Performative transactions: worlding compositional ecosystems

Lukawski, A.

Chapter 5. Autopoietic Rhizomatic Metamodelling Machine

Designing the Abstract Machine

The preceding chapters have aimed to describe, analyse, and expand upon current state-of-the-art regarding the diverse roles artists assume within an era shaped by emerging technologies and novel artistic methodologies. On the one hand, they formalised some of the ways in which artists currently deal with the task of composition in response to emerging technologies. On the other hand, they proposed new concepts such as allagmatic composition and the associated mode of worlding compositional ecosystems, identifying new directions for possible further development of artistic practices. The following text consolidates these previously explored roles and concepts. It does so by worlding a new compositional ecosystem, following the practice as described in the previous chapter by: 1) envisioning a set of abstract relations between various processes, elements and actants to world the operations of a new abstract machine (in the current chapter), and 2) to operationalise the workings of the newly envisioned abstract machine in a contrete technical realisation (in Chapter 6).

The abstract machine proposed here is conceived as an Autopoietic Rhizomatic Metamodelling Machine (ARMM)—a conceptual and operational framework designed to integrate the previously described roles of artists into a unified compositional ecosystem. Its purpose is to enable diverse types of artistic compositions—generative, interactive, and allagmatic—to be instantiated seamlessly, without significant friction or the need for extensive redesign of the framework that accomodates them. Such a creative machine is required precisely because contemporary artistic practice increasingly demands frameworks capable of accommodating multiple, overlapping roles and creative methodologies. As we have seen, artistic practice today moves fluidly between making discrete artefacts, curating relational interactions, designing generative and interactive processes, and structuring entire creative ecosystems. Traditional artistic tools and conceptual paradigms, typically designed to address specific, isolated needs, often struggle to support this fluidity. The ARMM addresses this gap by envisioning an explicitly flexible, modular, and operationally rich environment, allowing artists to seamlessly transition between—or simultaneously engage in—these varied roles within a single, coherent compositional ecology.

This abstract machine emerges directly from an artistic research process marked by iterative self-reflection, experimentation, and practical encounter. It is an outcome of the problematisation inherent in artistic research—where the reflective practice of the composer-researcher generates new conceptual and methodological insights that lead directly to the construction of new artistic objects, practices, and methods. In this context, the abstract machine functions as a machinic embodiment of problematisation: it operationalises the researcher's discoveries, translating theoretical advancements into practical compositional infrastructures. It is explicitly designed as an assemblage—drawing together diverse conceptual threads, artistic experiences, technological encounters, and scientific insights encountered throughout this research. Its design integrates philosophical frameworks (such as those derived from Deleuze and Guattari's notions of the rhizome and Guattari's redefinition of the concept of autopoiesis), computational methodologies (particularly metamodelling and generative processes), and artistic methodologies previously articulated in this volume.

Importantly, this envisioning of the ARMM sets the stage for its operationalisation in Chapter 6. There, this abstract machine will be instantiated as a concrete machinic assemblage—specifically through the concepts of Decentralised Creative Networks and Performative Transactions. ARMM will serve as a diagramatic generative engine for designing a new creative infrastructure, one that is both conceptually grounded and operationally viable. The readers should keep in mind that the operations of the abstract machine conceived in the current chapter are yet not to be understood as mechanisms of any concrete technical machine and should not be attempted to be understood in this way. They are mechanisms of a philosophical, abstract machine the purpose of which is to "generate" its own operationalisation within a concrete technical machine in Chapter 6. It is a conceptual preparation that aims to elucidate some of the practical and experiential reasons for which the technical operations of the machine introduced in Chapter 6 are functioning in the ways technically described. Thus, reading Chapters 5 and 6 together should allow the reader to follow the transition from conceptual design to practical implementation—revealing how a speculative machine can structure real-world systems of artistic production and interaction. What is more, this chapter aims to demonstrate how artistic research—through cycles of conceptual inquiry, experimental prototyping, speculative design and iterative reflection—can yield genuinely new artistic ecosystems.

Software Design Principles after Miller Puckette

No speculative machine can afford to overlook the lessons of lived practice. Venturing into the conceptual design of an abstract machine for future compositional ecosystems, we must engage with the practical knowledge of those who have already built the tools that shaped generations of artistic thinking. Few figures have been more central to this domain than Miller Puckette. As the creator of Max and later Pure Data—software environments that redefined how composers, performers, and researchers interact with sound—Puckette stands not only as a technical innovator but as a visioner of compositional infrastructure. In his reflections on music software development, Puckette identifies a range of principles that cut through trends and technological fads to address enduring compositional needs. His work provides a grounded, hard-earned perspective on what it means to design systems that are open yet stable, user-configurable yet technically rigorous, and conceptually rich without becoming inaccessible. In search for operational principles, this section draws selectively from Puckette's observations to distil a set of pragmatic design considerations that can inform the development of more speculative compositional systems. What follows is not a comprehensive summary of Puckette's approach, but an extraction of principles that resonate with the aims of this chapter.

At the core of Puckette's reflections lies a fundamental concern: what kind of tool would enable musicians to work with the same sonic precision and scope as the recording studio, but in real time? The initial motivation behind the creation of Max and later Pure Data was precisely this—an attempt to construct an instrument capable of translating live musical intuition into sound while preserving the structural complexity and flexibility of studio-based workflows (Puckette, 2022, 27). What emerged was a reactive infrastructure designed for real-time interaction: a system to be shaped and performed dynamically, continuously adapting to the composer's or performer's evolving intentions in the flow of creation (26). What enabled this responsiveness was a software model inspired by analogue computing: modular nodes connected by patch cords, transmitting messages and data across discrete components. These messages could take the form of control signals or discrete triggers—"bangs"—that marked a change in state at a specific logical time (27). By distinguishing between the transmission of data and the instantiation of events, Puckette laid the foundation for systems that handle both time-based reactivity and structural reconfiguration.

Equally significant is Puckette's sensitivity to cultural assumptions embedded in interface design. He deliberately avoided implementing a conventional score editor, recognising that such a move would have required encoding the norms of Western classical music—staffs, clefs, pitch names—into the software's core architecture. Doing so, he argued, would have restricted its usefulness to a narrow subset of musical cultures while undermining the experimental potential of the electronic medium (26). The resulting environment was rather not a tool for notation, but an instrument for action. This concern with minimal, culturally agnostic design shapes Puckette's broader philosophy of feature selection. He posed a deceptively simple question: What is the smallest set of features that would allow the widest range of musicians to achieve their aims? (Puckette, 2022, 24). This approach centres on designing for generality through modular abstraction. It offers a powerful model for thinking about compositional systems that reach beyond the specific demands of real-time synthesis. A well-formed environment begins with constraints. And these constraints function as the necessary boundary conditions within which experimentation gains coherence and depth.

Over the decades-long evolution of Max and Pure Data, additional capabilities were layered onto the foundational architecture. The integration of audio signal processing into the same visual patching language, for instance, introduced new expressive possibilities but also surfaced tensions between different paradigms of computation—between sample-accurate signal flow and message-driven event handling (27). Such interfaces between modalities reveal how structural clarity and usability must be carefully negotiated in any compositional environment. For developers of more speculative systems, this is an invitation to design interfaces that are both conceptually legible and operationally precise—particularly where different logics (such as symbolic, procedural, and economic) converge.

