Music Encoding Conference Proceedings

2015, 2016 and 2017

18–21 May 2015
Biblioteca Umanistica, Università degli Studi
Firenze, Italy

17–20 May 2016
Tanna Schulich Hall, Schulich School of Music, McGill University
Montréal, Québec

16–19 May 2017
Centre d'études supérieures de la Renaissance (CESR)
Tours, France

Edited by
Giuliano Di Bacco, Johannes Kepper and Perry D. Roland

DOI: https://doi.org/10.15463/music-1

MEC 2015 • Firenze

Organizing Committee

Giuliano Di Bacco, Indiana University Bloomington, USA (Co-Chair)
Federica Riva, IAML-Italia / Conservatorio di musica "L. Cherubini" di Firenze (Co-Chair)
Mila De Santis, Università di Firenze
Paola Gibbin, Biblioteca Nazionale di Firenze
Johannes Kepper, Musikwissenschaftliches Seminar Detmold/Paderborn, Germany
Perry Roland, University of Virginia, USA

Program Committee

Christine Siegert, Universität der Künste Berlin, Germany (Chair)
Stephen Downie, University of Illinois at Urbana-Champaign, USA
Richard Freedman, Haverford College, USA
Axel Teich Geertinger, The Royal Library, Copenhagen, Denmark
Franz Kelnreiter, Internationale Stiftung Mozarteum Salzburg, Austria

Keynote Speakers

Elena Pierazzo, Université Stendhal - Grenoble 3, France: *Encoding the Code: The Tale of a Textual Impostor*
Eleanor Selfridge-Field, CCARH, Stanford University, USA: *The Evolving Uses of Encoded Music*

MEC 2016 • Montréal

Organizing Committee

Ichiro Fujinaga, Schulich School of Music, McGill University (Co-Chair)
Andrew Hankinson, Schulich School of Music, McGill University (Co-Chair)
Ryan Bannon, Schulich School of Music, McGill University
Karen Desmond, Schulich School of Music, McGill University
Emily Hopkins, Schulich School of Music, McGill University
Audrey Laplante, Université de Montréal
Laura Risk, Schulich School of Music, McGill University

Program Committee

Franz Kelnreiter, Internationale Stiftung Mozarteum Salzburg, Austria / Digitale Mozart-Edition (Chair)
Richard Chesser, British Library. London, UK
Eleanor Selfridge-Field, CCARH, Stanford University, USA
Peter Stadler, Carl-Maria-von-Weber-Gesamtausgabe, Germany
Raffaele Viglianti, University of Maryland, USA

Keynote Speakers

Julia Flanders, Northeastern University, USA: *The Provocation of Music: Evolving Paradigms for Markup*
Richard Freedman, Haverford College, USA: *Music, MEI and the Arts of Quotation*

MEC 2015 • Tours

Organizing Committee

Philippe Vendrix, Centre d'études supérieures de la Renaissance (CESR), Université de Tours (Co-Chair)
Philippe Rigaux, Conservatoire national des arts et métiers (CNAM), Paris (Co-Chair)
Vincent Besson, CESR, Université de Tours
Hyacinthe Belliot, CESR, Université de Tours
Alice Loffredo-Nué, CESR, Université de Tours
Jean-Louis Bouteiller, CESR, Université de Tours

Program Committee

Richard Chesser, British Library, London, UK (Chair)
Laurent Pugin, RISM Switzerland, Bern
John Rink, University of Cambridge, UK
Karen Desmond, Brandeis University, USA
David Rizo Valero, University of Alicante, Spain

Keynote Speakers

Elaine Chew, Queen Mary University, London: *Time and the Limits of Music Notation*
Joachim Veit, Carl-Maria-von-Weber-Gesamtausgabe: *Music Encoding as Initiative and Inspiration: A Vision Through the Eyes of a Traditional Editor*

Contents

Contributions from MEC 2017

Introduction

Giuliano Di Bacco
Indiana University
gdibacco@indiana.edu

MEI is a fine example of a project which is both “a noun and a verb” [1] for it both pursues a definite aim, and projects trajectories of advancement in multiple directions. The declared goal of MEI is the creation of a universal and independent method for the encoding and annotation of music documents. However, at the heart of the initiative is an assorted community of technologists, historians, librarians and other scholars keen to develop shared approaches and tools to aid their manifold projects and questions. It is due to the range of experiences carried out by this community that the core project—the quest for the ideal music markup—gains validation and strength.

The annual conference organized by MEI is a key moment in this process. Since its inception, this conference has been a place where people discuss a number of topical issues, first of all the very concept of the encoding of music as a tool for research. The Music Encoding Conference (let it be remembered once more) by the virtue of its name and in its call for papers, invites scholars to share their experiences in the use of electronically encoded music, broadly conceived and irrespective of the format, and to explore the full range of research questions that could be addressed (and possibly answered) by applying any system for the digital representation of music documents. Then, indeed, the focal point of discussion is how MEI could be applied to these questions, and how the MEI model can be further developed, in order to be more effective and for a wider range of approaches.

It may come as a surprise that there is a need to reaffirm the usefulness of having music documents available in a suitable electronic format. Everyone would want to perform searches into large corpora of music documents, or to process them with computational analysis. However, it is a fact that computational methods are procedures not as commonly used by music scholars, as they are by linguists, or literary historians. The gap is not a theoretical or a technical one, but is due mostly to the shortage of available datasets. The reason lies first of all in the complexity of musical data, which raises the bar much higher for the encoder, and for the way musical documents are most often handled in our favorite environment, the World Wide Web. For decades we have enthusiastically welcomed the exponential increase of original sources, transcriptions, editions available on the internet. Most of the time, however, what we access is an image, either a high-resolution facsimile of a rare original source, or a PDF of a beautifully engraved edition, which is barely useful for searching or analysis. As Laurent Pugin rightly reminds us, our discipline needs to raise awareness of this issue [2]. So it is really useful to have regular appointments such as MEC where scholars can make manifest the range of research questions they can tackle when provided with a suitable electronic representation of their object of study.

Thus in the six MEC conferences held so far, delegates have reviewed several theoretical positions, discovered and tracked the progress of a number of research projects, and accompanied the development of four major releases of the MEI specifications. The present volume follows the 2015 publication [3] that documented the first two conferences in Mainz and Charlottesville. In 2015, MEC returned to Europe. Emblematically, the inaugural 2013 edition was held in the home city of Johannes Gutenberg, while the third edition took place in Florence, in the building that once belonged to the Santa Maria degli Angeli monastery, whose scriptorium produced the Squarcialupi codex and other musical monuments. In 2016 and 2017 MEC was hosted by two music research institutions, who are among the most active in the creation of innovative resources for music research: the Schulich School of Music in Montréal and the CESR in Tours. These three beautiful cities and welcoming scholarly communities have been the perfect setting for such exchanges. This volume gathers contributions by individuals and research groups who presented their work at these three editions of the MEC, in 2015-17. On behalf of the co-editors, I wish to express my appreciation to them for accepting our invitation to contribute a range of examples of how researchers use music encoding and MEI in their endeavors.

As for the topics of broader interest, this volume reminds us how, historically, the development of digital-based analytical methods and encoding systems has been strictly linked to the progress of specific technologies (Schuler, page 131). It emphasizes how the choice of encoding format in computational analysis matters, since it may influence the analytical process itself (Krämer, page 53), and it also stresses the crucially important decisions the encoder faces when they attach metadata to their objects of study, which may sometimes defy classification (Swanson & Lacoste, page 73). Considering the growing body of web resources that give access to digitized music corpora, a method is presented (Viglianti, page 47) that allows to specify and point to a specific portion of an encoded piece of music via an URL address, with endless possible applications. Another method permits to capture the results of individual queries made by users on a website, preserving them as part of the dataset as valuable annotations on existing patterns and relationships (Lewis & al., page 113). We also read about a crowdsourced project launch (*Keyboard Philarmonic*), which aims to create a repository of freely shareable, revisable digital music scores (DeLaHunt, page 107); and of a proposed XML markup format (the Music Performance Markup), designed to encode the nuances of a musical performance through a number of expressive parameters (Berndt & Bohl, page 91). For those, like me, who are fond of the engraving qualities of LilyPond, but lament the lack of support for prevalent file formats, you will be pleased to know that an import and export infrastructure is under construction (Voigt, page 135).

Concerning MEI-centered topics, in the following pages we read progress reports from the *Digital Mozart Edition* (Dubowy, page 7) and from the *Freischütz Digital* (Kepper, page 95), two examples of pioneering digital editions, which originated as an extension of a monumental printed edition, and as a born-digital large-scale application of MEI, respectively. The *Freischütz Digital*, along with the *Beethovens Werkstatt* project provides case studies regarding how digital means can enrich and go beyond a traditional edition. In the former case, MEI is used to capture and investigate performance-related annotations in a score, such as the one used for the first performance of Weber's opera (Seipelt, page 79). In the latter case, MEI is proposed as a tool for genetic text criticism, where the edition's ultimate goal (in this case, Beethoven's sketchbooks) is not to establish the "final" text, but to present an analysis of the writing (that is, compositional) process (Sänger & Kepper, page 37). A different outcome, focusing on stylistic and historiographical questions, is a corpus analysis powered by MEI (Harrison & Khalid, page 143) that investigates a sample of compositions by a group of early 20th-century authors, the "Apaches" (including Ravel, Schmitt, Sévérac and others), to evaluate the compositional patterns characterizing their *debussysme*.

One of the challenges faced by music encoding is the empowering of research on early music and non-Western repertories. Among others, medievalists are working on defining bespoke extensions of MEI, repurposing data collected over time in other formats for furthering research on notational features, manuscript sources, and for editorial purposes (Stinson & Stoessel, page 15; Di Bacco & Roland, page 25). Work is underway on defining a standard method to encode tablatures in MEI, and also confronting cases of compositions which utilize more than one style of notation (Shaw, page 63). Fascinating prospects emerge from the possibility of using MEI to represent both the visual and semantic details of the Hampartsum notation from Ottoman music manuscripts (Plaksin & Olley, page 119).

Most of the contributors to this volume discuss the difficulties they encountered in their projects and the strategies used to overcome them, and this should be helpful to those who are in the planning stage of projects with similar goals. One of the basic problems that anyone faces when embarking on an encoding campaign—and we find this discussed throughout the volume—is the coding itself, the acquisition of musical data and its transfer to an electronic file. Among the editors created to aid in writing and editing MEI data, a browser-based editor is presented (Horwitz & al., page 45) which synchronizes the code editing with graphical editors and score engravers. Other projects are making progress on Optical Music Recognition, as exemplified by TAMIR, which is a toolbox tailored for work on 16th century manuscripts (Chen et al., page 151).

To conclude, when a music encoding conference was suggested less than a decade ago, people doubted there could be enough material and interest from music technologists and would-be digital musicologists to make it a regular necessity. On the contrary, in the last few years we have seen a huge growth of interest in the encoding and digital annotation of music, which is also evident from analyzing other conference programs in our discipline. In addition, there is a clear demand to overcome the 'silo model' of the digital humanities also in the realm of music studies, to build common platforms for an increased interoperability of resources. I think it would be over-optimistic to imagine a future where all the existing formats and systems could merge into one. After all, each model was created to fulfil a specific goal; some encoding systems have become obsolete, superseded by more efficient ones, while others are still more suitable than others for certain purposes. MEI, with its community-based model is helping the process, creating a very powerful and flexible model for representation and interchange, which is also more human-readable than others, and now legitimately aiming to comprehensiveness. The growing number of people who found a common ground in MEI, a reason for adding an XML schema (yet another one!) to their research toolset, justifies the existence of an annual forum for discussion. This volume is a non-exhaustive but quite illustrative collection of examples representing its outcomes.

Works cited

[1] Burdick, Anne, Johanna Drucker, Peter Lunenfeld, Todd Presner, and Jeffrey Schnapp. *Digital Humanities*. Cambridge: MIT Press, 2012: 124 (mitpress.mit.edu/books/digitalhumanities or archive.org/details/DigitalHumanities_201701).

[2] Pugin, Laurent. "The Challenge of Data in Digital Musicology" *Frontiers in Digital Humanities* 2 (2015) (frontiersin.org/article/10.3389/fdigh.2015.00004).

[3] *Music Encoding Conference. Proceedings 2013 and 2014*, ed. by Perry D. Roland and Johannes Kepper. Mainz-Munich: Music Encoding Initiative - Bayerische Staatsbibliotek, 2015). (Online resource: langzeitarchivierung.bib-bvb.de/delivery/DeliveryManagerServlet?dps_pid=IE2860815& also reachable at music-encoding.org/archive).

Contributions from MEC 2015

Encoding the NMA in the Digital Mozart Edition: A Progress Report

Norbert Dubowy
Internationale Stiftung Mozarteum Salzburg
dubowy@mozarteum.at

Abstract

The *Digital Mozart Edition* (DME) has as its goal the digital presentation of documents and texts related to Mozart's life and oeuvre. The most challenging project within the DME is the digital edition of Mozart's music which is currently being developed. It will eventually make the complete compositional output of Mozart available to the public in critical, MEI based editions. The DME already hosts the *NMA Online*, the representation in digital images of the *Neue Mozart Ausgabe* (NMA) print edition, originally published between 1954 and 2007. While on the one hand, the DME seeks to transform the authoritative musical text of the NMA into digital format, the DME also intends to broaden the philological concept by the publication of new and alternative editions based on individual sources. At the present state of the project, several challenges have been identified, some connected to the publication history and the availability of sources used by the NMA. Others are related to the representation of the *Kritische Berichte* of the NMA within the DME. The paper reports on the development of the conceptual as well as practical model of the DME music edition.

Introduction

The purpose of this paper is to outline the project of a digital music edition within the framework of the *Digital Mozart Edition* (DME) which is jointly pursued by the Internationale Stiftung Mozarteum in Salzburg, and the Packard Humanities Institute in Los Altos/Ca. While the *Neue Mozart Ausgabe* (NMA), the complete critical edition of Mozart's music, may be known to many, a few words on the DME may not be out of place in this context.[1]

The *Digital Mozart Edition*, initiated in 2001, has developed into a group of several digital projects. Its goal is the digital presentation of documents and texts related to Mozart's life and oeuvre. The edition of the correspondence of the Mozart family and the critical edition of the librettos to Mozart's operas and oratorios in digital format are two segments of the *Digital Mozart Edition* that are already accessible online. The digital edition of Mozart's music will eventually make the complete compositional output of Mozart available to the public in critical, MEI based editions. It would not only complement the existing projects, it has to be seen as the core element of the family of DME projects. Hence, its acronym, DME, stands for three things, the edition of the music, the editorial project, and the whole group of digital Mozart projects that share the same critical approach towards textual scholarship and the goal of accessibility. Hopefully, at some point in the future, they will be fully integrated with each other. In order to distinguish the DME, which refers to the family of digital

1. This paper complements the perspectives on the *Digital Mozart Edition* given by Kelnreiter [3].

Mozart projects, from the DME in a stricter sense, which stands for the music edition only, we use the acronym DMEDME, the 'Digital Mozart Edition as Digital Music Edition', in our internal discussions. In the context of this paper, we will simply refer to the DME as the music edition.

Principles of the DME Music Edition

Since 2006, the Mozarteum Foundation has already hosted the *NMA Online*, the representation in digital images of the *Neue Mozart Ausgabe* print edition, enhanced with additional capabilities such as a synoptic view of the music edition and the corresponding critical report. As big an achievement as this already is, it cannot, in the long run, suffice in an age that is geared more and more towards a fully digital representation. Eventually replacing the images of the analog music prints currently available in the *NMA Online* with fully encoded music that is rendered on the spot is simply a logical consequence of the development of critical editing in general and the path the Mozarteum Foundation has embarked on. The NMA, however, continues, as we will see, to play a key role in the concept of the DME and remains a cornerstone of the future DME music edition.

There are two premises that determine the structure of the DME:[2]

The first premise regards the relationship between DME and NMA. The NMA is a major achievement in Mozart scholarship; it is widely distributed and is the authoritative edition of Mozart's music. As such it is the reference point for many, performers and scholars alike. It would be unwise to discard it. It is basic to the concept of the DME that the NMA has to be fully preserved within the DME, not only in the digitized format currently available through the *NMA Online*, which relies on scanned pages of the analogue print edition, but also as an entirely digital encoding of its musical text. On the one hand, the NMA text may be treated just as a musical source, as one musical text among others, i.e. it will be encoded in MEI along with the other musical sources, mansucripts and printed editions alike. On the other hand, it seems practical to use the NMA in a special way as a reference, a 'reference text', other, alternative editions of the same work can be pitted against in order to display their differences. In practical terms, the DME will clearly be based on the NMA even in its material aspect as the encoding we use is derived from the NMA print edition. We will take a look at the relationship of NMA and DME and its role within the DME in more detail shortly.

More than a mere transformation of the NMA into digital format, the DME also intends to broaden the philological and editorial approach. Here the second premise comes into play, the concept of 'alternative editions', which may deserve some explanation. In many cases the notion of authorial intention, either the search for a 'Fassung letzter Hand' or the principal of 'best text', i.e. the singling out of a source that is deemd to represent the author's intentions, have been and are still considered viable methods for critical editions. They are also among the editorial principals of the old NMA as the guidelines from 1962 state: The composer's autograph is the preferred source; in the case of several autograph sources, the "later and more mature state" is to be preferred.[3] In cases when there are several versions of a work that reflect, for example, different productions—the NMA called these "Eventualfassungen"—one version has to be picked.[4] It is well known that these principles have undergone criticism in recent years concomitant with the critique of the author and work concept. Alternative concepts emphasize the individual source and recognize the possibility of multiple

2. For an introduction on the principles of the DME see Leisinger [1].

3. "Sind von einem Werk zwei oder mehrere Autographe vorhanden, so ist die Frage zu prüfen, ob es sich dabei um 'sekundäre' Autographe, d. h. Autograph-Kopien handelt, oder ob diese Autographe verschiedene Entwicklungsstufen des betreffenden Werkes darstellen. In letzterem Fall ist dem späteren bzw. reiferem Stadium primäre Stellung in der Bewertung einzuräumen." [2, p.259].

4. Ibid.

versions, each of them potentially tied to different historical or institutional circumstances. While obvious limitations exist with the print media, the digital format makes it easier to show multiple versions alongside one another.

The DME intends to follow this idea for a selected number of works by offering, in addition to the NMA-based edition, the so-called ‘alternative editions’. In most cases, these alternative editions may be source editions based on one single source, the autograph, an early print, or even a later copy if it is historically relevant, e.g. for the reception of the work. In other cases, several sources may be combined if they relate to a specific version of the composition, etc. Thus, the DME may potentially offer more than one alternative edition of a specific work.

As a consequence of the emphasis on the individual source on the one hand, and the goal of transparency of the editorial process on the other hand, the procedural steps from source to edition should be laid out in front of the reader. For this purpose, the DME intends to offer various ‘views’ or stages of textual representation for display: The first stage is a transcript of the original source ‘as is’, i.e. with all its idiosyncrasies, ambiguities, and errors; the second stage is an interpreted text version of the previously encoded transcript version. The interpreted text provides a continuous musical text in regularized musical orthography, e.g. by resolving, mostly quite mechanically, abbreviated notation among other details. The third stage, finally, is the edited text; it is here where the editorial decisions have to be made. A similar trajectory from source to edition has already been put in place in the DME libretto edition that offers a transcript along with the edition of each individual textual source. In the music edition, all three stages are encoded and potentially rendered in regular notation. To make it clear, no source images are planned to be included; we do not intend to make an `Edirom`-like edition. The ‘edited text’ is the edition in the stricter sense which contains all of the features known from a traditional critical edition, i.e. editorial emendations, variant readings, critical commentaries. In addition, in order to take advantage of the potentialities of a digital edition, we plan to provide certain display options that can be activated by the user such as switching between modern and original scoring, original clefs, or variants of orchestration. The components of the DME and their interaction is tentatively summarized in figure 1:

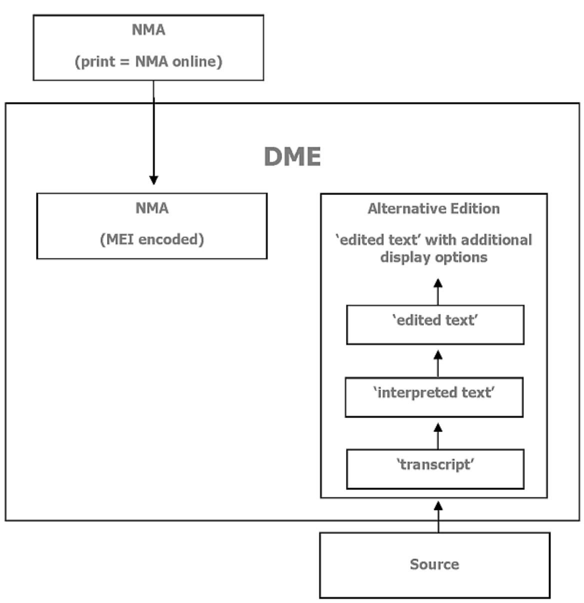


Figure 1: Schematic outline of the DME and its components.

The DME is both, a continuation of the NMA and its revision. It may involve, in some cases, new philological expertise and may even lead to editorial decisions different from those taken by the editors of the NMA. On the other hand, the NMA is also the reference point for the alternative editions. Ideally, we envision that both editions, the digitally encoded NMA and the alternative editions, will be automatically linked to each other in a way that potential differences can be easily documented and visualized; the technical implementation of this function, however, needs still to be developed. The preparation of the alternative editions in particular is a time-consuming and labor-intensive process. To make it clear, 99% (but any number given here is purely speculative) of the DME will consists of an enhanced and transformed NMA while the alternative editions are an option that is, at the moment, reserved for a few cases. The team of the DME is working on a number of compositions, K. 458 and K. 365 among others, in order to gain experience and test the workflow.

DME and NMA

The DME is materially derived from the NMA, since its encoding is based on the encoding of the NMA. The complete NMA print edition is already encoded in a proprietary format (DOX format) developed by David Packard and the *Packard Humanities Institute*. It is a format, elegant in its simplicity and powerful at the same time, which serves very well the purpose it was created for, i.e. the encoding of the printed NMA text including features of layout of the original NMA page, which can be rendered in such a way as to closely reproduce the printed NMA.[5]

As a proprietary format derived from and tailored to one single—albeit quite substantial—corpus and its printed representation, it is less suited for general encoding purposes, and does not provide any markup that would allow bringing any deep structure to the forefront or would enable the referencing to contents not found on the page the encoding is derived from.

Nevertheless having the musical substance and an authoritative text already available in an encoded form is an invaluable advantage. But in order to overcome the shortcomings, the best solution at the moment is to 'translate', transform, or convert the existing Packard encoding, the so-called DOX format, to MEI. We are working, with the help of external experts, on the tools to transfer files from DOX to MEI format, but it will still take a while before we can talk about a satisfactory result. Once this goal is achieved—hopefully with the least amount of effort needed for reworking and retouching—it will be possible to convert the complete NMA and make it available to the public within the framework of the *Digital Mozart Edition*. At the moment, however, all our sample codings are generated on the slow track, i.e. by creating a clone of the NMA text in `Sibelius` which is subsequently exported to MEI via the `sibmei` plug-in or through `MusicXML`.

Encoding the musical text of the NMA is not a problem in and of itself. It may even be fairly simple as the NMA relies on a standardized format and notational style, the style and layout used by the publishing house Bärenreiter, and does not need to deal with the uncertainties and inconsistencies often found in original sources. A few problems remain that have to do with the decisions the editorial board of the NMA made decades ago, namely the diacritic marking of editorial additions, especially the small sized staccato dots, spiccato strokes and accidentals, as well as dynamics and trills in non-italic fonts.

Another problem one may encounter is the presence of different versions within the NMA itself. As the NMA was published over a period of more than 50 years (officially 1954 to 2007), some of the volumes have received one or even more re-editions with addenda and corrigenda. These additions and corrections pertain not only to obvious printing errors but also to more substantial musical variants: Many works were edited without the benefit of having certain autograph scores by Mozart at hand; for example, many sources formerly housed in Berlin were dispersed during World War II and were thought to be lost at the time of the publication of the earlier volumes of the NMA. In reality, these sources were safely preserved and "re-discovered" at the Biblioteka Jagiellońska in Krakow, Poland, where they are now available again.

Of course, one could argue that the newest corrected NMA edition should be used, but it is necessary to keep in mind, that the DOX encoding utilized the first edition of the NMA. This means that additional corrections, either on the DOX or the MEI level, need to be made.

5. The format is briefly described in Kelnreiter [3]; at the time, the name of the format was `Fox`.

The DME and the *Kritische Bericht* of the NMA

A major issue involves the Critical Reports (*Kritische Berichte*) that usually accompany the printed edition of the musical text. In the case of the NMA, many Critical Reports were written years after the publication of the music edition, many of them after the re-discovery of the aforementioned autograph sources. This creates a problem as music edition and Critical Report may not be congruent. Things get further complicated when volumes are re-issued in a second or third edition after the publication of the Critical Report.

The question of whether the digital re-edition of the NMA within the DME should include also a digital re-edition of the *Kritische Bericht* volumes is open to debate. If so, how can it be integrated into the DME? The simplest solution for combining the music edition with the Critical Report would be a synoptic view of the encoded (and subsequently rendered) NMA with a similarly encoded Critical Report. The synoptic and coordinated display of the music edition and the Critical Report is one of the features of the current *NMA Online*: [6]

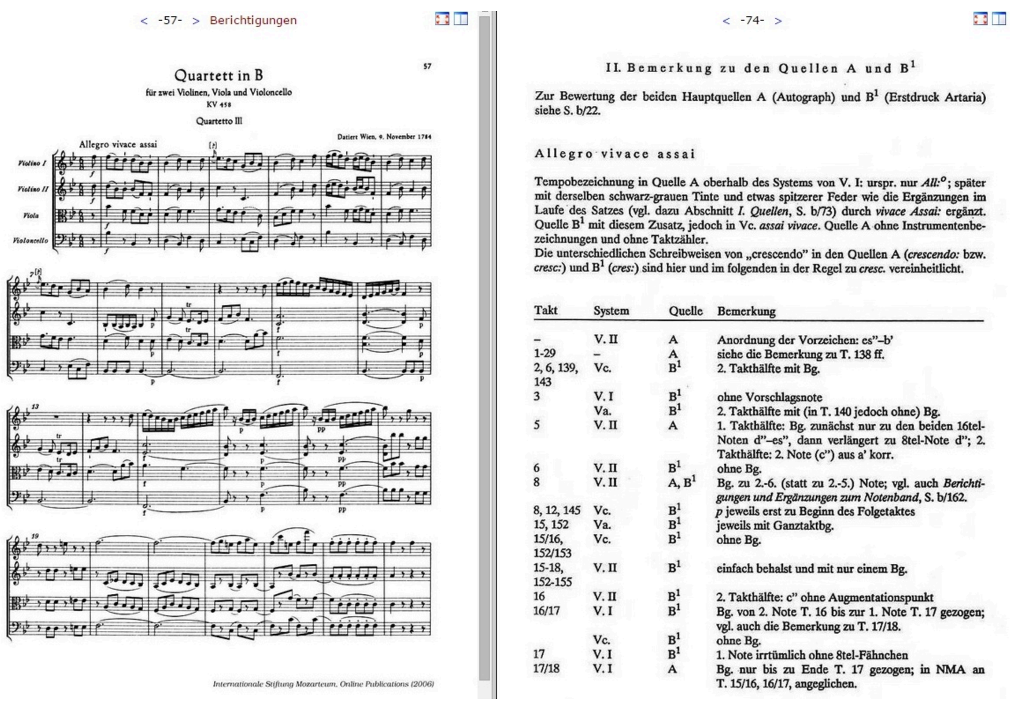


Figure 2: Synoptic view of an edition and its corresponding Critical Report in the *NMA Online* (screenshot).

This display solution, however, where the reader himself has to look up and correlate measures and potential readings, is outdated. Reading habits and user expectations have changed. Splitting the list view of the traditional 'Lesartenverzeichnis' (variant readings) into single annotations, each of them in its own pop-up window like in `Edirom` seems more appropriate as it allows us to link the content of the annotation directly with the notational item in the score.

6. *NMA Online*, ed. by the Internationale Stiftung Mozarteum (http://dme.mozarteum.at/DME/nma/start.php?l=2). The present example is taken from NMA, vol. VIII/20/2 [4] and the Critical Report to vol. VIII/20/2 [5].

All this, however, cannot hide the fact that it is the structure of the Critical Report of the NMA and its content that prevents any easy transfer into a digital medium. This is best illustrated by the readings to the string quartet K. 458, which deals with two sources only, the autograph score and the first print, here labeled A and B^1. Here is a partial transcript—in TEI—of the annotations to movement one; the individual entries (originally unnumbered) have been numbered for convenience: [7]

	Takt	System	Quelle	Bemerkung
1	–	V. II	A	Anordnung der Vorzeichen: es"–b'
2	1-29	–	A	siehe die Bemerkung zu T. 138 ff.
3	2, 6, 139, 142	Vc.	B^1	2. Takthälfte mit Bg.
4	3	V. I	B^1	ohne Vorschlagsnote
5		Va.	B^1	2. Takthälfte mit (in T. 140 jedoch ohne) Bg.
6	5	V. II	A	1. Takthälfte: Bg. zunächst nur zu den beiden 16tel-Noten d"–es", dann verlängert zu 8tel-Note d"; 2. Takthälfte: 2. Note (c") aus a' korr.
7	6	V. II	B^1	ohne Bg.
8	8	V. II	A, B^1	Bg. zu 2.-6. (statt zu 2.-5.) Note; vgl. auch *Berichtigungen und Ergänzungen zum Notenband*, S. b/162.
9	8, 12, 145	Vc.	B^1	*p* jeweils erst zu Beginn des Folgetaktes
10	15, 152	Va.	B^1	jeweils mit Ganztaktbg.
11	15/16, 152/153	Vc.	B^1	ohne Bg.
12	15-18, 152-155	V. II	B^1	einfach behalst und mit nur einem Bg.
13	16	V. II	B^1	2. Takthälfte: c" ohne Augmentationspunkt
14	16/17	V. I	B^1	Bg. von 2. Note T. 16 bis zur 1. Note T. 17 gezogen; vgl. auch die Bemerkung zu T. 17/18.

Figure 3: Rendered transcript of p. 74 of the Critical Report [5] to K. 458.

Annotation 3 reports the presence of a slur in measure 2, 6, 139 and 143 in source B^1. This is a collective annotation (as are annotations 9-12, and 14); it works fine as long as we accept that the 'Lesartenverzeichnis' is a simple list of occurrences or phenomena the editor deemed to be noteworthy. But already in the synoptic view as it is presented in *NMA Online*, it no longer works. In our days, readers expect to have the information available at the point of relevance. The moment we look at measure 139 of the score, the 'Lesartenverzeichnis' has moved four pages further on and the information given at measure 2 (but also pertaining to measure 139) is no longer visible on the screen. In order to avoid the inconvenience of having to jump back to the first page, the annotation needs to be split and repeated for measures 6, 139 and 143. But once this is done, the 'Lesartenverzeichnis' is no longer the same, and it opens the door for even more change such as the elimination of annotations whose essence can be easily reported by other, e.g. graphic means. This would result in an amended, purged, and totally overhauled Critical Report. It is questionable, however, whether it is wise to invest too much time and workforce into the preparation of a digitally refurbished Critical Report as long as the original analogue (though digitized) Critical Report continues to be accessible through the *NMA Online*.[8]

7. Based on the Critical Report to vol. VIII/20/2 [5, p.74].
8. In a resolution taken after the presentation of this paper, the Advisory Committee of the DME recommended not to pursue the digital transformation and integration of the Critical Reports of the NMA into the digital version of the NMA.

Outlook

The whole concept of the DME, with its multiple stages and views of sources and editions, is still a work in progress, which is highly dynamic and in some parts transformational. It goes without saying, that the concept is dependent upon an appropriate visualization, i.e. the rendering of the MEI data in all its forms, including editorial oddities such as the small staccato dot used in the NMA. Leaving the DME at the level of the XML code is not an option; the edition needs to be rendered in traditional music notation readable by anyone. The DME has not taken any steps to develop a rendering tool of its own, but will have to rely on third party input in this regard. The prospect of having thousands of pages of Mozart's music available in MEI may serve as an incentive. The future DME music edition will provide a service to the larger Mozart community and to anyone who wants to play, study, or simply enjoy Mozart's music.

Works cited

[1] Leisinger, Ulrich. "Methoden und Ziele der Digitalen Mozart-Edition" *Mozart-Jahrbuch 2013*: 43-63.

[2] Appel, Bernhard and Joachim Veit (eds.) *Editionsrichtlinien Musik*. Kassel: Bärenreiter, 2000. 249-279.

[3] Kelnreiter, Franz. "Explorations on a Digital Mozart-Edition: How to Make Mozart Compatible with MEI". Presented at the Music Encoding Conference 2014, Charlottesville, VA (unpublished).

[4] Finscher, Ludwig, ed. *Wolfgang Amadeus Mozart. Neue Ausgabe sämtlicher Werke*. Serie VIII, Werkgruppe 20, Abteilung 1: Streichquartette, Band 2. Kassel: Bärenreiter, 1962.

[5] Finscher, Ludwig and Wolf-Dieter Seiffert. *Wolfgang Amadeus Mozart. Neue Ausgabe sämtlicher Werke, Kritische Berichte*. Serie VIII, Werkgruppe 20, Abteilung 1: Streichquartette, Band 2. Kassel: Bärenreiter, 1993.

Revising MEI for Research on late Medieval Manuscripts

John A. Stinson
La Trobe University
johnastinson@gmail.com

Jason Stoessel
University of New England, Australia
jason.stoessel@une.edu.au

Abstract

This paper illustrates opportunities for revising the current `Chant` and `Mensural` modules of the Music Encoding Initiative for encoding music notation from before c.1500. Repurposing data collected over the last three decades in Scribe music encoding software required a bespoke MEI module, which we have called `NeoScribe`. The Scribe project has benefited from the long-term investigation and implementation of methods for the efficient encoding of late medieval music notation. With Scribe, users are able to represent every meaningful scribal mark on the written page, something that is not currently possible in the current MEI Chant and Mensural modules. In converting Scribe data to a MEI-compliant XML, we recognised the need to retain Scribe's nuance-rich encoding of medieval musical notation. For zero dataloss, new elements and data types were added to MEI for encoding late medieval chant and mensural notation. We demonstrate some of NeoScribe's enhanced features for encoding 14th-century repertoires. We conclude by discussing some of the benefits of revising the MEI Chant and Mensural modules for projects investigating music from before c.1500.

Introduction

In the mid 1980s, John Stinson set out to create a computerised system for encoding every meaningful mark in manuscripts of chant and mensural music from the twelfth to early fifteenth centuries. Over the next three decades, he and his assistants encoded the complete texts and musical notation for 6,331 works, including but not limited to the complete chants of Hildegard of Bingen, the complete polyphonic songs of Don Paolo Tenorista da Firenze and Francesco Landini, and the Dominican order's complete annual cycle of chant. In doing so, two central principles for encoding medieval music obtained: where the writing process was identical across chant and mensural repertoires, it was encoded in the same way; secondly, all music encodings organised neumes or notes according to corresponding syllables in this largely vocal repertoire. Encoding both chant and mensural notation consistently recognised the strong intertextuality of medieval music in which music from one repertoire could be readily transferred to another, as in Gregorian chant melodies being used in polyphonic motet tenors. In 2013, we embarked upon a collaborative project to convert the entirety of Scribe-encoded data into Music Encoding Initiative XML data in the expectation that eventually making this data available in this publicly documented format would provide music historians with a formidable resource for future research. In translating from one data format to another, thirty years of lessons learnt from building and using Scribe, our own and others' expertise in musical paleography, and the lessons of the MEI endeavour, necessitated the modification and extension of the elements and data types in MEI. The proposals that follow have emerged from this process, and we offer them here in the hope that they might fuel the ongoing development of MEI.

Overarching Issues

We begin with a highly abbreviated overview of the rapid notational developments in the thirteenth to fifteenth centuries. Late twelfth-century scribes started to use a wide-nibbed writing implement that, when moved at right angles to its edge, produced a square or rectangle. The pen strokes of older notational signs, known as neumes, were adapted so that, for example, the original fluid curve of the tick-like `pes` or `podatus` mutated into three conjoined strokes consisting of two vertically aligned rectangles joined on their right side by a ligature. The `virga`, originally a neume to indicate a relatively higher pitch, became a rectangle with a descending stem added to its right side. The `punctus`, originally a neume to indicate a relatively lower pitch, became either a rectangle or, when used in a so-called interrupted neume like a `climacus`, a rhomboid note. The new mendicant orders founded by Saints Dominic and Francis in the early thirteenth century rapidly adopted and standardised square chant notation, and it remained in use virtually unchanged for the next five centuries. [5]

These processes of writing square notation were central to the synchronic development of mensural notation in the second half of the thirteenth century. Although many of the written forms common to square chant notation and mensural notation had different musical meanings, their writing processes remained essentially the same. Unlike square chant notation, which had no rhythmic significance, the variation of standard neume forms in the ligatures of mensural notation signified different rhythms. The transformation of the `virga`, `punctum` and `inclinatum` into the mensurally significant long, breve and semibreve paved the way for an array of rhythmically significant notes used during the fourteenth century.

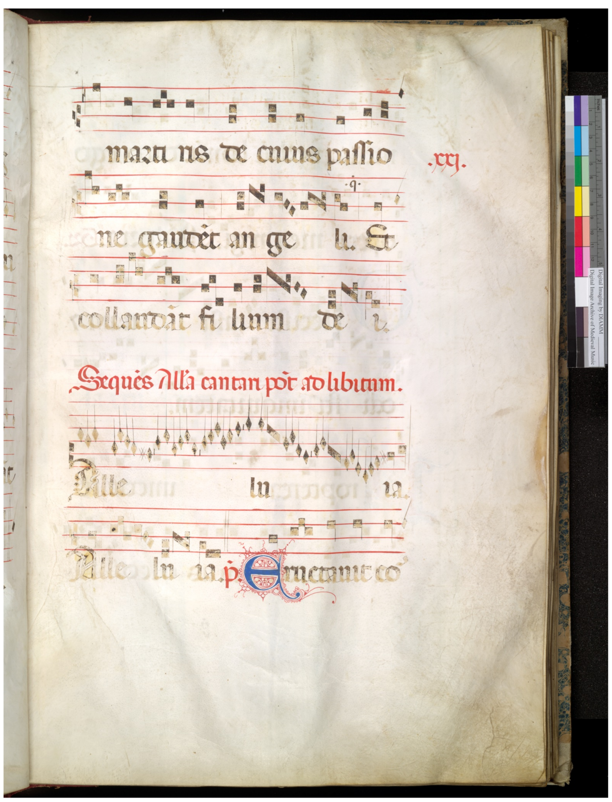


Figure 1: Florence, Biblioteca Medicea Laurenziana, Ms Ashburnham 999, fol. 21r. Reproduced by the kind permission of the Biblioteca Medicea Laurenziana. Further reproduction prohibited.

The proliferation of notational forms in chant and mensural notation presents a serious challenge to MEI, which relies upon verbal tags within XML elements to denote neumes and notes. Even though there is some consensus in the naming of simple chant neumes—but note the previous case of the `pes` and `podatus`—the naming of more complex neumes, for example the `pessubtripunctis`, is a more thorny issue. Stefan Morent [6] and Gregor Schräder [7] developed an elegant solution to the problem of compound neumes, but here we would like to propose further refinements. The nominalist (not to mention linguistic) conundrum at the heart of MEI made its presence felt even more when we sought to convert encodings of fourteenth-century mensural notation. Pointedly, several notes are found in medieval manuscripts that have no codified name. Simply put, MEI in its current state is not completely adequate for dealing with most fourteenth-century Italian Trecento notation, and most of the so-called *ars subtilior* repertoire from ca.1370 to ca.1430.

Gaudeamus omnes and fourteenth-century musical writing

An early fifteenth-century treatment of *Gaudeamus omnes in Domino*, the Introit chant used for the feasts of St Agatha, the Virgin Mary and All Saints, illustrates the need for cross-repertoire-consistent encoding of medieval music. In the manuscript Ashburnham 999 from the Laurenziana Library in Florence, the scribe has carefully laid out the chant in regular square notation (see Figure 1).[1] The neumation of the chant in Ashburnham 999 closely follows the tradition represented in many other neumed sources, including the contemporaneous Douai, Bibliothèque Municipale, Ms 1171, fol. 81r (Douai 1171) and the earlier Ms Gaddi 44, also now in the Laurenziana Library. Both Douai 1171 and Ashburnham 999 were copied in the very location of the 2015 Music Encoding Conference (at which this paper was first presented), namely in the scriptorium of Santa Maria degli Angeli, and under the nose of the composer and monastery rector Don Paolo Tenorista da Firenze [3].

The same scribe who copied the chant, also wrote a discantus in French mensural notation on the opposite folio of Ms Ashburnham 999 (see Figure 2). [2]

Both the mensural and chant parts were sung together polyphonically.[3] Yet the chant and mensural notations in this piece of music use completely differently notational systems, especially for indicating rhythm. The chant neumes lack rhythmic significance. The notes and ligatures of the mensural part denote precise relative durations. Indeed the only way to sing the polyphonic *Gaudeamus* is to sing all notes of the original chant as breves that are the same length as breves in the new mensural part. To emphasise: the chant part of *Gaudeamus* cannot be read according to the rules of ligatures in the mensural notation of the upper part. This piece of polyphony, as such, uses both mensural and chant notation simultaneously.

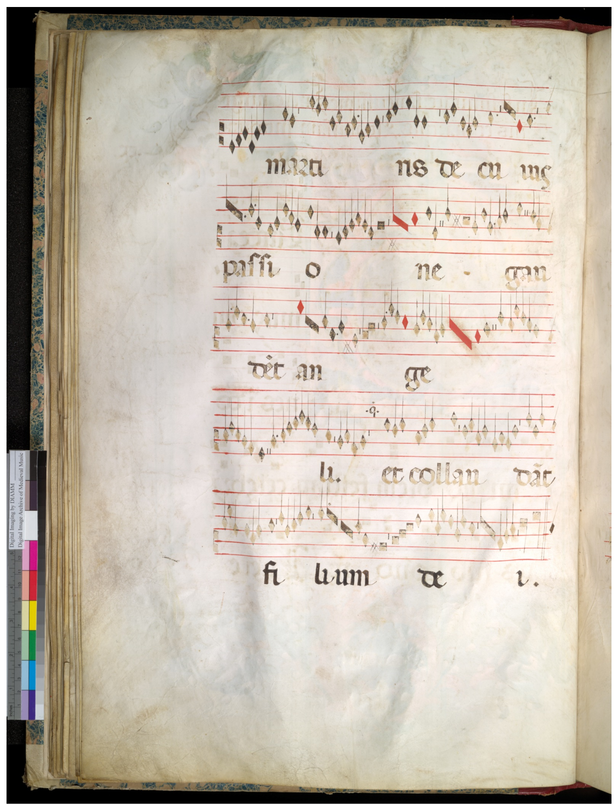



Figure 2: Florence, Biblioteca Medicea Laurenziana, Ms Ashburnham 999, fol. 20v. Reproduced by the kind permission of the Biblioteca Medicea Laurenziana. Further reproduction prohibited.

If MEI is to be useful for encoding pieces like this for the purposes of further research, including many examples from the monophonic *cantus fractus* repertoire from the 14th to 17th centuries,[4] and comparisons made with other encodings, both chant and mensural encodings must exist in the same layer of MEI. Furthermore, the element and data tags for clearly analogous structures need to be commensurate across mensural and chant elements. The clumsy nomenclature for some or all neumes might be dropped but the minimum to be retained should indicate that, for example in *Gaudeamus*, the pitches on

1. Florence, Biblioteca Medicea Laurenziana, Ms Ashburnham 999.
2. The copying of both parts is so organised that when folio 20r is to be turned, on its verso is the mensural notation just where you need it, opposite the continuation of the plainsong on fol. 21r. NB. transcription in [1], pp.375-77.
3. The introit *Gaudeamus* was so well-known that the chant was probably sung from memory.

the last syllable of “filium” are one compound neume comprising a group of two ascending pitches (*podatus*) followed by a group of five, the last two rhomboids really being part of the *porrectus subbipunctus* structure.

“Reforming” the neume branch

In making the leap from Scribe to MEI, a number of principles were adopted in our project that differed from the current MEI Neume branch. In particular it was decided that the `uneume` and `ineume` elements needed to be subsumed into a `neume` element in a manner already familiar to those researchers at McGill University who developed the Solesmes branch [4]. Neumes can be nested, just as in the present MEI model, but the names of neumes were drastically simplified and some new approaches taken for the encoding more complex neumes, both of the so-called uninterrupted and interrupted variety. The following principles were applied:

1. Only the names of simplest neumes were used: `punctum`, `virga`, `pes`, `clivis`, `torculus`, `porrectus`, and `apostropha`;
2. A new data class `data.NEUMESTOKEFORM`, adapted from `data.INEUMEFORM`, defined the appearance of a simple neume or note with the following att.form attributes: `liquescent`, `quilismatic`, `rectangular`, `rhombic`, `tied` (or `ligated`), `oriscus`, and `tractulus`;
3. All neumes containing liquescence, an `oriscus` or a `tractulus` were indicated by their simple from, and the `data.NEUMESTOKEFORM` set as a form attribute for the neume or individual note elements, eg. `pressus (major)`;

```
<syllable>
   <syl>Do-</syl>
   <neume>
      <neume name="virga"/>
      <neume name="punctum" form="oriscus"/>
      <neume name="punctum"/>
   </neume>
</syllable>
```

4. Complex and interrupted neumes like the `climacus`, `scandicus`, `torculus resupinus`, `porrectus flexus`, `pessubpunctis` and the `bi-/tristropha` were encoded as compound, nested simple neumes. For example, the `torculus resupinus` becomes a torculus plus a `virga`, which as nested neumes are implicitly ligated. However, the rhombic attribute implicitly prevents ligation in the `climacus` and `scandicus`;
5. The chant neume dialect, in our case square neumes and Hildegard’s `Hufnagelschrift` neumes, is indicated in the header, to permit the correct visualisation of the encoded neumes.

This model might be profitably adopted for the revised `Neumes` branch of MEI, `<ineumes>` and `<uneumes>` deprecated in favour of simple and nested `<neumes>` whose appearance is encoded as set out above.

4. See [2].

Stems, flags, signs

Converting mensural notation encoded in Scribe to MEI warranted the introduction of several new data as attributes to basic mensural elements that already exist in MEI Mensural. The following discussion presents selected examples from the Trecento and *ars subtilior* repertoire.

Several note forms are endemic to Trecento notation so as not to warrant specific examples from manuscripts. The semibrevis caudata, the dragma, oblique-stemmed notes, the reverse-flagged semiminim, the *signa divisionis*, and one-pitch ligatures require either special adaptation of existing MEI methods, or in most cases new data elements.

	Semibrevis caudata
	Dragma/fusa/fusiel
	Oblique stemmed minim
	Reverse-flagged semiminim
	One-note ligature
`.p., .i., .q., .sg., .si., .o., .n., .d.`	Signa divisionis (Trecento “time signatures”)

Figure 3: Trecento notation forms

In order to achieve these additions

1. a new attribute `@stem.type` needed to be added to `att.stemmed.cmm`
2. new `data.MENSURALSTEMTYPE` defined, which adapts stem names from the Standard Music Font Layout (SMuFL)[5] specification for medieval and Renaissance noteheads and stems.
3. New attribute class `stem.flag` added with attributes `@type` and `@dir` and `data.MENSURALFLAGTYPE` and `data.MENSURALFLAGDIRECTION` were defined
4. The definition of `data.MENSURATIONSIGN` was extended to include all *signa divisionis* letters for the element `<mensur>`. It is also proposed that an attribute `@modifer` be added to `<mensur>` that accounts for the array of modified mensuration signs found during the fifteenth century in sources like Ms Turin J.II.9 (Figure 4).

