MuseReduce: A Generic Framework for Hierarchical Music Analysis

Petter Ericson
Umeå University
Sweden
pettter@cs.umu.se

Yannis Rammos
École Polytechnique
Fédérale de Lausanne
Switzerland
yannis.rammos@epfl.ch

Martin Rohrmeier
École Polytechnique
Fédérale de Lausanne
Switzerland
martin.rohrmeier@epfl.ch

Abstract

In comparison to computational linguistics, with its abundance of natural-language datasets, corpora of music analyses are rather fewer and generally smaller. This is partly due to difficulties inherent to the encoding of music analyses, whose multimodal representations—typically a combination of music notation, graphic notation, and natural language—are designed for communication between human musician-analysts, not for automated large-scale data analysis. Analyses based on hierarchical models of tonal structure, such as Heinrich Schenker's, present additional notational and encoding challenges, since they establish relations between non-adjacent tones, and typically interpret successions of tones as expressions of abstract chordal sonorities, which may not be literally present in the music score. Building on a published XML format by Rizo and Marsden (2019), which stores analyses alongside symbolically encoded scores, this paper presents a generic graph model for reasoning about music analyses, as well as a graphical web application for creating and encoding music analyses in the aforementioned XML format. Several examples are given showing how various techniques of music analysis, primarily but not necessarily hierarchical, might be unambiguously represented through this model.[1]

1 Background

In contemporary music theory, theories of tonal structure often represent a piece of music as a hierarchy of simpler models known as "reductions". Beyond the mature theory of Schenker (1977), and the scholarship of disciples such as Salzer (1962), the discipline has seen a proliferation of such theories, and an accumulation of hierarchical analyses, especially since the 1970s and '80s (e.g., Benjamin, 1981; Finkensiep and Rohremeier, 2021; Komar, 1971; Lerdahl & Jackendoff, 1996; Rohrmeier, 2011; Yust, 2018; see also Rohermeier & Pearce, 2018). A common concern of hierarchical theories is the distinction between structural and ornamental functions of a given tonal entity at various levels of the tonal structure.

Examples of hierarchical analyses abound in textbooks and the music-theory literature, yet corpora of such analyses are scant at best. By way of comparison, they are generally smaller than data sets available in natural language processing (NLP), where access to large-scale data—not

[1] The source code of the web application is published under the AGPL license with DOI 10.5281/zenodo.6395095 and downloadable at https://github.com/DCMLab/reductive_analysis_app/.

only of actual utterances and text, but also of analyses—has proven valuable both in evaluating the coverage of various theories, and in measuring the performance of computerized approaches to analysis. An array of analogous problems remains open, especially in music-theory building, and would significantly benefit from the availability of large "ground-truth" corpora of hierarchical analyses. The pioneering work of Kirlin and Yust (2016), who extracted the implicit analytical criteria of human experts by applying machine-learning algorithms on a small corpus, is suggestive of the possibilities. Hierarchical-analysis corpora would also enable innovative studies on musical genres and styles, taking into account deep-level tonal structures as well as surface features.

Furthermore, corpus building provides an incentive for rigorously systematic thinking. The creation of large-scale data sets of hierarchical analyses necessitates an additional level of conceptual clarity in the models and theories, which may expose interesting edge-cases and exceptions, as well as discrepancies between theoretical principles and analytical practice. Despite these compelling research problems, the conceptual and notational complexity of hierarchical analyses, coupled with the lack of a general-purpose annotation tool for intuitively encoding them, have impeded the development of large-scale corpora.

In response to these needs, this paper presents a graphical web app for creating new analyses and interacting with them in various ways. Crucially, in contrast to models that establish relations between multi-note entities, such as key regions or form units, the building block in our model is the individual note. All higher-level tonal entities consist of sets of related notes. In principle, this design renders the app suitable for a wide range of models, including, beyond Schenkerian and neo-Schenkerian theories, also motivic or "semiotic" techniques, pitch-class set theory, and various transformational theories.

2 Related work

The lack of computer-accessible corpora of music analyses is actively being remedied in various forms. For harmonic analyses, scholarship and software by Hentschel et al. (Hentschel, Moss, et al., 2021; Hentschel, Neuwirth, et al., 2021; Neuwirth et al., 2018) applies a file format and annotation standard compatible with the MuseScore system to encode the analysed harmonies within the actual score file. For analysing form, Dezrann (Giraud et al., 2018) is a web application written to facilitate collaborative analysis of scores, annotating sections of the score using a graphical tool.

