Tools for Annotating Musical Measures in Digital Music Editions

Yevgen Mexin, Aristotelis Hadjakos, Axel Berndt, Simon Waloschek
Center of Music and Film Informatics
University of Music Detmold, Germany
OWL University of Applied Sciences, Germany
{e.mexin,a.hadjakos,a.berndt,s.waloschek}@cemfi.de

Anastasia Wawilow, Gerd Szwillus
Human-Computer Interaction and Software Technology Group
Paderborn University, Germany
{apetkau,szwillus}@upb.de

ABSTRACT

The process of creating historical-critical digital music editions involves the annotation of musical measures in the source materials (e.g. autographs, manuscripts or prints). This serves to chart the sources and create concordances between them. So far, this laborious task is barely supported by software tools. We address this shortcoming with two interface approaches that follow different functional and interaction concepts. *Measure Editor* is a web application that complies with the WIMP paradigm and puts the focus on detailed, polygonal editing of image zones. *Vertaktoid* is a multi-touch and pen interface where the focus was on quick and easy measure annotation. Both tools were evaluated with music editors giving us valuable clues to identify the best aspects of both approaches and motivate future development.

1. INTRODUCTION

Musicologists create historical-critical editions of musical works based on autographs by the composer and other source materials, e.g., letters or materials from various copyists and publishers. In order to create a musical score that is both accurate and usable by musicians, the editor has to spot and analyze differences between the different sources, decide which variant to use in the final score, and document the editorial decisions. These decisions are usually described in the critical apparatus, which may be contained in the same volume as the musical score or be published separately. A musician can consult the critical apparatus in order to understand editorial decisions, to examine differences between different editions of the same work, or to spot interesting variants that he or she might choose to play in opposition to the editor's choices.

Digital and hybrid editions, which are published both in digital and printed form, make it easier for a musician to delve into the original materials and understand its inherent ambiguities. On the one hand digital editions are not limited by printing costs and can provide scans of the original autographs, on the other hand suitable human-computer interaction makes it easier to navigate in the complex network of cross-references between various source materials and the editors annotations.

The hybrid edition of Max Reger's works may serve as an example. Two screenshots are depicted in Figure 1. On the left side, the typeset final score is shown together with small icons that represent annotations by the editor. When the user opens an annotation, a window shows the textual annotation as well as the source materials that the annotation refers to (see the lower-left sub-window of the right screenshot in Figure 1). The textual annotation in the depicted example says that one of the sources lacks the fortissimo in measure 3. By clicking on the autograph icons, the corresponding autograph scans are displayed (see the other three sub-windows of the right screenshot). This way, a musician can delve into the piece and understand its heritage and its ambiguities.

Music in common music notation is structured in musical *measures*. Larger works can additionally be split up in several *movements*. Musicians usually refer to specific positions within a score by measure numbers. Measure numbers therefore provide a reasonable granularity for links between textual annotations and the original material. In practice, this makes it necessary to define processable markings for every measure in all related autographs and create concordances, i.e., one-to-one mappings, between these annotation. The XML-based MEI (Music Encoding Initiative) [2] [3] data format, the main format for digital music editions, is able to handle measure definitions in images in the form of polygons. Since musical works can consists of hundreds or even thousands of measures, the annotation of them can, however, be a tedious task. Therefore specialized and time-efficient tools are needed.

We developed two different interfaces: The *Measure Editor* is a web application and part of a full featured music edition platform in development. *Vertaktoid* is a standalone Android Application that supports pen and touch interaction. Both software prototypes were evaluated by music

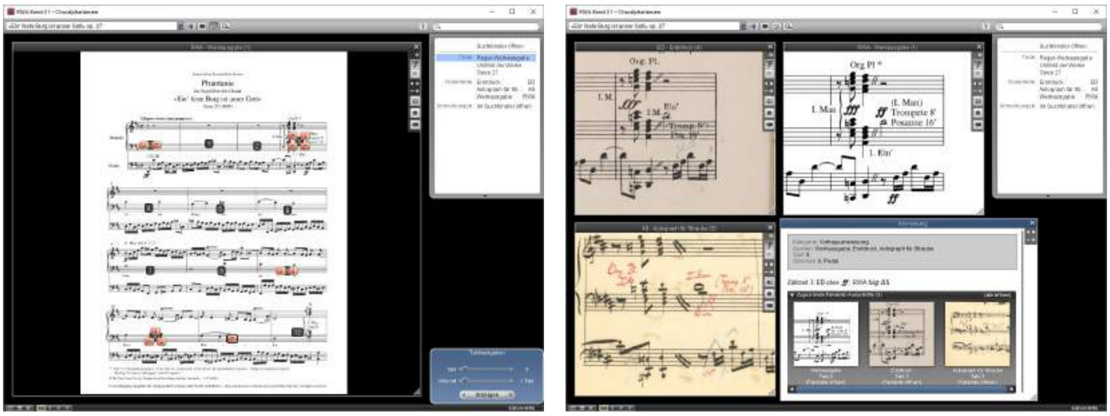


Figure 1. Screenshots from the digital Reger edition [1]

editors, showing potential paths towards an efficient measure annotation tool.