5. https://www.smufl.org

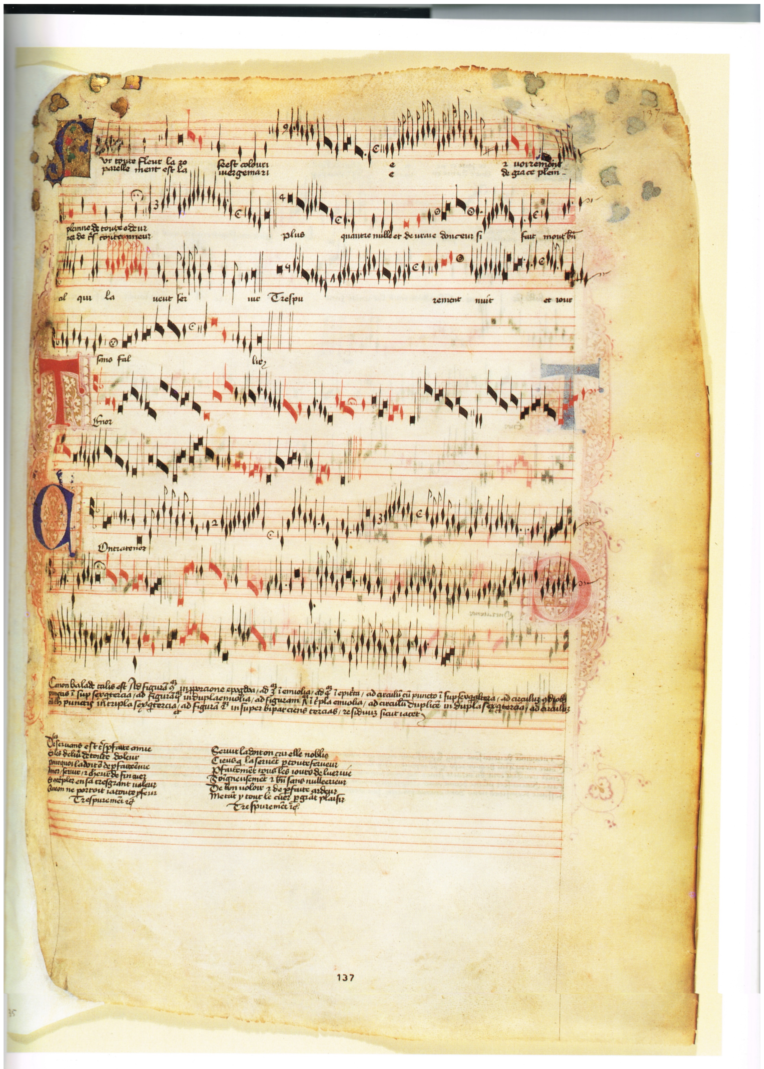



Figure 4: Cantus of *Sur toutes fleur*, Turin, Biblioteca Nazionale Universitaria, Ms J.II.9, fol. 137r. Reproduced by the kind permission of Libreria Musicale Italiana, and permission of the Ministero dei Beni e delle Attività Culturali e del Turismo, Biblioteca Nazionale Universitaria di Torino. Further reproduction prohibited.

Special attention is also directed to current problems in MEI Mensural associated with encoding double signatures, and mensuration signs that only indicated part of the mensuration, both of which Stoessel [8] has documented in his research. At present, we have no solution as to how MEI might encode the first class of mensuration signs, although element `<proport>` provides a model that might be adapted.[6]

In the case of tempus-only mensuration signs, which were used in the late fourteenth century, using the `@prolatio` attribute with `<mensur>` to indicate the implicit prolation avoids any possible misunderstanding.

The addition of new stem types was also deemed suitable for encoding examples of special notes found in the *ars subtilior*, for example that found in the notation of Jacob de Senleches (see Figure 5). Senleches's graphical score, which is not the first of its kind in the fourteenth century, also presents challenges for providing adequate descriptors of its appearance, use of a verbal text to describe how it needs to be performed, and dealing with unusual notational features. Most unusual is the tablature-like notation of Senleches's canonic work in which notes are only written on the lines or, in keeping with the visual representation itself, the strings of the harp.

6. An example of double signatures can be found in the Confiteor of *Missa 'L'Ardant desir'* found on fols 98v-99r of Ms Rome, Biblioteca Apostolica Vaticana, Ms Cappella Sistina 51 (http://digi.vatlib.it/view/MSS_Capp.Sist.51).

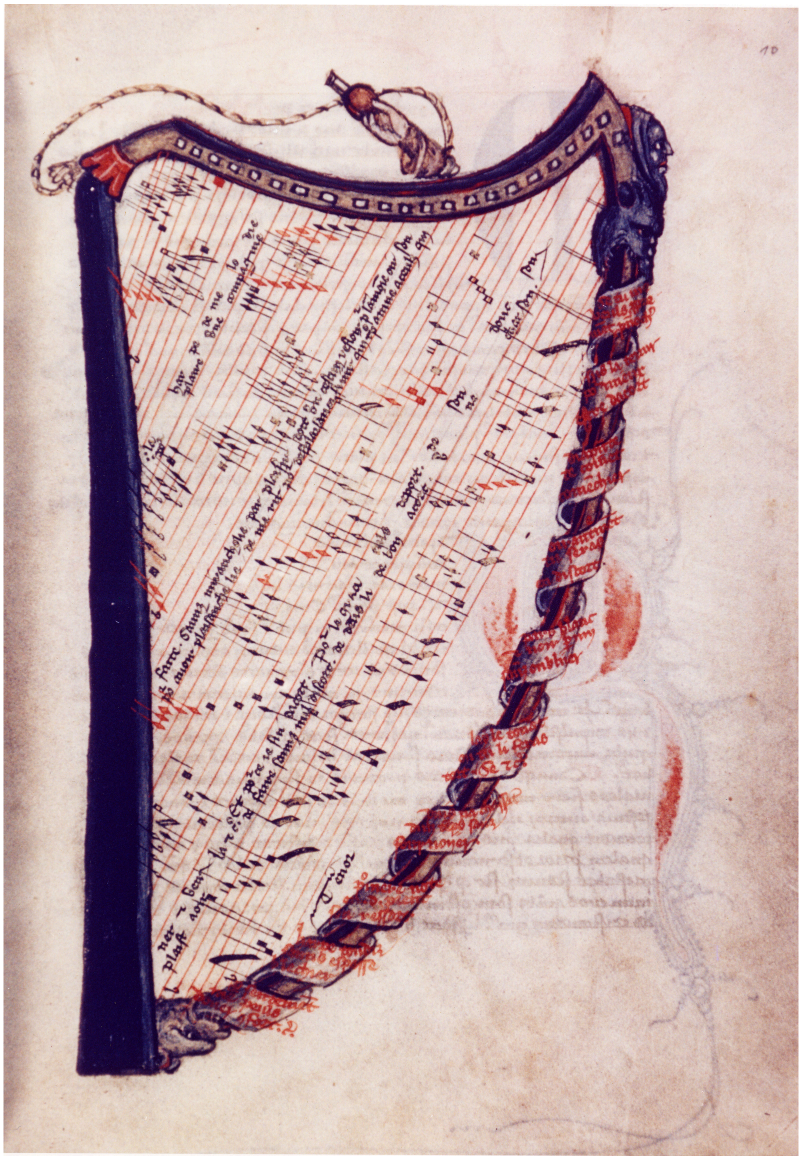



Figure 5: Jacob de Senleches, *La Harpe de melodie*, Chicago, Newberry Library, Ms Case 54.1, fol. 10r. Image by Newberry Library, Chicago.

More thorny issues can be demonstrated by further examples from the end of the long Trecento and the *ars subtilior* in Italy.

As is well known to students of notation, Lorenzo Masini's *Ita se n'era star in Paradiso* appears twice in succession in the famous Squarcialupi Codex.[7] The first copy (Figure 6) contains four different notes that consist of modified flags or little circles placed on the stem or below the note. The little circle could be added to the attributes data for stems and stem modifiers.

7. Florence, Biblioteca Medicea Laurenziana, Ms Pal. 87.

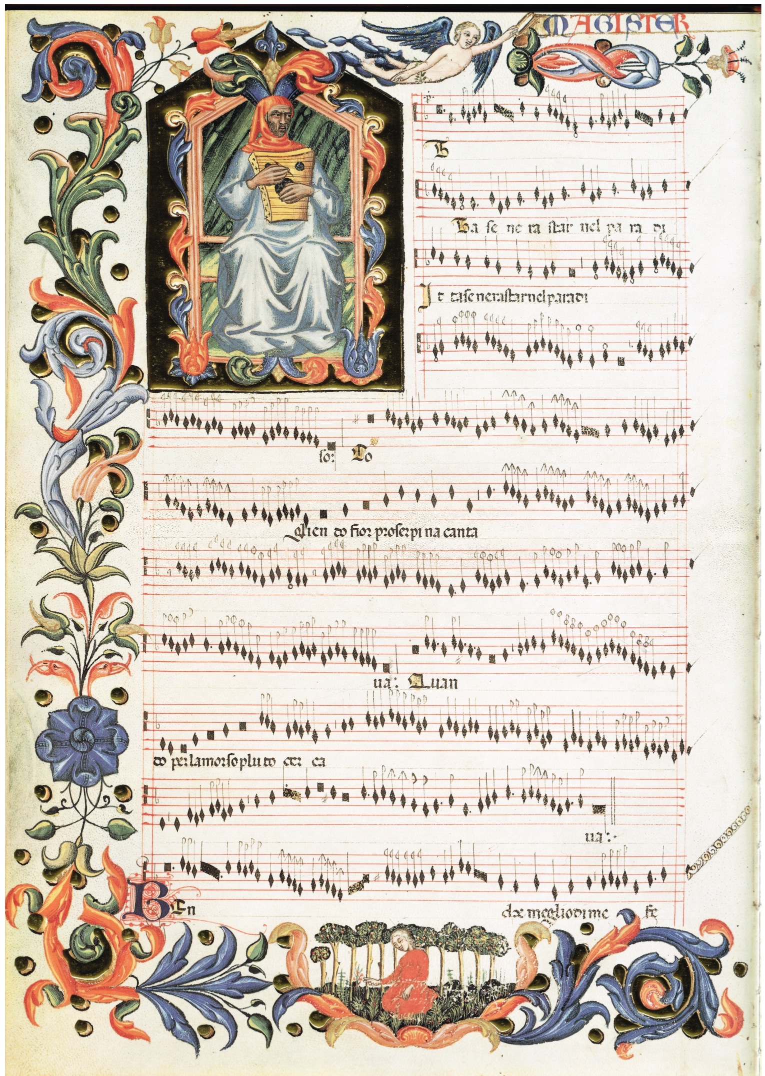
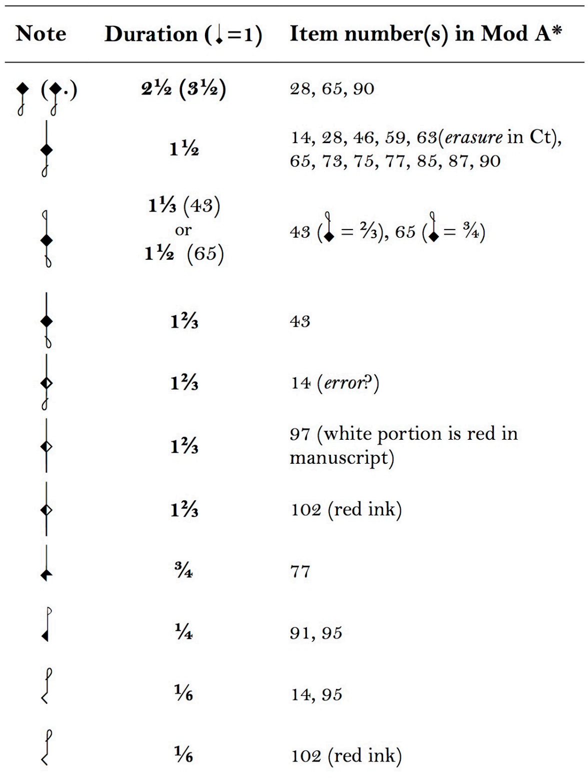



Figure 6: Lorenzo Masini, *Ita se n'era star in Paradiso*, Squarcialupi Codex, fol. 45v. Reproduced by the kind permission of the Biblioteca Medicea Laurenziana. Further reproduction prohibited.

Matteo da Perugia's notation presents several challenges, especially the scribal practice of carving off a half or even a quarter of the body of a note to indicate a corresponding reduction to its relative duration. Figure 7 illustrates some of these notational subtleties. The notes in Rows 5 to 7 also employ the notational device of partial coloration. Unlike full coloration where the whole black note is redrawn in solid red ink or as a hollowed out "white" note, partial coloration applies this writing process to only half of the note's body.[8] MEI currently has no simple way of encoding the visual appearance of these notes, although it can indicate their proportional reduction as a note attribute.

8. Accordingly, the note duration is affected only half as much as it would be for full coloration. Full coloration either lessens a note's normal (black) duration by a third or increases it by a half depending on context.



Note	Duration (♩=1)	Item number(s) in Mod A*
	2½ (3½)	28, 65, 90
	1½	14, 28, 46, 59, 63(*erasure* in Ct), 65, 73, 75, 77, 85, 87, 90
	1⅓ (43) or **1½** (65)	43 (♩ = ⅔), 65 (♩ = ¾)
	1⅔	43
	1⅔	14 (*error*?)
	1⅔	97 (white portion is red in manuscript)
	1⅔	102 (red ink)
	¾	77
	¼	91, 95
	⅙	14, 95
	⅙	102 (red ink)

Figure 7: Unusual notes in the works of Matteo da Perugia. The durations of these notes a signified by combining elements from two notes or changing the body of the note (including "cutting away" part of the note).

By way of a summary of some of the problems of encoding mensural notation into MEI, we note that mensural music often challenges the paradigm behind MEI. Compositions that have come down to us in the shapes of circles, mazes, harps and hearts cannot be easily rendered down into plain vanilla encodings. Their use of visual cues for indicating perpetual canon, for example, should not be subsumed into an arbitrary and foreign repeat sign. In the same vein, the mensurations, proportions and other transformations of a written part at the moment of performance, especially of what are referred to as isomorphic tenors [9], sometimes determined by verbal canons, produce musical results so removed from the notation that the crucial questions of "what are we really encoding?" and "how do we want to use these data?" becomes a pressing one.

In our case, encoding medieval music notation for research needs to provide an abstraction of the paleographic record. It also needs to provide data about the notated elements of music that might be useful for computer-assisted analysis. Of a lower priority is the building of diplomatic scores, although the solution to this question is really building the right tools for producing such scores, as Stinson has demonstrated by producing Score editions of several of the repertoires he has encoded. Making MEI fit for the challenge of encoding more medieval music will require careful work. Yet, just being aware of these issues raises the stakes for MEI as is progresses into the future as a model for the comprehensive encoding of chant and mensural notation.

Conclusions

The foregoing has served to demonstrate what we have learnt from the process of shifting from one encoding of medieval music notation to another. While the Neume and Mensural branches of MEI have proven adequate for several previous projects, our project, which represents the tip of the iceberg in the endeavour to encode medieval repertoires into something like MEI, has shown to us that while the established structures in these two branches are feasible, they are not completely adequate for the task at hand. Though we have needed to provide only selected examples in support of our case, we believe that they are sufficient to encourage any critic to put aside reservations that they might have held about the need to enhance MEI for the task of encoding chant and mensural music written down before c.1500. What scholarship stands to gain from the systematic encoding of these early music notations far outweighs the ordeal of revising either branch. Not only will a comprehensive method of encoding permit the large scale computer-assisted interrogation of complete repertoires, it will form an important partnership with the bold endeavour of developing optical music recognition systems for musical scores from every age. Already Stoessel is part of a project that will soon be using the NeoScribe flavour of MEI for the systematic investigation of canonic techniques and the role of musical notation in this process. It continues the endeavour that Stinson started over thirty years ago which aimed to encode medieval music and use this data to inform musicological enquiry.

Works cited

[1] Cuthbert, Michael Scott. “Trecento Fragments and Polyphony Beyond the Codex”. Harvard University, 2006. PhD dissertation.

[2] Gozzi, Marco. *Cantus fractus italiano: Un’antologia*. Musica Mensurabilis, 4. Hildesheim: Georg Olms, 2012.

[3] Günther, Ursula, John Nádas, and John Stinson. “Magister Dominus Paulus Abbas De Florentia: New Documentary Evidence” *Musica Disciplina* 41 (1987): 203-46.

[4] Hankinson, Andrew, Perry Roland, and Ichiro Fujinaga. “The Music Encoding Initiative as a Document-Encoding Framework” in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*. (http://ismir2011.ismir.net/papers/OS3-1.pdf)

[5] Huglo, Michel. “Dominican and Franciscan Books” in *The Calligraphy of Medieval Music*, ed. John Haines. Turnhout: Brepols, 2011, 195-202.

[6] Morent, Stefan. “Representing a Medieval Repertory and its Sources: The Music of Hildegard von Bingen” *Computing and Musicology* 12 (2001): 19-31.

[7] Schräder, Gregor. “Ein XML-Datenformat zur Repräsentation kritischer Musikedition unter besonderer Berücksichtigung von Neumennotation”. Eberhard Karls Universität Tübingen, 2007. Undergraduate dissertation.

[8] Stoessel, Jason. “Looking Back over the ‘Missa L’Ardant desir’: Double Signatures and Unusual Signs in Sources of Fifteenth-Century Music” *Music & Letters* 91.3 (2010): 311-42. (*doi: 10.1093/ml/gcq059*)

[9] Zazulia, Emily. “The transformative impulse” in *The Cambridge History of Fifteenth-Century Music*, ed. Anna Maria Busse Berger and Jesse Rodin. Cambridge: Cambridge University Press, 2015, 587-601.

MEI for Mensural Notation in the Thesaurus Musicarum Latinarum

Giuliano Di Bacco
Indiana University
gdibacco@indiana.edu

Perry Roland
University of Virginia
pdr4h@eservices.virginia.edu

Abstract

This paper stems from the project of redevelopment of the *Thesaurus Musicarum Latinarum* (TML), the online searchable archive of early music treatises in Latin. Over 25 years after the project's inception, work is in progress to enrich the large corpus of plain-text files by encoding them with TEI, to improve the resource's fruition and the discoverability of the text. Over that background, a quite more difficult challenge emerges: to make also discoverable the large quantity of notated examples found interleaved with the verbal text of the treatises. It was not in the original intent to make music notation systematically searchable, but attempts were made to capture some facets of mensural notation through a code specifically designed for the project. The aim of this paper is both to discuss these earlier encoding attempts, with their shortcomings, and to propose our customization of MEI as a possible starting point to expand the current specifications. This is proving useful in the present state of the project: more work will be necessary to take care of other mensural features and flavors, and of course of chant and non-mensural notations, but MEI is proving apt to the job. The TML provides an ideal benchmark for developing a more specialized model of markup that will be useful for many other scholarly purposes. (The transcript below reflects the status of the TML project and of MEI at the time of the conference.)

Introduction

With its 2.0 release, MEI has reached maturity as an encoding system permitting scholars working on traditional scores in Common Western Music Notation a high level of editorial and analytical sophistication. Of course the achievement is eloquent, but we are still at the beginning of the journey. It is significant that this conference hosts the founding meetings of new Interest Groups with the participation of several specialists who agree that the time is ripe to extend the current specifications to take care of pre-1600 music notations.[1]

This paper presents the case of the *Thesaurus Musicarum Latinarum* (TML), the digital archive of music treatises in Latin from the late Antiquity to the seventeenth century. Discussion of this well-established resource provides an example of how MEI can be very helpful to overcome the difficulties of the past in the encoding of early music repertories. The TML contains texts referring to and exemplifying most if not all known notational systems in use during the medieval period; below we focus on some facets relevant to the encoding of 14th-century mensural notation, a current priority for the project.

1. The conference hosted the inaugural meetings of the Interest Groups on Neumes, Mensural Notation, and Lute Tablatures.

The TML was started in 1988 by an international group of scholars and led by Thomas Mathiesen of Indiana University. It was one of the earliest projects in historical musicology to create a searchable archive of documents in electronic format and to distribute them through a promising new medium, the Internet. The main mission of the project is to make available in electronic format all existing treatises on music, intended in the widest sense. These treatises range from theoretical discussions concerning the general principles of music as a liberal art and music's intersection with intellectual and scientific history, to more practical treatises on specific techniques, genres and styles. A number of treatises are anonymously transmitted, but authors represented include a range of canonical music theorists from Aurelian of Réôme and Guido of Arezzo to Gioseffo Zarlino and Giambattista Doni. Polymaths are also included, such as Church fathers, philosophers, and early scientists from Aristotle and Augustine to Johannes Kepler and René Descartes.

At a time when editions of these works were often scattered in difficult-to-locate publications and journals, the TML was designed to be a resource making them all available in one place. The treatises were converted to ASCII characters through OCR or retyping and then proofread twice. The resulting plain-text files, organized in folders by century, were distributed via FTP or on a CD, while other basic metadata, bibliographic information, and indices were available separately in print.[2] The corpus consists mainly of verbal text, but a number of tables, diagrams, and other figures are present to illustrate the matters discussed. Most of the more practical treatises also have notated examples intermingled with the text. All these non-verbal materials, when in the public domain or whose reproduction was authorized, were scanned and added to the relevant folders as GIF files, sampled at a resolution low enough to accommodate the transfer rate of dial-up connections.

The ability to read these works on a personal computer was a true revolution for scholars. Even more sensational news was the possibility of searching the corpus for one or more terms, thus greatly facilitating the study of terminology and the identification of parallel passages or unattributed quotations. In 1998, a website was created that gave access to the texts in a basic HTML. Points of access are manually-compiled lists by century with links to the individual files, and, most important of all, a web-based search facility.[3] Because of these writings' importance to musicologists and early music performers, and given the accuracy of these electronic versions, the TML established itself as one of the most authoritative online resources for the history of music. There is also evidence of its use as a research aid by non-music scholars, such as historians of philosophy and science.

Redevelopment

As a result of over 25 years of work by a number of contributors, the TML currently contains 6.5 million words and several thousand graphic files for ca. 960 texts (and counting), all reproduced from published editions or directly transcribed from primary sources. The TML is still hosted in Bloomington, at the Center for the History of Music Theory and Literature of the Jacobs School of Music, along with three sister resources created over time and dealing with analogous documents in Italian, English, and French.[4] Work is underway to redevelop these projects with a three-fold goal: to update data model and technology, to increase discoverability of the content, and to extend purposes and intended audience.

2. The *Canon of Data Files* was a document updated yearly and distributed on-demand. It was published in book format as [1].
3. The website that was current at the time of the conference is still viewable, courtesy of the Wayback Machine of the Internet Archive: web.archive.org/web/20150304171939/http://www.chmtl.indiana.edu/tml. See one of the century indices: web.archive.org/web/20150213002358/http://www.chmtl.indiana.edu:80/tml/13th/13TH_INDEX.html; a short 13th-century anonymous treatise: web.archive.org/web/20150920024750/http://www.chmtl.indiana.edu/tml/13th/ANOQUA_TEXT.html (note the links to graphic files that open in a new window).

More specifically, new interim websites are under development for all these projects. They will still use the basic HTML files but will be driven by an SQL database accommodating existing metadata, new metadata and indices, and a new search mechanism.[5] At the same time, work is in progress to add layers of annotations through TEI markup (just to the TML corpus for now), and this will be a considerable improvement. The reasons for moving towards TEI markup are manifold. It allows introducing amendments to texts that were reproduced as in the source without editorial intervention. TEI also supports making structural information machine-readable and enriching the content with new metadata and scholarly annotations. Structural markup will improve readability and permit filtered searches and the generation of tables of contents. Content markup is aimed to identify people's names, titles of musical and non-musical works, places mentioned, and any quoted text that can be recognized, so that faceted browsing and searching will be available.

Beyond browsing and searching, today many computational methods can be employed when large archives of texts as the TML are available. These methods can then make the content discoverable further, which also helps to answer emerging research questions. For instance, topic modeling can provide statistically-based suggestions on existing relations between texts at content level, and further exploration is possible using tools for automated collation. Even more interestingly, network analysis can visually represent complex relations between people and works mentioned in the corpus. Tagging all places mentioned, along with places of provenance and origin of the manuscript sources of extant versions can show the circulation of the texts on a map. All these methods can be combined or applied to sub-corpora. And these methods work best when data is structured—thus providing another reason for having embarked on an extensive markup campaign. The future website, built on the TEI-encoded data and empowered by the above features, will also have more points of access, namely entries on the most influential authors and theoretical themes. Ultimately it is hoped that the TML will develop also as a place for publication of new scholarly contributions on these writings and their sources.

Musical examples: the TML Code

Over 10,000 notated examples are found interleaved with the text in the TML corpus. These examples range from small fragments of music to more substantial scores, either quoted from plainchant and other repertories, or created specifically to demonstrate the procedures and rules described. As a text-oriented project, the original plan for the TML did not include strategies to make the musical content fully discoverable, and a general policy has been set to make all non-verbal content available as graphics. Great care has been made to capture any text appearing in whatever portion of a work is reproduced as a graphic, so all text underlay and any other textual label present in these examples is searchable. However, in order to answer old and emerging research questions, the discoverability of musical content proper has become a growing priority.

Just to make an example, one may want to find out correspondences or similarities between the theorists' chosen examples and known pieces of music. The possibility to discover connections between treatises and extant music pieces can benefit the study of both, providing new indirect evidence about their circulation and contacts. So, actual quotations from known compositions are found, and the information may be explicit

4. *Saggi musicali italiani* (SMI), directed by Andreas Giger (Louisiana State University): www.chmtl.indiana.edu/smi; *Traités sur la Musique en Français* (TFM), directed by Peter Slemon (Indiana University): www.chmtl.indiana.edu/tfm; *Treatises on Music in English* (TME), directed by Peter Lefferts (University of Nebraska-Lincoln): www.chmtl.indiana.edu/tme.
5. The new TML website that was in preparation in 2015 is now online at the same URL: www.chmtl.indiana.edu/tml.

in the text. This is the case of the ballata by Francesco Landini illustrated in Figure 1, below. But it would be a hard task to find out other similar correspondences through visual examination only. As it would be, more generally, to find occurrences of specific notational details, or musical patterns or other features. The forthcoming website[6] adds a 'lightbox' gallery intended to help browsing more easily all the graphics available for each treatise, but an utility capable of searching for a sequence of intervals or a rhythmic pattern would be a highly desirable addition. The use of some machine-readable code is obviously a precondition to making this possible.

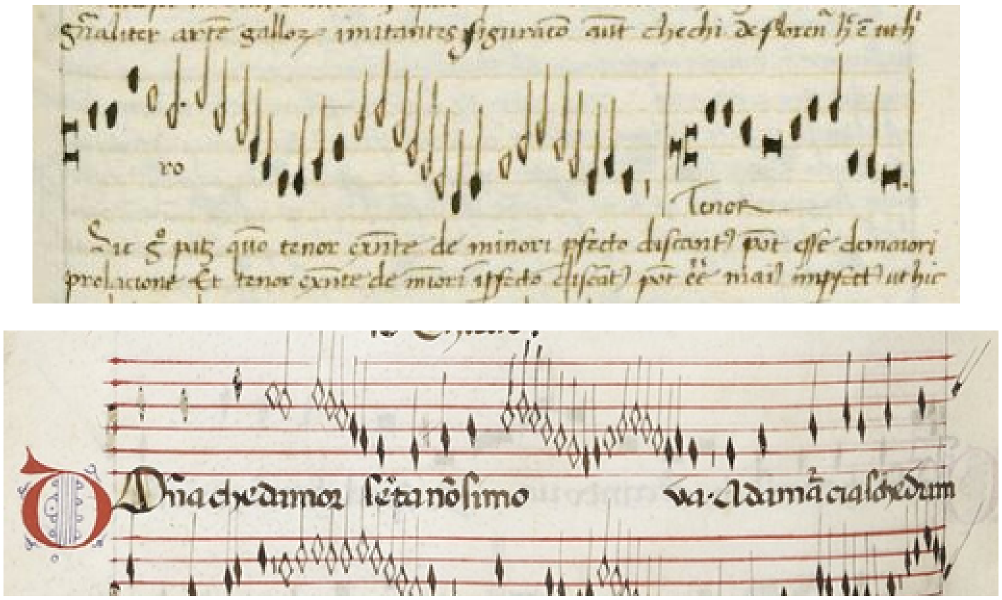


Figure 1: Above: beginning of Francesco Landini's ballata "Donna che d'amor senta" quoted in an anonymous treatise on mensural notation (Ms I-Fl Plut. XXIX 48, fol. 81r). Below, the beginning of the full piece in a Trecento anthology (Ms F-Pn 568, fol. 104v). See the discussion of this piece in [2]. Images of both manuscripts are reproduced by kind permissions of their library owners: Biblioteca Medicea Laurenziana and Bibliothèque nationale de France.

As the expert user of the TML knows, a code is already in use for a number of examples written in mensural notation ca. 1280-1450. This follows a system created by Mathiesen around 1990.[7] This is yet another case of an encoding system with a limited scope, like those examined by Eleanor Selfridge-Field in her illuminating Keynote presentation given at this conference. Thus the aim of the TML *Code for Noteshapes* is to represent the special musical symbols employed in this notation, from notes and ligatures to mensuration and proportion signs. Figure 2 below shows a musical example, and its encoding follows:[8]

Figure 2: Close-up from Ms GB-Ob Bodley 842, fol. 53r (Franco of Cologne, *Ars cantus mensurabilis*). Reproduced by kind permission of the Bodleian Library, Oxford.

6. Current at the time of publication.

7. The specifications are available on the TML websitewww.chmtl.indiana.edu/tml/about/policies but see also [3].

8. www.chmtl.indiana.edu/tml/13th/FRAACMO_MOBB842.

```
[ClefC2,Lig2art,Lig3ad,Lig3aart,Lig4add,Lig4adod,Lig5adodart,Lig5addd,Lig3ad,Lig9aaaddddd,Lig3ad]
```

The meaning is that, for instance, the first *ligatura* (a neume with mensural significance) is `Lig2art`, that is, "formed by 2 notes, ascending, and the final head is turned over"; similarly the sixth *ligatura* (`Lig5adodart`) is "formed by 5 notes, ascending, then descending twice (in oblique form), then ascending again with the final head turned over". In mensural terms, the motion (ascending or descending) and the form (*recta*=squared, or oblique) are the defining elements. The exact representation of the original shapes permits calculating the (relative) durations of the notes (performed durations depend on the context).

However, the code's shortcoming is quite obvious: pitches are not represented. As a review contemporary to the introduction of this code pointed out, the TML aims at "stressing the identity of each [sign] but not expressing the relationship between [them]".[9] Incidentally, another project on early music theory, started in around the same period, took a complementary approach: Sandra Pinegar for her THEMA project created an alphanumerical code that focuses on pitches and simplifies the representation of the actual signs. See again Figure 2, for which THEMA uses the following:

```
([s4c2] 2CCcd 3CCcdc 3CCcdf 4CCdedc 4CSdedc 5CCcdbac 5CCdedcb 3CCcdc 9CCcdefedcba 3CCcdc)
```

Here, in addition to lowercase letters representing pitches, letters C and S stand for "cum proprietate" and "sine proprietate"—the expressions used by the author of the treatise, Franco of Cologne, to categorize these shapes for their rhythmic meaning.[10]

Returning now to the TML, consider that for decades it was not uncommon for CHMTL to be denied permission to use in their websites even close-ups from published editions and library reproductions of original sources. In these cases, the TML alphanumerical code has been useful to represent in a human-readable format at least the form of the notation.[11] A few users even reported that they have used the coded examples to search for note shapes in the corpus. This is one of the reasons why, as work proceeds on marking-up the treatises in TEI, the original TML codes are being preserved with the element `notatedMusic` and will remain available.

However, the usability of this code, which has not been adopted elsewhere, remains quite limited. Nor does the possible integration with elements from the THEMA code look promising enough. Thus the plan is to move on with some other code. The ideal replacement code would be one that could be used across the whole corpus (that is, not specifically designed for mensural music); possibly able to generate a graphic rendering to be used in place of or in addition to the scanned imeges; possibly XML-based in order to be easily integrated

9. [4].
10. The THEMA project (Music THEory of the Middle Ages) included transcriptions of 18 manuscript copies of 14 Latin theoretical treatises related to *musica mensurabilis* of the thirteenth century. The original website is offline but can be retrieved through the Wayback Machine of the Internet Archive. See web.archive.org/web/20070519150932/http://www.uga.edu:80/~thema/fra-bod.html for the page with the cited example.
11. An overview of the encoding systems for medieval and Renaissance notations being developed around 1990 is given in [4]. The most advanced system of the time was SCRIBE, created by John Stinson and others. About the current prospects of redevelopment of the latter, see [5] and the paper by Stinson and Stoessel in this volume.

in the TEI markup. Ultimately MEI appears as the most promising solution, because it aims to be an universal system for structural and semantic representation of all kinds of music, and because of the readiness of integration in a TEI document implicit in its format and architecture.

Adopting MEI

While the ultimate goal would be to make the examples discoverable through a browse/search utility by duration and pitch, MEI markup would also bring several bonuses. For example, and first of all, the possibility to add transcriptions in modern notation;[12] Then, one day, multiple versions of the same example from different sources could be available encoded in TEI, to help producing collations, if not new critical editions. There are also a number of features that scholars may want to see implemented thanks to the possibility of encoding the music, for example connected to the fact mentioned earlier that the examples may be derived from actual compositions. In those cases where such derivation is known (see again Figure 1), the encoders are encoding the piece as a whole so that the quotation could be shown in its context. Along the same lines, cases are found of pieces cited in the text in reference to certain features, but not accompanied with actual musical quotations (notated examples): also these pieces could be made available in the website for the user to understand the reasons of such references.

There are three main obstacles to overcome towards the realization of these aims. One: MEI support is still scarce for the notations involved, especially those of the late medieval period. Two (a corollary of the former): it would be desirable that whatever is encoded can be then used for display in the browser. Three: encoding the whole corpus would take quite a lot of time. Let's proceed examining these issues in reverse order.

As timing is concerned, Michael McClimon (Bloomington) has compiled for CHMTL a routine transforming the TML Code in an XML MEI-like skeleton, to which the missing component (the pitches) could be manually supplemented. For all musical examples not previously TML-coded, CHMTL would like to make some experiments with Optical Music Recognition, perhaps starting with using Aruspix to process the sub-corpus of examples in white (hollow) notation, the focus of that software; also to this effect, CHMTL has started a campaign of new scanning of the musical examples to obtain graphics at a higher resolution. Finally, it is hoped that MEI-encoded data may be made available by other projects, such as chant databases.

As for engraving and display, CHMTL is currently testing the possibility of using, at least in the interim, an extension of LilyPond called *MensuralBlack* developed by Lukas Pietsch (Hamburg).[13] In a forthcoming release of this extension, planned as a collaboration with the CHMTL, `.ly` coding could be derived from the TML Code, with some manual adjustments and again input needed to add information on pitches. This will work as a kind of rendering front-end for MEI data, certainly helpful to provide a graphic representation in those cases when permission was denied to publish any proprietary reproductions, as mentioned above. Indeed, the ultimate goal would be to have all examples displayed dynamically in the browser: Don Byrd (Bloomington) has started working on a project-specific customization of Verovio.[14]

12. Concurrently with the plan of enlarging the scope and reach of the TML project by including translations in English and in other modern languages.
13. The current specifications of the *blackmensural.ly* template are available on the personal website of the author: www.lukas-pietsch.de/Music.
14. A collaboration between CHMTL and the Visualization and Analytics Group of the Pervasive Technology Institute of Indiana University.

However, the most obstructive obstacle is that the TML project deals mostly with early notations requiring very specialized markup that is not yet available. In particular, MEI works already quite well in representing white (hollow) notation, but the situation gets far more problematic for the sub-corpus currently in focus for the TEI-encoding project where various flavors of black notations (Franconian, Ars nova, Trecento) are used—further complicated by the multiplication of signs and rhythmical intricacies of the *subtilior* style. Echoing the wishes formulated at the beginning: this conference will see the start of a process of revision and extension of the specifications for encoding mensural notations within the new Interest Group. In looking forward to these discussions, we would like to share some preliminary thoughts originated by some problems we encountered.[15]

1. The current mensural module introduces very specific values for `@duration` (`brevis`, `semibrevis`, `minima`, and so on). They work well to represent relative durations across the entire repertory, but some ambiguities remain as far as the representation of the signs is concerned. This is because during the period of application of the mensural system the correspondence between sign and name is not univocal, since two different set of signs were used: what at the beginning was *black* (solid note heads) later became *white* (hollow). For example, depending on the chronology, a solid diamond with an upward stem could be either a *minima* or a *semiminima*. Conversely, an encoded `duration="brevis"` is ambivalent for either a solid or a hollow square. Moreover, in both black and white flavor, mensural notation has a feature called coloration, to be encoded in MEI with `@coloration`: this represents a rhythmic change for example from ternary to binary, but again leaves ambiguous the representation of the sign indicating it. A colored note head would be solid if hollow is the norm, and would be hollowed if the solid is the base sign. We don't want to have to specify the solid or hollow status of the head every time(!), so it seems necessary to add a declaration about the context, the base notational flavor we are operating within. Simple but much needed: we propose to use the attribute `@notationtype` with at least two values, `mensuralBlack` and `mensuralWhite`. Such an identifier will be useful to avoid ambiguity in other circumstances. For instance, while in white notation upward and downward stems in base noteshapes (not in a ligature) are rhythmically equivalent (depending on the position of the note on the staff), in black notation the direction of the stem is always a meaningful modifier of the basic shape.

2. Concerning modifiers, in black notation a flag drawn on the right of a stem has a different meaning than one drawn on the left. In the more sophisticated notation typical of the *ars subtilior* style such flags can be found either on upward or downward stems, or on both, again signaling distinct rhythmical meanings.

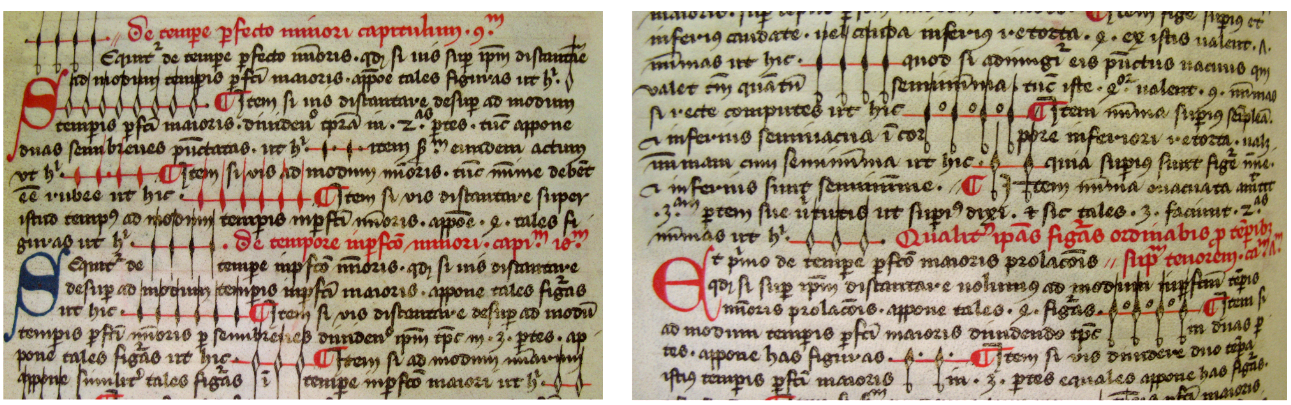


Figure 3: Close-ups from Ms US-Cn 54.1, fols 8v and 9r (*Tractatus figurarum*). Reproduced by kind permission of the Bodleian Library, Chicago.

15. At the time of the conference the authors were both members of the MEI Board, worked together at defining these initial priorities; Giuliano Di Bacco, project director of the TML, launched the initiative of an Interest Group on Mensural Notations.

Let's consider two examples from late fourteenth-century sources, a theoretical and a practical one.

In Figure 3 (above) the variously attributed treatise *Tractatus figurarum* deals with the definition of such special signs. In Figure 4 (below) a piece in the Chantilly Codex shows a concoction of some of the most idiosincratic noteshapes ever conceived: stems downward or upward or both, different types of flags on the right or left of upward or downward stems, and drawn in black or red, filled and unfilled. (Note that in both sources the note heads can be even half-filled.)

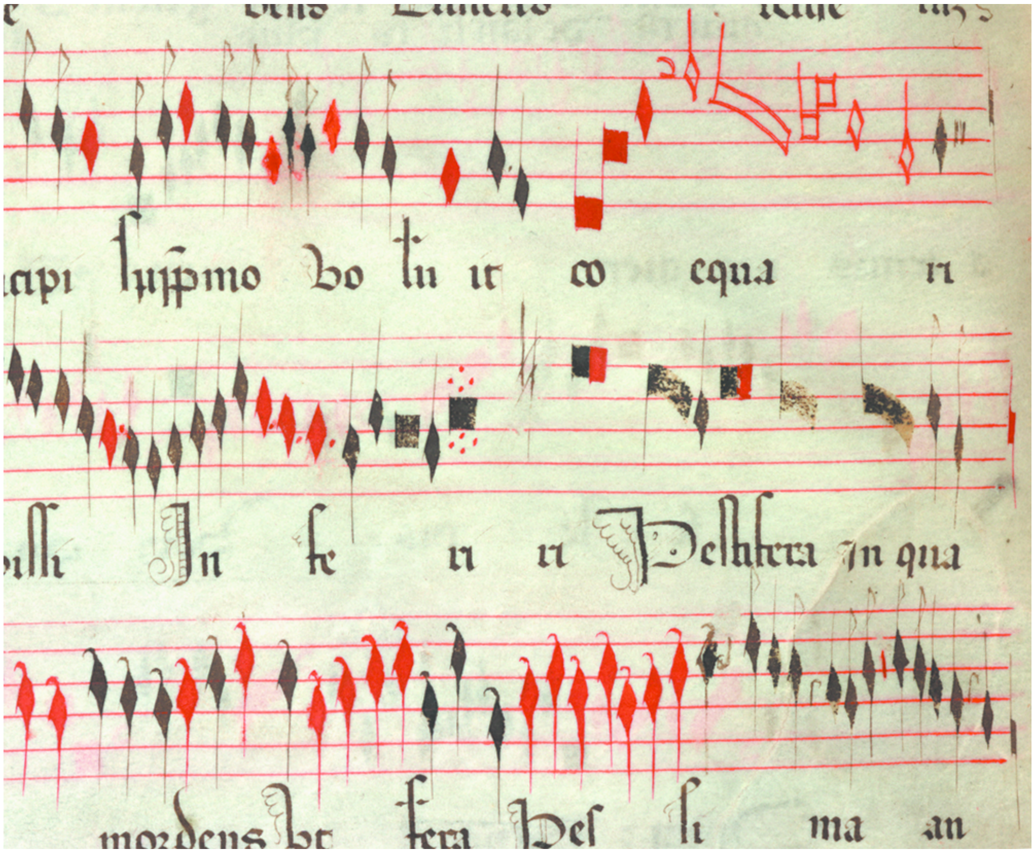


Figure 4: Close-up from MS F-CH 564 (the Chantilly Codex), fol. 48v. Reproduced by kind permission of the Bibliothèque du Château de Chantilly.

It is relatively simple for MEI to handle this situation just introducing new attributes and values. Then for clarity, and to prevent confusion, a mechanism (schematron rules) could be in place to make this markup available only when `notationtype` is `mensuralBlack`, as described earlier. Or to distinguish its usage and meaning between one notational dialect and another.

Parenthetically: this example serves to illustrate the situations that we encounter when encoding the treatises in the TML: sometimes short snippets of notation present new challenges, probably easier to analyze and fix than if we were dealing with full compositions; we are expanding our markup model step by step, and then entire compositions can be encoded.

3. MEI considers the classic mensuration signs and their modifiers (C and O, with or without a dot), and these are enough to represent most occurrences of mensural notation. However, other signs are found to represent the four degrees of division: *maximodus* and *modus* (later also called *modus maior* and *modus minor*), *tempus* and *prolatio*, signalling different notational habits, such as the use of two or three dots; or rectangular boxes with three or two strokes (vertical for the relation between *maxima* and long, horizontal between long and breve). Moreover, not all black notations obey to the same mensural principles. The most idiosyncratic and important case is perhaps that of the notation of the Italian Trecento, with its system of successive degrees of divisions of the breve. In this system, six special mensurations are operating, named after the maximum number of *minime* possible in each type of division, a range from three to twelve. These *divisiones*, as theorists call them, in the sources are indicated by letters referring to their names: *q(uaternaria)*, *s(enaria) p(erfecta)*, *s(enaria) i(mperfecta)*, *o(ctonaria)*, *n(ovenaria)*, and *d(uodenaria)*. It is not just a matter of names and symbols, because the system implies different rhythmic structures, a different interpretation of patterns of longer/shorter durations when compared to the contemporary French system, so these need to be represented univocally as well.

A few of the notational details mentioned above are shown as Figure 5:

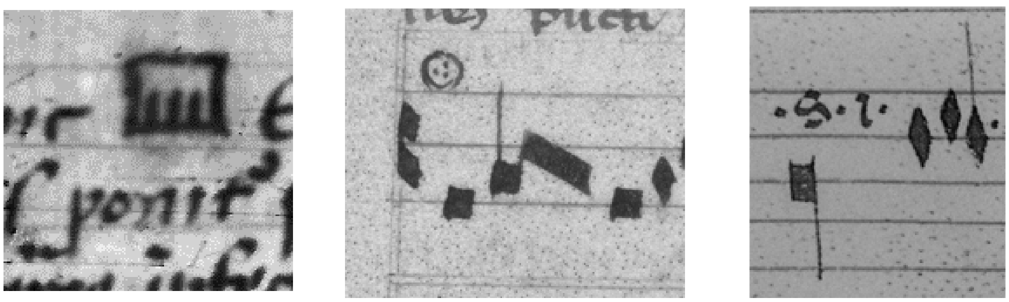


Figure 5: Close-ups from Mss I-Ma H.165 inf., fol. 21r; CH-E 689, fol. 87v; I-Fl Pal 87, fol. 74v. Images reproduced by kind permissions of their library owners: Biblioteca Ambrosiana di Milano, Stiftsbibliothek des Klosters Einsiedeln, Biblioteca Medicea Laurenziana.

Conclusions

Beyond the general scope and value of MEI, scholarly projects dealing with both music notations and text should find very convenient to adopt this encoding system because of the smooth integration of MEI code into a TEI document. Music sources from the period before the standardization of mensural (white) notation are rich with peculiarities that need to be taken into account, and MEI is not yet, but will be perfectly able to represent such idiosyncrasies. The necessary process includes the deconstruction and analysis of the components (for example the shape of a note head, the presence, number and direction of one or more stems and flags, their color) and their reconciliation in a chain of tags and attributes representing aspects of a specific notational dialect. We welcome the opportunity of this conference to start discussing solutions such as those summarized in the list appended below. In doing so, we hope that scholarly communities can start building rich repositories of music written in “Uncommon” Western Music Notations and encoded in MEI.

Appendix

Proposed additions and changes to the Mensural module. A simplified list

NOTATIONAL DIALECTS

@notationtype	(mensuralWhite | mensuralBlack)
@notationsubtype	consider: (franconian | arsnova | trecento)

NOTE HEADS

@head.shape	add: (square)
@head.reversed	(false | true)

STEMS

@stem.dir	consider adding: (updown)

UPWARD STEMS

@upstem.angle	add: (straight | obliqueleft | obliqueright | hirundinis)
@upstem.color	*data.COLOR*
@upstem.fillcolor	*data.COLOR*
@upstem.len	*xsd:decimal*
@upstem.pos	(left | right | center)
@upstem.x	*xsd:decimal*
@upstem.y	*xsd:decimal*
@upstem.flags	*xsd:positiveInteger*
@upstem.flags.pos	(left | right | end)
@upstem.flags.shape	(straight | angled | curled | circular | rounded | flared | extended | hooked)
@upstem.flags.color	*data.COLOR*
@upstem.flags.fillcolor	*data.COLOR*

DOWNWARD STEMS	
@dnstem*	as for @upstem* above

MENSURATIONS AND MENSURATION SIGNS	
@maximodus	(2 | 3) as an alternative to @modusmaior
@modus	(2 | 3) as an alternative to @modusminor
@divisio	(d | n | o | q | si | sp)
@mensur.sign	add: (quad | tract)
@mensur.dots	(1 | 2| 3) and deprecate @mensur.dot then, similarly:
@dots	(1 | 2| 3) deprecate @dot
@mensur.dots.align	(center | vertical | horizontal | trigonup | trigondn | right | left)
@mensur.sign.mod	(slashed | dblslashed | dashed | dbldashed | vcut | dblvcut | circled | dblstrokedn | dblstrokeup | trplstrokedn | trplstrokeup) and deprecate @mensur.slash

Acknowledgements, and Afterword

Support for the redevelopment of the TML and the TEI/MEI encoding comes from the OVPR and JSoM of Indiana University, and relies on the work of several people in the CHMTL staff, with major contributions from Dana Barron, Daniel Bishop, Magda Dragu, Adam Hochstetter, Michael McClimon and Sebastian Bisciglia. As of today, thanks to the MEI community effort, some of the proposals reported above have been revised and implemented. Substantial progress has been made by Don Byrd on extending the capabilities of Verovio. Several MEI-encoded musical examples and related full pieces will feature in a future release of the TML website.

Works cited

[1] Mathiesen, Thomas J., ed. *Thesaurus Musicarum Latinarum. Canon of Data Files*. Lincoln, NE: University of Nebraska Press, 1999 (Publications of the Center for the History of Music Theory and Literature, 1).

