AN ONTOLOGY FOR AUDIO FEATURES

Alo Allik, György Fazekas, Mark Sandler
Queen Mary University of London
a.allik, g.fazekas, mark.sandler@qmul.ac.uk

ABSTRACT

A plurality of audio feature extraction toolsets and feature datasets are used by the MIR community. Their different conceptual organisation of features and output formats however present difficulties in exchanging or comparing data, while very limited means are provided to link features with content and provenance. These issues are hindering research reproducibility and the use of multiple tools in combination. We propose novel Semantic Web ontologies (1) to provide a common structure for feature data formats and (2) to represent computational workflows of audio features facilitating their comparison. The Audio Feature Ontology provides a descriptive framework for expressing different conceptualisations of and designing linked data formats for content-based audio features. To accommodate different views in organising features, the ontology does not impose a strict hierarchical structure, leaving this open to task and tool specific ontologies that derive from a common vocabulary. The ontologies are based on the analysis of existing feature extraction tools and the MIR literature, which was instrumental in guiding the design process. They are harmonised into a library of modular interlinked ontologies that describe the different entities and activities involved in music creation, production and consumption.

1. INTRODUCTION

Several content based audio feature extraction frameworks and toolsets have been developed over the past decades of MIR research aiming to provide a platform for distributing, sharing or deploying algorithms. While most tools have the potential to become widely adopted common platforms, it is most likely that a plurality of them will continue to be used by the community as well as adopters of MIR technology. However, diverging conceptual organisation of features and different output formats present difficulties when it comes to exchanging and comparing data, or producing annotated datasets. These issues are hindering research reproducibility as well as the use of multiple data or tool sets in a single application or experiment.

A growing demand for shared representations of computational extraction workflows and interoperable data formats is signified by several proposed formats, some of which are associated with feature extractor tools or services [3, 4, 8, 13, 19]. While existing formats for structuring and exchanging content-based audio feature data may satisfy tool or task specific requirements, there are still significant limitations in linking features produced in different data sources, as well as in providing generalised descriptions of audio features that would allow easier identification and comparison of algorithms that produce the data.

Semantic Web technologies facilitate formal descriptions of concepts, terms and relationships that enable implementations of automated reasoning and data aggregation systems to manage large amounts of information within a knowledge domain. Both in research and commercial use cases, it is becoming increasingly important to fuse cultural, contextual and content-based information. This may be achieved by leveraging Linked Data enabled by the use of shared ontologies and unique identification of entities. This not only offers the potential to simplify experiments and increase productivity in research activities traditionally relying on Web scraping, proprietary application programming interfaces or manual data collection, but also enables incorporation of increasingly larger and more complex datasets into research workflows.

We propose a modular approach towards ontological representation of audio features. Since there are many different ways to structure features depending on a specific task or theoretically motivated organising principle, a common representation would have to account for multiple conceptualisations of the domain and facilitate diverging representations of common features. This may be due to the "semantic gap" between low-level computational representations of audio signals and theoretical representations founded in acoustics or musicology. This semantic gap could potentially be bridged using Semantic Web technologies if high-level feature identification can be inferred from computational signatures. However, this functionality is currently beyond the scope of existing technologies. For example, Mel Frequency Cepstral Coefficients (MFCC), which are widely calculated in many tools and workflows, can be categorised as a "timbral" feature in the psychoacoustic or musicological sense, while from the computational point of view, MFCC could be labelled as a "cepstral" or "spectral" representation. The complexity arising from this makes music somewhat unique calling for a robust ontological treatment, although ontological representation of content-based features have been proposed in other domains including image processing [23].

Our framework consists of two separate components to distinguish between the abstract concepts describing the audio feature domain and more concrete classes that represent specific audio features and their computational signatures. The Audio Feature Ontology (AFO) is the more abstract component representing entities in the feature extraction process on different levels of abstraction. It describes the structure of processes in feature extraction workflow through phases of conceptualisation, modelling and implementation. The Audio Feature Vocabulary (AFV) then lists existing audio features providing the terms for tool and task specific ontologies without attempting to organise the features into a taxonomy.

