

DEZRANN, A WEB FRAMEWORK TO SHARE MUSIC ANALYSIS

Mathieu Giraud

CRIStAL, CNRS, Univ. Lille, France
mathieu@algomus.fr

Richard Groult

MIS, UPJV, Amiens, France
richard@algomus.fr

Emmanuel Leguy

CRIStAL, Univ. Lille, France
emmanuel@algomus.fr

ABSTRACT

Music analysis on traditional scores is often based on *annotated* elements, such as patterns, harmonies, or sections. Music students, teachers, players, or researchers are used to annotate music and to discuss these analyses. Music lovers, even when they do not read music, also frequently talk about music and share their reaction to specific sections. We present Dezzrann, an open-source web platform for music annotation and analysis on scores in traditional notation, developed with the Web Components through the Polymer framework. Dezzrann enables to view, edit, and share music analysis through sets of *labels* on a score or a waveform. Labels are linked to musical positions and possibly to voices of the score, and can have a duration or not. They can be created or edited with simple mouse or finger gestures. A public server is available to test the application on Bach fugues and chorales as well as on Mozart string quartets.

1. INTRODUCTION

1.1 Music Annotation and Web Scores

Music analysis can be seen as “casting light upon music” [1], talking about music, possibly with an historical, comparative or aesthetic perspective. Any style of music can be analyzed – oral, notated traditionally or with other means, possibly stored with some electronic material – and hearing and feeling the music may be the first action of an analyst [2]. Music analysis is however well-established on *scores* in common music notation. Analyses on such scores are often grounded on *music annotation*, labeling on the score concepts such as patterns, harmonies, or sections.

How can digital tools help *visualize and annotate traditional scores*, or, more generally, to *talk about music*? One may add comments on scores by using *score editors and engravers* – either commercial ones, or open-source projects such as Lilypond¹, which puts a strong emphasis on music typography. By the way, many music teachers or scholars use score editors to prepare annotated materials for their talks or their courses.

¹lilypond.org

Today, scores can directly be rendered and heard from a web browser, with some projects such as Guido [3], and, more recently, VexFlow², MaxScore with NetCanvas [4], and, developed in the MEI initiative, Verovio [5]. Sending music notation across web notation applications was studied by [6]. Web scores are also proposed together with for-profit services, for example by NoteZilla³ or JellyNote⁴.

Music annotation elements can be encoded in platforms such as iAnalyse [7] or INScore [8, 9]. Toolkits for computational music analysis such as Humdrum [10] or music21 [11] may also output annotated scores [12]. The MEI consortium put lots of efforts into modeling not only music but annotations and metadata, and platforms built on Verovio can offer both engraving and annotation capabilities, such as with the Verovio Humdrum Viewer [13].

These software may give access to other representations than scores, such as waveforms, and some allow exporting videos with scores, or even representing interactive scores. Other projects focus on providing annotated music content, helping the discovery of music concepts for music lovers, such as *Guides d'écoute de la Philharmonie de Paris*⁵.

1.2 Who Needs to Annotate Music?

People may need to talk about music, or to annotate actual music content, notated on scores but not always:

- music pedagogues, teaching either in basic music education or in more specialized classrooms such as analysis, harmony, or composition;
- pupils or students of these classes, engaging into active participation [14];
- researchers in musicology, or in music information retrieval (MIR), more precisely in computational music analysis (CMA), focusing on manual, automated, or semi-automated music analysis;
- performers studying scores and preparing their own interpretation;
- general public or music lovers, sharing their reaction to music.

²vexflow.com

³notezilla.io

⁴jellynote.com

⁵media.citedelamusique.fr/guide-ecoute

The screenshot displays the Dezrann web application interface. At the top, a header bar shows the piece title "Fuga #02 in C minor, BWV 847 J.-S. Bach" and the analysis name "my-analysis". Below this, the musical score is presented on three staves. Various segments are highlighted with colored boxes and labels: "M" (orange), "R1" (orange), "R2" (orange), "CS1a" (blue), "CS1b" (blue), "CS2" (green), "CS3" (red), "S" (blue), "P" (orange), "I:IAC" (blue), "I:PAC" (blue), and "Tonic pedal" (grey). The bottom panel is a table for editing labels, with columns for Type, Tag, Comment, Start, Duration, and End. The selected label is a Subject S starting at measure 26+5/8, with a duration of 2-1/8, and ending at measure 28+2/4.

