Exploring Musical Expression on the Web: Deforming, Exaggerating, and Blending Decomposed Recordings

Florian Thalmann, Sebastian Ewert, Geraint Wiggins, Mark B. Sandler
Centre for Digital Music
Queen Mary University of London
f.thalmann@qmul.ac.uk, s.ewert@qmul.ac.uk

ABSTRACT

We introduce a prototype of an educational web application for comparative performance analysis based on source separation and object-based audio techniques. The underlying system decomposes recordings of classical music performances into note events using score-informed source separation and represents the decomposed material using semantic web technologies. In a visual and interactive way, users can explore individual performances by highlighting specific musical aspects directly within the audio and by altering the temporal characteristics to obtain versions in which the micro-timing is exaggerated or suppressed. Multiple performances of the same work can be compared by juxtaposing and blending between the corresponding recordings. Finally, by adjusting the timing of events, users can generate intermediates of multiple performances to investigate their commonalities and differences.

1. INTRODUCTION

Musical performance consists in shaping musical material, which may be given by a score, in precise and subtle ways in several of its dimensions including time, dynamics, pitch, and timbre, depending both on the musical context and instrumentation [2,18,20]. In the temporal dimension, for instance, this is generally referred to as expressive timing, microtiming, or agogics. It may take years of listening and playing experience with a particular style to be able to perceive these subtleties or distinguish them from plain errors [1], and an even longer training to be able to execute such timings [5]. Computational methods and tools have proven useful for the analysis and comparison of different performances and playing styles, both for musicologists [4,9,16,17] and musical performance students, [3,10,15,21], and may be equally useful for laypersons as a gateway to understanding performance practice [11]. A central approach in many of these tools is to temporally align multiple performances of a piece, which enables switching between corresponding positions across performances. Additionally, some of the tools offer an interactive visualization of the musical material or of analytical results obtained from the underlying audio; e.g. the tool presented in [10] allows pianists to visually represent the temporal and dynamic characteristics of notes in patterns or chords on a two-dimensional plane, whereas the one in [3] highlights the currently performed passage within a virtual representation of the sheet music.

Several studies have been led to investigate how such tools are found to be useful or valuable in practice. In [15], while the majority of the feedback was positive, switching between versions was sometimes mentioned as being too abrupt and the delays involved as potentially distracting from the music. Furthermore, while visualizations are generally seen as helpful [10,19], they may be perceived as too abstract and thus difficult to get acquainted with and understood musically.

In this paper we introduce the concepts behind a prototypical web application[1] for the exploration of performances and their underlying expressivity primarily through listening, by allowing users to deform the original audio recordings in specific ways in order to highlight various musical aspects. In particular, after decomposing a given music recording into individual note events using sound source separation techniques, subtle performance characteristics such as the timing, order, or relative dynamics of notes in a chord can be exaggerated locally without changing the global tempo. Other aspects of the piece or performance, related to the pitch or harmonic content, can be highlighted using spatialization and amplification techniques. Furthermore, different performances of the same material can be juxtaposed, combined, and blended into each other. Users can also generate new performances based on recombinations of existing ones[2]. In this sense, our *sonification-by-deformation* is a novel concept to highlight musical characteristics within a recording, and can be seen as an alternative to visualization or simpler alignment-based comparison techniques.

All this is enabled by the Web Audio API, which allows us to play back audio samples efficiently through optimized and modifiable audio graphs, which can be run in various browsers and platforms optimized for sample-based playback. Conceptually, our tool is based on the interplay of three main technological components: a score-informed audio decomposition technique [8], a hierarchical and graph-based

[1] https://github.com/florianthalmann/performance-playground

[2] The tool currently includes a visualization for orientation and interaction purposes, but may be potentially be realized without any visual interface in the future, e.g. as a teaching tool purely based on auditive communication and run on a contemporary smart speaker interface such as the Amazon Echo.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution**: owner/author(s).

Web Audio Conference WAC-2017, August 21–23, 2017, London, UK.

© 2017 Copyright held by the owner/author(s).

