

TROMPA

TROMPA: Towards Richer Online Music Public-domain Archives

Deliverable 5.2 Score Edition Component v2

Grant Agreement nr	770376
Project runtime	May 2018 - April 2021
Document Reference	TR-D5.2-Score Edition Component v2
Work Package	WP5 - TROMPA Contributor Environment
Deliverable Type	Report
Dissemination Level	PU
Document due date	28 February 2021
Date of submission	28 February 2021
Leader	GOLD
Contact Person	Tim Crawford (GOLD)
Authors	Tim Crawford (GOLD), David Weigl, Werner Goebel (MDW)
Reviewers	David Linssen (VD), Emilia Gomez, Aggelos Gkiokas (UPF)

Executive Summary

This document describes tooling developed as part of the TROMPA Digital Score Edition deliverable, including an extension for the Atom text editor desktop application to aid in the finalisation of digital score encodings, a Javascript module supporting selection of elements upon a rendered digital score, and the Music Scholars Annotation Tool, a web app integrating this component, in which users can select, view and interact with musical scores which have been encoded using the MEI (Music Encoding Initiative) XML format. The most important feature of the Annotation Tool is the facility for users to make annotations to the scores in a number of ways (textual comments or cued links to media such as audio or video, which can then be displayed in a dedicated area of the interface, or to external resources such as html pages) which they ‘own’, in the sense that they may be kept private or ‘published’ to other TROMPA users of the same scores. These were the main requirements of Task 5.2 of the TROMPA project.

Task 5.2 focuses on the integration of data and technologies (WP3–WP4) for the collaborative edition of digital scores. In this functionality, users will be able to access digital scores that can be annotated and linked; this further leads to detailed linking and annotation across and between score documents and audio recordings. This alignment is enabled by the adoption within TROMPA of the Music Encoding Initiative (MEI) as the principal storage and exchange medium for musical data, and the MELD (Music Encoding and Linked Data) framework for data management. The corresponding deliverable D5.2 is a Digital Score Edition component for flexible on-screen display, within a standard web browser, of musical scores, online user-interaction (e.g. passage-selection and highlighting for further processing) and annotation (including linkage to other documents, scores or audio/video recordings).

Digital music encodings

The XML music-encoding formats MusicXML and MEI (Music Encoding Initiative) offer comprehensive, although slightly different, capabilities which make a project such as TROMPA possible. MusicXML is supported by a wide range of score-editing software programs, and a large repertory of classical music in the public domain is available in this format. But MEI (into which MusicXML can be converted) was developed with the exacting needs of musicologists in mind, and has been selected for use within TROMPA because certain features make it more flexible and compliant to the complex needs of the project.

The DSE makes extensive use of Verovio, a high-quality score-rendering open-source library developed by Laurent Pugin at RISM Digital Center (formerly RISM Switzerland). This uses MEI as its ‘native’ score-encoding format. The graphical display of scores is rendered as SVG (Scalable Vector Graphics), wherein displayed elements can be manipulated separately (e.g. for hiding/showing, highlighting/colouring, editing, etc.) while retaining the same xml:id as in the originating MEI encodings. In this way, they make low-level user-interaction with a musical score possible in a way not hitherto available, in a cross-platform environment independent of the user’s operating system, and it does not require software installation on their own machine. In particular, we would wish to draw attention to the enhancements we have been able to develop for the Verovio score-rendering library (see 3.1.1, below).

Editing MEI

While MEI has seen considerable uptake in the music research community, a comprehensive, widespread repertoire of high-quality, publicly licensed MEI encodings is yet to be established, largely due to the lack of native support for MEI in commercial score typesetting packages such as Sibelius. Consequently, the generation of MEI from other sources, e.g., from encodings in formats such as MusicXML or Kern, or via Optical Music Recognition (OMR) from the printed page, is a common use-case for projects employing MEI-related technologies. Such translations of music information into the MEI formats are rarely error-free. Final editing by hand can be a laborious activity, requiring knowledge of and careful attention to the XML syntax, MEI schema hierarchy, and to the correct placement of and reference to XML identifiers. We have worked to improve automatic conversion from MusicXML to MEI as much as possible (Section 3.1.1), and we have implemented the *MEI-friend* package to enhance manual MEI editing, as discussed in Section 3.1.2.

MEI-Friend package for Atom editor

The motivation for designing and contributing this package was to alleviate final copy editing of MEI code derived either from exports of commercial or free notation editing programs (such as Sibelius, Finale, or MuseScore) or from OMR pipelines as described above.

Currently, MEI is usually edited using XML-supporting editing software, such as Oxygen or Visual Studio Code (in combination with common XML extensions, such as “XML Tools” available through the extension market place), with a combination of hand-crafted scripts and XSLT transforms. Atom is an open-source text editor that is extensible through user-contributed packages available through the Atom package manager. Atom is based on web technologies (the browser Chromium and Node.js) and the packages can be provided in Javascript, so any such package can be easily reusable as a web-based tool.

The MEI-Friend package features as-you-type rendering of MEI encoding into music notation, using Verovio provided by the npm package manager operating inside the Atom package ecosystem. The editing capabilities currently comprise inserting control elements on selected notes (slur, tie, hairpin, dynamics, directive, tempo indication, fermata, trill, mordent, turn, arpeggio, glissando, pedal) and to manipulate some of their properties (curve direction, placement, and stem direction) by specialized keyboard shortcuts. As Atom supports the github versioning system, the published MEI encodings can easily be loaded, inspected, corrected, and pushed back to the repository.

The MEI-Friend package has been published to the official Atom package manager with proper documentation and is currently in version 0.2.15 with 92 downloads (25 Feb 2021), even before being promoted to the MEI community.

Selectable Score Module

In order to flexibly accommodate the requirements of various use-cases for working with digital scores in the TROMPA project, we implemented a Javascript module that wraps more complex functionalities of displaying, selecting, and annotating elements of music notation in a simple, easily implemented API. The selectable-score module is implemented using MELD (Music Encoding and Linked Data), a semantic framework and set of client rendering and interaction libraries [1] designed to simplify the creation of web applications providing multimodal access and interaction with music score encodings. MELD handles the integration of rendered TROMPA score data (as in the case of MEI-encoded scores which are displayed within the browser using the Verovio Javascript library), and external files which may be displayed (as in the case of scanned score images), or played back (as in the case of audio or video files). Passages within the scores can be selected for annotation, but also to act as queries to a search-system or for other external processing. MELD is thus enabling

technology for the Digital Score Edition providing the data infrastructure on which the score-display interface was developed in Javascript.

The semantic framework combines and builds on W3C web standards including the RDF model for data interchange, the Web Annotation Data Model, the Linked Data Platform (LDP), and the Media Fragments URI specification, established Semantic Web ontologies including the Timeline and Segment Ontologies, and the Music Encoding Initiative's comprehensive XML schema (MEI) which expresses the classes, attributes, and data types required to encode a broad range of musical documents and structures. It provides for the abstract structural segmentation of musical works, and for the linking of structural segments to concrete multimedia representations, e.g., score fragments as collections of identified MEI elements; specified regions of audiovisual recordings; image regions; identified regions of marked-up text in HTML or TEI.

Through its basis in RDF Linked Data, each entity within this framework is identified by its own URI identifier. This supports the creation of Web Annotations targeting any combination of entities, including (fragments of) the abstract and concrete representations discussed above, as well as entities described within the TROMPA Contributor Environment (via its REST API), and any other URI-identified resources within the wider Semantic Web.

