A FRAMEWORK FOR DISTRIBUTED SEMANTIC ANNOTATION OF MUSICAL SCORE: "TAKE IT TO THE BRIDGE!"

David M. Weigl and Kevin R. Page
Oxford e-Research Centre
University of Oxford, United Kingdom
{david.weigl, kevin.page}@oerc.ox.ac.uk

ABSTRACT

Music notation expresses performance instructions in a way commonly understood by musicians, but printed paper parts are limited to encodings of static, a priori knowledge. In this paper we present a platform for multi-way communication between collaborating musicians through the dynamic modification of digital parts: the Music Encoding and Linked Data (MELD) framework for distributed real-time annotation of digital music scores. MELD users and software agents create semantic annotations of music concepts and relationships, which are associated with musical structure specified by the Music Encoding Initiative schema (MEI). Annotations are expressed in RDF, allowing alternative music vocabularies (e.g., popular vs. classical music structures) to be applied. The same underlying framework retrieves, distributes, and processes information that addresses semantically distinguishable music elements. Further knowledge is incorporated from external sources through the use of Linked Data. The RDF is also used to match annotation types and contexts to rendering actions which display the annotations upon the digital score. Here, we present a MELD implementation and deployment which augments the digital music scores used by musicians in a group performance, collaboratively changing the sequence within and between pieces in a set list.

1. INTRODUCTION

Music is a fundamental channel of communication [7], between musicians and an audience, but also among musicians performing together, and as a record of a performance. Inter-performer communication can support *semi-structured* performances such as jam sessions, where the set list is not entirely pre-determined, and repetitions and variations can be added within pieces. These decisions are made and communicated as the performance unfolds.

Music is richly structured, and annotations may serve to interlink musical content along such a structure. Annotations must be able to specifically *address* elements within this structure if they are to be described or related. In multimedia information systems, this is typically achieved using timeline anchors, offsets along a reference recording specified, e.g., in milliseconds. Such timed offsets are not intrinsically *musically* meaningful without context, limiting their use when no reference recording is available.

We can address part of this issue using the Music Encoding Initiative XML schema (MEI; [6]). MEI comprehensively expresses the classes, attributes, and data types required to encode a broad range of musical documents and structures. It does not, however, include or reference concepts, relationships, or existing descriptive forms of multimedia Linked Data external to its schema.

We present the Music Encoding and Linked Data (MELD) framework and implementation architecture that augments and extends MEI structures with semantic Web Annotations capable of addressing musically meaningful score sections. Through its use of Linked Data, our approach deploys knowledge structures expressing relationships unconstrained by boundaries of encoding schema, musical sub-domain, or use-case context, supporting retrieval of a wide range of music information. We employ the flexible and extensible Web Annotation model. New kinds of annotations are easily incorporated through customisation of the MELD JavaScript web-client via drop-in rendering and interaction handlers. Annotations, captured in the context of the performance session with provenance information, can seamlessly reference external data sources, and can in turn be referenced for external analysis, reuse, and repurposing in other contexts.

To demonstrate the feasibility of our approach, we present a prototypical implementation of a performance scenario which collects, distributes, and displays semantic annotations of digital music score in a live jam session.

2. RELATED WORK

Previous projects have applied digital notation to music performance scenarios for just-in-time composition and computer-assisted generation of musical score (e.g., [5, 23, 24]). While fascinating, these approaches are concerned with generating, rather than augmenting, musical score in real-time performance scenarios. Our work, in contrast, concerns the flexible targetting and interlinking of music resources and resource fragments within a framework of meaningfully structured music information.

Attribution: David M. Weigl and Kevin R. Page. "A framework for distributed semantic annotation of musical score: "Take it to the bridge!"", 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

2.1 Addressing Musical Content

Although typically delivered in a linear rendition, music is richly structured at various levels, from individual notes and performance directions to higher-level musical sections (intro, verse, chorus, bridge; all terms established in western popular music). Viewed as hyperstructures [2], these concepts can be annotated with extra-musical information. Such annotations may be anchored to representations of the music using media fragments [1] [20] expressing temporal positions along a reference timeline in milliseconds, beat instances, or MIDI clock ticks (e.g., [4, 16]).

