

Composing Monads for a Musical Performance

Nick Rossiter*, Michael Heather,

Computer and Information Sciences, Pandon Building, Northumbria University, Newcastle NE2 1XE, UK

Copyright ©2019 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract Music is a testing challenge for formal information systems. Here we apply the full power of category theory to the challenge, involving the topos for data structuring and the monad for process. The topos handles many aspects of the data for a performance including the score and variants, the orchestral players, the conductor and the supporting infrastructure such as funding bodies. The monad as process controls the adjointness between the functors representing articulation and intonation, based on perceived activity in the brain in professional musicians. We present a musical performance as a categorical composition over time signatures that proceed in successive adjoint steps with the monad looking back and its associated comonad looking forward. The physical complexity of each musical sound operates in its respective time-frame, represented by a limit, as a colimit. The formalism can be implemented in a functional programming language such as Haskell.

Keywords Music, Performance, Category Theory, Topos, Monad

1 Introduction

Much work has been done on computer representations of music at the physical level. Developments by Klumpenhouwer such as K-nets [10] provide a way for representing transformations from one pitch-class to another. A pitch class is all notes an octave apart, for example all C available on a piano. In a classical system there are 12 pitch classes, one for each note on the 12-note scale. K-nets represented a fundamental change from a set-theoretical approach to music theory into a transformational one. Earlier the emphasis had been on the pitch classes being treated as sets of elements, each element being a note within the item. This enabled chords and other combinations of notes to be modelled. The transformational approach extended this technique by adding a transition from one pitch class to another to capture the dynamic possibilities within a musical piece. Such a transition from, say, K1 to K2 was tightly structured, with the target K2 being isographic to a source K1. Isographism is similar to isomorphism but within

a graphical context. The cardinalities of the source and target nodes must be the same. K-nets are therefore a very disciplined but restricted way of moving from one pitch class to another: they can handle the situation where the labelling of nodes is consistent from one system to another and where the transformations are classical as, for example, within the 48 preludes and fugues (Well-Tempered Clavier) of JC Bach, handling all 24 keys.

A more general form of K-nets was defined by Lewin [13] in an extension of Klumpenhouwer's work, attempting to make the graphs and their transformations more general. These have been termed L-nets by O'Donnell [25]. L-nets extend a node from being a static collection of pitch classes to a network of transpositions, giving a transformational model, allowing numerous graphic possibilities for representing a single pitch-class set.

2 Previous Attempts with Category Theory

L-nets still have their origin in set-based graph theory. It was not long before the potential was seen for a move to category theory with the nodes in the graph becoming categories and the edges becoming functors. Category theory should facilitate the development of a logical approach to music, which can be mapped into one of the physical approaches for implementation. Towards this aim a massive tome *The Topos of Music*, 1335pp long, was produced by Mazzola [17], bringing together many of the recent advances in the theory of music. The title is, however, misleading with a formal topos approach, based on the Cartesian closed category, not attempted. In the preface (p.v) it is stated that the word topos is used in the style of Aristotle's or Kant's topic. Chapter 19 *Topoi of Music* gives an overview of the Grothendieck Topology but does not relate the topology to music. In later sections the word topology is frequently used but is nowhere elevated formally to a topos. Section XVI, containing Appendices C-I, deals with many categorical concepts but not in a musical context. Because the book is disjoint in its treatment of the topos and music, it has failed to achieve its aim as highlighted in the title. The most relevant section for the application of category theory to music is Chapter 6 *Denotators*, a concept developed further in collaboration

with Andreatta.

The subsequent paper by Mazzola and Andreatta [18] develops the idea of a category of directed graphs with objects as notes or chords and edges as musical operations such as transposition. The formalism of K-nets in category theory as denotators is developed in detail as a digraph, with vertices and arrows. In music the vertices are pitch classes and the arrows are operations; between any two vertices, there may be multiple arrows and an arrow may map from a vertex to the same node, a loop. A path in a digraph is a sequence $p = a_1, a_2, \dots, a_i$ where i is the number of arrows (a) in the digraph. The operations are the elements of a group T/I (translation/inversion), that is a bijective mapping ensuring that p is invertible. In category theory T/I is a category with one object Z_{12} (the 12-note scale) and automorphisms $f : Z_{12} \rightarrow Z_{12}$. The authors acknowledge that Z_{12} is far too restrictive from the articulation viewpoint, replacing it by a four-dimensional real vector space R_4 , where the coordinates represent onset o , pitch p , loudness l , and duration d , in a parameterization by real numbers. Their use of the powerobject for collections of notes as a basic type enables chords to be represented, the powerobject being any combination of notes permitted from Z_{12}/R_4 . The complex categorial formalism ultimately developed involves limits, colimits, presheaves, powerobjects and the Yoneda embedding. The references to the categorial literature are very general but it appears that their approach towards limit denotators owes much to the uncited Eilenberg-Moore category: the pullback of the category of presheaves on the Kleisli category along the Yoneda embedding.

More recent work by Popoff and his co-workers developed the generalised Poly-K-net or PK-net [26]. PK-nets enable heterogeneous collections of musical objects to be naturally compared and manipulated [27]. In particular the cardinalities of the source and target nodes do not have to be the same and the labelling of the nodes in two different approaches may be varied to suit the genre. Five main categories are developed, one PKNR for the underlying PK-net and four others as homographies of the PK-net. Four functors are defined, relating the categories. Natural transformations are used to generalise isographies. Their work does not employ explicitly Cartesian closure so does not appear to be from a topos viewpoint. The dynamic aspects involve a combination of functors and natural transformations, following a Godement calculus approach.

Problems occurred with the sets representing the graphs, resulting in their replacement by the category of relations REL [28]. This facilitates handling relationships but is inferior to the pullback, which can be locally Cartesian closed and hence adaptable to a topos view. Another fundamental difficulty is that the reliance on directed graphs means that the approach is Euclidean, with its limited dimensions.

The main findings from the literature review are that the approaches do not provide a natural correspondence with music. In particular the conversion of the K-nets and successors to categories, functors and natural transformations is categorification at a low-level of the set theoretic graphical structures, on a 1:1 basis. However the denotators approach [18] with the apparent use of the Eilenberg-Moore category comes closest to our approach presented here and the PK-nets or denotators could

be very useful as a basic representation of the notation in the score.

In the remainder of this paper we provide an introduction to natural category theory in Section 3, a description of the data structure employed of a topos in Section 4, an assignment of musical processes in the brain in Section 5, use of the monad and their composition for processes operating within the topos in Sections 6, 7 and a discussion of the applicability of the approach and future work in Section 8. A verbal presentation of the work was given at the 6th World Congress on Universal Logic, held at the University of Vichy, France, in June 2018 [31].