Music Scholars Annotation Tool

The MELD framework does not provide a user-facing interface for directly generating or editing annotations. In order to answer our need to digitally annotate musical scores, we have developed an interface that allows users to simultaneously view and share annotations of musical scores afforded by the MELD framework. The score annotation tool is based on previous work such as the open-source Audio Annotator interface -- a javascript tool developed to allow users to attach annotations to audio -- and in the same way we needed to be able to allow users to make both text and graphical annotations. While sharing some goals, our annotation software differs from tools such as the Audio Annotator or Sonic Visualizer, in that, as opposed to making annotations along a continuous axis of chronological time on the single dimension of an audio waveform, the Music Scholars Annotation Tool needed to be able to make finer grained annotations on discrete, ordered, and hierarchical events that unfold over musical time, and are represented in a musical score. Our goal was to adopt a similar strategy following the ability to make annotations with shaded, coloured boxes, but we also needed to allow users to have more control over what elements they were and were not annotating. This ability to annotate discrete and hierarchical events is afforded by using MEI. Using the Music Scholars Annotation Tool, users are able to:

- ❖ View rendered digital scores selected from the TROMPA Contributor Environment, or loaded from any web-accessible MEI encoding by entry of the corresponding URL
- ❖ Select score elements according to combinations of different types (notes, measures, dynamics, directives)
- ❖ Create annotations of their score selections of the following types:
 - Describing - add a textual description
 - Linking - link to a Web resource; supports browsing to the linked resource when the annotation is triggered
 - Cue media - load an audiovisual recording by selecting a node in the CE or indicating an external URI, and select regions along a visual representation of the audio waveform, using tooling described in D5.5 (Annotation Tools); supports playback within the UI from the appropriate position when the annotation is triggered.

- Image - supply the URI of an image to associate with the annotation; supports displaying the image when the annotation is triggered
- Reply - add a textual response to a previously created annotation; supports threaded display of replies at different hierarchical levels.
- ❖ Store these annotations within private, user-controlled space (Solid Pods) - see D5.1 (TROMPA Data Infrastructure).
- ❖ Control access rights on specified annotations, to share them with other users, make them publicly accessible, or publish them under open license with the TROMPA Contributor Environment (Section 5).

Relation to the TROMPA Data Infrastructure

The Digital Score Edition component interacts with TROMPA's Contributor Environment for the selection of scores for display, for saving annotations and for managing data-transmission for further processing (e.g. initiating searches external to the currently-displayed score).

Annotations generated by users of applications integrating TROMPA's *selectable-score* module (Section 3), including the Music Scholars Annotation Tool presented in Section 4, and of the working prototype for instrumental players (**D6.5-Working Prototype for Instrument Players**), are handled by default as private data. As such, these contributions are not automatically ingested into TROMPA's Contributor Environment, but rather reside within the contributing user's Solid Pod (see **D5.1-Data Infrastructure v2**). The user may choose to selectively share individual annotations through interactions with these interfaces, making them accessible to specified other users based on their (Solid) WebIDs; or, to make select annotations within their Solid Pod publicly readable, including by non-authenticated users.

Users may also choose to publish select annotations with TROMPA under an open Creative Commons (CC-BY 4.0) license. When this action is requested by the user, the selected annotation is first made publicly readable in the user's Pod. A copy of the annotation is then ingested into the TROMPA Contributor Environment using technologies and APIs provided by **D5.5 - Annotation Tools**. Particularly, these convert the Web Annotation structure from its Solid-native JSON-LD format into a neo4j property graph representation; generate corresponding nodes as members of ItemList objects associated with the score being annotated; set the `rights` property of the cloned Annotation object to point to the appropriate license URI; and reference the source annotation within the user's Pod by its URI using the `canonical` property as an explicit provenance trace and disambiguation cue. Thus published, the user's annotation becomes discoverable to other users of TROMPA's Contributor Environment.

Music repertoires

For the typical repertoires which need to be displayed frequently in the Digital Score Edition component, we give two examples from different periods which are contrasted in terms of their complexity and their notation: Early music: 16th-century vocal music; Beethoven piano scores (for details, see section 4).

Future work and evaluation

The interface and associated technologies described in this deliverable will be tested and their useability assessed during the final evaluation phase of TROMPA (months 35-36). The principal aim is to ensure that the interfaces are intuitively useable by a range of non-technical users (that is, without advanced computer skills) for a task they wish to carry out, in particular, the annotation of a score and the addition of comments to annotations previously done by others. Further to this, the evaluation exercise will offer users the possibility (for 16c choral music) of instigating searches of an

external musical database for a query generated from a score, or a selected region therefrom, in the TROMPA interface. The evaluation will be aimed at a broad range of users, both expert and non-expert (including those who are not able to read musical notation fluently) and uses examples from two highly contrasting musical repertoires: Gustav Mahler's symphonies; and Renaissance choral music.

Version Log		
#	Date	Description
v0.1	22 February 2021	Initial version submitted for internal review
v0.2	25 February 2021	Revised version after internal review
v1.0	28 February 2021	Final version submitted to EU

1. Introduction	9
1.1 Granularity of selection/interaction	9
1.2 Annotations and links	10
1.3 Audio linkage	10
1.4 MELD and TROMPA score-interface requirements	10
2. Generating and editing digital music encodings	10
2.1 Digital music encodings	11
2.2 Relation to TROMPA use cases	12
2.3 Semi-automatic workflows for generating MEI from PDFs	12
3 The Digital Score Edition Component	14
3.1 Creating and Editing MEI	14
3.1.1 Converting MusicXML to MEI with Verovio	14
3.1.2 MEI-Friend package for Atom editor	16
3.2 Selectable Score interface	19
3.3 Music Scholars Annotation Tool	21
3.4 Relation to the TROMPA Data Infrastructure	24
4 Music repertories	25
4.1 Early music: 16th-century vocal music	25
4.2 Beethoven piano scores	25
5 Future work and evaluation	25
6 Conclusions	26
9 References	28
List of abbreviations	28

1. Introduction

This document describes tooling developed as part of the the TROMPA Digital Score Edition (DSE) deliverable, including an extension for the Atom text editor desktop application to aid in the finalisation of digital score encodings, a Javascript module supporting selection of elements upon a rendered digital score, and the Music Scholars Annotation Tool, a web app integrating this component,, in which users can select, view and interact with musical scores which have been encoded using the MEI (Music Encoding Initiative) XML format. The most important feature is the facility for users make annotations to the scores in a number of ways (textual comments or cued links to media such as audio or video, which can then be displayed in a dedicated area of the interface, or to external resources such as html pages) which the ‘own’, in the sense that they may be kept private or ‘published’ to other TROMPA users of the same scores. These were the main requirements of Task 5.2 of the TROMPA project.

Task 5.2 focuses on the integration of data and technologies (WP3–WP4) for the collaborative edition of digital scores. In this functionality, users will be able to access digital scores that can be annotated and linked; this further leads to detailed linking and annotation across and between score documents and audio recordings. This alignment is enabled by the adoption within TROMPA of the Music Encoding Initiative (MEI)¹ as the principal storage and exchange medium for musical data, and the MELD (Music Encoding and Linked Data) framework² for data management. The corresponding deliverable D5.2 is a React (Javascript) component³ facilitating selection of note elements upon a rendered digital score (Section 3.2); and, a user interface⁴ (Section 3.3) wrapping this component for flexible on-screen display, within a standard web browser, of musical scores, online user-interaction (e.g. passage-selection and highlighting for further processing) and annotation (including linkage to other documents, scores or audio/video recordings).

For D5.2, described in this document, having chosen a score, the user can:

- I. select one (or maybe more) regions, whose encoded musical contents (or data such as locations of musical objects, depending on task) can be specified as the location for annotation, and/or passed to other TROMPA components via the data infrastructure’s API (see Granularity of selection/interaction, below);
- II. see visual feedback indicating the presence of annotations (which may contain links to other resources within or external to TROMPA) add their own new annotations, or comment on existing ones (i.e. annotate annotations);
- III. follow links (from within the annotations) to other TROMPA musical entities or score-pages or to external resources.

1.1 Granularity of selection/interaction

Users can select various types of element from those displayed in a rendered score: notes, measures, dynamics (‘piano’, ‘ff’, etc.) and other ‘directives’, such as textual expression markings or

¹ <https://music-encoding.org>

² <https://github.com/oerc-music/meld-clients-core>

³ <https://github.com/trompamusic/selectable-score>

⁴ <https://github.com/trompamusic/music-scholars-annotator>

tempo changes. These can be used as ‘anchors’ for annotations or (in the case of notes or measures) be passed to an external API for further processing (such as music database search).