Temporal anchors are widely used in multimedia information systems – for instance to link to specific scan positions within YouTube videos [2]. As such anchors are not musically meaningful, their usefulness is limited when targeting music in conceptual terms, rather than in terms of a reference recording, such as when annotating music score.

The Music Encoding Initative (MEI; [6]) provides an XML schema encompassing a comprehensive representation of musical structure. Content is cleanly separated from presentation [14], allowing the identification and addressing of (elements of) a musical work – from an entire composition, to a collection of notes constituting a phrase within a particular measure on a specific instrumental part. MEI arranges musical elements, each of which may be named with an XML identifier, within a well-specified hierarchy. These named elements provide anchor points for annotations targeting a musically meaningful structure.

The Open MEI Addressability Service (OMAS; [21]) addresses granular portions of music notation using offsets employing units of musical structure (measures, staves, beats), rather than temporal units. OMAS responds to such an offset specification (supplied via a templated URI syntax) by generating MEI documents containing copies of the specified portion of source MEI, resulting in new resources containing only the portions of music to be addressed. This enables the addressing of musical score without requiring a reference timeline. However, it is not equivalent to addressing a fragment of a resource within its (source) context, a requirement when using the score as a dynamic communication framework between multiple performers. Although specifying offsets in musical terms, OMAS does not directly address musically meaningful sections of the score (e.g., “verse 1,” “chorus 2,” “bridge”).

2.2 Expressing Musical Relationships

Linked Data extends the structure of the World Wide Web by employing URIs to specify directed relationships between data instances. These data instances are themselves encoded by URIs or represented by literal values. In the music information domain, Linked Data has been employed to describe musical resources in terms of associated catalogue metadata (e.g., [8, 22]); to publish features derived from audio-signal content along with associated provenance metadata [12, 13]; and to transcribe symbolic music content [11]. Throughout this article, we apply Linked Data to express annotations about musical structure from a music performance perspective.

Several ontologies – Linked Data formalisations of classes, properties, and relationships within musical subdomains – support the relation, interlinking, reuse, and repurposing of music information within and between data sets and associated applications. We now discuss three pertinent examples: the Music Ontology [17]; the Segment Ontology [3]; and the Common Hierarchical Abstract Representation of Music (CHARM) ontology [9].

While it does not primarily focus on music *performance*, the Music Ontology is a widely used data model describing terms and relationships around the production of musical works, actors (e.g., artists, composers), items (e.g., recordings, published scores), and events (e.g., performances). Its classes extend the Functional Requirements for Bibliographic Records (FRBR) ontology [3], discriminating between musical entities at different levels of abstraction, ranging from (at the most abstract level) the intellectual conception of a musical *work*, to its *expression* (conceptualised, e.g., as musical score), to an embodied *manifestation* (e.g., a publication of the musical score), to (at the most concrete level) a physical *item* representing a single exemplar of a manifestation (e.g., a musician’s personal copy of the published score).

The Segment Ontology represents music as comprised of segments ordered along an abstractly defined axis (the *segment line*). These music-generic segments are bridged to different ontological structures expressing elements of musical form appropriate to specific musical sub-domains (e.g., intro, verse, chorus, bridge; or sonata, minuet, trio, or fugue). This separation of concerns supports cross-application to different musical domains and use cases.

CHARM describes music at a fundamental level of pitches, times, and durations, expressing statements as logical formulae operating upon abstract data types. In this paper, we apply more concrete conceptualisations of musical score sufficient for the presented use case; but we invite the prospect of ontological mappings from CHARM to our framework, which could offer intriguing opportunities for re-use and extension for music analytical purposes.

3. MELD FRAMEWORK

The MELD [4] semantic framework combines and augments pertinent subsets of a number of ontologies in a semantic scaffold supporting dynamic distributed annotation of musical score (Figure 1). The Music and Segment Ontologies describe a musical *work*, the music score – a collection of musical segments ordered along a segment line that *express* the work (in FRBR terms) – and finally its *manifestation* as a *published* score encoded as MEI. Collections of MEI fragments, manifestations of musical segments embodied within the published score, anchor annotations processed by rendering and interaction handlers (Section 4). These elements form the core of the MELD semantic framework.