3 Natural Category Theory

The alternative approach to categorification is to search for a natural correspondence between music and category theory. Music is a composition of sounds from point to point as a succession of transitions. Category theory also involves, as a central tenet, the principle of composition, from the target object of one arrow to the source object of another. In both music and category theory the arrows have a direction from a starting point to a closing point, though loops may exist. An isolated point in music is a sound without context while an isolated object in category theory is simply a set. It is the processes that map from object to object that provide the naturality.

The practice by a performer of playing a score is the personal communication, often highly intensive, of the piece to a listener. When performing a player is at the same time both looking forward to what is to be played next and looking back at what has just been played. The process of music is indeed similar to that for transactions in a database system, where monads have been used to represent process [30]. There are however some significant differences. Aestheticism is an important part of music, covering aspects of articulation such as style and improvisation, subject to the rules of intonation. So while in database systems it would be a major deficiency if transactions were not always perfect to the letter of the requirements, in music variation through expression is an integral part of a performance, involving a departure from the score in aspects such as phrasing, rubato and articulation. It is necessary to move from the syntactic level of Shannon's communication theory to the semantic/aesthetic level.

The simplest starting point for a data structure suitable for music is the pullback, representing a binary relationship as a limit of a product. Such a structure can be enriched to a locally Cartesian closed category (LCCC), by incorporating connectivity (exponentials), an internal logic (λ -calculus), identity (from the limit), interchangeability of levels (objects can be categories, a category-object), hyperdoctrine (adjointness between existential and universal quantifiers and the diagonal). Ideally the LCCC should be embedded in a topos, the data structure of choice in applied category theory, requiring the definition of relationships within a coproduct (co-limits), an internal intuitionistic logic (Heyting), a subobject classifier (query) and a reflective subtopos viewpoint (query closure). These structures will be illustrated in a musical context in the

next section.

In music the structure of the nodes is potentially very complex and diverse ranging from a single note through tonal chords and dissonant combinations to microtones. On the piano powersets of integers may suffice but the Cartesian space will be more complex for the violin. Cartesian spaces as pullbacks can be constructed for real numbers through smooth manifolds expressed as differential forms. Another dimension is the articulation, described earlier.

The intension/extension relationship plays a central role in music. The intension is the type; the extension is the collection of instances that satisfy the type. It is not as simple though as a hierarchy of types. There remains a philosophical dimension to the design. The Universe contains everything. The Universe of Discourse (UoD) is that section of the Universe of interest to our application. By the laws of physics we cannot isolate any part of the Universe but we can identify a section for our work. In this case the intension is the Universe and the extension, UoD, is the world of music. A musical manuscript is extensional to the UoD of Music as one of the objects in this universe but intensional to the manuscript's variants and their performances. Variants include changes to the score (composer initiated or developments after composer's death), rehearsal (conductor initiated) and performance. No two performances are ever the same. In the next two sections, we bring together these ideas in the formal definitions of the topos for a data structure and of the monad as a process, operating inside the topos.

4 The Topos as the Data Structure

We develop the categorial data structures introduced in the last section. Figure 1 shows the pullback diagram for the relationship $\mathbf{S} \times_{\mathbf{O}} \mathbf{V}$ of Score (\mathbf{S}) by Variant (\mathbf{V}) in the context of Occasion (\mathbf{O}). The placement of \mathbf{S} and \mathbf{V} is not arbitrary: \mathbf{S} is the independent variable and \mathbf{V} the dependent variable. $\mathbf{S} \times_{\mathbf{O}} \mathbf{V}$ holds the intension and extension for pairs of \mathbf{S} and \mathbf{V} participating in the relationship; \mathbf{S} , \mathbf{V} and \mathbf{O} hold the intension/extension for the Score, Variant and Occasion respectively. The diagram illustrates a number of features of category theory: the diagram commutes through different paths between objects yielding the same result, that is $\iota_l \circ \pi_l = \iota_r \circ \pi_r$; the restricted product $\mathbf{S} \times_{\mathbf{O}} \mathbf{V}$ is projected by π into its component category-objects, to the left through π_l as \mathbf{S} and to the right through π_r as \mathbf{V} ; the category-objects \mathbf{S} and \mathbf{V} are included by ι in the coproduct, written conventionally in shorthand as \mathbf{O} but in truth being the coproduct $\mathbf{S} + \mathbf{O} + \mathbf{V}$; ι_l is the left inclusion and ι_r is the right inclusion. The lower limit ensures the diagram is natural (universal) with a unique morphism u from the limit to the restricted product $\mathbf{S} \times_{\mathbf{O}} \mathbf{V}$, providing a factorisation through the whole diagram. The objects in the diagram are categories, with their own internal structure of arrows, termed category-objects.

It is worth emphasising that the term Occasion has been selected with some care. The term has a philosophical basis from Whitehead's work on *Process & Reality* [34] where it is a temporal actual entity. Among the four types defined, the last 'occasions of experience of the fourth grade' is the most relevant

for music, involving experience in the mode of presentational immediacy, which is taken to mean the qualia of subjective experience.

The diagram in Figure 1 can be converted into the LCCC of Figure 2 (labelled LCCC 1) by showing the functors between the restricted product $\mathbf{S} \times_{\mathbf{O}} \mathbf{V}$ and the coproduct $\mathbf{S} + \mathbf{O} + \mathbf{V}$; as a hyperdoctrine, there is adjointness between the functors: the existential quantifier \exists is left adjoint to the diagonal Δ , which is right adjoint to the universal quantifier \forall . The quantifiers \exists and \forall provide the search facilities of an information retrieval or database system. The diagonal functor Δ identifies pairs in the relationship between \mathbf{S} and \mathbf{V} in the context of \mathbf{O} . The opposite arrows are shown with inverse projections as π^* and inverse inclusions as ι^{-1} . Two properties, part of the adjointness, remain to be defined: η is the unit of adjunction measuring freeness through π_l in the diagram and ϵ is the counit of adjunction measuring co-freeness through π_r^* . The category-objects in a LCCC are typically further LCCC so a category-object at the top-level can be decomposed into further category-objects, each representing more detail in the application. A locally Cartesian closed category therefore provides a recursive feature.

An interesting question is when can the colimit, a pushout in category theory, be constructed in a pullback diagram. The colimit represents the upper limit, the defined sum of the included categories in the context of the limit. The sum can only be defined when some special conditions occur in the pullback diagram, restricting its flexibility.