As mentioned, Rizo and Marsden have worked on an MEI-based storage format for hierarchical music analyses, though to date no corpus has been published using it. Other corpora of hierarchical analyses, such as that by Kirlin (2014) and Harasim et al. (2020), have opted to store analyses separately from the score, using relatively simple *ad hoc* file formats, since their main object of interest is the analytical data as such, not its relation to the score. A similar encoding strategy of separate files is adopted in the 300-piece corpus of Hamanaka et al. (2014), which provides a MusicXML encoding of each piece, as well as all its reductions according to Lerdahl & Jackendoff's generative theory of tonal music (GTTM).

The use of graphs to encode analyses has a long history in linguistics, especially for encoding semantics, with the large-scale Abstract Meaning Representation (AMR) graph-bank releases (Baranescu et al., 2013) being an especially salient example. Probably the most widely-known theories modelling music in graph- or tree-like fashion are those of the Schenkerian and neo-Schenkerian traditions, and the GTTM by Lerdahl and Jackendoff (1996). Recent work by Yust evidently draws inspiration from both of these approaches, though his Minimal Outerplanar Graphs are more strictly hierarchical than the freer graph-like structures of Schenkerian analysis (Yust, 2018).

Recent work by Finkensiep and Rohrmeier (2021) also uses an approach to music analysis with close affinities to the model presented here. Establishing note connections within a graph expressive of hierarchical relations, their model proposes a set of simple operations to generate more complex tonal structures. Also relevant is Robert Snarrenberg's *WesterParse*, a software tool and associated corpus for proto-Schenkerian linear analysis of species counterpoint (Snarrenberg, 2021). Snarrenberg's work is based on Peter Westergaard's seminal *Introduction to Tonal Theory* (Westergaard, 1975).

3 Graphs and graph encoding

In contrast to broader approaches which take key regions, themes or motifs, or harmonies as the objects of study, this framework primarily regards these types of musical objects as entities recursively generated by simpler ones, with the individual musical note as the minimal entity.[2] Moreover, the concept of an analysis is in itself abstracted into a generalised notion of *relations*. Finally, relations need to relate to not only notes, but also other relations. In short, for the purposes of this framework, a *music analysis* is a set of relations that makes precise claims about, on the one hand, a set of musical notes, and, on the other hand, themselves.

Analyses can relate certain of its members as *secondary* to others, e.g., a neighbour note is usually considered as less salient or important than the note(s) that it elaborates. The concept of rhythmic hierarchy is also familiar and has been well theorised. Likewise, there are many approaches to the hierarchical analysis of harmonies (see, e.g., Harasim, 2020; Lerdahl & Jackendoff, 1996; Rohrmeier, 2011; and others reviewed in Rohrmeier & Pearce, 2018).

The structure chosen to model and encode analyses is that of a *graph*—a collection of vertices connected by edges. This choice strikes an appropriate balance between non-local and interleaving relations and the need for formal and precise representations. As an additional benefit, tree analyses can be seamlessly embedded within the encoding, as tree structures are a special case of graph structures.

In particular, notes and analytical relations between them are encoded as vertices in the graph, while edges connect relation vertices to their member entities (other relations or individual notes). Though less free than an entirely unrestricted graph structure, where vertices and edges could also be introduced to represent other concepts, this extremely generic structure still places a large amount of responsibility on the part of the encoder to clarify the semantics of relation vertices—that is, what they actually represent in a music-analytical sense. This is especially important since there may be several ways of encoding any particular analytical concept. For example, a Schenkerian linear progression could be encoded as a single relation

[2] The notion of a "note" is not always self-evident in music analysis. Depending on the abstractions of the chosen theoretical model, it may lack, for example, a specific register, an exact metric placement of its onset, or an exact duration. Conversely, a tone may not necessarily exist in the original score as a material pitch event, yet may be implied by the voice leading and even play an important structural role in the analysis (Rothstein, 1991).

labelled as such, but could also be encoded as the combination of a passing relation and an arpeggiation between the endpoints of the passing motion. Unlike the former option, the latter would semantically represent the fundamental principle of linear progressions as stepwise unfoldings of deeper-level harmonic intervals. In general, the semantic neutrality of the low-level (meta)relation "primitives" of this model allows for the representation of higher-level analytical concepts of a wide range of individual analytical techniques, albeit at the inevitable cost of verbosity: multiple (meta)relations are often needed to represent a single analytical concept.