2. RELATED WORK

2.1 Current tools for Measure Annotation

Edirom [1] is a project that aims to assist the music editors providing special software. Two of them are explicitly relevant for this overview: Edirom-Editor [4] and Edirom-Online [5].

Edirom-Editor is a platform that combines different tools for creating digital and hybrid music editions. Edirom-Editor supports the following tasks:

- Cataloging of music works represented by multiple sources with musical and eventual textual content.
- Definition of the music score's structure by specifying the movements and their names.
- Measure annotation with sequence numbers and eventual additional meta-information. The measures will be also aligned to the movements in this step.
- Creation of concordances for the movements and measures that represent the relations between the corresponding elements from different sources.
- Creation of textual annotations and additional references between the elements.
- Export of the created digital music edition in the MEI format.

The MEI format can then be presented and published using Edirom-Online. Edirom-Online can present the music score, the corresponding measures from different sources, the annotations and references. The publishing can be performed both locally and remotely via the web.

Edirom-Editor is well known by music editors, who work on digital and hybrid music editions. Among other projects, Edirom-Editor is being used in the following projects: Detmolder Hoftheater [6], OPERA, [2] and Beethovens Werkstatt [7].

Since measure annotation is the most laborious repetitive task, we developed specialized tools. The development of these tools addressed shortcomings of Edirom-Editor with regard to its usability. In this we see the biggest potential for accelerating the process of measure annotation.

2.2 Measure Detection in Optical Music Recognition

Annotation of measures is a relatively time-consuming process that needs a lot of user attention. In this paper we discuss two tools that facilitate this task by introducing improved interaction techniques and intelligent approaches like automatic numbering (see Section 3.2.2), but the main part of the annotation process remains manual. On one hand, this allows the working process to be free from issues that can be caused by automatic recognition and allows non-standard context-based user decisions. On the other hand, automatic processing of big volume of data counts can be performed much faster.

Optical Music Recognition (OMR) [8] is the automatic conversion of scanned music scores into computer readable data in variable formats, e.g., MusicXML, [3] or MEI. [4] OMR processes the content of music scores, trying to recognize the notation and create a representation that provides the best correspondence to the original source, which is typically scanned. The effectiveness of optical recognition of printed music scores is greater than of hand-written scores, which are very common material for digital music editions. For our purposes the recognition of music notes themselves, including pitch and duration, is not relevant since we are mainly interested in finding measure boundaries. Therefore we will present related work that tackles this task below.

Dalitz and Crawford have investigated the possibility of OMR in relation to the lute tablature. The recognition of measures is discussed in detail in the paper [9]. Although the lute tablature does not rely to common Western music notation, it has a similar measure concept. The staff lines are removed during the preprocessing step and then the algorithm tries to detect the bar line candidates through their *aspect ratio* (ratio of width to height). The candidates can be also a combination of multiple closely located frag-

[1] `http://edirom.de`, last access: Feb. 2017
[2] `http://opera.adwmainz.de`, last access: Feb. 2017
[3] `https://musicxml.com`, last access: Feb. 2017
[4] `http://music-encoding.org`, last access: Feb. 2017

ments. The properties of bar line candidates are validated trough comparison with staff line and staff space parameters. The optical measure recognition for lute tablatures can be performed relatively robustly because they do not contain complex systems of multiple voices such as in other music score.

Vigliensoni, Burllet and Fujinaga have devised an approach for optical measure recognition of music scores in common Western music notation [10]. Vigliensoni et al. follow the task model from Bainbridge and Bell [11] that contains the following steps: image preprocessing and normalization, staff line identification and removal, musical object location and musical reasoning.

The approach by Vigliensoni et al. [10] needs additional information from the user about the structure of staves in music score. This information consists of the count of staves for each page, the count of systems, the relation between staves and systems and the kind of bar lines. During the process of bar line detection, the algorithm has to remove all horizontal lines and filter the resulting set to include only thin, vertical elements. The remaining bar candidates are further filtered by their *aspect ratio* using a user-defined parameter. The algorithm can handle the broken candidates by combining the vertical lines with the nearest horizontal positions. As a final step, the height and position of each bar candidate is compared with the related properties of the corresponding system. The candidates that do not match the system will be also removed. The second user-defined parameter, *vertical tolerance*, controls the sensitivity of the last step. This measure recognition technique was evaluated trough comparison with manually annotated music scores. Vigliensoni et al. have chosen one hundred music score pages from the International Music Score Library Project (IMSLP) [5] and had these pages marked by experienced annotators. The authors compared the these bounding boxes with those recognized by the machine. The results of machine recognition was considered correct if the divergence of the bounding boxes was less than 0.64 cm. The evaluation results yield to an average f-score of 0.91 by the best chosen *aspect ratio* and *vertical tolerance* parameters.