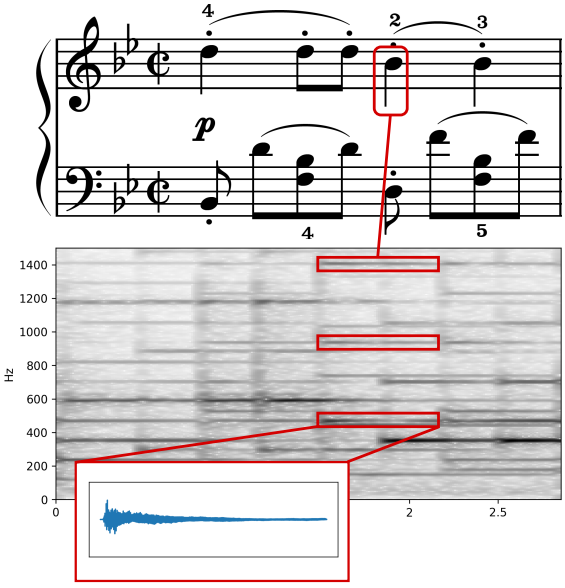

Figure 1: Note-level audio decomposition: Given an audio recording and a score of a piece of music (here: Schubert Op142No3), we employ the score as prior knowledge to identify, for each note in the score, the corresponding part of a time-frequency representation of the recording.

representation of the music using semantic web standards, and an object-based audio rendering techniques as introduced in [24]. In Sections 2-4, we discuss these three components in order followed by a brief overview of the functionality of the tool in Section 5. Finally, we conclude in Section 6 with a prospect on future work.

2. NOTE-LEVEL AUDIO RECORDING DECOMPOSITION

As a first step, we decompose the audio recordings of a number of performances of a particular musical piece into individual note-objects. Without additional information about the recording conditions and the piece, however, such a separation is a highly challenging task and leads to various ill-posed problems from a modeling point of view [7]. To simplify and guide the separation process, we incorporate information from a corresponding musical score, which we assume to be given – a concept referred to as score-informed source separation, see [7] for an overview. More precisely, we start by using the method proposed in [25] to automatically map each time position in the audio recording to a corresponding position in a MIDI file representing the underlying score. The method is a variant of Dynamic Time Warping and employs chroma-based representations to yield a first robust alignment, whose accuracy is further refined based on onset indicator features and by aligning musical voices independently. This way, we obtain a reasonable estimate for when each score note is played in the recording.

Next, we use this aligned MIDI information to learn how each note contributes to the audio recording, or more precisely, which part of a time-frequency representation of the recording is associated with each MIDI note, see Figure 1 for an illustration. To this end, following [8], we use a specific type of neural network: an *autoencoder*. Such networks are typically used to learn a function that condenses its input (i.e. the audio recording) into a smaller, low-dimensional representation in such a way that this representation still contains the necessary information to reconstruct the input. Usually, this learned representation does not provide directly interpretable information. However, employing the *structured dropout* technique presented in [8], we can use the aligned MIDI information to change this behavior. As a result, the modified autoencoder yields a representation that encodes the signal information separately for each musical pitch – this enables a direct modification of individual notes in this learned representation. In particular, by eliminating the information for all but one note, we can extract each individual note from the recording and create a corresponding *note-event audio object* or note-object for short. Using the alignment information from the first step, each note-object is richly annotated with score information, such as pitch, dynamics and start time as specified in the score. By further analyzing the extracted note-objects, we can also obtain refined information about the dynamics and start time as used in the performance [6].

3. ONTOLOGICAL REPRESENTATION

The information associated with note-objects and their relationships with and within the score are then represented using semantic web technologies, in particular, using a data model based on the Common Hierarchical Abstract Representation of Music (CHARM) [12, 13] and the Dynamic Music Object Ontology [23]. These structural representations are automatically generated using designated scripts that analyze and process the output of the decomposition step described above. Figure 2 shows a simplified example of such a representation, consisting of a score and two performances.