Modelling hierarchical tonal structure as graphs also enables the encoding of hierarchical violations, which are partly admissible, and certainly existent in actual analytical documents, at least in Schenkerian analysis. It also makes possible the encoding of associative (non-hierarchical) relations, most notably motivic repetitions.

3.1 Formal definitions

Graphs. Formally, a *directed graph* $g = (V, E, \text{att})$ is a set of *vertices* V, a disjoint set of *edges* E, and an *attachment function* $\text{att}: E \to V^2$. A *directed path* (of length l) in a graph is a string $s=v_1e_1v_2e_2 \dots e_{l-1}v_l$ of alternating vertices and edges such that $\text{att}(e_k) = (v_k, v_{k+1})$. A path from a vertex u to itself is a *cycle*. A (directed) graph without cycles is called a Directed Acyclic Graph (DAG).

Partial orders. A relation $R \subset S^2$ is an order if it is reflexive ($\forall a: R(a, a)$), antisymmetric ($R(a, b) \land R(b, a) \to a = b$), and transitive ($R(a, b) \land R(b, c) \to R(a, c)$). It is a *total order* if it is total ($\forall a, b: R(a, b) \lor R(b, a)$), and a *partial order* if it is not.

Each partial order corresponds to a DAG, and the reflexive and transitive closure of the edge relation $\text{edge} \subset V^2$ defined as $\forall (u, v \in V): \text{edge}(u, v) \equiv \exists e \in E: \text{att}(e) = (u, v)$ of a DAG $g = (V, E, \text{att})$ is a partial order.

3.2 Vertices and edges

A *note* is the basic unit of analysis, and each note is represented in the graph as a single vertex. As used in this framework, a note is a musical object that has a graphical representation in a score. Frequently, it has an associated pitch, rhythmic value and timbre, and perhaps also lyrics, accents, dynamics, or other musical information. At the initial state, there are no other vertices in the graph, and no edges.

In order to relate notes to each other in an analysis, *relation vertices* are introduced into the graph. Each relation can have an associated *type* (as well as other data), and is connected to the notes it relates using edges. For all notes n in a relation r, we add an edge $e = (r, n)$ marked with whether the note is primary or secondary in the relation. If there is no distinction between notes in a relation, they are usually considered secondary. Let $\text{primary}(r, v)(\text{secondary}(r, v))$ indicate that v is primary (secondary) in r. [3]

Similarly, to relate relations to each other, additional *meta-relation vertices* can be introduced into the graph, with the same marking of the edges which connect a meta-relation to

[3] An alternate way of modelling relations is as *hyperedges*, which also generally explicitly order the notes that a relation connects.

relations or other meta-relations. In short, edges in the graph can connect *i)* relations to notes, *ii)* meta-relations to relations, and *iii)* meta-relations to meta-relations. Henceforth we will be using the term "(meta)relation" as a shorthand of "relation or meta-relation."

3.3 Partial orders and hierarchies

The hierarchies implied by the distinction between primary and secondary members of a (meta)relation can be reified as (partial) orders on various subsets of the vertex set of the graph. Let $g = (V, E, \mathrm{att})$ be a graph, $R \subset V$ its (meta)relation vertices, $N \subset V$ its note vertices, and $S \subset R$ some specific set of (meta)relations of interest. We can then define the relation $<'_S$ on the set N as

$$\forall (u, v \in V): u <'_s v \Leftrightarrow (\exists r: r \in S \wedge primary(r, v) \wedge secondary(r, u))$$

and $<_s$ as the reflexive and transitive closure of $<'_S$. $<_s$ is not necessarily an order, as there may be note vertices $u, v, u \neq v$ such that $\mathrm{u} <_s \mathrm{v}$ and $\mathrm{v} <_s \mathrm{u}$, violating antisymmetry.

From a partial order $<_s$, we can extract one or more *maximal (minimal)* vertices v, such that $\nexists u: u \neq v \wedge (v <_s u) (\nexists u: u \neq v \wedge (u <_s v))$. Indeed, it is possible to partition the set ordered by a partial order into a collection of "levels" from the maximal or minimal direction, e.g., by iteratively removing all the maximal (minimal) elements (which would constitute the "next top (bottom) level"), and computing the next set of maximal (minimal) elements.

4 Workflow & use cases

4.1 Basic annotation workflow

When annotating a piece of music using the tools presented here, the user must provide the score in an encoding format that the Verovio tool (Pugin et al., 2014) can convert to MEI (MusicXML, ABC, etc.). The score will then be rendered to an SVG image by the Verovio Javascript engraver, and the user will be presented with the user interface. Annotating a relation between notes can be done simply by clicking on the notes, and then selecting or typing in the appropriate relation type in the pop-up window that appears; the annotated relations will be displayed as a curved hull enclosing the constituent notes. After a short annotation session, the score may look similar to that of Figure 3. For a more detailed look at an annotation process, see Section 4.4.