Although a lot of further research papers in OMR thematic field exist, we focus on these two articles because they have contributed largely to optical measure recognition. The approach from Padilla et al. [12] with the clever idea to compare and combine the OMR outputs for different sources of the same music score can be also mentioned. The majority voting approach is used to decide which elements are correctly recognized.

2.3 Interaction

The process of measure annotation comprises three interaction tasks: navigation through autographs, definition of zones (measures, respectively) in the autographs, and editing of zone numbering and movement association. The latter is conveniently solved via text input. Regarding navigation between and within images (autograph pages), standard gestures for panning and zooming are well established and implemented in every multitouch enabled operating system. These will be applied here, too. The definition of zones (typically bounding boxes) is the dominating task. It is essentially similar to the more general task of selecting image regions which is well-researched in the field of human-computer interaction. In Edirom Editor [4] zones are defined by the traditional rectangle spanning gesture that is known, e.g., from graphics editors and file explorers (selecting multiple files by spanning a box around them). Edirom Editor relies on mouse input. Users describe this interaction technique as slow and demanding, requiring a lot of corrections as the initially drawn bounding boxes rarely embrace their content perfectly. The original developers of Edirom Editor obtained this feedback from their own experiences with music edition projects, active communication with other editors and during the yearly Edirom Summer School. This was the situation that motivated the developments which we report here.

As an alternative to mouse input, pen (and touch) seems the most promising modality as it corresponds more closely to the way we naturally work with sheet music. However, there is a variety of selection gestures for pen input, each with its advantages and disadvantages depending on the application context. For some editions it may suffice when zones roughly embrace measures, others may need very precise bounding volumes.

Typical selection gestures in the literature are *spanning a rectangle* (as implemented in Edirom Editor), *lassoing* (well-known from graphics editors) and *tapping* (discrete object selection) [13]. These are sometimes combined with a *mode button* that is operated by the secondary hand to clarify conflicting gestures, phrase multiple strokes to one gesture, and separate the selection process from an operation that is applied to it [14]. Zeleznik & Miller [15] use terminal punctuation to define selection-action phrases. Several phrasing techniques (timeout, button, pigtail) where investigated by Hinckley et al. in [16, 17]. They also summarize that lassoing is the favored selection gesture "because it is well suited to selecting handwritten notes or diagrams, which typically contain many small ink strokes that would be tedious to tap on individually" [16, 17].

Interesting insight into the correlation of precision and quickness of selection gestures is given by Lank et al. [18]. Based on the motion dynamics of a gesture they made an "analysis of deliberateness versus sloppiness" and inferred "a linear relationship between tunnel width and speed, for paths of varying curvature" [18] which helps to automatically refine the selection.

3. TWO APPROACHES

The measure annotation task is a time-consuming process within the work on music editions. In practice, it includes a large number of interaction steps which have to be executed with high precision, putting a heavy load on the user. Hence, in our project we wanted to optimize the user interaction necessary for this task. We decided to follow two parallel paths: we included a measure annotation tool (Measure Editor) in the full featured music edition platform as a web application on one hand, and on the other hand implemented a native tablet tool (Vertaktoid) on an Android system for

[5] `http://imslp.org`, last access: Feb. 2017

the same purpose.

By doing so, we were enabled to compare the web-based solution to a solution dedicated to touch interaction on a popular device. Although the web-based application can be run on any device, it is primarily designed for the big screen on the desk, or even a multi-touch table, hence for a stationary work place. The tablet version on the other hand can be used in any mobile context and could, for instance, be used while working in a library or even during a musical performance.

The two tools worked in different contexts from the beginning:

- The Measure Editor is part of the music edition platform linked to an underlying data base (the back end), which is accessed via the front end representing the user interface. All semantic functions are triggered by the user from the front end. The input data for all operations come from the central data base and results are fed back into it. This includes, for instance, retrieval and storage of autographs, annotated autographs, concordances, and the complete set of possible FRBR [19] data. MEI imports and exports are also implemented through the data base, which implies that data exchange with Edirom is possible through this channel.