[1] http://www.w3.org/TR/media-frags/

[2] https://developers.google.com/youtube/player_parameters#start

[3] http://vocab.org/frbr/core.html

[4] http://github.com/oerc-music/meld

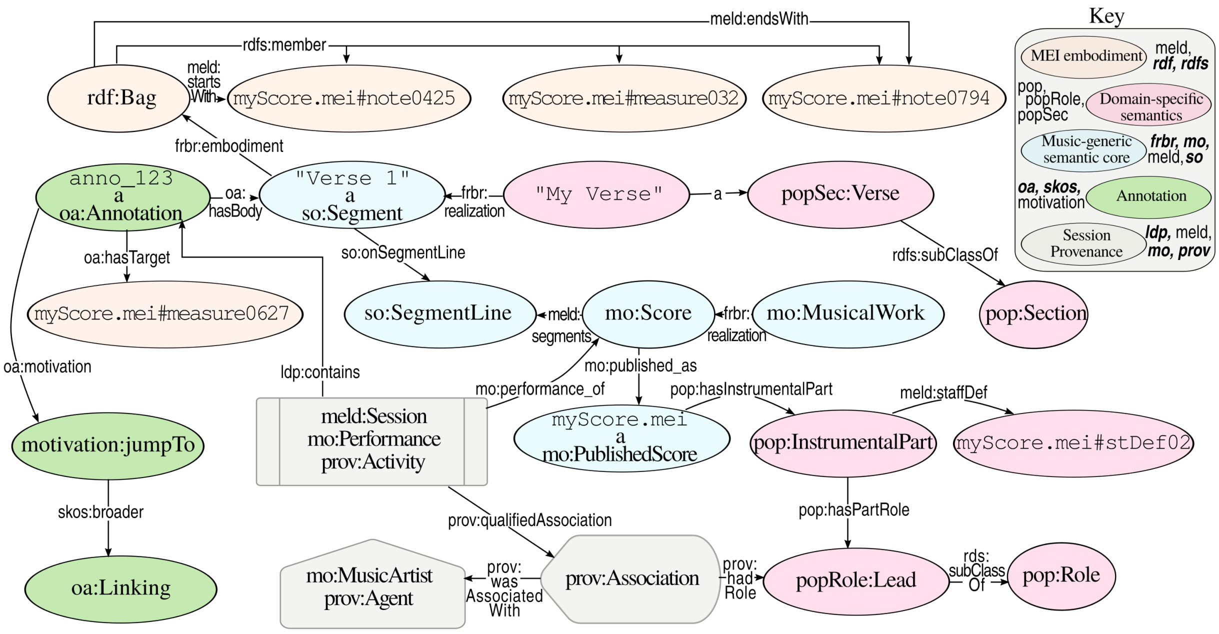

Figure 1. MELD semantic framework. Annotations address music segments embodied in a published score (MEI) resource. A music-generic core is linked to but seperable from domain-specific entities instantiating concrete music sections. Key (top right): External ontologies in ***bold italics***. ***frbr:*** Functional Requirements for Bibliographic Records Core; ***ldp***: Linked Data Platform Vocabulary; *meld*: MELD Vocabulary; ***mo:*** Music Ontology; *motivation:* MELD-specific oa motivations; ***oa:*** Web Annotation Ontology; *pop, popRole, popSec*: Pop music domain-specific semantics; ***prov***: PROV Namespace; ***rdf:*** RDF Concepts; ***rdfs:*** RDF Schema; ***skos:*** Simple Knowledge Organization System; ***so:*** Segment Ontology.

