synchronised countdown sequences visible to all connected clients. A sequence is a named group of timed sections that play consecutively, with optional looping and chaining between sequences. For example: *sparse texture* (60s), *build* (90s), *peak* (45s), *decay* (120s). The timer is server-owned: the server maintains authoritative state, and late-joining clients receive the current countdown immediately. Each section supports on-complete actions, enabling timed structural transitions: a section ending can trigger a navigation jump to a named marker (e.g. `nav(@mark:sectionB)`), coupling durational scaffolding directly to spatial and formal structure within the score.

Markers provide a complementary, lighter-weight mechanism. Performers drop visual waypoints onto the score during playback, rehearsal, or pre-performance planning, a simple way to sketch structure onto a shared timeline. Markers are colour-coded per client by default, with user-adjustable colour to support visual groupings, and may be shared across the network or kept local. They can be labelled with section names, solo assignments, or coordination notes. Named markers integrate directly into Oscilla's transport: they are navigable via keyboard shortcuts and transport buttons, targetable by `nav()` cues from within the SVG score, and addressable as destinations for timer on-complete actions, bridging casual annotation and structured performance navigation.

5.3 Sonic Contributions

Text and audio contributions add material directly to the score. Contributions may be purely textual (rehearsal notes, section labels, coordination cues) or executable, functioning as clickable or playhead-triggered sound events with configurable amplitude, pan, pitch, looping, and fade parameters. Audio pools and stochastic impulse generators can be assigned to contributions, and each contribution defines a horizontal trigger region using draggable extent lines that couple sound to spatial position in the score.

An integrated audio recorder allows performers to record short audio contributions directly into the score; recordings are stored within annotation-scoped directories and can immediately be assigned to existing triggers.

5.4 Freehand Drawing

A drawing layer allows performers to mark up the score freehand using finger, stylus, or mouse directly in the browser. Strokes are rendered as SVG paths in world coordinates, anchored to the

correct score position across devices and zoom levels. Drawings can be local or shared across the network, and function as visual annotations (rehearsal marks, dynamic contours, circled passages, conductor gestures) without affecting cues or synchronisation. Drawing and all other interactive elements behave uniformly across mouse, stylus, and finger touch; on multitouch devices the full interface is reachable, and authors can lay out controls so that performance gestures sit ergonomically under one or both hands.

5.5 The Session Layer as Score

Contributions, markers, timers, drawings, and drag positions may all be local or shared across the network, enabling real-time negotiation of structure during rehearsal or performance. The two strata serve different roles: the SVG score is the authoritative, durable layer authored in advance, while the Session Layer is ephemeral, performative, and annotational. Some compositional approaches exploit the Session Layer alone: those for which audio cues, timing markers, and text annotations suffice and animation or extensive OSC routing is not required. The author's sound score *What You Heard* [5] adopts this approach. More generally, an ensemble can load any image as a backdrop, drop markers for structural landmarks, set up timer sequences for durational scaffolding, add audio contributions as sonic material, draw annotations directly onto the score surface, and use o2p faders as gestural controllers—and that configuration *is* the score. A score may begin as an empty canvas and evolve into a complex instrument through cumulative contribution.

A live coding console further extends this mutability, allowing DSL expressions to be typed and executed directly during performance, modifying animations, triggering synthesis, and patching signal routes without leaving the score environment.

6 Practice

Oscilla has been used in networked ensemble performance and workshop contexts. A commissioned score for a professional trio was developed through iterative remote rehearsal and premiered on synchronised tablets without technical interruption; at the same event, a workshop introduced the system to ten composers who progressed within a single session from first encounter to modifying score behaviour in Inkscape.[1]

These contexts confirmed the system's viability for composed, ensemble-based work. The practice described below explores a different configuration: the score as a composer-performer's instrument, developed toward a forthcoming interdisciplinary collaboration.

6.1 Constellation Practice: Performed Signal Topologies

A developing solo practice explores Oscilla as a performed instrument without pre-composed linear notation. The configuration uses groups of small SVG objects (tens or hundreds) arranged in spatial constellations: arcs, circles, ellipses, grids, and irregular clusters. Each object is assigned an OSC trigger via Oscilla's `propagate()` preprocessor, which expands a single cue template across all children of a group, deriving pitch, envelope, and density parameters from each object's visual properties: position, size, and colour.