The first condition concerns the freedom of the mapping. The relationship between the independent and dependent variable in its most general case is N:M (many to many), for example if the independent variable was Orchestra and the dependent variable was Player: each orchestra contains many players and each player performs for many orchestras. In this case each projection arrow is N:1 and each inclusion arrow is 1:N. A more restricted case is 1:N [2], such as for our running Score/Variant example, where each score has many variants but each variant refers to only one score. In this case the left projection arrow is N:1 and the left inclusion arrow is 1:N; the right projection and right inclusion arrows are both 1:1. The most restricted case is 1:1. Very few natural relations are 1:1 over any length of time. For example a Player may generally be assigned to one instrument but flute players for instance often play the piccolo as well. An Orchestra may have one Manager but over time new managers will be appointed. Even more artificial examples may not always be 1:1. For example Player to National Insurance Number is not 1:1 if two players have the same name. In such cases 1:1 can be enforced by names being required to be unique for booking purposes. However, in spite of its scarcity in nature, it is 1:1 that is required for the colimit to be constructed in a pullback diagram. So we cannot construct a colimit for Figure 2 as the relationship between Score and Variant is 1:N.

The second condition concerns the mapping onto the sum. It is necessary for this sum to only include values for the independent and dependent variables with no extraneous information. This is enforced by both inclusion arrows being typed as epic (surjective) so that all values in the sum are assigned,

as in Pulation diagrams [1]. Another issue with Figure 2 is that the diagram looks to be entirely intensional with no provision for the extension. We find that constructing a diagram with both pullbacks and pushouts facilitates the design of the intension/extension relationship.

Figure 3 shows a diagram for a pullback with a limit that is also a pushout with a colimit when the arrow f_0 from the independent to the dependent variable is 1:1 and the arrows ι_l and ι_r are epic (surjective) onto the defined sum. Such a diagram is termed Dolittle, named after *The Story of Doctor Dolittle* by Hugh Lofting (1920), involving the mythical Pushmi-pullyu creature. This Dolittle diagram, holds the intension/extension relationship for the category object S as follows. The arrow f_0 is a set-valued functor relating the independent variable, the higher object S , the type for Score, to the dependent variable, the lower object S , the set of values for Score. Since the mapping f_0 is to the set as a whole, f_0 is 1:1 and the inclusion and projection arrows are also 1:1 with the restricted product and defined sum expressed as sets. Both the inclusion arrows are epic. The restricted product, the limit SX or $S \times_{S^+} S$, is pairs of type and value in the context of the colimit S^+ , that is $S +_{SX} S$. The relationship on the left-hand side is an explicit product while that on the right-hand side is an implicit coproduct.

The diagram is repeated in Figure 4 with the nodes labelled as type, value or combination of the two; this gives a more explicit view of the intension/extension relationship with $f_0 : S\text{-type} \rightarrow \{S\text{-value}\}$.

Figure 5 shows in outline form how the category-object S within the LCCC of Figure 2 is a Dolittle square. Every node in the diagram of Figure 2 will be a similar internal pullback-pushout square, with limits and colimits, relating the definition to the instances, as described in detail by the authors elsewhere [30]. The type of the data values needs further elaboration. While Figure 4 shows the type for the Score is S-type, this is rather abstract. if we look at the work of Mazzola and Andreatta [18] discussed earlier, they suggest the use of denotators for articulation purposes with a musical object being expressed as the powerobject of Z_{12}/R_4 , where Z_{12} is the 12-note scale and R_4 holds the parameters onset o, pitch p, loudness l, and duration d, as real numbers. Z_{12} could be expanded to Z_{53} (53 Equal Temperament) to satisfy some musicologists [11], who find such a structure appealing for music from the east and for perfect thirds and fifths. We would not include the transitions as digraphs in the structural side: these are processes within the music (and the topos), for which we use monads as discussed later.

Dolittle diagrams appear to be equivalent to extensive categories, which have coproducts that interact well with pullbacks, and adhesive categories, which have pullbacks and pushouts of monomorphisms as 1:1 mappings. This is important as such categories are readily embedded into a topos [23, 24], yielding our objective of a topos as the data structure.

The full data structure will be more complex than a single pullback with the combination of further categories representing other entities such as composers, conductors, musicians and venues into one comprehensive LCCC. The pullbacks are pasted together to give complex relationships as described for

information systems elsewhere [30]. Here we use as an example in Figure 6 how data for the composers, held in the category-object C , is linked to that for the scores and variants. We label the pullback from Figure 2 as **Pb1** and introduce another pullback **Pb2** with independent variable C and dependent variable the relationship for **Pb1**. We paste the left inclusion ι_l' of **Pb2** onto the right projection π_r of **Pb1** to give a pullback square, which contains three pullbacks **Pb1**, **Pb2** and **Pb2** \times_V **Pb1** with the last named factored through V . All of these three diagrams have to be valid pullbacks in their own right for the relationships to hold. The overall relationship, which we label LCCC 2, is of score with variant by composer in the context of occasion. The attachment of the composer to the variant gives flexibility for the variant to be made by the original composer of the score or another musician altogether.

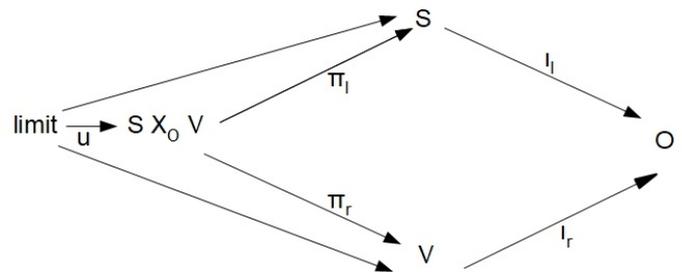


Figure 1. Relationship of Score by Variant in Context of Occasion as Pullback $S \times_O V$. Category-objects: S Score, V Variant, O Occasion

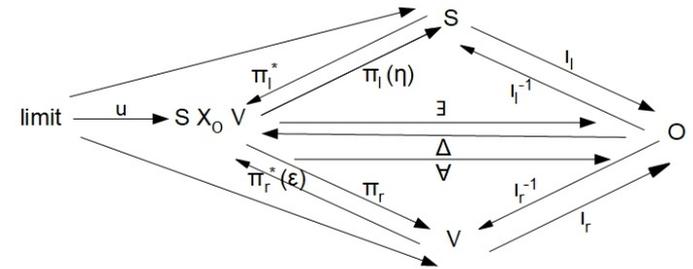


Figure 2. LCCC 1: Locally Cartesian Closed Category for Relationship $S \times_O V$. Category-objects: S Score, V Variant, O Occasion