These music-generic structures are linked with, but separable from, components expressing domain-specific entities associated with the concrete instantiations of musical segments. Here, we have specified a taxonomy of popular music performance terms sufficient to accomodate our use case. Notably, these domain- and use-case-specific taxonomies may be modularly replaced by other ontological structures reflecting different domains (e.g., popular vs. classical music) or use cases (e.g., annotations supporting music performance vs. musicological scholarship) without modification of the core.

As each entity, class, and relationship in the framework is assigned its own URI, the entire ontological structure, as well as the generated annotation and session metadata, is part of a wider web of Linked Data. This enables the seamless inclusion of external information within MELD annotations, as well as the referencing, reuse, and repurposing of the generated information by external services.

3.1 MELD Annotations

MELD annotations build upon the Web Annotation Data Model, [5] a W3C recommendation providing an extensible, interoperable, machine-readable means of creating annotations by asserting relationships between a set of connected resources, typically an annotation *body* and a *target* (or target resource fragment).

Web Annotations may be associated with an explicit *motivation* formalising the given annotation's intended purpose. In MELD, domain- and use-case-specific rendering and interaction clients (Section 4.2) make use of this information to map the annotation to corresponding rendering and interaction handlers to effect changes to the score displayed to the user. By defining MELD-specific motivations subclassing generic ones specified within the Web Annotation model, we promote reuse and repurposing of MELD annotations in external contexts implementing Web Annotation standards.

Web Annotations may also be associated with an intended *audience* to whom a given annotation applies. In MELD, this information is used to address annotations to only certain specified participants – for instance, the player of a given part within a session. By default, annotations that do not specify an audience are made available to every client associated with a given session.

3.2 Domain Ontologies

We have specified a taxonomy describing sections and part roles of popular music. Abstract, music-generic ordered segments of a score are associated with more concrete notions of musical sections appropriate to popular music (e.g., "intro", "verse", "chorus", "bridge") via the Segment Ontology. In its original conception, this ontology bridges music-generic and domain-specific conceptions of musical segmentation by mapping an abstract segment line to a concrete reference timeline manifested along a musical recording. To avoid the requirement of such a reference, we instead anchor music-generic segments to

[5] https://www.w3.org/TR/annotation-model/

domain-specific musical sections, embodied as manifestations within the published score. These are represented as collections of media fragment URIs specifying the named MEI elements that comprise the given section. Where reference recordings form a part of the use case, both approaches are applicable, enabling flexible, complementary structuring of music information via temporal, symbolic, and semantic anchors. The simplicity of repurposing the Segment Ontology's bridging mechanism within our novel context is afforded by our use of Linked Data.

The published score, represented in MELD by an MEI resource, is also associated with domain-specific part roles (e.g., "lead", "bass", "rhythm") anchored within the MEI via fragment URIs specifying the corresponding MEI staff definition container element. We employ the PROV Ontology to express and track the provenance of relationships associating specific musicians with such part roles within the context of a particular performance session.

By virtue of the clean separation between *music-generic* and *popular-music-performance-specific* ontological structures, the domain-specific structures may be swapped out to address other use contexts while retaining the rest of the presented framework, e.g. for an analytical ontology of musicological terms supporting the use of digital score annotations to illustrate points in scholarly musicological arguments. This flexibility of ontological schema is another key affordance of Linked Data.

4. MELD ARCHITECTURE

The MELD architecture (Figure 2) implements server- and client-side components: RESTful web services are used to manage session and annotation resources; client-side rendering and interaction handlers display and update annotated digital score parts relevant to each user, as well as capturing user interactions and updating server-side resources with interaction outcomes.

4.1 Web Services

MELD annotation and session management services are implemented using a Python web server capable of handling operations on RDF and JSON-LD datasets encoding the collection of MELD sessions, as well as the performer part-roles and annotations associated with each session.

4.1.1 Performance Session Service

The server exposes a RESTful web service providing access to a resource representing the list of all MELD sessions available to a user (requested via HTTP GET). In order to create a new session, a Linked Data representation of a basic session resource, related by a `mo:performance_of` predicate to the published score MEI resource, is posted (via HTTP POST) to the list. The server mints a URI to represent the new session. A 'join' resource is exposed to establish qualified associations between the respective performer and an instrumental part in the session context.