A further insight from Puckette's account concerns the nature of software as a medium-specific infrastructure. From the perspective of computer science, both Max and Pure Data qualify as "application-specific graphical programming environments for real-time multimedia" (Puckette, 2022, 28). But what distinguishes them in practice is the degree to which they allow music to be constructed not just with code, but through interaction with a persistent and reconfigurable logic. What matters is not only what the program does, but how it frames the relation between gesture and response. Any system that aspires to similar flexibility must also contend with a

familiar design dilemma: balancing ease of use with structural neutrality. As Puckette observes, software designers constantly face trade-offs between streamlining the interface and leaving compositional choices fully open. By introducing shortcuts—predefined templates that stand in for longer sequences—the software inevitably "nudges" the user toward certain conventions or patterns, privileging some musical outputs over others (28). Even small interface decisions, such as naming a pitch "C3" instead of 261.62 Hz, can anchor the user in culturally-specific assumptions. At the same time, avoiding such conventions entirely may make the software less accessible to those unfamiliar with its foundational abstractions.

Puckette's reflections thus highlight a central tension: any design that aspires to neutrality may still embed forms of exclusion, not through cultural dominance but through the technical thresholds it imposes. As he notes, Pure Data's decision to avoid simplifying shorthands made the software less culturally prescriptive, but also more difficult for users without a strong background in mathematics or signal theory (28). The cost of openness was a steeper learning curve. For designers of compositional systems today, this raises the question of how to maintain both epistemic clarity and broad accessibility. A possible response lies in modularity—not only in terms of functionality, but in how meaning is distributed. If the core compositional layer remains abstract and operable across diverse use cases, the expressive specificity can be layered on top, delegated to interfaces, visualisations, and performance environments. Such a design refrains from dictating how music should be represented or performed, while still enabling structured experimentation and contribution. The compositional act then becomes a negotiation between abstraction and instantiation, between infrastructural logic and contextual expression.

In thinking about scale and expressivity, Puckette offers another heuristic. He proposes that we imagine any given music software as a system that maps a bounded set of user gestures—defined by the time and effort available in a studio session—onto a corresponding space of possible outputs (28–29). The goal is to design this mapping such that minimal input variation yields a maximally rich set of outcomes. But this goal introduces its own contradiction: increasing expressive yield often requires predefining how certain inputs behave, and this reintroduces bias. "It is hard to imagine in this context", Puckette writes, "a way in which the expressive power of an input language could not be a manifestation, at bottom, of the designer's preconceived and presumably heavily biased notions of what good music is" (29). In other words,

automation—no matter how generative—carries the trace of aesthetic preselection. Rather than viewing this as a problem to be solved, it may be more productive to treat it as a condition to be managed. Systems that are open to compositional extension—where users can contribute, discover, and combine modular operations—allow aesthetic assumptions to become distributed. Each module reflects the perspective or intention of its author, but the system as a whole does not mandate a unified aesthetic framework. It becomes a field of possibilities, not a channel of constraint. Automation in such systems supports expressivity without encoding a single vision of what that expressivity should produce.

This brings us to one final, pragmatic concern: long-term viability. How can a piece of music software—or a compositional system more broadly—ensure that the works created within it remain performable across time? Puckette challenges the fatalism often associated with technological obsolescence. "Though some technologies do indeed fall out of use and availability", he writes, "others get frozen in place" (29). His example of the QWERTY keyboard—suboptimal yet durable—demonstrates that persistence is not necessarily tied to elegance. It is bound to ubiquity, standardisation, and infrastructural entrenchment. From this, Puckette draws an important lesson: future-proofing is achievable when systems are designed with long-term readability and executable coherence. In his view, Pure Data supports this kind of durability. It enables composers to produce "archival music documents", sacrificing some ease-of-use in favour of legibility, modularity, and transparent execution (30). While such systems may demand more from their users upfront, they offer resilience over time—a continuity not of interface or media format, but of operational intelligibility.

This perspective reorients our understanding of what compositional software must do. It should not only allow for experimentation in the present, but also preserve the intelligibility of those experiments in the future. It should support complexity, but not at the expense of clarity. And it should invite collaboration across boundaries of fluency, background, and expertise, not by hiding its logic, but by making that logic accessible, modular, and reusable. For an abstract machine to support new compositional ecologies, it must build on lessons drawn from the practical architectures that preceded it. From Puckette's writings, we inherit a set of durable imperatives to be added to the abstract diagram of the ARMM being here constructed: design minimally; avoid unnecessary prescription; distribute complexity; preserve intelligibility; and remember that compositional infrastructure is not a neutral

medium—it is a space where agency, culture, and thought are configured.

Intelligent Cultural Commons

The concept of the platform has undergone a profound transformation over the past decades, evolving from a political metaphor into a structural model for collective production and distributed agency. As Olga Goriunova notes, the term originally referred to a "program, an outline of theories or beliefs" shared among a group (2012, 8). In contemporary usage, however, it has come to signify a set of shared resources—material, organisational, or intentional—that inscribe practices and enable collaboration, production, and the potential for transformation. In this expanded sense, the platform is a socio-aesthetic construct: a machinic infrastructure capable of assembling publics, coordinating contributions, and fostering differentiated modes of engagement. Especially in the domain of art, platforms have played a central role in cultivating alternative publics—networks of geeks, amateurs, and artist-technologists whose collective activity constitutes a political as well as aesthetic formation. These publics participate in the making of new apparatuses, new circuits of enunciation, and new models of relation (Goriunova, 2012, 110). Such an understanding of the platform as an aesthetic and political machine reinvents how components are assembled, how knowledge is shared, and how futures are collectively imagined. Art platforms, as Goriunova argues, are capable of enacting radical aesthetic practices while remaining open to divergent political and economic formations. "The same processes to be found in art platforms", she warns, "can be operated upon by different interests" (2012, 111).

As the software paradigm has shifted away from static products toward services, infrastructures, and ecosystems, new models of cultural production and value generation have emerged. The rise of open-source practices, peer-to-peer infrastructures, and participatory coding communities has played a critical role in this shift. As Christoph Neubert—media and cultural theorist specialising in media history and infrastructure—observes, open-source software—especially systems like Linux—has undermined the product-based software economy of the 1980s, ushering in new strategies of value creation based on contribution, recognition, and infrastructural control (2015, 28). These dynamics are echoed in artistic contexts, where creators often develop software tools not only for personal use, but for collective benefit. As Miller Puckette reflects, motivations for writing such tools range

from the pursuit of reputation—"to be acknowledged as having done something cool"—to pragmatic economic incentives (2022, 31–32). Yet the enduring appeal of open-source systems like Pure Data lies in their epistemic transparency and infrastructural durability. Because the code is accessible and modifiable, composers can trust that their work will remain legible and executable across time, even if institutional or commercial support disappears (30).

In the context of these shifting infrastructures and socio-technical formations, Martin Zeilinger introduces the concept of the posthumanist cultural commons as a critical alternative to both private ownership and the liberal ideal of the public domain. Zeilinger reframes the commons itself as a posthuman ecology—one that emerges from and is co-constituted by posthumanist agential assemblages, in which human and non-human entities operate together in entangled, co-creative relations. "What is needed", he argues, "is a posthumanist cultural commons that is co-constituted and co-determined—in a spirit of unownability rather than of property-to-be—by the works and the workings of the posthumanist agential assemblage" (Zeilinger, 2021, 157). This proposal does not simply call for AI outputs to be released into the public domain. On the contrary, Zeilinger warns that the public domain, as conventionally construed, often functions as a preparatory space for further enclosure under anthropocentric intellectual property regimes. "The public domain", he writes, "does not strike me as a viable ecology for accommodating the works of posthumanist agential assemblages, in part because it effectively frames its contents for assimilation into the anthropocentric IP enclosures into which it feeds" (166). In simple terms, if something is available for free and to everyone, it potentially becomes a resource to be further incorporated into the schematics of proprietary regimes. Instead, Zeilinger proposes a more radical gesture: a commons grounded not in ownership or appropriation, but in unownability.