[2] Balensuela, C. Matthew, ed. and transl. *Ars Cantus Mensurabilis Mensurata Per Modos Juris*. Lincoln, NE: University of Nebraska Press, 1994 (Greek and Latin Music Theory, 10)

[3] Mathiesen, Thomas J. "Transmitting Text and Graphics in Online Databases: the 'Thesaurus Musicarum Latinarum' Model" *Computing and Musicology* 9 (1993-4): 33-48.

[4] "Encoding Neumes and Mensural Notation" *Computing and Musicology* 6 (1990): 23-35.

[5] Stinson, John and Jason Stoessel. "Encoding Medieval Music Notation for Research" *Early Music* 42 (2014): 613-617.

[6] "Thesaurus Musicarum Latinarum" *Computing and Musicology* 6 (1990): 133.

Encoding Genetical Processes

Richard Sänger
Beethovens Werkstatt
saenger@beethovens-werkstatt.de

Johannes Kepper
Beethovens Werkstatt
kepper@edirom.de

Abstract

This paper describes a model for genetic criticism in MEI, which takes slightly different approaches than the corresponding module in TEI.

Introduction

One hundred and eighty-eight years ago, Ludwig van Beethoven's original workshop was finally closed, and for obvious reasons, there is no chance to reopen it. However, Beethoven left behind a significant number of artifacts from his workshop, and there is great interest in exploring and understanding the highly complex processes in Beethoven's œuvre. Using an almost archeological approach, our project, called "Beethovens Werkstatt", seeks to dig into Beethoven's manuscripts and other materials. Our methodology is based on the concepts of the so-called `critique génétique` [1], and we present our findings in digital editions.

The ultimate goal is to publish a three-fold edition of the opus 120 Diabelli Variations by the end of our project, which is scheduled for 2029. Until then, the project seeks to develop concepts and techniques to combine a traditional work-centered edition with both a conventional source edition and an edition focussing on the genesis of the work.[1]

One of the first steps on that path is to identify different types of variants and classify their underlying compositional processes. In order to illustrate such variants, an initial prototype was developed in December 2014. It utilizes a passage of variants from the end of the first movement of the autograph of the piano sonata op. 111 (D-BNba BH 71). On page 14 of this manuscript, you will immediately notice several cancellations in the middle of the page. Furthermore, you will find "stacked" writing layers, written with both ink and pencil. From the five variants he produces on this page, it becomes immediately evident that Beethoven was struggling for a satisfying textual solution.

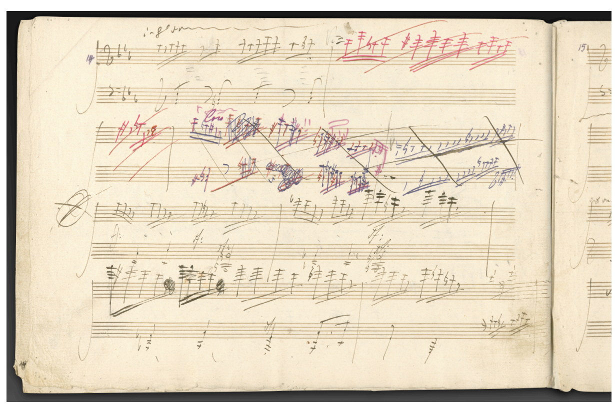



Figure 1: Digital Manipulation of Ludwig van Beethoven: Piano Sonata op. 111, Autograph (D-BNba BH 71), p. 14

1. For a complete overview of the project, please visit its website at http://beethovens-werkstatt.de.

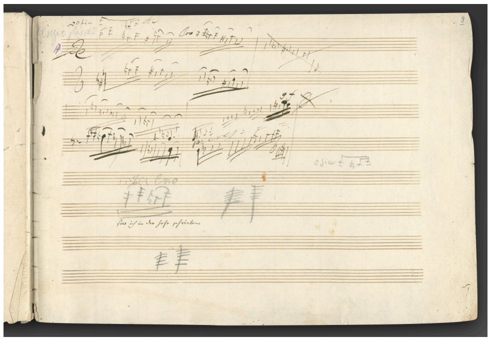



Figure 2: Digital Manipulation of Ludwig van Beethoven: Piano Sonata op. 111, Autograph (D-BNba BH 71), p. 17

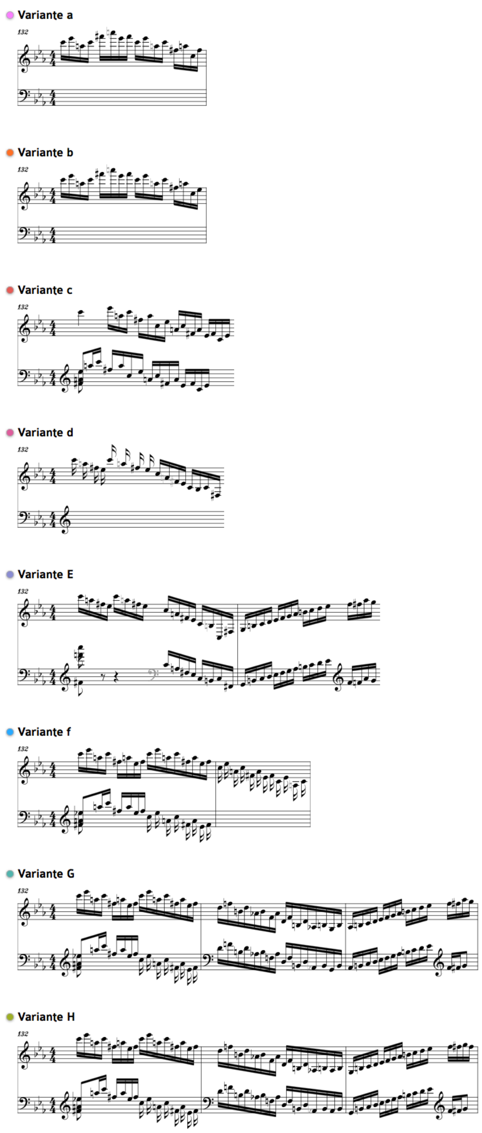


Figure 3: Overview of transcribed variants of op. 111, as seen in http://demo.beethovens-werkstatt.de.

However, after reaching the end of the piece on page 16, he again comes back to these measures on page 14 and revises them again. As there is no space left on this page for carrying out his modifications, Beethoven uses the empty page 17 of the manuscript for writing three more variants. In order to connect to this new workspace, Beethoven adds a `Vi=`-entry, which links to a corresponding `=de` on the new page.

Figure 3 shows how this variant passage looks when edited in a synoptical form. In these transcriptions, you can see the sequence of variants, sorted chronologically one below the other. What we are seeing here in the transcription are the textual layers: that is, the musical content of the text, without any other meta-textual symbols, such as deletions, cross references, verbal texts or musical letters. Thus, our transcription shows a “cleaned” text, which is completely detached from all graphical aspects of the writing process. These graphical aspects are fully contained in the manuscript itself, hence it is unnecessary to double that information in the transcriptions.

Beethoven’s op. 59.3 as second case study

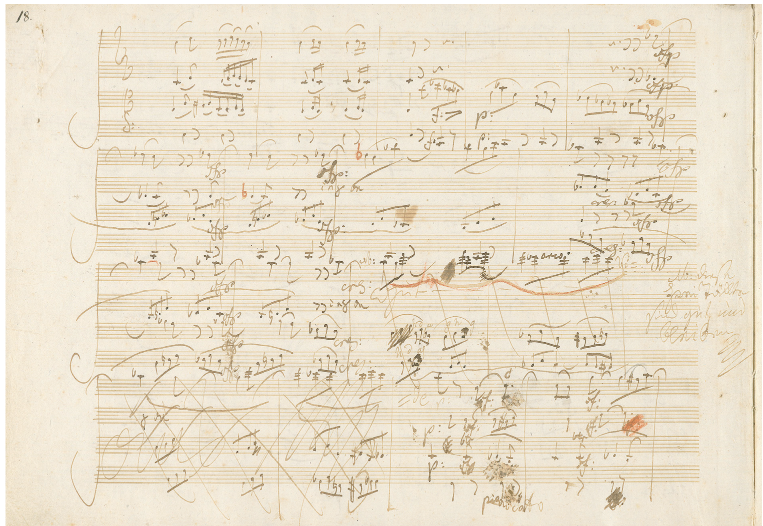



Figure 4: Ludwig van Beethoven, String Quartet op. 59.3, p.18

Recently, the project started to work on a second example, which will be published soon, along with an improved prototype. This time, we are working with Beethoven’s string quartet op. 59, number 3, which was composed in 1806. In the second movement, on page 18 of the autograph, we find another interesting passage. Among the various cancellations and crossed out text, it is the red pen and the verbal annotation at the margin which attract our attention. By looking closer, next to some notes one can spot letters, which verify the pitch. Moreover you find the words “gut” and “aus” as well as another “Vi=de” link. Once more, this instruction clarifies the correct succession of measures. But the essential information, the “key” to understand the variants and their progression, is the marginal note on the right side. Beethoven wrote: *"Nb: diese zwei Täkte sind gut und bleiben"* [*"Nb: These two measures are fine and have to remain"*]. The words “gut” and “aus” also help to reconstruct this passage.

The genesis of this passage may be explained that way: First, Beethoven wrote down two bars. He immediately cancelled them and wrote two more bars as a replacement. After some minor modifications, he decided to revert to his initial approach, and so he canceled the replacement bars, restored and subsequently modified his first variant. In order to clarify the sequence, Beethoven added the two comments “gut” and “aus” in the text, and the marginal note, which leave no doubt about his final decision.

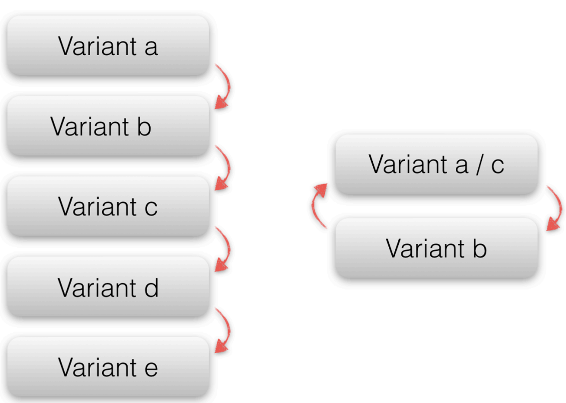


Figure 5: Different successions of variants: Series of substitutions (left) vs. restitution (right).

Comparing the sequences of variants from our two examples, the first shows a series of interventions, where Beethoven substitutes existing text with new text, which he substitutes again and so on. The second example is different: Here, Beethoven restores and then modifies an already existing text. The progress of restitution is different from that of substitution (see Figure 5). Those differences had consequences for our encoding model, which will be discussed below.

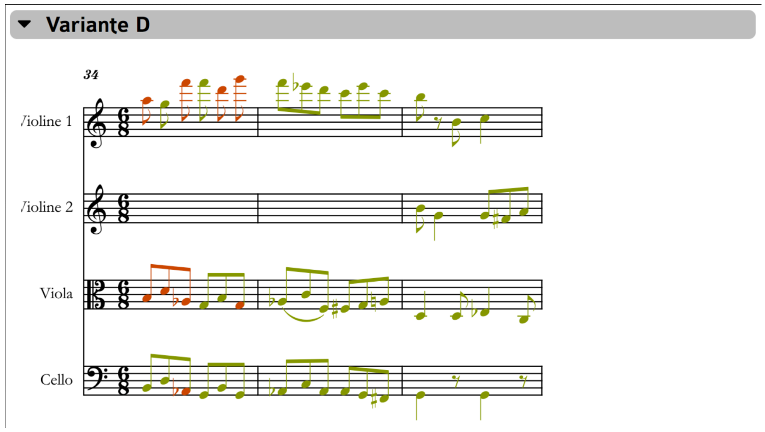


Figure 6: Variant D of the selected passage in Beethoven's op. 59.3, with colors according to when a note was written first. Here, green color indicates a note written in Variant A already, while the notes written in Variant D (after the restitution of Variant A) are highlighted in orange.

In our second prototype, we tried to improve on our transcriptions, in order to illustrate the textual-genetical processes more clearly. After all, Beethoven created this sequence of variants not by writing a clean text on a fresh page for every variant, but by modifying already existing text, and by adding additional writing layers on top. In order to better illustrate this stacking of writing layers and how they result in the textual states of each variant, we added an optional coloring mode to our transcriptions. In this mode, all notes get colored according to the variant in which they were written (see Figure 6). A note that remains unchanged from variant to variant keeps its initial color.

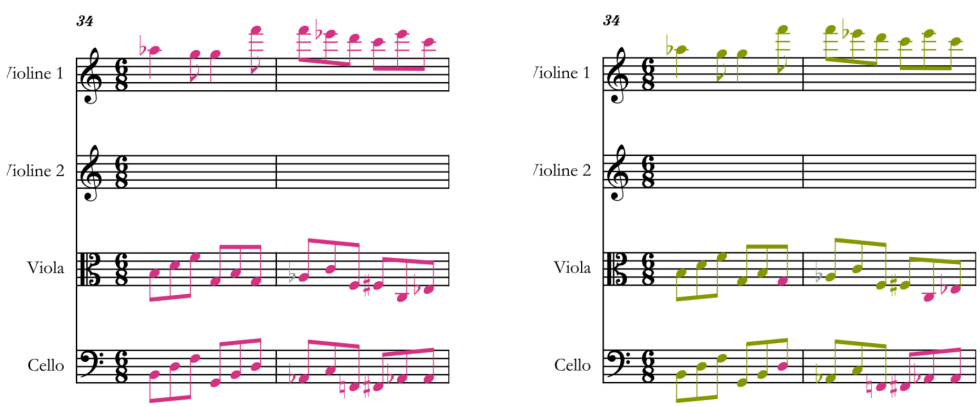


Figure 7: On the left, you see that Variant B has been rewritten completely—all notes are colored in pink. However, most notes have just been copied over, without any conceptual change. These notes are shown in green on the right side.

In order to understand a series of variants from a genetic perspective, and to better see how the musical text developed throughout this process, another coloring mode was implemented. Here, the coloring indicates which notes have changed with regard to their content, independent of their position on the page. That is, when Beethoven writes a new variant in a previously unused workspace, only those notes that have changed in comparison to the previous variant get a new color, while all notes that already existed there preserve their coloring (see Figure 7). In this mode, it becomes quite evident which parts of a variant were "recycled" by Beethoven, no matter if he implemented his modifications as alteration of existing text, by stacking writing layers, or by starting from scratch on a blank page.

Resulting requirements for an MEI encoding

Most of these aspects haven't been explored with MEI so far. While some aspects where easy to model with current MEI, others needed substantial enhancements. In the following, we'd like to introduce the most interesting aspects of our data model.

The relation between encoding and facsimile

MEI encodings typically serve two very different purposes: Describing the graphical appearance of an existing source as a (partially) diplomatic transcript, and enabling a dynamic layout of musical content, which is rendered into the confinements of the current browser window. While the latter is of minor importance to our project in general—we don't adjust our score layout depending on screen size—we still need to clarify the relation between our MEI encodings and the high-res facsimiles we get from the Beethoven-Haus Bonn.


```
<slur xml:id="bd9ad0022-d470-4de0-a05d-bd2b09fbf076"
  staff="3" tstamp="1" tstamp2="0m+3" curvedir="above"/>
```

Figure 8: An encoding of a slur, with both the corresponding original slur and a transcript rendered from the MEI code.

The question is to which degree the encodings should be used to capture information about the script: Unusual shapes, the room used for laying out the notation, the way older variants might be corrected to create newer variants—all this is relevant to understand the genetic processes in Beethoven's manuscripts. At the same time, all that is clearly visible in the facsimiles, and almost impossible to be captured faithfully in a semantically structured format like MEI. Therefore, Beethovens Werkstatt followed a new approach by separating graphical appearance from our interpretation of those signs. This was achieved by encoding all shapes written on the page as `SVG` (Scalable Vector Graphics) `path` elements. SVG is an XML-based format (just like MEI), which is used to describe graphics as resolution-independent vectors. Every path described in SVG may have an `@id`, and can thus be individually addressed. From within an MEI encoding, notes (and other parts of the notation) can then point at the shapes their transcription is based on, using the existing `@facs` attribute. This attribute, which is available on all required elements, is officially meant to point to the `@xml:id` of an `<mei:zone>` element, which provides the bounding box of the feature in question. As its datatype is `anyURIs`, however, it is possible to point any kind of markup, including the exact shape encoded as `<svg:path>`.

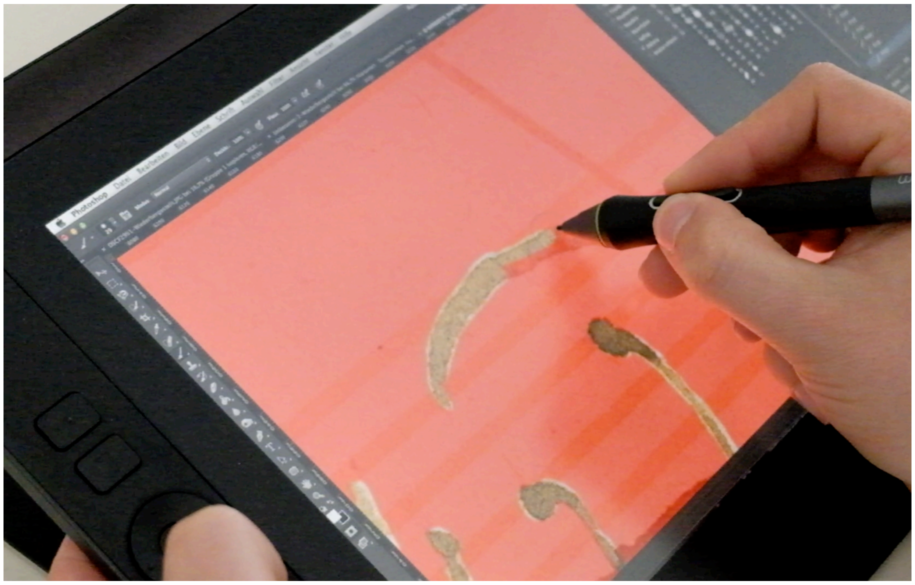


Figure 9: Tracing a slur in order to retrieve SVG shapes.

This significantly reduces the amount of information we need to include in our MEI encodings—only those aspects we consider to be relevant for the musical content have to be encoded in MEI, while all graphical aspects are delegated to the SVG shapes. If a later project wants to go deeper than we do, they can easily go back to our shapes and start from there. However, this complete setup relies heavily on the availability of these SVG shapes. To our knowledge, there is currently no sufficiently reliable automated process to extract these from our scans—we need to separate all layered strokes of ink in order to address specific writing layers, so already this seemingly objective task requires a good amount of interpretation. We retrieve those shapes by tracing them on a professional graphics tablet.[2] The time needed for this tedious task depends on the amount of corrections on a page. On average, it takes between eight and twelve hours to prepare a completely filled page.

However, with this information, it becomes possible to always go back to the shape a specific encoding is based on. In combination with `Verovio`, this becomes very powerful: As Verovio preserves the original `@xml:ids` of the MEI files it renders [2], it is possible to go back from a rendered score to the MEI file, and from there to the shapes it is based on. Obviously, this works in both directions. Therefore, it becomes possible to click on a note in a rendering and highlight the note it is based on in the facsimile, and vice versa. Additionally, all information stored in the MEI file can be used to provide additional information about that note. For instance, if a note is not easily legible because of multiple corrections, a click on it's shape can bring up information about it's pitch, duration and so

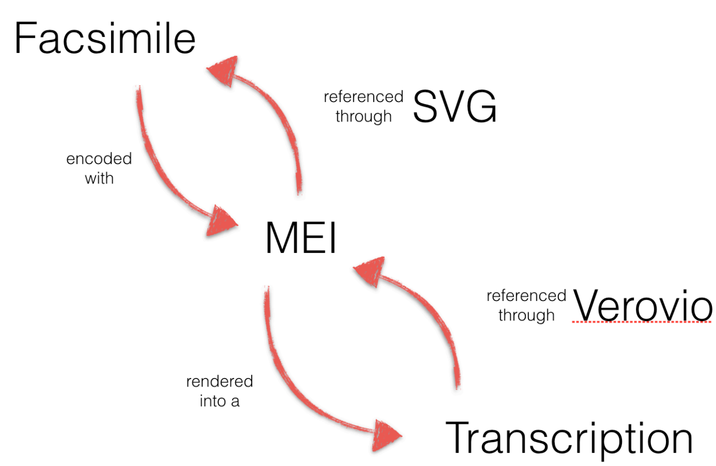


Figure 10: Roundtripping between Facsimile and Transcription through MEI, Verovio and SVG.

2. The tablet we use is by Wacom and allows to use a digitizer pen directly on top of our high-res scan. In this setup, we use Adobe Photoshop in `Quick Mask` mode to paint multiple, non-overlapping selection layers. These layers are then exported to Adobe Illustrator, where some post-processing is needed before the file can be exported to SVG. We then use a simple XSLT to remove all references to those Adobe applications from the files and generate the `@id`-attributes we need.

on, but can also redirect the user either to the full encoding of that note or a transcription of it. With these connections, it becomes much easier to make it clear what an editorial statement refers to—the corresponding feature can be highlighted very easily. So eventually, our combination of MEI with SVG enables a combined use of facsimile and encoding which is much more expressive than any other approach to digital editions of music seen so far.

Tracing genetic processes in MEI

In TEI's model for encoding genetic states, a sequence of `<change>` elements is used to describe a *set of revision campaigns or stages identified during the evolution of the original text* [3]. This means that this model focusses clearly on the modifications undertaken to compose a work, and it matches well with the elements used in the music (or, in TEI's case, texts) that capture this development. Additions, deletions, restitutions and substitutions are all changes to the text, and so aligning them chronologically through the means of a `<change>` element seems quite adequate.

However, when looking at manuscripts, we can usually identify or reconstruct *states*, that is, the appearance of the manuscripts between these changes. At least one state can be identified safely—the current state of the manuscript, after all changes have been implemented. Almost always, all preceding states and changes are a matter of interpretation. However, from a modelling perspective, it is evident that the development of the (musical) text is actually a strict alternation of *changes* and *states*: Each change leads to a new state, after which another change results in the next state and so on. With this in mind, it becomes possible to separately address genetic processes from the products they result in. In our data model, we therefore created a new element called `<genDesc>` (genetical description), which contains an alternation of `<change>`s and `<state>`s,[3] even though projects are not forced to use both elements. In essence, they can decide if they want to describe genetic products and / or processes separately, or within a single entity.

On elements like `<add>` or `<del>`, which trace textual modifications, we've added an attribute `@changeState`, which may point to both `<change>` and `<state>` elements within `<genDesc>`. Combined with the possibility to encode states only, this introduces a certain ambiguity to the model, which is addressed by documentation: In our customization to MEI, we defined that a textual modification pointing to a `<state>` need to be carried out to reach that state. This means that a deletion pointing at a state indicates that the contained, deleted text isn't part of the state anymore, while additions are included. With this clarification, it becomes possible to closely trace the textual development across different modifications, no matter how deep they are nested inside an MEI encoding. Eventually, this model makes it possible to reconstruct genetic states in their chronological order.

Although we can determine this chronological order with sufficient certainty for our first two case studies, this is surely not always possible. Very often, an editor can only make relative statements: *Correction A* must have been carried out before *Deletion B*, the ink layer was written after the pencil etc. In TEI, it is relatively complicated to faithfully reproduce such statements. A `<tei:listChange>` has an `@ordered`-attribute, which indicates whether it's contained `<change>` elements are in chronological order or not. As `<listChange>` may contain not only changes, but also other listChanges, it's possible to mix sections with known order with other sections. However, this requires a relatively complex setup of nested listChanges, which is neither easy to setup, nor to maintain or process. While we kept this possibility for our proposed `<genDesc>`, we also introduced another possibility. MEI, like TEI, offers the `@next` and `@prev` attributes,

3. Besides this, the proposed `<genDesc>` has great similarity with `<tei:listChange>`.

which indicate the immediate successor and predecessor in a sequence. For dates, there are also relative statements like `@notAfter`, which give no precise dates, but only one condition that needs to be matched. We introduced two additional attributes, `@follows` and `@precedes`, with the `anyURIs` datatype. These are available on `<state>`s and `<change>`s, and may point to other states and changes. With these pointers, it becomes very easy to describe sequences as far as they are known to the editor. Without the need for nested genDescs (or listChanges), this should be much easier to follow. While we haven't been able to test this on our current case studies, future examples will surely require this new feature.

Conclusions

MEI is definitely capable of encoding genetic processes, with only very small modifications to the schema. The only major change that was necessary was the addition of `<genDesc>` and its children in order to connect individual modifications spread across the score (and the respective XML hierarchies). The combination of facsimile and encoding through SVG, with which so far unimaginable representation possibilities become possible, required no changes to MEI, but a clever modelling utilizing existing markup options.

Beethovens Werkstatt is still at the beginning of exploring these new options. While we believe that the model we have described in this paper is generally applicable for genetic editions, we are also confident that future case studies will bring up new perspectives and challenges, which may affect this model. We will keep track of these developments in a separate GitHub[4] repository and invite to join our discussions there.

Works cited

[1] Grésillon, Almuth: "Critique génétique. Handschriften als Zeichen ästhetischer Prozesse" in *Ästhetische Erfahrung und Edition* (Beihefte zu editio, 27), ed. by Rainer Falk and Gert Mattenklott, Tübingen 2007: 73-86.

[2] Pugin, Laurent, et al. "Verovio: A Library for Engraving MEI Music Notation into SVG" in *Proceedings of the International Society for Music Information Retrieval*, 2014: 107-12 (https://dblp.org/rec/conf/ismir/2014)

[3] TEI Documentation for the <listChange> element, see www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-listChange.html

4. https://github.com/BeethovensWerkstatt/Data-Model

A Browser-Based MEI Editor

Andrew Horwitz
McGill University
andrew.horwitz@mail.mcgill.ca

Andrew Hankinson
McGill University
andrew.hankinson@mail.mcgill.ca

Ichiro Fujinaga
McGill University
ich@music.mcgill.ca

Abstract

This paper presents a web-based XML editor tailored for use with MEI. This editor provides an interface for synchronizing text edits with changes from a graphical editor. While initially developed to facilitate editing the results of optical music recognition, the editor is built to interface with other graphical editors. Various features such as loading multiple files at the same time, saving to the user's computer, and validation are also built in.

Introduction

This paper presents a web-based MEI XML editor, meix.js. The editor application is built with Javascript, CSS, and HTML, and uses Cloud9's Ace text editor (http://ace.c9.io/), which supports many features of a software code editor including syntax highlighting, find and replace, and code folding. Our research needs included editing MEI that contained a list of pixel coordinates for neumes on an image; we found that directly editing the MEI XML files was unnecessarily difficult for this task, and thus built a text editor that can communicate and share edits with other browser-based applications. We have since expanded meix.js to work with applications including the Diva.js document viewer, the Verovio score engraver, and the xmllint XML validator.

Applications

Our project using Diva.js gathers a list of MEI zone elements that correspond to regions on an image and generates highlighted overlays on the document for each of these zones (Figure 1). meix.js manages the MEI XML files that contain these zones; creating, resizing, or deleting a highlighted overlay automatically updates these files, allowing the user to edit the MEI markup without memorizing the necessary elements and attributes.

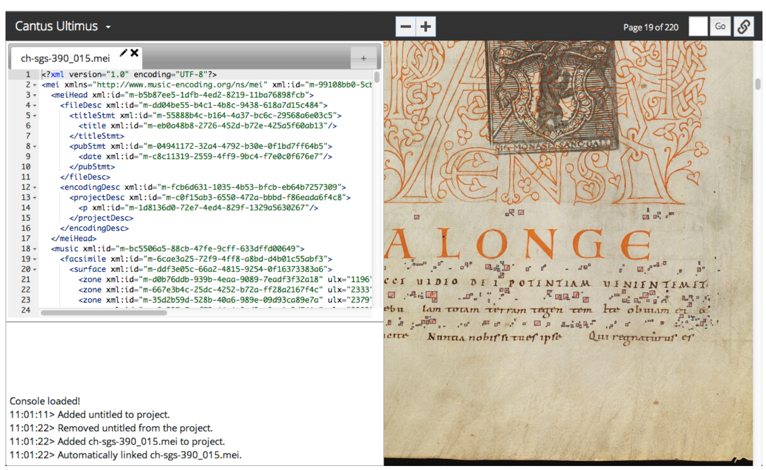


Figure 1: Highlighted overlays on an image and the corresponding MEI XML.

Meix.js can also be used as an XML editor for the JavaScript-based Verovio rendering system. While our application does not yet provide a graphical interface for editing the music, users can edit the MEI markup to change notes, durations, or other musical attributes. This is then rendered in the browser using Verovio with no server-side software needed (Figure 2).

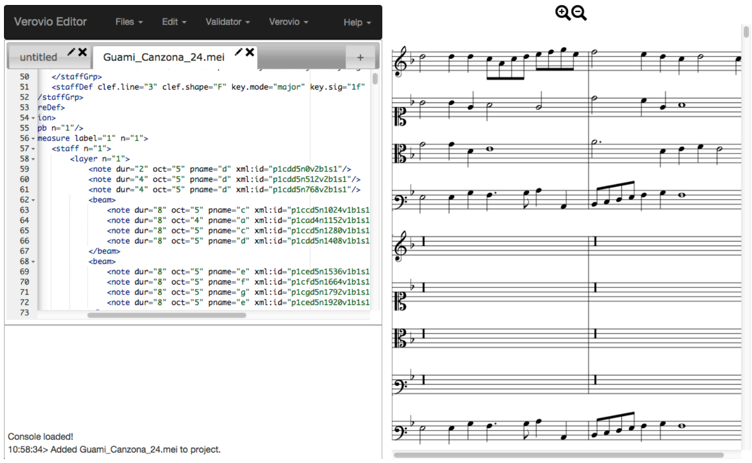


Figure 2: SVG rendering of a score and the corresponding MEI XML.

Xmllint is a program that can be used to validate MEI against a RelaxNG schema. The xmllint functionality in meix.js sends an MEI document to a remote Node.js server to be validated; if the server is not running, the validation service uses a JavaScript port of xmllint running in the browser using the WebWorker API.

Summary

Meix.js is open-source software, and is available at https://github.com/ddmal/meix.js. It may be used as a standalone web application or used to store, save, and edit files for existing web applications. Our list of supported applications is currently limited to our image analysis and score display needs at the moment, but we encourage contributions from the community for additional features.

A Specification for Addressing Encoded Music on the Web

Raffaele Viglianti
Maryland Institute for Technology in the Humanities, University of Maryland
rviglian@umd.edu

Introduction

Enhancing Music Notation Addressability (EMA) is a one-year project that investigates methods for addressing arbitrary portions of encoded music notation on the web. By "addressing" we mean being able to refer to, or cite, a passage of music in order to make a statement about it. This could be considered a virtual equivalent of "circling" some music notation on a printed score. Michael Witmore [1] has argued that text is a "massively addressable object"; that is, given certain abstractions and conventions, it is possible to identify areas of a text such as characters, words, as well as chapters or proper names. Compared to text, music notation is more complicated to represent digitally. Human-computer interaction has since its early days been built around the concept of character and line, which makes dealing with "plain" text a fairly straightforward matter for many basic operations; counting the number of characters in a given plain text document is trivial in any digital environment. Music notation, however, requires substantial computational modeling even for the simplest musical text before any further operation is possible. There are, indeed, many ways of representing a single note; some aspects are common to all representation systems, such as information about pitch and duration, but some systems will prioritize certain aspects over others.

The technical specification created by the EMA project are described below. The specification aims at defining a scheme for addressing a selection of music notation regardless of its representation. The expression is based on simple units that are commonly represented by music notation systems for common Western music notation, such as measure, staff, and beat. The expression is formulated as a URL, which makes it possible to target resources on the web.

A brief overview of the specification

Consider the following example (from *The Lost Voices* project),[1] and the notation highlighted in red:

1. http://digitalduchemin.org/piece/DC0519.

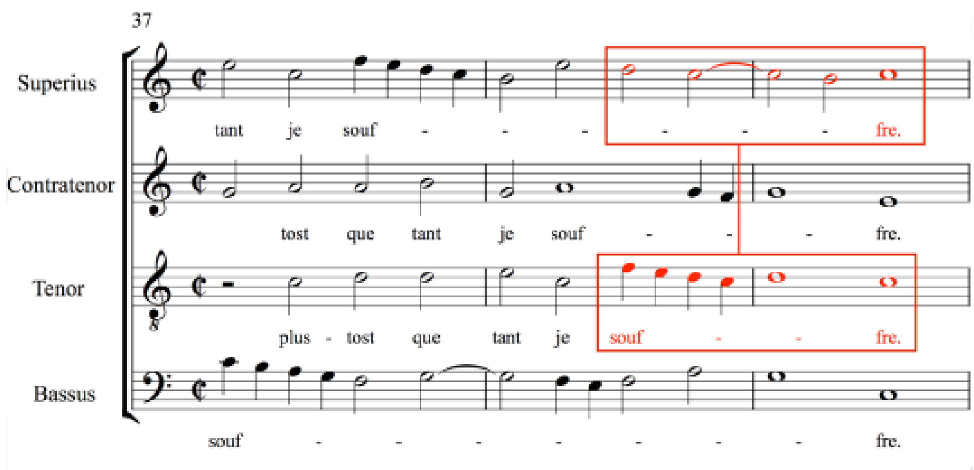


Figure 1: DC0519 L'huillier, *Si je te voy*.

The highlighted notation occurs between measure 38 and 39, on the first and third staves (labelled *Superius* and *Tenor* — this is a Renaissance choral piece). Measure 38, however, is not considered in full, but only starting from the third beat.

This selection can be expressed according to a URI syntax:

```
/{identifier}/{measures}/{staves}/{beats}/
```

```
/dc0519.mei/38-39/1,3/3-3
```

The measure is expressed as a range (38-39), staves can be selected through a range or separately with a comma (1,3), and the beats are always relative to their measure, so 3-3 means the third beat of the starting measure to the 3 beat of the ending measure. The EMA project is working on a specification for this URI syntax and an API for its implementation.[2]

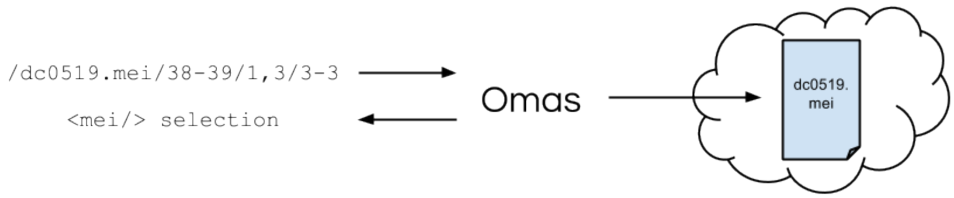
In this specification the beat is the primary driver of the selection: only selections that are contiguous in beat can be expressed with this system. Music notation, however, often breaks rules in favor of flexibility. Cadenzas, for example, are ornamental passages of an improvisational nature that can be written out with notation that disregards a measure's beat. The project is planning on extending the specification to be able to address music notation without relying on beat.

Evaluation

In order to evaluate the specification, EMA is producing an implementation of the API as a web service. While the specification can attempt to be absolute from a specific representation, the implementation must know how to operate on specific formats. The web service that we coding operates on the The Music Encoding

2. A first draft of the API is available on GitHub at https://github.com/umd-mith/ema/blob/master/docs/api.md. Future releases will be available at the same address.

Initiative format and is called Open MEI Addressability Service (Omas). Omas interprets a conformant URI, retrieves the specified MEI resource, applies the selection and returns it. An additional parameter on the URI can be used to determine how "complete" the retrieved selection should be (whether it should, for example, include time and key signatures, etc.).



Similarly to an image server, Omas assumes that the information specified by the URL can be retrieved in the target MEI file. If requested, the web service can return metadata information about an MEI file, such as number of measures, staves, beats and their changes throughout the document. This can be used to facilitate the creation of URL requests able to return the selection required.

Finally, EMA partnered with the *Lost Voices* project to model a number of micro-analyses addressing music notation from their existing collection of MEI documents. In a second phase of the project, these analyses will be re-modeled as Linked Open Data nanopublications.[3] Each EMA nanopublication will address an arbitrary portion of music notation using the URL specification described here. Omas will work as a web service to connect the nanopublications with the collection of MEI files in Lost Voices.

Works cited

[1] Witmore, Michael. "Text: A Massively Addressable Object". Blog post: *Wine Dark Sea*, December 2010 (http://winedarksea.org/?p=926).

3. www.nanopub.org/2013/WD-guidelines-20131215

Contributions from MEC 2016

On Intermediate Formats

Reiner Krämer
University of Northern Colorado
reiner.kramer@unco.edu

Abstract

Computational Music Analysis Platforms (or CMAPs for short) in general parse music data and transcode the data to an internal format in order to process the acquired data. We call the internal format an intermediate format, because it acts within a software architecture as an intermediary between input and output of a given CMAP's workflow. The paper examines three different types of intermediate formats, one historical, and two currently in use, as they occur in CMAPs, and posits how these intermediate formats influence the analytical process and thought.

Introduction

Most discourses deal with symbolically encoded music data consumed by CMAPs. We can talk about all aspects of how to properly structure and optimize this type of data, but, however well-encoded the data seems to be, in the end it will most likely be transcoded to an intermediate format within a software package. This is where we surrender our well-behaved data to the "machine." The "machine"—or CMAP's programmer, programming language, and music theory/analysis knowledge—influences how the intermediate format handles our symbolically encoded music, and thereby its music analytical output. The intermediate format is what a music analyst sees and works with while examining a piece of music, or a corpus of music. Even though intermediate formats may be as ephemeral as their host software packages, these formats are often the only remaining traceable artifacts of a particular démodé software package.

This paper (1) shows how CMAPs are structured, (2) examines three types of CMAPs: a historical platform, from a media archeological perspective, and two platforms in current use, (3) and points to usability of Intermediate Formats.

CMAPs ins and outs

Figure 1 shows the three main tasks of a CMAP. The first task is to programmatically import music data (top of the chart, or level 1). At level 1, interchange formats are brought into a software package. The formats can consist of symbolically encoded music data, such as .mei, MusicXML, .json, .krn, .md2, .midi, etc. The formats can also include preferably uncompressed audio or live processing formats. Likewise, graphically high-resolution pixel based images, or vector graphics can also be brought in, as well as text-based formats, such as .csv, .xml, .json, .osc, or any other possible future formats.

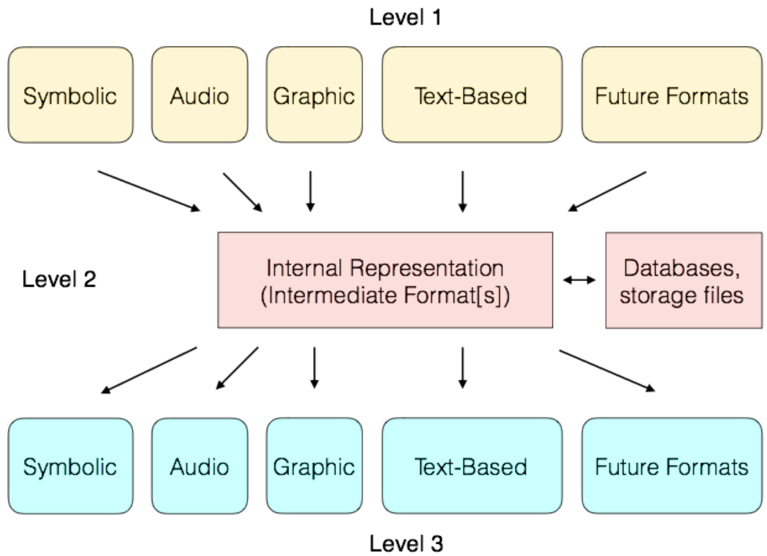


Figure 1: CMAPs I/Os.

At level 2, a CMAP translates data acquired from the interchange formats to a native data processing format that we shall call an **intermediate format**. In many cases CMAPs possess some sort of data storage capability for the intermediate format either as plain text based files, or databases.

At level 3, CMAPs can display analysis results in various formats, such as symbolic, audio, graphic, text-based, or any other future formats, which are usually interchange formats (as the formats consumed at level 1).

David Huron created the much beloved Humdrum[1] CMAP in the 1980s. The humdrum.org website states "Humdrum is a set of command-line tools that facilitate musical analysis, as well as a generalized syntax for representing sequential streams of data.

Because it is a set of command-line tools, it's program-language agnostic. Many have employed Humdrum tools in larger scripts that use PERL, Ruby, Python, Bash, LISP, and C++." Most people will be familiar with humdrum and its intermediate/exchange format named `kern`, carrying the ".krn" extension.

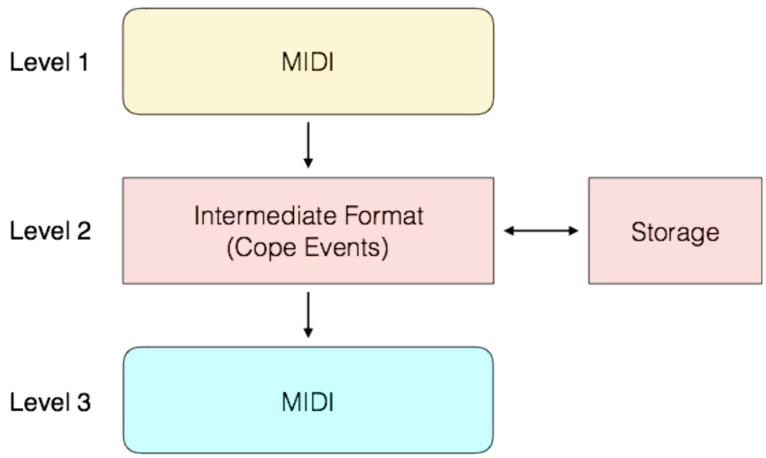


Figure 2: David Cope's CMAPs.

We will briefly return to the kern format at a later point. For now, we will turn our discussion toward David Cope's work, who is a well-known American composer, composition pedagogue, music theorist, and musicologist.

1. http://humdrum.org

David Cope's CMAPs

During the 1980s David Cope began designing computer-aided compositional tools at the University of California, Santa Cruz. At their core, the tools can be divided into two categories: (1) analysis, and (2) re-composition from analysis. Cope covers a large range of computational music analysis problems with his programs, including pattern matching, harmonic/melodic successions, forms, and style. Cope's "Trilogy" (*Computers and Musical Style*, *Experiments in Musical Intelligence*, *The Algorithmic Composer*) covers most aspects of his work on recombinant music (a pastiche-like technique similar to 18th-century musical dice games). The musical output usually is titled "In the style of …," or "After …". Cope summarizes his in *Virtual Music: Computer Synthesis of Musical Style*, which also features critical responses. In Cope's *Computer Models of Musical Creativity*, he uses association networks for music composition in hope to stylistically "synthesize" new musical styles from his previous recombinant work. However, let's discuss Cope's working process for music analysis.

As Figure 2 shows, at level 1 (as was the case in Figure 1) Cope first parses the midi interchange format. At level 2, the parsed music data is placed into an intermediate format, also called "Cope Events." All computational operations are conducted on this intermediate format. "Cope Events" can also be stored as text files for later consumption with his CMAPs, meaning that MIDI data does not have to be re-parsed ad infinitum, or showing Cope's commitment to the DRY (don't repeat yourself) software development principle. At level 3, after music data has been processed either as analysis or perhaps a new composition the outcome is translated back into the MIDI interchange format. A Cope event looks like the following example grouped by parentheses:

```
(0 38 147 2 90)
```

Figure 3: A Cope event.

A Cope event breaks down breaks down in the ensuing way: (1) the first number is the start time in milliseconds (here it is the first event of a piece of music, thus the event starts at 0); (2) the second number is a MIDI pitch value (between 0 and 127) (here 38 == D1); (3) the third number is the duration in milliseconds (it is a 16th note durational value); (4) the fourth number is a channel number (between 0 and 127), which can be used to assign an instrumental sound, or somewhat of a "timbral" specification (channel 2, perhaps a piano); and the fifth number describes a dynamic level between 0 and 127 (90 would be similar to a *forte* intensity level designation) [2]. Now, let's consider the first few measures of Nicolas Champion's motet *De profundis* (Figure 4).

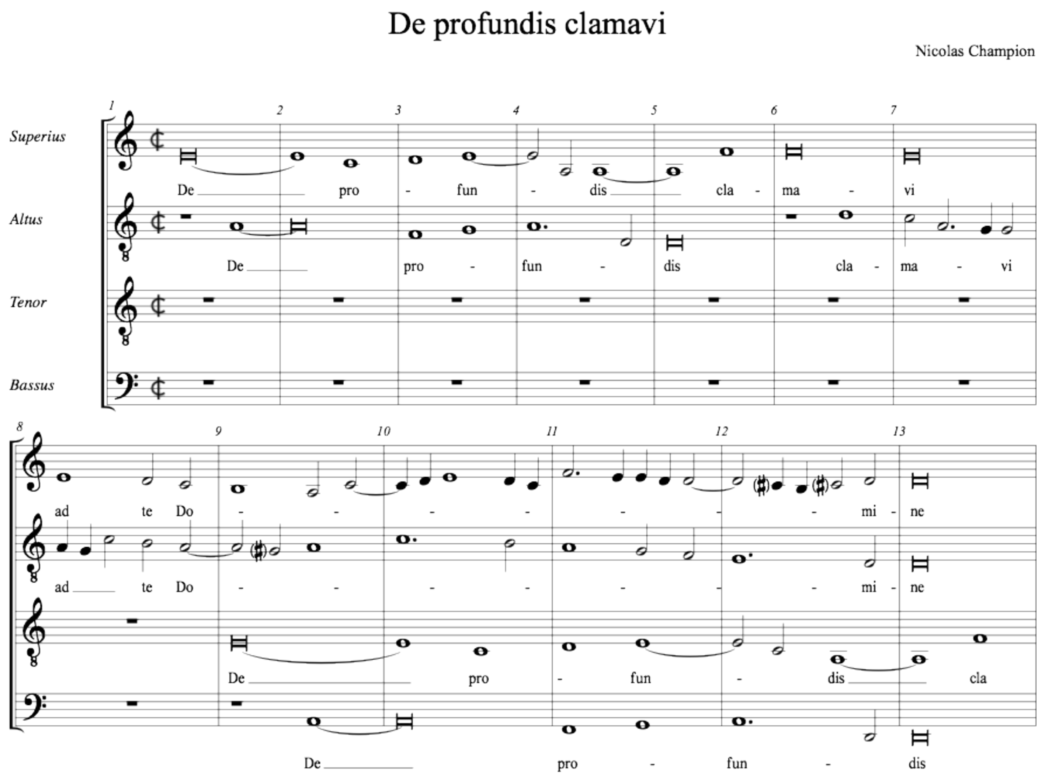



Figure 4: Nicolas Champion's *De profundis*.

Figure 4 looks the following way encoded in Cope's intermediate format:

```
((0 64 5999 1 80) (2000 57 5999 2 80) (6000 60 1999 1 80) (8000 62 1999 1 80) (8000 53 1999 2 80) (10000
  64 2999 1 80) (10000 55 1999 2 80) (12000 57 2999 2 80) (13000 57 999 1 80) (14000 57 3999 1 80)
  (15000 50 999 2 80) (16000 50 3999 2 80) (18000 65 1999 1 80) (20000 65 3999 1 80) (22000 62 1999 2
  80) (24000 64 3999 1 80) (24000 60 999 2 80) (25000 57 14 99 2 80) (26500 55 499 2 80) (27000 55 999 2
  80) (28000 64 1999 1 80) (28000 57 499 2 80) (28500 55 499 2 80) (29000 60 99 9 2 80) (30000 62 999 1
  80) (30000 59 999 2 80) (31000 60 999 1 80) (31000 57 1999 2 80) (32000 59 1999 1 80) (32000 52 59 99
  3 80) (33000 56 999 2 80) (34000 57 999 1 80) (34000 57 1999 2 80) (34000 45 5999 4 80) (35000 60 1499
  1 80) (36000 60 2999 2 80) (36500 62 499 1 80) (37000 64 1999 1 80) (38000 48 1999 3 80) (39000 62 499
  1 80) (39000 59 999 2 80) (39500 60 499 1 80) (40000 65 1499 1 80) (40000 57 1999 2 80) (40000 50 1999
  3 80) (40000 41 1999 4 80) (41500 64 499 1 80) (42000 64 499 1 80) (42000 55 999 2 80) (42000 52 2999
  3 80) (42000 43 1999 4 80) (42500 62 499 1 80) (43000 62 1999 1 80) (43000 53 999 2 80) (44000 52 2999
  2 80) (44000 45 2999 4 80) (45000 61 499 1 80) (45000 48 999 3 80) (45500 59 499 1 80) (46000 61 999 1
  80) (46000 45 3999 3 80) (47000 62 999 1 80) (47000 50 999 2 80) (47000 38 999 4 80) (48000 62 3999 1
  80) (48000 50 3999 2 80) (48000 38 3999 4 80) (50000 53 1999 3 80))
```

Figure 5: *De profundis* as Cope events.