4.1.2 Annotation Service

Clients interact with the annotation service using an API based on the Web Annotation Protocol [6], which specifies transport mechanisms for creating and managing annotations that are consistent with Web Architecture and REST best practices. This involves casting each session as an *annotation container*, a form of Linked Data Platform (LDP) container [7] with additional constraints derived from the Web Annotation data model. New annotations are posted to the annotation container, where they are associated with the session using `ldp:contains` relationships.

MELD extends the Web Annotation model by tracking the *state* of each annotation associated with each performer, in order to support the dynamic, real-time nature of the distributed annotation activity. Annotations are created in a *raw* state. Upon the occurrence of certain events (e.g., user interaction), clients may optionally effect a *processed* annotation state (via HTTP PATCH), signifying that the annotation has been handled and is no longer of relevance within the session context. This approach is preferable to simply deleting the handled annotation (e.g., via HTTP DELETE), as it may remain relevant in external contexts – for instance for post-session review by the performers, or by other interested observers.

4.2 Rendering and Interaction Clients

A JavaScript web client is responsible for rendering MEI score parts to the user. The client dynamically augments this display with currently relevant annotations; handles user interactions; and communicates interaction outcomes using the MELD web services (Section 4.1).

The procedure is illustrated in Figure 2. The client processes a JSON-LD [19] representation of the graph associated with the session resource, framed [18] to include only relevant annotations by filtering on audience and state (Sections 3.1 and 4.1.2). It retrieves the MEI resource associated with the session (HTTP GET), and renders the encoded score using Verovio [15], a tool that produces SVG engravings of MEI-encoded music notation. Crucially, Verovio retains the MEI hierarchy and element identifiers into the produced SVG output, supporting the addressing of graphical score elements for visual markup and dynamic interaction through the web browser.

Each annotation is mapped to a corresponding handler, which may be customised to a specific motivation and use-case using CSS styling and drop-in JavaScript functions. User-interaction outcomes trigger AJAX calls using the web services to POST a new annotation, or to PATCH an annotation's state from *raw* to *processed*.

Each client continuously polls the session resource (using HTTP GET), maintaining the annotated score at its latest state in near real-time. Simple resource hashes (HTTP entity tags) and atomic server-side file writing reduce network traffic and address race conditions inherent in distributed real-time interaction: the former by supplying

[6] `https://www.w3.org/TR/annotation-protocol/`

[7] `https://www.w3.org/TR/ldp-primer/`. LDP provides read-write access to RDF datasets via RESTful HTTP services.

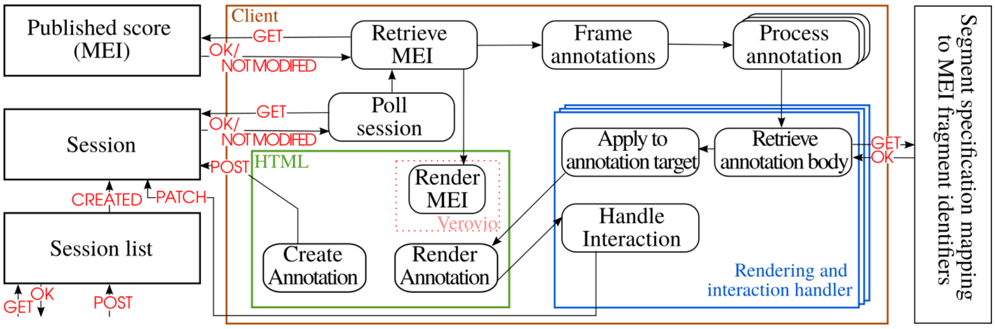

Figure 2. MELD architecture: performance session and annotation web services; MELD client; and rendering and interaction handlers. A tight polling loop, employing entity tagging techniques to reduce network load and address concurrency concerns, distributes and maintains annotation state between performers in near-real-time.

lightweight HTTP 304 ("Not Modified") responses that exclude the resource body when a resource has not changed from its state last seen by the client; and the latter by the server rejecting changes (via HTTP 412 – "Precondition Failed") upon an entity tag mismatch, prompting the client to GET the latest version of the session resource before reattempting its modification.