4.2 Hierarchies, reductions and layers

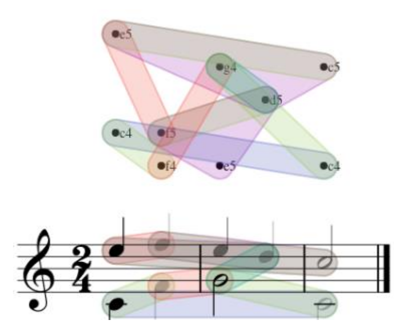

Figure 1: Partial order visualisation in the web app

The web app can show the partial orders implied by the annotated relations, as described in Section 3.3. Figure 1 shows a hierarchical placement from the "minimal" perspective, where the "more ornamental" notes are placed at the bottom, and successive higher levels show the new set of minimal notes once the lower level is discounted. There is moreover the option to "reduce out" successive levels from the minimal direction, removing, at each step, the minimal note vertices and all relations connected to them from view. In effect, this procedure produces a progressively more abstract model of the original score (a “reduction”) by deleting notes of lowermost hierarchical value from it. Of special relevance to music analysis is the additional option to limit the reduction process to the scope of selected relations, providing the user with fine-grained control over which notes to hide. However, this user-guided procedure currently reduces-out only note vertices that are globally minimal, and will also hide the relations involving them as well. Work is ongoing to overcome these limitations. The implementation of more refined reduction algorithms, which would match the music-theoretical intuition, remains an open research problem.

Also provided is the option to re-render the score into a new layer, which removes all hidden notes from the score, and deletes any measures thus made empty. Further options allow the removal of rhythmic information by placing each onset into its own quarter-note time-span—a simple attempt at rhythmic “normalization”—with notes either sustained throughout the time-span, or shifted to its onset. New layers can also be edited by adding new notes to the score, for instance notes implied by the harmonic context. At present, new notes need to have the same onset and length as existing notes.

4.3 XML encoding

At any stage of the music-analytic annotation process, the analysis can be stored within the MEI file of the original score, using a variant of the scheme introduced in Rizo and Marsden (2019). The original score element is not changed; instead, the graph is encoded in a separate XML subtree, whose vertices (nodes) establish links between corresponding notes of the analysis and original score. We use the `@corresp` attribute to this end (Rizo and Marsden, 2019 used `@sameas`). Relations exist only in the graph subtree, while layers are encoded as additional `mdiv` elements, with `@corresp` attributes linking elements to their origin instance: a note either in the original score, or in the layer where a new note was introduced.

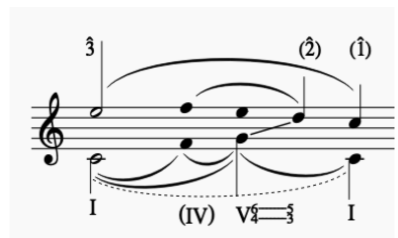

Figure 2: Example of a simple Schenkerian annotation

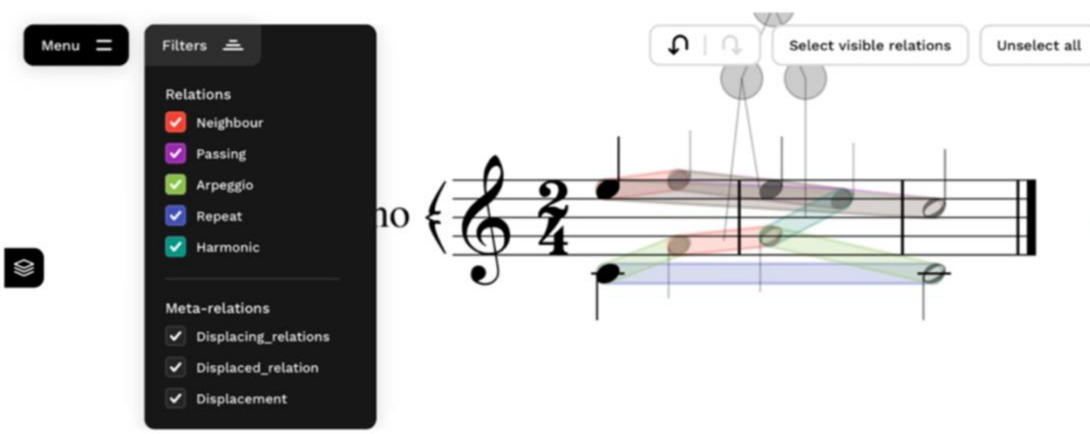

Figure 3: An annotation of the Schenkerian example of Figure 2 as encoded and depicated in the web app.