1.2 Annotations and links

The MELD framework permits a fully adequate range of annotation types which have been deployed in earlier projects (see Section 3.2 below). MELD does not provide any authoring environment for user-generated annotations, so we have developed a user-friendly interface to achieve this for music scholars. Associating annotations with credentialed users (so that they maintain control of access, sharing and display of their own annotations) was new ground which required us to undertake significant original research, but, since it further requires significant interactions with the TROMPA Contributor Environment, this work on the Digital Score Edition component has proved a useful test-case for the project as a whole. MELD operates upon Linked Data employing standard web URIs (where linking to web resources; e.g., an MEI file or an audio recording) or media fragment identifiers (where linking to resource fragments; e.g., a particular measure of an MEI file, or a particular time range within an audio recording). This provides for transparent interlinking between resources hosted within the Contributor Environment (via its REST API) and resources hosted anywhere on the web.

1.3 Audio linkage

Synchronised linkage to audio recordings formed part of our original specification for this deliverable. However, due to the rescheduling of the T5.2 deliverables caused by the COVID crisis we have focused rather on the provision of direct links to audio files by means of annotations. In this way we can provide a similar functionality, albeit with the loss of automatic page-turning, which can easily be done manually.

1.4 MELD and TROMPA score-interface requirements

The MELD-clients-core library⁵ provides a starting point for the development of the various interfaces required. Existing MELD components cater for these requirements, but alignments between a score and a media file, such as an audio or video recording of a performance of the score, need to be generated separately. Automated (rather than manual/hard-wired) generation of alignments forms part of Task 3.5, ‘Multimodal music information alignment’, and is implemented according to use-case requirements (see, e.g. **D6.5-Working Prototype for Instrumental Players**⁶).

2. Generating and editing digital music encodings

Although not strictly part of the DSE per se, in this section we discuss some enhancements to the problem of acquiring, converting and editing high-quality score encodings in MEI format suitable for the various uses to which they are put in TROMPA. These largely arise from the experience of working with such resources during the course of the project. In particular, we would wish to draw

⁵ See footnote 2.

⁶ https://trompamusic.eu/deliverables/TR-D6.4-Working_Prototype_for_Instrument_Players_v2.pdf

attention to the enhancements we have been able to develop for the Verovio open-source score-rendering library (see 3.1.1, below).

2.1 Digital music encodings

In order for a score to be electronically preserved and shared between performers, composers, and other musically literate people, the notes on the page need to be digitized in a way that fully captures the essential features notated on the written page. Digitizing musical scores presents special challenges that do not exist for text documents. Music notation, which has evolved over many centuries, is a highly pragmatic solution to the problem of capturing the ‘essence’ of an essentially time-based phenomenon in a static document, and serves several purposes depending on the use to which it is put. For performers, including singers, instrumentalists and orchestral conductors, it serves as a record which preserves in some degree the intentions of the composer, while for the scholar it may be a source of data for analysis or comparison with other instances (including arrangements, revisions and derivative works). Similarly, the intention behind an encoding may be to capture the layout and detailed graphical appearance of a printed or manuscript score, or alternatively to systematically record the underlying structure of the music at multiple hierarchical levels. Further types of encoding (such as MIDI files) preserve the limited subset of this structural information necessary to reconstruct an aural ‘performance’ of the score (using appropriate software) without any consideration of its visual appearance.

The XML music-encoding formats MusicXML⁷ and MEI (Music Encoding Initiative)⁸ offer comprehensive, although slightly different, capabilities which make a project such as TROMPA possible. MusicXML has the advantage that, since it is supported by a wide range of score-editing software programs, a large repertory of classical music in the public domain is available in this format. MEI, on the other hand, was developed with the exacting needs of musicologists in mind, and has been selected for use within TROMPA owing to the fact that certain features make it more flexible and compliant to the complex needs of the project. In particular, the main advantage of MEI over MusicXML for our purposes is the universal use of `xml:ids` for all elements of an MEI encoding, meaning that any musical ‘object’ in a score (a note, barline, expression mark, etc.) can be referenced uniquely and easily addressed even after some transformation (such as transposition, or excerpting of a score-fragment). MusicXML documents can be automatically converted to MEI, although the resulting encodings typically require significant manual validation and fixing to achieve high quality results.

The DSE makes extensive use of Verovio,⁹ a high-quality score-rendering library developed by Laurent Pugin at RISM Digital Center (formerly RISM Switzerland). This uses MEI as its ‘native’ score-encoding format, and it supports many specialist MEI flavours as needed by Music Scholars (e.g. tablature, mensural notation, neumes, etc.). (See [2] for a summary introduction to MEI.) The graphical display of scores is rendered as SVG (Scalable Vector Graphics),¹⁰ wherein displayed elements can be manipulated separately (e.g. for hiding/showing, highlighting/colouring, editing, etc.) while retaining the same `xml:id` as in the originating MEI encodings. In this way, they make low-level user-interaction with a musical score possible in a way not hitherto available, in a cross-platform environment which is independent of the user’s operating system and does not

⁷ <https://www.musicxml.com>

⁸ <https://music-encoding.org>. See also [2].

⁹ <https://www.verovio.org> (accessed 25 Feb 2021)

¹⁰ <https://www.w3.org/Graphics/SVG/> (accessed 25 Feb 2021)

require software installation on their own machine. Similarly, groups of SVG elements (adjacent or non-adjacent) can be addressed simultaneously (e.g. for editing, as ‘anchors’ for annotations, or as selected items for further processing, etc.)

2.2 Relation to TROMPA use cases

As mentioned above (2.1), many public-domain scores in MusicXML format are available for download, the reason for this being the widespread adoption of MusicXML as a file-exchange format by many music-notation programs,¹¹ including the two in most common use by music professionals, Finale¹² and Sibelius.¹³ For this reason it is desirable that reliable translation of MusicXML to MEI can be achieved. The open-source Verovio score-engraving library has this capability, although for our purposes it has been necessary to implement certain improvements (see 3.1.1, below).

An alternative approach, where no score-encoding is available, is to use public-domain PDF files of suitable high graphic quality, subject them to OMR, and encode the recognised score-data as MEI. We carried this out in two case-studies to provide data which we can then use in the working prototype demos for D6.5 and **D6.3-Working Prototype for Scholars**¹⁴. The resulting semi-automatic workflows are described in the next section.

2.3 Semi-automatic workflows for generating MEI from PDFs

Since the procedures followed were slightly different in the two cases, we describe them separately here.

Study 1: Beethoven piano works using commercial OMR

The recent 250th birthday of Ludwig van Beethoven and the available encodings of his 32 sonatas for piano in the Humdrum format¹⁵ were the initial motivation to attempt to encode the remaining piano solo works by Beethoven into MEI format. These pieces comprise several variation cycles (among them the Eroica Variations Op. 35, the Diabelli variations Op. 120, or the c-minor variations WoO. 80), Rondos (Op. 51, Bagatelles (Op. 33, 119, 126), the Polonaise (Op. 89), Sonatinas (WoO Anh. 5), Phantasie (Op. 77), and pieces, such as Rondo a Capriccio Op. 129 and Clavierstück “Für Elise” WoO. 59. As the target edition to encode, we chose Breitkopf and Härtel’s first complete edition from 1862–1868, fully available as scanned images at IMSLP¹⁶. The aim of the encoding was to follow the target edition as closely as possible by the current version 4.0.1 of MEI as engraved by Verovio (currently version 3.2). The corpus encoded now comprises 18 works by Beethoven spanning over 220 pages, all published on GitHub¹⁷.