5. SCENARIO

To validate the feasibility of our proposed approach, we have produced an implementation of MELD to support a simple scenario where musical performers collaborate within a semi-structured performance environment.

5.1 Motivation

The selected use case is a performance – such as a jam session – where collaborating musicians make fluid, ad-hoc decisions about song repetitions and transitions, rather than adhering to a pre-determined set list. Musicians may add directions to change dynamic or stylistic elements of the performance, e.g., to reference a particular prior recording they wished the group to emulate in style or interpretations; or they may incorporate structural directions, e.g., to repeat a chorus or a verse, or to move to a bridge section. Such directions transcend the symbolic representation of the music being performed, and may draw upon significant contextual information external to the performance itself, for instance to adopt the style of a certain artist, or to transition to another song by this artist.

5.2 Implementation

A session, corresponding to the performance of a given song, is represented as an LDP container. It contains annotations, and connects participating performers to instrumental part roles via qualified associations (Figure 1). The corresponding MEI staff definition elements are used to filter the music structure in order to display only the relevant instrumental parts to each performer on a personal touchscreen (Figure 4). Only annotations pertinent to their performance are displayed, using the Web Annotation *audience* property as a filtering mechanism (Section 3.1).

Figure 3. The MELD client displays the annotated digital score. A modal action pane enables users to generate annotations enacting jumps within a piece, or queueing actions determining the next piece to be performed.

Rendering and interaction handlers operate on annotations that enable performers to collaboratively change the performance sequence within and between musical scores. Annotations are generated by using the touch screen to interact with a modal action pane situated below the musical score (Figure 3).

Navigational changes of sequence within a score – e.g., a jump to the bridge section – are requested by specifying a jump *source* (highlighted on the score in red) and *destination* (highlighted in green). The former is specified by tapping on a measure of the score. The action pane then displays a list of musical sections representing potential destinations. This list is automatically retrieved from the Linked Data, by traversal of the segments on the segment line associated with the current session's score. When a selection is made, an annotation is POSTed to the session, specifying an annotation target (the fragment URI of the jump source measure); an annotation body (the URI of the selected destination); and `motivation:jumpTo` (a custom specialisation of Web Annotation's `oa:linking`) as the motivation. All performer clients retrieve the annotation upon the next polling cycle.

Figure 4. MEI staff definition element identifiers are associated with each performer in a session resource (Figure 1), enabling filtering according to role. Here, an annotation targeting a segment embodied as `myScore.mei#msr0823` retrieves the corresponding notes for Lead or Bass, according to the performer's part-role association within the session.

Upon tapping on the red source measure, the rendering and interaction handler flips to the score page containing the destination measure. The highlights then fade, and the performer's copy of the annotation is PATCHed as *processed* to avoid rerendering of stale information. [8]

Annotations representing queueing instructions for navigation between songs may target individual source measures as described above, or the URI of the current score (in which case the rendering and interaction handler creates a "next piece" button in place of the "next page" button on the last page of the current piece). Songs to transition to can be selected according to a criteria such as "More songs by this artist". These parameterise SPARQL queries [9], in this case with the URI of the artist from the current score's MEI responsibility statement. The SPARQL query retrieves a list of songs by this artist from DBPedia [10] [1], matched against a cache of available MEI resources stored on a local triplestore (Linked Data database). Upon selecting a song, the annotator's client requests the creation of a new session associated with the corresponding MEI resource. It then posts an annotation to the current session, instructing all performers' clients to join the next session when an interaction event on the annotation target (the jump source) is handled.

6. CONCLUSIONS AND FUTURE WORK

We have presented the MELD framework and architecture applying musical structure as a semantic spine for real-time annotation of digital music score. RESTful web services manage the retrieval and distribution of annotations created by user interactions in a performance session. Annotations address MEI-encoded score elements using Linked Data, enabling flexible reuse, repurposing, and interlinking of the generated information in external contexts. The framework and associated implementation architecture comprise separable music-generic and domain-specific semantic structures, and modular rendering and interaction handlers, allowing components to target different music domains and use cases. We have validated the feasibility of the proposed framework through a MELD implementation supporting multi-way communication between musicians collaboratively changing the performance sequence within and between pieces in a group performance.