All events that were previously displayed in the score in all parts are here placed into one large list of lists (the latter being the event list). The musical data has been truncated and now contains only pitches in MIDI values, disregarding any gradation of sharps or flats, i.e. 61 is E flat, but so is D sharp (current iterations of the MIDI standard, i.e. expanded/extended MIDI, have solved this problem). All parts are lumped together into one stream of data, however they are separated by channel numbers so that channel one is the Superius, channel 2 is the Altus, and so forth. Another piece of musical data that has been truncated is the text, although technically meta data.

All of Cope's databases use this format (it should be noted that Cope's databases do include several additional bits of meta data in relationship to the music they represent). The intermediate format (Cope events) is machine parse-able, since lists remain being data types in many other programming languages. Further processing a list of events as displayed in Figure 5 invites users/programmers to think through a score in a recursive manner, utilizing the "functional programming" paradigm employed by Common Lisp.

Music21

Music21[2] is a CMAP developed by Mike Cuthbert and Christopher Ariza at MIT, and started appearing during the late 2000s [3].

The CMAP can parse many symbolic music file formats, or interchange formats, like ABC, Braille, Capella, Humdrum, Lilypond, MEI, MIDI, MuseData, MusicXML, Noteworthy, NoteworthyBinary, RomanText, Text, TextLine, TinyNotation, and more. Due to music21 being open source, other interchange formats can be added as they emerge. Music21's intermediate format is the music21 stream. The music21 streams can be stored as its own entity, not unlike Cope events.

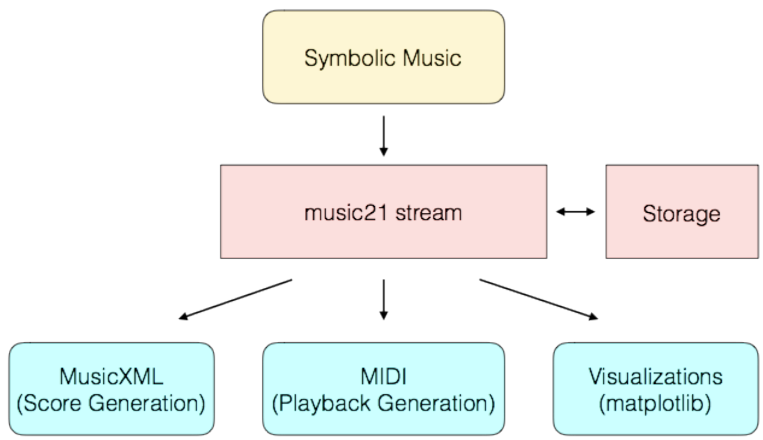


Figure 6: Music21.

Music21 can output numerous intermediate formats, such as for MusicXML score generation, MIDI for playback, and data visualizations in .pdf or .svg formats.

Figure 6 summarizes music21's workflow process. In many regards music21 uses a kitchen sink approach. Let's refresh out memory of *De profundis* (Figure 4). Figure 7 below shows its music21 stream.

2. http://web.mit.edu/music21

```
{0.0} <music21.text.TextBox "De profund…">
{0.0} <music21.text.TextBox "Nicolas Ch...">
{0.0} <music21.metadata.Metadata object at 0x10ca6dd50>
{0.0} <music21.stream.Part Superius>
    {0.0} <music21.instrument.Instrument P1: Superius: Instrument2>
    {0.0} <music21.stream.Measure 1 offset=0.0>
        {0.0} <music21.layout.SystemLayout>
        {0.0} <music21.clef.TrebleClef>
        {0.0} <music21.key.KeySignature of no sharps or flats, mode minor>
        {0.0} <music21.note.Note E>
    {8.0} <music21.stream.Measure 2 offset=8.0>
        {0.0} <music21.note.Note E>
        {4.0} <music21.note.Note C>
    {16.0} <music21.stream.Measure 3 offset=16.0>
        {0.0} <music21.note.Note D>
        {4.0} <music21.note.Note E>
    {24.0} <music21.stream.Measure 4 offset=24.0>
        {0.0} <music21.note.Note E>
        {2.0} <music21.note.Note A>
        {4.0} <music21.note.Note A>
```

Figure 7: *De profundis* music21 stream (excerpt).

During an initial observation we notice that the music21 stream itself resembles XML structure. One could say that music21's intermediate format is XML-centric. The other main feature that jumps out at a user/programmer is that everything is an object, rather than a list. Object-oriented thought is what drives music21. Features of the music21 stream that are not visible are lyrics, but also how pitches are stored. For example, a pitch object can contain multiple attributes, like: name, nameWithOctave, pitchClass, MIDI value, frequency, German, Italian, and Spanish names.

Music21 solves many issues in its intermediate format, in comparison to the Cope events, and is fully extensible. However, it's display of musical data to the casual music-theorist/musicologist programmer seems still a bit cryptic (although music21's show method, as its default, displays an XML score visualized by invoking score editors like Sibelius, Finale, Dorico, or MuseScore, if a user has them installed).

VIS-Framework

As was the case with previously examined CMAPs, the VIS-Framework[3] (initially developed at McGill University by Christopher Antila and Jamie Klassen, and under current development by the analysis axis of the SIMSSA team), parses symbolically encoded music by way of using music21's parsing mechanisms. Therefore, all music21 stream objects become available to the VIS-Framework [1].

Music21 streams are then indexed with pandas, a Python data analysis toolkit[4] that is well suited with the following types of data: (1) tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet, (2) ordered and unordered (not necessarily fixed-frequency) time series data, (3) arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels, and (4) any other form of observational/statistical data sets. Data actually need not be labeled at all to be placed into a pandas data structure.

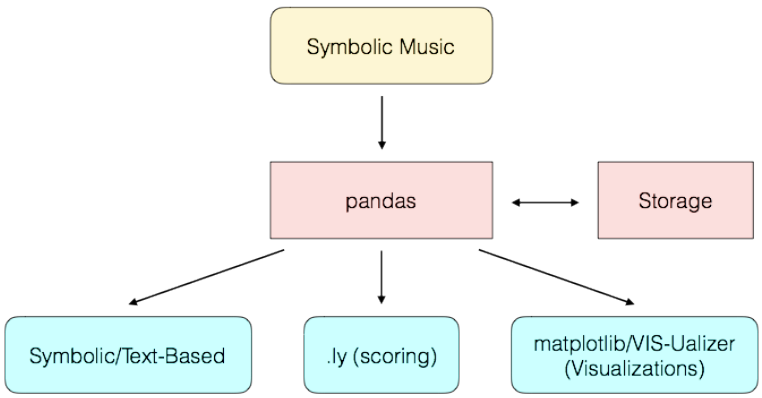


Figure 8: VIS-Framework.

There are two main datatypes within pandas: (1) a series (ideal for processing one-dimensional arrays: as in a single line voice/instrument in a score), and (2) a DataFrame (ideal for processing two-dimensional arrays: as in multiple voices/instruments in a score). Pandas' DataFrame can also be found in the R programming language[5], and in Apache Spark[6], and programming libraries exist that can bridge any combination of DataFrames for the different platforms. Pandas has mechanisms that enable it to store its Series, or DataFrames in a multitude of popular data storage formats.

The VIS-Framework can output analytical data as text-based files (similar to the storage formats previously mentioned). The CMAP can also generate score output in LilyPond, in addition to data visualizations by way of the popular matplotlib[7] Python visualization tool. Figure 8 summarizes how the VIS-Framework dataflow process functions.

3. https://github.com/ELVIS-Project/vis-framework
4. http://pandas.pydata.org
5. https://www.r-project.org
6. https://spark.apache.org
7. https://matplotlib.org/

Let's re-familiarize ourselves with our music example *De profundis* (Figure 4). Once the VIS-Framework has processed the data and stored it in its intermediate DataFrame (or Series) format, a user can observe the data (Figure 9), reminiscent of Humdrum's kern format, where different voices/instruments are represented as spines.

```
Indexer noterest.NoteRestIndexer
  Parts                         0     1     2     3
  0                            E4  Rest  Rest  Rest
  4                             -    A3     -     -
  8                             -     -  Rest  Rest
  12                           C4     -     -     -
  16                           D4    F3  Rest  Rest
  20                           E4    G3     -     -
  24                            -    A3  Rest  Rest
  26                           A3     -     -     -
```

Figure 9: *De profundis* in VIS' pandas DataFrame (excerpt).

However, a user can also transpose the DataFrame. Now the intermediate format appears more native to a music analyst, since it closely resembles an actual musical score, in regards to a right to left timeline (Figure 10).

```
                                0    4     8   12    16   20    24   26
  Indexer                 Parts
  noterest.NoteRestIndexer 0        E4    -     -  C4    D4  E4     -  A3
                           1      Rest  A3     -    -    F3  G3    A3   -
                           2      Rest   -  Rest    -  Rest   -  Rest   -
                           3      Rest   -  Rest    -  Rest   -  Rest   -
```

Figure 10: *De profundis* in VIS' pandas DataFrame, transposed (excerpt).

Again, as was the case with music21, using the VIS-Framework invites the user into object-oriented thought.

Conclusion

We have examined the structure of CMAPs. We have surveyed three intermediate formats through three different CMAPs, namely Cope Events, music21 streams, and pandas Series/DataFrames. We have hinted at how thinking through a different programming languages and platforms can influence the analytical process and thought. Since a music analyst will be working mostly with an intermediate format during a music analytical session, we have also pointed out that the readability of an intermediate format is of importance to the design of a CMAP.

Works cited

[1] Antila, Christopher and Julie Cumming. "The VIS Framework: Analyzing Counterpoint in Large Datasets." in *Proceedings of the International Society for Music Information Retrieval*, 2014: 71-76 (https://dblp.org/rec/conf/ismir/2014)

[2] Cope, David. *Virtual Music: Computer Synthesis of Musical Style*. Cambridge, MA: MIT Press, 2001, 141-143.

[3] Cuthbert, Michael Scott and Christopher Ariza. "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data" in *Proceedings of the International Society for Music Information Retrieval*, 2010, 637-42 (https://dblp.org/rec/conf/ismir/2010).

[4] Kester, Walt. *Data Conversion Handbook*. Amsterdam: Elsevier, 2005.

[5] Li, Tao. *Music Data Mining*. Boca Raton: CRC Press, 2012.

[6] Nartovich, Aleksandr. *WebSphere Development Studio Client for ISeries V5.0*. Rochester, MN: IBM, 2004.

[7] Selfridge-Field, Eleanor. *Beyond MIDI: The Handbook of Musical Codes*. Cambridge, MA: MIT Press, 1997.

[8] Steyn, Jacques. *Structuring Music Through Markup Language Designs and Architectures*. Hershey, Pa.: IGI Global, 2013.

The Kassel-Wolfenbüttel Tablature System: A Convergence of Lute Tablature and Mensural Notation

Rebecca A. Shaw
Western University
shawbecky1@gmail.com

Abstract

The *Wolfenbüttel Lute Tablature* (c. 1460) consists of two fragmented folios with intabulations for five secular songs and is the only extant source of a notation system described theoretically in *Kassel Lautenkragen*. The notation, named the Kassel-Wolfenbüttel Tablature System (KWTS) by Marc Lewon, combines a quasi-mensural notation, which closely resembles the upper voice of organ tablature, and an alphabetic notation, parallel to organ tablature. Because of its connections to both tablature and mensural notations, in order to encode KWTS, one must combine the encoding systems for both tablature and mensural notation. In this paper, I examine some of the problems inherent in encoding KWTS; suggest possible solutions; and define a musicological application for the project. As the earliest lute-specific notation, KWTS contains valuable information on the changing lute performance practices of the mid-fifteenth century, despite the sources' brevity. Developing an encoding system for this notation will enable further exploration of fifteenth-century lute repertoire, including that prior to lute-specific notation.

Introduction

In 2011, Martin Staehelin introduced the earliest extant source of lute-specific notation to the musicological community: the *Wolfenbüttel Lute Tablature* (c. 1460) [10]. Consisting of two folios with fragmented intabulations of five secular songs, it is the only surviving source of the notation system described theoretically in *Kassel Lautenkragen*, a pedagogical drawing of a lute neck with definitions for various notational symbols, largely drawn from Old German Organ Tablature [12][1]. This notation system has subsequently been named the Kassel-Wolfenbüttel Tablature System (KWTS) by Marc Lewon. [12, p.172] [8] KWTS itself consists of two styles of notation. The first is closely related to mensural notation and the upper voice of organ tablature, with the addition of some signs specific to the lute, including its clefs, chords, and a sign for an upward stroke of the plectrum. The second style of notation in KWTS is an alphabetic notation parallel to that used in organ tablature. Because of these connections to organ tablature, it is thought that the *Kassel Lautenkragen* served as a guide to mid-fifteenth-century lutenists who played from organ tablatures.[2] [12, p.189] In this scenario, the descant, or higher, lute, would play the upper voice of a given piece, whilst the tenorist, on a lower lute,

1. Marc Lewon provides a thorough analysis of this source, including high-quality scans of the fragments on his blog [8].
2. The *Kassel Lautenkragen* was glued to the front cover of a codex, although it was originally created as an independent manuscript. Because of the notational concordances between this theoretical source and the more practical Wolfenbüttel Lute Tablature, Marc Lewon named the notation system the "Kassel-Wolfenbüttel Tablature System", nomenclature that I have adopted in this essay.

would play the tenor and contratenor lines. [12, p.189] To differentiate between these two subsets of KWTS, I will refer to the first as KWTS-High and the second as KWTS-Low. The *Wolfenbüttel Lute Tablature* is the only surviving source that employs this notation for lute-specific repertoire, and it makes use of KWTS-High only.

Because of its connections to both tablature and mensural notations, in order to encode KWTS, one must combine the encoding systems for both tablature and mensural notation. [2] [7] In this paper I examine some of the problems inherent in encoding KWTS; suggest possible solutions; and define a musicological application for this project. As the earliest source of lute-specific notation, the *Wolfenbüttel Lute Tablature* and its theoretical counterpart, the *Kassel Lautenkragen*, contain valuable information on the changing lute performance practices of the mid-fifteenth century, despite the sources' brevity. Developing an encoding system for this notation will enable further exploration of fifteenth-century lute repertoire, including that prior to lute-specific notation.

Comparison of notations

Before considering the specifics of encoding KWTS, one must first understand its similarities and dissimilarities to mensural notation, and to later lute tablatures, for which Tim Crawford's `TabCode` was devised. [7] While lute tablatures generate pitch at the convergence of a pair of fret and string coordinates, instructing the performer where and when to put her finger on the fingerboard, this is only partially true for KWTS-Low, as I will explain later, and has little bearing on KWTS-High. Written on an eight-line staff system, with each line corresponding to a diatonic pitch, KWTS-High has more in common with mensural notation; however, the traditional clefs of mensural notation are replaced by five alphabetic signs specific to the lute. The two lowest clefs correspond to the two lowest courses of the lute, and the three upper clefs are spread over the top three courses, defining a two-octave range. Whilst KWTS-High does not have the same degree of prescription as later lute tablature systems, some of its notational particulars are specific to the lute, and cannot be fully encoded using MEI's mensural notation guidelines; it must necessarily draw on some aspects of TabCode as well.

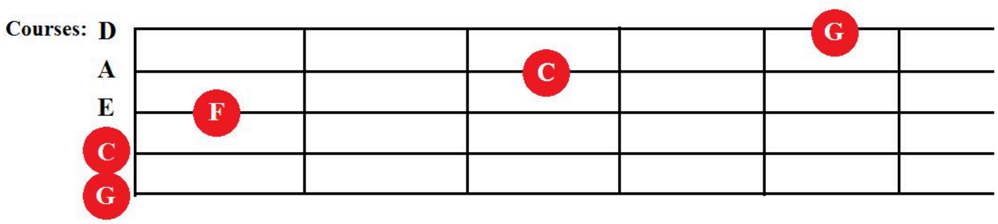


Figure 1: Placement of clefs in the *Wolfenbüttel Lute Tablature* in relation to the frets of a five-string Renaissance Lute.

Encoding KWTS

If we use the *Kassel Lautenkragen* as a glossary of symbols, it is possible to assign most of these to a pre-existing element and attribute (see Figure 2). Because KWTS simultaneously serves as a lute-specific notation system in its own right—as evidenced by the *Wolfenbüttel Lute Tablature*—and as a translation guide for organ tablatures, I have provided equivalent signs to the *Kassel Lautenkragen* from both the *Wolfenbüttel Lute Tablature* and the *Buxheimer Orgelbuch*, which is from the same timeframe and with which the *Wolfenbüttel Lute Tablature* shares some of its repertoire. Starting with KWTS-High, the only signs for which new descriptors are required are that for the "upward stroke pluck," which can be housed under the `@articulation` attribute and the "Cfaut" clef, which is specific to this notation system. The clef signs of KWTS-High can be encoded using the `clefGrp` element (see Figure 3). The only change, here, is that, in addition to the `@clef.line` attribute, I have added a `@clef.space` attribute to account for the Cfaut clef, which is placed in the second space from the bottom of the staff. I have also included the `@trans.diat` (transposition) attribute as the sounding pitch of the lute for KWTS-High would have been a fourth or fifth higher than notated. [8] To describe each of the clefs, I used their typed equivalent, all in lowercase, with a single letter for the lower octave and a double letter for the upper octave.

Sign from "Kassel Lautenkragen"	Appearance in *Buxheimer Orgelbuch*	Equivalent Sign in *Wolfenbüttel Lute Tablature*	Meaning	Encoding Element/ Attribute
		x	*suspira* (rest equivalent to a semiminim)	<rest dur="seminimina">
		or	*semitongis* (accidentals: # if attached to anything other than E or B)	<accid>
			cardinalia (fermata signs; possibly referring to the length of a breve (ternary); breve (binary); and semibreve, respectively)	<fermata> and @dur.ges
	x	x	*signa sursum traxionis* (upward stroke pluck) **specific to lute	@articulation
			reincepciones (repeat signs)	<section> and <expansion> (as in CMN)
		x	*mordante* (ornament)	<mordent>
	x		*concordancie* (simultaneous plucking of notes; not arpeggiated)	<chord>
			Longa	<note>, @dur
		x	*Breuis* [semibreve]	<note>, @dur
			Minim	<note>, @dur
			Semiminima	<note>, @dur
		x	*Tripla*	<note>, @dur
			Fusa	<note>, @dur
		x	Rest, Perfect Breve	<rest>, @dur
		x	Rest, Longa (6 semibreves)	<rest>, @dur
		x	Rest, semibreve	<rest>, @dur
		x	Rest, minima	<rest>, @dur
			Claues (clefs): *ggsolreut* (gg) *ccsolfaut* (cc) *faut* (f) *Cfaut* (c) *Gamaut* (g) **Cfaut* is specific to this tablature. *Buxheimer Orgelbuch* provides the C clef with markings for G and D	<clefGrp>, @clef.line, @clef.shape, @trans.diat

Figure 2: Symbols from *Kassel Lautenkragen*, their equivalents in the *Buxheimer Orgelbuch* and the *Wolfenbüttel Lute Tablature*, and the means by which they are encoded.

```
<staff n="1">
    <app>
        <rdg>
            <staffDef n="1" lines="8" trans.diat="4" trans.semi="5"></staffDef>
            <staffDef n="1" lines="8" trans.diat="5" trans.semi="7"></staffDef>
        </rdg>
    </app>
    <staffDef n="1" lines="8">
        <clefGrp>
            <clef n="1" line="1" shape="g"></clef>
            <clef n="2" space="2" shape="c"></clef>
            <clef n="3" line="4" shape="f"></clef>
            <clef n="4" line="6" shape="cc"></clef>
            <clef n="5" line="8" shape="gg"></clef>
        </clefGrp>
    </staffDef>
</staff>
```

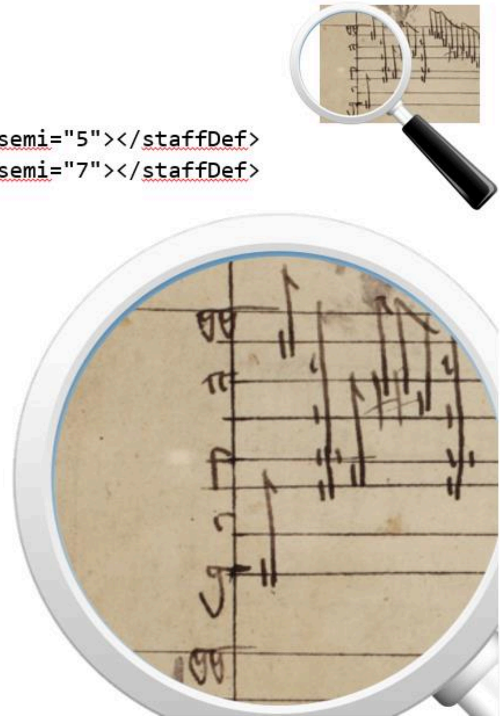


Figure 3: Staff Definition `staffDef` and Clef Groups `ClefGrp` for the *KWolfenbüttel Lute Tablature*.

The encoding system for the notes and rhythms of the *Wolfenbüttel Lute Tablature* can be similarly borrowed from that devised for mensural notation. For instance, the first few beats of the third fragment of the *Wolfenbüttel Lute Tablature* ("Gruß senen Ich im hertzen traghe") can be encoded as you see in Example 4. This portion requires no changes or additions to the existing encoding vocabulary. However, because this music is particular to the lute, we cannot stop here; instead, we must turn to some of the conventions of TabCode. As Richard Lewis has described, encoding lute music necessarily consists of two components: (1) that for the tablature and (2) that for the instrument and its respective tuning. [7] The same is true for KWTS-High, although, in this case, the tablature module is dependent upon the instrument module to transcribe the notation into other lute tablatures, whilst other lute tablatures are dependent on the instrument module for transcription into common music notation (see Figure 5). For the instrument module of KWTS, I have borrowed that used for encoding later lute tablatures in TabCode (see Figure 6). [7]

```
<staff>
    <layer>
        <beam>
            <note dur="fusa" oct="5" pname="c"></note>
            <note dur="fusa" oct="4" pname="b"></note>
            <note dur="fusa" oct="4" pname="a"></note>
            <note breaksec="1" dur="fusa" oct="4" pname="b"></note>
            <note dur="fusa" oct="5" pname="c"></note>
            <note dur="fusa" oct="4" pname="b"></note>
            <note dur="fusa" oct="4" pname="a"></note>
            <note dur="fusa" oct="4" pname="b"></note></beam>
        <chord dur="minima">
            <note oct="5" pname="c"></note>
            <note oct="4" pname="g"></note>
            <note oct="4" pname="c"></note></chord>
        <note dur="semiminima" oct="5" pname="c"></note>
        <beam>
            <note dur="fusa" oct="5" pname="d"></note>
            <note dur="fusa" oct="5" pname="c"></note>
            <note dur="fusa" oct="4" pname="b"></note>
            <note dur="fusa" oct="4" pname="a"></note></beam>
        <chord dur="semiminim">
            <note oct="4" pname="g"></note>
            <note oct="4" pname="e" accid="f"></note>
            <note oct="4" pname="c"></note></chord></layer></staff>
```

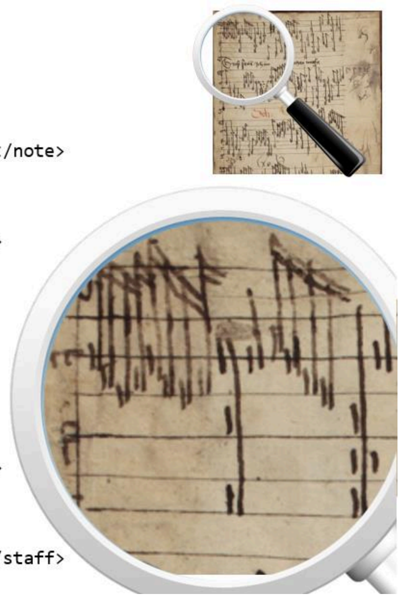


Figure 4: Encoding system for “Gruß senen Ich im hertzen traghe” from the *Wolfenbüttel Lute Tablature*.

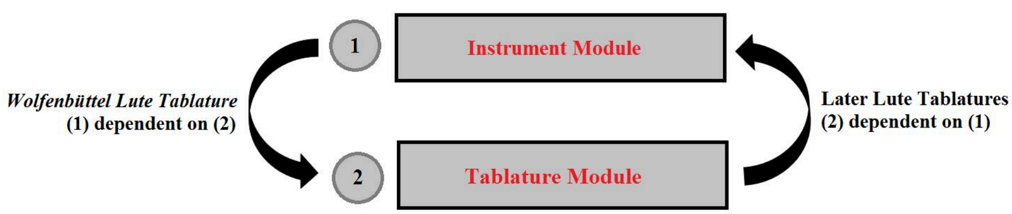


Figure 5: Relationship between the Instrument Module and Tablature Module for later lute tablature (specifically French and Italian) and for KWTS-High (as exemplified in the *Wolfenbüttel Lute Tablature*.

```
<instrumentation>
    <instrVoice solo="true" code="lute">
        <instrDesc>
            <instrName>Lute</instrName>
            <courseCount>5</courseCount></instrDesc>
        <instrConfig label="renaissance">
            <courseTuning>
                <course n="1" pname="d" oct="5"/>
                <course n="2" pname="a" oct="4"/>
                <course n="3" pname="e" oct="4"/>
                <course n="4" pname="c" oct="4">
                    <string pname="c" oct="5"/>
                    <string pname="c" oct="4"/></course>
                <course n="5" pname="g" oct="3">
                    <string pname="g" oct="4"/>
                    <string pname="g" oct="3"/></course>
            </courseTuning>
        </instrConfig>
    </instrVoice>
</instrumentation>
```

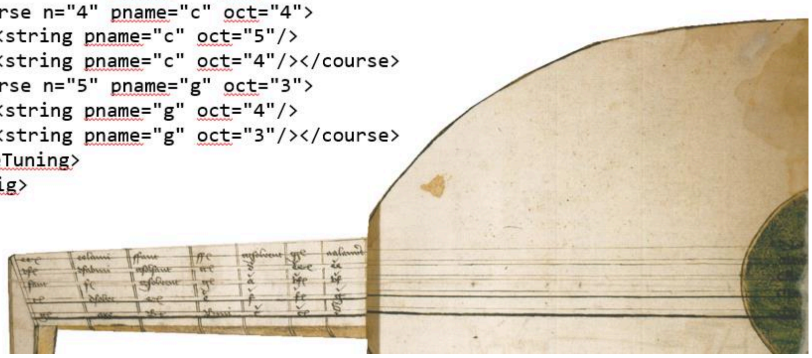


Figure 6: Encoding of instrument for the *Wolfenbüttel Lute Tablature*(KWTS-High).

There are, however, issues with this encoding system that are yet to be resolved. As with later lute tablatures, the music—and, therefore, its encoding system—is notated as a series of vertical, rhythmic durations, rather than as horizontal voices that exist individually and in relation to each other. Quite simply, the notation lacks contrapuntal information; the interaction of the voices is only realized in performance. While organ tablature—and other polyphonic works from this time—spatially demarcate the voices on the page, the *Wolfenbüttel Lute Tablature* combines all of its voices, plus idiomatic chords, on the same staff system, obscuring the voice leading of each line (see Figure 7). If notated differently, the polyphonic voices could be separated in the encoding process using the `layers` element, but this is not possible because of the manuscript's "strike" notation. Rather, the voices must be separated in a separate stage so that both diplomatic and polyphonic transcriptions are possible.

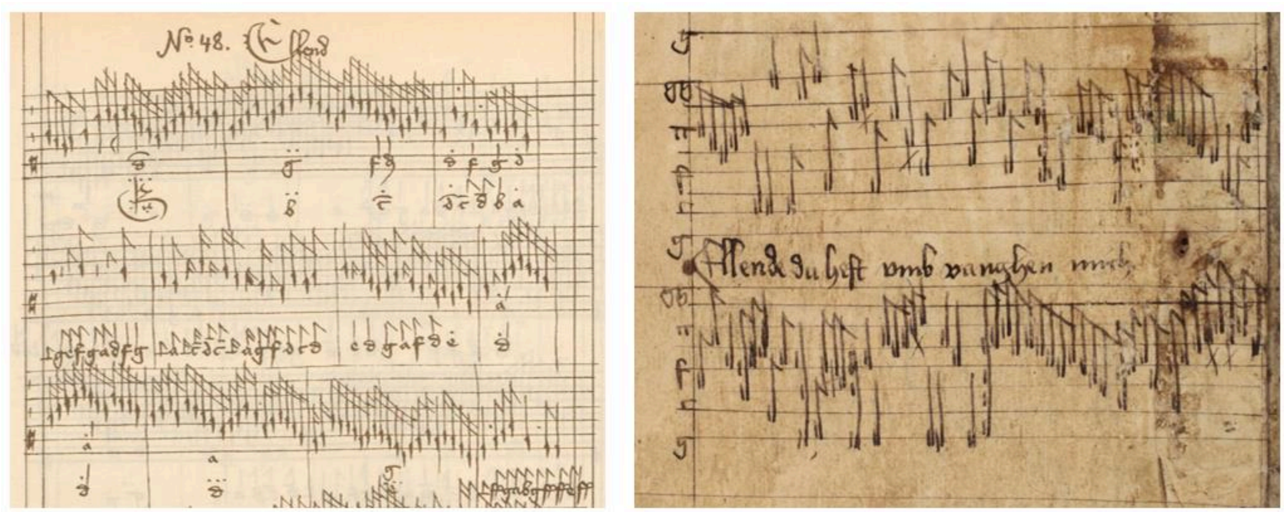


Figure 7: Note the spatial demarcation of the voices in (left) the Buxheimer Orgelbuch as opposed to their equivalent in (right) the *Wolfenbüttel Lute Tablature.*

Reiner de Valk and Tillman Weyde have recently developed a means of voice extraction for sixteenth-century lute tablature that also accounts for duration reconstruction, as the notated and performed note values are not always synonymous in lute tablature due to the inherent nature of "strike" notation. Their system uses pitch, time, and course proximity to assign notes to their correct voice, and determines each note's duration by considering the next note in that voice or on that course. [11, p.563-576] Although it was developed for later lute tablatures, I see no reason why it could not also be applied to the *Wolfenbüttel Lute Tablature*.

We will now turn, briefly, to the second portion of KWTS, KTWS-Low, for which no lute-specific music survives, if it ever existed. *Kassel Lautenkragen* defines a couple of signs particular to this portion of the notation (see Figure 8), apart from the fingerboard diagram. The first, the "equivalence signs" are used in the fingerboard diagram to show which pitches are also playable on a higher course and lower fret, and likely have no notational purpose outside of this source, so I have not assigned them an encoding element or attribute. The second sign is the common shorthand for a chromatically altered pitch, which is also used in contemporaneous organ tablatures, and indicates a sharp when attached to all but B or E, when it is used as a flat sign.

Sign from “Kassel Lautenkragen”	Appearance in *Buxheimer Orgelbuch*	Equivalent Sign in *Wolfenbüttel Lute Tablature*	Meaning	Encoding Element/ Attribute
	x	x	*Signa equivalencia* (equivalence signs) **specific to the lute neck diagram to show equivalent pitches	x
		x	Sign attached to letter to indicate chromatic alteration: ♯ if attached to anything except E or B (then, ♭)	@accid

Figure 8: Encoding KWTS-Low: signs from *Kassel Lautenkragen* and their equivalent in the *Buxheimer Orgelbuch*.

The next table (see Figure 9) shows the signs assigned to each course and fret of the lute on the left, and their corresponding sign in Old German Organ Tablature (taken from the Buxheimer Orgelbuch) on the right. The only differences are that the *Kassel Lautenkragen*, for the most part, writes the full name of each pitch, where the Buxheimer Orgelbuch, being a practical source, uses their respective shorthand, and where the *Kassel Lautenkragen* uses a repeated letter to indicate the upper octave, the Buxheimer Orgelbuch uses a line above the letter. Using TabCode to encode this notation is relatively simple, as I alluded to earlier. The only major alteration required is that the `@course` and `@fret` attributes would need to be combined into a single attribute that simultaneously encodes both pieces of information. As the signs in *Kassel Lautenkragen* correspond to actual pitches, not just the fret and course positions, I would suggest adopting the `@pname` (pitch name) attribute and incorporating it into the `tabNote` element of TabCode. Just as with KWTS-High, the instrument module would then be applied to the encoded material of KWTS-Low to transcribe the music into later lute tablatures.

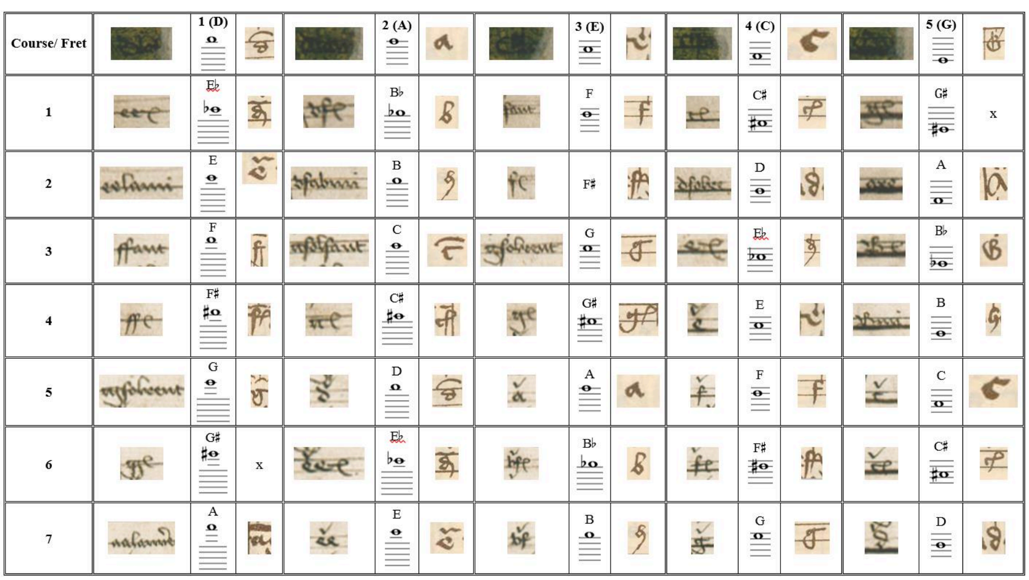


Figure 9: Lute neck diagram with the pitches of each course (column) and fret (row) combination as they appear in *Kassel Lautenkragen* (left-hand column of each group of three); their modern pitch-equivalent in bass clef (middle); and their equivalent in Old German Organ Tablature, taken from the *Buxheimer Orgelbuch* (right).

Musicological applications

It is in its use as a transcription tool that the main value of encoding KWTS lies. Given the limited extant sources for lute in this notation—namely the *Wolfenbüttel Lute Tablature*—it is hardly worth encoding these pieces: five fragmented songs are hardly going to unlock the mysteries of the universe. However, if we employ *Kassel Lautenkragen* as a translation guide from organ tablature to lute performance, then we unlock a much larger dataset that would benefit from the empirical analysis that encoding enables. By using the methods outlined for KWTS-High and -Low to encode the upper and lower voices of organ tablature respectively, and then applying the instrument module to this encoded material, it would be possible to substantiate the longstanding claim that lutenists, prior to the development of lute-specific notation, often played from organ tablatures, including the *Buxheimer Orgelbuch*.[3] Apart from a note above one of the pieces in the *Buxheimer Orgelbuch* (*Je loe amours* by Gilles Binchois) that suggests its suitability to lute performance, and the fact that lute duos—as well as lutes in a variety of multi-instrument ensembles—existed in the fifteenth century, not much else can be said with any degree of certainty.[4].

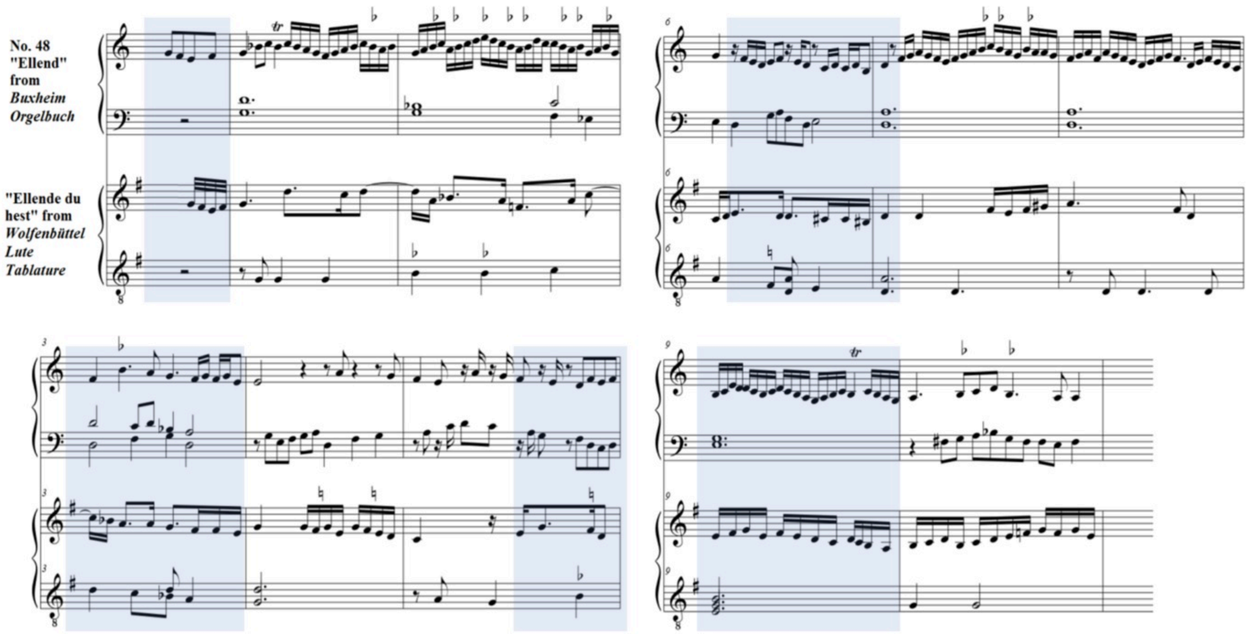



Figure 10: Comparison of the *Wolfenbüttel* arrangement of *Ellend du hast* with one of the versions found in the *Buxheimer Orgelbuch* (No. 48).

By encoding the tools offered by the *Kassel Lautenkragen*, this could change. Moreover, as the first source of lute-specific notation, and because of its links to *Kassel Lautenkragen*, and thus to organ tablatures, the *Wolfenbüttel Lute Tablature* gains significance in the music encoding realm as the link between a time without lute-specific notation to a time with such notations. Three of the fragments from the *Wolfenbüttel Lute Tablature* have concordances with the *Buxheimer Orgelbuch*, among other manuscripts. [5, p.434-579] *Ellend du hast* (Sadness You Have), the fifth fragment of the *Wolfenbüttel Lute Tablature*, in particular, shares

3. As Martin Kirnbauer states in his commentary on the *Kassel Lautenkragen*, "a majority of surviving fifteenth-century sources of the so-called German organ tablature are playable with the help of the Kassel source, on lute(s)" [12, p.189]. See also: [5, p.9], [12, p.10-12], [1, p.32].
4. The note below *Je loe amours* by Gilles Binchois reads "in cytaris vel etiam in organis" (on lutes or organs): [4, p.31-32]. See also: [12, p.12], [9, p.30]

polyphonic material with at least one of the versions of the same piece in the *Buxheimer Orgelbuch*. As the earliest source of *Ellend du hast* was monophonic, not polyphonic, these similarities cannot be attributed to a shared polyphonic vocal source. Nonetheless, the similarities between the two arrangements suggest a shared inspirational source—whether written or performed. Not only are the tenor lines nearly identical but so too are aspects of the cantus line, with the *Wolfenbüttel* arrangement a simplified version of the *Buxheim* version (see Figure 10). As professional musicians in Germany regularly performed instrumental arrangements of the songs found in the *Buxheimer Orgelbuch* and the *Wolfenbüttel Lute Tablature*, it is not unlikely that the musicians behind both sources were the same, or heard a similar version of *Ellend du hast* performed. Perhaps the *Wolfenbüttel Lute Tablature* rendition is a simplified version of a lute duo arrangement, altered for an aspiring solo musician who did not want to be dependent on his tenorist for polyphony. By encoding both the music in the *Wolfenbüttel Lute Tablature* and the *Buxheimer Orgelbuch*, more similarities may present themselves, whether these are direct quotations, intervallic concordances, or stylistic similarities, all of which could be identifiable via "search and match operations" between the two sources, but not by traditional analysis. [3, p.644]

Conclusion

By applying KWT as a translation tool to fifteenth-century organ tablatures, perhaps the connection between early organ tablatures and lute performances prior to lute notation can be confirmed. By establishing the alterations needed to turn a "keyboard" piece into one suitable for lute performance, if changes are indeed necessary, it would be possible to conceptually recreate lute performances prior to lute notation, and, thus, widen our knowledge of this instrument and its early repertory. In doing so, one would add a hitherto-little-explored dataset to the existing collection of encoded lute tablature, most of which is from the sixteenth century. In short, it would add a valuable resource to the arsenal of anyone seeking to understand the development of the Western lute and its repertory.

Works cited

[1] Coelho, Victor. *The Manuscript Sources of Seventeenth Century Italian Lute Music*. New York: Garland, 1995.

[2] Crawford, Tim, Geraint Wiggins, Michael Gale, and David Lewis. *ECOLM—An Electronic Corpus of Lute Music*. Goldsmiths College, 2011. (http://ecolm.org).

[3] Crawford, Tim, Richard J. Lewis, and David Lewis. "Exploring Information Retrieval, Semantic Technologies and Workflows for Music Scholarship: The Transforming Musicology Project" *Early Music* 43.4 (2015): 635-647.

[4] Fallows, David. "Fifteenth-Century Tablatures for Plucked Instruments: A Summary, a Revision, and a Suggestion" *Lute Society Journal* 19 (1977): 7-33.

[5] Fallows, David. *A Catalogue of Polyphonic Songs, 1415-1480*. Oxford: Oxford University Press, 1999.

[6] Ivanoff, Vladimir. "An Invitation to the Fifteenth-Century Plectrum Lute: The Pesaro Manuscript" in *Performance on Lute, Guitar, and Vihuela*, ed. by Victor Coelho. Cambridge: Cambridge University Press, 1997, 1-15.

[7] Lewis, Richard. "Customising MEI for Lute Tablature". *Transforming Musicology: An AHRC Digital Transformations Project* Blog post, 21 May 2014 (www.transforming-musicology.org/blog/2014-05-21_customising-mei-for-lute-tablature).

[8] Lewon, Marc. "The *Wolfenbüttel Lute Tablature*". *mLewon: Research News from Marc Lewon* Blog post, 22 February 2014 (http://mlewon.wordpress.com/category/wolfenbuttel/).

[9] Polk, Keith. *German Instrumental Music of the Late Middle Ages: Players, Patrons, and Performance Practice*. Cambridge: Cambridge University Press, 1992.

[10] Staehelin, Martin. "Norddeutsche Fragment emit Lautenmusik um 1460 in Wolfenbüttel" in *Kleinüberlieferung mehrstimmiger Musik vor 1550 in deutschen Sprachgebiet*, Series IX, Neue Quellen des Spätmittelalters aus Deutschland und der Schweiz. Berlin: Gruyter, 2012, 67-88.

[11] De Valk, Reiner and Tillman Weyde. "Bringing 'Music into the Tablature': Machine-Learning Models for Polyphonic Transcription of Sixteenth-Century Lute Tablature" *Early Music* 43.4 (2015): 563-76.

[12] Young, Crawford and Martin Kirnbauer, eds. *Early Lute Tablatures in Facsimile*. Winterthur: Amadeus, 2003.

Chants That Defy Classification: Implications of Categorization in the Cantus Database

Barbara Swanson
University of Waterloo
bswanson@dal.ca

Debra Lacoste
University of Waterloo
dlacoste@uwaterloo.ca

Abstract

Entering metadata for medieval chants in the Cantus Database often poses challenges for the indexer. Although many of the chants follow the standard, expected forms, some items are difficult to classify. For example, certain melodies do not conform to the standard modes and some chants are unique in their usage for a particular day or service within the liturgical year. Although it is easy to create new classification tags, the prudent database manager exercises caution, as the modern need to categorize does not always conform to medieval custom. By focusing on chants from Holy Week including the *Exsultet*, *Improperia* and *Trisagion*, this paper addresses the various criteria used to create two new genre codes, "Varia" (Va) and "Holy Week Varia" (VaHW). Although these classifiers provide less specificity than other possible options, they avoid over-interpretation, are easy-to-use by novice indexers, and usefully differentiate the unique chants of Holy Week from other chant miscellany.

Introduction

The *Cantus Database* is a digital archive containing inventories of medieval chant manuscripts.[1]. The database has more than 400,000 records of individual chants as well as information regarding every indexed manuscript. Each individual chant record includes substantial metadata, including genre, mode, liturgy, and feast, with the option to include more, such as full melodies. Used by individuals world-wide through the Internet, the data in the *Cantus Database* is also integral to the development of the *Cantus Ultimus* project, which involves the scanning and processing of digital images of chant manuscripts with Optical Music Recognition (OMR).[2]

Identifying the contents of medieval chant manuscripts requires more than simple data-entry, and even skilled musicologists experience challenges. For example, although most chants in a manuscript fit standard types, some chants can be difficult to classify, such as those texts that are unique to a particular day within the liturgical year. As well, chant melodies do not always conform to the standard modes. Medieval scribes further complicate matters when they leave scant information regarding how and when to sing the chants, or

1. Debra Lacoste (Project Manager and Principal Researcher) and Jan Koláček (Web Developer), *Cantus: A Database for Latin Ecclesiastical Chant — Inventories of Chant Sources* (http://cantus.uwaterloo.ca)
2. The *Cantus Database* is housed at the University of Waterloo, and the *Cantus Ultimus* project is in development primarily at McGill University alongside *SIMSSA: Single Interface for Music Score Searching and Analysis*, Project Director: Ichiro Fujinaga (https://cantus.simssa.ca).

squeeze seemingly miscellaneous chants into the margins of a folio. Careful decision-making is essential so that users can successfully retrieve difficult-to-classify data. While entering ambiguous information into the database is not desirable, entering a value can sometimes seem preferable to leaving a field blank.

As this paper will demonstrate, a “Miscellaneous” category is sometimes necessary, in part because it provides a value which, however vague, links difficult-to-classify chants into a searchable group. Even though some users might prefer more specificity, the database serves a large user group including experienced academics and students, librarians, amateur chant choir directors, interested members of the general public, and those who have purchased manuscript fragments from art dealers and want to find out more about their artefacts. A “Miscellaneous” category accommodates the wide-ranging expertise of *Cantus* users, and avoids over-interpreting content or introducing potential errors.

Background: *Cantus Manuscript Database* and *Cantus Index*

Over the past few years, the original *Cantus: A Database for Latin Ecclesiastical Chant* has been divided into two separate databases, the *Cantus Manuscript Database* and the *Cantus Index*. A recent rebranding has attempted to clarify the purposes of each website. The *Cantus Manuscript Database* more closely resembles the original chant research resource, which was begun in the late 1980s. This is the site where manuscript records are stored and where searching and analysis of chants and their traditions takes place. The *Cantus Index* is the central catalogue of chant texts and melodies for both the Office and the Mass.[3] Several other partner databases containing chants from manuscript traditions across Europe are interconnected through the use of *Cantus* ID Numbers. A single search for a chant text (or in some cases a melody) in *Cantus Index* will return results in all partner databases, including the original *Cantus Database*.

Until recently, *Cantus* was primarily a database for chants of the liturgical Offices. In August 2015, Graduals and Mass chants were added to the database, beginning with the Bellelay Gradual (Porrentruy, Bibliothèque cantonale jurassienne, Ms 18) [1]. As the next section of this paper will demonstrate, many issues arose during the indexing of the Bellelay Gradual, including questions of classification. Decisions regarding difficult-to-categorize chants were made collaboratively by a team of scholars including Jennifer Bain, Inga Behrendt, and Elsa de Luca, as well as Debra Lacoste and Barbara Swanson. Other scholars were consulted as the need arose, including Franz Karl Prassl, Kate Helsen, and Armin Karim.