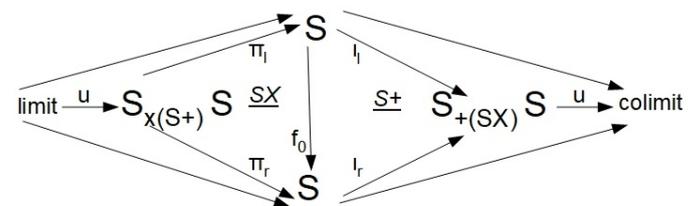


Figure 3. The Intension/Extension Relationship in a detailed Dolittle Diagram, as both a pullback and a pushout, for the Category-object S (Score) in Figure 2. f_0 relates intension S to extension S . The colimit is the pushout S^+ and the limit the pullback S_x

Pastings provide a way for extending a data description horizontally to represent relationships between independent entities or types. There are also relationships in which one entity-

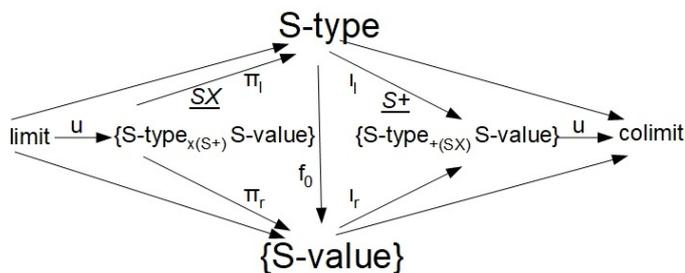


Figure 4. The Dolittle Diagram of Figure 3 for the Category-object **S** (Score) repeated with the nodes labelled as **S-type**, **S-value** or a combination of the two

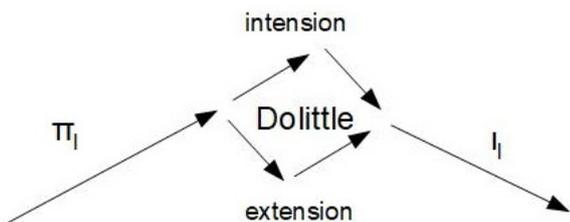


Figure 5. Category-object **S** (Score) as node in Figure 2, expanded in outline as the Dolittle Diagram in Figure 4

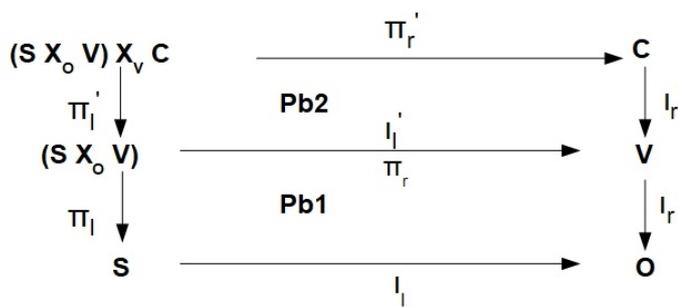


Figure 6. LCCC 2: Pasting of the pullback for the relationship $S \times_O V$ (**Pb1**) with a pullback with independent variable C (**Pb2**) to give overall relationship $(S \times_O V) \times_V C$ (**Pb2** \times_V **Pb1**), factored through V . Category-objects: C Composer, S Score, V Variant, O Occasion

type is dependent on another, in which we can extend the data description vertically by decomposition. Examining the components of an orchestra provides an example of this technique. An orchestra comprises sections such as strings, woodwind, brass and percussion. LCCC 3 can be constructed for the relationship $ORCH \times_O SCT$ between the category-object **ORCH** for Orchestra and the category-object **SCT** for Section in the context of category-object **O** for Occasion; this is shown in Figure 7 as **Pb3**. Occasion has the same meaning as before so this pullback records the orchestra and its sections for a point in the score at a particular event. Sections can be further specified, for instance, as by violin, viola, cello, double bass, for strings, with violin subdivided again into 1st and 2nd sections. However, in the present paper for simplicity we restrict the subdivision to one level.

Two vertical expansions come to mind. The sections can be expanded into the instruments and their players for an event and the orchestra can be expanded into its current administration, such as the roles and appointees, necessary for the orchestra to be funded and operational. Both these relationships can be handled as further pullbacks within LCCC 3 based on **Pb3**. The pullback **Pb4** in Figure 7 is a decomposition of the category-object **SCT**, showing the relationship between the category-object **INSTR** for Instrument and the category-object **PLR** for Player in the context of category-object **R** for Ranking, representing aspects such as first and second flute. The pullback **Pb5** in Figure 7 is a decomposition of the category-object **ORCH**, showing the relationship between the category-object **ADMIN** for Administrative-role and the category-object **APPT** for Appointee in the context of category-object **F** for From (date of appointment). Pullbacks **Pb3**, **Pb4** and **Pb5** collectively make up LCCC 3.

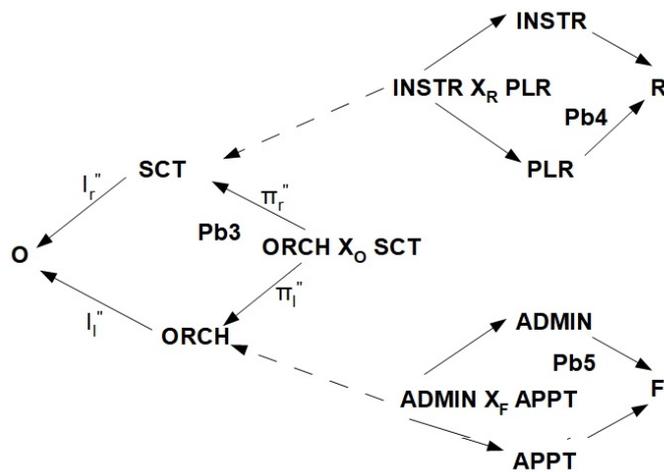


Figure 7. LCCC 3: the relationship $ORCH \times_O SCT$ (**Pb3**) between Orchestra and Section in the context of category-object **O** for Occasion, including the decomposition of category-object **SCT** for Section into the relationship $INSTR \times_R PLR$ (**Pb4**) and of category-object **ORCH** into the relationship $ADMIN \times_F APPT$ (**Pb5**)

The LCCC 2 of Figure 6 and the LCCC 3 of Figure 7 contain a common category-object **O** for Occasion. This means that the two LCCC can be merged into a single category, which we call **PERF** representing an instance of a performance. Figure

in more detail are A for articulation and I for intonation. A is the free functor, the creative step operating on one instance \mathbf{PERF} in the left-hand hemisphere of the brain. I is the underlying functor, enforcing the rules, operating on the same instance \mathbf{PERF} in the right-hand hemisphere of the brain. If the two functors are adjoint then there is a natural relationship $A \dashv I$, defined by the 4-tuple $\langle A, I, \eta, \epsilon \rangle$ where η is the unit of adjunction and ϵ is the counit of adjunction. The unit measures the creativity in one cycle of the free and underlying functors, starting with the free functor. The counit measures the adherence to quality in one cycle of the free and underlying functors, starting with the underlying functor. The adjointness therefore gives a measure of the creativity and quality of the performance. The monad is a process, controlling the composition of the adjoint functors IA across the two hemispheres of the brain and therefore acting as the *corpus callosum*. Also coordinating activity in the *corpus callosum* is the comonad, dual to the monad, which controls the composition AI . The monad and comonad represent the intense mental effort required in coordinating the left and right hands of the player. In the next section we look at the category theory in more detail. A schematic diagram of the processes involved in shown in Figure 9.