Such a commons must satisfy at least two criteria. First, it must persist beyond assumptions of the centrality of human agency; and second, it must be capable of resisting assimilation into conventional IP frameworks (Zeilinger, 2021, 169, 171). This is a fundamentally ecological model of commoning—an ecology not of natural resources, but of practices, systems, and relations. Drawing on Merima Bruncevic's legal philosophy, Zeilinger describes the commons as "an in-between or the entanglement of personhood-property space" (168)—a space that does not clearly separate human from non-human, intellectual from material, or producer from

product. In this configuration, the non-human is not simply part of the commons as resource; it becomes an integral agent in the production, maintenance, and transformation of the commons itself (169). Posthumanist agential assemblages—such as AI systems engaged in creative processes—thus disrupt the foundations upon which copyright and ownership have historically been justified. These systems operate through distributed configurations of data, code, networks, and training sets, all co-constituted by human and non-human inputs, rather than conforming to the image of a unified, intentional author. Zeilinger's analysis of GAN systems (Generative Adversarial Networks, widely used AI models) exemplifies this logic: framed as "bodies without organs", these systems engage in expressive processes that cannot be attributed to a single agent or mapped onto conventional aesthetic categories. Instead, they represent "mutual constitutions of entangled agencies" (Barad, 2007, 33, in Zeilinger, 2021, 160), in which meaning and creativity emerge from the interplay of divergent forces (Zeilinger, 2021, 160).

In Zeilinger's view, then, the emergence of the posthumanist cultural commons requires a new legal framework and a new aesthetic and infrastructural imagination. The commons must be reconceived as a site of commoning—an ongoing, performative process in which agency, value, and relation are continually reconfigured. In this sense, commons are not merely resource pools or legal categories; they are machinic ecologies, dynamic assemblages in which subjects and objects, authors and infrastructures, become indistinct. As Zeilinger writes, "the emergence of the posthumanist assemblage… hinges on a radical rethinking of what property means and how it operates, what we mean by cultural ownership, by creativity, by calling something a creative expression" (173). This is not a call to abandon ownership altogether, but to rethink its basis in a world where creativity is no longer confined to the human subject.

To conceive of a commons adequate to the complexity of posthumanist assemblages, a rethinking is needed of how intelligence itself can be organised—beyond individual cognition or machine learning models—through shared infrastructures of data, participation, and reflection. In this light, the posthumanist cultural commons becomes an ecology of shared works and processes, and a substrate for collective intelligence: a mode of cognition distributed across human and non-human agents, platforms, and artefacts. This vision challenges the individualist paradigm of intelligence that has long structured technological imaginaries. As Gilbert Simondon

noted in the 1950s, modern technics has often been driven by the desire to outsource volition and knowledge to machines in pursuit of power and mastery. "The man who wants to dominate his peers calls the android machine into being", Simondon writes, describing a fantasy of delegation whereby the individual retreats behind a constructed apparatus, "freed of all danger, exempt from all feelings of weakness, and triumphant through the mediation of what he [of course] invented" (2017, 16). Yet Simondon also suggests that this desire need not be pathological if it is reoriented: rather than building machines for individual supremacy, we might build them for collective insight—machines that support processes of co-individuation rather than concealment.

Such an orientation finds support in the evolution of the web toward semantic and relational architectures. As described by Tim Berners-Lee—computer scientist and inventor of the World Wide Web—the development of a machine-understandable web was envisioned as enabling "intelligent agents" capable of analysing not only content but also the links and transactions between people and machines (Berners-Lee 2000, in Hui, 2015, 157). This vision is one of networked cognition, where the intelligent agent is a node in a semantic field of meanings, actions, and contexts. As Yuk Hui notes, digital objects serve a critical role in this ecology: they allow agents—human and machinic alike—to process and relate information automatically, creating a symbolic infrastructure for collective sense-making. In such a system, data is an operational field: a space of relations from which new understandings emerge. From this perspective, the commons besides collecting data, enables it to become intelligent through participation. This form of intelligence builds itself from the strata of interaction—structured, accessible, and re-usable data that reflects the ongoing actions of a networked community. A "collective intelligence" does not necessarily emerge through machine training, but through the active participation of various actants in a shared infrastructure. It is closer to symbolic than connectionist AI: not predictive modelling but procedural insight. Not probabilistic guesswork, but collective compositional reflection through participation.

In such systems, the intelligence is not located in any one agent or algorithm, but in the entire infrastructure—the platform, the contributors, the encoded transformations, and the evolving database of actions. Each contribution to the system—whether a compositional operation, a metadata annotation, or a code modification—becomes part of a growing ecology of meaning. As Future Art Ecosystems 4 notes,

organisations in such systems become "interfaces between data producers and AI models" (2024, 46). But when reframed in light of symbolic logic, this interface does not simply train a model; it becomes itself a space of epistemic articulation—a machinic commons through which a network reflects on and transforms itself. The data produced by participants thus becomes a form of shared cognition, through which the community can generate "a richer level of insight into how [it] functions, and what it can infer about its own operations" (46). This is the potential of an intelligent cultural commons: a system that not only stores and distributes creative content, but one that enacts a recursive relation between participation and understanding. It offers not simply access, but agency—not only archives, but operational diagrams. And because each act of contribution adds to the system's epistemic topology, it becomes possible to think of collective intelligence not as a model, but as a process. The commons, in this light is the infrastructure for a new kind of thinking, a machinic ecology whose insights emerge from relational depth of encoded interactions between various actants.

To realise the vision of collectively-intelligent posthumanist cultural commons, such systems must be grounded in infrastructures that are both technically robust and philosophically and politically aligned with the distributed nature of agential creativity. The durability of software lies not in its programmability alone but in its insertion into a broader ecology of social, economic, and cultural intention (Erickson and Kelty, 2015, 42-43). Software systems must be designed with time in mind, not just space; with duration, not just code. This temporal embedding is essential for ensuring the long-term viability of cultural systems that seek to be open, generative, and participatory. Blockchain technologies—particularly when framed beyond speculative finance—offer a promising substrate for such infrastructures. One of the main observations of this doctoral dissertation in this context is that blockchain technologies, in this view, are not simply ledgers serving as a new distributed medium—they are enablers of distributed compositional infrastructures that align with the ethos of the intelligent posthumanist commons.

Blockchain as the Art's Medium for Composing Agencies

As noted by Marcus O'Dair (2024), blockchain enables mass collaboration while also allowing for differentiated systems of economic reward (79). More than royalties and

ownership splits, smart contracts can also encode conditions of access, execution, and recomposition. This opens up the possibility for commons, where certain features or artefacts are free, while others are monetised or conditionally accessible—not to privatise the commons, but to sustain its infrastructure through transparent, programmable exchange. Crucially, this model resists the regressions of "competitive individualism" that Ruth Catlow and Penny Rafferty (2022) identify in the NFT space, instead promoting DAO-like models that frame culture as a collaborative and programmable ecology of participation (emphasis of O'Dair, 2024, 79).