We begin by creating a multi-hierarchical representation of the score S, in which each of its N note events e_i^S with $1 \leq i \leq N$ becomes a lowest-level constituent. Any hierarchical musical structural characteristic, e.g. corresponding to sections, phrases, bars, beats, chords, voices, or instruments is then represented as mid-level nodes, grouping multiple lower-level objects as appropriate. These levels of grouping are assumed to be defined by the performed work or a basic analysis thereof and are thought to be common to all potential performances of the work. We then annotate these note events and their parent objects with a set of appropriate attribute or `feature`[3] values from the score F_S, such as onset time, duration, pitch, metrical position, dynamic value, and any other desired labeling. See [24] for JSON-LD[4] examples of such annotated structures.

We then create a similar representation for each performance P_k of S with $1 \leq k \leq K$ where K is the number of performances. The performance's events $e_i^{P_k}$ are linked to events e_i^S in the score representation via a relation s, assuming the same indexing $1 \leq i \leq N$ for both[5]. So far

[3]In the Dynamic Music Object Ontology, there are two different kinds of attributes: immutable `features` and mutable `parameters`.

[4]http://json-ld.org

[5]Due to the nature of the decomposition method described in Section 2, we obtain exactly one performed note event for each score note event, even if due to playing errors or expressive deviations the performance contains additional notes or some notes are missing.

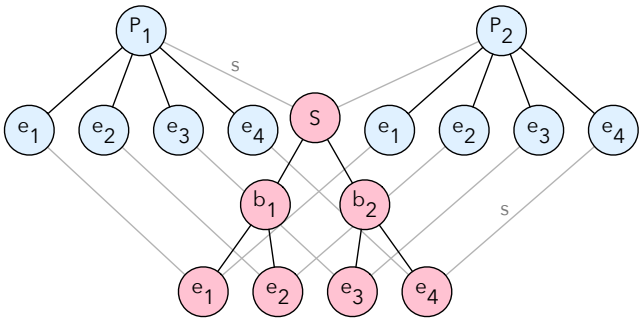

Figure 2: A simplified representation of a score S and two performances P_1 and P_2. The constituents of the performances are linked to the corresponding ones in the score via the relation s. The score has an additional level of hierarchy b_k.

we merely experimented with the simplest case, a two-level structure with a parent constituent P_k which directly links to all of the performance's note events. However, the data representation format also allows us to introduce hierarchies for performance representations, which may be used to represent characteristics of a particular interpretation, such as specific temporal groupings, articulation, or particular techniques, such as fingerings, chosen by the performer.

Each of the performance's lowest-level events $e_i^{P_k}$ then also links to the corresponding audio data generated during the decomposition step. Analogously to the score, we annotate the performance constituents with a set of `features` F_{P_k} of performed values, including onset time, duration, amplitude, pitch, etc. These values are either given by the output of the decomposition process or can be extracted from the resulting separated audio data. Finally, for each P_k we define the necessary performance `parameters`, P_{P_k} which determine the values considered by the scheduler during playback. As opposed to the immutable features F_{P_k}, these parameters are what is modifiable about the particular events, e.g. their time-stretch ratio, spatial positioning, or scheduled onset time.

4. PERFORMANCE DEFORMATIONS

Our application uses the core library of the Semantic Player framework [24][6] to read these representations and play them back in the browser. The library's scheduler is specialized on object-based playback and allows us to recombine the separated note-objects while changing their playback characteristics such as their amplitude, playback rate, local time-stretch ratio, or spatial positioning, in relation to the analytical and structural data stored in the representation. Furthermore, the latest version of the Semantic Player framework allows us to define logical constraints on any selections of values of parameters, features, and controls. These will then be maintained by the player whenever a value changes in the representation graph.[7]

A main purpose of our application is to highlight and illustrate aspects specific to a performance directly in the audio domain, i.e. without a visualization. In order to do so, we will use the structured information as described in Section 3 to control our parametric player framework in such a way that certain musical characteristics are attenuated or emphasized. We can do this by defining directed constraints that relate the available feature information F_S and F_{P_k} to specific player parameters P_{P_k}, while only solving for the latter. For each type of constraint, we introduce some high-level parameters that adjust their behavior, e.g. the degree to which a certain constraint is affects the original constellation of the audio. These high-level parameters can then be tied to designated user controls, such as GUI elements or mobile sensors, in most cases by a simple identity constraint.