For repeatability, we sketch the semi-automatic workflow of the OMR process established in this case-study:

¹¹ See <https://www.musicxml.com/software/> (accessed 25 Feb 2021) for a list of software programs supporting MusicXML

¹² <https://www.finalemusic.com/> (accessed 25 Feb 2021)

¹³ <https://www.avid.com/sibelius#> (accessed 25 Feb 2021)

¹⁴ https://trompamusic.eu/deliverables/TR-D6.3-Working_Prototype_for_Scholars_v2.pdf

¹⁵ <https://github.com/craigsapp/beethoven-piano-sonatas> (accessed 10 Feb 2021)

¹⁶ [https://imslp.org/wiki/Ludwig_van_Beethovens_Werke_\(Beethoven,_Ludwig_van\)](https://imslp.org/wiki/Ludwig_van_Beethovens_Werke_(Beethoven,_Ludwig_van)) (acc’d 10 Feb 2021)

¹⁷ <https://github.com/trompamusic-encodings/> (accessed 25 Feb 2021)

- ❖ Scan a commercial edition with a Konica Minolta C308 printer at 300dpi grayscale at A3 landscape with the saturation set 2 steps darker. This step has proven to work considerably more robustly than to take the scanned PDF of the target edition as provided from IMSLP, though such a PDF remains a viable starting point where the use of a printed edition is prohibitive.
- ❖ Adjust the derived PDF with Acrobat Pro (Version 9) through the automatic Optical Character Recognition (OCR) to rotate the pages. If necessary, crop the black margins.
- ❖ With Neuraton PhotoScore (Version 2018 8.8.7): Import with highest resolution and perform OMR; manually fix evident errors (such as key signatures, meter indications, triplets, some obviously incorrectly-OMRed notes) and export as MusicXML (uncompressed).
- ❖ Import into the free Editor MuseScore (Version 3.1.22425) and manually fix OMR errors as much as possible and use target edition for comparison. Export to MusicXml again.
- ❖ Import MusicXML to Verovio and convert it to MEI. During this step, several systematic MusicXML parsing errors have been observed. Most of them have been fixed by editing the C++ code of Verovio and creating numerous pull requests to the main development branch of Verovio. All of them were subsequently merged with the main branch and are now part of the Verovio code base (see section 2.3.1 for a detailed list of contributions to Verovio).
- ❖ The derived MEI are then processed systematically with customized bash scripts (clean_accid.sh to remove double `@accid` and `@accid.ges` attributes, adids.sh to insert missing xml.ids, renumber_measures.sh to renumber the measures from the beginning, taking into account measures that do not conform to the meter signature (@metcon="false") and measures inside multiple endings.
- ❖ Final manual copy editing was performed using the “mei-tools-atom”¹⁸ package for the Atom editor and, starting from Jan 2021, the TROMPA contributed package “mei-friend”¹⁹ with basic editing functionality (see below).

Study 2: Mahler’s 4th symphony using Audiveris

From the earlier PHENICX project,²⁰ we already had access to a complete MusicXML encoding of the full score of his 4th Symphony, which we successfully converted to MEI. However, for the purposes of the Mahler annotation pilot demo (D6.3), we soon realised that we needed a score-encoding which, when rendered, occupied less vertical space than the full score, since passages with many instruments involved are impossible to view on a normal tablet/laptop screen.

To achieve this, we located a public-domain PDF score of the piano-duet arrangement by Josef Wöss (Vienna: Doblinger, 1902) – there exists documentary evidence that Mahler and Wöss played through this arrangement, and that Mahler took care that it conformed to his wishes.

The PDF of the piano-duet version was run through the open-source Audiveris OMR program;²¹ although this does not have a command-line version, it can process the multiple pages of a PDF together. The resulting recognised pages were exported separately as MusicXML files.

A confusing feature of piano duets is that the parts for the two pianists (who typically sit side-by-side on the same stool) are printed on facing pages, not as a conventional score where they would be above one another. For this reason, the recognised MusicXML pages had to be aligned and rendered into a normal score format. While it would in principle be possible to automate this

¹⁸ <https://atom.io/packages/mei-tools-atom> (accessed 11 Feb 2021)

¹⁹ <https://atom.io/packages/mei-friend> (accessed 11 Feb 2021)

²⁰ <http://phenicx.upf.edu> (accessed 8 March 2021)

²¹ <https://github.com/Audiveris/audiveris> (accessed 9 March 2021)

process by processing the MusicXML files, this would have taken a good deal of time for programming and testing; also, for the current purpose (the Mahler showcase) we only needed the first 103 measures of the first movement. The pages were imported separately into the open-source score-editor, MuseScore,²² where we manually copied and pasted them into a new ‘two-piano’ score. Despite some problems due to the relatively few serious OMR errors from Audiveris, this took only just over a day’s work. Final editing was carried out in MuseScore to ensure the new score matched the original PDF as closely as possible.

The resulting ‘two-piano’ score was exported from MuseScore as MusicXML, and subsequently converted without difficulty into MEI.

3 The Digital Score Edition Component

3.1 Creating and Editing MEI

While MEI has seen considerable uptake in the music research community, a comprehensive, widespread repertoire of high-quality, publicly licensed MEI encodings is yet to be established, largely due to the lack of native support for MEI in commercial score typesetting packages such as Sibelius. Consequently, the generation of MEI from other sources, e.g., from encodings in formats such as MusicXML or Kern,²³ or via OMR from the printed page, is a common use-case for projects employing MEI-related technologies. While such translations of music information into the MEI formats are achievable (see Section 3.1.1 below), they rarely operate without error. Arriving at comprehensive-but-imperfect MEI files that require final “polishing” to achieve an optimal state of encoding is thus not unusual.

Doing such finalisation by hand can be a laborious activity, requiring knowledge of and careful attention to the XML syntax, MEI schema hierarchy, and to the correct placement of and reference to XML identifiers. One option to address the manual labour inherent in this task is to split it up among a larger group, as described in TROMPA’s **D6.4-Working Prototype for Orchestras**²⁴ (D6.4). While useful for generating larger quantities of MEI quickly, an alternative workflow suitable for scholarly purposes will put the task under centralised expert purview. In order to assist individual editors and reduce the load of this laborious task, we have worked to improve automatic conversion from MusicXML to MEI as much as possible (Section 3.1.1), and we have implemented the *MEI-friend* package to facilitate manual MEI editing (discussed in Section 3.1.2).

3.1.1 Converting MusicXML to MEI with Verovio

As mentioned above, converting a digital encoding from one format to another will never operate without error, on the one hand because there is not a one-to-one mapping from one encoding format to the other, but also – on the other hand – because the conversion process has some problems and requires improvement. When importing MusicXML into MEI format with Verovio, several systematic MusicXML parsing errors or un-implemented features have been observed (Section 3.1.2). Rather than cleaning up these errors manually at each occurrence, we have made

²² <https://musescore.org/en> (accessed 9 March 2021)

²³ <https://www.humdrum.org> (accessed 8 March 2021)

²⁴ https://trompamusic.eu/deliverables/TR-D6.4-Working_Prototype_for_Orchestras_v2.pdf

contributions to the C++ repository of Verovio²⁵ by creating several pull requests to the development branch of Verovio. After lively discussions with the MEI community and the core developers of Verovio (above all Laurent Pugin who deserves highest appreciation and gratitude for his astute and tireless commitment to this professional open-source project), all of them were subsequently merged with the main branch and are now part of the Verovio code base.

The contribution milestones as listed in the release history²⁶ were:

- ❖ Method Toolkit::GetTimesForElement for accessing score times
- ❖ Improved backup handling in MusicXML import
- ❖ Support for expansion implementation with --expand option
- ❖ Support for <turn> in MusicXML import
- ❖ Support for <arpeg> in MusicXML import
- ❖ Improved support for pedal marks in MusicXML import
- ❖ Improved cross-staff MusicXML import
- ❖ Improved clef changes in MusicXML import
- ❖ Improved hairpin MusicXML import
- ❖ Improved slur MusicXML import
- ❖ Support for <ending> in MusicXML import

Between April 2019 and February 2021 a large number of Commits²⁷ (smallest unit of code contribution) and a total of 28 merged Pull Requests²⁸ are documented on Github. See Figure 2.1. for some illustrating examples.