In this context, blockchain could be re-envisioned for the arts as a medium for composing with agencies. The posthumanist cultural commons, instantiated through smart contracts, could become a generative infrastructure for creating with and across agential assemblages. These assemblages would include artists, audiences, AI models, algorithms, economic conditions, semantic metadata, and executable procedures. Compositional practice would operate at the meta-level of agency itself—not composing artworks, but constructing systems that generate, transform, and interrelate procedures capable of generating artworks. A network in which not the artistic outputs but the relations between the artistic procedures that produce them persist would resist the commodification of art, because its value lies in the artistic processes themselves and, through them, in the participation of the actants who perform them. Thus, such compositional ecology would not oppose economy—it would be economic through participation, where the value lies in the act of performing rather than in the speculative bubble surrounding any single artistic object (precisely because these would not be stored in the network). It would be transactional by design. Every operation would be an exchange: of data, value, energy, or attention. This way, it would allow for the construction of self-regulating systems that embed economic logic into aesthetic and epistemic processes. The cultural commons, in this model, would be neither free nor owned, but programmable. Smart contracts would first of all encode how artefacts behave and what are the conditions to execute this behaviour, rather than primarly specifying who owns them. Ownership would be reframed as participation in a dynamic system of co-individuation. Such an infrastructure would make the speculative potential of Zeilinger's posthumanist cultural commons real by giving it operational form. It would provide a technical substrate for a new class of artworks: not representations, but autopoietic machines. Not outputs, but systems of individuation. And within these systems, the act of composing would become a practice of diagramming the real—not by modelling it,

but by setting in motion a field of relational operations whose outcomes are emergent, distributed, and ontogenetically open.

At its core, blockchain provides a Turing-complete execution environment that is stateless between runs: each transaction executes in isolation and only updates the blockchain's persistent global state if it explicitly writes data to it. Any such update—such as adding a new transformation to a compositional system or changing a condition of its execution—requires a transaction and the payment of a gas fee[7]. By contrast, once a smart contract is deployed, functions that do not modify state can often be called for free when executed off-chain. Retrieving metadata, running a view function, or accessing an existing compositional feature incurs no cost because these operations leave the state untouched. Adding a new operation, minting a feature, or contributing a subcomponent does alter the state and must be recorded on-chain. This incurs a cost not because the system is commercialised, but because computation, verification, and consensus on the blockchain consume scarce network resources.

This difference is fundamental. Calling a function without changing its internal state is free (for instance to generate some artistic output); changing the system (such as making someone owner of something) is not. In compositional terms, this means the ARMM could serve as a programmable infrastructure for collective creativity. It would support both read-only epistemic exploration (e.g. playing back, querying, or recombining existing compositional elements) and write-enabled agential operations (e.g. contributing new operations, features, or reconfigurations). Once open-sourced and deployed, the machine operationalised by the diagram of the ARMM would persist independently of its creator, accessible to other agents and capable of being recursively extended. Its structure would become a commons, not by virtue of being "public domain" in the legal sense, but because it would exist as a shared executable environment governed by compositional logic rather than private ownership. Importantly, while certain aspects of the system could be monetised (such as contributing new modules, minting new versions, or algorithmically restricting access to the execution of smart contracts), the base-level access to its structure would remain free (and let's remember that all code on the blockchain is principally open-sourced anyway). This balance would enable an economy of contribution rather than

[7] On blockchain networks, a small payment called a *gas fee* is required to perform any action that changes the state of the virtual machine, such as sending a transaction. This fee is paid in the network's cryptocurrency and compensates the computers that process and verify the action.

consumption. As long as the infrastructure is designed with modular openness and semantic transparency, it becomes possible to construct complex compositional ecosystems where aesthetic, epistemic, and economic operations coexist without collapsing into extractive logic.

What emerges from the considerations above is a diagrammatic vision of an infrastructure designed to condition its ongoing emergence through structured participation. The abstract machine described so far describes a compositional substrate that would be:

1) durable and transparent, ensuring legibility and reusability through accessible code and persistent structures,
2) open and modular, allowing contributors to access, recombine, and extend existing operations,
3) economically sustainable, embedding transactional logic without subordinating creativity to commodification,
4) epistemically generative and infrastructurally self-extending, enabling contributions that reorganise the system's own operational logic and expand its capacity to know, and capable of supporting recursive compositional acts that reconfigure not only outputs, but the system of composition itself,
5) collectively intelligent, integrating inputs from human and non-human agents into a shared ecology.

In this way, the machine sketched so far provides the foundational guideling for a new class of creative environments: living, procedural systems that cultivate the ongoing individuation of collectivelly intelligent, aesthetic, epistemic, procedural and relational forms.

Rhizomatic Metamodelling

What kinds of processes would the machinic assemblages envisioned in the ARMM enact? How would it individuate compositional structures, and at what level of abstraction would it intervene? What are the procedures that this machine would contain as a new kind of intelligent cultural commons? What processes encoded in the blockchain infrastructure would it need to consist of to enable the described new kind of compositional framework? Before the technical operationalisation of the principles

diagrammed so far as the Autopoietic Rhizomatic Metamodelling Machine, it is important to envision the principial processes through which such operationalised machine would operate. Thus, this section introduces the abstract process of the abstract machine: *rhizomatic metamodelling*, as a proto-concept to the technical processes described in the next chapter as Performative Transactions.

A foundational idea comes from Iannis Xenakis's canonical account of the compositional process, as outlined in *Formalized Music*. There, Xenakis distinguishes eight sequential fundamental phases involved in the production of a musical work. His articulation provides a rare and valuable schema for distinguishing between conceptual, operational, and performative layers of compositional labour:

"1. Initial conceptions (intuitions, provisional or definitive data);

2. Definition of the sonic entities and of their symbolism communicable with the limits of possible means (sounds of musical instruments, electronic sounds, noises, sets of ordered sonic elements, granular or continuous formations, etc.);

3. Definition of the transformations which these sonic entities must undergo in the course of the composition (macrocomposition: general choice of logical framework, i.e., of the elementary algebraic operations and the setting up of relations between entities, sets, and their symbols as defined in 2.); and the arrangement of these operations in lexicographic time with the aid of succession and simultaneity) ;

4. Microcomposition (choice and detailed fixing of the functional or stochastic relations of the elements of 2.), i.e., algebra outside-time, and algebra in-time;

5. Sequential programming of 3. and 4. (the schema and pattern of the work in its entirety);

6. Implementation of calculations, verifications, feedbacks, and definitive modifications of the sequential program;

7. Final symbolic result of the programming (setting out the music on paper in traditional notation, numerical expressions, graphs, or other means of solfeggio) ;

8. Sonic realization of the program (direct orchestral performance,

manipulations of the type of electromagnetic music, computerized construction of the sonic entities and their transformations)."

— Xenakis (2001/[1963], 22)

This breakdown offers a rigorous scaffold for distinguishing between levels of musical abstraction, from the intuitions that initiate composition (1–2), through the symbolic and algorithmic transformations that structure it (3–6), to the notational and sonic actualisations that render it performable (7–8). What interests us here is not the entire arc from concept to sound, but the operational middle ground—phases 3 to 6—where the compositional system is defined, structured, and internally validated. These are the domains of metamodelling.

Rhizomatic metamodelling, as envisioned in the ARMM, does not model sound directly. It models the space of possible transformations that structure sound-producing systems. The model is not a representation of music, but a scaffold of operational potentials—a space of transductive relations capable of generating a diversity of musical behaviours. This is why Xenakis's third through sixth phases are especially pertinent: they concern the definition, programming, and iterative reconfiguration of transformations before the work becomes fixed into notation or sound. These phases do not yet belong to the work as a musical object, but belong to what might be called the abstract diagram of compositional individuation.

