VideApp$_{Arr}$ – An Advanced Diff Tool for MEI

Kristin Herold, Beethovens Werkstatt (herold@beethovens-werkstatt.de)
Dr. Johannes Kepper, Beethovens Werkstatt (kepper@beethovens-werkstatt.de)
Ran Mo, Beethovens Werkstatt (mo@beethovens-werkstatt.de)
Agnes Seipelt, Beethovens Werkstatt (seipelt@beethovens-werkstatt.de)

Beethovens Werkstatt seeks to explore compositional processes from different perspectives. In its recently completed second module, the project dealt with a number of Beethoven's works that the composer re-arranged for other performing forces. For these works, printed editions of both the original works and their respective arrangements were fully encoded in MEI, following a rather plain style, i.e. no typographical or genetical details about the sources were preserved. Instead, an additional file per comparison, with merely more than pointers to the files, was provided. With these, it became possible to automatically align both files and present them in various ways with an application called VideApp$_{Arr}$.[1] A first perspective was a plain synoptical alignment of both versions in one score, with the original version on top, and the re-arranged version below. Here, the user may tint individual notes and rests with a limited palette of eight different colors. This provides an admittedly low-level mechanism for annotating the versions compared.

Most other perspectives provided with the VideApp$_{Arr}$ are based on this plain comparison and follow the same general layout. This certainly applies to the so-called "Single Note Comparison". Here, notes are already colored, according to the results of an automated comparison of both versions. For every note in the score, a set of XSLTs identifies notes in the other version, which are either invariant, variant or different, based on a defined set of criteria. It will look strictly at the same position in the piece only, i.e. notes need to sound at the same time in order to be considered for comparison.[2] The three parameters compared are pitch class, octave, and duration of the note. If all of these parameters match, then a note is considered to be invariant. Variance is evaluated as four different categories: octave shifts, differing rhythms (the note in the other version has a longer or shorter duration), the combination of octave shift and differing rhythms, and a mere match of the duration, while the pitch class differs. Everything else is considered to be different. No attention is paid to voice leading or similar aspects, as this isn't reliably available in all combinations of works.[3] However, if the arrangement is in a different key than the original work, the comparison will operate independent of that shifted key, and will compare scale degrees instead of pitch classes. This also considers octave shifts happening at different scale degrees in different keys. The VideApp$_{Arr}$ also allows the user to customize the

---

[1] The software developed was originally labelled the VideApp. As all prototypes of the project will be merged in the future, all applications got that name with an addendum referencing the corresponding project module – VideApp$_{Var}$ for the Variance that the first module dealt with, and VideApp$_{Arr}$ for the arrangements of the second module.

[2] At this point, it is not possible to automatically align

[3] Especially in a piano reduction, "voices" from multiple instruments are condensed in a way that frequently fails to show the same "melodic lines" for middle voices and others, so that the aspect of preceding and / or succeeding notes can hardly be made a default criterium for comparing two arrangements.

comparison: She may optionally select to transpose one of both versions into the key of the other, or to minimize the number of accidentals by transposing to C major / A minor. It is also possible to hide individual staves from both versions in order to focus the comparison on a selection of instruments only.
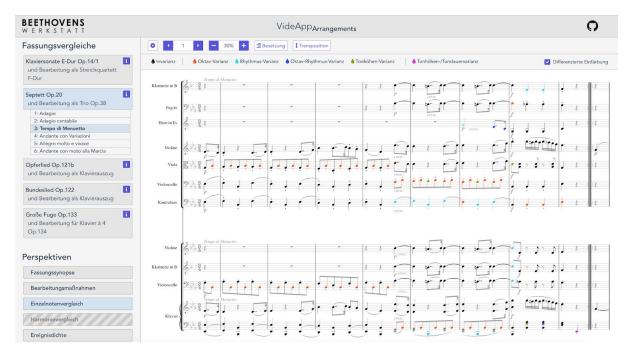

Figure 1: VideApp_{Arr} showing the "Single Note Comparison" of Beethoven's op.20 (top) and op.38 (bottom).

As mentioned above, the MEI encodings used by VideApp_{Arr} are pretty straightforward. They were generated with regular scorewriter applications, and from there converted to MEI. While significant effort was invested into proofreading these files, very little effort went into enriching them with additional information not available from the original scorewriter files. The VideApp_{Arr} turned out to be very helpful for proofreading, as it consequently highlights differences which may or may not be visible from a score rendition alone. This observation led to the idea of opening up the tool beyond the scope of *Beethovens Werkstatt*, and modify it so that users can actually upload their own MEI files. While some of the perspectives offered by the VideApp_{Arr} might be useful for individual files as well, this will be particularly helpful as a Diff tool for MEI instances, either when trying to compare arrangements as in the case of *Beethovens Werkstatt*, or when comparing revisions of a file while editing. Obviously, such a tool has strict expectations on how an MEI file needs to look like in order to work properly, so one might think it would be complicated to use VideApp_{Arr} for other data. However, while the MEI Schema[4] used to validate files is not fully compatible with MEI Basic[5], the level of information allowed (and required) is almost identical. As MEI Basic is supposed to serve as common ground for Common Western Music Notation-based projects, this means that the tool is basically applicable to all files from that

---

[4] https://github.com/BeethovensWerkstatt/module2/blob/dev/data/odd/bw_module2_works.odd.
[5] While this customization isn't released at the time of submission, we expect it to be released by the time of the Music Encoding Conference in May 2020. The current state is available from https://github.com/music-encoding/music-encoding/blob/basic/customizations/mei-basic.xml.

repertoire. Of course, this requires some additional adjustments, not available in the version of VideApp$_{Arr}$ currently available from https://videapp-arr.beethovens-werkstatt.de. This includes an interface that lets the user upload her own files, which will be checked to be valid MEI Basic instances. From there, these files will be automatically converted to match the internal data model used for all further calculations and processes. Some restrictions, of course, still apply. Most significantly, the files to be processed need to be of equal duration, to allow the position-based comparison described above.

Even then, a generalised VideApp$_{Arr}$ will be a major contribution to MEI-based workflows in the CWMN repertoire. We will illustrate the usefulness of that tool by applying it to selected files from the Pasticcio project[6], which, amongst other things, deals with arias customized for various singers.

---

[6] https://www.pasticcio-project.eu