- Vertaktoid is a stand-alone prototype written explicitly for Android systems. As such, it can make use of native interaction techniques. The tool is not based on the central data base, but works with local storage on the device. Communication with other tools, including Edirom, is realized with MEI import and export functionality. MEI is also the internal format on which Vertaktoid operates directly.

While the Measure Editor is an integral part of the music edition, Vertaktoid could be taken as prototype to be linked into that environment as a practical tool for special purposes or usage situations. It could eventually work as a component of a distributed user interface approach, side by side with the web application Measure Editor and/or special purpose tools implemented for a multi-touch table. The developing of both tools is the main part of the contribution described in this paper.

3.1 Measure Editor

Measure Editor is a web-based application that is used for annotation of measures in music scores. The editor is supported by most popular browsers, such as Internet Explorer, Chrome, Opera, Firefox and Safari. The user interface has been designed allowing the user to work not only on the desktop, but also on any mobile device. A tablet with a stylus can be very suitable for precise work.

There are a lot of methods available for measure annotation task in the Measure Editor, which can be combined with each other. This provides especially quickly and easy annotation of measures of any shape. Since the measures mostly have a rectangular shape, the user always begins with bounding boxes. Measure Editor on desktop provides

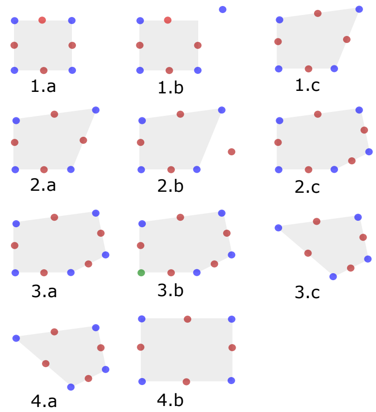


Figure 2. The transformation operations on the polygon.

a spanning of rectangles by mouse as input device. The same task can be performed on the mobile devices by definition of the diagonal of rectangle. So, the user can create a new bounding box with two single clicks. Measures of regular rectangular shape can be easily transformed into polygons. It can be useful when the measure or another object with a complex geometric shape has to be marked. Figure 2 visualizes the transformation of a rectangle to polygon and back.

Each rectangle consists of four vertices that are shown as blue dots. Red dots represent the midpoints of the edges. The user can change the shape and its size by moving its vertices. The first line in the Figure 2, 1.a - 1.c, demonstrates this effect. The drag & drop operation applied to midpoints will create new vertices (2.a - 2.c). The existing vertices can be removed by double-clicking with the condition that the number of vertices remains greater than or equal 3 (3.a - 3.c). The bottom row demonstrates the effect of converting a polygon to rectangle. Through this set of operations, the user can transform simple rectangles to complex geometric shapes.

Another equally important feature of Measure Editor is the pair of functions called *horizontal scissors* and *vertical scissors*. By means of these functions, the annotation of measures in music scores can be very quickly performed. The *scissors* function performs the division of rectangle into two smaller rectangles and so a new measure will be created. The division is either horizontally or vertically depending on the selected function. The new measure becomes automatically a number and inherits the attributes (e.g. movement alignment, suffix, upbeat) from the parent measure.

The GUI of Measure Editor is shown in Figure 3. The largest area is used for the representation of a score which is to be annotated. All functions available in a certain step are to be found in the toolbar. These are e.g. *select* , *move*, *draw*, *edit* a shape, *horizontal* and *vertical scissors*, *convert to rectangle*.

The right side of the GUI contains the list of all measures and the properties view. The user can select a specific measure graphically in the working area or directly as text in the list. The parameters of the selected measure can be set or adjusted in the properties view. For example, the