Miscellaneous *plus*: Varia (Va) and Holy Week Varia (VaHW)

The classifications used within *Cantus* are carefully chosen. Most are standard across the field of medieval musicology, drawn from printed manuscript indexes and other chant research resources. Others have been established as the standard by *Cantus* and now are being followed by other projects and individuals. For instance, several of the partner databases now connected with *Cantus Index* were formed using the *Cantus Database* structure. The *Portuguese Early Music Database* and *Gradualia*, independent pre-existing projects which are now networked into *Cantus Index*, have also adopted *Cantus* ID Numbers for their chants texts.

3. Debra Lacoste (Project Manager) and Jan Koláček (Project Founder and Developer), *Cantus Index: Catalogue of Chant Texts and Melodies*, available from http://cantusindex.org.

A number of chants in the Bellelay Gradual did not conform to the established taxonomy of genre categories in the *Cantus Database* and *Cantus Index*. Leaving the “Genre” field blank for such chants was not an option, as genre is one of the most important classifiers in *Cantus*. For instance, the genre of a chant is tied to its liturgical function; it suggests the general patterns of pitches and performance practices, such as where the cantor and choir sing, as well as where musical sections should be repeated. Essentially, it is a concentrated descriptor that in a single word or code reveals a whole association of musical and extra-musical factors. The genre category further groups similar chants together, making them searchable as a unit across sources.

Our team of scholars initially considered classifying these chants in the Bellelay Gradual as paraliturgical. The term paraliturgical has sometimes been used to classify rituals, including chants, that are not an element of the standard Mass or Offices of the Catholic church (for example, tropes, sequences, conductus, and the liturgical dramas of Easter and Christmas). As a category, paraliturgical suggests an unofficial, optional, or extra-liturgical element, as opposed to the official and required elements of regular liturgies. Had we adopted this as a genre, the abbreviation would have been **pL**. In the end, we decided against paraliturgical as a descriptor, since medieval sources do not distinguish between liturgical and paraliturgical elements. In the words of Keith Falconer, “there is little trace of this notion in the ordinals. I think we must assume for the sake of consistency that any music can be liturgical if performed in a church” [2, p. 70] [3, p. 40].

Rather than “Paraliturgical,” we instead created the genre “Varia” with the abbreviation **Va** in keeping with René-Jean Hesbert’s same category in the *Corpus Antiphonalium Officii* [4]. In consultation with Armin Karim, we created another new genre code specifically for Holy Week: **VaHW** [5]. While these categories are essentially a “Miscellaneous” grouping, **VaHW** in particular allows researchers to more easily explore the traditions specific to Holy Week. The usefulness of this latter code emerged after extensive conversation regarding three Holy Week chants in particular: the *Exsultet*, *Improperia* and the *Trisagion*.

Case Study 1: The *Exsultet*

The *Exsultet* is sung by a deacon during the Easter Vigil for the blessing of the Paschal candle. The text is tripartite, as seen in Table 1. Until recently, the *Exsultet* was categorized in the *Cantus Database* as a Hymn (**H**). We revisited this while indexing the Bellelay Gradual, as “Hymn” is not a completely accurate description of the chant. Although the *Exsultet* has a strophic musical structure in part, this structure is not continuous throughout the entire chant. Our research led to a variety of other possible descriptors: from Prayer and Preface to a hybrid of both.[4]

Section	**Exsultet text**
Introduction	*Exsultet iam angelica … / Gaudeat … / Laetetur …*
Dialogue	*Dominus vobiscum. Et cum spiritu tuo. Sursum corda*
Preface	*Dignum et justum est, vere quia dignum …* OR *Vere dignum et justum est …*

4. Hiley, *Western Plainchant*: [6], p. 38; Huglo and Kelly, “Exultet” [7]; Harper, *Western Liturgy* [8], p. 147; Hughes, *Medieval Manuscripts for Mass and Office* [9], p. 22, 248.

In choosing the most appropriate genre code, we considered various factors: the needs of the database and its users, the characteristics of the chant, medieval descriptors of the chant, and classification systems used in comparable collections or databases. Regarding needs of the database and users, we asked what would be easiest for indexers to implement. We also asked what would be most useful to a wide variety of researchers, each with different ideas about the chant. In considering the characteristics of the *Exsultet*, we prioritized accurate description of the chant, and asked if perhaps the *Exsultet* merited its own genre code. We wondered if medieval users considered the *Exsultet* unique within the liturgical year and if this should influence our decision-making. We also considered the suitability of the genre code for possible future melodic encoding.

As we engaged with these criteria, Inga Behrendt suggested creating a new code that would identify the chant as both a recitational chant and as a preface, revealing information about the musical characteristics and liturgical function of the chant (**RecPref**). For the sake of the database and users, Debra Lacoste advocated using a category currently in existence, as fewer genres simplify the indexing process, and genre descriptions can be lengthened to include a wider range of chants. With this method we would have used the existing genre code **Pr** for "Preface" and we would have expanded the description for that genre to include the *Exsultet*. Taking into account the unique character of the *Exsultet* within the medieval liturgy, Franz Karl Prassl suggested that it might merit its own code (**Ex**) to best represent its unique musical mixing of preface with extended recitational sections. We wondered, however, if this code would be redundant, as the text is unique to the *Exsultet* and a text search would bring up only this chant for this liturgical occasion. Barbara Swanson suggested following David Hiley's classification of the *Exsultet* as a prayer genre, which could further encompass Hiley's prayer categories of Prefaces, Litanies, *Exsultet*, and *Pater noster*. This would require repurposing **Pr** for Prayer or introducing **Or**, for the Latin *Oratio*.

In the end, we instead chose to categorize the *Exsultet* as **VaHW**. This new code arguably coincides with a more medieval way of thinking about this chant: a special chant among others that are unique to Holy Week. This code further facilitates comparison of the unique chants of Holy Week within and between sources, perhaps generating new research findings. Ultimately, **VaHW** allowed us to avoid taking sides in various debates regarding the character of the *Exsultet*: a Preface? a Hymn? a Prayer? or other?

Case Study 2: *Improperia* and *Trisagion*

The *Improperia* and *Trisagion* are chants sung on Good Friday during the Veneration of the Cross. There are many variant versions of the *Improperia*, but one of the most common begins "Popule meus" and is sung in alternation with the *Trisagion*, as seen in Table 2. The *Trisagion* is an accretion from the Byzantine Divine Liturgy and comprises three invocations of God as "Holy," often sung in both Greek and Latin after each section of the *Improperia*.

Improperia verses (which alternate with Trisagion)	Trisagion
Popule meus, quid feci tibi? …	Agios O Theos, Sanctus Deus
Quia eduxi te de terra Aegypti …	Agios ischyros, Sanctus fortis
Quia eduxi te per desertum …	Agios athanatos, Sanctus immortalis
Quid ultra debui facere tibi …	eleison imas, miserere nobis

While discussing possible genre codes for the *Exsultet*, we also considered categorizing the *Improperia* with the code **Im**. Whereas a unique genre code for the *Exsultet* seemed redundant given that it is a unique chant, the *Popule meus* text also functions as an Antiphon and as a Responsory during the Offices. Thus, while the *Exsultet* could be searched as a text string and would always return the Holy Week chant for the Easter Vigil, a search on the text *Popule meus* would return chants with different liturgical functions. **Im** was also an attractive descriptor as it would link together the many different chants that qualify as *Improperia*, including the chants often referred to as the Greater Reproaches, the Lesser Reproaches, and the Aquitanian Reproaches. Using **Im** would clarify when a *Popule meus* text belonged to the Good Friday Veneration of the Cross and when it did not.

The possibility of using **Im** as a genre code introduced a further question: given that the *Trisagion* is typically sung in alternation with the *Improperia*, should the genre code include the *Trisagion*? It was felt that the *Trisagion* merited its own code, as it is a unique chant with its own liturgical history. **Tg** was one option, and **Gk** for "Greek" was another, given that the *Trisagion* is an accretion to the Western liturgy from the Byzantine rite. **Gk** would then include other Greek chants used in the Latin liturgy, like the *Kyrie*.

Although each of these codes were defensible options, distinct drawbacks discouraged us from implementing them. For one, each code required expert knowledge from the indexer and, in the case of "Greek," would require a complicated re-classification of other genres in the database. **VaHW**, however, could be easily implemented by users and would require much less adjustment within the database overall. While **VaHW** provides less specificity than the precision of an **Im** or **Tg** code, it provides more specificity than **Va**. Although any loss of specificity is in some ways lamentable, it is important to note that most *Cantus* genres include hundreds of different chant texts. **Im** would have included at most twenty chants and **Tg** perhaps ten.

Conclusions

Multiple criteria determined the classification decisions described above, including the character of specific chants, research merits, and usability. Careful consideration of the various criteria led to the creation of **Va** and **VaHW**, with usability ultimately being the most important decision-making factor. Although it is possible to teach a novice indexer about the primary chant genres, many difficult-to-categorize chants require expert knowledge. Even among experts, however, there may be little agreement about the best descriptor for a difficult-to-classify chant. A "Varia" category serves as a catchall, with the "Holy Week Varia" category identifying a group of unique chants within a specific season of the liturgical year, more in keeping with how medieval users likely thought.

The case studies presented here reveal only a few of the many classification issues within the *Cantus Database*. A much longer list of classification issues includes the question of *Kyrie* tropes, "C-mode" chants, chants with uncertain modes, the relationship of Alleluias and their verses, differences between long and short responsories, as well as regional chant practices with unique genres and liturgies, such as the Ambrosian rite. As well, there are questions of database management: for example, would a level-of-certainty indicator for each field be useful, revealing the ease or difficulty with which a classification decision was made?

Although it is possible and easy to create new classification tags, the prudent database manager should be cautious regarding new and existing assignments for items of the medieval liturgy. The monks and canons of the Middle Ages knew what to sing and when to sing it, even if it occurred just once every year, and our modern need to categorize does not always suit medieval customs. The *Cantus Database* helps researchers to find meaning in manuscripts by accurately representing and not over-interpreting the data. When we at *Cantus* debate (sometimes seemingly endlessly) the merits of a category, or when we change our minds regarding a genre, it is because we are carefully considering the implications of classification.

Works cited

[1] Swanson, Barbara. "The Cantus Gradual Project: Porrentruy, Bibliothèque cantonale jurassienne, 18" in *New Research on Old Chants in Honour of Andrew Hughes. Proceedings of the Eleventh Annual Colloquium of the Gregorian Institute of Canada, Saint Augustine's Seminary, Toronto, August 11-14, 2016* ed. by Kate Helsen. Kitchener, ON: Institute of Mediaeval Music, forthcoming.

[2] Falconer, Keith. "Ritual Reflections" in *Companion to Medieval and Renaissance Music*, ed. by Tess Knighton and David Fallows. Berkeley: University of California Press, 1992: 69-73.

[3] Flanigan, C. Clifford. "'Quid Quaeritis, O Clerici?' A Review of Some Recent Scholarship on the Medieval Latin Music Drama" in *The Ritual Life of Medieval Europe: Papers by and for C. Clifford Flanigan* (ROMARD: Research on Medieval and Renaissance Drama, 52-53). London, Ontario: Western University, 2014, 35-60.

[4] René-Jean Hesbert, ed., *Corpus Antiphonalium Officii*, vol. 1: *Manuscripti "Cursus Romanus"*; vol. 2: *Manuscripti "Cursus Monasticus"*; vol. 3: *Invitatoria et Antiphonae*; vol. 4: *Responsoria, Versus, Hymni et Varia*; vol. 5: *Fontes Earumque Prima Ordinatio*; vol. 6: *Secunda et tertia ordinationes*. Rome: Herder, 1963-1979.

[5] Karim, Armin. "'My People, What Have I Done to You?': The Good Friday *Popule meus* Verses in Chant and Exegesis, c. 380-880". Case Western Reserve University, 2014. PhD Dissertation.

[6] Hiley, David. *A History of Western Plainchant*. Oxford: Clarendon Press, 1995.

[7] Huglo, Michel and Thomas Forrest Kelly. S.v. "Exultet" *Oxford Music Online*. Available on subscription at *www.oxfordmusiconline.com/subscriber/article/grove/music/09148*. Accessed 17 November 2015.

[8] Harper, John. *Forms and Orders of Western Liturgy from the Tenth to the Eighteenth Century*. Oxford: Clarendon Press, 1991.

[9] Hughes, Andrew. *Medieval Manuscripts for Mass and Office*. Toronto: University of Toronto Press, 1995.

[10] Huglo, Michael. S.v. "Gallican chant." *Oxford Music Online*. Available on subscription at *www.oxfordmusiconline.com/subscriber/article/grove/music/09148*. Accessed 17 November 2015.

The Freischütz Performance in Vienna: Encoded Representation of Performance-Related Modifications of the Score

Agnes Seipelt
Musikwissenschaftliches Seminar Detmold/Paderborn - Unversity of Paderborn
aseipelt@mail.upb.de

Abstract

The first performance of Weber's *Freischütz* in Vienna took place November 3rd, 1822. A manuscript copy was bought from Weber, which closely followed the composer's autograph. But a close look to the manuscript reveals a great number of performance-related modifications of both the musical and textual content. These transformations have mostly been major ones, like deletions of whole measures or text passages, and insertions of new pages. On the basis of the BMBF-sponsored project "Freischütz Digital" (Digital edition of the *Freischütz*, hereafter "FreiDi"), this paper tries to develop a way to describe the textual and physical modifications of the manuscript not only verbally but in a semantic, i.e. logic, encoding, based on MEI. For this, the paper explores possible modifications of MEI elements, or the creation of new elements and attributes for the special problems. It will explore the usefulness of markup from other projects, most importantly the *Beethoven Werkstatt* project with its genetic encodings. It will also discuss the possibilities, problems and limitations of the differents solutions.

Introduction

The manuscript of the *Freischütz* first performance in Vienna in November 1821 is interesting for historical musicology, music philology, music editing and encoding. The manuscript contains multiple modifications which can mostly be traced back to the censorship regulations in Vienna at around 1821.

There are several numbers in the *Freischütz* containing interesting modifications of both the textual content and the music manuscript itself. This paper aims to show a few of them in two numers of the opera: The 4th movement contains modifications of the textual content and the whole number was completely replaced by a new, modified copy of Vienna. Number 13 is characterized by modifications of the original manuscript itself like deletions of measures, radical curtailments by foldings of pages and insertions of new pages that show several processes of transformation of the score.

The digital edition of the *Freischütz*, based on Edirom, presents all important sources of the opera in scans with the possibility of a bar-by-bar concordance.[1] The MEI-encoding is the basis of all these features. While textural modifications in the librettos are recorded in TEI, the music encoding mostly ignores the textual

1. The FreiDi is accessible at http://freischuetz-digital.de/edition.

modifications by the censorship and other later modifications. The only exception to this is an encoding of deleted passages or additions of new copies, that are described by `subst`, `del` and `add` elements so that the structure of the numbers is roughly documented at the `measure`-level.

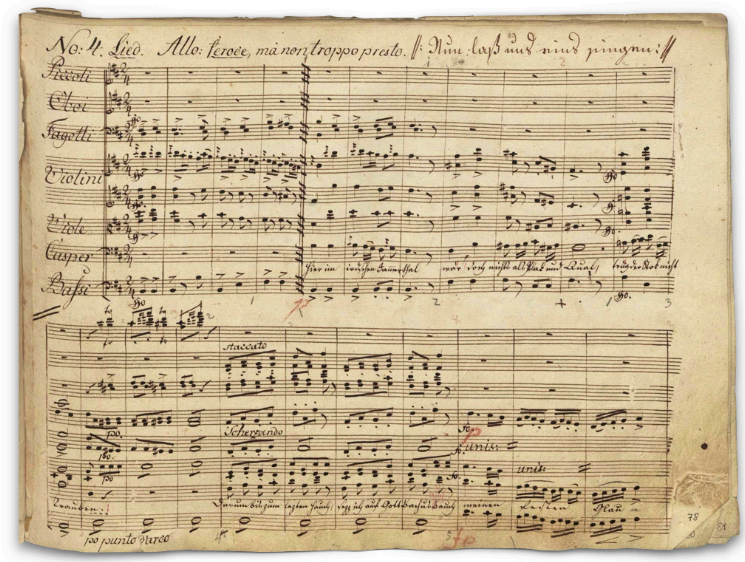



Figure 1: Carl Maria von Weber: Der Freyschütz, Viennese copy, A-Wn OA.373, No.4, page 1 (fol. 78r)

No.4

The first example is Caspar's drinking song in the first act. Weber's original version of this short song is written on two pages and has 31 measures. The text for the second and third stanza is written with interspersed dialogues on the second page. Both stanzas are to be sung to measures 5 to 31, which are thus to be repeated two times (see figures 1 and 2). In the Vienna copy, we find an inserted empty page after this original version of the song, as well as 7 following pages containing a new copy of this piece. In this, both repetitions have been written out and patches with dialogs are to be found between the pages. The new copy starts at measure 23—that corresponds to the beginning of the second page of the original copy. A rest of sewing cotton on this page reveals that there has been a page attached by sewing. It seems reasonable to assume the reason for this new copy in the modifications to the lyrics (see figure 2).

On the second page of the original copy, one can clearly see modifications in the second and third stanza, the latter being additionally rewritten beside the old text with red pencil and not very readable. It is very clear from the changes in stanza 2 that both religious and frivolous vocabulary had to be replaced for Vienna; so the words "runder Brust" ("round bosom") is replaced by "treuer Brust" ("faithful bosom") and "ewgen Leben" ("eternal live") by "frohen Leben" ("happy live"). In the musical text only dynamics were changed. All later modifications of the old copy are also to be found in the new copy. Additionally, a new staff above the bass staff is added in the new copy which strangely enough consists only of a colla parte indication for the cello players. All in all, there are no substantial changes in the musical content. So we could ask why the Vienna copyist did not imitate the original structure of the piece with the two stanzas given separately. Maybe the integration of the cello staff indicates that this new structure is more clear for the process of writing out the parts—or it is simply a hint that the copyist was paid by pages.

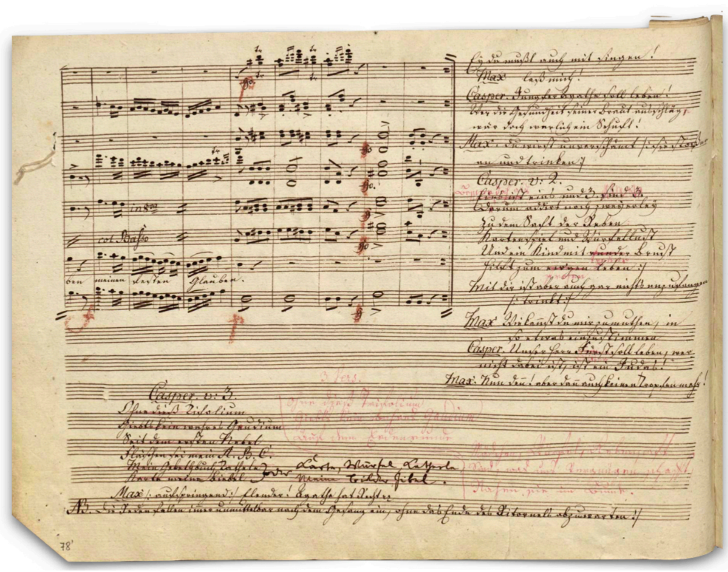



Figure 2: Carl Maria von Weber: Der Freyschütz, Viennese copy, A-Wn OA.373, No.4, fol. 78v

Relations between elements

@sameas and *@copyof*

In the source file of FreiDi the substitution of the original text and the new copy is encoded with `del` and `add`—elements that are nested in a `subst` element. Based on the [core-concept](), `@sameas` attributes in the `measure` elements refer to the copied notes and other correspondent elements of the `core`. Because there is no other source in FreiDi that has an extension of the repetition and the core-file only consists of the measures that are part of the autograph, there is no way to refer for example from measure 32 (first measure of the second stanza) of the source-file to the correspondent measure 32 of the core-file because it does not exist. But, fortunately that it is a repetition, one can also refer to the equivalent measure of the first stanza—measure 5 (if we disregard the different songtext):

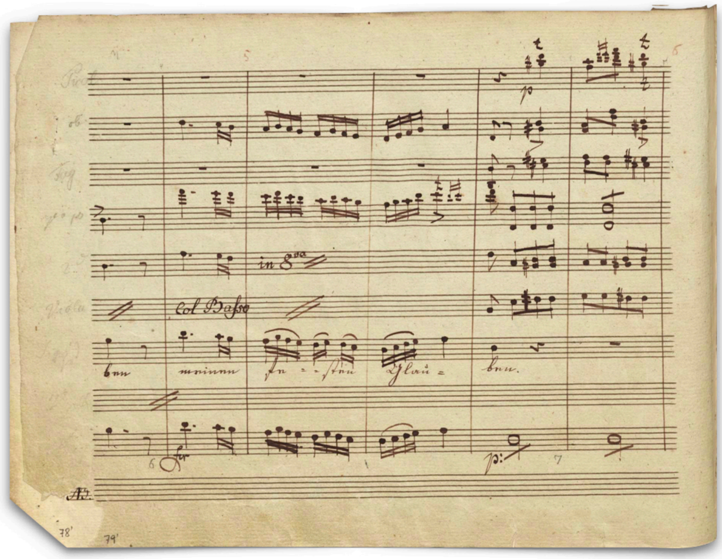



Figure 3: Carl Maria von Weber: Der Freyschütz, Viennese copy, [A-Wn OA.373](), No.4, fol. 79r

```
<section xml:id="stanza2">
   <measure n="32" xml:id="KA2_stage2_mov4_measure32" facs="#KA2_zoneOf_stage2_mov4_measure32"
      sameas="../core.xml#core_mov4_measure5"/>
   <measure n="33" xml:id="KA2_stage2_mov4_measure33" facs="#KA2_zoneOf_stage2_mov4_measure33"
      sameas="../core.xml#core_mov4_measure6"/>
   <measure n="34" xml:id="KA2_stage2_mov4_measure34" facs="#KA2_zoneOf_stage2_mov4_measure34"
      sameas="../core.xml#core_mov4_measure7"/>
  </section>
```

The combination of the original copy and the new vienna copy with the new, copied elements (measures 32 to 85) forms a complex network of different relations between these elements. In addition to the relations between the source-file and the core-file, there is also a network of relations *within* the source-file. For instance, the measures of the third stanza are structurally and musically not only in a relationship with the equal measures of the first stanza but also with the new copied measures of the second stanza. And these are also connected in a way with the other stanzas because they are all quasi copies from each other. These relationships can also be applied to other elements like `section`, that are also linked in a way:

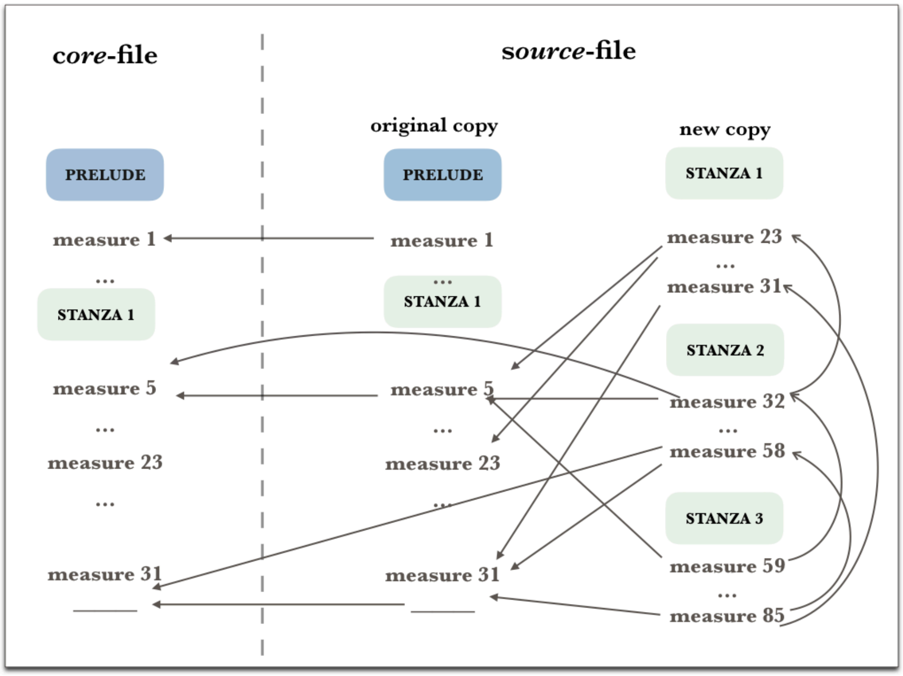


Figure 4: Network of relations between MEI-elements

The attribute @sameas is part of the MEI module 7: Analytical Information / 7.1: General Relationships between Elements and *"points to an element that is the same as the current element but is not a literal copy of the current element."* [3, p.141] It can be used to refer to the exact same element, for example if two layers share the same notehead. In this case, @sameas is not exactly the perfect attribute for the copied elements. The chapter *General Relationships between Elements* of the Guidelines offers more attributes like @copyof, which *"points to an element of which the current element is a copy."* [3, p.141] For example, @copyof can be used when an ecoding of several repeated elements is necessary. Unique xml:id can then refer to the copied elements:

```
<section xml:id="stanza2">
   <measure n="32" xml:id="KA2_stage2_mov4_measure32" facs="#KA2_zoneOf_stage2_mov4_measure32"
      copyof="../core.xml#core_mov4_measure5"/>
   <measure n="33" xml:id="KA2_stage2_mov4_measure33" facs="#KA2_zoneOf_stage2_mov4_measure33"
      copyof="../core.xml#core_mov4_measure6"/>
   <measure n="34" xml:id="KA2_stage2_mov4_measure34" facs="#KA2_zoneOf_stage2_mov4_measure34"
      copyof="../core.xml#core_mov4_measure7"/>
  </section>
```

The description of relations between elements by the suggested attributes of *General Relationships* is very limited: while @sameas and @copyof have definite purposes, @corresp has a very broad meaning, it *"marks the correspondence between the current element and one or more other entities"* [3, p.146].

relation

The presentation of relations between measures, notes etc. can be refined by adding different types of a relation. For that, module 3 of the MEI Guidelines [3, p.72-82] offers the possibility to define bibliographic entities and relate them to each other. This FRBR module[2] offers a `relation` element that can describe the relation between the basic entities of the module, this is to say the `work`, `expression`, `manifestation` or `item`. This `relation` element has an `@rel` attribut that offers a lot of different values like `isImitationOf` or `isRealizationOf`, `isPartOf` and many more. [3, p.79]

Because attributes are more flexible and easier to reuse in other projects, and because it is easier to add or modify `attribute` values than to create new `attributes`, my suggestion is to use the `relation` element for all MEI elements that are in a relationship. An important advantage of describing the relation via an own element is that it is not bound to the particular element. To minimize the `markup` within the whole file, the `relation` elements could be collected in a `relationList` in the header:

```
<relationList>
   <relation n="1" origin="#KA2_stage2_mov4_measure23" rel="isImitationOf"
      target="#KA2_mov4_measure23"/>
    …
   <relation n="10" origin="#KA2_stage2_mov4_measure32" rel="isImitationOf"
      target="#KA2_mov4_measure5"/>
    …
   <relation n="37" origin="#KA2_stage2_mov4_measure59" rel="isImitationOf"
      target="#KA2_mov4_measure5 #KA2_stage2_mov4_measure32"/>
    …
  </relationList>
```

The `@target` attribute refers to the element which is the starting point of the imitation (for example). In addition, the `relation` element in the relationlist needs also an `attribute` like `@origin` that shows of which element the relation is part of.

The examples of values for the `relation` element are primary for the needs of FRBR, thus, for bibliographical context. To extend the utilization of `relation` it is necessary to generalize the relations and to bring up values like for example `isCopyOf` for copied elements, `isOctavationOf` or `isTranspositionOf` for musical elements that were transposed, `isCollaparteWith` for elements, especially parts that go colla parte with other parts (in addition to `cpMark`). The value `isBasedOn` could be used for any relationship of similarity, for example when a passage of a musical work is copied and more or less changed. A `@copyof` declaration would be incorrect in this case.

2. Functional Requirements for Bibliographic Records, see [4].

Marks of added pages

The rest of sewing thread on the second, and the tiny stitches of maybe a needle on page 3 of the number prove that there has been a page sewed on the original page that is now lost. Although we cannot know the content of this lost page, this "operation" is very important for the interpretation of the source, its historical context and should therefore be described. The physical appearance of the source could be described verbally in the `physDesc` (physical description) of the `meiHead` and would not be too complex in this case. But verbal descriptions of an operation do not have the advantages of a machine-readable encoding. To do so, one can define all those physical operations that were made in an open list in the `header` to refer to the items of this list in the `body` where it appears. For this `stand-off markup` we add a `classification` element in the `workDesc` that contains a `termist` with `term` elements that describe different physical operations like sewing or pasting a page over another page:

```
<classification>
   <termList>
      <term xml:id="sewn-over" label="sewn-over">A new page is sewn over the old one</term>
      <term xml:id="paste-over" label="paste-over">A new page is pasted over the old one</term>
     </termList>
  </classification>
```

No.13

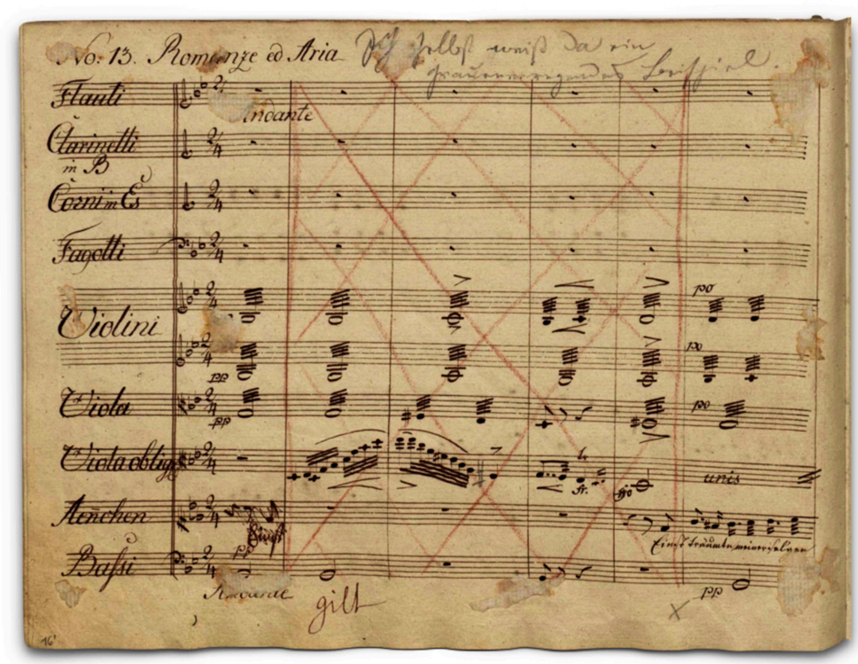



Figure 5: Carl Maria von Weber: Der Freyschütz, Viennese copy, [A-Wn OA.373](), No.13, page 1 (fol. 16r)

Already one the first page of Annchen's Romance one can see that this number experienced multiple stages of changing and editing in Vienna. There is a deletion of four measures in red pencil, a displacement of the upbeat of the vocal part to the beginning, a restoration of the deletion with the words "gilt" and a cancellation of the new upbeat as well as several marks of former glue (see figure 5).

In addition, there are several dialog pages inserted and a double leaf with three newly copied music pages containing a deletion of the two first measures in pencil and glue dots as well as folding marks on several pages in the middle of the number (see figures 6 and 7).

The reason for the deletion in red on the first page is probably the difficult solo viola, since in the new copy the solo is not contained anymore. This is surprising, as we have to suppose that the theatre in Vienna had a problem to find a skilled viola soloist to play this part—at least for some of the performances. Later on, this deletion was restored by the word “gilt” which means “is valid” and the upbeat, which had been copied to the first measure originally, is also deleted. So there was a jump from the first to the 6th bar, which corresponds to the copy-fragment. Marks of glue on this and two other pages indicate that there was an additional page attached by glue that is now lost.

The dialog pages after the first page of the original copy are followed by the new Viennese copy which only contains 9 measures of the number with the mentioned deletion at the beginning. The first page has also traces of glue in the same extent as the deletion. So if we restore the deleted first original page (as indicated by “gilt”), the following bar corresponds to the last one on the first page of the new copy (after the pencil deletion, see figure 6).

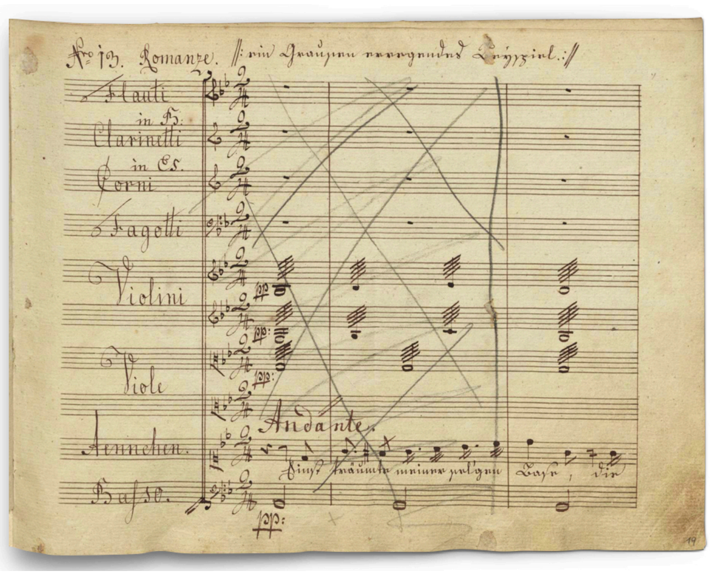



Figure 6: Carl Maria von Weber: Der Freyschütz, Viennese copy, A-Wn OA.373, No.13, fol. 19r

The fragment ends with bar 13 and bar 14 is to be found at the beginning on the verso page of the original manuscript. A possible explanation is, that an empty page was glued or attached to the original page with bars 7 to 13. This is indicated by tiny stitches in the lower right corner of the page. So one could easily jump from bar 13 of the new version to bar 14 of the original version (see figure 8).

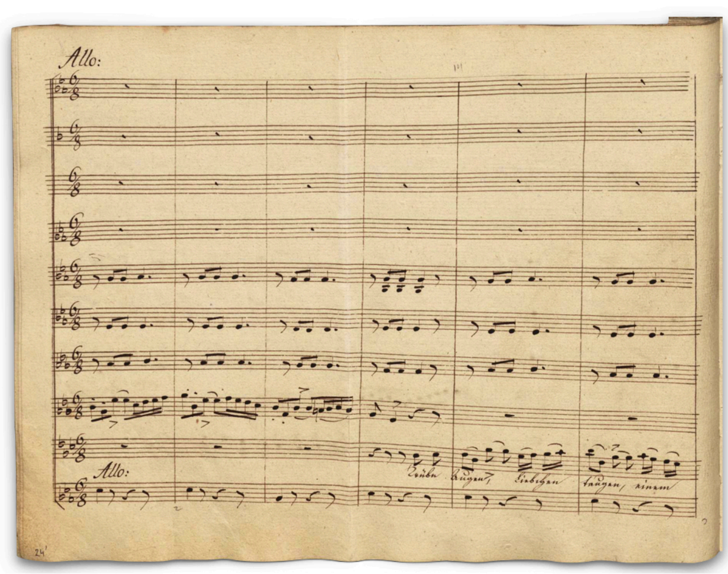



Figure 7: Carl Maria von Weber: Der Freyschütz, Viennese copy, A-Wn OA.373, No.13, fol. 24v, folding mark

From folio 24r running on to the last but one page (fol. 32v), you can see folding marks indicating that these and the following pages were folded vertically. On the preceding page (fol. 23v), the three last bars are deleted in red pencil together with the viola-solo. Also there is an end-mark and a “NB” (which means “nota bene” or “be careful”). It appears that this number should have ended here because the folding of the following pages was a usual way to abbreviate documents. A letter from Weber to his wife proves this assumption when he describes a performance of the *Freischütz* in Vienna while he visited the city: *Die große Arie. das Gebet schnell, und alles übereilt, aber doch nicht ohne Ausdruk. das Terzett auch so holterpolter. Die Wolf:schl: nun -- zusammen gefegt. aber allerley hübsches in Dekorationen. Die Cavatine der Agathe, das einzige was ganz gut war. Romanze, ohne Bratsche. und bei Nero, aus. Finale -- Kopf und Schwanz.* [1] He describes that in this performance the romance was played without the viola and ended at the word "Nero". This is exactly the point where we can find the end-mark in red. So it is sure that this number was played without the following recitative and aria.

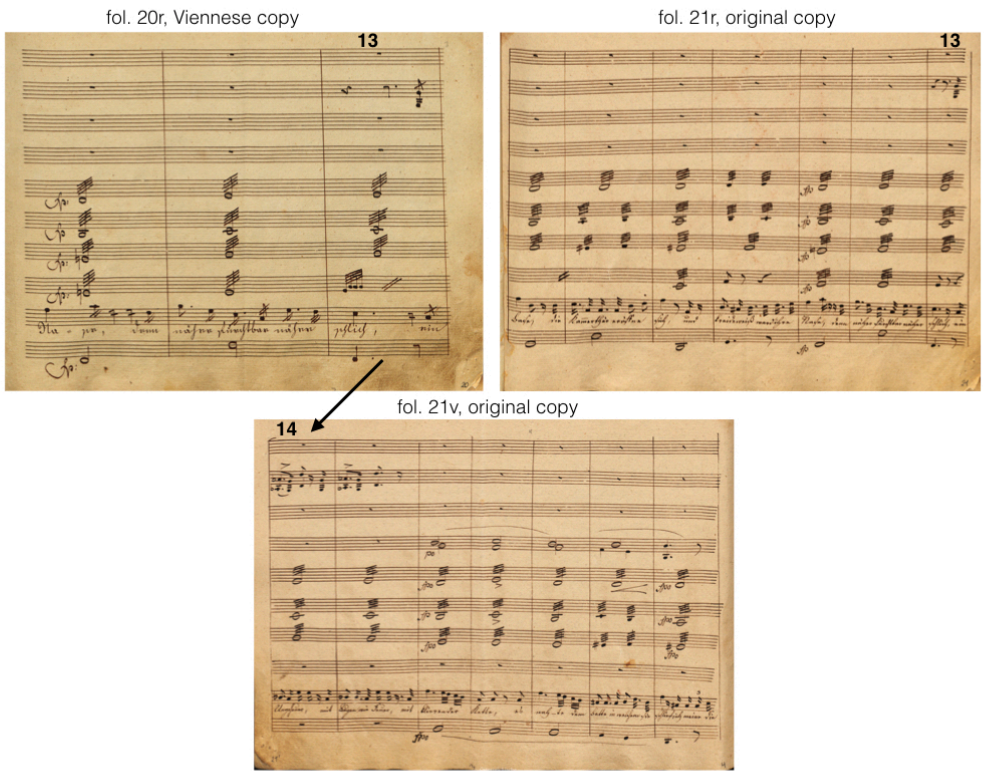


Figure 8: Carl Maria von Weber: Der Freyschütz, Viennese copy, [A-Wn OA.373](), No.13, fol. 20r – 21v

Encoding

The folding of several pages in this number is a special encoding problem. Because the folding mark is basically a feature of the material itself, that is the page, one could describe the folding in an `annot` element with the `type` "pagefold" and a verbal description. The `annot` element is integrated in a `pgDesc` (page description) element, that is nested in a `pb` (page break) element:

```
<pb label="f.23v">
   <pgDesc>
      <annot plist="Pf1" type="pageFold">This page features a folding mark</annot>
     </pgDesc>
  </pb>
<measure n="41">…</measure>
```

This solution is problematic because it only describes an occurence on the physical, graphical level, it does not describe anything on the logical, textual level. But the folding is obviously an abridgement of the number and should thus also be encoded.

The "deleted" measures of the folded page could be nested in a del element with @type attribute that refines the way of the deletion with the value "pageFold":

```
<del type="pageFold">
   <measure n="65">…</measure>
   <measure n="66">…</measure>
   <measure n="67">…</measure>
   <measure n="68">…</measure>
   <measure n="69">…</measure>
   <measure n="70">…</measure>
   <measure n="71">…</measure>
  </del>
```

It is also possible to describe the act of folding in a term element in the classification with a reference to the term on the particular point with a @decls attribute. For this, the measures should be nested in section elements:

```
<section label="f.26" decls="#pageFold">
   <measure n="72">…</measure>
   <measure n="73">…</measure>
   <measure n="74">…</measure>
   <measure n="75">…</measure>
   <measure n="76">…</measure>
   <measure n="77">…</measure>
   <measure n="78">…</measure>
  </section>
```

In addition, the measure elements in the section can also be surrounded by a del element to indicate that the folding is accompanied by a truncation.

These three approaches are very different from each other. annot can only describe the folding itself but in fact, in the encoding, the page is still existent. The same problem appears when using @decls but the operation can be described verbally in the classification and the term elements can also be described in an extern file to use them for other sources aswell. Nevertheless, the focus here is on the description of the manuscript and less on the logical content. The approach using del in combination with @type could serve the manusript level as well as the logical, textual level. In this case it depends on what the user/editor wants to explain and encode.

A few of the encoding solutions mentioned before combine the description of the musical content with the description of the manuscript in the body that contradicts the principles of the encoding with MEI. My suggestion is to separate the description of changes of the manuscript itself like the addition or removal of pages, foldings or the rearrangement of pages, from the description of the musical text. For this purpose, the project *Beethovens Werkstatt* uses an element describing the arrangement of the pages of a source (foliumSetup) which contains the order of the physical pages. The element is nested within the physDesc in the sourceDesc. The element folium means a single page (= recto and verso) and bifolium is a double page (=two pages with recto and verso). Like the real manuscript that consists of double pages which are nested into each other, the `bifolium` element can contain other `bifolium` or `folium` elements so that the original structure of the manuscript can be reproduced.

With their attributes `@recto` and `@verso` respectively `@outer.recto`, `@inner.verso`, `@inner.recto` and `@outer.verso` the `folium` and `bifolium`-elements can refer to the relevant `surface` elements that are nested in `facsimile`. Figure 8 shows the `foliumSetup` of No.13. The pages foliated 16v, 21r and 21v are described by a `bifolium` element that contains the dialog pages and the new Viennese copy (surface 451-453). The Viennese copy was added later, so it is described by an `add` element. The glue dots on the first page of this Viennese copy (surface 451 in the digital edition) and on the first page of No.13 show that there were sheets of paper. These sheets are described by a `patch` element, that was at first added (`add`) and later removed (`del`).

The textual and physical changes in the manuscript make it possible to discover previous states of the source. On the basis of the TEI Enconding Model for Genetic Editions [2], *Beethoven Werkstatt* developed a model where a group of elements allow a formal description of the textual and physical genesis of a source. With this `genDesc` (`genetic description`) it is possible to add a `@changeState` attribute to every modification (like `add` or `del`) in the `body` or in the `foliumSetup`. With this, every change or multiple changes can be associated with a unique state. These states are collected in the `genDesc` in the `meiHead` (through `work` > `history` > `creation`). In `genDesc` it is possible to describe the order of the states with `@ordered="true"/"false"`.

```
<foliumSetup>
   <bifolium inner.verso="#KA2_surface446" inner.recto="#KA2_surface455" outerVerso="#KA2_surface456">
      <add changeState="#X">
         <patch xml:id="Blatt1" attachement="glued" attached.to="innerVerso"/>
        </add>
      <del changeState="#X1">
         <patch xml:id="Blatt1"/>
        </del>
      <folium recto="#KA2_surface447" verso="#KA2_surface448"/>
      <folium recto="#KA2_surface449" verso="#KA2_surface450"/>
      <add>
         <folium recto="#KA2_surface451" verso="#KA2_surface452">
            <add changeState="Y">
               <patch xml:id="Blatt2" attachement="glued" attached.to="recto"/>
              </add>
            <del changeState="Y1">
               <patch xml:id="Blatt1"/>
              </del>
           </folium>
         <folium recto="#KA2_surface453" verso="#KA2_surface454"/>
        </add>
     </bifolium>
  </foliumSetup>
```

No.13, `foliumSetup` and `@changeState`: a patch was glued on the page in state x while it was removed in state x1.

```
<history>
   <creation>
      <genDesc ordered="true">
         <state xml:id="X" next="X1"/>
         <state xml:id="X1" prev="X"/>
        </genDesc>
     </creation>
  </history>
```

No.13, the states and their order are described in the `genDesc`.

Conclusion

To summarize my thoughts, I prefer a separation of the encoding of the content and the physical modifications of the manuscript for a better understanding of the modifications and detailed description of the genesis of the source if the changes are as complex as we saw in No.13. Even if this is not a genetic encoding of the source as in *Beethoven Werkstatt* one could apply the `foliumSetup` element to better describe manuscripts in their physical construction and modification. Like always with MEI, it depends either on the encoding targets for the musical content of the manuscript, or on the document itself. For instance, it could be sufficient for No.4 to describe the textual and physical changes in the body. There is no single right answer on how to encode physical changes and complex relations between elements, it all depends on the available source and the user's needs.

Works cited

[1] Complete Works of Carl Maria von Weber. Digital Edition, http://weber-gesamtausgabe.de/A041911 (Version 3.4.2 of February 8, 2019). Last change of this document on December 8, 2018.

[2] Jannidis, Fotis et al., An Encoding Model for Genetic Editions, URL: www.tei-c.org/Activities/Council/Working/tcw19.html

[3] Music Encoding Initiative. Guidelines, v.3 (www.music-encoding.org/docs/MEI_Guidelines_v3.0.0.pdf.

[4] Teich Geertinger, Axel and Kristina Richts "Implementing the Functional Requirements for Bibliographic Records (FRBR) model in MEI", in: *Music Encoding Conference. Proceedings 2013-2014*, ed. by Perry D. Roland and Johannes Kepper. (Mainz-Munich: Music Encoding Initiative - Bayerische Staatsbibliotek, 2015): 17-26. Online resource: langzeitarchivierung.bib-bvb.de/delivery/DeliveryManagerServlet?dps_pid=IE2860815 also reachable at music-encoding.org/archive.

XML Music Performance Description: Reflections on and Future Developments of the Music Performance Markup Format

Axel Berndt
Center of Music and Film Informatics, University of Music Detmold, Detmold, Germany
berndt@hfm-detmold.de

Benjamin W. Bohl
Institut für Musikwissenschaft, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt am Main, Germany
bohl@em.uni-frankfurt.de

Abstract

This paper introduces the `Music Performance Markup` format for encoding expressive performance parameters. The main focus is the general data organization in global vs. local data and header vs. dated data, illustrated by some musical examples. Moreover future work and use cases are envisioned.

Motivation and related work

Haus and Longari [6] distinguish six layers of music information: general, structural, music logic, notation, performance, and audio. Regarding the performance layer, most music formats provide either very technical information (e.g., MIDI) or very rough information (MusicXML, MEI). While the technical information is rich in details, precise and unambiguous it lacks a musically meaningful abstraction. Abstract representations, on the other hand, are easy to read and write into music scores but are ambiguous and incomplete. E.g.: how loud is "forte"? How fast is "allegro"?

Throughout the past two decades, several research projects contributed to answering such questions in order to bridge the gap between abstract information and technical implementation. While analytical projects investigated the phenomena of music performance and their shaping by human musicians [7], generative projects developed systems that (semi-)automatically create expressive performances (see [8], [10], [4], [5]). The data models to describe these performances were rarely a matter of deeper discussions. Hence, they never reached any relevance beyond the scope of their corresponding systems.

The XML-based Music Performance Markup (MPM) model for performance description initially had been developed for exclusive usage within a specific performance rendering engine [1]. Since their publication however, the format and the engine have been used in several different contexts, e.g. research on video.

General concept

An expressive performance of a piece of music transforms the musical raw material (typically notes) into sounding output. One piece of music can be performed in many different ways. In our approach this is reflected by—as in SMDL [9]—generally separating logical and gestural domain data. While a MIDI file provides the musical “raw material” (and should not contain anything else but the note-on and note-off events) a corresponding MPM file provides the description of the performative aspects. Our software implements the techniques to render this information into expressive MIDI sequences that are output directly or sent to a *Digital Audio Workstation* for music production.

The MPM format is based on the formal models that were introduced by Berndt [2], and thus allows descriptions of rich, nuanced music performances that reach down to a variety of subtle details. These cover a wide range of performance phenomena, including tempo (macro timing) and dynamics curves (macro dynamics), articulations (micro dynamics and micro timing), metrical accentuation (micro dynamics), asynchrony, rubati, and quasi-random imprecision (micro timing). The models are based on a series of studies, experiments with musicians, and measurements of music recordings. From the observed characteristics we deduced mathematical functions and developed a parameterization that is meaningful in the musical context and provides access to the full range of variability. For example, continuous tempo transitions, i.e. accelerando and ritardando, can feature different shapes depending on various factors such as the musical context or the abilities of the musicians The tempo model provides all these shapes of tempo curves so that they can be used to recreate existing performances or simulate hypothetic performances.