Figure 2.1 Examples of merged pull requests improving MusicXML import and adding functionality to Verovio.

²⁵ <https://github.com/rism-ch/verovio/>

²⁶ <https://github.com/rism-ch/verovio/releases?after=version-2.2.1>

²⁷ <https://github.com/rism-ch/verovio/commits?author=wergo>

²⁸ <https://github.com/rism-ch/verovio/issues?q=is%3Amerged+author%3Awergo>

3.1.2 MEI-Friend package for Atom editor

The motivation for designing and contributing this package was to alleviate final copy editing of MEI code derived either from exports of commercial or free notation editing programs (such as Sibelius, Finale, or MuseScore) or from OMR pipelines as described above.

Currently, MEI is usually edited by a XML-supporting editing software, such as Oxygen or Visual Studio Code²⁹ (in combination with common XML extensions, such as “XML Tools” available through the extension market place), using a combination of hand-crafted scripts and XSLT³⁰ operations. The edited MEI encoding then is pasted to the online Verovio tools on verovio.org or compiled locally to SVG files.

Atom³¹ is an open-source text editor that is extendable through user-contributed packages available through the Atom package manager. Atom is based on web technologies (the browser Chromium and Node.js) and the packages can be provided in Javascript. This makes code written for any such package easily reusable as a web-based tool in later deployment stages. An existing Atom package (“mei-tools-atom” by nCoda³²) provides displaying and page turning capabilities of MEI encodings, but lacks editing functionality, operates on an outdated Verovio version (2.0.0.4-dev) and is not supported anymore (last minor update in April 2019).

²⁹ <https://code.visualstudio.com/>

³⁰ <https://developer.mozilla.org/en-US/docs/Web/XSLT>

³¹ <https://atom.io/>

³² <https://atom.io/packages/mei-tools-atom> (accessed 11 Feb 2021)



Figure 2.2 Annotated screenshot of the MEI-Friend package in the Atom editor showing an excerpt of the encoding of Beethoven’s Diabelli variations (top), its automatic rendering with Verovio (middle) and the same system in the target edition by Breitkopf & Härtel (bottom, other window outside the Atom editor).

The MEI-Friend package now automatically loads the latest stable release version of Verovio (version 3.2 as of 10 Feb 2021) and features alongside flexible notation rendering several basic editing operations. It features as-you-type rendering of MEI encoding into music notation, using Verovio provided by the npm package manager operating inside the Atom package ecosystem.

A screenshot of the user interface is shown in Figure 2.2 above. The menu bar contains (from left to right):

- ❖ Viewing controls to scale the notation in the notation panel and to invert the notation colors to white and black
- ❖ Controlling page turning (jumping first, previous, next and last page) and setting notation to current encoding cursor position

- ❖ Setting page and system break behavior (corresponding to the `--breaks` option of Verovio)
- ❖ Setting the updating behavior (automatic, so that after each edit in the MEI encoding the notation gets re-rendered versus manually upon button click)
- ❖ Navigating the selected note or rest inside the notation through buttons or key-bindings (note-wise back-/forwards, measure-wise back-/forwards, page-turning)
- ❖ Re-rendering encoding through Verovio (to standardize encoding and insert missing xml:ids)
- ❖ Displaying the Verovio version loaded and the key bindings help page

The editing capabilities currently comprise inserting control elements on selected notes (slur, tie, hairpin, dynamics, directive, tempo indication, fermata, trill, mordent, turn, arpeggio, glissando, pedal) and to manipulate some of its properties (curve direction, placement, and stem direction) by specialized keyboard shortcuts as listed in the key bindings help page (Figure 2.3). The underlying logic is to select one or two notes and use a keyboard shortcut to insert the control element to the MEI code. For example, pressing “S” on two selected notes would insert a slur at the end of the measure of the first selected note, spanning from the first to the second selected note. Subsequent key-presses of “X” would change the placement of the slur from automatic to above or below. Combining the keyboard shortcut with “CTRL” always inserts the control element with the placement attribute set to “below”. There are specialized key-bindings for crescendo versus diminuendo hairpins or starting versus ending pedal markings. The complete list of key bindings is shown in Figure 2.3.

As Atom is tightly supporting the versioning system github, the published MEI encodings on github/trompamusic-encodings can easily be loaded, inspected, corrected, and pushed back to the repository.

The MEI-Friend package has been published to the official Atom package manager³³ with proper documentation and is currently in version 0.2.15 with 92 downloads (25 Feb 2021), even before being promoted to the MEI community.

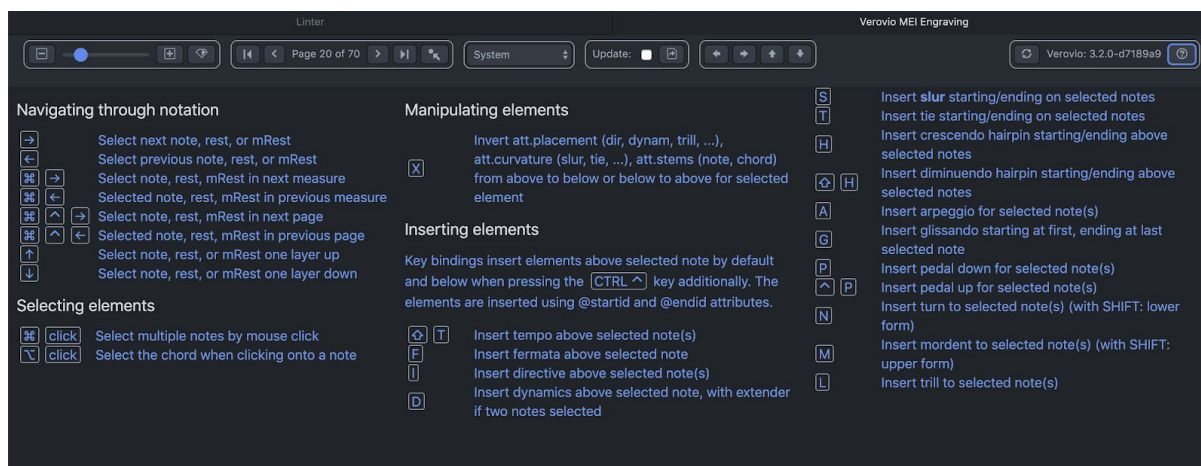


Figure 2.3 Screenshot of the help page of MEI-Friend listing the implemented key bindings for navigating through the notation, selecting elements, and performing several basic insert operations (control elements, such as slurs, hairpins and ornaments) and manipulation (inverting the placement, curve direction, or stem direction of multiple selected elements).

³³ <https://atom.io/packages/mei-friend> (accessed 25 Feb 2021)

3.2 Selectable Score interface

In order to flexibly accommodate the requirements of various use-cases for working with digital scores in the TROMPA project, we have implemented a Javascript module that wraps more complex functionalities of displaying, selecting, and annotating elements of music notation in a simple, easily implemented API. The TROMPA `selectable-score` module³⁴ exposes a JSX / React component that takes the following properties as attributes:

- ❖ `uri`: The URI of the MEI file to be rendered (optionally retrieved from a corresponding MediaObject in the TROMPA CE; otherwise, any MEI file addressable on the Web can be specified)
- ❖ `vrvoOptions`: a Verovio layout options JSON object to customise the engraving performed by the underlying Verovio MEI rendering tool
- ❖ `selectorString`: A CSS selector string that specifies the score elements to be subject to selection; for example, `“.note”` to select individual notes
- ❖ `onSelectionChange`: A callback function that is called whenever the user makes a new selection; it receives a list of selected DOM Node objects as input.
- ❖ `onScoreUpdate`: A callback function that is called whenever the rendered score is updated, e.g. when a page is turned.
- ❖ `annotationContainerUri`: The URI of an LDP container (as provided by Solid Pods, see D5.1 - Data infrastructure) into which annotations of selected elements should be posted.
- ❖ `toggleAnnotationRetrieval`: A Boolean flag that triggers retrieval of annotations from this container, e.g., for rendering or update.
- ❖ `onReceiveAnotationContainerContent`: A callback function that is called when annotations are retrieved.