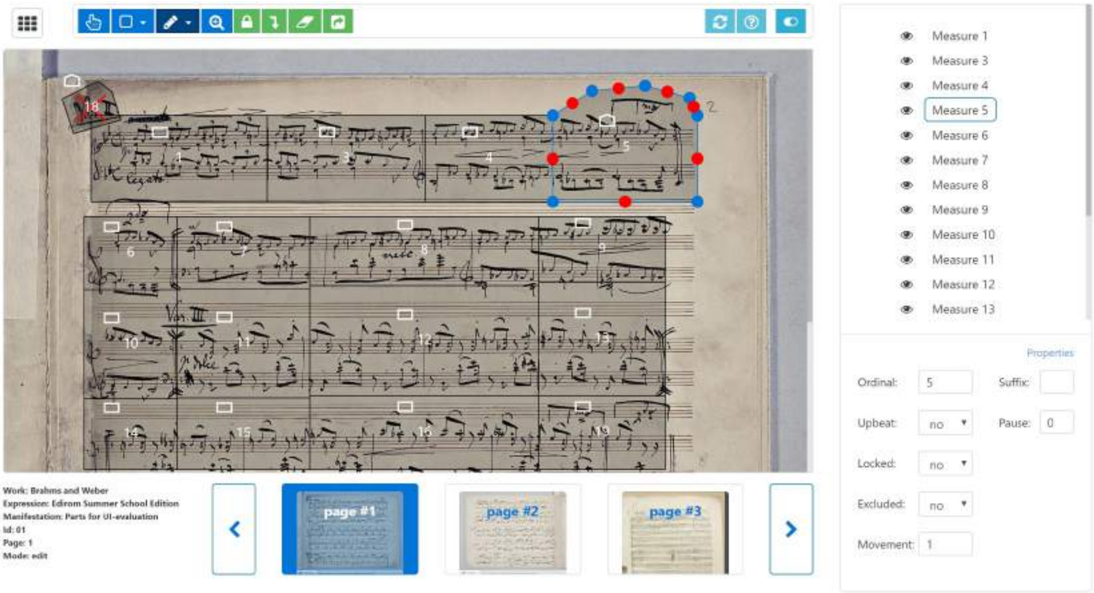


Figure 3. User interface of Measure Editor.

user can change the ordinal number of a measure or the corresponding movement. The other parameters that can be adjusted are:

- *Suffix* - the additional textual information that can be added to the measure name after the ordinal number
- *Upbeat* - describes whether a measure is an upbeat of the parent movement.
- *Pause* - the number that represents the repeated rest in measure. It may be considered during calculating the sequence number of next measure and can have a meaning when defining concordances
- *Locked* - makes the measure uneditable.
- *Excluded* - marks the measure as excluded (not to be considered).

Measure Editor supports the music editors providing numerous features for comfortable and fast annotation of music scores. Inasmuch as measure annotation is a part of the whole process of music edition creation, Measure Editor will be also developed as a part of the complex system that was designed to provide different tools for music editors. Measure Editor is an open source application and is available [6] under LGPL 3.0 license.

3.2 Vertaktoid

Created as an android application for tablets with a pen, Vertaktoid wants to cover a need of mobile user-friendly instruments for comfortable and quickly annotation of measures in handwritten and printed music scores. As mentioned above, the measure annotation is an important and relatively time consuming part of whole creation process for digital music editions. Therefore, the main task of Vertaktoid is to minimize the man-hour costs and, on the other hand, to provide a comfortable interface for precise user-defined measure annotation process. Vertaktoid supports natively the MEI format and can be used in combination with Edirom Editor and Edirom Online programs. Vertaktoid is an open source application and was published [7] under LGPL 3.0 license.

3.2.1 *Marking measures*

To mark measures, the user selects the pen tool and draws an outline of the measure with the interactive stylus. The start of user's stroke is marked by a small circle. When the user closes the outline by reaching that circle again, the interaction is completed and the measure is defined by the rectangular bounding box of the outline. This interaction is very easy to understand and therefore well-suited for users that just start working with Vertaktoid. However, when marking several hundreds measures in a large music piece or even thousands of measures spread over several autographs, the efficiency of the interaction becomes much more important than its learnability. Therefore, Vertaktoid supports two further interaction methods:

- **"Click, don't draw":** If, while drawing the outline, the user lifts the stylus and puts it down at another position, the line from the last to the new position is automatically drawn. Now, the user can continue drawing the outline or click again at another position. When the user reaches the initial position (marked by the small circle) by clicking or drawing, the outline is completed.
- **"Cut, don't repeat":** Oftentimes the user marks several measures that are aligned as they are part of the same grand staff (see, e.g., measures 1-4 in Figure 4).