2. BACKGROUND

Many recent developments in audio feature data formats employ JavaScript Object Notation (JSON), which is rapidly becoming a ubiquitous data interchange mechanism in a wide range of systems regardless of domain. Particularly, the JSON Annotated Music Specification (JAMS) [8] is a notable attempt to provide meaningful structure to audio feature data while maintaining simplicity and sustainability in representations, which the authors deem as the most crucial factors for wider adoption in the MIR community. A different JSON format for features has been developed in the AcousticBrainz project [19] with the intention of making available low and high level audio features for millions of music tracks. This resource provides the largest feature dataset to date while exposing the extraction algorithms in an open source environment.

It is evident that the simplicity of JSON combined with its structuring capabilities make it an attractive option, particularly compared to preceding alternatives including YAML, XML, Weka Attribute-Relation File Format (ARFF), the Sound Description Interchange Format (SDIF), and various delimited formats. All these formats enable communication between various workflow components and offer varying degrees of flexibility and expressivity. However, even the most recent JSON methods only provide a structured representation of feature data without the facility of linking these concepts semantically to other music related entities or data sources. For example, the JAMS definition does not address methodology that would enable detailed description and comparison of audio features nor does it provide a structured way of linking the feature data to the rest of the available metadata for a particular music track. Admittedly, the AcousticBrainz dataset does provide links to the MusicBrainz [1] database by global identifiers, but there is no mechanism to identify the features or compare them to those available in other extraction libraries. The Essentia library [3] that is used in the AcousticBrainz infrastructure for feature extraction is open source, thus providing access to the algorithms, but there is no formalised description of audio features beyond source code and documentation yet. Other feature extraction frameworks provide data exchange formats designed for particular workflows or specific tools. However, there is no common format shared by all the different tools and libraries. The motley of output formats is well demonstrated in the representations category of a recent evaluation of feature extraction toolboxes [15]. For example, the popular MATLAB MIR Toolbox export function outputs delimited files as well as ARFF, while Essentia provides YAML and JSON and the YAAFE library outputs CSV and HDF5. The MPEG-7 standard, used as benchmarks for other extraction tools provides an XML schema for a set of low-level descriptors, but the deficiencies highlighted above also apply in this case.

Semantic Web technologies provide domain modelling and linking methods considerably beyond the expressivity and interoperability of any of the solutions described above. The OWL family of ontology languages is designed to be flexible enough to deal with heterogeneous Web-based data sources. It is also built on strong logical foundations. It implies a conceptual difference between developing data formats and ontological modelling. The authors of [8] mention a common criticism that RDF-based MIR formats similarly to XML suffer from being non-obvious, verbose or confusing. However, the potential of meaningful representation of audio features and linking ability to other music related information outweighs these concerns. Ontological representation and linking of divergent domains is a difficult task, but should not be discarded lightly in favour of simplicity. The benefits of even relatively limited Semantic Web technologies for MIR research have been demonstrated on a number of occasions. For example, the proof-of-concept system described in [17] enables increased automation and simplification of research workflows and encourages resource reuse and validation by combining several existing ontologies and Semantic Web resources, including the Music Ontology [2], GeoNames [3], DBTune [4], and the Open Archives Initiative Object Reuse and Exchange (OAI-ORE). A system for MIR workflow preservation has been proposed in [12], which emphasises the importance of representing and preserving the context of entire research processes and describes a Context Model of a typical MIR workflow as a Semantic Web ontology.