Type	Tag	Comment	Start	Duration	End
S	S		26+5/8	2-1/8	28+2/4

Figure 1. Current Dezrann web application (`<dezrann-app>` component, here loaded with a `<dez-music-canvas type="score">` component, see Section 3.1) showing an extract at the end of a Bach’s fugue in C minor annotated with thematic patterns (subject S in blue, and counter-subjects CS in green and red) as well as cadences (IAC, PAC) and other labels. Such fugues were written for the keyboard and are usually presented on two staves. Here the three voices of the fugue are presented onto separate staves to easily label voice-related elements. The lower panel enables to edit the selected label, here a fugue subject starting at the 6th eighth (5/8) of the measure 26. The duration of the label, 2 measures minus one eighth, is computed including the first note but excluding the last note. The top panel displays piece information and enables to save or export the analysis, named here “my-analysis”.

Most people are not experimented music readers. However, everyone likes to talk about music: A platform to discuss music should thus give convenient access to several representations (score, waveform) as well as audio rendering – such as the social network SoundCloud⁶ that enables to annotate instants in waveforms.

1.3 Motivation and Contents

The software and platforms listed in the previous paragraphs have some strong points, and many of them bring innovation for traditional scores, and far beyond for some. Nevertheless, there has not been any open-source platform designed to *annotate scores with traditional notation* that is easily accessible on the web, enabling to visualize, to compare, to edit, and to save elements involved in music analysis. For example, researchers in MIR/CMA designing music annotation algorithms frequently need to “get back to the score” and to compare the output of their algorithms to actual music, or sometimes to existing analyses. This can be painful without adapted tools.

We present here Dezrann, an open-source web platform to view, to edit, and to share “music analyses” as annotations on scores in common musical notation, but also on waveforms. Dezrann was designed to help discuss music between people: We do not develop a music editor or engraving software – score rendering is a challenge already tackled by existing software – and focus here on the *anno-*

tation part, allowing users to label music elements, either on a score or a waveform.

In the following, we thus present Dezrann from the point of view of users and developers (Sections 2 and 3), we detail the availability of the platform, the associated corpora, and the development roadmap (Section 4), and we conclude (Section 5).

2. USER PERSPECTIVE

Dezrann means “analysis” in Breton language, a Celtic language spoken in Brittany. Working with Dezrann is free and requires no software installation – the platform works with recent web browsers, at least with recent Safari, Firefox, and Google Chrome.

The user starts by selecting a piece inside available corpora (Figure 3) or by giving a URL. She lands on a “raw” score, or, when it is available, on an annotated score. For example, `dezrann.net/#/bwv847` leads the user to the Bach fugue in C minor BWV 847 with a pre-loaded analysis (Figure 1). The score is here split into three linear staves spanning the whole piece. The user can browse and possibly hear the score, and navigate where she wants.

To annotate music, she may click or point anywhere on the score to add or edit *labels*. Labels, as we defined in [12], are annotations on one or several staves that may have a duration, or not, such as patterns, cadences, harmonic markers, or structural elements. The creation of labels is done by *gestures*. Two gestures are now implemented:

⁶soundcloud.com

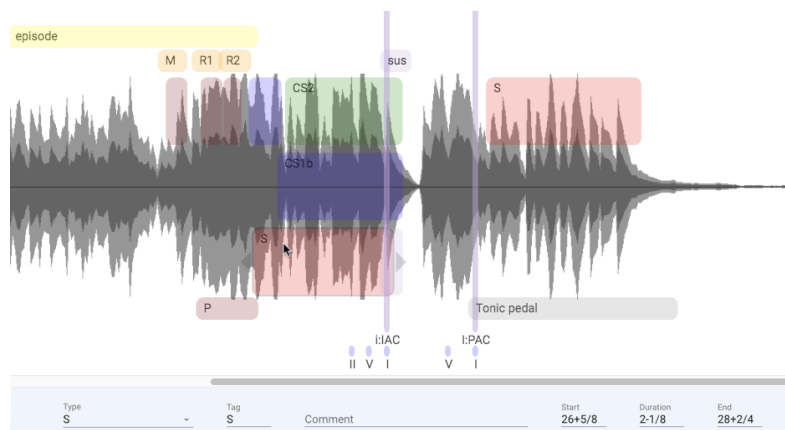


Figure 2. Annotation on a waveform through a `<dez-music-canvas type="wave">` component. The same extract than Figure 1 is displayed, with the same selected label. Labels are here synchronized both to audio time and to musical symbolic time, allowing to see or edit these values in musical time. The invisible grid for editing the labels is here on beats, although more precise values can be entered through the label editor in the bottom panel.

- A left-to-right gesture creates a label with some duration. Such labels can be created either on a staff, to highlight a musical element in one voice, or on the spaces above or below the score, for example to annotate concepts related to the structure or the overall harmony;
- A top-to-bottom gesture across the staves creates a label with no duration, that can be used for example for cadences or as a section marker.

The user can also create labels by using buttons (Figure 4) corresponding to label presets. These labels can be created on the fly while the music plays.

Labels coordinates (start, end, and duration) are shown in symbolic *musical time*, relatively to bar and beat. They can be edited:

- Either on the score or the waveform, by dragging them or by extending them to the left or to the right. These operations snap on an invisible grid built on beats, or on onsets and offsets of the notes appearing in the score;
- Or by editing text fields, entering values such as 23 (first beat of measure 23), $25-1/4$ (one beat before the measure 25, works even on ternary measures), or arbitrary float values such as 26.75 or even 26.71, allowing more flexibility in label placement.

When an audio file and its associated synchronization data are both available, the user can switch back and forth between the score view and the waveform view (Figure 2), still preserving the labels with their musical time. The *analysis*, that is the annotation made as a set of labels, can be saved for later usage and shared with other people.

The user may share links to scores through permanent URLs. Going to dezrann.net/#/bwv847#28.5 jumps to the extract displayed on Figures 1 and 2, around the half of the measure 28 of Fugue BWV 847 in C minor – this is where there is an incomplete authentic cadence concluding the *strettos* and preceding the final cadence. Discussing this passage inside her talk or lesson, the user may thus share a link to his audience.

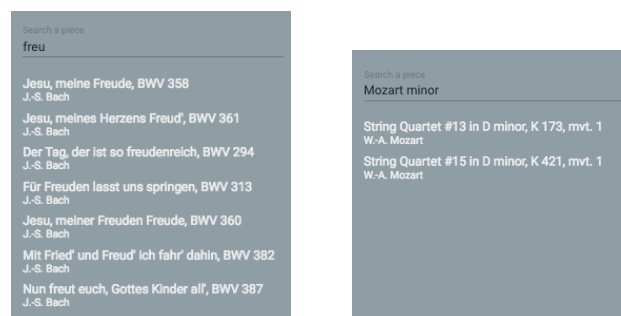


Figure 3. Corpora querying from `<dez-corpus>`.



Figure 4. Label bars for annotating song structures (top) and fugues (bottom). Each button triggers a label with a preset type, tag and comment. Label presets can be edited by the user.

3. UNDERLYING TECHNOLOGY

3.1 Dezzrann Components

The Dezzrann web client is written in object-oriented JavaScript ES6, and uses the *Web Components* model through Polymer 2.0 framework⁷. Proposed in 2011, Web Components “are a set of features being added by the W3C to the HTML and DOM specifications that allow for the creation of reusable widgets or components in web documents and web applications. (...) The components model allows for encapsulation and interoperability of individual HTML elements”⁸. Within this model, new “components” can be used as HTML tags such as the `<audio>` or `<video>` HTML5 tags. We implemented the following components (Figure 5):

- `<dez-analysis>` stores the set of labels. It can be imported or exported into a .dez json file (Figure 6).