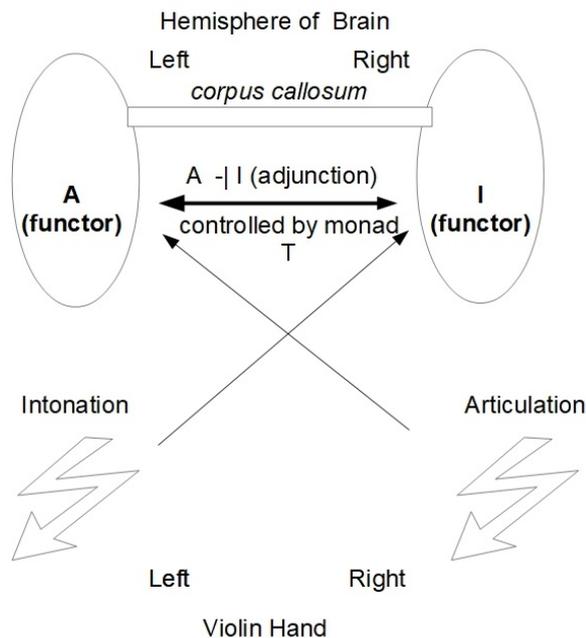


Figure 9. The Processes in the Brain of Articulation A and Intonation I as Categorical Concepts. A is left adjoint to I and I right adjoint to A , written $A \dashv I$, all under the supervision of the monad T

6 The Monad operating within the Topos

To achieve a musical performance we need to define an activity on the topos to realise the sound. This requires a process, as indicated by Heraclitus (all is flux) and discussed at length by Whitehead in *Process & Reality* [34]. In information

systems process corresponds to transaction, representing activity on the contents of a database system [29]. A transaction requires three cycles: the first makes the change, the second checks the change against the rules, the third rolls back the change if the rules are broken or records the change if it is acceptable. A transaction can be very complex but the whole is viewed as an atomic action with a binary outcome: success or failure. The before and after states must be consistent in terms of rules; intermediate results are not revealed to others and the results persist after a successful outcome.

In a musical performance matters are not so binary as in the world of databases. In the first cycle of adjointness, a performer will apply articulation and intonation to the time bar; in the second cycle the performer will apply the rules and may, for instance, make minor adjustments on the violin to the left-hand finger positions (intonation); in the third cycle the performer reviews the whole sound effect; only in extreme circumstances would the performer rollback, by reverting to the previous timebar; for the unit of adjunction, the change represents creativity by the performer in articulation with no change indicating a lack of expression; for the counit of adjunction, no change represents strict adherence to the score by the performer in intonation with change indicating departure from the score. Clearly there is a balance between articulation and intonation with a lifeless, over-strict performance resulting in no change to the unit and counit of adjunction and an expressive performance, deviating from the strict score, being measured by the unit of adjunction for creativity and by the counit of adjunction for the extent of the deviation.

The monad is the preferred way of representing process in category theory and functional languages. The term originates from Leibniz for an elementary ‘substance’ whose interior cannot be examined. In the object-oriented programming paradigm, Leibniz’s ideas correspond to encapsulation, used to define a process with a clearly-defined interface to hidden internal workings. A monad is based on an endofunctor, a functor with the same source and target category. Such an endofunctor may be written $F : \mathbf{X} \rightarrow \mathbf{X}$ where \mathbf{X} is a category. An endofunctor can also be a pair of adjoint functors: IA where $A : \mathbf{PERF} \rightarrow \mathbf{PERF}$, $I : \mathbf{PERF} \rightarrow \mathbf{PERF}$, as the source and target category is the same, \mathbf{PERF} . It is this application of an endofunctor that we adopt in this paper. The diagram in Figure 10 shows the adjointness $A \dashv I$ with its unit $\eta :: 1_P \rightarrow IA(P)$ and counit $\epsilon :: AI(P') \rightarrow 1_{P'}$ of adjunction, where P and P' are objects in \mathbf{PERF} .

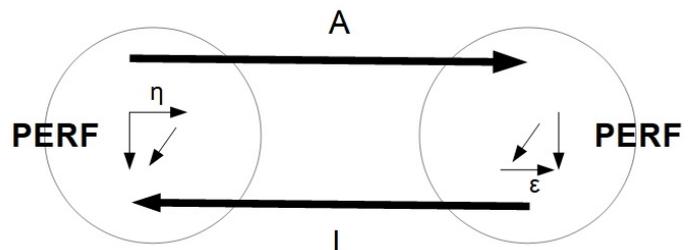


Figure 10. The pair of Adjoint Functors $A \dashv I$ as an endofunctor with source and target category \mathbf{PERF} . A is left adjoint to I and I right adjoint to A . I intonation, A articulation

As in the world of transactions, in category theory a monad can be viewed as involving three cycles: $IAIAIA$ of the free functor A and the underlying functor I defined above, where $A \dashv I$, that is A is left adjoint to I and I is right adjoint to A . The monad therefore involves three cycles of the diagram in Figure 10. It should be noted that while the term cycle is appropriate from the practical point of view it is slightly misleading: the monad is actually a snap rather than three cycles in turn, corresponding to the prehension (or grasping) of Whitehead [34]. Writing IA as T , the monad is shown in Figure 11(a). There is a dual comonad shown in Figure 11 (b) where $S = AI$, S^3 is $AIAIAI$. The monad $\langle T, \eta, \mu \rangle$ operates within the topos **PERF** as $T : \mathbf{PERF} \rightarrow \mathbf{PERF}$, and the comonad $\langle S, \epsilon, \delta \rangle$ as $S : \mathbf{PERF} \rightarrow \mathbf{PERF}$, where η and ϵ are the unit and counit of adjunction respectively and μ and δ are multiplication and comultiplication respectively.