Furthermore, the format defines global information, applying to all musical parts, and local information, applying to single parts. Technically, the parts correspond with MIDI channels. Local information may be defined for each part individually, and if present, local information dominates the global. This pays off e.g. in solo-plus-accompaniment constellations. Global performance instructions are executed by the whole ensemble; only the solo part features its own local instructions that “overwrite” the global.

Both, global and local information, are subdivided into header and dated information. Header information applies to the whole movement, e.g. the definition of articulation styles and MIDI controller mappings. Dated information is organized in sequential lists, so-called *maps*, one for each type of performance features. Each element in a map demands a date attribute (in MIDI ticks). Discrete performance features, e.g. articulation instructions, are applied at the respective time position in the MIDI sequence. Most performance features, however, extend over a certain timeframe. A dynamics instruction, like *forte* for instance, lasts from its date until the date of a succeeding instruction in the same dynamics map. Other penomena like rubato instructions or metrical accentuations define a certain timeframe of their application—in case of metrical accentuations typically one measure. The schemes are then repeatedly applied (e.g., measure-wise) until the succeeding instruction in the same map defines a new scheme and frame length.

The tempo and dynamics models define macro curves, i.e. absolute timing and loudness curves with no other details than rough ritardandi, accelerandi, crescendi, and decrescendi, etc. All other models are called micro features. They are defined as relative features, i.e. they are mathematically added onto the macro curves and, thereby, introduce rich details to the performance description. One requirement to the model design of micro features was self-containedness. Even though many features affect the same domain (e.g. the timing domain is addressed by the micro features rubato, random imprecision and asynchrony) the features should be self-contained, i.e., give the user independent control. E.g., editing a Viennese waltz timing is achieved only via rubato, other feature types do not interfere with it and it is not necessary to balance rubato and several other features to achieve the desired rubato timing. This self-containedness is not only of practical value when defining and editing music performances. It is also relevant to analytical applications as it provides guidance for the analytical decomposition of complex timing and dynamics curves.

The software is also capable of rendering seamless transitions between different performance styles, which is, for instance, used for adaptive game scoring. In a listener study we used this interactivity for an analysis-by-synthesis approach where the participants could adjust certain performance parameters according to the tasks they were given [3].

Reflections and perspectives

The interplay of global and local information is the key to complex polyphonic performance structures. Each part can feature its own performance plan. One part may perform a decrescendo while another part plays a crescendo. One part may perform a swing timing, another an even timing, while they all follow the same basic tempo. A key feature of our models to allow such multifarious performance representations is the self-containedness of each single model. This gives precise and independent control of each performance detail without any distorting interference from another detail. It is possible, for instance, to change the macro timing (tempo curve) completely while keeping all micro timing features unaltered.

But where do these performance details come from? The performance instructions that the format conveys are explicit and numerically precise. For instance, a dynamics instruction *forte* corresponds with a clearly defined numeric value, even if each part defines a different such value.[1] The models underlying the format act as interfaces, the user adjusts their parameters. However, the creation of a fully fledged performance is, nonetheless, an excessive amount of work. This must be supported by editing tools, such as music notation software plug-ins. Reduced editing workload can also be achieved by semiautomatic approaches. Rough performances may be generated automatically. The human editor then corrects and incorporates further details. The analysis of preexistent performances is another way to generate a detailed performance in MPM. Ultimately, this makes the computer play the music as the human musician did. Such formalized human music performances might become a subject of closer analysis to learn more about performance styles, e.g. in the context of historically informed performance practice and its research.

The format's close connection to the MIDI standard reflects the direct coupling to music production as the core application domain. This may imply some disadvantages such as bad readability. A better compatibility with other formats also demands a flexible numeric basis of the attributes, for instance by user-defined reference systems. We therefore decided to make the timing, dynamics and pitch domain floating point with the next development iteration. Although this precision might be lost when the performance is rendered to an expressive MIDI sequence, other (for instance analytical) application contexts can benefit from higher numeric resolution and accuracy. To break the reliance on the MIDI standard even further, we will generalize the relation to musical score data. Via converters we plan to add support for formats such as MusicXML and MEI.

Currently, MPM undergoes a drastic redesign addressing issues of generalization and cross-format references (e.g. between MPM and score data). We plan to extend the library of models in many respects. For instance, delay and (de-)tuning are added to the articulation model. Random imprecision—thus far available only on the timing axis—are added to the dynamics and articulation domain. Several alternative imprecision models are offered, from a simple uniform distribution, Gaussian and triangular distribution over correlated noise types such as pink noise up to an imprecision table that can be created from measurements of human performances.

1. The *forte* of, e.g., a flute can be defined different to that of a trumpet.

Further potential features include playing techniques (e.g. fingering, breathing), ornamentations, ensemble communication and how it is influenced by seating order and acoustics. A formal representation of a more general "aesthetic agenda" could capture specifics of a certain musician, school or musical era and might be used to generate expressive performances in an automatic or semi-automatic fashion.

The redesign will also make a redevelopment of the rendering engine necessary which we plan to add to the *meico* converter framework.[2] With this we hope to support interaction of the two musicological disciplines performance research and (digital) music edition. We see further potential applications of MPM in computer-based music production and interactive media scoring [1].

Works cited

[1] Berndt, Axel. *Musik für interaktive Medien: Arrangement- und Interpretationstechniken*. Munich: Hut, 2011.

[2] Berndt, Axel. "Formalizing Expressive Music Performance Phenomena" in *Works in Audio and Music Technology*. Dresden: TUDpress, 2015, 7-128.

[3] Berndt, Axel and Hähnel, Tilo. "Studying Music Performance and Perception via Interaction" in *Works in Audio and Music Technology*. Dresden: TUDpress, 2015, 129-53.

[4] Flossmann, Sebastian, et al. "Expressive Performance Rendering: Introducing Performance Context" in *Proceedings of the 6th Sound and Music Computing Conference (SMC). Porto, Portugal, 23-25 July 2009*, 155-60 (http://smc2009.smcnetwork.org/proceedings)

[5] Friberg, Andreas et al. "Overview of the KTH Rule System for Musical Performance" in *Advances in Cognitive Psychology* 2.2-3 (2006): 145-61.

[6] Haus, Goffredo and Longari, Maurizio. "A Multi-Layered, Time-Based Music Description Approach Based on XML" *Computer Music Journal* 29.1 (2005): 70-85.

[7] Hähnel, Tilo. *Baroque Performance: a Research Study on Characteristic Parameters of 18th-Century Music*. Studies in Cognitive Musicology. Osnabrück: Electronic Publishing Osnabrück, 2013.

[8] Mazzola, Guerino. *The Topos of Music: Geometric Logic of Concepts, Theory, and Performance*, in collaboration with Stefan Göller and Stefan Müller. Basel: Birkhäuser, 2002.

[9] Newcomb, Steven R. "Standards: Standard Music Description Language Complies with Hypermedia Standard" *Computer* 24.7 (1991): 76-79.

[10] Widmer, Gerhard, and Werner Goebl. "Computational Models of Expressive Music Performance: The State of the Art" *Journal of New Music Research* 33.3 (2004): 203-16.

2. https://github.com/cemfi/meico, accessed 1 September 2017.

Wie? Was? Entsetzen! Lessons learned from the Freischütz Digital project

Johannes Kepper
Beethovens Werkstatt
kepper@edirom.de

Abstract

The Freischütz Digital project (FreiDi) was one of the pioneer projects employing MEI in large scale. It did not only try to encode a huge quantity of music material, it also sought to capture as many aspects of the available sources as possible, effectively creating data of almost unrivaled richness. This paper discusses the outcomes of and experiences made in the FreiDi project.[1]

Introduction

When the FreiDi project started in 2012, there was no significant number of real-world MEI encodings available. Most encodings at that time were part of the Sample Collection[2], which is an artificial compilation of hand-crafted files, intended to show-case different aspects of the MEI format. No major musical work of the classical period had been completely encoded with MEI so far, so that the proof of MEI's feasibility and applicability for such quantities was still outstanding. Likewise, Frans Wiering's *Multidimensional Model* [1] had never before been tested in a real-world scenario.

In this situation, Freischütz Digital was initiated to showcase the full potential of a genuinely digital edition. As the printed edition of Carl Maria von Weber's *Freischütz* was scheduled for the Complete Works edition [2], it was an obvious decision to utilize this most prominent work by Weber and share at least the work for preparing materials. Since the *Freischütz* was extremely successful, Weber sold dozens of manuscript copies to theaters across Europe. He listed more than forty such copies, including dates, prices and the venues to which they've been sold. This means that there were some forty-plus copies that were authorized by the composer. Out of the eight surviving manuscripts we know so far, five carry at least some autograph corrections by Weber. A full autograph, which served as master for most of these copies, also survived. Finally, we considered a 1849 print of the score for our project. As the *Freischütz* has about 3100 measures, and we looked at a collection of ten relevant sources, it was clear that this would easily exceed anything done with MEI so far.

At the same time, the scope of FreiDi was not restricted to the music edition. Parallel to that, we prepared a genetic edition of Friedrich Kind's libretto for Weber's opera, and we tried to trace the development of the underlying subject matter through predecessing texts, which we called “reference texts”. While both reference texts and libretto edition are supposed to help describe the establishment of the musical text, we also

1. While this paper has a single author only, it is based on input from the whole project team. The author would like to thank his colleagues, most notably Benjamin W. Bohl, Daniel Röwenstrunk, Joachim Veit and Raffaele Viglianti.
2. The Sample Collection is now available at https://github.com/music-encoding/sample-encodings.

continued from there on to actual performances of the *Freischütz*. We managed to secure rights to include audio files from some well-known recordings of this work, and we also made our own recording of three numbers. The intention here was to allow a comparison between different performances, and to explore novel approaches in audio-processing informed by encodings of the performed musical text. In summary, the scope of FreiDi ranged from the first ideas on the subject matter topic to specific features of actual performances of the completed work. Doing so, we dealt with material of sufficient diversity to trial Wiering's concept of a multi-dimensional model and to identify the challenges of a genuine digital edition. This paper will focus on the preparation of the music edition, as this aspect may inform future editions and recommend best practices.

The plan

Our original intention was to fully encode the autograph, all eight existing copies mentioned above, and at least one copy of the first prints in score and for piano. Each source was supposed to be encoded faithfully to its appearance – not with full diplomatic precision, but still reflecting all typographical features. This means, we wanted to be able to trace in our encodings where and when two woodwinds on one staff were written with shared stems (i.e. as chords), or as two separate layers with stems in opposite directions. We wanted to be able to see where and when Weber used "shortcuts" and abbreviations to repeat the preceding measure, another staff, or something completely different, perhaps in a different octave. We wanted to know how the different copyists responded to that – either by resolving those abbreviations or keeping them in their respective versions. In essence, we wanted to capture and preserve relevant features of the specific scripture of our sources.

When FreiDi started, there was no real-world experience for generating, proofreading and correcting such quantities of MEI data. MEI already provided a sufficiently robust XSLT to convert MusicXML files to MEI.[3] However, there was no satisfactory way to display MEI files at that time – today's omnipresent Verovio[4] was not introduced until two years later. Accordingly, we had to come up with a reasonable plan on how to model and generate our dataset. As mentioned earlier, the WeGA was preparing the text for their printed edition at the same time. This allowed us to share the task of transcribing the score into digital media. Using existing expertise, we created and proofread transcriptions of the autograph with Finale, which where then exported to MusicXML. From there on, we could use the XSLT mentioned above to convert these files to MEI. That path proved perfectly reliable for our purposes. However, the files created reflected the layout of our Finale transcriptions, which obviously differed from the layout of the autograph or any of the other sources we where interested in. Accordingly, we used a custom XSLT to remove all layout-related information from our MEI files. At this point, the data reflected the pure text of the autograph, but other than the WeGA, we where also interested in its scripture. At the same time, we wanted to capture both content and scripture of the other relevant sources.[5] This turned out to be one of the most challenging and momentous requirements of Freischütz Digital.

3. The most recent version of this XSLT can be found at https://github.com/music-encoding/encoding-tools/blob/master/musicxml2mei/musicxml2mei-3.0.xsl. Most other XSLT mentioned in this article are also available from the Freischütz Digital / Tools repository at GitHub. However, they are closely tied to the workflows of the project, and may not be immediately useful in other contexts. The author will happily support others to make these scripts more widely applicable.
4. See https://verovio.org
5. As it turned out, the textual content of all sources was almost identical – there where almost no substantial differences of the primary text, i.e. pitches and durations.

Modelling differences: the Core Model

In MEI, the predominant approach to encoding variation is the so-called `parallel segmentation`[6] paradigm. This means that content unaltered between multiple documents / texts is encoded only once, reflecting the commonalities between those texts. Whenever relevant differences occur, that “common stream of code” is suspended for the moment, and multiple differing segments are provided in parallel, each of which representing the content of one or more sources. MEI (just like TEI) uses the <app> (apparatus) element to indicate such a section, which then contains multiple <rdg> elements. These readings utilise a @source attribute to refer to the source(s) whose text they encode.

```
<note pname="c" oct="4"/>
<note pname="d" oct="4"/>
<app>
   <rdg source="#A">
      <note pname="e" oct="4"/>
   </rdg>
   <rdg source="#B #C">
      <note pname="e" accid="f" oct="4"/>
   </rdg>
</app>
<note pname="f" oct="4"/>
<note pname="g" oct="4"/>
```

Figure 1: The (fictional) sources **A**, **B** and **C** share the first and last two notes, but differ in the third: **A** reads an E4, while both **B** and **C** feature an Eb4.

Such <rdg> elements may also be empty to indicate the fact that some content is available only in some of the encoded sources. In the Freischütz material, this was a frequent case with slurs and other “secondary” aspects of the notation.

In the example above, all but the most relevant information was omitted for clarity. It should be obvious that considering more aspects of the notation will result in a significantly higher likeliness of differences between the sources, and will thus increase the number of <app> elements – as will the number of sources considered. As we were interested in preserving scriptural details, it was clear that following this approach would lead to highly cluttered encodings. Extracting information from such fragmented files becomes overly complicated: relevant details are easily hidden between trivialities. As a consequence, we sought methods to distinguish between the textual content and it's scriptural manifestations. Our answer was the `Core Model`, which facilitates this separation of concerns at the cost of a significantly increased complexity.

The idea of the Core Model is to split the information about every single source up into two separate files, with links connecting the information back into a bigger “virtual” file. In one document, called `core`, only information relevant for the text is contained – pitches, durations, but also slurs, dynamics etc. The other document, called `source`, completely avoids that information, and only provides scriptural details such as abbreviations used (see below), curve and stem directions. The source also holds pointers to the core file. Every note etc. is therefore encoded twice, but every information is given only once.

6. See [TEI Guidelines, chapter 12.2.3: The Parallel Segmentation Method](#).

```
<-- core.xml file -->
<note xml:id="core-note1" pname="g" oct="4" dur="4"/>
<-- source.xml file -->
<note xml:id="source-note1" sameas="core.xml#core-note1" stem.dir="down"/>
```

Figure 2: In this encoding, four bits of information about a given note are split between two files: While pitch name, octave placement and duration are given in *core.xml*, the stem direction is provided in *source.xml*. The latter also holds a reference to the note in the first file in its @sameas attribute.

Considering that Freischütz Digital used UUIDs (such as “b1182026-4e75-49b9-b976-de3ab6a62624”) as XML:IDs[7], it is evident that the Core Model is not suitable for manual maintenance: The final dataset holds almost half a million of these @sameas references, each based on an absolutely incomprehensible XML:ID. It is thus costly in that it requires automated processes to ensure consistency of the data and avoid the potentially fatal breaking of links through manual interventions.

```
<-- core.xml file -->
<note xml:id="core-note1" pname="g" oct="4" dur="4"/>
<app>
   <rdg source="#A">
      <note xml:id="core-note2a" pname="b" oct="4" dur="4"/>
   </rdg>
   <rdg source="#B #C">
      <note xml:id="core-note2b" pname="c" oct="5" dur="4"/>
   </rdg>
</app>
<note xml:id="core-note3" pname="d" oct="5" dur="4"/>
<-- source A file -->
<note xml:id="a1" sameas="core.xml#core-note1" stem.dir="up"/>
<note xml:id="a2" sameas="core.xml#core-note2a" stem.dir="down"/>
<note xml:id="a3" sameas="core.xml#core-note3" stem.dir="down"/>
<-- source B file -->
<note xml:id="b1" sameas="core.xml#core-note1" stem.dir="up"/>
<note xml:id="b2" sameas="core.xml#core-note2b" stem.dir="down"/>
<note xml:id="b3" sameas="core.xml#core-note3" stem.dir="down"/>
<-- source C file -->
<note xml:id="c1" sameas="core.xml#core-note1" stem.dir="up"/>
<note xml:id="c2" sameas="core.xml#core-note2b" stem.dir="up"/>
<note xml:id="c3" sameas="core.xml#core-note3" stem.dir="down"/>
```

Figure 3: Source **A** reads *G4*, *B4*, *D5*, while **B** and **C** read *G4*, *C5*, *C5*. This difference is immediately available from the core file. The fact that **B** and **C** have a different stem direction does not obfuscate the core file, but can still be derived from a comparison of these two files.

7. For selected structural elements like movements, measures and staves, however, FreiDi used predictable IDs like “core_mov6_measure12” to facilitate better manual orientation and navigation in the files.

While the Core Model has an obvious and undoubtedly large impact on the complexity of the data, it facilitates a two-level approach to variation between source. Differences on the level of primary aspects of the notation can still be encoded using the "parallel segmentation" mentioned above, that is by utilizing <app> and <rdg> in the core file, while minor differences on the scriptural level will not trigger such an encoding. If interested in a "full" encoding of a given source, it is possible to parse its source file, follow every link to the corresponding notes etc. in the core file, and copy the information provided there into the source file.

The Core Model allows to focus the encoding and promote some differences over others. However, it depends on a comparable text. As we were interested in preserving the scriptural specifics of the different manuscripts, it was necessary to implement some level of normalization that would still allow to access the original form. The main phenomenon to consider were abbreviations like measure or beat repetitions. For that, MEI provides the <choice> element and its childs <abbr> (abbreviation) and <expan> (expansion). These elements are used inside our source encodings: Here, we transcribe both the original, abbreviated notation, and the expanded text. Only the latter is then linked to the core file. That way, two (or more) manuscripts with different notational shortcuts can still be compared in a consistent way.

```
  <-- core.xml file -->
  <note xml:id="core-note1" pname="c" oct="4" dur="4"/>
  <note xml:id="core-note2" pname="c" oct="4" dur="4"/>
  <-- source.xml file -->
  <note xml:id="source-note1" sameas="core.xml#core-note1" stem.dir="up"/>
  <choice>
     <abbr>
        <beatRpt/>
   </abbr>
     <expan>
        <note xml:id="source-note2" sameas="core.xml#core-note2"/>
   </expan>
</choice>
```

Figure 4: The source reads a note, followed by a typical beat repeat. The meaning of this shortcut is resolved by another <note> element. Please note that this element does not provide information about the stem direction – since the note is not written down in the source, no such statement can be made in the encoding.

The Core Model definitely imposes considerable requirements on an implementation (see below). However, the complexity of the model is a direct result of the complexity of its object: It seems unlikely that there are significantly simpler ways to model the information addressed by the Core Model. It is still based on the parallel segmentation approach, but introduces a "sweet spot", where some differences are more accessible or easier to process than others. This "sweet spot" is adjustable to different needs by deciding which information goes into the core, and which information stays in the sources. The concept is even applicable to other types of material.[8] In summary, the Core Model offers a specific approach to data modelling. It is certainly not the ideal solution for all types of use cases, but has proven itself extremely helpful in the context of FreiDi.

8. Even within FreiDi, the team adopted it for the TEI-based encoding of the Libretto; see [3].

The data preparation workflow

As mentioned earlier, we had raw transcriptions of the autograph, converted to regular MEI, at a very early stage of the project. However, as those transcriptions were also used for the WeGA, all abbreviations used by Carl Maria von Weber had been resolved due to the need for a complete edited text, while we were specifically interested in such features of the script. We therefore implemented a workflow to re-insert those features not only for the autograph, but also all other sources considered.[9] Eventually, that workflow was supposed to produce data conforming to our Core Model.

With our layout-free MEI encodings for each source, the next step was to include the measure positions we had generated with the old Edirom Editor software.[10] The only challenge here was to make sure the measure count in the facsimiles matched the encodings we had, so that a simple matching by index (per movement) was possible. In some sources, repetitions had been written our, or individual sections were deleted for various reasons[11], but other than that, this was a straightforward task. Our student assistants had drawn rectangular zones around each measure, and those zones were now associated with our measure elements. However, as we already knew the number of staves in each measure, on top of these measure zones, we were able to generate approximate values for every staff by dividing the height of the measure zones and including some overlaps. On some pages with skewed staff lines, those derived staff zones are not always perfectly accurate, but in general, they fit the manuscript very well. With tolerable precision losses, we were able to automatically increase the amount of data by a factor of about twelf.

The next step was to initialize the Core Model. With a semi-generated concordance file, we created a preliminary core file, which at this point only contained markup down to the staff level, but no musical content. However, it's only purpose at that point was to provide anchors that the individual source files could refer to. With those links established, it was later possible to clearly identify related materials that had to be compared and integrated into the full core. At this point, we were able to start the actual proofreading process.

It was obvious that proofreading large quantities of data was not possible based on manual inspection of the MEI files. At the same time, there was no way to render those MEI files into scores – this was in 2013, and Verovio, which is now the standard for rendering MEI, was not introduced until 2014. We therefore engineered a two-part proofreading workflow, which was based on individual files for each page of our sources. This separation left us with smaller units to deal with which also allowed better tracking of progress.[12] Then, each of those files was first proofread regarding MEI's *events* (notes, rests, chords, etc.), and after that regarding `controlevents` (slurs, dynamic markings, directions, etc.). For both steps, we created dedicated tools that greatly facilitated the task.

9. For the other sources, we just started with copies of our initial autograph-based encoding. Given the small number of substantial differences between all sources, and the fact that in almost all cases we only had to replace fully encoded material with spaces (which require much less elements), this was a sensible decision. However, with other, potentially more diverse material, other workflows might be more suitable.
10. The Edirom project (https://edirom.de/edirom-projekt/) was a research project funded by the German Research Foundation (DFG) from 2006 to 2012. It tried to develop generic tools for digital scholarly editions of music. The publication software is used in various digital editions, like the Reger-Werke-Ausgabe or the OPERA project. Most Detmold-based staff of the FreiDi project formerly worked for the Edirom project.
11. See Agnes Seipelt's article about the Viennese version in this volume.
12. The biggest challenge of this separation was to generate <scoreDef> elements at the start of each page that matched the current state of each staff, i.e. that factored in clefs that might have changed on preceding pages, but also altered meters or keys.

The first tool, internally called the `PitchTool`, was supposed to operate on a staff by staff basis, and combine a facsimile excerpt of that staff with a rendered transcription of that staff plus the underlying MEI in an XML editor. With the staff zones auto-generated from the measure zones, it was possible to calculate a bounding box around the complete staff, und request that portion of the image from an image server. Getting a rendition of the staff was much more complicated, given that there was no renderer for MEI yet. Since we were aiming for a web-based approach, we investigated options for other encoding standards than MEI. Eventually, we realized that ABC.js, a javascript-based editor for the ABC notation standard, could be instrumental for our project.[13]It uses a simple textarea where users could insert ABC code and see their changes immediately rendered into a score. In our setup, we changed that setup significantly. For us, the correct rendition of the code was less important than the MEI code itself, so it was no option to utilize the ABC notation as an interface for our users. Instead, we added a javascript-based XML editor, in which we inserted a version of the page's MEI file which only contained the current staff, so that the proofreading user would not be distracted by MEI code not relevant for her or his current task. Changes made in that XML editor where validated on our server, and at the same time converted into ABC code by a custom-built XSLT. This ABC code was inserted into the (invisible) textarea of our ABC editor, and from there rendered into a score image we could display on our page. Through a very sophisticated setup it was also possible to click on a note in that rendered score and move the cursor to the MEI code of that note in the XML editor. Whenever the user saved his changes, the modified MEI was uploaded to our server, and through an XSLT that corrected staff was re-inserted into the file for the complete page, alongside a <change> providing information on who worked on when on which staff.[14] In this step of our workflow, we removed all notation that was actually abbreviated in the source at hand, and replaced it with the correct abbreviations, mostly <mRpt>, <beatRpt>, <space>, and <cpMark>.[15] With this setup, it became very easy to compare the facsimile and a rendering of our MEI code, which made it possible to carefully proofread large quantities of encoded

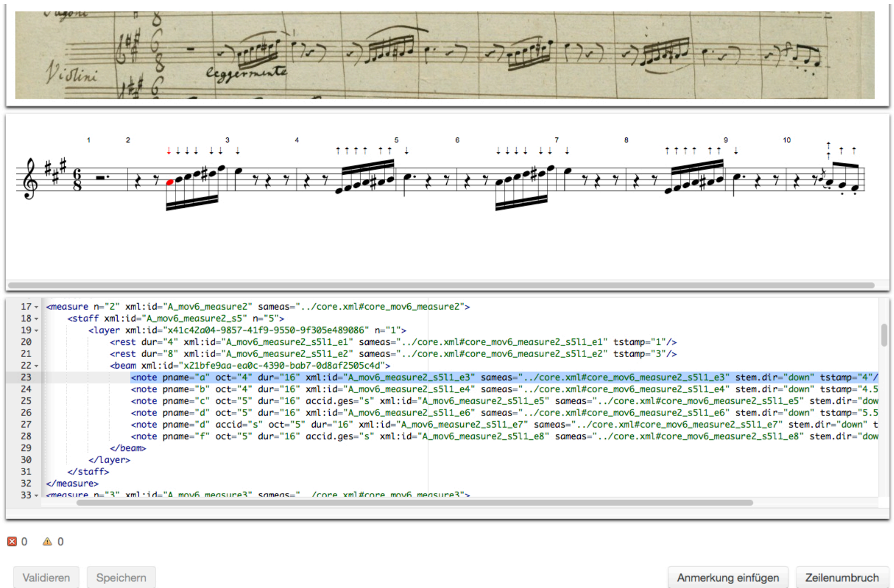


Figure 5: The PitchTool used for proofreading notes, chords, rests and other MEI "events".

13. ABC is a much simpler encoding format for music notation, mostly targeting at single-voice song collections. See https://abcjs.net/abcjs-editor.html and http://abcnotation.com/.
14. This information was mostly used to trace our progress and to make sure we give credits to everyone involved in the preparation of our encodings.
15. The cpMark element was introduced to MEI following a proposal from the Freischütz Digital project. It handles abbreviations in a machine-readable, i.e. resolvable way. cpMark stands both for "copy mark" and "colla parte mark".

material. However, the capabilities of ABC are significantly smaller than MEI's, so that various features of the code could not be properly interfaced to the user. [16] While the concept of our “PitchTool” proved itself as extremely helpful, the implementation clearly suffered from the technological limitations of that time.

Before we were able to also proofread controlevents, we had to resolve the abbreviations we just introduced into our MEI files again, at least with regard to those events. For this purpose we merged our page-based files back into movement-based files, eliminating superfluous <scoreDef>s and ensuring that every (freshly added) element reliably had unique identifiers. We then used an XSLT script to resolve all abbreviations with regard to MEI events: For every staff with a measure repeat symbol (or any other “scriptural shortcut”), we introduced a <choice> element with the two childs <abbr> and <expan>. The first contained the abbreviated form (very frequently just a <space> element with the proper duration), while the latter was filled with copies of the material to be repeated.[17] That way, it became possible to offer two perspectives on the data, and extract either a “diplomatic” transcript of the source with all its scriptural omissions, or the full musical text that the source transports.

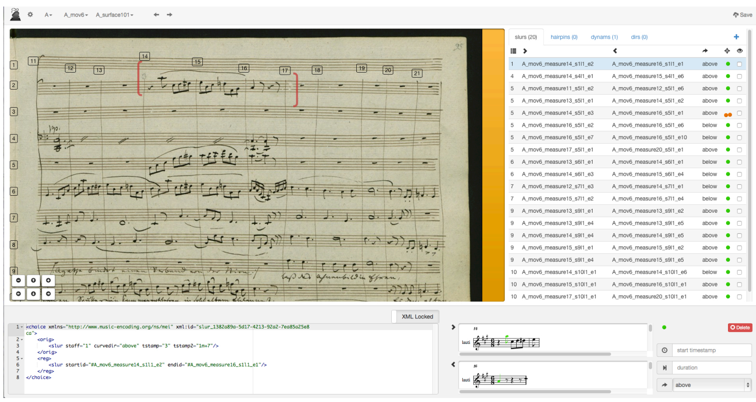


Figure 6: The tool used for proofreading controlevents like slurs.

With all gaps in the musical text being automatically (re-)filled, we were now able to proofread our controlevents.[18] Here, we built a set of specific tools, each focussing on just one type of controlevents, but all following the same principle. The user was displayed with one full page of the current source, which had overlays and the top and left margins indicating the measure and staff numbers respectively. Next to that, there was a list with all controlevents of the current type (i.e., slurs). If one of those was selected, a small, read-only XML editor was added to page, which showed the markup for this very element. In addition, two boxes where added that displayed a rendering of the staff in which the controlevent started and ended respectively.[19] In the facsimile, red brackets roughly indicated the position of the controlevent to check. In the renderings, the starting and ending note was highlighted. The now had to check if those matched the actual element on the page. In case those notes were wrong, it was possible just to click on the right note in the rendering, which replaced the incorrectly assigned xml:id with the correct one. However, it was also possible to indicate that an event was ambigously written in the score by clicking on more than one note. In that case, the user had to provide both an additional timestamp reflecting the correct visual placement. The tool would then generate a <choice> element, now with an <orig> (original form) child and the right number of <reg> (regularized form) childs. The first contained the element with @tstamp and @tstamp2 used for placement, while the latter contained one of the technically possible interpretations each. Only those

16. The ABC renderer we used does not allow to specify stem directions, and instead always uses his own algorithms to calculate them. As a workaround, we had to add arrow symbols above each note that indicated the stem direction specified in our MEI file.
17. Our XSLT introduced references from those copies to their corresponding original notes and rest, so that for every note etc. it's possible to trace where it was actually written down in the sources.
18. The reason for postponing this issue is that very often, controlevents like ties end in abbreviated measures with measure repeats only. Without those shortcuts resolved, there would be nothing to attach those controlevents to, and so they could not be reliably corrected before that.
19. As development for this tool started a year later than for the PitchTool, Verovio was already available. While still being in its infancy, it already greatly simplified the technical setup of this tool.

where later on synchronized with our core file. Of course the tool allowed to add new controlevents or delete existing as needed. All in all, it allowed us to reflect the sources with all their ambiguity into our code, without having to normalized and interprete at this stage already. At the same time, since all user input was channeled through form fields (and we used the rendered score as some kind of form here as well), we had absolute control that no links got broken due to incorrectly copied identifiers or some such.

With both events and controlevents being proofread, we were now able to finally align our data according to the Core Model. This was done in a semi-automatic way through a fairly complex XSLT.[20] The intention was to create an encoding with a) all references being correct, and b) an as fine-grained use of <app> and <rdg> as possible, reflecting human encoding styles. For every movement, a first XSLT splitted up the data from the Autograph[21], resulting in a new core file with all the information that was supposed to go in there, and an emptied encoding for the source, which had the correct pointers into the core. The next steps in the process were considerably harder. With a second XSLT, we now prepared one source after the other for inclusion into the core. Drawing on the alignment of staves build at the begin of our process, we compared the content of each staff from the new source to the core. When no difference was found, i.e. the source was giving the same musical text as the core, the was easy to automatically add pointers to the core from the cleansed source file.

In case of differences between core and source to be added, the smallest possible section of actual differences was identified, and wrapped by an <app>. The existing encoding for that section was then to be moved into a <rdg> within there[22], and a new <rdg> was created with the corresponding text for the current source. If the the core already had different readings from the previous sources, then the XSLT compared the current source to all of the existing readings, trying to see if there was a perfect match in any of the other sources. If so, all references where set to match that source. If not, the source with the smallest variation from the current source was identified, and then a “sub-branch” was created from there, utilizing a nested <app>. With this approach, the resulting markup matched the encoding style of a human encoder, and made it simple to identify commonalities and differences between the sources. One major complication though was that the comparison could not operate directly on the XML nodes, as we intended to ignore certain scriptural artifacts.[23] For that reason, the comparison had to operate on an abstraction of the markup, which only contained the information we considered relevant for our comparison.

Because of the significant complexity of this XSLT, and the fact that any error would potentially render parts of our data useless, it was set up to operate in two runs. On the first run, the XSLT would not actually change the markup of the core, but only include additional markup containing information what it *would* do if executed a second time. Then, a human editor had to go through these proposals, and decide whether they would be correct or not. These proposals were easy to find with a specific XPath expression, and had a special attribute to indicate whether they were checked already or not. Only when all proposals were accepted by

20. Our initial plans where to write another web-based tool for this, but there were no capacities left for that. Ultimately, after having written the XSLT, it became evident that it might have been possible to simplify the human control over that process through a specialized user interface, but the underlying transformations would not have been any easier in that case.
21. We chose this source as it was the only one for which all movements where encoded, and had been used as copy template for most other manuscripts. Other than that, it could have been any other source, with no further consequences to the model or the resulting data.
22. During this transformation, it was extremely important to make sure that all pointers from the sources which were already merged into the core stayed intact.
23. As the comparison operated on the level of staves, the visual distinction whether two instruments sharing a staff were written as indipendent layers or as stem-sharing chords, as is often the case with woodwinds, made a significant difference for the markup. We wanted to be able to trace the layout of any given source in this regard, but we did not want this to trigger an <app> in our core file, given that the actual notes were the same.

the editor, a second run of the same XSLT would actually implement these changes into the core and current source file. This procedure helped to identify some errors in the markup that slipped all previous steps of proofreading, and thus helped to increase overall correctness of our data.

Conclusions

The dataset generated for Freischütz Digital is massive. In raw format, it has about 230MB of XML, about ~500,000 @sameas-relations, ~75,000 @corresp-relations, ~16,000 <choice>s, ~11,000 different <rdg>s, and ~4,000 <annot> (annotations). Setting up an efficient data workflow for this project required significantly more time than we expected upfront, as there was nearly no software support for MEI to begin with. Accordingly, we had to build various assistive tools[24], ranging from very simple scripts that just helped with repetitive tasks, over highly complex XSLTs doing advanced comparisons and data maintenance work, to full-blown web applications that provided a user-friendly interface to the task of proofreading. Even though these tools are closely tied to the project's workflow, as an outcome they are surely as attractive as the dataset itself, and may serve as model for future developments.

With the substantial amount of time needed to develop these tools, it was necessary to reduce the amount of material that could be encoded. Accordingly, the only source that has been fully encoded throughout all 3,100 measures of the Freischütz is the autograph. For the existing eight original copies of the Freischütz and the first print (Schlesinger 1849), we were able to fully encode Numbers 6 (Duetto, 149 measures), 8 (Scena ed Aria, 198 measures), and 9 (Terzetto, 199 measures). This sums up to ~8,000 measures of fully encoded music, which provide both a "vertical" comparison of multiple sources, and a "horizontal" overview of the full Freischütz. With the CoreViewer prototype, it is possible to browse these encodings and highlight features of the script (such as abbreviations, but also custom XPath query results). For all other parts of the sources mentioned (plus some additional sources), structural data like measures (including facsimile positions) are available.

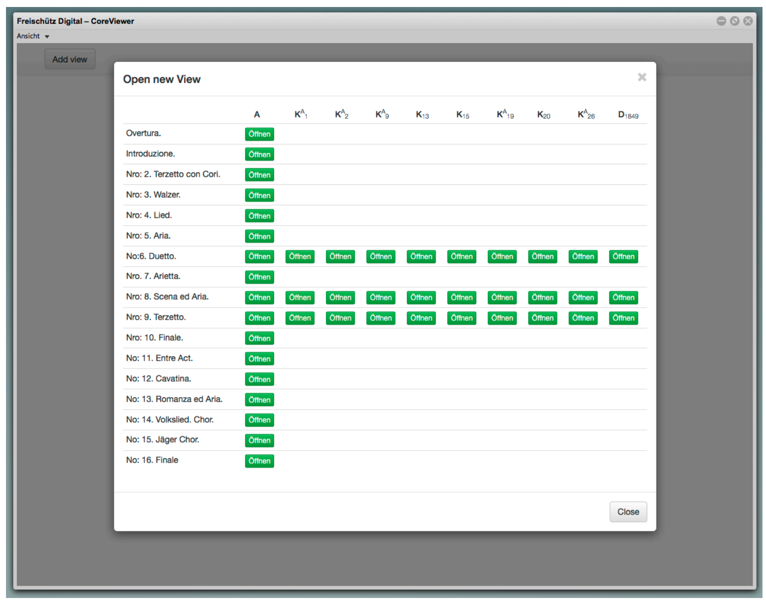


Figure 7: Screenshot from the CoreViewer, a prototype application that gives access to all full encodings of Freischütz Digital.

When the project started in late 2012, we intended to build a digital edition not unlike prior Edirom applications, but with encodings of the full score, linked audio recordings, and a complete diplomatic edition of libretto included. Very soon, it became obvious that this expectation could not be met, for various reasons. This was not only due to the complexities of the data preparation workflow described above (which led to

24. Obviously, one of the most important tools to control an MEI-based data workflow is the use of a customized version of the MEI schema itself. For us, it proved extremely helpful to use additional Schematron rules alongside MEI's RelaxNG schema, as it is possible to have multiple such rules, each covering a specific aspect. Without Schematron, it would have been hard to ensure consistency of our data.

statements similar to Agathe's famous exclamation “*Wie? Was? Entsetzen!*” – “*How? What? Oh, horror!*” more than once), but also to the lack of workable concepts for advanced digital edition in general. While we were able to actually integrate all our data and prototypes into the Edirom, it became clear that this approach of an “one-size-fits-all” application is not ideally suited to satisfy varying expectations on an edition. Instead, a data-driven approach, with individual, only loosely connected tools, each focussed on a given research question, seems much more versatile. Accordingly, all our prototype applications are also available as standalone applications, and offer different perspectives on the Freischütz Digital dataset. With such an open architecture for digital editions, it becomes much easier to integrate other tools and perspectives on the data.[25] It also helps to keep the technological complexities at a manageable level. In that sense, Freischütz Digital clearly illustrates the benefits and potentials of genuinely digital scholarly editions of music.

Works cited

[1] Wiering, Frans. "Digital Critical Editions of Music: A Multidimensional Model", in: "Modern Methods for Musicology. Prospects, Proposals and Realities", ed. by Tim Crawford and Loma Gibson, Farnham 2009, p.23–45.

[2] Complete Works of Carl Maria von Weber (WeGA). Digital Edition, http://weber-gesamtausgabe.de/A070006 (Version 3.4.2 of February 8, 2019) Last change of this document on December 17, 2017.

[3] Viglianti, Raffaele, et al., “A stand-off critical apparatus for the libretto of Der Freischütz”, Paper delivered at the TEI Members' Meeting 2013 in Rome. Abstract available at http://digilab2.let.uniroma1.it/teiconf2013/program/papers/abstracts-paper/#C151.

25. On FreiDi Plus, we collect such efforts. At the time of this writing, two student projects are built on top of the Freischütz dataset.

All Classical Music, Freely Available As Revisable Digital Scores, Via Crowdsourcing

Jim DeLaHunt
The Keyboard Philharmonic Project
jdlh@KeyboardPhilharmonic.org

Abstract

Our heritage of music scores need to be delivered in 21st century packaging: the symbolic, digital score, instead of the paper-printed book. This will enable better music-making, but also new tools for using digital scores. The digital score plus the new tools will enable dramatically better music-making. Thus it is important to create a corpus of digital scores for the works in our musical heritage. A gap analysis shows that what is missing is an organization which mobilises participation and produces results, and which provides content in symbolic, digital score form, and freely licensed. Such an organization has recently begun operations: the Keyboard Philarhmonic project.

About the Keyboard Philharmonic

The Keyboard Philharmonic project organises music lovers to transcribe music scores of public domain opera and classical music into revisable, shareable, reusable digital files — and to give those digital scores away freely. We bring great scores into the 21st century, and enable new digital tools for musicians, educators, and music lovers.

Key elements of the project:

- **People, people, people**. Identify people who share our goals, reduce the barriers to participation, and motivate them to contribute their time. Gamification is part of the motivation.
- **Revisable, digital scores**. Symbolic information content helps elevate the whole ecosystem beyond what printed scores permit.
- **Public domain and free culture**. Public domain content means no permission or licensing needed. Plenty of compelling content is available. Giving away free content enables us to ask for free labour: the Wikipedia and MusicBrainz model. Giving away content freely lets musicians, teachers, and students use it freely.
- **Music Charity** as corporate form. Trying to charge rent on free content has failed many times. Charity model enables: free scores in + free labour → free digital scores out.
- **Technology Startup** behaviour. Newly begun. Web-hosted community. Agile. Minimum viable products. Initial budget is approximately $0. Fail fast and often.
- **Complementary for-profit businesses**. New good content will spark new opportunities to add value. This is not a contradiction: commercial and free/libre have proven coexistence models.

Workflow and products

We have draft editorial standards for two products:

1. A `Fidelio edition` digital score. Corresponds faithfully, note-for-note and page-for-page and error-for-error, to a specific source printed score. In MusicXML format (MEI under consideration). The primary product. The faithful correspondence to the source score permits easier distributed proofreading.
2. A `433 edition` digital score. Corresponds measure-for-measure and page-for-page to a specific source printed score, but each measure consists of rests only, no notes. Shows measure counts, key and time signatures. Advantage: far faster to transcribe. The `433` edition provides a structure on which to add notation. It also permits initial proofreading. `433` editions of two different printed scores is enough to make a page cross-reference product.

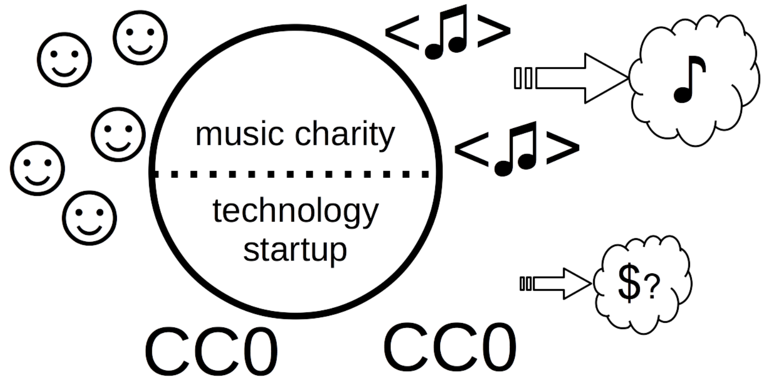


Figure 1: Key elements of the Keyboard Philharmonic project

Critical editions, improved editions, etc. are not planned as Keyboard Philharmonic products. Individual musicians and musicologists will be empowered to make these editions themselves, using `Fidelio` editions as starting points.

The workflow starts from a `reference score`.

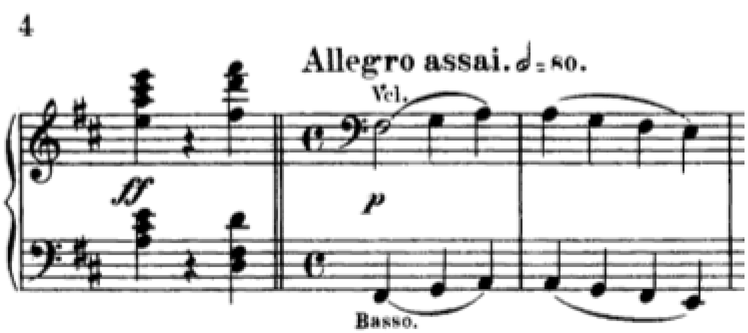



Figure 2: Reference score example

A coordinating person breaks the `reference` score into fragments (pages, perhaps), and distributes them to helpers. Each helper creates the `433` transcription of their fragment. The coordinator gathers the transcriptions, and assembles them into one `433` edition of the entire score. Then the coordinator breaks the `433` edition into fragments, and distributes them and the `reference` score to another set of helpers. These helpers check the `433` edition against the structure of the `reference` score. The coordinator gathers the corrections and applies them to the `433` edition.

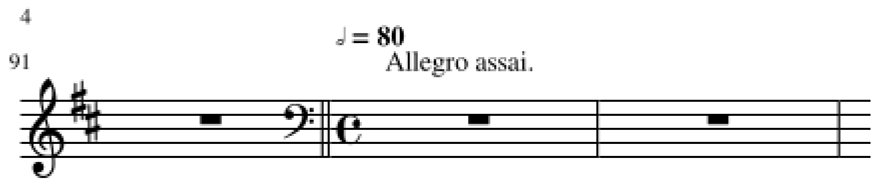


Figure 3: 433 edition example

In the next step, the coordinator breaks again the `reference` score into fragments, and distributes them to helpers along with the `433` edition. Each helper enters notation into the `433` edition, creating a `Fidelio` edition of their fragment. Again the coordinator gathers the transcriptions, and assembles them into a draft `Fidelio` full edition. Finally the coordinator runs another review cycle, this time checking fragments of the `Fidelio` edition against corresponding parts of the `reference` score.

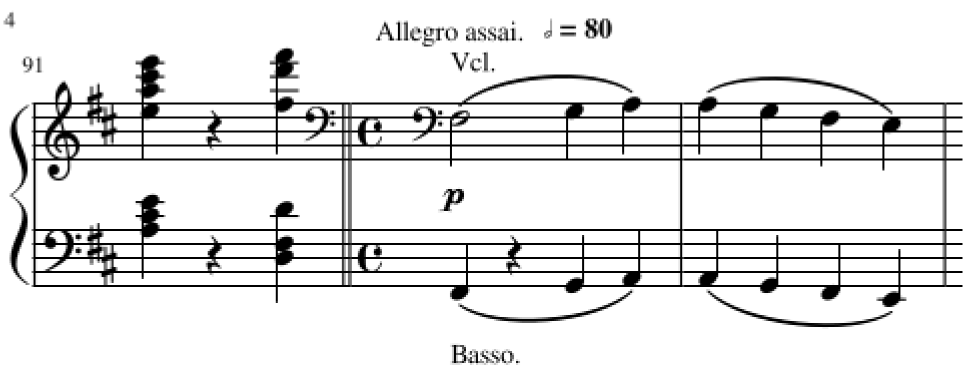


Figure 4: Fidelio edition example

The `Fidelio` edition contains symbolic notation information, rather than image data, the format of the `reference` score. It may be in MusicXML and/or MEI format.

```
<measure number="92" …> <note …><pitch><step>F</step>…<duration>2</duration>…
```

Figure 5: Simplified view of Fidelio edition’s symbolic content

These products are freely distributable. They may be shared with repositories like IMSLP, CPDL, archive.org, and with other projects.

Contributor and beneficiary engagement

Two critical resources for the Keyboard Philharmonic project are contributors and beneficiaries. The project workflow is designed to attract both. We have a staged process for finding and gradually enlisting people.

1. **Following**. Subscribe to an email announcement list. Provides a low-effort step for people who are interested in the project, and attaches them to the project. Also: Twitter, Facebook.

2. **Benefitting** from a Keyboard Philharmonic digital score. Low-effort, high gratification. Builds loyalty to the project. Spreads word of project to other people. Obstacles: only possible if catalogue of digital scores includes desired work; catalogue will be small compared to long tail of existing works for a few years; catalogue of scores will be tiny at first.
3. **Contributing simply**. E.g. proofread a page of a score, transcribe one page of a score. Receive assignment from lead editor, complete it, submit to lead editor, done. Name appears in contributors list for that score.
4. **Contributing more**. E.g. proofread many pages or transcribe many pages, solve difficult encoding problems, supervise layout of score. Name appears prominently in contributors list for that score.
5. **Organising contributors**. Invite a group of people to make simple contributions. Could be part of a project to "sponsor" transcription of a work. E.g. music teachers, a choir transcribing a work they want to perform from digital scores. Digital score transcription could be dedicated to that group.
6. **Leading editions**. Organise work of contributors to a score project. See a project through from beginning to end. Praise contributors, encourage them to move to larger contributions.
7. **Administering**. Organise the work of the Keyboard Philharmonic project. Maintain project web site. Develop social norms and build community. Set strategy.

We expect to use gamification to motivate and reward participation. We will award points for contributions, give badges for specific accomplishments, and unlock increased authority accordingly. Our models are StackExchange sites, and Wikipedia.

There will probably be some kind of software developer engagement track also. This will have a similar structure to other free software projects.

Challenges

Current MusicXML and MEI formats are good enough to get started, but have limitations in both functionality and in application support. Notation applications have bugs and limitations, and challenges are present on just about every score page.

We lack in-web-page score editing. Contributors must download a file, make changes, and upload it again. This is much more difficult than being able to edit with a click on a link, and it may discourage potential contributors.