⁷ polymer-project.org

⁸ en.wikipedia.org/wiki/Web_Components

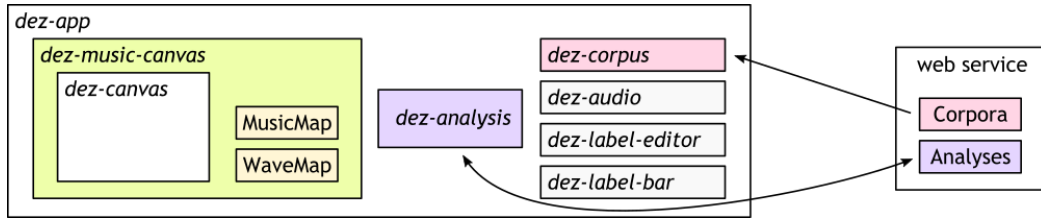


Figure 5. Dezrann architecture, see details in the text. The client side (the *<dezrann-app>* component) communicates with the web service both to get corpus files and to load and store analyses.

```

"labels": [
  { "type": "Structure", "tag": "episode",    "start": 84,    "duration": 18,    "line": "top.1" },
  { "type": "Pattern D",    "start": 96.5,  "duration": 0.5,  "staff": 1 },
  { "type": "Pattern D",    "start": 98.5,  "duration": 0.5,  "staff": 1 },
  { "type": "Pattern D",    "start": 100.5, "duration": 0.5,  "staff": 1 },

  { "type": "Harmonic sequence", "tag": "M", "start": 95.75, "duration": 1.75 },
  { "type": "Harmonic sequence", "tag": "R1", "start": 98,    "duration": 1.75 },
  { "type": "Harmonic sequence", "tag": "R2", "start": 100,   "duration": 1.75 },

  { "type": "CS1",    "start": 104,    "duration": 6.5,  "staff": 2 },
  { "type": "S",      "start": 102.5,  "duration": 7.5,  "staff": 3 },

  { "type": "Pedal", "tag": "I",    "start": 114,    "duration": 10,   "line": "bot.1" },

  { "type": "Cadence", "tag": "I:IAC", "start": 110,    "duration": 0,    "line": "all" },
  { "type": "Cadence", "tag": "I:PAC", "start": 114,    "duration": 0,    "line": "all" },

  { "type": "Degree", "tag": "II",    "start": 109,    "duration": 0,    "line": "bot.3" },
  { "type": "Degree", "tag": "V",     "start": 109.5,  "duration": 0,    "line": "bot.3" },
  { "type": "Degree", "tag": "I",     "start": 110,    "duration": 0,    "line": "bot.3" },
  ...
]
    
```

Figure 6. Dezrann analyses, here represented as a *.dez* json file, are collection of *labels*. These lines correspond to some labels displayed on the Bach fugue on the Figures 1 and 2. Each label has a onset, an optional duration, a type, optional tags and comments, and may have optional staff or line information. Onsets and durations are stored in musical time.

The files can also be read and written through our extension to music21 [12]. Label types are expendable by editing the `lib/LabelTypes.js` file.

- *<dez-canvas>* handles SVG graphical objects on top of the `snap.svg` library⁹ (while being agnostic to the musical significance of these objects), as well as generic mouse/pointer gestures and interaction with these elements for label creation and edition. Rectangles and triangles markers are now defined. Labels are put onto one or several horizontal *lines*, each line being defined by a pair of *y*-positions. The *x*-positions of the labels can be snapped to an optional grid with the graphical positions of the notes.
- *<dez-music-canvas>* embeds a *<dez-canvas>* and handles music content, either a score or a waveform, with proper handling of *x*- and *y*- positions.
 - *<dez-music-canvas type="score">* handles a score image. The default view is now produced by Lilypond extended with scripts borrowed from `ly2video`¹⁰. These scripts output graphical *x*-positions of the notes, linked to their musical times, as well as *y*-positions of the staves.