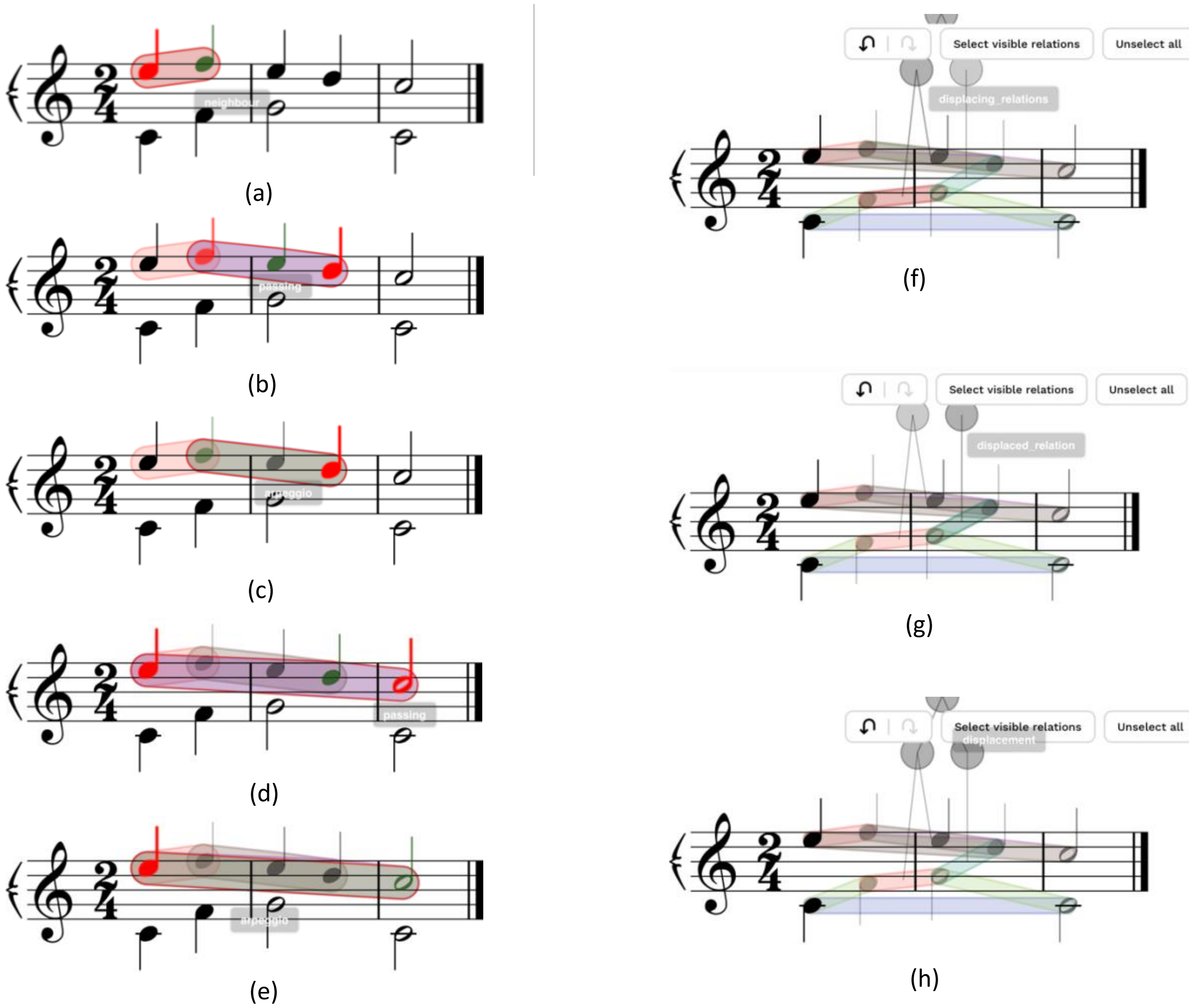

Figure 4: Successive stages of the annotation process, with "tool tips" displaying the labels of various elements under the user's cursor.