In future work, we will focus on extending the capabilities of the described framework to incorporate annotations enacting modifications of the MEI structure, for instance requesting changes in dynamics. Such modifications are readily incorporated within the client polling cycle (Section 4.2), given the speedy rendering performance of Verovio and the relation of the polled session resource (via `mo:performance_of`) to the MEI notation being rendered. However, session management complexities including MEI resource duplication, versioning, and the capture of provenance information will need to be accomodated, and potential licensing issues carefully considered.

While physical interactions with musical score are commonplace in music performance – consider the necessity of paging through paper parts – care must be taken in ongoing interface development to minimise additional cognitive load upon the performer. We have developed an alternative, multimodal interaction mechanism by integrating the MELD framework with a technology supporting real-time triggering of HTTP actions in response to specified patterns matched to an audio or MIDI stream. This variation of MELD has been successfully applied to drive a gamified composition for disklavier and electronics [10], demonstrating a more complex performance interaction than the simple scenario used to explain the MELD framework in this paper. Other interaction paradigms are imaginable, for instance by means of foot pedals, or voice commands.

Finally, we are exploring the application of MELD in non-performance-related use cases. MELD annotations may be specified to target MEI and other digital media, including audio, images, and textual commentary. Thus interlinked, such media resources, annotated to target musically meaningful sections, may be used to illustrate scholarly musicological arguments. Together with the work presented here, these applications demonstrate the utility and flexibility of adopting a semantic framework anchored in addressable musical structure to express, retrieve, and distribute information in a variety of contexts and use cases.

[8] Measures are defined above staves within the MEI hierarchy, meaning that the notes contained in the targeted measures, and the pages they appear on, likely differ between performers, according to their instrumental part associations

[9] SPARQL is a query language for Linked Data resources, analogous to SQL queries against relational databases, but capable of retrieving data spanning local and external data sets

[10] DBPedia publishes Wikipedia information as structured Linked Data

7. ACKNOWLEDGEMENTS

This work is undertaken as part of the Fusing Audio and Semantic Technologies for Intelligent Music Production and Consumption project, funded by the UK Engineering and Physical Sciences Research Council under grant number EP/L019981/1, a collaboration between Queen Mary University of London, the University of Nottingham, and the University of Oxford.

We gratefully acknowledge the support of our colleagues. Particularly, we thank David Lewis for guidance and insight during the development of the MELD framework, and his collaboration in extending our approach to applications in musicological scholarship; Graham Klyne for consultation regarding PROV-O mappings; and Leia Yeomans for her valuable assistance in realising the graph layout for Figure 1 of this paper.

8. REFERENCES

[1] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sren Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009. The Web of Data.

[2] David C. De Roure, Don G. Cruickshank, Danius T. Michaelides, Kevin R. Page, and Mark J. Weal. On hyperstructure and musical structure. In *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*, pages 95–104. ACM, 2002.

[3] Ben Fields, Kevin Page, David De Roure, and Tim Crawford. The segment ontology: Bridging music-generic and domain-specific. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[4] Martin Gasser, Andreas Arzt, Thassilo Gadermaier, Maarten Grachten, and Gerhard Widmer. Classical music on the web–user interfaces and data representations. In *Proc. 16th International Society for Music Information Retrieval Conference*, 2015.

[5] Georg Hajdu. Real-time composition and notation in network music environments. In *International Computer Music Conference*, 2008.

[6] Andrew Hankinson, Perry Roland, and Ichiro Fujinaga. The Music Encoding Initiative as a Document-Encoding Framework. In *Proc. 12th International Society for Music Information Retrieval Conference*, pages 293–298, 2011.

[7] David J. Hargreaves, Raymond MacDonald, and Dorothy Miell. How do people communicate using music? In *Musical communication*, pages 1–25. Oxford University Press Oxford, 2005.