- *<dez-music-canvas type="wave">* handles a waveform, always with proper handling of *x*- and *y*- positions. Now the regular *x*-positions of the waveform are mapped, through synchronization data of *<dez-audio>* (see below and Figure 7), to the musical *x*-positions.

Both the *score* and *wave* views set the grid of the underlying *<dez-canvas>* component, such that label creation or edition with the mouse or the finger are snapped to musical positions.

- *<dez-label-editor>* enables to see and edit properties of one label. It is focused on the musical time, even linked to *<dez-music-canvas type="wave">*, keeping a symbolic view on audio data.
- *<dez-label-bar>* eases the creation of labels with buttons encoding a given preset (type, tag and comments).
- *<dez-audio>* handles audio output to play .mp3 or other formats through the Web API Audio. It works on its own but is better handled with a manual (or automatic) synchronization file between musical time and audio time (see Figure 7). The synchronization points can be located at each beat, at each measure, or even at only a few places in the piece. Moreover, a MIDI output (processed through `midi.js`, under

⁹ `snapsvg.io`

¹⁰ `github.com/aspiers/ly2video`

```
[
  { "onset" : 0, "time" : 0.45 },
  { "onset" : 40, "time" : 34.40 },
  { "onset" : 80, "time" : 67.91 },
  { "onset" : 117, "time" : 100.89 },
  { "onset" : 118, "time" : 101.91 },
  { "onset" : 119, "time" : 103.01 },
  { "onset" : 120, "time" : 104.18 }
]
```

Figure 7. Synchronization between musical and audio time. Synchronization points can be given at any resolution. In this piece, the tempo is somewhat regular, except for a slow-down at the last four offsets.

development) is available as soon as there is a score, and further allows hearing a selected voice.

- *<dez-corpus>* provides a searchable view on a corpus of pieces (Figure 3).
- *<dezrann-app>* is the single web page application that wraps all previous components to display one piece with one analysis. The analysis is a `.dez.json` file (Figure 6) that can be either saved on the web service or exported. Analyses can be loaded and saved. Moreover, the component recognizes URLs to select a piece and optionally to link to a particular position in the piece by feeding its underlying components.

Components can be used as simple HTML elements, and created and modified with some attributes. For example, a *<dez-music-canvas not-editable>* component displays a read-only annotated score, without any edition or saving capabilities. Note that such a *<dez-music-canvas not-editable>* component fully works on a mobile device and allows scrolling with fingers. Some attributes can be set or changed even after the component has been created. Consider for example *<dez-music-canvas onset=114>*: Setting or changing the *onset* attribute by JavaScript from the outside updates the *<dez-music-canvas>* and jumps to the given offset without reloading the component.

The two synchronizations – graphical score to musical time, and audio to musical time – enable flexible combinations of the components. For example, one can change the score image, using other parameters or even another score rendering engine, and still conserve the synchronization to audio.

Note that *<dez-audio>* can even be used without any reference to any score: Labels are then referenced by their audio time. Should an audio/musical time synchronization be later available, the labels can then be converted back to properly reference the musical time.

3.2 Corpus and Analysis Web Service

A simple web service, written in `node.js`, provides a set of corpora and interactions through AJAX requests (Figure 5, right):

- The *<dez-corpus>* component queries the piece list from the web service. It then feeds the required files for a piece (images, audio, positions and synchronizations) to *<dezrann-app>* and the underlying components;
- For a given piece, *<dezrann-app>* uploads and downloads analyses into the underlying *<dez-analysis>*.

4. AVAILABILITY AND ROADMAP

The web platform with developer and web service documentation¹¹ as well as some corpora¹² are available under open-source licenses (GPLv3+ for the code, ODBL for the data). Components are distributed with usage examples allowing individual reuse in other projects. Tests include 130 unit tests and 40 functional tests using the Mocha toolkit and Selenium^{13, 14}.