Figure 2 depicts a minimal tonal structure.[4] The example is presented in standard Schenkerian notation, with white notes hierarchically superior to black ones. Overall, the structure represents a prolongation of the tonic harmony. The upper voice delineates a stepwise descending motion $\hat{3}$–$\hat{2}$–$\hat{1}$ supported, at the deepest level, by a bass arpeggiation I–V–I. Note that, at a lower level, a melodic neighbour-note $\hat{4}$ is harmonically supported by a predominant IV harmony and initiates a lower-level descending-third progression from $\hat{4}$ to the more structural $\hat{2}$. The counterpoint of this progression against the bass has rhythmic implications: the structural dominant harmony now appears horizontally expressed, as a displacement between its upper-voice tone D5 and its bass G4 (see diagonal line), while the passing tone E produces a passing 64 sonority.

[4] Despite possible first impressions, this structure could not be an *Ursatz* proper because scale degree $\hat{3}$ in the upper voice is assumed to be hierarchically superior to $\hat{2}$ and $\hat{1}$. In an *Ursatz* all three melody tones would have been irreducible.

The complete annotation of such a tonal context in our web app is shown in Figure 3. Every Schenkerian operation of the previous, conventionally-notated figure is now represented by one or more graph *relations*, each visualised with a straight arc between two or more notes. Relation arcs are colour-coded by the app, and labelled by the annotator, according to the type of relation that they represent.

Figure 4a demonstrates a potential first step of this annotation process, in which $\hat{4}$ is declared as an upper neighbour to $\hat{3}$. Red note-heads represent the primary notes in a given relation (hierarchically higher), green note-heads the secondary ones (hierarchically lower). Less trivial is the annotation of the aforementioned linear progression from $\hat{4}$ to $\hat{2}$ via a passing $\hat{3}$. While a number of encoding strategies could be conceived for such a pattern, we choose to explicitly represent the double nature of the Schenkerian linear progression: both as a passing motion between two hierarchically higher tones (Figure 4b), and as the elaboration of a harmonic relation (arpeggiation) between its outer tones (Figure 4c). The overarching descending third progression $\hat{3}$–$\hat{2}$–$\hat{1}$ is encoded in similar fashion, using two relations (Figures 4d and 4e). As for the encoding of the bass line, it should now be self-evident given the colour coding of the filters panel (Figure 3).

The thorniest aspect of our encoding pertains to the "diagonal" displacement. Encoding Schenkerian displacements as simple relations between two individual tones would make an economical but not quite eloquent representation, since it would not directly capture the deeper contrapuntal causes of this rhythmic phenomenon. Instead, we opt for an encoding with a degree of expressive redundancy, grouping together not only the two displaced tones themselves, but also the respective voice-leading transformations which generate the displacement. This is achieved with the meta-relation shown in Figure 4f. We then assign a second meta-relation to the displacement proper (Figure 4g). Finally, we group the two meta-relations together (Figure 4h). The finished annotation is depicted in Figure 3.

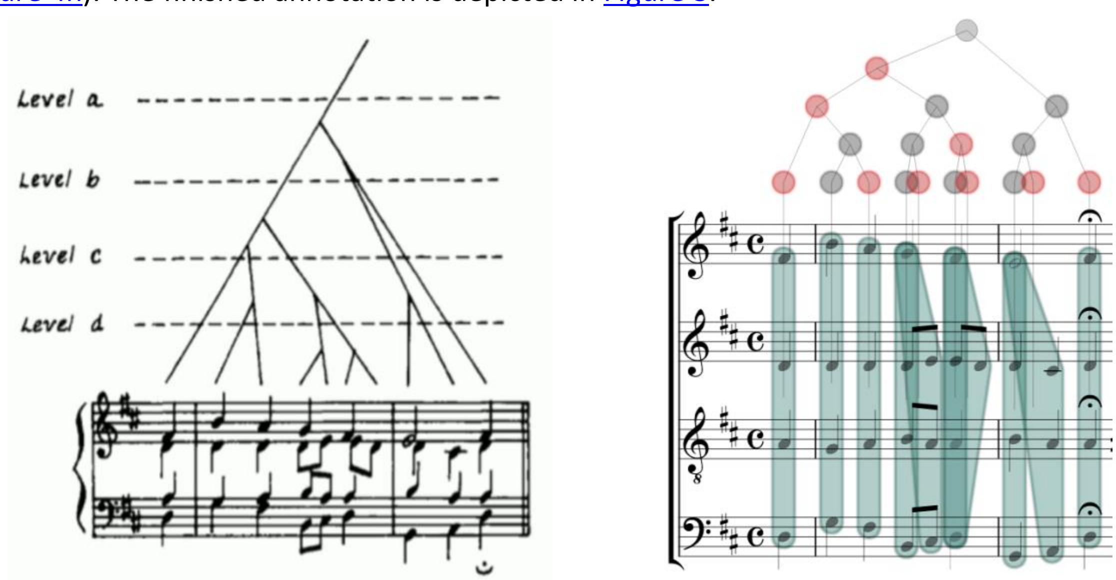

Figure 5 (a) Prolongational tree of J. S. Bach's *O Haupt voll Blut und Wunden*, reproduced from Lerdahl and Jackendoff 1996, p. 202.