[1]These deployments are described in detail in the authors' companion paper presented at TENOR 2025.

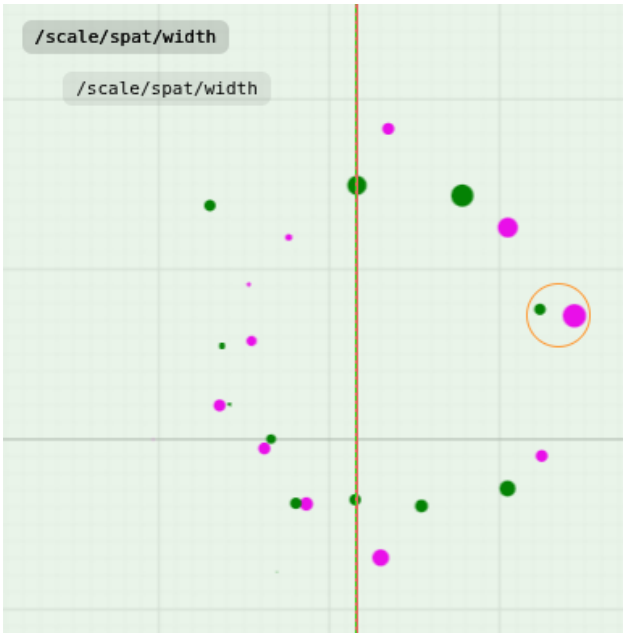

Figure 2: Constellation practice: groups of SVG objects rotate at different speeds around fixed centres. As objects intersect the playhead (vertical line), they trigger synthesis events via OSC. The performer repositions groups using `drag()`, and controls rotation speed and scale through o2p touch faders routed via the signal bus.

These groups are then animated: rotated at different speeds and in different directions, scaled, and layered (Figure 2). The performer's primary gesture is to push these rotating constellations into and out of the playhead's active region. As individual objects intersect the playhead, they trigger synthesis events or samples in the external audio engine via OSC. The resulting patterns depend on the geometry of the constellation, the rotation speed, the playhead position, and the rate of traversal, producing textures that range from sparse, quasi-periodic pulses to dense, rapidly shifting clouds of events.

The playhead itself becomes a performed parameter. When paused, a rotating constellation produces a repeating rhythmic pattern determined by the spatial distribution of objects passing through the stationary playhead. When the playhead moves, the intersection patterns shift continuously. Symmetrical arrangements produce regular periodicities; asymmetric or clustered distributions yield irregular, evolving textures.

Oscilla's `drag()` mechanism extends this practice further. Any group can be made draggable while retaining its running animations: a compound identifier such as `rotate(dir:1, dur:4) drag(1)` produces a group that rotates continuously and can be grabbed and repositioned at any time. Drag positions persist across page reloads and synchronise across all connected clients. In the constellation practice, this allows the performer to pick up rotating groups and place them directly over the playhead, stack multiple constellations in overlapping configurations, or pull them away entirely, treating the score surface as a spatial instrument played by hand.

With `drag(1, osc:1)`, the position of each group is itself emitted as OSC data, adding a continuous positional control dimension to the discrete trigger events generated by playhead intersection.