[6] `https://bitbucket.org/zenmem/zenmem-frontend`, last access: May 2017

[7] `https://github.com/cemfi/vertaktoid`, last access: May 2017

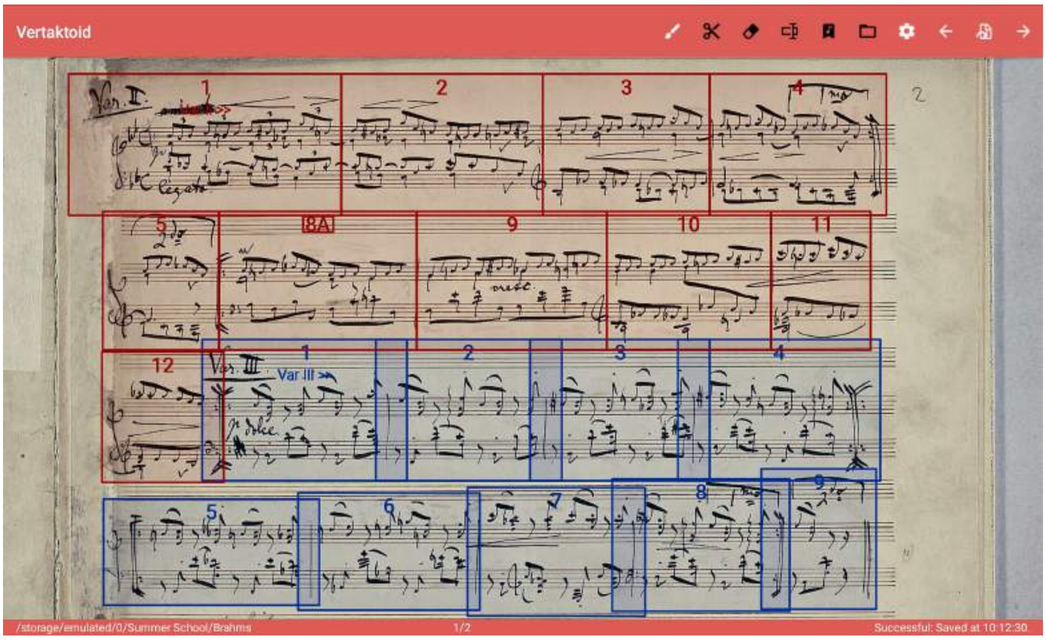


Figure 4. User interface of Vertaktoid.

> Instead of repeating and drawing the outline for each measure separately, the user can outline the entire grand staff and then use the scissors tool to cut the outline in pieces by clicking at the measure bars.

By clicking and cutting, the user can perform the task considerably faster. Consider, e.g., the red Measures 1-4 in Figure 4. With clicking and cutting, a user would typically use ten clicks to mark those four measures: Five clicks would be needed to mark the four corners of the grand staff and close the shape by clicking on the first corner for a second time. Three clicks would be needed for cutting. Further two clicks would be needed for first selecting the pen and then switching to the scissors tool. Sometimes the vertical alignment of the measures is not wanted.

Some music editors prefer to include a small horizontal overlap from the previous and the next measure. This way, the measure does not look isolated when, e.g., a musician views an annotation by the music editor which refers to that measure. Sometimes an overlap is not purely aesthetic but necessary in order to understand the content of the measure, e.g., an overlap may be necessary if a sustained note is written with a tie across the bar line. In order to enable the use of the scissors tool in such situations, the user can configure the amount of horizontal overlap. While the blue Measures 1–4 in Figure 4 were made with an overlap, the red measures were made without. Also if the scissors tool is not able to create the desired result, the user can always define the measure outlines manually (see Figure 4, blue Measures 5–9).

3.2.2 Automatic measure numbering

Another feature of Vertaktoid is automatic measure numbering and coloring. The latter indicates the movement the measure belongs to. The measure numbers are calculated through comparison of the measure positions on the facsimile page and their vertical overlapping with each other.

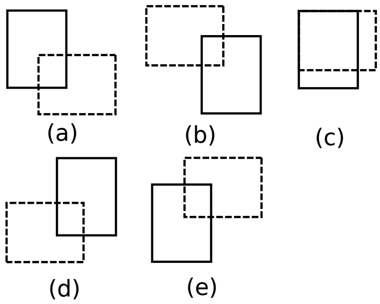


Figure 5. Five cases by measure positions comparison.

When a new measure is added, it is compared to all other measures on the page. In Figure 5, the new measure is depicted as the dashed rectangle while the solid rectangle represents an already existing measure. Figure 5 demonstrates the five possible cases that can occur.

(a) The new measure is located vertically and horizontally after the existing measure.

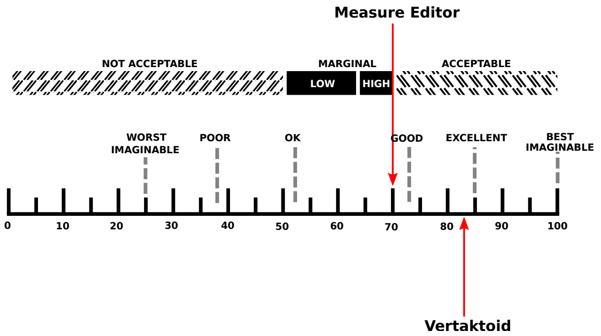
(b) The new measure is located vertically and horizontally before the existing measure.

(c) The new measure and the existing measure have the same start positions.

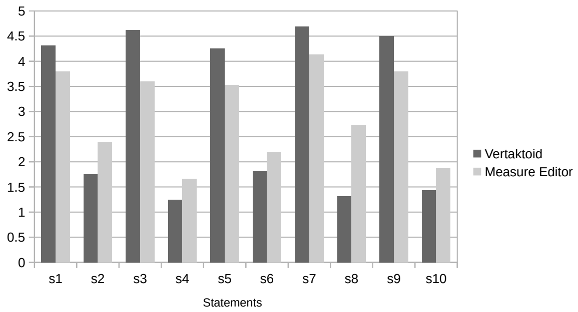
(d) The new measure lies vertically after and horizontally before the existing measure.