(b) Encoding of prolongational tree using hierarchy of meta-relations.

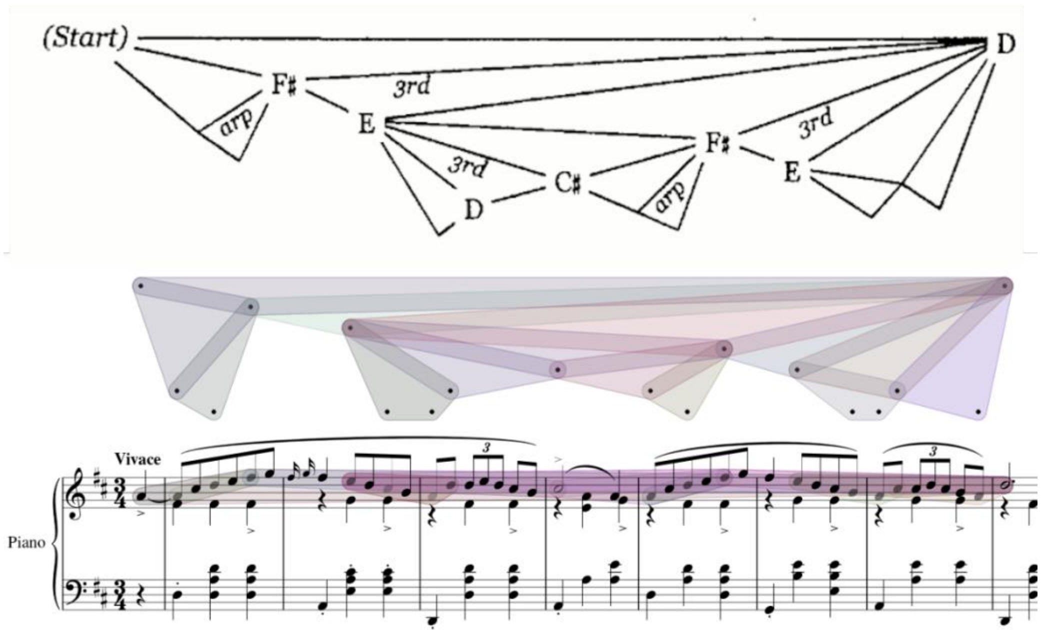

Figure 6: Analysis of Chopin's Mazurka op. 33, no. 2, reproduced from Yust (2018), p. 34, and visualization of its encoding in the web app.

The encoding process for a GTTM prolongational reduction is similar. Figure 5 shows one possible annotation of Lerdahl and Jackendoff's analysis of Bach's *O Haupt* (Lerdahl & Jackendoff, 1996, p. 202). We use relations (highlighted here in green) to demarcate the notes of each sonority in the chorale, then apply meta-relations to encode the prolongational tree in bottom-up fashion. Left-branch prolongations are encoded as secondary–primary relations in our model, right-branching prolongations as primary–secondary. The resulting tree of meta-relations in 5b effectively reproduces the original tree of 5a.

Our final example (Figure 6) demonstrates an application of our model on Yust's Maximally Outerplanar Graphs (MOPs) (Yust, 2018, p. 34). The coloured tree in the screenshot is a partial-order visualisation, produced without human intervention or supervision by the web app once the triangles of the original MOP graph had been encoded. The close match between the visualisation of Yust and the automatic output of the web app shows that Yust's model and visualisation is suited for automatic processing, and highlights the ways the web app enforces precision: in the Yust graph, two of the triangles actually span four notes of the reduction each. This shows up in the web app visualisation as quadrangles, in both cases with no hierarchical distinction between the two lower notes.

5 Conclusion

Although the model and annotation tool described above were initially motivated by needs of corpus construction, their current scope evolved to be broader, spanning music teaching and computer-assisted music analysis, among other fields. Additional features are under construction to support such applications alongside corpus building. Among them is an automatic "spelling checker" for annotated relations, ensuring, for example, that "passing motions" are indeed stepwise, sequential, and monotonic, or that neighbour notes are indeed a step apart from their main note(s).