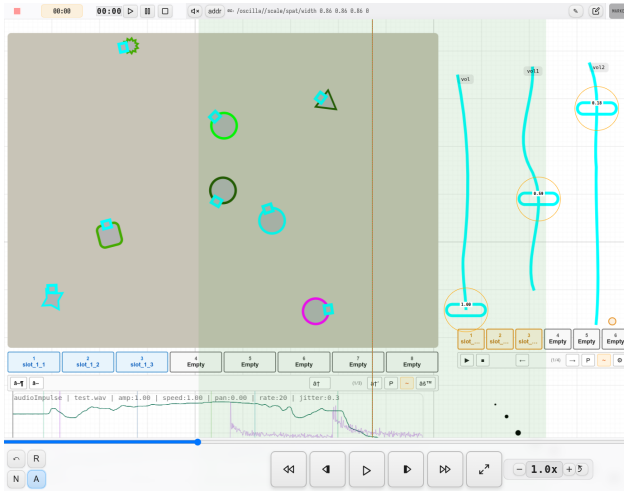

Figure 3: A control surface section of a constellation score. The playhead (vertical line) has paused at a region containing controlXY touch points and o2p faders, following a musique concrète passage whose audioImpulse waveform is visible completing at the bottom of the screen. The touch points (circled objects) send XY position data via OSC to SuperCollider; each is also twistable, functioning as a potentiometer for an additional control dimension. Drawn-path faders at right control volume and spatialisation parameters.

O2p touch faders, routed through the signal bus (Section 4.4), can control rotation speed and scale factor in real time, allowing the performer to reshape the generative topology during performance. Preset sequences with tweened transitions compose movement between stored fader configurations, enabling the performer to design trajectories through parameter space and trigger them as structural gestures. Figure 3 shows a control surface section embedded within a constellation score, combining controlXY touch points and o2p faders alongside audioImpulse playback.

6.2 Scored Improvisation with Dance

This constellation practice forms the basis of a forthcoming work with contemporary dancer Jasmina Križaj, a co-production by the NAGIB and EManat institutes in Slovenia, currently in rehearsal and scheduled for premiere in summer 2026. The audio component of the work will be released separately on the Kamizdat label in autumn 2026. The project requires the score to operate simultaneously in two modes: as a pre-composed formal structure that can be rehearsed and refined across a development period, and as a responsive instrument that allows the musician to react to the dancer's movement in real time during performance.

The constellation configuration supports both demands. Formal structure is composed through the spatial arrangement of groups, the selection of synthesis parameters via `propagate()`, and the design of preset sequences that define trajectories through timbral and rhythmic space. Real-time responsiveness is achieved through drag interaction, fader control of animation parameters, and the ability to activate or suspend constellations selectively, entering and withdrawing from sound-producing regions as the choreography unfolds. The musician can follow a composed arc while remaining free to respond to what is happening on stage: accelerating or delaying transitions, foregrounding or suppressing material, reshaping density and texture in dialogue with the dancer's presence.

In this configuration, the score surface is simultaneously notation (the visual arrangement of objects), instrument (the gestural control of constellations and faders), and signal topology (the modulation routing between controllers and animation parameters). No element of the musician's performance exists outside the score environment. The practice demonstrates that Oscilla's score-as-interface architecture supports not only the performance of composed works but the construction of personal instruments in which composition and improvisation coexist within a single performed surface.

7 Discussion

Treating the score as a performable interface challenges conventional distinctions between notation, interface, and controller. In Oscilla, these roles are deliberately entangled. The same drawn shape may function as notation to be read, a region that triggers sound, and a controller that emits OSC data, often simultaneously.

The practice contexts described above illustrate a spectrum of use. In the commissioned performance, the composer authored a detailed score with embedded cues, and performers engaged with it as a composed instrument whose technical infrastructure became invisible through rehearsal. In the workshop, composers with no prior experience began building interactive scores within a single session, demonstrating that the authoring model is accessible without programming skills.

In the constellation practice, the system operates as a personal instrument for improvisation, with no pre-composed linear material. The interface layer itself is the score, and musical structure arises from the real-time interaction between spatial geometry, animation, and performed gesture. These three contexts are unified not by a common workflow but by the underlying entanglement of notation, interaction, and sound within a single performed surface.

Features such as path-based touch interaction and the Session Layer redistribute authorship between composer and performer. Composers design systems of behaviour; performers realise them through interaction and choice. This redistribution is not unlimited: the composer retains control over the topology of possibilities, the available interactions, and the structural constraints. But within those constraints, performers exercise genuine agency over how the work unfolds.

The notion of selective agency foregrounds not only what performers can do but *when they choose to do it*. In ensemble contexts, the decision to engage or withdraw from a parameter becomes a musical gesture in itself, one that shapes the collective texture as much as any pitch or rhythm. In the constellation practice, selective agency operates differently: the performer negotiates between active gestural control and designed automation, choosing when to intervene in processes that continue independently.