(e) The new measure lies vertically before and horizontally after the existing measure.

The first three cases are trivial. In case (a) the new measure will be ordered after the existing, in case (b) - before the existing and in case (c) - the both measures have equal start positions and can be randomly ordered.



(a) The meaning of the SUS Score depicted on adjective ratings and acceptability ranges.



(b) Average spreading of participants votes over SUS statements.

Figure 6. The results of SUS survey.

For the last two cases, a calculation of the vertical overlap is needed. Let M be the rectangle that represents the existing measure and M' - the rectangle that represents the new measure. The rectangles can be defined by two points: ul - upper lefter vertex and lr - lower righter vertex. Each vertex consists of x and y coordinates. Then the vertical overlapping $Over_y$ is calculated as follows:

$$Over_y = \frac{\min(lr_y^M, lr_y^{M'}) - \max(ul_y^M, ul_y^{M'})}{\min(lr_y^M - ul_y^{M'}, lr_y^M - ul_y^{M'})} \qquad (1)$$

$Over_y$ is the vertical intersection of both rectangles relative to the height of the smaller shape. Rectangles, which doesn't have a horizontal intersection, can be also processed. Now that the vertical overlapping factor is calculated, the final order of two examined measures can be decided. The algorithm will take the horizontal order of measures if $Over_y \geq 0.5$ and the reverse horizontal order otherwise. This approach is applied to each measure from the corresponding movement until the measures are sorted by their positions.

Sometimes, such a strict numbering scheme is not appropriate for the musical content, e.g., for upbeat measures, multiple measure rests or measures that are split between two successive pages or staves. In those cases the user can manually assign an arbitrary name for the measure. Such names are marked with a frame (see measure ”8A” in Figure 4). The alignment of measures to the movements can be also changed. Both user-performed operations causes further automatically renumbering of following measures inside of the corresponding movements.

4. EVALUATION & DISCUSSION

To evaluate the usability of both measure annotation tools, a quantitative study was carried out. The study was based on the System Usability Scala (SUS) [20]—a simple and quick survey that can be used to measure the usability of a product or service based on the subjective assessments of participants. The SUS survey contains ten statements, which must be voted with a value in the range from 1 (strongly disagree) to 5 (strongly agree). The statements alternate between positive and negative expressions about the examined software. A simple formula calculates the SUS score from the individual votes. Figure 6(a) shows the meaning of a SUS score translated into adjective ratings [21].

Our study was conducted as part of the Edirom Summer School event [8], which was created to teach musicologists the creation of digital music editions using the Edirom Editor tool and to inform experienced music editors about the new techniques in this field. We recruited sixteen music editors for our study. The familiarity of participants with the music annotation software was heterogeneous: some of them have learned the Edirom Editor during the Summer School, the other part have already a good expirience with it. They used prototype versions of Measure Editor and Vertaktoid and annotated autographs with them. Figure 6 shows the average votes for both applications. The resulting SUS scores were 70 for Measure Editor and 87 for Vertaktoid. This means that both tools were found as acceptable and Vertaktoid even as excellent (see Figure 6(a)).

To decide about future development, we interviewed the study participants. The combination of the survey and these consultations was taken into consideration during the developing of next versions for both applications. Their feedback contains a lot of clever suggestions. So, the cut function can be performed in both dimensions, what can make the annotation process more effective. Some users found the often repeating selection of suitable control elements as uncomfortable. The next control element in some cases can be automatically chosen following the common workflow. The whole annotation process can be also performed in the predefined sequence of steps. Thus, the application can offer the user to annotate the measures with draft bounding boxes that can be refined during the next step.

Furthermore, Vertaktoid is currently used at Detmolder Hoftheater project, which provides continuous helpful feedback for further evaluation and troubleshooting.

A number of features could be added in the future. Optical measure recognition and subsequent automatic annotation would be a helpful and time-saving function at least for annotation of printed music sheets. It would also be productive if several users could simultaneously work on one music score. This would mean that users could access data from a shared database to continue the work of another

[8] `http://ess.uni-paderborn.de`, last access: May 2017

users, make corrections or discuss further work flow. It would also be useful if the users could make additional textual (and not only textual) annotations for individual measures, music notes or even any other positions on the facsimile. These annotations could be considered in the later steps during the creation of a music edition.

Both tools can also contribute to each other and, using the experience of the sibling tool, inherit its best techniques. Thus, the possibility to annotate an area via polygon can be very helpful in Vertaktoid as well as the editing functions. On the other hand, Measure Editor can be improved by additional interaction techniques, movement colorizing or automatic numbering.