To articulate this operational logic, we must clarify what is meant by metamodelling. The term is used here in two specific and complementary senses: one drawn from Félix Guattari's *Chaosmosis* (1995), the other from computer science. In *Chaosmosis*, Guattari defines metamodelisation (Guattari, 1995, 21-22, 58-69, see also Watson, 2009, 10-12) as a pragmatic method for mapping how subjectivity is produced across heterogeneous systems—social, semiotic, machinic, affective. For Guattari, subjectivity is not a given, fixed entity (like "the self" or "the individual"). It is produced—emerges through relations between bodies, technologies, signs, institutions, affects, and practices. In *Chaosmosis*, he writes that subjectivity is machinic and processual, formed at the intersection of multiple heterogeneous systems: social, psychological, biological, technical, and aesthetic. To model how subjectivity is produced means to: map the interacting forces that shape a subject (e.g., media, language, work, environment), trace how a subject's identity, perception, or agency

emerges over time, not explain who someone is, but how different processes make someone who they are. Unlike fixed models, metamodels are open-ended, partial, and functional. They are tools for diagramming processes of individuation, not for representing stable forms. Metamodels remain modifiable and situated, adapting to shifting creative or existential contexts. For Guattari, they operate on the level of processual consistency, not truth or completeness.

In software engineering, a metamodel is a model that defines the structure of other models—a schema that describes how components can be composed, validated, and transformed. For example, in visual modelling languages like UML, the metamodel defines the rules by which diagrams are constructed and interpreted. This allows for interoperability, modularity, and introspection within complex systems. Metamodelling means creating a model of a modelling language—a model that defines how other models are structured, validated, and interpreted. If a model represents a specific system (like a diagram of a software app or a musical transformation), a metamodel defines the rules and components allowed in that diagram: What kinds of elements can exist? How can they be connected? What properties must or can they have?

This aligns the ARMM's function with a long-standing philosophical concept: Kant's notion of the *schema*. As Yuk Hui (2015) observes, the Kantian schema mediates between the abstract concept and the concrete intuition, functioning as a temporal operation that enables general forms to be instantiated in particular ways. Crucially, Hui reframes the schema not simply as an act of imagination (as in Kant's first *Critique*), but as a technical standard that ensures interoperability across instances. As he puts it: "Schema is thus a standard that assures identity across several instances of an object… Schemas also become the base of the production of relations, and allow them to be more explicit… [they] are used widely in the design of digital objects: known as ontologies, they are the structures that produce, reproduce, and model objects" (Hui, 2015, 9).

In this light, the machine envisioned in the ARMM can be seen as a machine for composing schemas—not digital objects themselves, but the generative scaffolds through which such objects can emerge and relate. These schemas are not symbolic representations; they are operational configurations that modulate the emergence of artworks, performances, features, and processes across a distributed compositional system. Here, the notion of schema becomes a conceptual hinge between Kantian

transcendental philosophy, Guattarian diagrammatics, and the ontological infrastructure of digital art. Agents within the envisioned machine construct generative schemas—compositional metamodels that define procedures, transformations, and relations. These are open, interoperable configurations: they can be detached, recombined, or recontextualised into new assemblages. Schemas are not only constructed within the system—they also construct the system itself. They recursively generate the very ecosystem that supports them.

However, while the schemas produced by agents are generative, the machine itself is not generative in the conventional sense. Its logic is not to produce artefacts (as it also doesn't start from nor at any point contain any objects in the first place), but to support their production through the configuration and reconfiguration of plain operations. In this respect, the envisioned machine is not a generator—it is allagmatic. Its space of operations can be performed on any set of objects and it is independent from the objects themselves. It orchestrates reconfigurable transformations, structuring the space of possible operations rather than producing any fixed outcome (from any set of previously known inputs). The agents that operate within it may be generative, but the machine they inhabit is allagmatic and autopoietic—constantly folding operations back into its own topology, allowing it to evolve without predefining its direction. To return to Xenakis: phases 3–6 are where schema and metamodelling converge. They represent the space of procedural experimentation—where operations are tested, sequenced, parameterised, and recursively adjusted. The abstract machine is then a system that conditions the emergence of musical systems.

If the machine envisioned by ARMM intervenes exclusively within phases 3 to 6 of Xenakis's schema, its operational field remains strictly confined to the abstract layer of transformations, structural mappings, and procedural recombinations. It does not concern itself with the origins of sonic material (phases 1–2) or with its final representation or sounding (phases 7–8). This raises a critical question: what is this machine modelling, if not musical material itself? And how do the operations within the machine relate to the musical works, processes, and events that populate our lived experience of composition? A powerful lens for answering this comes from Paulo de Assis's extension of Timothy Morton's theory of hyperobjects into the domain of music. As Morton argues, hyperobjects are entities that cannot be fully grasped at any single moment; they are massively distributed in time, space, and relationality (Morton, 2013, 1–4). They are never directly accessible—only their local manifestations

are. And yet, they persist and exert effects across networks of entangled interactions. One prominent example throughout Morton's book is the hyperobject of global warming, which cannot be grasped as one single object, but can be measured, proved and certainly exists, occurring in various contingent phenomena. Morton notes that hyperobjects are not "collections" or "systems" of smaller objects. They are entities in their own right, irreducible to their components, and radically nonlocal. Their structure is always withdrawn, only ever encountered through aesthetic interference patterns, or what he calls interobjectivity (2).

De Assis applies this logic to musical practice, proposing that musical works could be understood not as bounded objects, but as hyperobjects—or more precisely, as hypermusic (de Assis, 2024, 18–20). Hypermusic consists of distributed networks of performative acts, conceptual decisions, latent potentials, and historical sedimentations. It is never entirely given. It manifests only partially through notations, performances, interpretations, and analyses. And critically, new musical assemblages always emerge from existing ones. Unlike in generative music, where a self-contained system produces a work from internal rules, hypermusical practice proceeds through the reconfiguration of a network of *existing* materials, processes, and relations. This is a crucial distinction for situating what would the envisioned machine metamodel. In generative composition, the system is autonomous: it creates outputs from within its own rule space. In interactive composition, the system responds dynamically to external stimuli—usually in real time—but the underlying structure remains fixed. By contrast, the machine being envisioned here would transform the operational framework itself. It would not just enable variation in behaviour; it would enable variation in the conditions of variation. Each execution of the system would become a transductive event—a moment of individuation that reconfigures not the output, but the structure that enables outputs.

From this perspective, the ARMM does not diagram principles for operationalising a generative engine nor for operationalising an interactive interface. It is envisioning an allagmatic machine (in a sense described in the previous chapter) that would metamodulate the operational relations between *existing* compositional elements. But—and this is the key point—these elements would not be contained within the machine. The machine would operate only on numerical indexes symbolising objects and further on indexes referring to other indexes in an intricate network of metamodelling processes, not on objects themselves. Its domain would be purely

procedural. The particles of music—the recordings, gestures, timbres, samples, motifs, scores—would all exist outside of the machine. They correspond to Xenakis's phases 1–2 (intuitions and definitions of sonic entities) and phases 7–8 (notation and sonic realisation). The envisioned machine would neither define nor produce them. It would diagram how they might relate, recombine, or transform one another. It would build metamodels—abstract operational schemas through which external musical materials could be rearranged only by linking them back to the indexical values they were previously assigned.