4.1 Deforming Expressivity

The first set of constraints we introduce can be used to deform the degree of expressivity of a performance with respect to a particular musical dimension. We select a feature pair $f \in F_S$ and $g \in F_{P_k}$, and a corresponding parameter $p \in P_{P_k}$, e.g. score onset, performed onset, and playback time, for which we introduce an *expressivity* parameter $E_{fgp} \in \mathbb{R}$, which determines the amount by which we deform. The constraint then looks as follows, operating on the features and parameters of all $e_i^{P_k}$ with $1 \leq i \leq N$:

$$p_i = g_i + (E_{fgp} - 1)(g_i - f_i')$$

where

$$f_i' = (f_i - f_{min})\frac{g_{max} - g_{min}}{f_{max} - f_{min}} + g_{min}$$

is the feature f_i linearly mapped onto the performed feature space of g_i.[8] x_{max} and x_{min} are the maximum or minimum of feature or parameter x on all $e_i^{P_k}$.

This definition entails that for $E_{fgp} = 1$ the parameter p_i assumes the original performed value g_i and for $E_{fgp} = 0$ the mapped score value f_i'. For $E_{fgp} > 1$ we get an *exaggerated* version of the performance, where each performed value is scaled to be farther away from the score by the chosen factor. We can even choose $E_{fgp} < 0$, where we obtain a kind of a *performance negative*, where all the parameters are away from the score in the opposite direction.

The most straightforward deformations we can create with such expressivity parameters are the ones we obtain by pairing analogous performance and score features with the corresponding playback parameter, e.g. for onset deformation we choose f = score onset, g = performed onset, and p =playback time. Figure 3 shows how this works for a simple two-note example. With an expressivity factor of 1, we get the original performed onset values. With a factor of 0, we get the score values, and with 2 an exaggerated performance, where each note is moved further away from the score.[9] We can do the same for duration, dynamics/amplitude, pitch, etc. This way we can emphasize the characteristics of a particular performance and the way it differs from the score, e.g.

[6]https://github.com/florianthalmann/dymo-core

[7]In previous versions of dymo-core, and the dymo ontology, as well as in previous papers about the framework, these constraints were more limited and were referred to as *mappings*, as it is common in computer music [24].

[8]Depending on the feature pair, we may also need to choose other ways of mapping between score and performance, for instance logarithmic mapping for certain dynamic or frequency values. Here, however, for simplicity, we assume that we can find pairs where linear mapping makes sense.

[9]Note that the positions of the score events visualized in this example are in fact the values mapped onto the temporal axis of the performance (f_i').

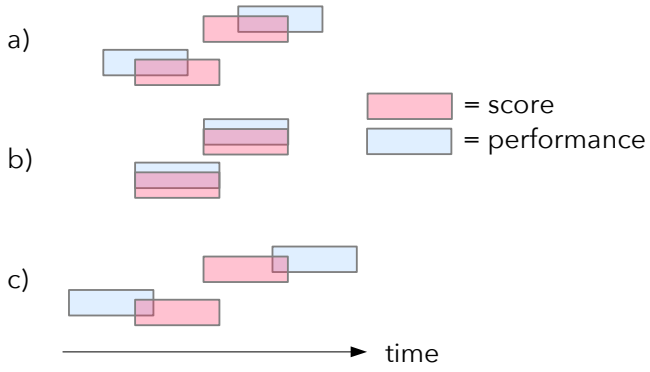

Figure 3: An illustration of onset deformation. a) original performed onset values with $E_{fgp} = 1$, b) full deformation towards the score with $E_{fgp} = 0$, c) exaggerated performance with $E_{fgp} = 2$.

expressive timing, dynamics, and tuning. To control all of these aspects simultaneously and to the same degree, we can introduce a *global expressivity* parameter E which is directly mapped to all desired specific expressivity parameters E_{fgp}.