The associated styleable `PrevPageButton` and `NextPageButton` components exposed by the module simplify performing these interactions without requiring access to the underlying Verovio renderer.

Finally, the associated styleable `SubmitButton` component simplifies the authenticated posting of annotations into LDP (Solid) containers. This component takes the following properties as attributes:

- ❖ `buttonContent`: Custom JSX button content if the default is to be overwritten.
- ❖ `submitUri`: The URI to POST to. If used in an authenticated Solid context, this could be a location in the user's Pod
- ❖ `submitHandler`: Callback function to call when the button is clicked. This is expected to return a JSON object, the contents of which forms the HTTP POST request body.
- ❖ `submitHandlerArgs`: An optional JSON object handed to the `submitHandler` callback when triggered. If not supplied, an empty object is used instead.
- ❖ `onResponse`: Callback function to call when the POST is completed. The application may then set the `SelectableScore` component `toggleAnnotationRetrieval` property to `true` when this happens, so that newly POSTed annotations can be retrieved and rendered.

A minimal example implementation of an application using the `selectable-score` module is provided³⁵.

³⁴ <https://github.com/trompamusic/selectable-score/releases/tag/v1.0>

³⁵ <https://github.com/trompamusic/selectable-score-demo>

The `selectable-score` module is implemented using MELD³⁶ (Music Encoding and Linked Data), a semantic framework and set of client rendering and interaction libraries [1] designed to simplify the creation of web applications providing multimodal access and interaction with music score encodings. MELD handles the integration of rendered TROMPA score data (as in the case of MEI-encoded scores which are displayed within the browser using the Verovio Javascript library), and external files which may be displayed (as in the case of scanned score images), or played back (as in the case of audio or video files). Passages within the scores can be selected for annotation, but also to act as queries to a search-system or for other external processing. MELD is thus enabling technology for the Digital Score Edition providing the data infrastructure on which the score-display interface was developed in Javascript.

The semantic framework combines and builds on W3C web standards including the RDF³⁷ model for data interchange, the Web Annotation Data Model³⁸, the Linked Data Platform (LDP)³⁹, and the Media Fragments URI specification⁴⁰, established Semantic Web ontologies including the Timeline and Segment Ontologies [5-6], and the Music Encoding Initiative's⁴¹ comprehensive XML schema (MEI) which expresses the classes, attributes, and data types required to encode a broad range of musical documents and structures [7]. It provides for the abstract structural segmentation of musical works, and for the linking of structural segments to concrete multimedia representations, e.g., score fragments as collections of identified MEI elements; specified regions of audiovisual recordings; image regions; identified regions of marked-up text in HTML or TEI.

Through its basis in RDF Linked Data, each entity within this framework is identified by its own URI identifier. This supports the creation of Web Annotations targeting any combination of entities, including (fragments of) the abstract and concrete representations discussed above, as well as entities described within the TROMPA Contributor Environment (via its REST API), and any other URI-identified resources within the wider Semantic Web.

The TROMPA alignment data model, which instantiates the MELD semantic framework, further enables the inclusion of feature data derived from musical representations by quantifying specific aspects of e.g. an audio signal, or of a symbolic score. This data model is discussed in **D3.5-Multimodal music information alignment**⁴², and illustrated in Figure 3.1 below, and drives performance assessment and user interactions in TROMPA's instrumental players use-case (see **D5.4-Music Performance Assessment**⁴³; **D6.5-Working Prototype for Instrumental Players**).

The MELD client libraries⁴⁴ facilitate the creation of web interfaces that render and facilitate interactions between instances of the different musical representations interlinked within the MELD semantic framework. They are implemented in a set of Javascript (React) components, actions, and reducers, which can be configured and flexibly extended to build applications for specific use contexts. The `selectable-score` component has been developed with a view to simplifying developer access to this tooling in order to facilitate integration across TROMPA project use-cases.

³⁶ <https://github.com/oerc-music/meld>

³⁷ <https://www.w3.org/RDF/>

³⁸ <https://www.w3.org/TR/annotation-model/>

³⁹ <https://www.w3.org/TR/ldp/>

⁴⁰ <https://www.w3.org/TR/media-frags/>

⁴¹ <https://music-encoding.org>

⁴² https://trompamusic.eu/deliverables/TR-D3.5-Multimodal_Music_Information_Alignment_v2.pdf

⁴³ https://trompamusic.eu/deliverables/TR-D6.4-Music_Performance_Assessment_v2.pdf

⁴⁴ <http://github.com/oerc-music/meld-clients-core>

To date, MELD applications have been created around uses in scholarly publication [7], music performance [8], and composition [9], demonstrating the applicability of this technology to TROMPA's varied use cases. However, MELD has thus far lacked an authoring environment for the definition of user-defined structural segmentations (beyond automatic note-, measure-, and section-level structurings retrievable from MEI encodings) and corresponding annotations, relying instead on a priori specification by the application developer through manual creation of RDF triples. This limitation is addressed by the new technologies developed in the TROMPA project, which explicitly support the creation of annotations within a user-friendly authoring environment (Section 3.3; see also [D6.5](#)).

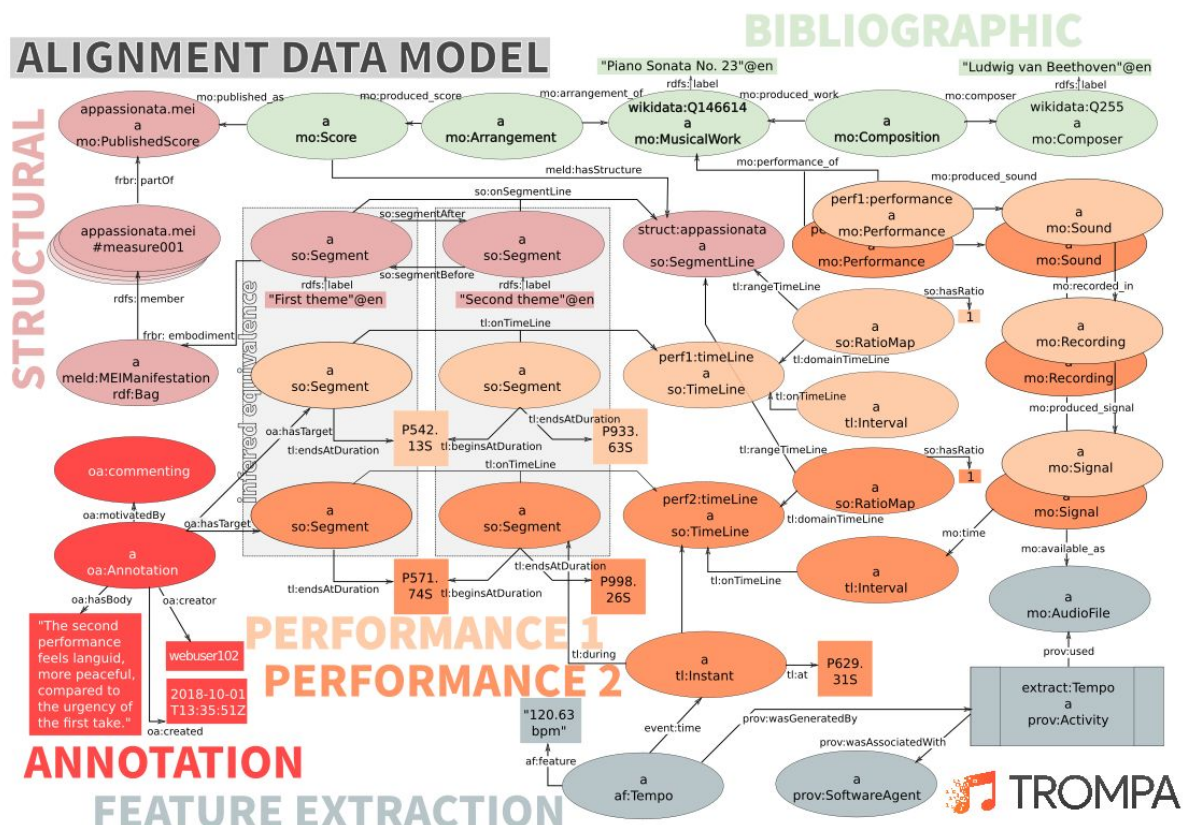


Figure 3.1 - TROMPA alignment data model based on the MELD semantic framework.