4.1 Platform Availability and Roadmap

Following suggestions on a first prototype of Dezzrann [15], the platform evolved, most notably with a complete refactor leading to a clean separation between the *<dez-canvas>* on one side, and the *<dez-music-canvas>* on the other side. This separation helped to implement the *<dez-music-canvas type="wave">* component and to expand the set of labels.

The components and the application can now be used from the public server `dezrann.net`. The MIDI parts of *<dez-audio>* are still under development.

Beside generic improvements to the ergonomics and the features, notable perspectives include:

- *user accounts and flexible access patterns.* Authentication is now limited to a custom installation of Dezzrann behind protected pages. We plan to implement a more generic authentication scheme to run scenarios needing fine-grained access to the analyses through *roles* such as teacher or student. For example, a homework might be assigned, needing first individual or group analysis, then returned and discussed or compared with other analyses such as a reference analysis provided by the teacher.
- *real-time remote collaboration.* We would like to allow simultaneous users to use Dezzrann over the network through the use of `socket.io` library. This could make the platform evolve towards a “social network” to annotate music.

We welcome ideas or contributions from other groups and are ready to help the links between Dezzrann components and third-party code. In particular, new links with automated or semi-automated analysis pipelines could be worthwhile.

¹¹ `dezrann.net/dev`

¹² `algonus.fr/datasets`

¹³ `mocha.js.org`

¹⁴ `seleniumhq.org`

Bach four-voices chorales	2 staves (2 voices each)	181 pieces
Bach fugues, first book of the <i>Well-Tempered Clavier</i>	2 to 5 staves	24 annotated pieces, > 600 labels [16] including 12 with synchronized audio
Mozart string quartets, sonata form movements	4 staves	28 annotated pieces, > 500 labels [18]

Table 1. Corpus and annotations available through Dezrann.

Type	Start	Duration	Tag / comment
Degree	6+1/4	0	ii6/5

Figure 8. Annotation of the start of the four-part Bach chorale “Aus meines Herzens Grunde” (BWV 269), showing degrees coming from the `riemenschneider001` analysis from `music21`, imported by a script based on [12], and completed by cadences, neighbor and passing notes, annotated from within Dezrann. The resulting analysis can be saved or exported for future use.

4.2 Corpus Availability and Roadmap

Table 1 lists the available corpora on the public Dezrann server. As the current graphical capabilities of the platform are to put labels on staves, we focus on music with distinct voices – hence string quartets, or voice-separated fugues. We also included 4-voices chorales laid out on 2 staves, drawing labels on each half of the staff (Figure 8).

More than 200 scores are now available on the public server. New pieces are gradually included, and we target 1,000 scores available by the end of 2018, notably using corpora available through `music21` [11]. Individual pieces or corpora can be added by the `piece.py` and `corpus.py` scripts on the server side. Perspectives include to let the user upload score files (MIDI, MusicXML, MEI, or `**kern`), possibly with audio files.

Dezrann is made in order that people experiment their own annotations and analyses. However, we also gather reference analyses and make them available through the platform:

- There are now analyses of the fugues of the first Book of Bach’s *Well-Tempered Clavier*. Previously published in [16], this dataset was adapted to match Dezrann format by adapting the extension [12] of the python `music21` framework [11].
- Colleagues encoded within Dezrann cadences and structural markers in Mozart string quartets [17] in an ongoing study on sonata forms [18], allowing us to benefit from their feedback in developing the platform.

5. CONCLUSION

“Look at the chord progression at measure 42!” Such a comment – sometimes hard to follow – is often heard in music analysis or harmony classrooms, in music lectures, when discussing on music scores between colleagues, or when practicing into a band. Even if we love paper scores, pencil and erasers, even if we love music teaching through oral practice, we believe that new numeric tools may be pertinent to navigate easily through scores, to share people’s view, possibly different, on a same score, and to benchmark or discuss results from automated or semi-automated software. We designed Dezrann in this spirit, hoping to propose a modern and efficient way to talk about music encoded onto traditional notation.