In this sense, the envisioned machine's role would be composing the conditions under which music can individuate. The musical work is a hyperobject—a distributed, emergent field whose local manifestations (scores, performances, features) always exceed their encoding. The envisioned machine would operates within this field, not by shaping its objects, but by diagramming its conditions of transformation. It would be a platform not for outputs, but for structural individuation. The works that would appear around it would not be its contents. They would be its external expressions—its local, contingent, and partial manifestations. Campbell (2013, 148) describes a similar process in Deleuzian music practices, noting that the most compelling composers today do not operate within fixed symbolic grammars, but instead "molecularise, deconstruct or dissolve existing musical material", generating new semiotic units through hybridisation and technological experimentation. This is precisely the operational space of the machine envisioned in ARMM. It would be a machine not for preserving meaning, but for reorganising expressive potentials. Its allagmatic capacity would reside in its ability to reconfigure the relations among already-fragmented elements, allowing for emergent musical assemblages without ever prescribing their form.

The capacity to support recursive operations—where transformations can act not only on musical material but also on other transformations—would position such an abstract machine not as a tool for composition but as a platform for meta-composition. In this sense, it would enable what Sofian Audry and Jon Ippolito describe as the meta-artist function: the authorship not of an artwork, but of a system that can itself produce authors. The meta-artist, following Annemarie Bridy's definition, is "the author of the author of the works" (Audry & Ippolito, 2019, 3). This recursive layering of agency directly parallels the logic of metamodelling: the machine envisioned in ARMM would compose not musical works, but compositional agents and structures capable

of producing them. Within this machine, every new transformation or feature would be both a tool for composing and a model of how composition itself can unfold. The machine would enable contributors to write new compositional behaviours (e.g., a transformation module), but also to define new types of transformations (e.g., a transformation of transformations). In doing so, contributors would occupy a meta-compositional role, shaping not outputs but the grammar of future outputs.

Audry and Ippolito argue that such recursion is perceptually framed by the receiver, who decides whether to ascribe authorship to the artwork, the artist, or the meta-artist depending on the conceptual distance they choose to traverse. The same holds in the machine envisioned in the ARMM: some may perceive a musical output as authored by an individual agent; others may trace it back to the underlying schema, infrastructure, or ecosystem of operations. In this framework, metamodels become epistemic scaffolds, not only for compositional action, but for the negotiation of agency and authorship across a distributed system. By enabling agents to define the structure of compositional interaction itself—not only what is done, but how doing is defined—the envisioned framework would become a machine for composing composition. This is its metamodelled, autopoietic core: a system that writes its own operational grammar through the contributions it hosts. It would not contain works—it would condition the emergence of worlds of works. And in doing so, it would fulfill a speculative compositional function: not being a tool for making music, but a platform for rewriting what it means to make music at all.

The recursive architecture of the envisioned machine, in which compositional operations define not only musical materials but also the conditions of their transformation, resonates strongly with Xenakis's vision of the evolving role of the composer. Already in *Formalized Music*, Xenakis identified a paradigmatic shift: the composer is no longer simply a shaper of forms, but a designer of schemes—formal, abstract systems for generating sound structures: "We may further establish that the role of the living composer seems to have evolved, on the one hand, to one of inventing schemes (previously forms) and exploring the limits of these schemes, and on the other, to effecting the scientific synthesis of the new methods of construction and of sound emission" (Xenakis, 2001, 133).

In this formulation, the composer becomes a meta-composer—an agent who no longer composes works directly, but composes the conditions for compositional emergence. These conditions take the form of logical frameworks, mathematical processes, and

experimental configurations that are treated algorithmically and explored through computation. Xenakis names Poisson and Markov processes, probabilistic constraints, and musical games as early examples of this procedural logic, already lending themselves to computer-based implementation (133). In the envisioned machine, this logic would be extended further: the system would support not only compositional processes but compositional processes about compositional processes. In other words, it would enable users to build schemes that themselves generate or transform other schemes—exactly the recursive condition of metamodelisation.

Xenakis's broader conceptual project—particularly his vision of metamusic—provides another useful frame. In his later work, he imagines a universal musical framework encompassing all musical traditions and formal possibilities, grounded in a "general harmony" and operationalised through the theory of sieves (180). The sieve, in Xenakis's use, is a modular, arithmetical mechanism that filters integers into musical intervals, scales, or rhythmic patterns. Crucially, sieves can be added, subtracted, nested, and recombined, making them not only generators of structure but, crucially, operators on other operators. This transforms them from tools of music-making into tools of metamodelling—machinic schemas that modulate how composition itself unfolds. It is precisely this logic that the machine envisioned here would generalise. Within the machine, all transformations—whether stochastic, deterministic, or combinatorial—would be treated as operable entities that can be chained, edited, duplicated, or composed into higher-order behaviours. Like Xenakis's sieves, they would be the operational primitives for diagramming and transforming the compositional space. Even Xenakis's metaphysical reflections echo this procedural ontology. Late in *Formalized Music*, he suggests that space may be nothing more than "an appearance of […] chains of energy transformations" (257). This notion aligns with the ARMM's view of the compositional system as a topology of indexed operations, where every feature, transformation, or schema is not a static entity but a node in a dynamic network of relational energies. In such a system, no singular object is final. Everything is an event in a broader field of metamodelling.

Social Complexity of the Autopoietic Rhizomatic Metamodelling Machine

The procedural relations between various compositional elements described as parts of the abstract machine envisioned in this chapter so far, should at this point form a

blurred picture of a compositional system in which various human and non-human actants co-constitute. A system that is capable of reorganising its own principles through the actions of its agents, who in response acquire access to a broader collective intelligence resulting from the commons of compositional processes. In that context, it is useful to sharpen this picture further by discussing the social level of such an assemblage.

Assemblage theory, particularly as articulated by writer, artist and philosopher Manuel DeLanda (2006), provides a framework for analysing complex, multi-scalar systems composed of heterogeneous elements—systems in which parts maintain operational independence even as they participate in larger configurations. These insights are especially pertinent when designing a compositional infrastructure intended to be both decentralised and recursive: a system that must support individual creative acts while enabling their aggregation into durable, evolving structures. Crucially, DeLanda insists on understanding assemblages in terms of relations of exteriority. A component part of an assemblage "may be detached from it and plugged into a different assemblage in which its interactions are different" (DeLanda, 2006, 10–11). This principle sharply distinguishes assemblages from organismic or structuralist models, where parts are defined by their place in the whole and cannot function independently. For the ARMM, this distinction is foundational. The envisioned machine must accommodate compositional units that can migrate across projects, contexts, or communities without losing their internal coherence (similarly as patchers in MAX/MSP or PureData). Each contribution must remain legible and functional outside its initial instantiation, enabling reconfiguration.

In my previous research (Łukawski, 2023), this principle was articulated diagrammatically—envisioned as a method for modelling spatiotemporal change across levels of complexity, from individual assets to entire social networks. Such a diagram, understood in Guattarian terms as an "autopoietic machine", does not map interior relations but instead outlines the topological limits within which transduction can occur (Łukawski, 2023, 62–63; see also Guattari, as cited in Łukawski, 2023, 63). Diagrammatic thinking allows for procedural reasoning about multiplicities—configurations that unfold in time without becoming fixed into singular identities. In this context, the ARMM envisions a socially-dynamic infrastructure for processing heterogeneous contributions.

Assemblage theory also foregrounds the multiscalar nature of such infrastructures.