3.3 Music Scholars Annotation Tool

As mentioned above, the MELD framework does not provide a user-facing interface for directly generating or editing annotations. In order to integrate the MELD framework with our need to digitally annotate musical scores, we have developed an interface that allows users to simultaneously view and share annotations of musical scores afforded by the MELD framework. We detail the development of our Music Scholars Annotation Tool below.

Following the 1st version of the TROMPA report, **D5.5-Annotation Tools**⁴⁵, we aimed to base our score annotation tool on previous work such as the open-source Audio Annotator interface⁴⁶-- a javascript tool developed to allow users to attach annotations to audio. Similar to the Audio

⁴⁵ https://trompamusic.eu/deliverables/TR-D5.5-Annotation_Tools_v1.pdf

⁴⁶ <https://github.com/CrowdCurio/audio-annotator>

Annotator interface, we needed to be able to allow users to make both text and graphical notations of musicological importance. This software would not stand alone, but also be used across the Performers, Choir Singers, and Music Enthusiasts work packages. Further, integration between all use cases would be done via TROMPA's Contributor Environment.

While similar in goals to a tool such as the Audio Annotator or Sonic Visualizer, our annotation software differs from those previously mentioned in that, as opposed to making annotations along a continuous axis of chronological time on the single dimension of an audio waveform, the Music Scholars Annotation Tool needed to be able to make finer grained annotations on discrete, ordered, and hierarchical events that unfold over musical time. Our goal was to adopt a similar strategy following the ability to make annotations with shaded, coloured boxes, but we also needed to allow users to have more control over what elements they were and were not annotating. This ability to annotate discrete and hierarchical events is afforded by using MEI.

Initial development of the Music Scholars Annotation Tool began in Spring of 2020 with an initial prototype developed using the React web framework. Early versions of the prototype integrated the software developed by other packages of the TROMPA project, namely the selectable score component from the Instrumental Players use case. After initial prototyping and experimenting with offering different selection options to the users as well as integration of Solid Pod logic, minimum viable product (MVP) was reached in Summer of 2020.

At MVP, users of the digital score annotator were able to enter, save, and share the data they created on their local session. Following MVP, the development team spent the Autumn and Winter of 2020 building up the functionalities of the Music Scholars Annotation Tool with features such as allowing users to select different MEI files from a larger repository (eventually CE), integration with refactored Solid Pod logic, and the ability to select additional MEI objects such as dynamics, directives, and (cue media). Finally, the early months of 2021 have focused on developing a more user friendly, accessible user interface, refactoring of the existing code, and functionality to allow users to make annotations on existing annotations, thus allowing for users to interact with one another using a single score as a general reference.

Using the Music Scholars Annotation Tool, users are able to:

- ❖ View rendered digital scores selected from the TROMPA Contributor Environment using the multimodal component (**D5.3-TROMPA Processing Library**⁴⁷), or loaded from any web-accessible MEI encoding by entry of the corresponding URL
- ❖ Select score elements according to combinations of different types (notes, measures, dynamics, directives)
- ❖ Create annotations of their score selections of the following types:
 - Describing - add a textual description
 - Linking - link to a Web resource; supports browsing to the linked resource when the annotation is triggered
 - Cue media - load an audiovisual recording by selecting a node in the CE or indicating an external URI, and select regions along a visual representation of the audio waveform, using tooling described in **D5.5-Annotation Tools**; supports playback within the UI from the appropriate position when the annotation is triggered.
 - Image - supply the URI of an image to associate with the annotation; supports displaying the image when the annotation is triggered

⁴⁷ https://trompamusic.eu/deliverables/TR-D5.3-TROMPA_Processing_Library_v2.pdf

- Reply - add a textual response to a previously created annotation; supports threaded display of replies at different hierarchical levels.
- ❖ Store these annotations within private, user-controlled space (Solid Pods) (see **D5.1-TROMPA Data Infrastructure**⁴⁸).
- ❖ Control access rights on specified annotations, to share them with other users, make them publicly accessible, or publish them under open license with the TROMPA Contributor Environment (Section 5).

Figure 4.1. All elements within two complete measures (highlighted in blue) have been selected by the user dragging across them with ‘Measure’ chosen as Selection Type

⁴⁸ https://trompamusic.eu/deliverables/TR-D5.1-Data_Infrastructure_v2.pdf

Specify the annotation container path inside your Pod:

Selection Type

Note Measure Dynamics Directives

Annotation type

Describing Linking Cue Media Reply

The textual content of this annotation is new different anno

Created on: 2021-02-10T19:27:38.626Z by <https://federicozubani.trompa-solid.upf.edu/profile/card#me> with describing motivation

Show replies Reply

This reply contains: reply to new different anno

Created on: 2021-02-10T19:27:55.650Z by <https://federicozubani.trompa-solid.upf.edu/profile/card#me> with replying motivation

The textual content of this annotation is test

Created on: 2021-02-10T19:26:24.624Z by <https://federicozubani.trompa-solid.upf.edu/profile/card#me> with describing motivation

Show replies Reply

This reply contains: second reply to test

Created on: 2021-02-10T19:30:08.307Z by <https://federicozubani.trompa-solid.upf.edu/profile/card#me> with replying motivation

This reply contains: reply to test

Created on: 2021-02-10T19:27:02.876Z by <https://federicozubani.trompa-solid.upf.edu/profile/card#me> with replying motivation

help

Figure 4.2. After the user has pressed the ‘Fetch Annotations’ button, the annotations are listed and their location indicated by highlight in the score; the user can then add their own annotations.

3.4 Relation to the TROMPA Data Infrastructure

The Digital Score Edition component interacts with TROMPA’s Contributor Environment for the selection of scores for display, for saving annotations and for managing data-transmission for further processing (e.g. initiating searches external to the currently-displayed score). The Contributor Environment’s API is described in technical detail in the CE API manual.⁴⁹ Specific CE components for multimodal integration of music data⁵⁰ have also been provided.

Annotations generated by users of applications integrating TROMPA’s *selectable-score* module (Section 3), including the Music Scholars Annotation Tool presented in Section 4, and of the working prototype for instrumental players (D6.5), are handled by default as private data. As such, these contributions are not automatically ingested into TROMPA’s Contributor Environment, but rather reside within the contributing user’s Solid Pod (see D5.1). The user may choose to selectively share individual annotations through interactions with these interfaces, making them accessible to specified other users based on their (Solid) WebIDs; or, to make select annotations within their Solid Pod publicly readable, including by non-authenticated users.

Users may also choose to publish select annotations with TROMPA under an open Creative Commons (CC-BY 4.0) license. When this action is requested by the user, the selected annotation is first made publicly readable in the user’s Pod. A copy of the annotation is then ingested into the TROMPA Contributor Environment using technologies and APIs provided by D5.5 (Annotation Tools). Particularly, these convert the Web Annotation structure from its Solid-native JSON-LD format into a neo4j property graph representation; generate corresponding nodes as members of ItemList objects associated with the score being annotated; set the rights property of the cloned Annotation object

⁴⁹ <https://github.com/trompamusic/ce-api>

⁵⁰ <https://github.com/trompamusic/ce-multimodal-component>

to point to the appropriate license URI⁵¹ and, reference the source annotation within the user's Pod by its URI using the canonical property as an explicit provenance trace and disambiguation cue. Thus published, the user's annotation becomes discoverable to other users of TROMPA's Contributor Environment.

4 Music repertoires

Through its integration as a component in several of TROMPA's use cases (e.g., D6.3 - Music scholars, D6.5 - Instrumental players), Digital Score Edition component technologies are applied to score encodings representing significantly different repertoires. To illustrate, we give two examples from different periods which are contrasted in terms of their complexity and their notation.