As demonstrated in Section 4.4, both app and model are versatile enough to support multiple tonal theories and styles of annotation. However, as the Schenkerian example suggests, the visual rendering of these relations bears little resemblance to any standard notations of graphic analysis, nor is it currently optimal from a readability perspective. The design of alternative visualisations for hierarchical relation annotations is thus another field of active research.

Acknowledgements

This project was conducted at the Latour Chair in Digital and Cognitive Musicology, generously funded by Mr. Claude Latour and has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 760081 – PMSB.

References

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., & Schneider, N. (2013). Abstract meaning representation for sembanking. *Proceedings of the 7th linguistic annotation workshop*, 178–186.

Benjamin, W. E. (1981). Pitch-class counterpoint in tonal music. In R. Browne (Ed.), *Music theory: Special topics*, 1–32. Academic Press.

Finkensiep, C., & Rohrmeier, M. (2021). Modeling and inferring proto-voice structure in free polyphony. *Proceedings of the 22nd International Society for Music Information Retrieval Conference*.

Giraud, M., Groult, R., & Leguy, E. (2018). Dezrann, a web framework to share music analysis. *International Conference on Technologies for Music Notation and Representation (TENOR 2018)*, 104–110.

Hamanaka, M., Hirata, K., & Tojo, S. (2014). Musical Structural Analysis Database based on GTTM. *Proceedings of the 15th International Society for Music Information Retrieval Conference*.

Harasim, D. (2020). *The learnability of the grammar of jazz: Bayesian inference of hierarchical structures in harmony*. EPFL.

Harasim, D., Finkensiep, C., Ericson, P., O'Donnell, T. J., & Rohrmeier, M. (2020). The jazz harmony treebank. *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 11–16.

Hentschel, J., Moss, F. C., Neuwirth, M., & Rohrmeier, M. (2021). A semi-automated workflow paradigm for the distributed creation and curation of expert annotations. *Proceedings of the 22nd International Society for Music Information Retrieval Conference.*

Hentschel, J., Neuwirth, M., & Rohrmeier, M. (2021). The Annotated Mozart Sonatas: Score, harmony, and cadence. *Transactions of the International Society for Music Information Retrieval* 4(1).

Kirlin, P. B. (2014). A data set for computational studies of Schenkerian analysis. *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 213–218.

Kirlin, P. B., & Yust, J. (2016). Analysis of analysis: Using machine learning to evaluate the importance of music parameters for Schenkerian analysis. *Journal of Mathematics and Music* 10(2), 127–148.

Komar, A. (1971). *Theory of suspensions: A study of metrical and pitch relations in tonal music*. Princeton University Press.

Lerdahl, F., & Jackendoff, R. S. (1996). *A generative theory of tonal music*. 2nd edition, with a new preface. MIT press.

Neuwirth, M., Harasim, D., Moss, F. C., & Rohrmeier, M. (2018). The Annotated Beethoven Corpus (ABC): A dataset of harmonic analyses of all Beethoven string quartets. *Frontiers in Digital Humanities* 5, 16.

Pugin, L. X., Zitellini, R., & Roland, P. (2014). Verovio: A library for engraving MEI music notation into SVG. *Proceedings of the 15th International Society for Music Information Retrieval Conference*.

Rizo, D., & Marsden, A. (2019). An MEI-based standard encoding for hierarchical music analyses. *International Journal on Digital Libraries* 20(1), 93–105.

Rohrmeier, M. (2011). Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music* 5(1), 35–53.

Rohrmeier, M., & Pearce, M. (2018). Musical syntax I: Theoretical perspectives. *Springer handbook of systematic musicology*, 473–486. Springer.

Rothstein, W. (1991). On implied tones. *Music Analysi*s 10(3), 289–328. https://doi.org/10.2307/853970

Salzer, F. (1962). *Structural hearing: Tonal coherence in music* (2 Volumes). Dover.

Schenker, H. (1977). *Free composition* (2 volumes). Translated by Ernst Oster. Pendragon Press. (Original work published 1935.)

Snarrenberg, R. (2021). WesterParse: A transition-based dependency parser for tonal species counterpoint. *Proceedings of the 13th International Conference on Computer Supported Education CSEDU 2021* (1), 669–679.

Westergaard, P. (1975). *An introduction to tonal theory*. Norton.

Yust, J. (2018). *Organized time: Rhythm, tonality, and form*. Oxford University Press.