We can also define more adventurous deformations by pairing not directly analogous features and parameters, e.g. onset features and the amplitude parameter, which has the effect of dynamically emphasizing performed events depending on their temporal distance from the score. However, none such pairings are used in the current application prototype.

4.2 Highlighting Musical Characteristics

Another way of transforming a particular performance, one use case of which was demonstrated in [22], is a transformation based on a given set of feature values, which can be used to highlight particular characteristics of either the performance or the score, as opposed to their relationship as described in the previous section. This way, we can for example emphasize certain types or groups of events by increasing their amplitude or spatializing them. These can be melodic lines, the hands of a pianist, instruments, harmonic content, or metrical values, as long as they are annotated within the score or the performance representations, as described in Section 3. However, none of these techniques are yet applied in the current prototype.

4.2.1 Proportional Scaling by Feature Values

For any feature $f \in F_S \cup F_{P_k}$, we can define a proportional scaling of performance feature $g_i \in F_{P_k}$ and parameter $p \in P_{P_k}$ as follows:

$$p_i = g_i + H_{fgp} f'_i$$

where H_{fgp} is the high-level parameter representing the scaling factor.

This way we can for instance highlight dynamic values of the score or the performance, or deform pitch space around a given reference value.

4.2.2 Other Functions

By using different functions and combining several features, we can achieve more dramatic effects, such as in the example described in [22], where we arranged all note events based on their symbolic pitch values on a three-dimensional spiral[10] around the listener. We did this using the following function for the binaural spatial dimension parameters $x, y, z \in P_{P_k}$ and the pitch feature $p \in F_S$:

$$(x, y, z) = (cos(2\pi p'), sin(2\pi p'), p/12)$$

with $p' = (p \bmod 12)/12$. Even though no such transformations are featured by the current implementation of the tool presented in this paper, we may include similar sonifications in future versions.

4.2.3 Selecting Events with Constraints

Instead of transforming all the events of a performance, we can also choose to select a particular subset by defining an appropriate constraint that limits the objects that are being mapped to by the high-level parameter. For instance, we can define a constraint that selects all downbeat events with metrical position 1, a typical score feature. This allows us, for example, to emphasize the beginnings of bars by increasing the amplitude of these particular events. The same type of constraint can be used to emphasize different voices or instruments, or particular harmonic entities if they are annotated in the score representation.

4.3 Interpolating Between Performances

The third type of parameters we define allows us to interpolate between a selection of performances. In the simplest case with just two performances P_1 and P_2, two parameters $p \in P_{P_1}$ and $q \in P_{P_2}$, two features $f \in F_{P_1}$ and $g \in F_{P_2}$, and an interpolation parameter I_{pq} we define a constraint

$$p_i = q_i = I_{pq} f_i + (1 - I_{pq}) g_i$$

This means that the performed value of the given features will be the same for all corresponding events in the two recordings. We can do this for any feature/parameter pair, but it makes most sense to combine analogous pairs as we did in Section 4.1. For example, by choosing all types of onset, we can interpolate between the agogics of the two performances. Again, it is useful to combine multiple interpolation parameters under one global parameter I, analogous to the parameter E defined in Section 4.1, we can generate intermediate versions of performances.

In practice, in order to smoothly blend the audio samples of the two performances, we also need to interpolate all their amplitudes. We can do this by defining a cross-fade parameter as described in [24]. This parameter is then also controlled by the global interpolation parameter I.

We can similarly interpolate between more than two performances. This lets us create average performances which contain the commonalities of multiple performances. This can be useful to study general performance norms of a particular piece, such as for instance calculated in [14] in terms of normative tempo.