[8] Kurt Jacobson, Simon Dixon, and Mark Sandler. Linkedbrainz: Providing the musicbrainz next generation schema as linked data. In *Late-breaking demo session at the 11th International Society for Music Information Retrieval Conference*, 2010.

[9] Kurt Jacobson, Simon Dixon, and Mark Sandler. An ontology for abstract, hierarchical music representation. In *Late-breaking demo session at the 16th International Society for Music Information Retrieval Conference*, 2015.

[10] Maria Kallionpää, Chris Greenhalgh, Adrian Hazzard, David M. Weigl, Kevin R. Page, and Steve Benford. Composing and realising a game-like performance for disklavier and electronics. In *New Interfaces for Musical Expression (NIME)*, 2017.

[11] Albert Meroño-Peñuela and Rinke Hoekstra. The song remains the same: Lossless conversion and streaming of MIDI to RDF and back. In *International Semantic Web Conference*, pages 194–199. Springer, 2016.

[12] Kevin R. Page, Sean Bechhofer, Georgy Fazekas, David M. Weigl, and Thomas Wilmering. Realising a layered digital library: Exploration and analysis of the live music archive through linked data. In *Joint Conference on Digital Libraries*, 2017.

[13] Kevin R. Page, Benjamin Fields, Bart J. Nagel, Gianni O'Neill, David C. De Roure, and Tim Crawford. Semantics for music analysis through linked data: How country is my country? In *e-Science (e-Science), 2010 IEEE Sixth International Conference on*, pages 41–48. IEEE, 2010.

[14] Laurent Pugin, Johannes Kepper, Perry Roland, Maja Hartwig, and Andrew Hankinson. Separating Presentation and Content in MEI. In *Proc. 13th International Society for Music Information Retrieval Conference*, pages 505–510, 2012.

[15] Laurent Pugin, Rodolfo Zitellini, and Perry Roland. Verovio: A library for engraving MEI music notation into SVG. In *Proc. 15th International Society for Music Information Retrieval Conference*, pages 107–112, 2014.

[16] Fazilatur Rahman and Jawed Siddiqi. Semantic annotation of digital music. *Journal of Computer and System Sciences*, 78(4):1219 – 1231, 2012.

[17] Yves Raimond, Samer A Abdallah, Mark B Sandler, and Frederick Giasson. The music ontology. In *8th International Conference on Music Information Retrieval*, pages 417–422, 2007.

[18] Manu Sporny, Gregg Kellogg, Dave Longley, and Markus Lanthaler. JSON-LD framing 1.1: An application programming interface for the JSON-LD syntax. Community group draft report, W3C, 2017. `http://json-ld.org/spec/latest/json-ld-framing/`.

[19] Manu Sporny, Dave Longley, Gregg Kellogg, and Markus Lanthaler. JSON-LD 1.0: A JSON-based serialization for linked data. Recommendation, W3C, January 2014. http://www.w3.org/TR/json-ld/.

[20] Raphaël Troncy, Lynda Hardman, Jacco Van Ossenbruggen, and Michael Hausenblas. Identifying spatial and temporal media fragments on the web. In *W3C Video on the Web Workshop*, pages 4–9, 2007.

[21] Raffaele Viglianti. The music addressability API: A draft specification for addressing portions of music notation on the web. In *Proceedings of the 3rd International Workshop on Digital Libraries for Musicology*, DLfM 2016, pages 57–60, New York, NY, USA, 2016. ACM.

[22] David M. Weigl, David Lewis, Tim Crawford, Ian Knopke, and Kevin R. Page. On providing semantic alignment and unified access to music-library metadata. *International Journal on Digital Libraries*. Accepted for publication: Special issue on Digital Libraries for Musicology.

[23] Gerhard E Winkler. The realtime-score: A missing-link in computer-music performance. *Sound and Music Computing*, 4, 2004.

[24] Lonce Wyse and Jude Yew. A real-time score for collaborative just-in-time composition. *Organised Sound*, 19(3):260–267, 2014.