The signal routing architecture, in which any control source can modulate any target parameter, positions the score as a signal topology rather than a flat trigger surface. This connects Oscilla to traditions of modular synthesis and dataflow programming exemplified by Pure Data [19, 21], but situates the patching within a visual, spatially organised score environment rather than a separate programming interface. The constellation practice makes this particularly explicit: the performer constructs and reshapes

a signal network in real time, using the score surface as both the instrument and the control interface.

Emmerson's proposal for a "superscore" [10] (a score incorporating video, sound, and notation to preserve live electronic works) anticipated the need for scores beyond static representation. Oscilla can be read as a realisation of something like the superscore, but reframed: rather than an archival document that records what a work was, the score is a live, executable environment that *is* the work. Its notation, its instrument, and its sounding behaviour coexist in a single performed surface.

What Oscilla contributes beyond existing systems is the combination of three properties not found together elsewhere: behaviour authored locally in visual elements rather than in scripts or external applications; a signal routing layer that connects any control source to any target parameter within the score; and a session layer that allows the score to be restructured, extended, and sonically populated during performance without editing the underlying file. Individually, each has precedent; their integration within a single browser-based, openly licensed environment is new.

For the NIME community, Oscilla offers an alternative approach to interface design in which scores themselves function as dynamic, performable instruments, an engagement with screen-based performance and digital organology [19]. Its reliance on open web standards and free software positions accessibility and openness as design features rather than constraints. Scores authored in Oscilla are instruments that can be shared, modified, and redistributed, lowering barriers to participation while preserving compositional intent.

8 Conclusion

This paper has presented Oscilla as a browser-based system in which the score functions as an interface and instrument. By embedding navigation, interaction, audio behaviour, signal routing, and OSC control directly into the score environment, Oscilla supports a range of performance practices from fully composed networked scores to improvised configurations built from markers, timers, and gestural controllers.

The interaction models described here (spatial traversal, page-based navigation, path-constrained touch control, continuous control lanes, cross-cue modulation, and the Session Layer) position the score as an active site of performer agency rather than a passive representational object. The concept of selective agency offers a framework for thinking about interaction as a temporal, negotiated, and social practice within composed and improvised musical structures.

Future work will focus on extended performance documentation, particularly of the constellation practice described in Section 6.1; refinement of interaction models around multi-performer coordination; and broader investigation of score-as-instrument approaches within the NIME community.

The system, documentation, and example scores are available at https://oscilla.cc.

9 Ethical Standards

This research did not involve human participants, personal data collection, or animal subjects. The system described is developed as free and open source software and does not collect user data. No conflicts of interest are declared.

References

[1] Andrea Agostini and Daniele Ghisi. 2012. Bach: An Environment for Computer-Aided Composition in Max. In *Proceedings of the International Computer Music Conference (ICMC)*. Ljubljana, Slovenia.

[2] Rob Canning. 2012. Real-Time Web Technologies in the Networked Performance Environment. In *Proceedings of the International Computer Music Conference (ICMC)*. Michigan Publishing, Ljubljana, Slovenia.

[3] Rob Canning. 2014. Interactive Parallax Scrolling Score Interface for Composed Networked Improvisation. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*. London, UK. https://doi.org/10.5281/zenodo.1178728

[4] Rob Canning. 2025. OscillaScore: A Modular Platform for Graphic Notation in Networked Music Performance. In *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*. Beijing, China, 214–221.

[5] Rob Canning. 2026. What You Heard. Performance and workshop, International Conference on Technologies for Music Notation and Representation (TENOR).

[6] Thierry Coduys and Guillaume Ferry. 2004. IanniX Aesthetical/Symbolic Visualisations for Hypermedia Composition. In *Proceedings of the 1st Sound and Music Computing Conference (SMC)*. Paris, France.

[7] Thierry Coduys and Guillaume Jacquemin. 2014. Partitions rétroactives avec IanniX. In *Proceedings of the 21st Journées d'Informatique Musicale (JIM)*. Bourges, France.