5. THE WEB APPLICATION

In order to test our ideas we built an Angular application[11] using the Semantic Player core library and our Music Visualization library which uses D3.js[12]. We chose a small set of piano performances to test the functionality and decomposed

[10] an immersive Drobisch/Shepard chroma helix
[11] http://angular.io
[12] https://github.com/florianthalmann/music-visualization

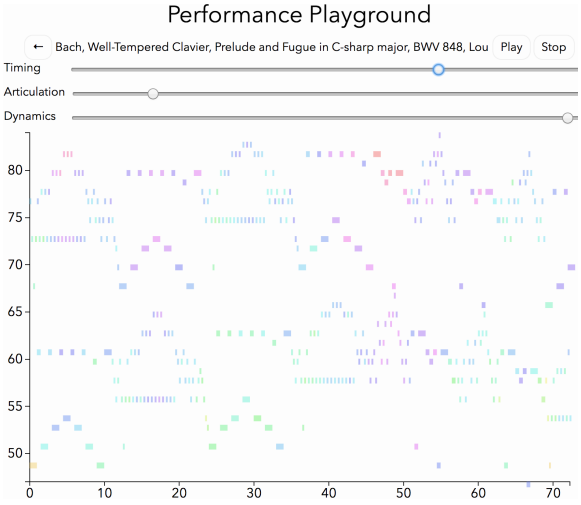

Figure 4: A screen shot of the web app prototype showing a particular performance with a sliders to deform time (onset), articulation (duration), and dynamics (loudness).

and represented them structurally as described above. The dymo-core library allows us to generate, save and load, and play back such structures, and tie our interface elements to the constraints defined to deform the performances. At this stage, all of the structures are pre-generated using the scripts described in Section 3. However, later on, for more advanced use cases, we will also be able to generate new structures and constraints on the fly depending on the user's specific needs.

Currently, as a simple proof of concept, there are two application modes. In the first mode, users can select a *single performance* and using four sliders, each for an individual expressivity parameter for time, articulation, and dynamics, they can deform the selected performance (with slider ranges $E_{fgp} \in [-1, \ldots, 2]$). Figure 4 shows the application in the single-performance mode. A simple piano roll representation illustrates the note events of the performance and keeps track as they are deformed. The x and y axes represent time and pitch, respectively, and the color of each note shows its current dynamic value.

In the second mode, a user can use a slider to blend two performances of the same piece and can thus create interpolated versions between them – see also Section 4.3. Again, a visualization shows the current interpolated performance. In the near future, we plan to add an advanced mode with more options, where users can define additional constraints such as the ones described in Section 4.2 in a simple way by selecting a feature pair and a parameter to map to.

6. CONCLUSION AND OUTLOOK

We presented the underlying concepts of a web application for the discovery of musical performances, based on note-level source separation applied to real recordings. Using the simple prototype developed in the context of this paper we are now planning on running user studies with both laypersons and performance students, in order to determine the usefulness of sonification-by-deformation, as opposed to simple alignment and juxtaposition as proposed in other tools. With more complex and randomized constraints mapping user interface elements to the deformation parameters, we could also investigate the individual listeners' preferences for differing degrees of expressivity.

Furthermore, we are going to experiment with ways of representing and deforming larger sets of performances – the current implementation is limited to two at once. We could thus calculate averages or norms of sets of performances as described in 4.3 and highlight an individual performance's divergences from the norm. Also, in [15] performance students expressed interest in exploring their progress over time, which could be realized using our model by interpolating between subsequent elements in a sequence of performances.

After further experiments, and once the application is more mature, a version of it could be created specifically for larger public archives or library collections, where users could explore and compare historical performances.

7. REFERENCES

[1] E. F. Clarke. The perception of expressive timing in music. *Psychological research*, 51(1):2–9, 1989.

[2] E. T. Cone. *Musical form and musical performance*. W. W. Norton and Co., New York, 1968.

[3] D. Damm, C. Fremerey, F. Kurth, M. Müller, and M. Clausen. Multimodal presentation and browsing of music. In *Proceedings of the International Conference on Multimodal Interfaces (ICMI)*, pages 205–208, Chania, Crete, Greece, 2008.

[4] S. Dixon and G. Widmer. MATCH: A music alignment tool chest. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 492–497, London, GB, 2005.