The original version of the Audio Feature Ontology was created within a framework of a harmonised library of modular music-related Semantic Web ontologies [4], built around the core Music Ontology [20]. This library relies on widely adopted Semantic Web ontologies such as the Friend of a Friend (FOAF) vocabulary, as well as domain specific ontologies for describing intellectual works (FRBR) and complex associations of domain objects with time-based events (Event and Timeline ontologies). The library also provides a set of extensions describing music specific concepts including music similarity [9] and the production of musical works in the recording studio [5]. Since its publication, it has been integrated in several research projects, including the Networked Environment for Music Analysis (NEMA) [24], the Computational Analysis of the Live Music Archive (CALMA) [2] as well as commercial applications, e.g. the BBC and its music Web-

[1] `http://musicbrainz.org`

[2] `http://musicontology.com/`

[3] `http://www.geonames.org/`

[4] `http://dbtune.org/`

site [5]. This ontology provides a model for structuring and publishing content-derived information about audio recordings and allows linking this information to concepts in the Music Ontology framework. However, it does not provide a comprehensive vocabulary of audio features or computational feature extraction workflows. It also lacks concepts to support development of more specific feature extraction ontologies. Structurally, it conflates musicological and computational concepts to an extent that makes it inflexible for certain modelling requirements as suggested in [7]. In order to address these issues, the updated Audio Feature Ontology separates abstract ontological concepts from more specific vocabulary terminology, supplies methodology for extraction workflow descriptions, and increases flexibility for modelling of task and tool specific ontologies.

3. ONTOLOGY ENGINEERING

In order to gain a better understanding of the MIR domain and user needs, a catalogue of audio features was first compiled based on a review of relevant literature, existing feature extraction tools and research workflows. The first phase of this process involved extracting information about features from journal articles, source code and existing structured data sources. This information was subsequently collated into a linked data resource to serve as a foundation for the ontology engineering process. There was no attempt at explicit classification of features into a hierarchical taxonomy. Source code was parsed from a number of open source feature extraction packages including CLAM [1], CoMIRVA [21] jMIR (jAudio) [13], LibXtract [6], Marsyas [7], Essentia [3] and YAAFE [11]. Existing linked resource representations of the Vamp plugins provided easy access to all the features available for download on the Vamp plugins Website [8]. Manual extraction was used for the packages which did not provide suitable access for automatic parsing or which were reviewed in journal articles, including the Matlab toolboxes (MIR Toolbox and Timbre Toolbox), Aubio [9], Cuidado [18], PsySound3 [10], sMIRk [6], SuperCollider SCMIR toolkit, and some of the more recent MIREX submissions. A simple automatic matching procedure was employed to identify synonymous features using a Levenshtein distance algorithm, which aided the compilation of a feature synonym dictionary.

The catalogue was created in linked data format using the Python RDFLib library [11], which enables quick and easy serialisation of linked data into various formats. The catalogue lists feature objects and their attributes and serves as the foundation for a hybrid ontology engineering process combining manual and semi-automatic approaches. The catalogue identifies approximately 400 distinct features and thereby significantly increases the scope of the original ontology, which supports identifying about 30 entities. The catalogue has been published online [12] and allows querying subsets of popular features computed in feature extraction tools to help define the scope and domain boundaries of the ontology. It also sheds light on the range of classifications of features inherent in different software tools and libraries, as well as conceptualisations of the domain in journal articles. Figure 1 shows three divergent organisations of features from very different sources.

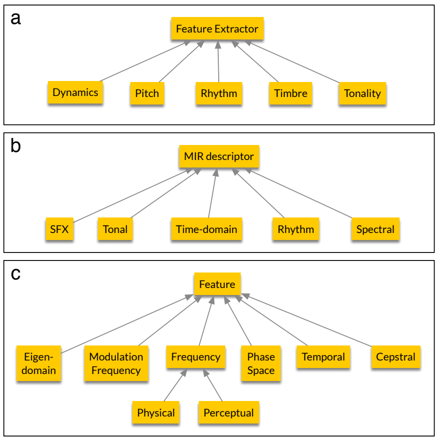

Figure 1. Three different taxonomies of audio features extracted from (a) MIR Toolbox, (b) Essentia, and (c) Mitrovic et al. [14]

The catalogue exemplifies the diversity of viewpoints on classification of features within the community. It is clear that in some cases audio features are categorised according to musicological concepts, such as pitch, rhythm and timbre, while in others, the classification is based on the computational workflows used in calculating the features or a combination of different domains depending on the task. Consequently, there is no need to impose a deeply taxonomical structure on the collected audio features, rather the resulting ontology should be focused on facilitating structured feature data representation that is flexible enough to accommodate all these diverging organisational principles.