Find out more

Fill out the subscription form and join the Keyboard Philharmonic email announcement list.[1] It is low volume, spam-free, and is your best way to stay in touch.

1. www.keyboardphilharmonic.org

Contributions from MEC 2017

Capturing Context and Provenance of Musicology Research

David Lewis
University of Oxford e-Research Centre
david.lewis@oerc.ox.ac.uk

Kevin Page
University of Oxford e-Research Centre
kevin.page@oerc.ox.ac.uk

Andrew Hankinson
Bodleian Libraries, University of Oxford
andrew.hankinson@bodleian.ox.ac.uk

Introduction

As the number of music corpora published online increases, with many of these containing symbolic music encodings, the importance of search activities is also growing. Although it is seldom explicitly recognised as such, the act of querying these corpora is a form of research activity. Despite this, no common standards have emerged for preserving and disseminating these activities, or their results and provenance. In this paper, we discuss two examples of current search tasks—each with a musicological motivation—and consider what information might be preserved and communicated, and why. Capturing the method of a search, the timing of the query and the motivation for the investigation may be as important as preserving results. We show a preliminary Linked Data implementation, and consider how it can support visualisation of the results and evaluation of the methods.

In the first of our use cases, an *ad hoc*, document-retrieval task is recorded, allowing a user to label, store and communicate to others an interesting set of results. Since the system is live, there is no guarantee that the dataset or the algorithm will not change, so a 'save' functionality must permit the recording of a snapshot of the queries and search results as they were at the time of retrieval, as well as providing provenance information.

In our other use case, many searches are carried out, the sum of which form part of a larger musicological inquiry. The results of the search must be considered by the musicologist, subjected to analysis then disseminated for future research. The instigator of the search has had to create mechanisms for preserving the results and subjecting them to analysis; our motivation is to provide infrastructure to give the same affordances to other future researchers, so they can replicate, criticise or extend the analysis.

Background and motivation

Re-use of *Liber Usualis* searches

Our first case study centers around the online *Search the Liber Usualis* interface developed in the SIMSSA project.[1] a database of chant melodies based on Optical Music Recognition (OMR) applied to a modern edition. The database can be searched using interval sequence, and users might query such a corpus in order to find pre-existing examples of an element of a chant melody or identifying chant quotations in polyphonic music.

Many such queries would be seen as one-off information gathering activities—if the chant melody in a newly-discovered fragment is unequivocally identified with this method, there is no obvious need to preserve the search and its results. Other queries, particularly ones for partial matches, or where counts or corpus-dependant information are important (for example, the earliest instance of a motive in the database), are less straightforwardly true or false, and may yield different results based on the state of the database and its associated technologies. In these latter cases, there is a strong argument for providing the ability to annotate and preserve searches and their results.

Intabulation research

Our second case study explores one of the most common types of solo instrumental music in the 16th century, the *intabulation*. (The name stems from the concept of a piece of music as a 'table' of its polyphonic voices stacked in columns. This gave rise to the use of the word "tablature" for the specialised form of music notation used for instrumental music, most notably for lute and related instruments and for keyboard instruments.) Such pieces, which constitute roughly half of the printed and manuscript instrumental repertory, consist of music originally for vocal ensemble arranged for a solo instrument capable of playing most, if not all, of the parts at once—generally instruments from the keyboard or the guitar/lute family. This process was not a simple mechanical one, and the arrangements were almost always given some degree of embellishment. Brown [1] lists all the elaboration patterns used in three lute intabulations of the same popular vocal work (Jachet Berchem's chanson, *O s'io potessi donna*) classifying them by the original vocal diatonic interval-class pattern that they replace. Lewis, Crawford & Müllensiefen [2] use the patterns from Brown as search terms, querying corpora of lute music, showing that the patterns used were more frequent in lute intabulations in general than in the rest of the instrument's repertory, and, furthermore, that a difference can be discerned between their usage in arrangements of sacred and secular vocal music.

Brown gives 78 patterns, and the approach used by Lewis, Crawford & Müllensiefen required the generation of 88,000 queries from these, accounting for all possible chromatic variations of the diatonic patterns and different degrees of rhythmic scaling. Applying these to a corpus of over 6,000 pieces meant carrying out over $5\text{x}10^8$ searches, the results of which were stored in a database and presented only as summary statistics. Such large amounts of data must be carefully handled if they are to be genuinely re-usable, but also made intelligible to humanists.

An alternative way to carry out the task—using the same set of elaborations, corpus and search algorithm, but applying a pitch-spelling algorithm and searching using the diatonic pitches rather than chromatic pitch—would generate another data set approximately one hundred times smaller and execute

1. http://liber.simssa.ca

proportionately more quickly. Comparing these two methods requires musicological skills and a user interface that facilitates quick interrogation and comparison of results, but it also requires that the provenance of each be unambiguously recorded.

Approach

In each case, we use the W3C PROV data model and ontology [4] to annotate the relationships between information from each step of the retrieval process. PROV provides a simple model based on `Entities` that are used or produced by `Activities`, either of which may be associated with or attributed to `Agents`, which may be humans or pieces of software (see diagram below). This simplicity is designed to supplement rather than replace more specialised ontologies either for specific domains or for modelling workflows in detail.

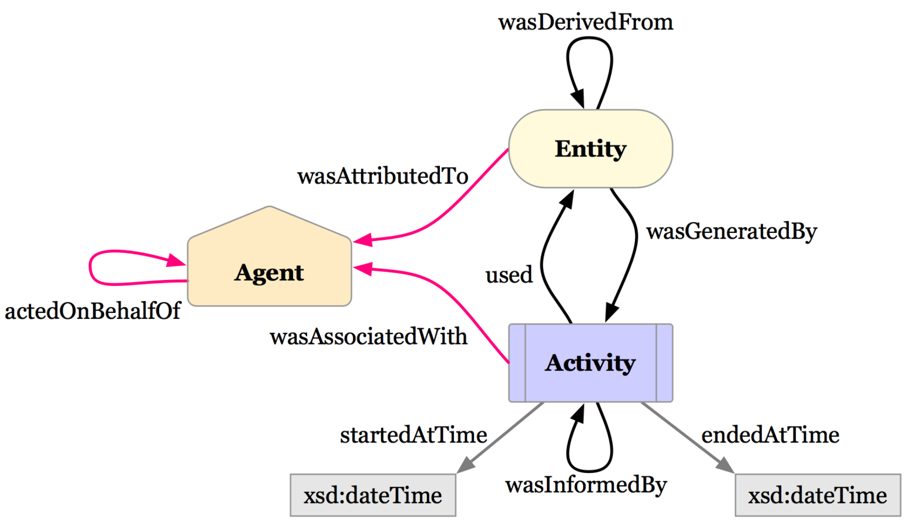


Figure 1: The core classes and properties of PROV (reproduced from [4]).

Application to the *Liber Usualis*

Liber Usualis queries can be entered as a string of pitch-codes which are then parsed by the query engine. In this instance, a simple approach for documenting the query is to make no attempt at parsing this string—simply recording its content—and to regard the processing of the query and generation of results as a black box. We record the query string and a user-provided label for the search, and then timestamps and URL for the search.

Like the query, the results returned by the service are also not explicitly musical—they consist of a sequence of bounding boxes for the images of the chant book used. Again, this can be accommodated—we record what information we have, and do not try to associate musical meaning with it. The semantics that we have used are shown in Figure 2.

Since the SIMSSA *Liber Usualis* system functions through a stateless interaction with server via HTTP GET, we can easily construct a near identical version of the web interface that can either query the SIMSSA servers or save or retrieve searches published (locally or publicly) using these semantic structures.

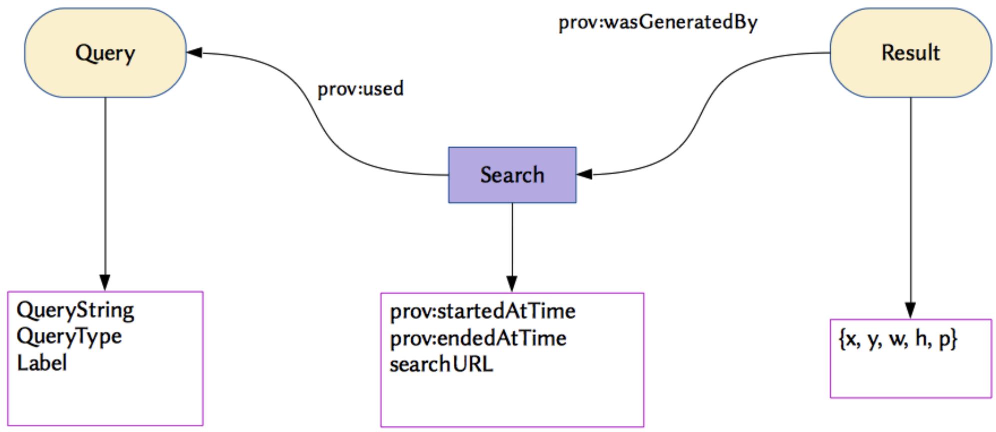


Figure 2: A simple PROV model for a *Liber Usualis* search.

Intabulations research

The intabulations queries both require and permit more detailed annotation, partly because the process is more complicated and partly because we do not regard the computation as a black box for the purposes of this paper. Where the simple query discussed earlier is modelled as a single activity, here we model the process of turning a single instance of Brown's patterns (here called 'Template') into multiple queries ('Expansion') and that of searching the corpus ('Search') as two separate activities. That allows alternative methods for performing either task to be compared or substituted independently.

Further richness is available here because the queries we generate and results can both be published as MEI. This is particularly valuable for recording the precise location of the discovered matches in the search result. This information could be subsequently used to inform further analysis, but also as data for presentation interfaces where matches are highlighted in the browser and optionally made to support further user interaction.

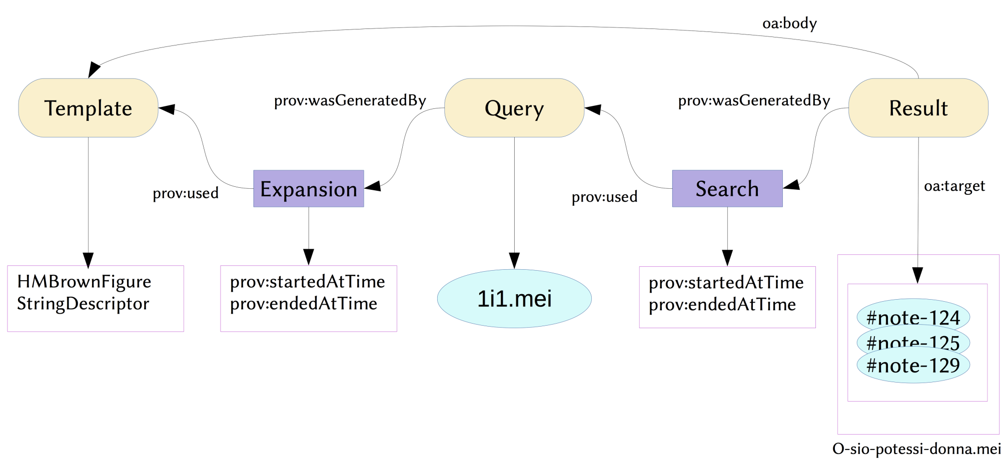


Figure 3: A richer model for capturing the process of intabulation pattern detection, recording separately the steps of turning Brown's patterns into concrete queries, and then running those queries on a corpus.

Since another approach might create different intermediate stages, or even have no intermediate stages, it is convenient for later comparison to record the direct connection between the search results and the original pattern. This is achieved using an open annotation.[2] .

Discussion

We have shown how the PROV data model can be used to capture key aspects of workflows in generating different musical search results. The potential benefits for communicating reusable research, with transparent, clearly-described processes, are significant. The low complexity of the models we describe here makes them easily implemented as extensions of existing functionality. Using these structures, even small investigations can be citable as a whole, while individual components can also be referenced. Most of the information that is not domain specific is provided through PROV, which also lowers the barrier for those from outside the discipline in reusing data published in this way.

The benefits extend beyond simple cross compatibility and reuse. In the case of the *Search the Liber Usualis* we can easily use the existing publicly-available functionality, adding our own extension to permit saving and retrieval of queries.

In the case of the lute music search, the preservation of MEI identifiers throughout the search process and their publication as linked data gives the opportunity to present them within a more generalised framework than would normally be the case. The Music Encoding and Linked Data (MELD) approach [3] allows these musical structures, associated with semantic tags that specify the nature of the relationship between them, to be explored programmatically. The MELD framework permits the exploration of musical annotations, displaying music notation using Verovio and highlighting regions as necessary. Applying this system to the annotations described would allow users to explore the intabulations dataset, viewing results lists with metadata, and then to select a result to view the piece in tablature or staff notation, with matches highlighted as desired. In future work, we will attempt to model musicological observations using similar linked-data structures and explore ways of accessing these using the MELD framework.

Conclusion

In this brief exposition, we have shown a simple PROV mapping that focusses on recording the process and components of search activity. Once we can publish and record these, more information can be added depending on the aspects of the research process that are deemed most interesting to capture. PROV provides the ability to make attributions of responsibility to human or software agents, along with mechanisms for describing the versions of corpora and software used and how they relate.

Search activities such as these are special, constrained cases of musicological investigation in general. There is considerable value in capturing the output of such investigations, associating them with information about provenance and musicological motivation, and sharing the whole thing as a unit.

Works cited

[1] Brown, Howard Mayer. "Embellishment in Early Sixteenth-century Italian Intabulations" *Proceedings of the Royal Musical*

2. www.w3.org/community/openannotation

Association 100 (1973): 49-83.

[2] Lewis, David, Tim Crawford and Daniel Müllensiefen. “Instrumental Idiom in the 16th Century: Embellishment Patterns in Arrangements of Vocal Music” in *Proceedings of the 17th Conference of the International Society for Music Information Retrieval, New York, 2016*, p. 524-30 (https://dblp.uni-trier.de/db/conf/ismir/ismir2016.html).

[3] Page, Kevin R., David M. Weigl, David Lewis and Carolin Rindfleisch. “Prototypical Scenarios for Contextual Navigation with MEI and Linked Data”. Paper presented at the Music Encoding Conference, Montréal, 2016. Unpublished.

[4] Lebo, Timothy, Satya Sahoo and Deborah McGuinness, eds. “PROV-O: The PROV Ontology”. World Wide Web Consortium, Recommendation, 30 April 2013 (www.w3.org/TR/prov-o). Accessed August 2017.

Creating an Encoding Workflow for a Critical Edition of Ottoman Music Manuscripts: Challenges and Solutions

Anna Plaksin
Technische Universität Darmstadt
anna.plaksin@stud.tu-darmstadt.de

Jacob Olley
Westfälische Wilhelms-Universität Münster
olley@uni-muenster.de

Abstract

This paper outlines an encoding workflow for a critical edition of Ottoman music manuscripts based on the MEI schema. The encoding workflow is one aspect of a larger digital edition project entitled Corpus Musicae Ottomanicae (CMO). The paper gives a brief introduction to CMO, its scholarly goals and digital infrastructure. It offers some basic information on Ottoman music, focusing on the scholarly transcription into staff notation of manuscript sources written in Hampartsum notation. We discuss the modelling of metadata according to FRBR and MEI, and the challenges involved in adapting the source material to schema that were developed with different musical concepts and practices as their model. We describe the workarounds, tools and procedures we developed to encode Ottoman music in a way that meets both visual and semantic demands, and reflects musical as well as philological aspects of the sources. These solutions are offered as a possible first step towards integrating non-Western musical repertoires and sources into the MEI schema.

Introduction

The encoding workflow described in the present paper represents an attempt to integrate the critical edition of Ottoman music manuscripts with current standards of digital musicology. The scholarly material (i.e. scores and metadata) was produced within the framework of the digital edition project Corpus Musicae Ottomanicae (CMO), which is described in more detail below. The encoding standards are based on the Music Encoding Initiative (MEI) and, for the metadata, Functional Requirements of Bibliographic Records (FRBR).

The CMO Edition and Source Catalogue are in principle designed to be compatible with the MEI schema. However, there are a number of obstacles to integrating CMO with MEI, which stem from the particular scholarly demands of Ottomanist musicology; differences in musical concepts and notational techniques or systems; the limited possibilities of existing notation software; and the philological diversity of the source material. The workflow that is outlined here constitutes a tentative first step towards establishing MEI-compatible encoding standards for Ottoman or other types of non-Western music, and is not intended as a definitive solution to any of the challenges we describe.

The solutions that we arrived at are mainly the product of experiments and discussions between the authors in the period from October 2015 to January 2017. The paper also reflects the input of several other individuals involved in the CMO project during this period, especially Fabian Cremer, Zeynep Helvacı, Ralf Martin Jäger, Salah Eddin Maraqa and Ersin Mıhçı. The paper deals only with the encoding of the musical transcriptions, and does not tackle the encoding of the critical commentary or the edition of lyrics, both of which demand a separate and fuller treatment.

While the paper does not discuss in detail previous research into the digital possibilities of Ottoman and Turkish music studies, we hope that it contributes to this small but growing field, to which researchers such as Kemal Karaosmanoğlu [5], Sertan Şentürk [10], Erdem Ünal, [11], Barış Bozkurt and André Holzapfel [1] have made significant contributions. One of the limitations of this previous work, which focuses mainly on data retrieval and computational analysis, is that it takes as its starting point the theoretical and practical norms of the contemporary era, based on scores and recordings compiled mostly during the twentieth century. However, as a number of scholars have shown, the sources of Ottoman music display a high degree of variability in terms of musical content and notational conventions, and need to be analysed according to their historical context. (See [12], [4] and [7].

A central aim of the present paper is thus to suggest how encoding workflows might take greater account of the historical and cultural specificity of musical sources. One way of doing this, we suggest, is by developing a workflow that reflects the particular characteristics of the notation system from which a piece is transcribed, and which includes not only musical data, but also paleographic and philological data. The MEI schema is particularly well suited to the latter task, since it has been developed with similar goals in mind, albeit mainly in relation to Western tonal music. We hope that the paper serves to stimulate further debate on the challenges involved in encoding non-Western music and making the MEI schema more inclusive, and we welcome feedback or critical suggestions regarding the encoding solutions described herein.

Corpus Musicae Ottomanicae

Corpus Musicae Ottomanicae (CMO)[1] is a long-term research project established in 2015, funded by the Deutsche Forschungsgemeinschaft (DFG) and based at Westfälische Wilhelms-Universität Münster (WWU). The project is led by Prof. Dr. Ralf Martin Jäger and comprises an international group of scholars based in Münster and Istanbul.

The central aim of CMO is to produce critical editions of notated music manuscripts originating in the Ottoman Empire and the wider Middle East region. In the first phase, the project focuses on manuscripts written in Hampartsum notation, a reformed system of Armenian church notation that was developed shortly before 1812 and used by both Armenian and Muslim musicians in Istanbul to notate diverse repertoires during the nineteenth century [6]. In the second phase of the project, CMO will focus on manuscripts written in Western staff notation during the same period. Parallel and complementary to the CMO music edition, the project also includes an edition of the lyrics of the notated sources and an online catalogue of Ottoman music sources.

The CMO Edition is published via perspectivia.net, and is an open-access publication. The CMO online publication platform, developed in collaboration with the Max Weber Stiftung and the Gemeinsamer Bibliotheksverbund (GBV), provides access to volumes or individual edited pieces in a number of formats (PDF, HTML, XML). Via the same platform, users can also access the CMO Source Catalogue, which contains data sets for several thousand musical compositions, composers and lyricists, printed and manuscript sources and secondary literature related to Ottoman music. Developed on the models of FRBR and MEI 3.0, the Source Catalogue enables users to make detailed searches for a wide range of musical data (in various languages and scripts), which are compiled according to the standards of musicological and Ottomanist scholarship. The CMO Edition is fully integrated into the Source Catalogue, so that users can access further data related to the

1. www.uni-muenster.de/CMO-Edition/en/cmo/cmo.html

edition (e.g. biographical data of a composer) via the catalogue. Most importantly, edited pieces are linked with musical and textual variants within the Source Catalogue, so that users can easily locate and consult concordances.

CMO is the first large-scale critical edition and catalogue of Ottoman music sources, and represents a major step towards integrating non-Western musical traditions into the methodological, intellectual and institutional frameworks of historical musicology, particularly in relation to evolving paradigms of digital scholarship. It has the potential to establish new standards for the edition, cataloguing, publication and encoding of non-Western musical sources and systems of notation, while at the same time contributing to a diversification and further development of the existing tools of digital musicology, including the MEI schema.

Transcribing Hampartsum notation into staff notation

The present section provides a brief outline of the conventions used when transcribing music from Hampartsum notation into staff notation accordıng to the CMO guidelines.[2] The CMO Edition uses the notation software Sibelius. Pieces are transcribed individually, and editorial conventions may vary somewhat depending on particular modal or rhythmic characteristics or individual scribal practices. Since Hampartsum notation was used over a period of a century or more by musicians of diverse educational, linguistic and cultural backgrounds, the corpus displays a wide range of musical and paleographic features.

The fundamental elements of Ottoman music are **makâm** (mode) and **usûl** (rhythmic cycle). The heading of a piece in a manuscript collection of Hampartsum notation typically states the makâm, usûl and genre in which it is composed. In addition, the heading often includes an attribution to a composer, and sometimes a programmatic title. In the case of vocal pieces, it may include an incipit. This information is used within the CMO Edition and Source Catalogue in order to make pieces searchable and identifiable. The metadata for a single piece or transcription also includes a CMO identifier and the physical location of the piece according to the library sigla of the Répertoire International des Sources Musicales (RISM). For ease of searchability, composer names and dates are standardized as far as possible, while further biographical data, alternative names or spellings and scholarly citations are accessible through the Source Catalogue.[3]

Hampartsum notation has two levels: signs indicating pitch are written at base level, while signs indicating duration are written above the pitch signs. Durational signs may also indicate rests when they are written at base level. A number of other signs are used to indicate structural features (e.g. formal sections or repetitions) or other musical elements such as phrasing or different types of articulation.

The **usûl** or rhythmic cycle is a skeletal pattern of drum strokes that can be embellished in performance, and which may be simple or highly complex. The total number of time units in an usûl cycle can range from 2 to 120, within which a unique pattern of strokes is defined according to both relative duration and timbre (low-pitched sounds are designated *düm*, high-pitched sounds *tek*). The number of units in an usûl is referred to as *darb* (lit. *beat*), and this is provided at the beginning of the transcription. The value of the basic time unit is relative rather than absolute, but for the sake of legibility is most often transcribed as a half-note. The value (i.e. the denominator) of the basic time unit is provided above the darb in the transcription (e.g. 1 = 𝅗𝅥).

2. For a concise introduction to Hampartsum notation as a method of transcription, see [9].
3. Thanks to the work of Elif Damla Yavuz, several hundred Ottoman composers and lyricists have been assigned Integrated Authority File (GND) numbers through the cooperation of CMO with the German National Library (DNB).

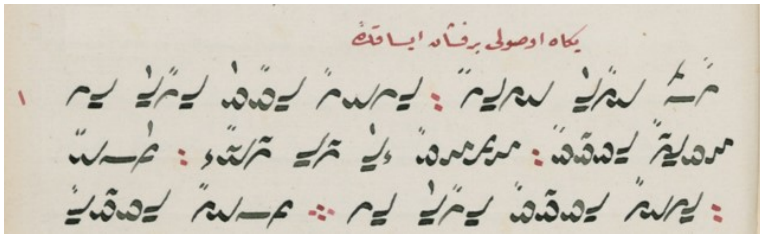


Figure 1: Hampartsum notation, "Yegāh ūṣūli berefşān Īsāķıñ" (TR-Iüne 207-5, p. 27).

The usûl pattern is added by the editor in the lower staff, based on the description of the pattern given in contemporary Ottoman sources. The usûl plays a central role in the formal and melodic structure of a composition. When written in Hampartsum notation, long usûls are divided into smaller sections (referred to here as *divisions*) by means of colons, while the end of an usûl cycle is usually indicated by a double colon (sometimes placed diagonally). Within a division, Hampartsum symbols are grouped into smaller blocks (referred to here as *groups*) that help to determine the relative duration of pitches or rests. Groups are represented in the transcription by corner brackets, which enable the user to reconstruct the distribution of signs in the original source. Users can navigate between the transcription, commentary and source by means of division and group numbers, as well as page, column or line breaks.

Structural signs vary from source to source, and may consist of words, stylized letters or other abstract symbols indicating the formal sections of piece and the order in which they should be performed. Using a specially developed font for Hampartsum notation,[4] we are able to insert the original signs into the transcriptions. Patterns in the distribution of such signs can enable researchers to identify relationships between different scribes or manuscripts.

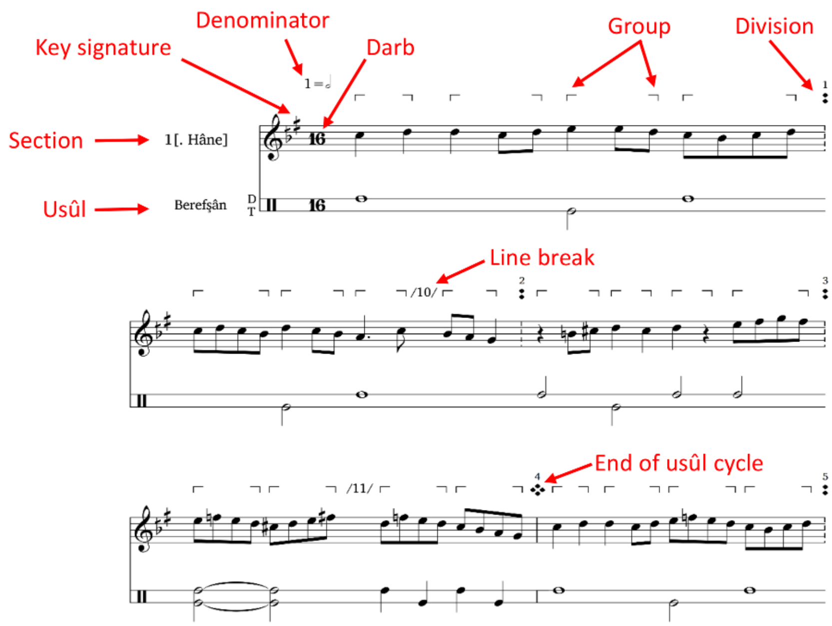
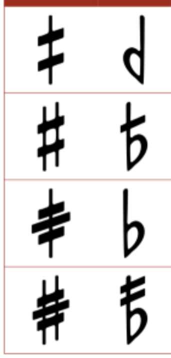


Figure 2: TR-Iüne 207-5, p. 27 (part), transcribed by Salah Eddin Maraqa.

Hampartsum notation has seven basic pitch signs, which are modified to indicate higher or lower octaves. Each sign corresponds to a named pitch within the Ottoman musical system. A small tilde (known as *gisver*) placed above or below a pitch sign raises the pitch by a degree that can vary from a large half-step to an interval less than a quarter-step. In the final decade of the nineteenth century, partly under the influence of European musicology, Ottoman intellectuals began to conceptualize the pitch system according to the intervals described in Arabic- and Persian-language treatises of the thirteenth to sixteenth centuries. This resulted in a theoretical system that is widely accepted in Turkey today, and which divides the whole tone into 9 Pythagorean commas. In the **Arel-**

4. VF OttoAneumatic, developed in collaboration with Vladimír Faltus and Haig Utidjian at Charles University in Prague.

Ezgi-Uzdilek (AEU) system, modified sharp and flat signs indicate the precise number of commas by which a pitch should be raised or lowered. Each type of interval is designated with a name: koma (1 comma), bakîye (4 commas), küçük mücenneb (5 commas) or büyük mücenneb (8 commas).



Sharp	Flat	Name	Value
		Koma	1 comma
		Bakîye	4 commas
		Küçük mücenneb	5 commas
		Büyük mücenneb	8 commas

Figure 3: Alteration signs in the Arel-Ezgi-Uzdilek (AEU) system.

The Ottoman pitch system is relative rather than absolute. However, it is a widely adopted convention that the central pitch (known as *râst*) should be transcribed as g. The treble clef is used in the transcriptions, followed by a key signature where necessary in order to reduce the number of accidentals within the score. Accidentals apply for the duration of a division, unless cancelled by a subsequent accidental within the same division. A single key signature is usually applicable to a number of different **makâms** (modes).

Although it is generally acknowledged that the AEU system does not precisely represent the intervals used by Turkish musicians in practice, it has become a common standard for theoretical and educational purposes. For the sake of simplification and standardization, the AEU system is also used for encoding the CMO Edition. In future, it may be desirable to develop the encoding of alteration signs further, so that they reflect more accurately historical concepts of pitch rather than theoretical frameworks established in the twentieth century.

Creating a digital edition of Ottoman music manuscripts

On metadata

Besides the more complex issue of creating an encoding for the transcribed music, modelling the metadata in conformity with the specific character of the music culture requires creative solutions. Firstly, the work descriptions require special elements, e.g. instead of `<key>`, the **makâm** should be indicated, and `<meter>` should be replaced by **usûl**. In addition, it is necessary to provide further details about the lyrics of vocal compositions and additional information related to the description of sources. In a music culture that encompasses multiple, coexisting languages, scripts, and notation systems, the accurate description of these features is essential.

Yet the most challenging issue in modelling metadata for Ottoman music is the fact that the notations are unique snapshots of a living musical tradition that was for the most part orally transmitted. Rather than representing fixed musical works, the manuscripts record multiple variants and lines of musical transmission. Thanks to the FRBR module [8] it was possible to deal with this special situation through a bottom-up approach: the variants were first catalogued as expressions and then grouped into works, with the latter serving mainly as *collections of variants*. This approach also makes it possible to deal with complex cyclic genres (such as the Mevlevî âyîn) and unique performance cycles consisting of a group of individual works (i.e. fasıl or takım).

Processing modified Western music notation

In order to make the corpus easily accessible to a wide range of scholars and performers, the CMO digital edition provides transcriptions from Hampartsum notation into modified Western staff notation. An important and innovative goal of the edition is to make the transcriptions available in a machine-readable form that is both textually and semantically accurate. To fulfil this aim, it was necessary to develop a semantically rich music encoding system that provides not only an adequate representation of the musical content, but also of the special characteristics of Hampartsum notation, while at the same time meeting internationally acknowledged standards for the critical edition of music. Moreover, a key requirement was to build the encoding workflow into a score-writing software.

While there have recently been efforts to develop encoding standards for Turkish modal music, for example the notation software Mus2 and the symbolic data collection SymbTr [5],[5] they focus mainly on music information retrieval, and omit philological aspects relating to the original sources and notation systems. Moreover, although the SymbTr collection provides a MusicXML version and Mus2 includes an inbuilt conversion tool, this entails major losses of data quality due to the fact that MusicXML is unable to represent key aspects of the Ottoman-Turkish musical system. Likewise, mainstream notation programs are not designed to represent or encode non-Western musics in a semantically adequate way, leading to the heavy use of special symbols and other workarounds which are visually acceptable but logically defective.

We have attempted to remedy some of these problems by developing methods for converting Sibelius transcriptions of Ottoman music into CMN and MEI. Nonetheless, the process could be vastly improved through the development of existing notation software to accommodate both the musical and scholarly requirements of Ottoman music. As a starting point, this might entail the inclusion of the AEU accidentals within the Sibelius symbols repertoire and the possibility to create non-standard key signatures, including not only graphic but also semantic information. Alternatively, it might be desirable to integrate additional features into Mus2 in order to support the scholarly and MEI-oriented goals of the CMO Edition. However, until such features are made available in existing score-writing programs, the MEI schema offers some possible solutions to the challenge of encoding Ottoman music in a way that is both visually and semantically acceptable.

MEI is a suitable means of achieving these goals because it already supports the demands of critical music edition and offers a continually growing set of tools for further processing. In particular, because it is designed as a modular framework for music encoding [3] it also includes a customization method that enables the re-use of basic core concepts alongside adaptations to handle new music features, while the Sibelius-to-MEI plug-in provides support for the conversion of standard Western music notation. On this basis, and with further customizations of the plug-in, it is possible to preserve most of the intended logic for the subsequent conversion process.

5. Created as a part of the CompMusic research project.

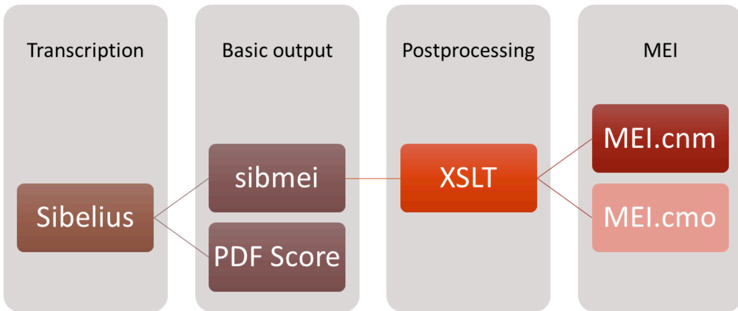


Figure 4: Encoding workflow

Since Sibelius was chosen as the score-writing software for the CMO Edition, the use of the Sibelius to MEI plug-in[6] was an obvious choice. Accordingly, the workflow begins with the transcription process in Sibelius. Sibelius generates the PDF scores and serves as the point of origin for the transformation into MEI. The `sibmei` plug-in provides an adequate basic conversion, on the basis of which the expanded features can be realized. Going further, two broad sets of criteria need to be met. Firstly, the special features of Ottoman music need to be taken into account. Secondly, additional features of the transcription reflecting the norms of a critical musicological edition have to be incorporated. In addition, assignment issues resulting from the extensive use of diacritics and editorial markup need to be resolved. The most reasonable approach to meet these varied requirements was to adopt a multi-stage conversion workflow.[7] Accordingly, it was necessary to keep the adjustments of the `sibmei` plug-in to a minimum, with only the project-related symbols and text styles being added to produce a preliminary output.

A postprocessing routine was introduced to encapsulate the project-related bug fixes and editorial transformation. It processes the *"flat"* [2, p.2], and in this case also defective MEI, with a series of XSLT scripts, run by a build software, to correct frequent assignment errors and to generally refine the encoding.[8] First of all, non-necessary data is removed from the score and the placement of the group signs and other editorial diacritics is corrected. The measures are labelled according to their position within an usûl cycle and invalid time signatures are replaced. In addition, the structuring of the encoding is enhanced: sections and subsections are created according to text markings in the Sibelius output and editorial markup is inserted according to editorial symbols.[9] Other diacritics are enhanced and linked to a symbol table.

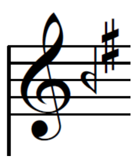



Figure 5: Manipulated key signature.

Perhaps the most important feature is the handling of the intended accidentals and key signature. Here, the workflow relies on a set of editorial guidelines to substitute the lack of support for certain pitch-related features. For example, since it is not possible to define custom key signatures in Sibelius, they are added by means of a visual manipulation of clef symbols and the semantic information is thus not stored within the score. As part of the transcription guidelines, a system of abbreviations has been established in order to transfer the intended key signature via the instrument label of the transcribed staff.

6. http://github.com/music-encoding/sibmei

7. Our approach is derived from the Lost Voices project (http://digitalduchemin.org). [2] offers a detailed description of their routines.

8. http://github.com/maxweberstiftung/CMO_MEI

9. E.g. <supplied> elements are inserted in correspondence with vertical bracket symbols.

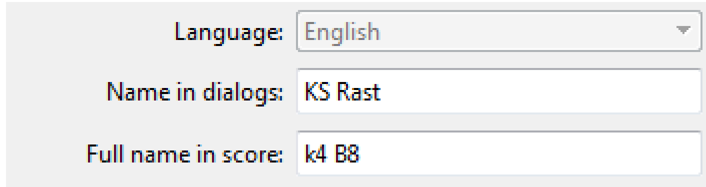


Figure 6: Key signature shortcut in instrument label.

Another advantage of this multi-staged workflow is the possibility to create multiple MEI files. After a series of general scripts, the process branches and transforms an interim version into different output files. In this way, it is possible to create different MEI versions that match the disparate needs of compatibility and enriched, adequate representation. First, we created a version validating with MEI.cmn to provide a basic minimum standard version. The cmn script mainly cleans non-valid accidentals from the shared transformation process and adds tremolo elements. But this output version consists mainly of visual markup and does not contain a semantically adequate representation of the transcribed music. The development of this version is still in progress, but nonetheless the current state of basic solutions could be built upon.

Towards the encoding of an edition of Ottoman music manuscripts

To achieve a semantically adequate MEI version, it is first necessary to find solutions for modelling the features of Ottoman music in a customization. Then, the postprocessing routine needs to be expanded to reconstruct the intended transcription from the `sibmei` output. The required changes deal mainly with core concepts of Ottoman music, i.e. **usûl** and **makâm**. It is thus necessary to transform the measure-based structure into an usûl-based score, to give the correct beat qualities of the usûl staff and to represent the basic pitch set of the makâm.

To build a rhythmic structure on the basis of an usûl, a three-level hierarchy needs to be implemented, with the *whole usûl cycle* as top level, the *divisions* as second level, and the *groups* as the lowest. Theoretically, this could also be solved by using `<section>` elements, but a semantically adequate model seems favorable, in order to distinguish the genre-related sectioning of a piece from the metric structure. The group element could, in a transcription in modified Western notation, be a control event, but in a representation of Hampartsum notation a parent element for the notes and rests would be more suitable.

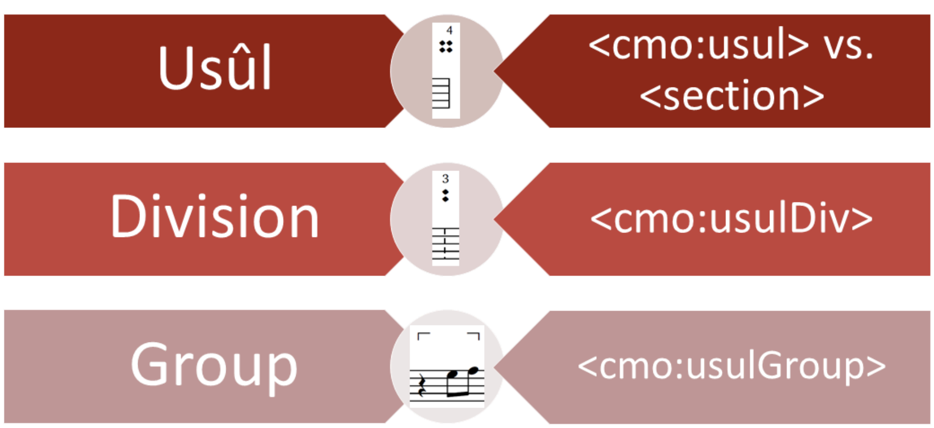


Figure 7: Hierarchy of metric structure

The signs indicating the end of usûl cycles also have to be considered in the transformation process. As they are an integral part of the editorial apparatus and provide important information about the source, they could not simply be replaced by a customized structure. The beat qualities in the usûl staff also need to be correctly transformed. Especially for composite beat qualities, solutions have to be found.

The other crucial task of the customized MEI is the semantic processing of the pitches of a makâm-based composition. In the CMN-validating version, the accidentals are only visually defined as glyphs, likewise the key signatures. The encoding of gestural accidentals is not possible, though one possibility is the usage of 'empty' accidentals, linked to the previous accidental with the SMuFL glyph.

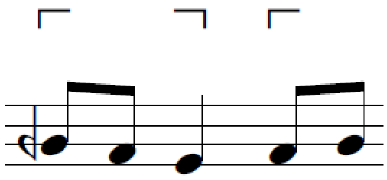


Figure 8: Accidentals in score

```
<beam xml:id="m-1174">
   <note xml:id="m-1172" dur="8" oct="4" pname="b" stem.dir="up">
      <accid glyphnum="U+E443" xml:id="m-1173"/>
   </note>
   <note xml:id="m-1175" dur="8" oct="4" pname="a" stem.dir="up"/>
</beam>
<note xml:id="m-1176" dur="4" oct="4" pname="g" stem.dir="up"/>
<beam xml:id="m-1178">
   <note xml:id="m-1177" dur="8" oct="4" pname="a" stem.dir="up"/>
   <note xml:id="m-1179" dur="8" oct="4" pname="b" stem.dir="up">
      <accid xml:id="m-1180" corresp="#m-1173"/>
   </note>
</beam>
```

Figure 9: Accidentals MEI.cmn

This workaround still seems to be defective when compared with a possible extension of the supported accidental values:

```
<beam xml:id="m-1174">
   <note xml:id="m-1172" dur="8" oct="4" pname="b" stem.dir="up">
      <accid accid="k" glyphnum="U+E443" xml:id="m-1173"/>
   </note>
   <note xml:id="m-1175" dur="8" oct="4" pname="a" stem.dir="up"/>
</beam>
<note xml:id="m-1176" dur="4" oct="4" pname="g" stem.dir="up"/>
<beam xml:id="m-1178">
   <note xml:id="m-1177" dur="8" oct="4" pname="a" stem.dir="up"/>
   <note xml:id="m-1179" dur="8" oct="4" pname="b" stem.dir="up">
      <accid xml:id="m-1180" accid.ges="k"/>
   </note>
</beam>
```

Figure 10: Accidentals MEI.cmo

However, thinking in more general terms, even if the project-related set of accidentals is manageable, the number of special accidentals in SMuFL is overwhelming. In order to enable general support for unusual accidentals, simply modelling every possible accidental is not a workable solution. In this regard, one way to enhance interchangeability could be to provide the definition of the used accidentals within the file. In this case, it would also be necessary to provide further essential information—which is currently part of the critical commentary—within the encoding. Amongst other elements within the commentary, the 'pitch set' is of special importance for encoding the pitch- and mode-related aspects of a transcription.

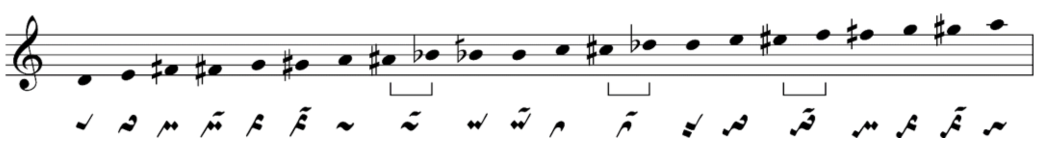



Figure 11: Pitch set

The pitch set provides information about how each pitch is transcribed from Hampartsum notation into staff notation in a given piece. Making these transcription rules available in a machine-readable form within the editorial apparatus will be an important step in the further development of an encoded critical edition of Ottoman music manuscripts.

Outlook

There are numerous possibilities for the future development of the encoding workflow of the CMO Edition. Most importantly, it is necessary to make the encoded data compatible with a standard MEI viewer such as Verovio, which currently does not support many of the expanded features in the CMO data. The basic conversion to MEI should be refined further to improve the quality of the output, and to resolve some outstanding issues in the encoding of Ottoman music, for example taking account of pre-twentieth-century systems of pitch representation, or establishing standards for the classification and encoding of makâms according to their modal characteristics. In this regard, it would be a useful step to close the gap with existing data retrieval and computational analysis projects that focus on Ottoman-Turkish music. Similarly, further refinement of the conversion process may help to make the encoded data compatible with other score-writing programs such as Mus2. The development of an MEI module for Hampartsum notation might be a starting point for the encoding of other (letter-based) notation systems in Ottoman or other non-Western traditions.

At a later stage, it would be useful to integrate the transcriptions more closely with the critical commentary (and with the edition of lyrics). This would entail a partial reorientation of the concept of the critical commentary away from a traditional, print-based model and towards a more fully digitized mode of presentation. Nonetheless, an intermediary stage might be achieved by linking critical comments to the score by means of division or group numbers. Within the Source Catalogue, another desideratum would be to display and compare incipits for individual pieces that have been transcribed and encoded.

A body of high-quality data would be the basis for achieving higher-level analytical goals. This might include the development of tools to search for particular modal or rhythmic phrases or structural features of a composition, which could enable semi-automated identification of concordances or, especially if philological data are also included, relationships between sources or scribes. Analysis of a larger body of data could be the basis for more ambitious analytical projects related to the processes of repertoire transmission or the historical evolution of makâm and usûl patterns. In conclusion, bringing together the specialized requirements of non-Western repertoires with the concepts and practices of critical music editing suggests new approaches to integrated, multi-faceted encoding problems and a possible road map for future research.

Acknowledgements

The research for this paper was completed within the framework of the long-term project “Corpus Musicae Ottomanicae: Critical Editions of Near Eastern Music Manuscripts” (CMO), funded by the Deutsche Forschungsgemeinschaft (DFG) and supported by the Max Weber Stiftung, Orient-Institut Istanbul and Westfälische Wilhelms-Universität Münster. The authors are grateful to their colleagues in all branches of the CMO project and at the Max Weber Stiftung for their support and advice.

Works cited

[1] Bozkurt, Barış, Ruhi Ayangil, and André Holzapfel. “Computational Analysis of Turkish Makam Music: Review of State-of-the-Art and Challenges” *Journal of New Music Research* 43.1 (2014): 3-23.

[2] Freedman, Richard et al. “Sibelius to MEI: From Editor’s Desk to Dynamic Editions on the Web”. Unpublished, 2014 (Slides: www.dropbox.com/sh/n8v844ks3pbbee8/AAC3OYI68Ch38YIBqgU8-YyVa/Sibelius_2_Mei_Example_9_9_2014.pdf - accessed 1 July 2018

[3] Hankinson, Andrew, Perry Roland, and Ichiro Fujinaga. “The Music Encoding Initiative as a Document-Encoding Framework” in *Proceedings of the 12th International Society for Music Information Retrieval Conference. Miami, 2011*, 293-98.

[4] Jäger, Ralf Martin. *Türkische Kunstmusik und ihre handschriftlichen Quellen aus dem 19. Jahrhundert*. Eisenach: Wagner, 1996.

[5] Karaosmanoğlu, M. Kemal. “A Turkish Makam Music Symbolic Database for Music Information Retrieval: SymbTr” in *Proceedings of the 13th International Society for Music Information Retrieval Conference. Porto, 2012*, 223-8 (https://dblp.uni-trier.de/db/conf/ismir/ismir2012.html).

[6] Olley, Jacob. “Writing Music in Nineteenth-Century Istanbul: Ottoman Armenians and the Invention of Hampartsum Notation”. King’s College London, 2017. PhD Dissertation.

[7] Olley, Jacob. “Towards a New Theory of Historical Change in the Ottoman Instrumental Repertoire” in *Theory and Practice in the Music of the Islamic World: Essays in Honour of Owen Wright* ed. by Rachel Harris and Martin Stokes. Aldershot: Ashgate, 2018, 22-41.

[8] Richts, Kristina and Axel Teich Geertinger. “Implementing the Functional Requirements for Bibliographic Records (FRBR) Model in MEI” in *Music Encoding Conference. Proceedings 2013 and 2014*, ed. by Perry D. Roland and Johannes Kepper. (Mainz-Munich: Music Encoding Initiative - Bayerische Staatsbibliotek, 2015): 17-26. Online resource: langzeitarchivierung.bib-bvb.de/delivery/DeliveryManagerServlet?dps_pid=IE2860815 also reachable at music-encoding.org/archive

[9] Seidel, Heinz-Peter. “Die Notenschrift des Hamparsum Limonciyan: Ein Schlüssel” *Mitteilungen der deutschen Gesellschaft für Musik des Orients* 12 (1973/74): 72-124.

[10] Şentürk, Sertan and Xavier Serra. "A Method for Structural Analysis of Ottoman-Turkish Makam Music Scores" in *Proceedings of the 6th International Workshop on Folk Music Analysis, Dublin, June 2016*, ed. by Pierre Beauguitte, Bryan Duggan, and John Kelleher, p. 39-46 (https://arrow.dit.ie/fema/19/)

[11] Ünal, Erdem, Barış Bozkurt, and M. Kemal Karaosmanoğlu. "A Hierarchical Approach to Makam Classification of Turkish Makam Music, Using Symbolic Data" *Journal of New Music Research* 43.1 (2014): 132-146.

[12] Wright, Owen. "Aspects of Historical Change in the Turkish Classical Repertoire" *Musica Asiatica* 5 (1988): 1-107.

The Development of Computing Technology and Its Influence on Music-Analytical Methods and Encoding: 1940s through 1980s

Nico Schüler
Texas State University
nico.schuler@txstate.edu

Abstract

This paper summarizes the development of computing technology from the 1940s through the 1980s and draws parallels to its influence on the development of music encoding and music-analytical methods. The emphasis on how the developments of computing and computers influenced the development of computer-assisted music analysis (and, thus, various representations) is related to the fact that this knowledge is often absent in contemporary discussions of music encoding and of computer applications in music analysis.