Social assemblages, according to DeLanda, range from individual bodies to buildings, neighbourhoods, cities, and nation-states, each with their own internal diagram and corresponding space of possibilities (DeLanda, 2006, 30). These spaces are characterised by degrees of freedom and universal singularities—features that define what kinds of events or transformations are possible within them. For the ARMM, this implies a layered structure: each compositional unit operates within a local topology of constraints, but these local operations feed into a broader ecology whose dimensions are defined not by central control but by distributed participation. This vision directly informs the need for machinic infrastructure that can accommodate radically different types of contributions—generated by human artists, AI agents, or procedural algorithms—and integrate them into coherent networks of transformation. In my earlier writing on the metaverse (Łukawski, 2023), I argued that both human and non-human agents could act as observers and participants in such a system, transducing themselves across dynamic topologies that resemble "interactive, constantly evolving, and self-composing virtual realities" (Łukawski, 2023, 63). The same logic applies to the machine envisioned in ARMM: each agent, whether embodied as an interface, a generative process, or a dataset, would participate in a field of compositional individuation governed by procedural rules rather than aesthetic norms.

DeLanda also re-introduces (after Deleuze and Guattari) the concepts of territorialisation and deterritorialisation to describe how assemblages stabilise and destabilise themselves through processes of spatial, symbolic, and functional consolidation (DeLanda, 2006, 13). For creative systems, this means recognising how compositional elements may temporarily cohere—forming genres, styles, or community conventions—only to be later reconfigured through new interventions. The envisioned machine must support this continual negotiation. Its design must allow for contributions to crystallise into recognisable structures, while preserving the ability for those same structures to be re-aggregated into novel assemblages. By situating the envisioned machine within the conceptual terrain of assemblage theory, we begin to understand it not as a singular machine but as a system of systems—an operational ecology in which each element retains autonomy while contributing to the emergence of a shared compositional environment.

Building on this understanding of the machine envisioned in the ARMM as a social assemblage, we can now turn to musical practices themselves. Edward Campbell's

Music after Deleuze (2013) provides a crucial perspective here, aligning Deleuzian thought with compositional methodologies. His reading of Deleuze and Guattari's work allows us to view artistic process as a rhizomatic process—a process that operates through transversals, detours, and cuts across established structures. Campbell identifies a key principle at the heart of Deleuzian aesthetics: every significant creative act is a response to a problem, not a refinement of a form (Campbell, 2013, 36). In this sense, creativity is diagonally constituted—it does not operate vertically within traditions or horizontally across contemporaneous styles, but obliquely, inventing a new line in a space that was previously unformed. Deleuze and Guattari borrow Pierre Boulez's term diagonal to name this transversal vector of creation. Each innovation traces a new line between pre-existing trajectories, resulting in a new assemblage, a new plane of consistency. The machine envisioned in ARMM must be structured to support precisely this kind of transversal operation: a compositional system in which each contribution not only adds to the network but configures a new vector through it.

Such transversality defines the rhizome. Unlike a tree, which branches from a single trunk and follows a hierarchy, the rhizome proliferates laterally, with multiple entry and exit points, no predetermined direction, and no unified structure (Campbell, 2013, 21, 37). Rhizomes are processes, not forms; they unfold in time through variation and recombination. As Campbell writes, they are "irreducible to prior or overarching identities", and they "pertain to a map that must be produced, constructed [...] and has multiple entryways and exits and its own lines of flight" (21). To the rules diagrammed in the ARMM, this implies a mode of operation that resists static classification. The envisioned machine must function as a diagrammatic machine whose paths are continuously drawn and redrawn, shaped by the flows of creative contribution rather than by adherence to fixed schemas.

In musical terms, Deleuze and Guattari contrast the arborescent tonal system with the potential of "generalised chromaticism"—a condition in which durations, timbres, intensities, and articulations are all placed into continuous variation (Campbell, 2013, 37–38). Under such conditions, music becomes a "superlinear system", a rhizome of sound rather than a tree of notation. The envisioned machine must accommodate this logic at the compositional level of the network: a space of continuous operational variation rather than discrete, rule-bound structures. Its function would not be to stabilise musical forms but to enable their ongoing differentiation. Rhizomatic

systems are not devoid of structure; rather, they construct their own immanent logics as they unfold—what Deleuze and Guattari call planes of consistency. The ARMM, too, generates its consistency through the accumulation and recomposition of its own operational strata (as various actors continuously contribute to the shared network).

Processes of deterritorialisation and reterritorialisation further illuminate this dynamic. Campbell notes that every diagonal, every creative transversal, functions by escaping a previous configuration—deterritorialising it—only to reassemble itself elsewhere as a new assemblage (Campbell, 2013, 39–40). This continual unmaking and remaking of form is central to both musical and philosophical creation. In this way, the envisioned machine should be understood as a field of operational deterritorialisations: each compositional act would subtract itself from an existing state and recompose the field it emerged from. It would produce not works, but vectors of transformation—lines that would remain open to new affiliations, and in doing so, they would reconfigure the very ecology that sustains them.

As Campbell explores through the work of Ivan Wyschnegradsky and Pierre Boulez, contemporary music has foregrounded the spatiality of sound, revealing not only its timbral or pitch structures, but its capacity to delineate new experiential spaces (Campbell, 2013, 69–70). For the envisioned machine, this spatiality is both literal and diagrammatic: the system would compose not only sonic artefacts but new territories of interaction and relation. Its diagram would not trace an a priori structure; it would map intensities, enable traversal, and manifest potential fields of transformation. In this context, musical time itself would become a variable in what Deleuze called a "cartography of variables"—a framework of continuous differentiation rather than punctual inscription (Campbell, 2013, 101).

To conceive the machine envisioned in the ARMM as a system capable of continuously transforming itself while maintaining compositional intelligibility, we must articulate the internal mechanics of its individuation. Here, the notion of parametrisation becomes crucial. As Liam Magee and Ned Rossiter—scholars of digital data infrastructure and labor—note, Manuel DeLanda has described assemblages as entities that individuate through the modulation of their parameters—adjustable "knobs" whose values configure the emergent identity of the system at any given moment. "Changing these values", they write, "alters the configuration of thought and practice" (Magee and Rossiter, 2015, 78). The envisioned machine, similarly, would not be defined by a fixed structure but by a field of potential operational states,

each shaped by the shifting relations. In this model, the artwork would not be the product of the system—but the system itself. Or more precisely, it would be the system as it expresses itself under particular parametric conditions. This view aligns closely with Gordon Pask's concept of behaviour, as revisited by Sofian Audry (2021). For Pask, behaviour is not a fixed output but a form: "an unchanging form of events due to the activity within an assembly" (as cited in Audry, 2021, 47). Though the system changes over time, its behaviour can remain recognisable as an invariant morphology. This definition is particularly apt for artistic systems, where the form of a work is often legible not as a discrete object but as a pattern of performance unfolding across space and time. Such behaviour is not strictly algorithmic; it depends on the perceptual and experiential encounter between system and observer. In this sense, the envisioned machine would be a machine of aesthetic transmission, wherein behaviours would be enacted and recognised within a shared environment of interpretation.

Audry further distinguishes between different orders of behaviour in artistic systems, drawing a line between stateless mappings and stateful agents. Stateless systems—like simple input-output mappings or traditional patch-based media artworks—lack memory and cannot evolve in time; they are "zero-order behaviours" (Audry, 2021, 49). In contrast, agent-based systems possess internal state, enabling them to modify their own behaviour in response to environmental interactions. These are "first-order behaviours", and when such systems become adaptive or generative—able to transform their own behavioural logic—they achieve "second-order behaviour" (50). Within the envisioned machine, these distinctions would not be separate layers but continuous dimensions of operation. The system must be capable of hosting mappings, agents, and meta-agents—operations that are not only performed but can perform themselves differently over time. Compositional features must not merely execute transformations; they must be able to transform the conditions under which they transform. In this framework, each contribution—whether a code module, a behavioural rule, or a performative condition—acts as a local expression of a more complex topological field. As Aaron Sloman—a philosopher and researcher on artificial intelligence and cognitive science—has proposed (1984), one can imagine a "space of possible minds", a multidimensional conceptual space in which different systems—human, machinic, hybrid—occupy various cognitive or behavioural niches. The envisioned machine would map a similar kind of space, though articulated in compositional rather than cognitive terms. It would not contain a singular logic of musical thought; it would comprise multiple, interacting logics that would co-

determine one another through the operations they would enact. Each compositional unit within this machine would thus correspond to a behavioural vector—a path through the space of possible operations, determined by its internal configuration and its relations to other units.

