The Diff Concept of the Digital Interactive Mozart Edition

Poster presentation, Music Encoding Conference 2019

Oleksii Sapov, Internationale Stiftung Mozarteum, sapov@mozarteum.at

The *Digital Interactive Mozart Edition* (DIME) provides two types of editions: *Reference Texts* (the musical text of the *Neue Mozart Ausgabe* encoded in MEI) and *Alternative Texts* (entirely new editions of the selected works). In order to show the differences between editions of the same work (a reference text and one or more alternative texts), a specific workflow has been developed. It relies on three strands: the data model, the recognition of the differences, and the rendering of the results.

The premise of the DIME project has an essential impact on the data model: "The digital file is complete in and of itself; it contains all information in terms of musical notation, textual criticism (editorial markup), and metadata." Therefore, an individual edition is not allowed to have any straight reference to other edition (via <app/>, @corresp, etc.). However, it is possible to create a workaround by relating the MEI-elements through the morphology of their IDs. Therefore, a *Reference Text* is copied with some modifications (including the addition of a suffix to any of the IDs[1]). Subsequently, the copied file is used as a 'base' for the edition of the *Alternative Texts*.

The process of recognition of differences is semi-automatic. It is supported by the script diff-DIME.xsl which creates a list of possible differences. However, the core file where the differences are stored — the diff-list — is edited manually. This is important in order to give the editors full control over the process and to be able to add any potentially complicated cases which might not be recognized or were not recognized in a proper way by the script. Additionally, diff-DIME checks if there are any potential errors in the diff-list made by the editor during the manual editing and creates a report; the editor then decides whether if to accept or to decline these error warnings. The whole process is iterative, i. e. changes can be made to the editions or to the diff-list at any time. The diff-list is transformed into app-diff.xml that lists the elements as alternative readings. This file serves as an input for rendering in the DIME graphic interface MOVI.

```
<appDiff xmlns="http://www.mozarteum.at/ns/dme">
  <appGrp xml:id="ref-B1">
    <app xml:id="app_1">
      <rdg n="1" target="dmeref_458-002_5494.xml#m8_k458_002_02#0"/>
      <rdg n="2" target="dmeedtB1_458-002_5494.xml#m8_k458_002_02-B1#dynam_d1e3351-B1"/>
    </app>
```

Figure 1 Alternative readings in the app--diff.xml.

---

[1] E. g. @xml:id="note_123" becomes "note_123-B1". The diff-algorithm relies on this idea of the corresponding elements.

An important aspect is a definition of what should be considered as a difference. Currently, the diff-DIME covers the most essential features like different note pitches, articulation, accidentals, length of slurs, dynamics, etc. A more sophisticated case is a difference like split/merged elements. For instance, instead of one long slur in one edition, the other edition has two short slurs. Child elements of <rdg/>, <orig/>, <sic/> as well as elements which point to them (e. g. slurs) are excluded from comparison.

An important aspect is a definition of what should be considered as a difference. Currently, the diff-DIME covers the most essential features like different note pitches, articulation, accidentals, length of slurs, dynamics, etc. A more sophisticated case is a difference like split/merged elements. For instance, instead of one long slur in one edition, the other edition has two short slurs. Child elements of <rdg/>, <orig/>, <sic/> as well as elements which point to them (e. g. slurs) are excluded from comparison.

Figure 2: An example of the diff rendering in the DIME graphic interface MoVI: split / merged slurs.