4. CORE ONTOLOGY MODEL

The most significant updates to the original ontology model are designed to address a number of requirements determined during the engineering process. The proposed updates are intended to:

- provide comprehensive vocabulary of audio features
- define terms for capturing computational feature extraction workflows
- support development of domain and task specific ontologies for existing extraction tools
- restructure concept inheritance for more flexible and sustainable feature data representation

[5] http://bbc.co.uk/music/
[6] http://libxtract.sourceforge.net
[7] http://marsyas.info
[8] http://www.vamp-plugins.org/
[9] http://aubio.org/
[10] http://psysound.wikidot.com/
[11] https://github.com/RDFLib/rdflib
[12] http://sovarr.c4dm.eecs.qmul.ac.uk/af/catalog/1.0#

- facilitate design of linked data formats that combine strong logical foundations of ontological structuring with simplicity of representations.

The fundamental structure of the ontology has changed in a couple of key aspects. The core structure of the framework separates the underlying classes that represent abstract concepts in the domain from specific named entities. This results in the two main components of the framework defined as Audio Feature Ontology (AFO, `http://w3id.org/afo/onto/1.1#`) and Audio Feature Vocabulary (AFV, `http://w3id.org/afo/vocab/1.1#`). The main differences also include introducing levels of abstraction to the class structure and reorganising the inheritance model. The different layers of abstraction represent the feature design process from conceptualisation of a feature, through modelling a computational workflow, to implementation and instantiation in a specific computational context. For example, the abstract concept of Chromagram is separate from its model which involves a sequence of computational operations like cutting an audio signal into frames, calculating the Discrete Fourier Transform for each frame, etc. (see Section 5.1 for a more detailed example, and [16] for different methods for extracting Chroma-based features). The abstract workflow model can be implemented using various programming languages as components of different feature extraction software applications or libraries. Thus, this layer enables distinguishing a Chromagram implementation as a Vamp plugin from a Chromagram extractor in MIR Toolbox. The most concrete layer represents the feature extraction instance, for example, to reflect the differences of operating systems or hardware on which the extraction occurred. The layered model is shown in Figure 2.

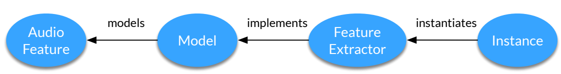

Figure 2. The Audio Feature Ontology core model with four levels of abstraction

The core model of the ontology retains original attributes to distinguish audio features by temporal characteristics and data density. It relies on the Event and Timeline ontologies to provide the primary structuring concepts for feature data representation. Temporal characteristics classify feature data either into instantaneous points in time - e.g. event onsets or tonal change moments - or events with known time duration. Data density attributes allow describing how a feature relates to the extent of an audio file: whether it is scattered and occurs irregularly over the course of the audio signal, or the feature is calculated at regular intervals and fixed duration. The change in the inheritance model removes the music-specific subclassing of **afo:Point**, **afo:Segment**, and **afo:Signal** classes which was claimed to make feature representation less flexible in certain use cases [7]. The Segment Ontology was proposed as a solution to get around these limitations [7], in which the Segment class functions as a music-generic dimension between explicitly temporal and implicitly temporal concepts, thus enabling multiple concurrent domain-specific concepts to be represented. An alternative solution is to subclass **afo:Point**, **afo:Segment**, and **afo:Signal** directly from **afo:AudioFeature**, which, in turn, is a subclass of **event:Event**. In this case, the feature extraction data can be directly linked to the corresponding term in AFV without being constrained by domain or task specific class definitions. This way, it is not necessary to add the Segment Ontology concepts to feature representations, thereby simplifying the descriptions.