This ontogenetic model of composition draws directly from Simondon's theory of individuation. As Deleuze summarises, "the individual is not only the result but the element [milieu] of individuation" (2001, 43). The preliminary condition for individuation, Simondon argues, is the presence of a metastable system—one that contains internal tensions or potentials that have not yet resolved (44). This model allows us to understand compositional acts as events within an ongoing process of individuation—each one creating new topologies of relation. For Simondon, even the individual itself is a phase—"a phase of becoming that will lead to new operations" (Deleuze, 2001, 48). This emphasis on the processual, the transductive, and the relational defines the envisioned machine's function as a posthumanist compositional machine. Its operations would not be reducible to symbolic notation or signal flow; they would constitute topological events. They would enact individuation through their performative effect, structuring space and time through behaviour. As Deleuze notes in his review of Simondon, "a living being grows from both the inside and the outside", operating in topological contact with its environment (2001, 47). The envisioned machine must operate in this mode—at once internally differentiated and externally responsive, capable of inscribing its own transformations into the network it constitutes.

Autopoietic Rhizomatic Metamodelling Machine

This chapter has introduced and explored the Autopoietic Rhizomatic Metamodelling Machine (ARMM)—an abstract machine in the sense defined by Felix Guattari as a diagram that conditions the emergence of systems, behaviours, and relations. The abstract machine described here is one that diagrams technological, philosophical, social and artistic conditions enabling their operationalisation towards a new kind of concrete machinic assemblage. ARMM serves then as an abstract preparatory diagramatic scheme of principles, elements, and relations that can now be used in the next chapter for the operationalisation of a concrete technical system.

The primary function of the machine envisioned to be operationalised by the ARMM's principles is to configure the conditions under which musical agency, form, and process can emerge—it should enable an infrastructure which artists can use to world new artistic ecosystems. In this respect, it should participate in what Martin Rohrmeier describes as creativity at the meta-level—a kind of invention that reshapes the very task of composition, often by redefining its constraints, goals, or modes of expression. Rather than solving musical problems within a predefined style or grammar, meta-level creativity instantiates or alters the problem space itself (Rohrmeier, 2022, 54). This aligns with what Aaron Sloman calls the space of possible virtual machines: a conceptual topology in which different systems—human, computational, or hybrid—can instantiate minds or behaviours with different levels of complexity and responsiveness (Sloman, 1984, 2–3). For Sloman, we must abandon the binary between things that "have minds" and those that don't. Instead, we should examine how systems behave—how they process, learn, relate, or reflect. The ARMM treats compositional entities (features, agents, conditions) as modular, virtual machines. Each can be indexed, invoked, or recombined across compositional acts, which enact procedural behaviours across a shared infrastructure. If, as Liam Magee and Ned Rossiter argue, the relational database has had more impact on the transformation of organisational culture than the Internet itself (Magee & Rossiter, 2015, 74), the system to be operationalised by the workings of the abstract machine envisioned here, aspires to a similar role in composition: it redefines the infrastructure of musical agency by enabling new forms of compositional relation—between features, between composers, and between human and non-human agents. It makes room for systems that behave differently, think differently, or reconfigure compositional space in unexpected ways. As Sloman insists, "minds are not static things—it's what they do that is so important" (1984, 3). The system to be operationalised by the ARMM does not attempt to simulate the mind of any single composer; it scaffolds a field in which multiple kinds of compositional behaviour—human, machinic, hybrid—can co-exist and co-individuate.

The key idea behind the system to be operationalised by the ARMM is metamodelling. In simple terms, a metamodel is a model about models: a structure that defines how other structures—such as musical transformations, creative procedures, or generative systems—can be built, validated, distributed and recombined. In this way, the ARMM should operationalise a system that allows artists and agents to design not only compositions, but systems for making compositions. These systems can themselves

evolve, be shared, recombined, or restructured across multiple creative contexts. This is what gives them their autopoietic character: they are self-producing. Every operation enabled by the system resulting from the ARMM should be capable of being folded back into the system as a reusable building block, allowing the machine to modify its own structure over time.

The ARMM also operationalises a system that follows a distributed rhizomatic logic. Borrowed from the philosophy of Deleuze and Guattari, the rhizome is a model of organisation without a single root or central structure. Instead, it spreads in all directions, with many interconnected paths, entrances, and exits. In the ARMM, this means that creative operations are not chained in a fixed hierarchy, but can be linked, rearranged, and remixed in many different ways leveraging the possibilities of decentralised modes of creation. This supports a kind of creativity that is flexible, non-linear, and open to unexpected recombinations—what we have called rhizomatic metamodelling.

Rather than enforcing a single style, method, or aesthetic, the system to be operationalised by the ARMM provides a space where different creative strategies—whether generative (rule-based systems that produce outputs), interactive (systems that respond to inputs), or allagmatic (systems that transform their own rules)—can all coexist and evolve. The term allagmatic composition, as defined in Chapter 4, refers to a compositional approach where what is composed is not a sound or a piece, but a system of operations that itself can be modified and transformed. The system to be operationalised by ARMM is designed precisely to support this kind of higher-order composition.

In this way, the system to be operationalised by the ARMM functions like an infrastructure rather than a tool. It is a framework for configuring creative relations, which enables artists, agents, and systems to collaborate by contributing operations that are modular (can be reused), interoperable (can work with other components), and procedural (can be executed dynamically over time). The outcomes that arise from such a system emerge through interaction, experimentation, and transformation. This is what we have described as structural individuation: the process through which a new artistic form takes shape not by applying a fixed template, but by navigating a field of evolving relations.

Through presenting the Autopoietic Rhizomatic Metamodelling Machine as an

abstract diagram of relations and principles (operations of the abstract machine), this chapter completed the first step of what has previously been defined as worlding a compositional ecosystem. It envisioned the abstract machine, which should now be operationalised in a concrete technical system. The ARMM is then a speculative design—a conceptual diagram of a system that should now be built. But to make it function in practice, we must translate its principles into concrete operations, agents, and infrastructures. This is the task of the next chapter.

In Chapter 6, we will shift from theory to implementation. The ARMM will take shape as a real-world compositional infrastructure, built on blockchain technologies and realised as Decentralised Creative Networks—distributed systems in which creative agents interact, contribute, and collaborate. These agents may include human composers, machine learning models, or algorithmic processes. Within these networks, creative action will be driven by the Performative Transaction: a formal operational unit for rhizomatic metamodelling—that executes transformations, records contributions, and enables new compositional relations to emerge.

In short, if this chapter outlined what the Autopoietic Rhizomatic Metamodelling Machine is and why it matters, the next chapter will show what it enables—both conceptually and technically, procedurally, and operationally. It will move from speculation to implementation, from the abstract diagram to the executable machine. Through the concepts of Decentralised Creative Networks and Performative Transactions, we will see how the compositional logic of the ARMM can be translated into a functioning, programmable, and open-ended system of artistic creation.