5. REFERENCES

[1] M. Reger, *Reger-Werkausgabe, Bd. I/1: Choralphantasien für Orgel*, A. Becker, S. König, C. Grafschmidt, and S. Steiner-Grage, Eds. Leinfelden-Echterdingen, Germany: Carus-Verlag, 2008.

[2] P. Roland, "The Music Encoding Initiative (MEI)," in *Proceedings of the International Conference on Musical Applications Using XML*, 2002, pp. 55–59.

[3] A. Hankinson, P. Roland, and I. Fujinaga, "The music encoding initiative as a document-encoding framework," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami (Florida), USA, October 24-28 2011, pp. 293–298.

[4] Virtueller Forschungsverbund Edirom (ViFE), "Edirom Editor 1.1.26," https://github.com/Edirom/Edirom-Editor, June 2016, accessed: 2017-20-02.

[5] Virtueller Forschungsverbund Edirom (ViFE), "Edirom Online," https://github.com/Edirom/Edirom-Online, accessed: 2017-20-02.

[6] I. Capelle and K. Richts, "Cataloguing of musical holdings with MEI and TEI. The Detmold Court Theatre Project," in *Music Encoding Conference 2015*. Florence, Italy: Music Encoding Initiative, May 2015.

[7] S. Cox, M. Hartwig, and R. Sänger, "Beethovens Werkstatt: Genetische Textkritik und Digitale Musikedition," *Forum Musikbibliothek*, vol. 36, no. 2, pp. 13–20, July 2015.

[8] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 173–190, 2012.

[9] C. Dalitz and T. Crawford, "From Facsimile to Content Based Retrieval: the Electronic Corpus of Lute Music," *Phoibos – Zeitschrift fur Zupfmusik*, pp. 167–185, February 2013.

[10] G. Vigliensoni, G. Burlet, and I. Fujinaga, "Optical Measure Recognition in Common Music Notation," in *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013*, 2013, pp. 125–130.

[11] D. Bainbridge and T. Bell, "The challenge of optical music recognition," *Computers and the Humanities*, vol. 35, no. 2, pp. 95 – 121, 2001.

[12] V. Padilla, A. McLean, A. Marsden, and K. Ng, "Improving optical music recognition by combining outputs from multiple sources," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, 2015, pp. 517–523.

[13] E. Artinger, T. Coskun, M. Schanzenbach, F. Echtler, S. Nestler, and G. Klinker, "Exploring Multi-touch Gestures for Map Interaction in Mass Casualty Incidents," in *INFORMATIK 2011 – Informatik schafft Communities (Proc. of the 41st GI-Jahrestagung)*, H.-U. Heiß, P. Pepper, H. Schlingloff, and J. Schneider, Eds., Gesellschaft für Informatik (GI). Berlin: Bonner Köllen Verlag, Oct. 2011.

[14] K. Hinckley, F. Guimbretiere, M. Agrawala, G. Apitz, and N. Chen, "Phrasing Techniques for Multi-Stroke Selection Gestures," in *Proc. of Graphics Interface 2006*. Canadian Human-Computer Communications Society, June 2006, pp. 147–154.

[15] R. Zeleznik and T. Miller, "Fluid Inking: Augmenting the Medium of Free-Form Inking with Gestures," in *Proc. of Graphics Interface 2006*. Canadian Human-Computer Communications Society, June 2006, pp. 155–162.

[16] K. Hinckley, P. Baudisch, G. Ramos, and F. Guimbretiere, "Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli," in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI 2005)*. Portland, Oregon, USA: ACM SIGCHI, Apr. 2005, pp. 451–460.

[17] Hinckley, K. and Baudisch, P. and Ramos, G. and Guimbretiere, F., "Delimiters for Selection-Action Pen Gesture Phrases," United States Patent No. US 7,454,717 B2, USA, Nov. 2008, filed Oct. 2004.

[18] E. Lank, E. Saund, and L. May, "Sloppy Selection: Providing an Accurate Interpretation of Imprecise Selection Gestures," *Computers & Graphics*, vol. 29, no. 4, pp. 490–500, Aug. 2005.

[19] B. Tillett, "What is FRBR? a conceptual model for the bibliographic universe," Library of Congress, Tech. Rep., February 2004.

[20] J. Brooke, "SUS: a "quick and dirty" usability scale," in *Usability Evaluation In Industry*, P. W. Jordan, I. L. McClelland, B. Thomas, and B. A. Weerdmeester, Eds. London: Taylor and Francis, 1996, pp. 189–194.

[21] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," *J. Usability Studies*, vol. 4, no. 3, pp. 114–123, May 2009.