4.1 Early music: 16th-century vocal music

This repertory has the advantage of relative simplicity in terms of score-display requirements. Also, GOLD has long-term experience in this field and continues to engage in active research. The Early Music Online (EMO) resource of ~300 early printed music books has been used extensively for this work, and is freely available, not only in high-quality digital scans from archival microfilms (c32,000 page images), but also in MEI encodings from the output of the OMR program Aruspix, and indexes derived therefrom. Many of the works in EMO are freely available online in score editions as PDFs, sometimes with digital encodings and even with individual voice parts in MIDI format.⁵²

Recent work on full-text musical-content searching of EMO allows fast retrieval of music to match a given example (currently, either as query code or as a page-image). Such searches will be activated by user-selection of a note-sequence within a rendered score to find similar music in the collection: the query is generated by extracting the sequence of MEI pitches from the selected notes which is then appropriately encoded and passed as query by URL to the EMO server's API.

4.2 Beethoven piano scores

Later music than that described in 4.1, above, is much more complex in texture and in notation. An example which is central to the TROMPA Music Performance Assessment Component (D5.4) is 19c piano music; the associated collection of digitised (MEI) Beethoven piano scores was generated as part of the TROMPA project⁵³. These can be associated with simple structural analyses described by visual annotations (using colour-shading and text-labelling), to inform performers about salient segments of their own performance and those of others. This is an example of a non-verbal (i.e. non-textual) form of annotation, made possible through adoption of MEI and the data-sharing capacities of the TROMPA environment enabled by MELD.

5 Future work and evaluation

The interface and associated technologies described in this deliverable will be tested and their useability assessed during the final evaluation phase of TROMPA (months 35-6). This is related to the use-cases outlined in the Music Scholars deliverable (D6.3). The principal aim is to ensure that the interfaces are intuitively useable by a range of non-technical users (that is, without advanced

⁵¹ <https://creativecommons.org/licenses/by/4.0/>

⁵² Over 1,000 such works are available in PDF at <https://www.uma.es/victoria/> together with digital encodings.

⁵³ <https://github.com/trompamusic-encodings>

computer skills) for a task they wish to carry out, in particular, the annotation of a score and the addition of comments to annotations previously done by others. Further to this, the evaluation exercise will offer users the possibility (for 16c choral music) of instigating searches of an external musical database for a query generated from a score, or a selected region therefrom, in the TROMPA interface.

The evaluation will be aimed at a broad range of users, both expert and non-expert (including those who are not able to read musical notation fluently) and uses examples from two highly contrasting musical repertoires: Gustav Mahler's symphonies; and Renaissance choral music.

In the Mahler case, we shall ask users to add their own comments, as further annotations, to a set of annotations previously entered by expert scholars. These will include references and links to other media sources, such as audio/video recordings, which users can audition; they will be able to add their own examples of such links. A simple, and non-obtrusive questionnaire will be used to record their subjective responses to the interface, its useability and potential for their future work or amusement.

Similarly, an evaluation of the use of the interface with a contrasting repertoire of 16th-century choral music (already registered within TROMPA's Contributor Environment, and coordinated with the TROMPA Choir Singers' Use Case) will allow users to add annotations freely, and view those contributed by others. However, in this case, a further extension of the possibilities of the interface will be demonstrated by allowing users to search a musical database - external to, and independent from, TROMPA - using a query generated from notes in the displayed score.

6 Conclusions

In this document we have described the final state of the Digital Score Edition Component, which is used in various contexts in TROMPA.

In section 2 we presented an account of our work on generating MEI scores from the starting point of PDF documents, using two test-cases: Beethoven piano works and Mahler's 4th symphony. Our workflows for each use-case are described in Section 2.3.

Enhancements we have developed for the conversion of MusicXML scores to MEI are described in Section 3.1. A number of improvements we have contributed to the Verovio open-source score-rendering engine are described in 3.1.1; as a tool to aid the eventual editing of an MEI score within the Atom editor, we developed the MEI-friend Atom package, which is described in detail in 3.1.2.

Section 3.2 describes the Selectable Score interface; this module exposes a JSX/React component which ensures efficient real-time update of the state of the interface, and is implemented using MELD (Music Encoding and Linked Data), a semantic framework and set of client rendering and interaction libraries which is optimised for applications such as this.

A further enhancement is the Music Scholars Annotation Tool described in section 3.3. This allows users without advanced computer skills to annotate a score in an intuitive and musician-friendly manner. Annotations can be of various kinds: plain text; URL links to other documents of any kind; media files cued to a specific start-time, which then display in a pane within the main interface; images, similarly displayed; or they can be 'replies' to previously-entered annotations. We show how the TROMPA Contributor Environment enables the storage and retrieval of annotations for future use, such as those outlined in section 5.

As explained in section 5 (Future work and evaluation) the Digital Score edition component will be used and evaluated by groups of users in the final evaluation phase of TROMPA (months 35-6). The chosen repertoires are of contrasting music from the 16th and early 20th centuries (Mahler's 4th symphony and a collection of Renaissance choral music). In this way, we shall test the extent to which the current interface answers the needs of two levels of user: expert music scholars and music-lovers who are not able to fluently read the scores. A secondary aim of the evaluations is to build small collections of annotations plus a discourse upon the music based upon them which can lead to further discussion, all in a way which is linked and re-discoverable. In this way, we shall show the way forward to further explorations of musical repertoires using the facilities provided by TROMPA.

9 References

- [1] Weigl, D., & Page, K. (2017). A framework for distributed semantic annotation of musical score: "Take it to the bridge!". In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, Suzhou, China.
- [2] Crawford, T., & Lewis, R. (2016). Review: Music Encoding Initiative. In *Journal of the American Musicological Society*, 69 (1): 273–285. DOI: <https://doi.org/10.1525/jams.2016.69.1.273>
- [3] Lewis, D., Weigl, D. M., Bullivant, J., & Page, K. R. (2018). Publishing musicology using multimedia digital libraries: creating interactive articles through a framework for linked data and MEI. In *Proceedings of the 5th International Conference on Digital Libraries for Musicology (DLfM '18)*. ACM, New York, NY, USA, 21-25. DOI: <https://doi.org/10.1145/3273024.3273038>
- [4] Pugin, L., Zitellini, R., & Roland, P. (2014). Verovio: A library for Engraving MEI Music Notation into SVG. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pp. 107-112.
- [5] Raimond, Y., Abdallah, S. A., Sandler, M. B., & Giasson, F. (2007). The music ontology. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, pp. 417–422.
- [6] Fields, B., Page, K., De Roure, D., & Crawford, T. (2011). The segment ontology: Bridging music-generic and domain-specific. In *2011 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE.
- [7] Hankinson, A., Roland, P., & Fujinaga, I. (2011). The Music Encoding Initiative as a Document-Encoding Framework. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pp. 293–298
- [8] Kallionpää, M., Greenhalgh, C., Hazzard, A., Weigl, D., Page, K. R., & Benford, S. (2017). Composing and realising a game-like performance for disklavier and electronics. In *New Interfaces for Musical Expression (NIME 2017)*.
- [9] De Roure, D., Klyne, G., Pybus, J., Weigl, D. M., & Page, K. (2018). Music SOFA: An architecture for semantically informed recomposition of Digital Music Objects. In *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music* (pp. 33-41). ACM.

List of abbreviations

Abbreviation	Description
API	Application Programming Interface
CE	Trompa Contributor Environment

DSE	Digital Score Edition
EMO	Early Music Online
HTML	HyperText Markup Language
IMSLP	International Music Score Library Project
LDP	Linked Data Platform
MEI	Music Encoding Initiative
MELD	Music Encoding and Linked Data
OCR	Optical Character Recognition
OMR	Optical Music Recognition
PDF	Portable Document Format
REST	REpresentational State Transfer
TEI	Text Encoding Initiative
URI	Uniform Resource Identifier
XML	EXtensible Markup Language