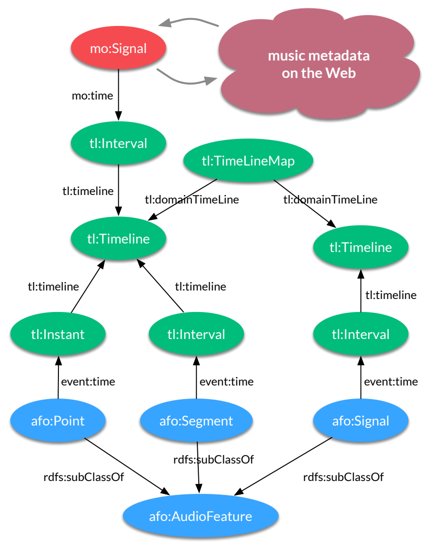

Figure 3. Framework model showing how feature data representation is linked with music metadata resources on the Web using temporal entities defined in the Timeline ontology

Audio features collated from literature and extraction software are defined as subclasses in the AFV. An illustrative Turtle-syntax representation that shows the basic principle of how subclassing **afo:AudioFeature** functions in the context of annotating both sparse and dense features is provided in Section 5.2. The other purpose of the vocabulary is to define computational extraction workflow descriptions, so that features can be more easily identified and compared by their respective computational signatures. The following section delves into this in more detail.

5. CASE STUDIES AND EVALUATION

5.1 Representing computational workflows

AFV defines terms for the tool and task specific ontologies and implements the model layer of the ontology framework. It is a clean version of the catalogue which only lists the features without any of their properties with many duplications of terms consolidated. This enables the definition of tool and task specific feature implementations and leaves any categorisation or taxonomic organisation to be specified in the implementation layer.

The vocabulary also specifies computational workflow models for some of the features which can be linked to from lower level ontologies. The computational workflow models are based on feature signatures as described in [14]. The signatures represent mathematical operations employed in the feature extraction process with each operation assigned a lexical symbol. It offers a compact description of each feature and enables an easier way of comparing features according to their extraction workflows. Converting the signatures into a linked data format to include them in the vocabulary involves defining a set of OWL classes that handle the representation and sequential nature of the calculations. The operations are implemented as sub-classes of three general classes: transformations, filters and aggregations. For each abstract feature, we define a model property. The OWL range of the model property is a ComputationalModel class in the Audio Feature Ontology namespace. The operation sequence can be defined through this object's operation sequence property. For example, the signature of the Chromagram feature defined in [14] as "f F l Σ", which designates a sequence of (1) windowing (f), (2) Discrete Fourier Transform (F), (3) logarithm (l) and (4) sum (Σ) is expressed as a sequence of RDF statements in Listing 1.

```
afv:Chromagram a owl:Class ;
  afo:model afv:ChromagramModel ;
  rdfs:subClassOf afo:AudioFeature .

afv:ChromagramModel a afo:Model;
  afo:sequence afv:Chromagram_operation_sequence_1 .

afv:Chromagram_operation_sequence_1 a afv:Windowing;
  afo:next_operation
       afv:Chromagram_operation_sequence_2 .

afv:Chromagram_operation_sequence_2 a
     afv:DiscreteFourierTransform;
  afo:next_operation
       afv:Chromagram_operation_sequence_3 .

afv:Chromagram_operation_sequence_3 a afv:Logarithm;
  afo:next_operation
       afv:Chromagram_operation_sequence_4 .

afv:Chromagram_operation_sequence_4 a
     afo:LastOperation, afv:Sum .
```

Listing 1. Description of a chromagram computation.

```
SELECT DISTINCT ?feature
WHERE {
  ?opid a afv:DiscreteCosineTransform .
  ?seqid afo:first_operation ?fopid .
  ?fopid afo:next_operation+ ?opid .

  OPTIONAL {
    ?model afo:operation_sequence ?seqid .
    ?feature afo:model ?model .
  }
}
```

Listing 2. Retrieving feature types involving the DCT.