[8] John Croft. 2007. Theses on Liveness. *Organised Sound* 12, 1 (2007), 59–66. https://doi.org/10.1017/S1355771807001604

[9] Nick Didkovsky and Georg Hajdu. 2008. MaxScore: Music Notation in Max/MSP. In *Proceedings of the International Computer Music Conference (ICMC)*. Belfast, UK.

[10] Simon Emmerson. 2006. In What Form Can 'Live Electronic Music' Live On? *Organised Sound* 11, 3 (2006), 209–219. https://doi.org/10.1017/S1355771806001427

[11] Morwaread M. Farbood, Egon Pasztor, and Kevin Jennings. 2004. Hyperscore: A Graphical Sketchpad for Novice Composers. *IEEE Computer Graphics and Applications* 24, 1 (2004), 50–54. https://doi.org/10.1109/MCG.2004.1255809

[12] Dominique Fober, Stéphane Letz, Yann Orlarey, and Frédéric Bevilacqua. 2013. Programming Interactive Music Scores with INScore. In *Proceedings of the Sound and Music Computing Conference*. Stockholm, Sweden.

[13] Dominique Fober, Yann Orlarey, and Stéphane Letz. 2012. INScore – An Environment for the Design of Live Music Scores. In *Proceedings of the Linux Audio Conference (LAC)*. Stanford, CA, USA, 47–54.

[14] Georg Hajdu. 2005. Quintet.net: An Environment for Composing and Performing Music on the Internet. *Leonardo* 38, 1 (2005), 23–30.

[15] Cat Hope, Aaron Wyatt, and Lindsay Vickery. 2015. The Decibel ScorePlayer: New Developments and Improved Functionality. In *Proceedings of the International Computer Music Conference (ICMC)*. Denton, TX.

[16] Erhard Karkoschka. 1972. *Notation in New Music: A Critical Guide to Interpretation and Realisation*. Praeger, New York. Originally published as *Das Schriftbild der neuen Musik*, Moeck, 1966.

[17] Friedrich A. Kittler. 1999. *Gramophone, Film, Typewriter*. Stanford University Press, Stanford, CA. Translated by Geoffrey Winthrop-Young and Michael Wutz.

[18] Gabriel Lepetit-Aimon, Dominique Fober, Yann Orlarey, and Stéphane Letz. 2016. INScore Expressions to Compose Symbolic Scores. In *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR 2016*. Cambridge, UK, 137–143.

[19] Thor Magnusson. 2019. *Sonic Writing: Technologies of Material, Symbolic, and Signal Inscriptions*. Bloomsbury Academic, London.

[20] James McCartney. 1996. SuperCollider, a New Real-Time Synthesis Language. In *Proceedings of the International Computer Music Conference (ICMC)*. Hong Kong, 257–258.

[21] Miller S. Puckette. 1997. Pure Data. In *Proceedings of the International Computer Music Conference (ICMC)*. International Computer Music Association, Thessaloniki, Greece.

[22] Andrew Schmeder, Adrian Freed, and David Wessel. 2010. Best Practices for Open Sound Control. In *Proceedings of the Linux Audio Conference (LAC)*.

[23] Julian Scordato. 2017. Composing with IanniX. In *Proceedings of the 5th Conference on Computation, Communication, Aesthetics & X (xCoAx)*. Lisbon, Portugal.

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

[25] Aaron Wyatt and Cat Hope. 2022. The Decibel ScorePlayer – Learning From And For Usability. In *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'22*. Marseille, France, 143–148.

[26] Aaron Wyatt, Lindsay Vickery, and Stuart James. 2019. Unlocking the Decibel ScorePlayer. In *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*. Melbourne, Australia, 61–68.

[27] Iannis Xenakis. 1992. *Formalized Music: Thought and Mathematics in Composition*. Pendragon Press.

[28] Andrew Yoon. 2022. neoscore: A Python Library for Notating Music in a Graphics-First Paradigm. Software. https://github.com/DigiScore/neoscore BSD-3-Clause licence.