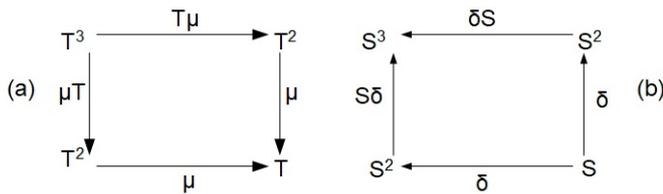


Figure 11. (a) The monad construction $T^3 \rightarrow T^2 \rightarrow T$ where $T = IA$, $\mu : T^2 \rightarrow T$ is multiplication; (b) the comonad construction $S \rightarrow S^2 \rightarrow S^3$ where $S = AI$, $\delta : S \rightarrow S^2$ is comultiplication. I intonation, A articulation

In keeping with its economical notation, a monad is commonly identified in category theory by its endofunctor rather than by an additional symbol. So the monad construction $\langle \mathbf{PERF}, T, \eta, \mu \rangle$ is referenced as the monad T . Similarly the comonad construction $\langle \mathbf{PERF}, S, \epsilon, \delta \rangle$ is referenced as the comonad S . Some sources use bold font for monad names indicating that the monad is a category [15]. While a monad does satisfy the properties of a category, we prefer to treat it as a functor, as done by Sankar [32], to indicate its active role. We add to the definition the category upon which the monad is based, here **PERF**, after Mac Lane [14].

7 Composition of Monads

One monad operation or instance T relates articulation and intonation in a single time-frame. We now need to compose monad instances across all the time-frames in a musical work. Compositionality is a cornerstone of category theory and at the applied level is straight-forward with, say, $\dots \circ T'' \circ T' \circ T$ representing the composition of the three monads: T'' with T' and T , as the progression from a time-slice T to T' to T'' , \dots . This naturality or abstraction at the applied level is maintained in the functional programming language Haskell [6], named after Haskell B Curry, who developed the transformation of functions through currying in the λ -calculus. In Haskell the monad is an increasingly popular construction as an abstract data class with the composition from monad to monad being natural as above. A motivation for Haskell in employing the

monad construction was to assist in compositionality of functional programs. Indeed Haskell provides a practical testbed for the categorial concepts developed in this paper.

In pure category theory the composition of monads has generated much theoretical discussion. The monad in Haskell is formally classified as an extension of the monad developed by Eilenberg-Moore, involving the notion of a strong monad [15, 16, 21] after work by Kleisli. In more concrete terms a strong monad is defined as a (categorial) monad with strengthening with respect to products and idempotency. The strengthening with products leads to the concept of a Cartesian monad where, if the underlying categories are pullbacks, the monad T preserves pullbacks and μ and η are Cartesian, then the monad is Cartesian. This strengthening of the monad results in composition of monads in Haskell being natural for all monads, including Cartesian ones. Our monads are Cartesian as **PERF** is Cartesian, involving products as pullbacks. More technical detail on the Kleisli monad can be found in earlier work by the authors [30], showing how the Kleisli lift is applied to a monad to improve its behaviour with respect to the laws of distributivity in comparisons and composition [16]. Such laws ensure that an operation applied to each of the parts of an expression gives the same result when applied to the whole e.g. $2(a+b) = 2a + 2b$. A distributive operation provides reproducible results.

To explore the detail for the music application, we need to define the three monads T, T', T'' and their respective dual comonads S, S', S'' :

$$\begin{aligned} T &= \langle \mathbf{PERF}, T, \eta, \mu \rangle \\ T' &= \langle \mathbf{PERF}, T', \eta', \mu' \rangle \\ T'' &= \langle \mathbf{PERF}, T'', \eta'', \mu'' \rangle \\ S &= \langle \mathbf{PERF}, S, \epsilon, \delta \rangle \\ S' &= \langle \mathbf{PERF}, S', \epsilon', \delta' \rangle \\ S'' &= \langle \mathbf{PERF}, S'', \epsilon'', \delta'' \rangle \end{aligned}$$

The underlying topos category remains unchanged as **PERF** but the adjunction is different, reflecting the different time bar and how it is performed with unique unit and counit of adjunction. So a player can be expressive or slightly off-pitch in one part of the score and not in another, as reflected in an actual performance.

For the subtleties in a performance of a musical composition we need to examine the potential connections between successive monads and comonads. There are at least three facilities available in category theory, involving 1) a composition of the functorial components of two monads; 2) a distributive law governing the composition of functorial parts of two monads; 3) adjointness between the functorial parts of a monad and a comonad. The functorial parts of monads, rather than the monads themselves, are always considered in composition as they are actions while the monads are classes. We now consider these facilities in more detail:

1. We can compose the functorial components of monads if a number of housekeeping rules are satisfied as defined by Barr & Wells ([3] p.259) with the expression: $T' \circ T$ a composite of monads with functors

T' and T respectively. Such a composition of monads $T_0 = \langle \mathbf{PERF}, T' \circ T; \eta_0, \mu_0 \rangle$ is compatible with monads $T = \langle \mathbf{PERF}, T, \eta, \mu \rangle$ and $T' = \langle \mathbf{PERF}, T', \eta', \mu' \rangle$.

2. We have a distributive law [4] with a natural transformation defined as $\lambda : T' \circ T \rightarrow T \circ T'$ comparing any two endofunctors on the category \mathbf{PERF} . The distributive law enables Cartesian monads to be composed *naturally*, as discussed above for the Kleisli lift.
3. There is an adjunction, between the functor component of a monad and the functor component of a successor comonad. The free functor in such an adjointness is the functor in the comonad and the underlying functor is the functor in the monad [3, 4]. So from the above definitions we can write the following possible adjunctions for the monad/comonad structures: $S' \dashv T$ and $S'' \dashv T'$. These adjunctions hold if appropriate units and counits can be defined. They then provide a chain through the performance from one timebar to another with the following complete definitions respectively as examples: $\langle S', T, \eta_{S'}, \epsilon_T \rangle$ and $\langle S'', T', \eta_{S''}, \epsilon_{T'} \rangle$. Note that at this level it is the comonad with its apparently forward-looking action that is driving the creativity with the monad, looking backwards, enforcing the rules.

From the musical perspective the composition of the functorial components of monads drives the performance forward from one bar to another, with the distributive law ensuring that the composition is natural. Such a composition would be 'wooden' in the sense that the relationship between successive bars is fixed through the timeline. The adjunction adds scope for expression between one bar and another with the unit of adjunction measuring expressiveness, such as variation in timing between one bar and another, and the counit of adjunction measuring quality in adherence to the score in moving from one bar to another, such as in phrasing. Earlier work by the authors [7] on handling time jitter in category theory is relevant for handling subtle differences in phrasing.