Historical sketch

During the 1940s, several electro-mechanical and the first electronic computers were developed: Z3 (1941) by Konrad Zuse and the Harvard Mark I (early 1940s) by Howard H. Aiken. Other computers of the 1940s were the ENIAC (1946), EDVAC (1949), and EDSAC (1949), the latter of which became the first full-scale universal digital computer with saved programs based on John von Neumann's principles. Although these early developments were very exciting at the time, those early computers were not easily accessible and not used for any music projects. Some scholars did use, however, electro-mechanical calculating machines. For example, in 1949, Bertrand Bronson described a procedure for using an electro-mechanical calculator to carry out a comparative study of British-American folk tunes. He used punched cards for encoding general information, regional information, and musical characteristics. Then, the sorting machine was able to automatically pick out cards with desired characteristics.

The late 1940s and early 1950s saw the rise of the Card Programmed Calculator and the development of several full-scale computers. IBM developed the magnetic drum for the main memory. The first commercially available computer was the UNIVAC (1951). The 1950s also saw a change from mathematical calculations to data processing. Thus, the first music applications date from the mid-1950s: Frederick P. Brooks conducted an analysis-synthesis-project at the Computation Laboratory at Harvard University, using a Harvard MARK IV (1952). For this project, high-order probabilities of 37 hymn tunes were calculated. Those probabilities were then used for the synthesis of new melodies, using Markov chains of orders one through eight. In the late 1950s, Hiller and Isaacson suggested several computer applications to music analysis: statistical and information theoretical applications, analysis of musical similarity, pattern search, analysis of sounds and their physical constitution, optical music recognition; and based on analytical results: realization of *continuo* and figured bass and to complete part writing, missing parts could be reproduced based on statistical style analysis, systematic generation of musical materials for teaching purposes. Most or all of these suggested projects were launched during the late 1950s and early 1960.

In the early 1960s, IBM continued with a standardization process to achieve compatibility among all their computer components as well as their software components. The result was their System/360 in 1964. However, programming turned out to be more costly than the hardware itself. Further developments were fully integrated circuits, giant number-crunching computers, and time-sharing computer systems. The military pursued real-time computing, and the airline industry developed a reservation system. The UPC was developed by the early 1970s. First scanners were manufactured by IBM and NCR. The possibility of quickly processing a large amount of data with new computer techniques was recognized in the 1960s, especially by US scholars in musicology and music theory. Allen Forte described a computer project for the determination of similarities and differences of sets, and Gerald Lefkoff (described a system for score-derived musical models. Other analytical methods during the 1960s involved the application of statistical and information-theoretical measurements. Many music codes were developed during the 1960s: MUSTRAN, ALMA, DARMS, IML, etc.

The 1970s brought faster integrated circuits, random-access disk storage units, semiconductor memory, more effective time-sharing, communication-based on-line computing, and virtual memory. The first personal computer came on the market in 1977: the Apple II. With the personal computer, new programming languages became important, especially BASIC, FORTRAN, C, and LISP. From its beginning, computer-assisted music analysis was frequently used in folk song research. Computer-assisted analysis continued to be centered on the less complex folk songs, even though computers became more and more powerful, starting to enable the greater complexity of multi-part art music. In the 1970s, the search for a musical grammar was probably the most important development in computer-assisted music analysis. Furthermore, existing (statistical and information-theoretical) methods were refined, separate measurements were converted into complex, multi-factor analyses, and, most importantly, the core of computer-assisted, analytical methods was extended by psychological and set theoretical approaches as well as approaches drawing on Schenkerian analysis and generative grammars.

During the 1980s, computer hardware costs started to fall dramatically. Changes in the nature of data processing have generally occurred invisibly. More sophisticated applications are possible, and real-time systems as well as time-sharing systems have become common. Large computers / supercomputers also continued to develop, and most music-analytical applications continued to be run on such computers. Programming languages and program design made also great strides, and MIDI was developed. Many research projects conducted during the 1980s aimed at the development of new methods of computer-assisted music analysis. Some projects discovered new possibilities to simulate human cognition and perception. Approaches that had already established historical precedents, especially those drawing on statistics and information theory, also developed further. The main focus in terms of the type of music continued to be folk music, but more and more projects focused on either set-theoretical or statistical and information-theoretical analysis of art music. The development of new models of syntactic structures in linguistics suggested new ways to describe syntactic structures in musical compositions. Specifically the application of concepts derived from Noam Chomsky's generative-transformational grammar to the analysis of music was of special importance to several developments in music theory.

Bibliography

[1] Bronson, H. Bertrand. "Toward the Comparative Analysis of British-American Folk Tunes" *Journal of American Folklore* 72 (1959): 165-91.

[2] Brooks, Frederick, et al. "An Experiment in Musical Composition" *IRE Transactions on Electronic Computers* 6 (1957): 175-82.

[3] Ceruzzi, Paul. *A History of Modern Computing*, 2nd edition. London, UK and Cambridge, MA: MIT Press, 2003.

[4] Hiller, Lejaren A., and Leonard M. Isaacson. *Experimental Music. Composition with an Electronic Computer*. New York: McGraw-Hill, 1959.

[5] Lefkoff, Gerald, ed. *Papers from the West Virginia University Conference on Computer Applications in Music*. Morgantown: West

Virginia University Library, 1967.

[6] Schüler, Nico. “Methods of Computer-Assisted Music Analysis: History, Classification, and Evaluation”. Michigan State University, East Lansing, MI, 2000. PhD Dissertation.

[7] Selfridge-Field, Eleanor, ed. *Beyond MIDI. The Handbook of Musical Codes*. Cambridge, MA: MIT Press, 1997.

Creating an Import and Export Infrastructure for LilyPond

Jan-Peter Voigt
GNU LilyPond, openLilyLib, beautifulscores.net
jpv@schoene-noten.de

Abstract

This paper discusses the implementation of an infrastructure for LilyPond that allows import and export of different file formats, to connect LilyPond with the rest of the music-engraving world and make it a valuable chain link in a multi format environment.

Introduction

Let me first reflect on somethning that Urs Liska and I have written in preparation of this conference:

> *While LilyPond is known for its exceptional engraving quality, format interchange with other tools has not been focused by the LilyPond community until a few years ago. There are some tools to convert scores, but LilyPond had always been regarded as the last point of a one-way street.*

Therefore, it doesn't suit well for archiving and some other tools we already heard about today. And it is no surprise that it has not even been mentioned in one of the presentations we already heard today. But this is a pity because beside its exceptional engraving quality LilyPond is a powerful tool based on a comprehensive scripting engine. Fortunately the severe limitation of not being able to work with other prevalent file formats has by now been acknowledged, and the need for two-way conversion options with various other formats is clearly recognized.

Below, I will first of all show up the current state of import and export options in LilyPond; then I will explain some strategies to implement foreign file exchange, name some inherent problems and have a look on doing it with Humdrum, MusicXML and MEI.

Current state of import and export

Import options

By now import is only possible using an external converter because LilyPond can't read other formats by itself. So the tools to use foreign formats in LilyPond have to produce source files to be compiled afterwards.

To name a few: The tools delivered with LilyPond, like `musicxml2ly`[1] and its siblings, are used to convert MusicXML and some other formats into LilyPond format. MEILER[2] is an interesting XSLT-based solution by Klaus Rettinghaus to transcode MEI. Denemo[3] is an Application intended as a graphical frontend for LilyPond that is able to import MusicXML. And there is `hum2ly`[4], an application prototype or sketch by Craig Sapp that creates LilyPond code from a Humdrum file.

Thus, it is possible to convert MusicXML and MEI to LilyPond, but the fact that LilyPond is not able to read the files by itself still leaves a gap because external tools don't have access to the internal objects of LilyPond.

Export options

LilyPond writes out Postscript, PDF, SVG and MIDI. Bitmap formats like PNG are created implicitly with ghostscript. But these are graphical representations that do not contain any musical data or at least some structural information.

In Google's Summer of Code 2015[5] there has been an attempt to integrate support for MusicXML. But that approach depends on `guile` version 2 and therefore didn't make it into the main development branch yet. `guile` is the scheme interpreter LilyPond is build upon. And even though quite a number of developers are working on switching to `guile 2`, there are still performance issues preventing a stable release.

There is an option in recent versions of `Frescobaldi`[6] that converts from LilyPond source to MusicXML. That functionality is also available as part of `python-ly`[7] the python core toolset of `Frescobaldi`. It can be used in shell-scripts for fast batch-processing. The disadvantage is that the source has to be formatted in a special way without any scripting parts. Therefore this tool does not work in a generic way for any LilyPond file.

Strategies

If the situation regarding import and export is still unsatisfactory. But I want to describe ways to change that and to make LilyPond a valuable chain link in a multiformat environment.

Integration of import and export

One might ask: "Why is it that difficult to integrate support for well-known and well-documented formats into LilyPond?"

1. http://lilypond.org/doc/v2.18/Documentation/usage/invoking-musicxml2ly.html
2. https://github.com/rettinghaus/MEILER/
3. http://denemo.org/
4. https://github.com/craigsapp/hum2ly
5. https://www.google-melange.com/archive/gsoc/2015/orgs/gnu/projects/david156.html
6. http://frescobaldi.org/
7. https://github.com/wbsoft/python-ly

One reason is that a LilyPond file is not bound to a fixed file structure. An instrumental part for example may be written inside the score construct *or* read from a variable defined in another file. And it may use the scripting engine. In fact a LilyPond file contains a program that is executed to layout the score on a virtual canvas. You can see two versions of the same score in Figure 1. This allows to keep projects organized, to reuse the content and to separate it from the layout.

```
\score {
  \relative { \key g \minor \time 3/4 c''4. bes8 g4 \bar "|." }
  \addlyrics { Frie -- den gib'. }
}
```


```
melody = { c''4. bes'8 g'4 }
global = { \key g \minor \time 3/4 $(skip-of-length melody) \bar "|." }
words = \lyricmode { Frie -- den gib'. }
\score {
  <<
    \new Staff = "singer" <<
      \global
      \new Voice = "melody" { \melody }
    >>
    \new Lyrics = "words" \lyricsto "melody" { \words }
  >>
}
```

Figure 1: Defining music inside or outside the score

But an *external* converter has to write a LilyPond program. And so there is no one-to-one relationship between the source and the target and the converted code depends heavily on both, the quality of the code-templates and the LilyPond-version in use.

If LilyPond would read foreign files by itself it could create LilyPond source code that fits the current version. It could wrap the content into a script and prepare an environment with variables named by the user. Of course it could add source-links to the generated objects to allow point-and-click.

Not having point-and-click into foreign files is not a real impediment. And as long as the code-templates are curated and well designed snippets the import will work satisfyingly. My intention is absolutely not to disregard the work of those who write external tools—like for example Klaus Rettinghaus with Meiler!

Importing data into LilyPond

Lets have a look on a real world example — lets import a humdrum file. I created a library[8] to be loaded by guile which uses the code found in Craig Sapps `hum2ly` tool. It provides a function `\scoreHumdrum` which translates the content of a `.krn` file into a string. This string is then fed into the parser and the score is typeset as an artefact of the LilyPond file containing `\scoreHumdrum`.

```
\scoreHumdrum "chor001.krn"
```

Figure 2: Importing Humdrum into LilyPond

8. https://github.com/jpvoigt/libguile_humdrum

This proof-of-concept actually wraps the external conversion in a LilyPond command. So it still transfers the data in text form. The next step will be to create directly the resulting objects instead of their string representation. The converter could add an origin-attribute with the filename, line- and columnnumber to the objects. Whenever LilyPond finds such an attribute, it adds a Link in the PDF to allow point-and-click like shown in Figure 2.

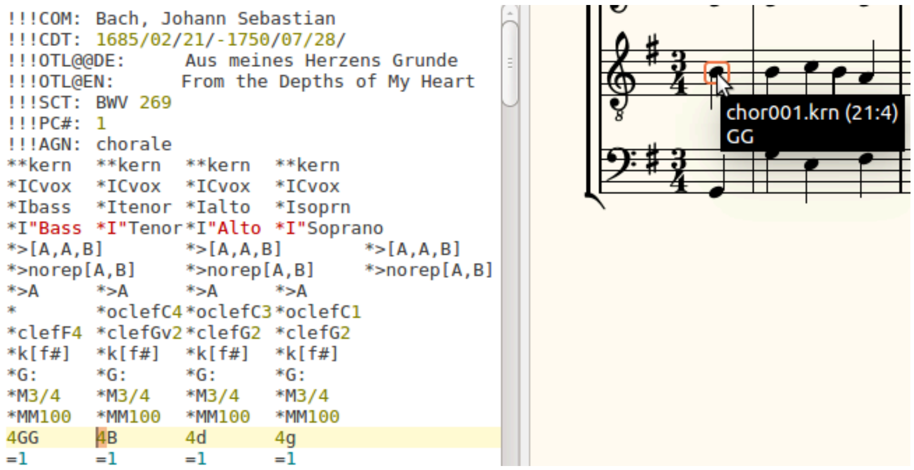


Figure 3: Point-and-click in Frescobaldi with a Humdrum file

This demo works with Humdrum. As a side note I'd like to mention that my first impression of Humdrum was that it is the opposite of human-readable and more like coding Fortran or Assembler than typing music. But I have to admit this assessment was pretty much biased by fondness for my own toolchain. Humdrum is a stable and consistent format and now I know *some* use cases where it is the format of choice. So building a bridge from Humdrum to LilyPond would add a very valuable option in a multiformat setup.

Other containers suit other use-cases. And after we have extended LilyPond with the ability to read Humdrum we might want the same for MEI. Both import routines will share a lot of code and it is obvious to provide that shared code behind a common API. I propose an entry point like the following example:

```
addEvent(measure, moment, system, voice, origin, event);
```

Figure 4

This method inserts a music-element, for example one note, into a two-dimensional grid. One dimension is the time identified by the bar-number and the moment inside this measure. The other dimension is the vertical position. That is not the pitch, but the context to place the event, identified by the staff and the voice — or in MEI terms the layer — inside this system. This grid provides a reasonable starting point to build a score from it. And with including the origins for each event we can link target and source to provide the useful point-and-click feature.[9]

The structure of the grid splits the time into measure and moment like MEI and MusicXML do. While we parse an MEI file it will be natural to call the proposed function and afterwards typeset the score with LilyPond.

9. Excuse me for alluding to point-and-click again and again. My intention is that a tool like `Frescobaldi` can make a suitable tool to edit foreign document types to be typeset by LilyPond.

Exporting data out of LilyPond

The situation regarding export is more complicated. To convert a file that is to be imported by another application, say MusicXML for Finale, the converter needs to look for scripted parts. That means an external application will need to mimic the LilyPond parser and part of the application logic to convey the whole score into the target file.

If LilyPond would export files by itself it could normalize and prepare the music data while typesetting the score and afterwards delegate the actual serialization to a module written specifically for the target format. Additionally it might trigger external applications like for example the Humdrum tools.

To export a score we need a collection of so called engravers which catch events and track the timing. I recently created a module that collects all notes, rests and some other elements into a grid sorted by measure, moment, staff and voice context.

That sounds familiar doesn't it? The grid is build as a tree where most elements can be found with a path consisting of measure, moment, staff and voice. And when processing the LilyPond score is finished the tree is filled with all the elements we caught. That tree is a normalized representation of this particular piece of music and it is given to the requested export function to iterate over and serialize it into the target file.

I have created two functions (export callbacks) to export Humdrum and MusicXML. The export procedure is implemented as an OpenLilyLib package.[10]

```
% load the export-package
\include "oll-core/package.ily"
\loadPackage lilypond-export
% define some music
music = { c'' } % the music
% export the music to 'music.krn'
\exportMusic \default hum \music
```

Figure 5: exporting music `music.ly`

The first argument of the `\exportMusic` function is the export filename without the suffix. With `\default` it uses the basename of the current file. '`hum`' calls for Humdrum export, '`xml`' would start the MusicXML export. On the agenda, but not developed yet, are '`mei`' and '`lily`' to export MEI and LilyPond respectively.

The tiny example in Figure 5 exports[11] the Humdrum file shown in Figure 6. The resulting MusicXML file is shown in Figure 7.

10. https://github.com/openlilylib/oll-core and https://github.com/jpvoigt/lilypond-export.

11. With just one note this doesn't look very fascinating, but the Github project provides a more elaborate example.

```
**kern
*staff1
=1
4cc
*-
```

Figure 6: The note C'' exported to Humdrum by LilyPond

```xml
<score-partwise version="3.0">
   <part-list>
      <score-part id="P1">
         <part-name>Part 1</part-name>
      </score-part>
   </part-list>
   <part id="P1">
      <measure number="1">
         <attributes>
            <divisions>64</divisions>
         </attributes>
         <note>
            <pitch>
               <step>C</step>
               <octave>5</octave>
            </pitch>
            <duration>64</duration>
            <voice>1</voice>
            <type>quarter</type>
         </note>
      </measure>
   </part>
</score-partwise>
```

Figure 7: The note C'' exported to MusicXML

Now I proved that it is possible to export MusicXML and Humdrum from LilyPond. Not all Elements are transcoded yet. I will complete that soon. The same applies for the MEI export.

Both export functions of the demo are created without coding any c++ but using only Scheme. To construct a stable exporter it would be reasonable to use libraries that are specialized to serialize a certain format. It doesn't make sense to reinvent the wheel over and over again. But that shall be part of upcoming projects.

The main problems remain the same irrespective of the used programming language. There are severe differences in the way the timing is handled and how Voices are separated in LilyPond and in other contexts. In MusicXML for example multiple voices are simulated using a disturbing mechanism to jump around in time. LilyPond on the other hand doesn't need any separation of measures and barlines are created separately by the respective timing engraver.

Prospects: Lychee

I am not going to explain each technical detail on how to construct the exporter functions, but I would like to point out something else. The grid- or tree-structure used to transcode and export files is similar to the one I proposed for importing files. This implies that a file might be exported right after importing it—just like `pandoc` does with text encoded in different markuplanguages. This is what the nCoda project has created Lychee for. One might ask if this strategy competes with nCoda, namely the Lychee project: the answer is no.

The nCoda project uses a format called Lychee-MEI which is an adapted version of MEI. It will be natural to serialize the grid into Lychee-MEI and to deserialize it again. As I said at the beginning, I want to build bridges and connect LilyPond. And if I understand it correctly the LilyPond inbound-converter of Lychee will be source based. So this import and export API might be a missing link between Lychee and LilyPond to circumvent the problems with source-based conversion to and from LilyPond.

Conclusion

In this paper I presented strategies to convert to and from prevalent music file formats with LilyPond. Some external tools already exist to convert to LilyPond-format, but I explained the benefit of integrating import into LilyPond. In addition to some attempts tried in the past, an idea is sketched of how to export music from LilyPond. The next steps will be, on the one hand, to develop stable export modules for Humdrum, MusicXML and MEI. On the other hand, to integrate conversion of external file formats into LilyPond.

Introducing a Corpus of French Compositions for Exploring Social Interaction and Musical Change

Jane Harrison
Istanbul Technical University
jane.harrison39@gmail.com

Farhan Khalid
Takhleeq App Development
farhan@takhleeqhq.com

Abstract

In this paper we offer a new corpus and a new method for how to consider musical style from a sociological perspective, by looking at the phenomenon of *debussysme* in early 20th-century France. We focused on the solo piano scores of the "Apaches" subset of potential *debussystes*, comparing them to a randomly selected control group of their peers. Due to the challenge of manually correcting these complex scores, we experimented with leaving the errors generated during the OMR process uncorrected in our MEI files, planning to correct them progressively in later stages of hypothesis testing. We then validated the error level and successfully analyzed nearly all of the scores for parallel 5ths and meter changes using customized MEI search programs. Even when taking the estimated error levels into account, a statistically significant difference was discovered between the Apaches and control groups. In addition to presenting these initial findings, our description of the methodological challenges of encoding and analyzing these complex scores might serve as a model for how to deploy a multi-hypothesis corpus study involving difficult scores over several, more manageable steps.

Introduction

Following Meyer's well known definition [4, p.1], we believe that musical style can be conceptualized as the replication of patterns in the choices and behaviors of creators of musical sound (whether in written or directly sonic form) that are influenced by a variety of types of constraints. While many studies have examined the relationships between certain internal changes in music and cognitive constraints, the role of social structure in such changes has received little attention. This paper presents a new corpus and a new method for analyzing the relationship between social interaction and the development of musical style, by looking in particular at the phenomenon of *debussysme* in early 20th-century France. In this first stage of empirical research, we focus on the interactions of a specific social circle named *les Apaches*or the Apaches, avowedly rebellious artists who came together around their admiration of Debussy's music. Encoding and analyzing the complex solo piano scores we assembled for our corpus study has presented several methodological challenges, and in addition to sharing some preliminary results, we will also describe some of the solutions for these problems. Especially relevant at this MEI annual conference are the customized search tools that we have begun to develop specifically for the MEI format, and the ways in which we have tailored our encoding process for these tools. We hope that in addition to adding to the body of scholarly knowledge with our findings, these descriptions of our methodology might serve as a model for how to deploy a multi-hypothesis corpus study involving complex scores over several, more manageable steps that might be of use to interested colleagues.

In verbal written sources, many French music critics and composers active around the year 1900 mentioned the existence of composers following Debussy's stylistic lead (*debussystes*), but no scholar had previously subjected this claim to serious investigation. Based on extensive score analysis and examination of other primary source material, Harrison [2] put forth the possibility that an identifiably distinctive compositional trend encompassing several hallmarks of Debussy's innovative idiom did take place among a sizeable group of French composers, with the period of greatest activity occurring between 1900 and 1920. Furthermore, many individuals seemed to adopt these techniques at the same time that they started or strengthened friendships with other composers who were already using them, a pattern reminiscent of the role social structures often play in the diffusion of new ideas and technologies [7] and in language change [3].

Methodology

To test this hypothesis, our two corpora consisted of all available solo piano pieces from the Apaches (n=61) and a randomly selected corpus of solo piano scores available as PDF files on IMSLP (n=108) from other French composers in the same time period of 1895-1925. Piano scores were utilized in this initial test because during this era in France 1) many composers were still writing solo piano pieces and 2) composers often used this genre for experimentation versus more conventionalized and/or economically risky genres such as opera or instrumental chamber music. Later studies will, we hope, turn to other genres. While the corpora are a great match for our hypothesis, they are unfortunately difficult to work with. The onset of a modernist "research" approach to music composition in France in the late 19th century involved pushing beyond common-practice boundaries in all the conventional musical parameters, and the notational system had not yet been appropriately updated (Ravel and Debussy's pieces were commonly criticized by conservative writers with terms like "recherché" or "raffiné"). Within the control set as well as the Apaches corpora we discovered many examples of modernist scores whose tricky note combinations and generally high level of detail might be easily readable for a contemporary pianist, but are downright perplexing to an OMR program like Photoscore. Consider, for example, one well-known score that we encoded with Photoscore 8, Ravel's triptych *Gaspard de la nuit*. Imagine how difficult it was for Photoscore to properly recognize the stave-crossing runs, the sudden switches between note value subdivisions like 32nd-notes and nontuplets, passages full of chords foreign to the notated key, or the occasional unmetered ad libitum bars of "Scarbo"?

If we wanted to test our hypothesis about musical style and social structure as it relates to the Apaches' piano music, the notational complexity of the required score corpora were an unavoidable problem to be solved. Turning to an extant encoded corpus might have saved time and effort, but it would have also meant drastically rethinking the hypothesis. Additionally, a robust set of early-20th-century French piano music is worth the effort to encode and make available to other scholars, given the popularity these pieces still have in the concert repertoire and in music historiography. After a few years of making little progress in encoding these pieces, we realized that our project would be doomed if we did not recalibrate and find a way to move past the sheer psychological difficulty of perhaps spending hundreds of hours correcting the encoded scores only to find that our hypothesis was not supported by the data. In theory a negative result is just as good as a positive one, but there is a moment when the rubber of theory meets the road of the real world. To put our car in gear, we recalibrated our ideal for the digitized version of these scores, coming to view the encoding as a multi-stage process that could still allow us to test certain hypotheses along the way. Instead of working immediately towards a beautiful and complete score such as would be the goal of a critical or performance edition, we merely aimed for enough accuracy of the MEI code so that our customized search tools could complete a search that might rise to statistical significance. If this initial test showed promise, it would boost our enthusiasm for manually correcting the entire corpus. The enduring concept of music scores that Lydia Goehr has described as "complete and discrete, original and fixed, personally owned units" [1, p.206], developing in Germany in the early 19th century and persisting in today's art music culture, becomes irrelevant in our "good enough" encodings. We also lose the visual beauty of the original scores and the historical trace of the printing art as a result of this approach.

After converting PDFs (mostly downloaded from IMSLP.org) of all 169 scores into Music XML with Photoscore 8, we estimated the accuracy of the uncorrected encoded scores by validating 100 randomly selected beats each in the control and Apaches groups. We compared the temporal and pitch accuracy in terms of several different error categories for these randomly selected beats in the XML files to the original PDF. Accuracy was initially computed as a simple percent correct, and then as a 95% confidence interval. After analyzing the two corpora for parallel 5ths pairs and meter changes, we applied the confidence interval to properly adjust our t-test of difference between the two groups, given the amount of errors we estimated them to contain.

	Wrong PC	Missing PC	Wrong duration	Wrong meter	Wrong key.sig	Wrong articulation
Apaches	31	25	33	8	27	10
Apaches at lower 95% CI	0.5937	0.657	0.5731	0.85	0.6357	0.8256
Control	25	16	29	16	24	7
Control at lower 95% CI	0.657	0.7558	0.6146	0.7558	0.6677	0.8625

Table 1: Selected Error Validation Results with Confidence Intervals.

The above table shows selected validation results. Columns 1 and 2 were applied to our parallel 5ths result, and column 4 was applied to the meter changes result. The 2nd and 4th lines show the lowest possible accuracy proportion for the two groups, respectively, with 95% confidence. Interestingly, although we assumed that the Apaches scores would be more complicated and as a result have a higher number of errors, for many of the error categories the two groups did not have a statistically significantly different percentage according to p-value. One reason for this surprising statistical similarity may have been that the PDF scores available on IMSLP for the three most prolific composers in the Apaches group (Ravel, Florent Schmitt, and Déodat de Séverac) were generally of high quality, whereas the control group contained many low-quality scans. The lower quality can be attributed to two steps in the document generation, one of which goes back to turn-of-the-century French music publishing: many control-group composers were published by smaller publishing houses whereas houses with greater resources and prestige such as Durand and Senart published many of the Apaches scores (Jacques Durand was in fact an important relational node in the *debussyste* network). Furthermore, the six Apaches composers have generally been better remembered and more often performed since their deaths than control-group composers like Justin Clérice, Reynaldo Hahn, or Rene Lenormand, whose already likely poor quality original scores have not been well preserved. There might only be a single extant copy of a forgotten score, and it might have been scanned without the effort to make it as clear and high-resolution as possible. This is a small yet clear example of how combining quantitative and qualitative methodology can yield rich outcomes. We see that as a kind of byproduct of our methodology of encoding with errors and making our error check, we have glimpsed some meaningful details about the nature of the histories of musical scores. The process funneled our minds towards specific differences between the Apaches versus control scores and their possible explanations. As many of the control scores seem to be entirely unknown pieces today, our study even offers the basic benefit of bringing those scores to scholarly attention.

Encoding challenges

In order to guard the integrity of the sampling process for our control and experimental groups, complex scores could not be discarded. This was particularly germane to the control group, because whereas we took as many Apaches piano scores as possible, the control scores were randomly selected from IMSLP. If we began to discard poor quality or unusual scores from this group, we would have jeopardized the "random" nature and thus the representativeness of the control sample. As reviewed above, since the quality of a particular digitized score probably correlates with its current popularity, discarding messier scores would have introduced a bias towards the values of our own culture, whereas our control is meant to be a kind of average, neutral ground from the Apaches' own culture against which to view their work.

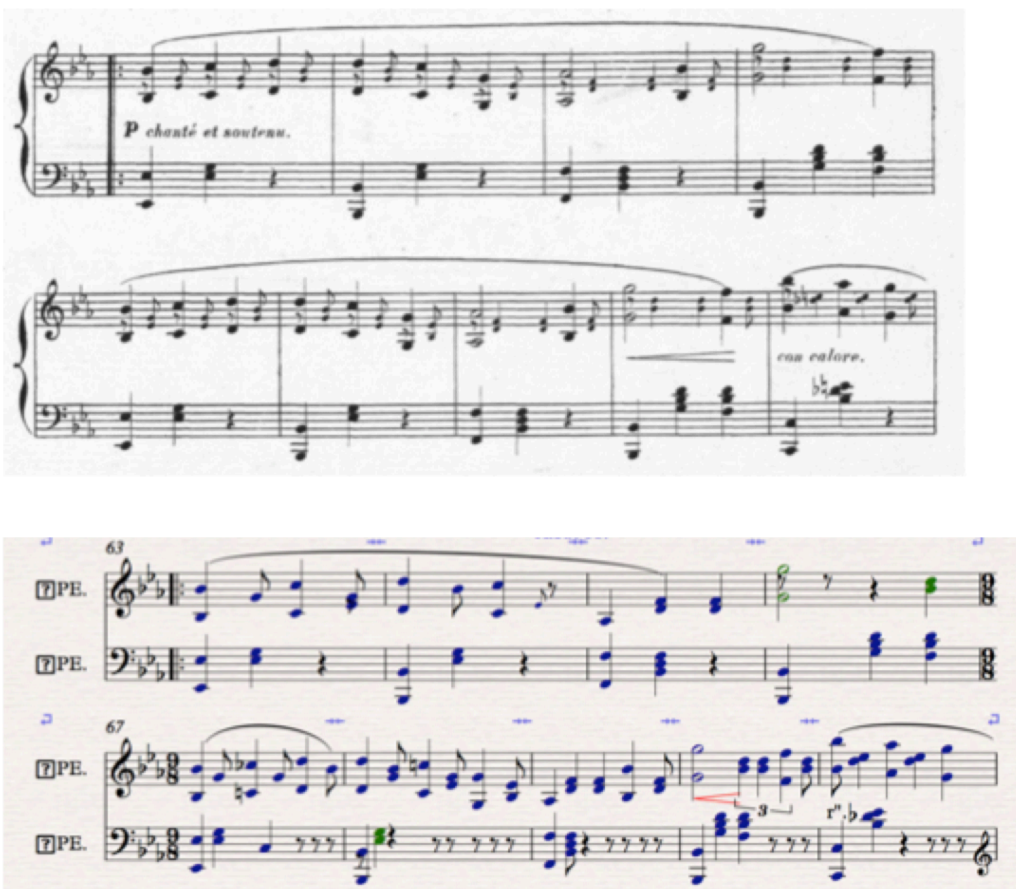



Figure 1: Justin Clérice, Heures grises, published by Bosc around 1901; top image shows the PDF and bottom image shows the XML file generated by Photoscore 8 and rendered in Sibelius 7.5

To meet our sampling conditions, we had to deal with scores of varying quality and with many pieces written in a burgeoning modernist culture with a taste for stylistic and notational complexity. In addition to optical misreading due to poor printing technology, low resolution, or damaged physical pages, we also saw many scores containing frequent meter and key changes, varied tuplets over varied timespans (including aberrations such as 15 notes over 2 beats), cross-staff beaming, grace notes, 3 or 4 simultaneous staves, up to 4 simultaneous voice layers, missing rests in layers, and extreme registers. In Figure 1, we can see the difficulty Photoscore 8 had with the overall low quality of the PDF and the persistent grace notes in one of the control group scores, Justin Clérice's *Heures grises*, published by Bosc. Simple enough for a good amateur pianist to sightread, Heures grises seems to confound the expectations of Photoscore's latest predictive algorithms. Contrastingly, although the PDF copy we started with was a pristine digitization of the Durand first edition, the cross-staff beaming employed in the "Prelude" movement of Ravel's Le Tombeau de Couperin, shown in Figure 2, still led to misalignment of parts and many rhythm errors in its OMR output.

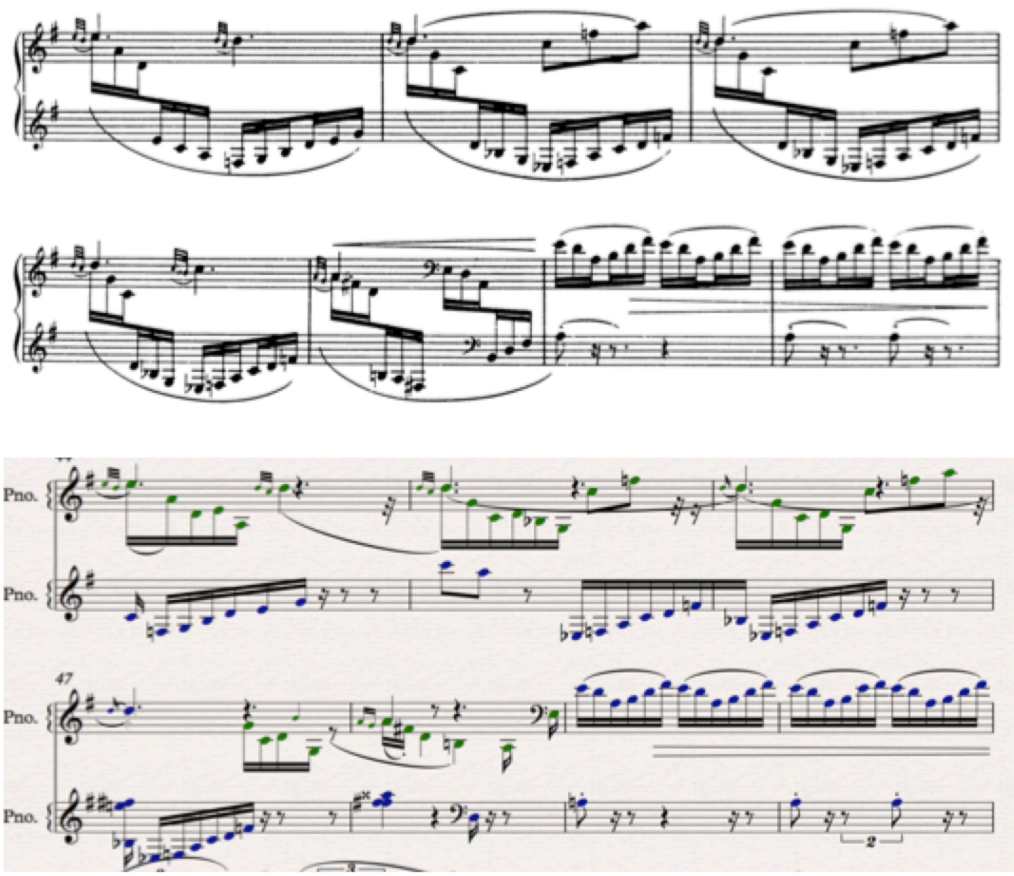



Figure 2: Maurice Ravel, Le Tombeau de Couperin, "Prélude", published by Durand in 1918; top image shows the PDF and bottom image shows the XML file generated by Photoscore 8 and rendered in Sibelius 7.5

Specialized computer analysis of MEI files

After encoding and estimating the error rates of both the Apaches and control corpora, we proceeded to convert them into MEI format and analyze them for parallel-5th pairs and meter changes. Harrison [2] had identified several potential *debussyste* stylistic elements, some easier to operationalize and parse computationally than others. We chose to make our first parsing tools for parallel 5ths and meter changes because they are both simple to define and categorically precise, making them easy for a computer algorithm to spot without ambiguity. Furthermore, in early 20th-century France they were both easily identifiable and culturally salient types of innovation. That is, all of the composers of our scores had been taught not to write parallel 5ths, yet a few composers began to dare to break this rule towards 1900. Debussy was one of the first, reportedly writing parallel 5ths into harmony exercises and improvising them in classes at the piano by 1884 [5, p.21-22]. Gradually, more and more composers began to use parallel 5ths until by 1920 it was a common sound, especially in exotic music. As is well known, European art music composers tended to use a single meter for one movement of a piece throughout the 19th century, and like parallel 5ths, changing the meter within the same composition became a daring, modern technique in the early 20th century that eventually achieved normative status within that culture. Taken together, parallel 5th pairs and notated meter changes index rebellious creative behavior nicely in the context of early 20th-century France in both the pitch and temporal domains. The Apaches artists were avowedly rebellious [6], and accordingly we might expect them to launch themselves into writing parallel 5ths and creating metric complications with special fervor.

Our analysis tools were tailored to read MEI files. We gravitated towards the MEI markup language because it has been developed specifically for academic use. Just like Music XML it is flexible, clear, and hierarchically intuitive for art music score encoding. Furthermore, MEI carries an education community license, with no strings attached by for-profit companies. One can easily see and manipulate MEI lines of code and even make new RNG schema files. All in all, it was worth our time to make the tools for MEI, knowing we wouldn't have to worry about losing access to the code or running into problems with future releases. MEI also encoded the vast majority of the errors in our scores, such as improper number of beats in a measure, without making the files corrupt.

Our preliminary MEI processor counts all successive pairs of diminished or perfect 5th intervals both within and between all simultaneous piano staves. It does this by building a multi-dimensional array for each simultaneous pair of notes that form a perfect 5th or diminished 5th, and these pairs are then compared to each other sequentially. In multiple tests, the parser in its final form showed that it could recognize pairs between any two voices located on any of the staves.

We processed each corpus as a whole batch, and the result displayed a single proportion of average number of incidences per measure, which we refer to as the "density" of the particular stylistic trait across the entire group of scores. We could then insert the two density numbers into a t-test. Except in 2 cases, all 169 uncorrected scores were successfully parsed. Table 2 below shows the densities of the Apaches and control groups, both as raw numbers and with an adjustment for the 95% confidence interval based on the error check for the relevant pitch or temporal categories shown in Table 1. A t-test for statistically significant difference between the two groups was significant for both counts as raw scores. However, when these counts were adjusted according to the worst possible confidence interval boundaries, only the meter change density remained significant. Because our hypothesis was that the Apaches would display a higher number of parallel 5ths and meter changes, we took the lower boundary of their 95% confidence intervals and the upper boundary of the control group confidence interval as the inputs for our adjusted t-tests. The CI boundaries actually overlapped for parallel 5ths, so that the control group showed a higher density than the Apaches! That is, the uncorrected scores contain the statistical possibility for confirming either corpus as having more parallel 5ths.

	Parallel 5ths Density	Par 5ths Density 95% CI	Meter Change Density	Meter Change Density 95% CI
Apaches	0.66	0.392	0.0967	0.0822
Control	0.44	0.67	0.0427	0.0565
p value	p=.005	p=.009	p=.00001	p=.03

Table 2: Results of Both Parser Tests in Terms of Density (average number of incidents per measure)

Next steps in the Apaches Project

Based on our initial tests of two stylistic dimensions in the uncorrected score corpora, we are enthusiastic to continue improving the encoding of the scores and analyzing them with MEI parser computer applications. The statistical significance of our analysis of meter changes as an index of metric ambiguity is especially promising. We also plan to investigate these results at the level of individual compositions, chronologically, looking at how composers' usage of parallel 5ths and notated meter changes changed over the period of 1888 and 1925 that is covered by our representative sample of scores. A composer might have included an especially high number of meter changes in one composition as compared to the baseline of the entire culture that we have now established, and we can try to discover the other conditions that might have led to this decision. We advocate the combination of methods both qualitative and quantitative, and on varying scales, as ideal for the empirical study of musical phenomena. As powerful as they can be, in our view, computational and big data analysis should never supplant the direct human analysis of musical data, because there are advantages to the small-scale view that takes the unit of analysis down to the level of the individual (the individual composer, composition, or even a single motive). Peering into the counts for individual composers and compositions against the batch counts can also give us a good idea of unusual individual instances, which we can then go back through and further analyze computationally or manually. To deepen the sociological dimension of the project, we envision pairing the analysis of the notes in scores with social data, especially positional information related to professional networks, artistic salons, socio-economic class, publishers, and the marketing of the scores. Such analyses will benefit from the MEI format, which gives us plenty of options for how to add sociological metadata to the encoded scores.

Our two MEI processors can be efficiently transformed to count other kinds of stylistic elements, such as certain vertical interval collections or certain chord voicings. After experimenting with parallel 5ths and parallel 4ths, many of the composers Harrison [2] identified as part of the earliest wave of *debussystes* began to write other kinds of parallel interval motions like parallel 7ths, and the parallel 5th counter could be easily modified to count them. After sufficiently revising our two current tools we plan to make them available soon at janeharrison.me, and to add to this collection of MEI computational analysis tools as our project grows.

One long term goal remains the complete correction of the 169 solo piano scores of our two corpora, and the analysis results we have reviewed here give us great hope that doing so will be a worthwhile endeavor. Because we will likely not complete this goal for several more years, we must still contend with how to keep submitting hypotheses to rigorous analysis tests while we work on making manual corrections. Our intended solution is to accomplish these two aspects in tandem, taking random sample sections from the scores in the corpus that are carefully tailored to each hypothesis test and correcting only those sections for that test. For further tests we will take a different random sample from the same score, likely catching new measures along the way until the whole encoded document is complete and free of errors. Once these pieces have been entirely corrected, we plan to make the corpus along with our computational analysis tools available to other researchers. To our knowledge, this would be the largest set of highly complex scores to be encoded to date, along with a handful of short piano pieces by Debussy and Ravel that have been already completed.

Corpus studies are popular right now among music scholars seeking to answer analytical and cultural questions with modern empirical rigor for good reason, yet certain pitfalls need to be conscientiously avoided, especially poor matching between hypothesis and test data. One might get a beautifully low p-value, but the significant numbers might not reflect a meaningful or even plausible story about the data at hand, given the limitations of its cultural provenance. For example, we could compare parallel 5ths among Apaches' scores with the scores of Palestrina, but that result wouldn't tell us how unusual the Apaches were as compared to their peers, in their own moment in history. While the need for carefully assembled corpora of musical data seems clear, our work with the debussystes reveals that this is not usually so easily done. Our way forward has involved choosing subsets of composers to stand in for larger groups of interest (Apaches for all *debussystes*), focusing on one genre at a time, and especially as we have shown in this paper, dealing with the encoding of highly diversified and detailed scores in multiple steps that coordinate closely with sampling and hypothesis testing. We would like to thank the whole MEI community for providing us with such a flexible and open format to help bring to life what once seemed to be impossibly out of reach.

Works cited

[1] Goehr, Lydia. *The Imaginary Museum of Musical Works. An Essay in the Philosophy of Music*. Oxford: Oxford University Press, 1992.

[2] Harrison, Jane E. "Fashionable Innovation: Debussysme in Early Twentieth-Century France" Ohio State University, 2011. PhD Dissertation.

[3] Labov, William. *Sociolinguistic Patterns*. Philadelphia: University of Pennsylvania Press, 1973.

[4] Meyer, Leonard. *Style and Music. Theory, History, and Idea*. Chicago: University of Chicago Press, 1989.

[5] Nichols, Roger ed. *Debussy Remembered*. London: Faber and Faber, 1992.

[6] Pasler, Jann. "A Sociology of the Apaches: 'Sacred Batallion' for Pelléas" in *Berlioz and Debussy: Sources, Contexts, Legacies*. Ed. by Barbara Kelly and Kerry Murphy. Burlington, VT: Ashgate, 2007.

[7] Rogers, Everett. *Diffusion of Innovations*. 5th ed. New York: Free Press, 2003.

TAMIR: A Toolbox for Recognition and Transcription of Music Manuscripts in Mensural Notation

Xuanli Chen
ESAT-PSI, KU Leuven, Belgium
xuanli.chen@esat.kuleuven.be

Yu-Hui Huang
ESAT-PSI, KU Leuven, Belgium
Yu-Hui.Huang@esat.kuleuven.be

Serafina Beck
Musicology, KU Leuven, Belgium
Serafina.Beck@arts.kuleuven.be

Radu Timofte
D-ITET, ETH Zurich
radu.timofte@vision.ee.ethz.ch

David Burn
Musicology, KU Leuven, Belgium
david.burn@kuleuven.be

Luc Van Gool
D-ITET, ETH Zurich
vangool@vision.ee.ethz.ch

The toolbox

We wish to introduce a toolbox that transforms handwritten mensural notation as found in 16th-century music manuscripts into modern music notation. To the best of our knowledge, this is the first such toolbox that can deal with the whole pipeline from symbol recognition to transcription. Along with the toolbox, we present our solution to the problem of ligature recognition, transcription, and encoding, which has remained an obstacle in previous mensural Optical Music Recognition (OMR) systems. We propose a visual encoding method as a reference to recognize and describe ligatures, then transform them to modern notation according to the time signature. Given an image as input, our toolbox recognizes the contents, transcribes them into modern musical notation, and encodes the result in a MusicXML file, with MEI support under development. Availability of the encoding enables further processing with third party tools.

Our toolbox is written in standard scripting languages (Matlab and Python), allowing for an easy installation under any operating system and for modifications according to a user's needs.

Our research interests

We regard our mensural music recognition and transcription tool as an opener leading to further analysis of mensural music. Encoding this information in an accessible format opens the possibility of mensural music analysis using data mining / machine learning methods. Among other things, our method allows the transcription of *tempus perfectum*, as well as additional capabilities such as searching for concordances, and counterpoint-based error checking. Our work has already provided a tool that saves musicologists time when reading and transcribing manuscripts. By releasing the source code, we hope to further support research into Renaissance music.

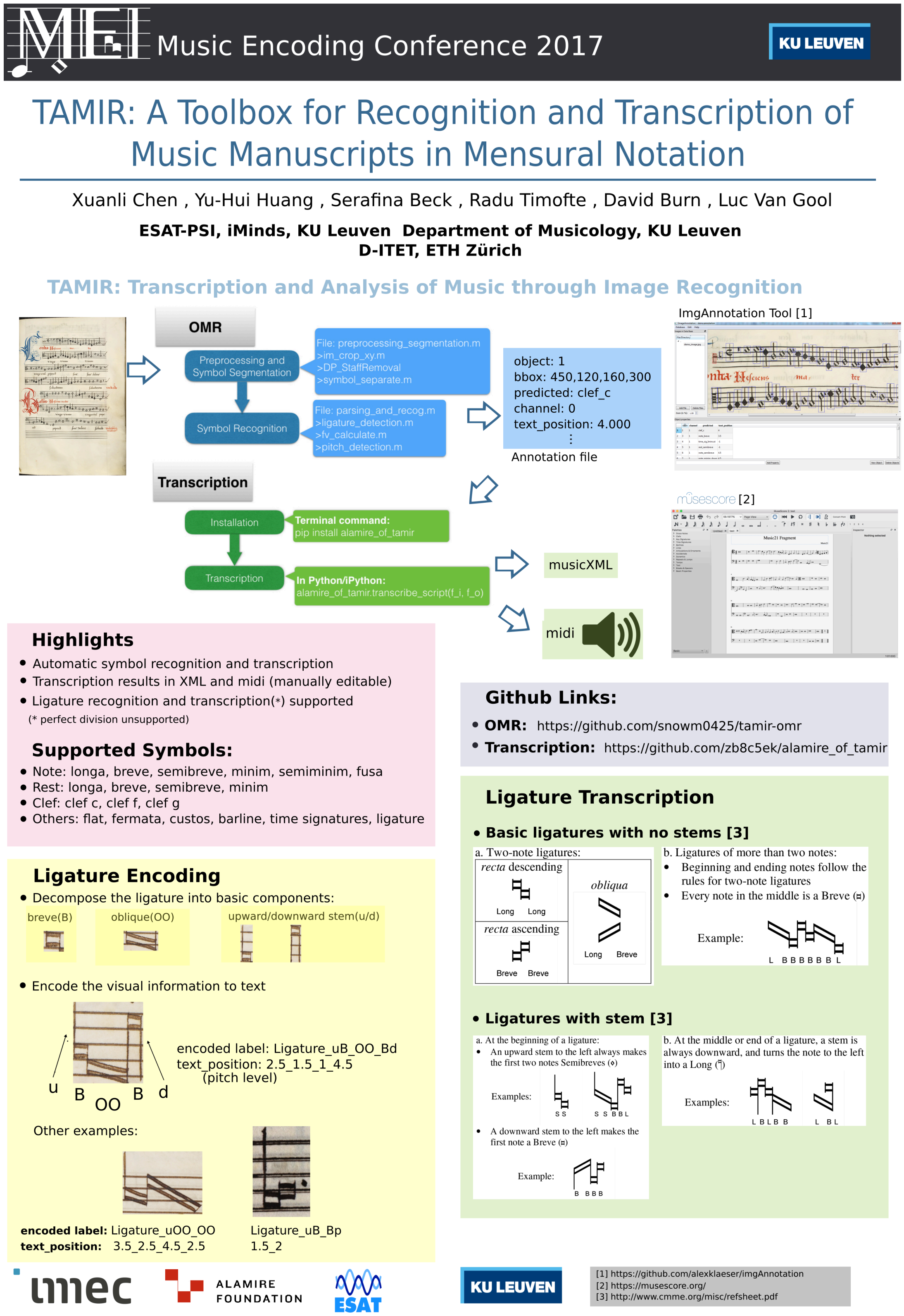


Figure 1: Poster presented at the conference

Author Index