We welcome comments or contributions from developers, users, music and MIR researchers, teachers, students, or pupils. to make the platform evolve for everyone needs, possibly with some custom components. Tests in music classrooms of a secondary school (age 10–13) are scheduled in relationship with the *inspection académique* (local education authority) supervising music teachers in Amiens district.

Acknowledgments

This project is partially funded by French CPER MAuVE (ERDF, Région Hauts-de-France) and by a grant from the French Research Agency (ANR-11-EQPX-0023 IRDIVE). We thank all users of Dezrann for their feedback and suggestions.

6. REFERENCES

- [1] I. Bent and W. Drabkin, *Analysis*, 1987.
- [2] F. Delalande, *Analyser la musique, pourquoi, comment ?* INA, 2013.
- [3] H. H. Hoos, K. A. Hamel, K. Renz, and J. Kilian, “The GUIDO music notation format,” in *Proceedings of the International Computer Music Conference (ICMC'98)*, Ann Arbor, USA, 1998, pp. 451–454.
- [4] B. Carey and G. Hajdu, “Netscore: An image server/client package for transmitting notated music t. browser and virtual reality interfaces,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'16)*, Cambridge, UK, 2016.
- [5] L. Pugin, R. Zitellini, and P. Roland, “Verovio: A library for engraving MEI music notation into svg,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 107–112.
- [6] S. James, C. Hope, L. Vickery, A. W. B. Carey, X. Fu, and G. Hajdu, “Establishing connectivity between the existing networked music notation packages,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, A Coruña, Spain, 2017.
- [7] P. Couprie, “iAnalyse : un logiciel d’aide à l’analyse musicale,” in *Actes des Journées d’Informatique Musicale (JIM'08)*, Albi, France, 2008, pp. 115–121.
- [8] D. Fober, C. Daudin, Y. Orlarey, and S. Letz, “Partitions musicales augmentées,” in *Actes des Journées d’Informatique Musicale (JIM'10)*, Rennes, France, 2010.
- [9] G. Lepetit-Aimon, D. Fober, Y. Orlarey, and S. Letz, “INScore expressions to compose symbolic scores,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'16)*, Cambridge, UK, 2016.
- [10] D. Huron, “Music information processing using the Humdrum toolkit: Concepts, examples, and lessons,” *Computer Music Journal*, vol. 26, no. 2, pp. 11–26, 2002.
- [11] M. S. Cuthbert and C. Ariza, “music21: A toolkit for computer-aided musicology and symbolic music data,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, Netherlands, 2010, pp. 637–642.
- [12] G. Bagan, M. Giraud, R. Groult, and E. Leguy, “Modélisation et visualisation de schémas d’analyse musicale avec music21,” in *Actes des Journées d’Informatique Musicale (JIM'15)*, Montréal, Canada, 2015.
- [13] C. S. Sapp, “Verovio Humdrum Viewer,” in *Music Encoding Conference (MEC)*, Tours, France, 2017.
- [14] E. Venn, “Watch this space: On blended learning and music analysis in the classroom,” in *European Music Analysis Conference (EuroMAC)*, Strasbourg, France, 2017.
- [15] E. Leguy, R. Groult, and M. Giraud, “Une application web interactive pour l’analyse musicale,” (poster) in *European Music Analysis Conference (EuroMAC)*, Strasbourg, France, 2017.
- [16] M. Giraud, R. Groult, E. Leguy, and F. Levé, “Computational fugue analysis,” *Computer Music Journal*, vol. 39, no. 2, 2015.
- [17] M. Flothuis, *Mozarts Streichquartette: Ein musikalischer Werkführer*. C. H. Beck, 1998.
- [18] L. Bigo, L. Feisthauer, M. Giraud, R. Groult, E. Leguy, and F. Levé, “Learning sonata form structure on Mozart string quartets,” *in preparation*.