[5] C. Drake and C. Palmer. Skill acquisition in music performance: relations between planning and temporal control. *Cognition*, 74(1):1–32, 2000.

[6] S. Ewert and M. Müller. Estimating note intensities in music recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 385–388, Prague, Czech Republic, 2011.

[7] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley. Score-informed source separation for musical audio recordings: An overview. *IEEE Signal Processing Magazine*, 31(3):116–124, May 2014.

[8] S. Ewert and M. B. Sandler. Structured dropout for weak label and multi-instance learning and its application to score-informed source separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2277–2281, New Orleans, USA, 2017.

[9] C. Fremerey, F. Kurth, M. Müller, and M. Clausen. A demonstration of the SyncPlayer system. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 131–132, Vienna, Austria, 2007.

[10] W. Goebl and G. Widmer. Unobstrusive practice tools for pianists. In *Proceedings of the International Conference on Music Perception and Cognition (ICMPC)*, pages 209–214, 2006.

[11] E. Gómez, M. Grachten, A. Hanjalic, J. Janer, S. Jorda, C. F. Julia, C. Liem, A. Martorell, M. Schedl, and G. Widmer. Phenicx: Performances as highly

enriched and interactive concert experiences. In *Proceedings of the Sound and Music Computing Conference (SMC)*, 2013.

[12] N. Harley. An ontology for abstract, hierarchical music representation. In *Demo at the International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015.

[13] M. Harris, A. Smaill, and G. Wiggins. Representing music symbolically. In *Proceedings of the Colloquio di Informatica Musicale*, pages 55–69, Venice, 1991.

[14] T. Hoshishiba, S. Horiguchi, and I. Fujinaga. Study of expression and individuality in music performance using normative data derived from midi recordings of piano music. In *International Conference on Music Perception and Cognition*, pages 465–470. Citeseer, 1996.

[15] V. Konz and M. Müller. Introducing the Interpretation Switcher interface to music education. In *Proceedings of the International Conference on Computer Supported Education (CSEDU)*, pages 135–140, Valencia, Spain, 2010.

[16] A. Lerch. *Software-based extraction of objective parameters from music performances*. PhD thesis, Technical University of Berlin Berlin, Germany, 2008.

[17] O. Mayor, J. Llop, and E. Maestre Gómez. RepoVizz: A multi-modal on-line database and browsing tool for music performance research. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[18] C. Palmer. Mapping musical thought to musical performance. *Journal of Experimental Psychology: Human Perception and Performance*, 15(2):331–346, 1989.

[19] K. Riley-Butler. Teaching expressivity: An aural-visual feedback-replication model. In *ESCOM 10th Anniversary Conference on Musical Creativity, April 5–8*, 2002.

[20] J. Rink. *Musical performance: a guide to understanding*. Cambridge University Press, 2002.

[21] S. W. Smoliar, J. A. Waterworth, and P. R. Kellock. pianoFORTE: A system for piano education beyond notation literacy. In *Proceedings of the ACM International Conference on Multimedia (MM)*, pages 457–465, 1995.

[22] F. Thalmann, S. Ewert, M. Sandler, and G. A. Wiggins. Rendering decomposed recordings spatially – integrating score-informed source separation and semantic playback technologies. In *Demo at the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, Malaga, Spain, 2015.

[23] F. Thalmann, A. Perez Carillo, G. Fazekas, G. A. Wiggins, and M. Sandler. The mobile audio ontology: Experiencing dynamic music objects on mobile devices. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC)*, pages 47–54, Laguna Hills, CA, 2016.

[24] F. Thalmann, A. Perez Carillo, G. Fazekas, G. A. Wiggins, and M. Sandler. The semantic music player: A smart mobile player based on ontological structures and analytical feature metadata. In *Web Audio Conference WAC-2016*, Atlanta, GA, 2016.

[25] S. Wang, S. Ewert, and S. Dixon. Compensating for asynchronies between musical voices in score-performance alignment. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 589–593, Brisbane, Australia, 2015.