This structure enables building SPARQL queries to retrieve comparative information on features from the vocabulary. For example, we can inquire which features in the vocabulary employ the Discrete Cosine Transform calculation by executing the query of Listing 2. The query will produce the following result:

```
afv:AutocorrelationMFCCs
afv:BarkscaleFrequencyCepstralCoefficients
afv:MelscaleFrequencyCepstralCoefficients
afv:ModifiedGroupDelay
afv:ModulationHarmonicCoefficients
afv:NoiseRobustAuditoryFeature
afv:PerceptualLinearPrediction
afv:RelativeSpectralPLP
```

5.2 Audio content description

In order to determine how well the AFO framework represents the audio feature extraction domain, we need to test its suitability for representing audio features in the context of particular use cases. We employ a task-based methodology to focus on evaluating the suitability of AFO in a feature extraction workflow. Task-based evaluation is based on having a set of pre-defined requirements and it may offer a measure of practical aspects, such as the human ability to formulate queries using an ontology, or the accuracy of responses provided by the system's inferential component. In order to qualitatively evaluate the AFO framework, we need to define a set of requirements from the perspective of music information retrieval workflows. Reviewing common research workflows, the following main requirements for audio feature annotations have been discovered:

- identify an extracted audio feature by linking it to a corresponding term in the Audio Feature Vocabulary
- identify the computational steps involved in the process
- describe the temporal structure and density of output
- associate audio features with the audio signal timeline
- identify the feature extraction software tools used in the extraction process

Sparse point-like and dense signal-like features of an audio file - such as onsets or MFCC - can be linked directly to their respective classes in AFV in the feature extraction process as shown in Listing 3.

The Turtle representation is but one of the possible means of serialisation. AFO can facilitate development of other data formats that are aligned with linked data principles, including binary RDF representations. One of the goals of the development process has been to look for alternative formats that could be used in different contexts. Due to the wide appeal of JSON, the ontology also enables publishing feature data in its linked data version. JSON-LD is an extension to the standard JSON format that provides an entity-centric representation of RDF/OWL semantics and a means to define a linked data context with URI connections to external ontologies and resources [10]. It has the potential to simplify feature representations while maintaining ontological structuring of the data. The format enables establishing links to ontologies where the structure of the data is defined by using the key word "@context". OWL class types are annotated with "@type" and unique identifiers are with "@id". The latter functions as a linking mechanism between nodes when an RDF graph is converted into a JSON tree structure. The JSON-LD representation of audio features has been tested in the context of an adaptive music player.

```
@prefix afv: <http://w3id.org/afo/vocab/1.1#> .
@prefix mo: <http://purl.org/ontology/mo/> .
@prefix tl:
      <http://purl.org/c4dm/timeline.owl#> .
@prefix vamp: <http://purl.org/ontology/vamp/> .

:signal_f6261475 a mo:Signal ;
  mo:time [
    a tl:Interval ;
    tl:onTimeLine :timeline_aec1cb82
  ] .

:timeline_aec1cb82 a tl:Timeline .

:transform_onsets a vamp:Transform ;
  vamp:plugin plugbase:qm-onsetdetector ;
  vamp:output
       plugbase:qm-onsetdetector_output_onsets .

:transform_mfcc a vamp:Transform ;
  vamp:plugin plugbase:qm-mfcc ;
  vamp:output
       plugbase:qm-mfcc_output_coefficients .

:event_1 a afv:Onset ;
  event:time [
    a tl:Instant ;
    tl:onTimeLine :timeline_aec1cb82 ;
    tl:at "PT1.98S"^^xsd:duration ;
  ] ;
  vamp:computed_by :transform_onsets .

:feature_1 a afv:MFCC ;
  mo:time [
    a tl:Interval ;
    tl:onTimeLine :timeline_aec1cb82 ;
  ] ;
  vamp:computed_by :transform_mfcc ;
  afo:value ( -26.9344 0.188319 0.106938 ..) .
```