8 Discussion

The net-based approaches discussed earlier in Section 2 provide a graphical approach to capturing musical performance. In the early stages the approaches were clearly set-based but later attempts moved to category theory. Unfortunately these later attempts do not represent a major advance as they are mainly categorification of the earlier set-based approaches, that is a conversion of set structures to categories on a 1:1 basis. Such techniques do not make use of the natural applicability of category theory to music as described in Section 3.

In our work we regard, as a significant development, the use of the topos in the form of a LCCC with intension/extension relationships as Dolittle diagrams in the nodes of the data structure. The closest approach to our work in structural terms is that on the denotators by Mazzola and Andreatta [18], who use the Eilenberg-Moore category, a lower level approach but

with significant handling of musical notation. Denotators could complement our work by providing a detailed description of the score in category theory terms. The topos also offers data structuring across the whole of the musical application from more organisational issues such as the infrastructure of an orchestra through to composers and scores and their variants. The limit of the topos provides an index within the score to the bar being played; the colimit represents the sound generated at that particular index value. The sound therefore is influenced by everything in the application from the infrastructure, the calibre of the performers and the fundraising through to the physical sounds generated from the variant of the score by the instruments under the directions of the conductor.

We would regard the monad as another significant breakthrough in representing musical performance, handling process, the dynamic aspects of the performance. The monad and its dual the comonad enable the performance of a musical piece to not only be monitored in simple terms as composition from one timeline to another but also in terms of adjointness between the intonation and articulation in the performance as it progresses. Such adjointness gives a measure of accuracy and expressiveness. The processes involved are linked to those in the brain in Section 5 for a violin with intonation as a process I played by the left hand and articulation A played by the right hand.

The aesthetic aspects of music, such as anticipation, can also be realised in our approach. There are two distinct viewpoints of anticipation in music: the performer's and the listener's. The player anticipates the sequence of notes to come by extrapolating intuitively from preceding notes while at the same time physically keeping an eye on the subsequent score and the listener builds up a mental image of the music as it evolves. Tension builds for listeners as their anticipation of the performance is realised or denied. Such experience is captured naturally by the monad/comonad structure with its forward/backward nature and inherent adjointness. A specific case of this is discussed in Section 7 where adjointness occurs between the active part of the monad in one timeline and the active part of the comonad in the following timeline. Overall the monad looks backwards ($T^3 \rightarrow T^2 \rightarrow T, T = IA$) and its comonad forwards ($S \rightarrow S^2 \rightarrow S^3, S = AI$) in their three cycles. However, the situation is more subtle than this: in each cycle the monad looks forwards (functor A for articulation) and then backwards (functor I for intonation) and its comonad looks backwards (I) and then forwards (A). The duality of the monad/comonad represents communication in an orderly manner within initially defined colimits and adjointness. Values for η (unit of adjunction) and ϵ (counit) represent rhetoric and dialectic respectively for the performance, giving a measure of expressiveness and accuracy. It is possible that there is a faltering in the communication, resulting in a roll-back with revised adjointness.

Category theory is no longer solely a theoretical exercise. As discussed earlier in Section 7, implementations are available in the functional programming language Haskell [6]. Indeed the implementation fed back into the theory with the need for the Kleisli lift clearly established in monad composition. Besides the use of an implementation in Haskell of the category theory

for the more abstract levels in the music, there are routes in Haskell through to the music itself as signals via Paul Hudak's work on the Haskell School of Music [8].

There are a number of areas where further work is desirable. The violin is a special case of a musical instrument with a clear delineation between the roles of the hands in generating the music. A number of other string instruments are similar, such as the viola, cello, double bass and lute but the pedal harp involves additionally foot control. Woodwind and brass instruments are more complicated with the mouth and both hands involved and the piano, while notionally having the melody played by the right hand and the harmony by the left hand, is much more complicated in advanced composition. An attempt to produce a more general match between the brain and the instrument is desirable. The notion of anticipation, as key to any communication process, could also be developed further from a more detailed examination of the interplay between the monad and its associated dual, the comonad. An attempt to link the structures developed in this work with the denotators of Mazzola and Andreatta and the Haskell School of Music of Hudak would provide a very interesting demonstrator project.

9 Conclusions

Our approach naturally handles both the static and dynamic aspects of a musical performance in a general and flexible manner within the coherent theoretical framework of category theory. The static data structuring involves the topos, based on the locally Cartesian closed category (LCCC), providing connectivity, relationships through products and limits and colimits. The dynamic process involves the monad, controlling the adjointness between articulation and intonation, mirroring the activity in the brain. The formalism produced can be implemented in a functional programming language such as Haskell.

Glossary

Adjoint: a natural relationship between a functor and its dual.

Arrow: a mapping or morphism from one object to another.

Cartesian: a product space.

Categorification: conversion of concepts to categories by rote.

Category: a collection of arrows, that are connected to each other.

Closed: a category with limit, colimit and λ -calculus logic.

Colimit: the upper boundary of a category.

Composition: an action taking the output from one task as the input to another.

Dolittle: a diagram that is both a pullback and a pushout.

Dual: an arrow in the opposite direction.

Endofunctor: a functor with source and target the same.

Extension: the values for a category according to some intension.

Functor: a mapping from one category to another.

Identity: a mapping from an object onto itself.

Intension: the definition of a category.

Isomorphism: indistinguishable mappings.

Limit: the lower boundary of a category.

Monad: an operation on an adjoint.

Natural: unique up to some isomorphism.

Natural transformation: a mapping from one functor to another.

Object: the source and target of arrows.

Pasting: the amalgamation of one pullback with another.

Pullback: a product expressed as a relationship over some objects.

Pushout: a coproduct expressed as a sum over some objects.

Source: the domain of an arrow.

Subobject classifier: an object which returns a logical value, such as true or false.

Target: the codomain of an arrow.

Topos: a closed category with intuitionistic logic and a subobject classifier.

Acknowledgements

For discussions on pure category theory: Michael Brockway; for insights into musical performance: members of the Royal Northern Sinfonia, The Sage, Gateshead, UK, particularly Alexandra Raikhlina, Sub-Principal 1st violin; for discussions on music in practice: Gillian Reed, Denise Jordon, William Wilson, Alexander Rossiter; for suggestions on improving readability of the manuscript: two anonymous reviewers.

REFERENCES

- [1] Jiří Adámek, Horst Herrlich & George E Strecker, *Abstract and Concrete Categories*, John Wiley (1990). Recent edition (2005) online from <http://katmat.math.uni-bremen.de/acc>.
- [2] R Banach, *Regular Relations and Bicartesian Squares*, *Theoretical Computer Science* 129(1) 187-192, 1994. Online available from [https://doi.org/10.1016/0304-3975\(94\)90086-8](https://doi.org/10.1016/0304-3975(94)90086-8).
- [3] Michael Barr & Charles Wells, *Toposes, Triples and Theories*, Version 1.1, Reprinted by *Theory and Applications of Categories*, Version 1.1, Reprinted by *Theory and Applications of Categories*, 2000,. Online available from <http://www.math.mcgill.ca/barr/papers/tttall.pdf>.
- [4] J Beck, *Distributive Laws*, *Lecture Notes in Mathematics* 80, Springer-Verlag, 1969, 119-140. Online available from <http://www.math.mcgill.ca/barr/ftp/pdf/ftpfiles/distlaw.pdf>.
- [5] S Dehaene, E Spelke, P Pinel, R Stanescu & S Tsivkin, *Sources of mathematical thinking: behavioral and brain-imaging evidence*, *Science*, 284 (5416), 970-974, 1999.
- [6] λ -Haskell: an advanced, purely functional programming language, 2017. Online available from <https://www.haskell.org/>.

- [7] Michael Heather, Nick Rossiter & Dimitris Sisiaridis, The Semantics of Jitter in Anticipating Time Itself within Nano-Technology, CASYS'07, Liège, AIP Conference Proceedings 1051, 326-337, 2008. Online available from <http://nickrossiter.org.uk/process/liege07jitter.pdf>
- [8] Paul Hudak, The Haskell School of Music - From Signals to Symphonies - Yale University, Department of Computer Science, Version 2.4 353 pp, February 22 (2012).
- [9] KL Hyde, J Lerch, A Norton, A Forgeard, E Winner, AC Evans & G Schlaug, Musical training shapes structural brain development, *J Neurosci* 29(10), 3019-3025, 2009.
- [10] Henry Klumpenhouwer, A Generalized Model of Voice-Leading for Atonal Music, PhD Thesis, Harvard University, 1991.
- [11] Jim Kukula, Fractal Microtonal Music, in: *Toward a Buddhist Philosophy of Science*. Online available from <http://www.interdependentscience.com/music/calliopist.html>.
- [12] Daniel J. Levitin, *This Is Your Brain on Music: The Science of a Human Obsession*, Plume/Penguin, 2007.
- [13] David Lewin, Thoughts on Klumpenhouwer Networks and Perle-Lansky Cycles, *Music Theory Spectrum* 24(2) 196-230, 2002.
- [14] Saunders Mac Lane, *Categories for the Working Mathematician*, 2nd ed, Springer, 1998.
- [15] Ernie Manes & Philip Mulry, Monad Compositions I: General Constructions and Recursive Distributive Laws, *Theory and Applications of Categories*, 18 (7), 172-208, 2007.
- [16] Ernie Manes & Philip Mulry, Monad compositions II: Kleisli strength, *Mathematical Structures in Computer Science* 18(03):613-643, 2008.
- [17] Guerino Mazzola, *The Topos of Music - Geometric Logic of Concepts, Theory, and Performance*, Birkhäuser, Basel, 1335pp, 2002.
- [18] Guerino Mazzola & Moreno Andreatta, From a Categorical Point of View: K-Nets as Limit Denotators, *Perspectives New Music* 44(2) 88-113, 2006.
- [19] Greg Miller, Music Builds Bridges in the Brain, *Science: Brain & Behavior*, Apr. 16, 2008. Online available from <https://www.sciencemag.org/news/2008/04/music-builds-bridges-brain>.
- [20] A.Milne, W.A Sethares & J Plamondon, Isomorphic Controllers and Dynamic Tuning: Invariant Fingerings Across a Tuning Continuum, *Computer Music Journal*, Winter, 31(4) 15-32, 2007.
- [21] Eugenio Moggi, Computational Lambda-Calculus and Monads, *Proceedings of the Fourth Annual Symposium on Logic in Computer Science* 14-23, 1989.
- [22] Eugenio Moggi, Notions of Computation and Monads, *Information and Computation*, 93, 55-92, 1991.
- [23] ncatlab, Adhesive Categories. Online available from <https://ncatlab.org/nlab/show/adhesive+category>
- [24] ncatlab, Extensive Categories. Online available from <https://ncatlab.org/nlab/show/extensive+category>
- [25] Shaun O'Donnell, Klumpenhouwer Networks, Isography, and the Molecular Metaphor, *Intégral*, 12 53-80, 1998.
- [26] Alexandre Popoff, Moreno Andreatta & Andrée Ehresmann, A Categorical Generalization of Klumpenhouwer Networks, 5th Int Conf MCM 303-314, 2015.
- [27] Alexandre Popoff, Carlos Agon, Moreno Andreatta & Andrée Ehresmann, From K-Nets to PK-Nets: A Categorical Approach, *Perspectives New Music*, 54(2) 5-63, 2016.
- [28] Alexandre Popoff, Moreno Andreatta & Andrée Ehresmann, Relational PK-Nets for Transformational Music Analysis, arXiv:1611.02249, 19pp, submitted, 2016.
- [29] B N Rossiter, M A Heather & D Sisiaridis, Process as a World Transaction, *Proceedings ANPA 27 Conceptions*, 122-157, 2006. Online available from <http://nickrossiter.org.uk/process/anpa064.pdf>
- [30] Nick Rossiter, Michael Heather & Michael Brockway, Monadic Design for Universal Systems, ANPA 37-38, Anton L Vrba (ed.), 369-399, 2018. Online available from <http://nickrossiter.org.uk/process/Rossiter-ANPA-PROC-37-38-2final.pdf>
- [31] Nick Rossiter & Michael Heather, Musical Performance: a Composition of Monads, UNILOG'2018, 6th World Congress on Universal Logic, University of Vichy, France, 21-26 June 2018, in: *Handbook of the 6th World Congress and School on Universal Logic*, Jean-Yves Beziau, Arthur Buchsbaum & Christophe Rey, with the assistance of Alvaro Altair & Yannis Ayari (edd) <https://www.uni-log.org/vichy2018> Workshop: Logic and Music, pp.212-213, 2018.
- [32] Anirudh Sankar, Monads and Algebraic Structures, University of Chicago REU, 2012. Online available from <http://math.uchicago.edu/~may/REU2012/REUPapers/Sankar.pdf>.
- [33] CJ Steele, JA Bailey, RJ Zatorre & V Penhune, Early musical training and white-matter plasticity in the corpus callosum: evidence for a sensitive period, *J Neurosci* 33(3), 1282-1290, 2013.
- [34] Alfred North Whitehead, *Process and Reality: An Essay in Cosmology*, Macmillan, New York (1929); corr.ed., eds. David Ray Griffin and Donald W. Sherburne, New York: Free Press, 1978. Online available from https://monoskop.org/images/4/40/Whitehead_Alfred_North_Process_and_Reality_corr_ed_1978.pdf.