Listing 3. An abbreviated example of linking onsets and MFCC features to AFV and the Music Ontology

5.3 Case study: adaptive music player

Beyond representing audio feature data in research workflows, there are many other practical applications for the ontology framework. One of the test cases is providing data services for an adaptive music player that uses audio features to enrich user experience and enables novel ways to search or browse large music collections. Feature data of the music tracks available in the player is stored in a CouchDB [13] instance in JSON-LD. The data is used by Semantic Web entities called Dynamic Music Objects (dymos) [22] that control the audio mixing functionality of the player. Dymos make song selections and determine tempo alignment for cross-fading based on features. Listing 4 shows an example of JSON-LD representation of a track used in the system linked to feature annotations.

6. CONCLUSIONS

The Audio Feature Ontology and Vocabulary provide a framework for representing audio features using Semantic Web methods and linked data technologies. It provides terminology to facilitate task and tool specific ontology development and serves as a descriptive framework

[13] http://couchdb.apache.org/

```
{
   "@context": {
      "foaf": "http://xmlns.com/foaf/0.1/",
      "afo": "http://w3id.org/afo/onto/1.1#",
      "afv": "http://w3id.org/afo/vocab/1.1#",
      "mo": "http://purl.org/ontology/mo/",
      "dc": "http://purl.org/dc/elements/1.1/",
      "tl": "http://purl.org/NET/c4dm/timeline.owl#",
      "vamp": "http://purl.org/ontology/vamp/"
   },
   "@type": "mo:Track",
   "dc:title": "Open My Eyes",
   "mo:artist": {
      "@type": "mo:MusicArtist",
      "foaf:name": "The Nazz"
   },
   "mo:available_as": "/home/snd/250062-15.01.wav",
   "mo:encodes": {
      "@type": "mo:Signal",
      "mo:time": {
         "@type": "tl:Interval",
         "tl:duration": "PT163S",
         "tl:timeline": {
            "@type": "tl:Timeline",
            "@id": "98cfa995.."
   }}},
   "afo:features": [
      {
         "@type": "afv:Key",
         "vamp:computed_by": {
            "@type": "vamp:Transform",
            "vamp:plugin_id":
               "vamp:qm-vamp-plugins:qm-keydetector"
         },
         "afo:values": [
            { "tl:at": 1.4 , "rdfs:label": "C# minor",
               "tl:timeline": "98cfa995.."
            },
            { "tl:at": 5.9 , "rdfs:label": "D minor",
               "tl:timeline": "98cfa995.."
            }
         ]
      }]}
```

Listing 4. JSON-LD representation of an audio feature linked with track metadata

for audio feature extraction. The updates to the original ontology for audio features strive to simplify feature representations and make them more flexible while maintaining ontological structuring and linking capabilities. JSON-LD has been shown to function as a linked data format that enables converting RDF graph structures to key-value representation. This could also apply for other similar data formats and NoSQL database systems. The ontology engineering process has produced example ontologies for existing tools including MIR Toolbox, Essentia, Marsyas and others available from the ontology Website http://w3id.org/afo/onto/1.1#.

7. ACKNOWLEDGEMENTS

This work was part funded by the FAST IMPACt EPSRC Grant EP/L019981/1 and the European Commission H2020 research and innovation grant AudioCommons (688382). Sandler acknowledges the support of the Royal Society as a recipient of a Wolfson Research Merit Award.

8. REFERENCES

[1] X. Amatriain, P. Arumi, and D. Garcia. Clam: A framework for efficient and rapid development of cross-

platform audio applications. In *proc. 14th ACM International Conference on Multimedia*, pages 951–954, New York, USA, 2006.

[2] S. Bechhofer, S. Dixon, G. Fazekas, T. Wilmering, and K. Page. Computational analysis of the live music archive. In *proc. 15th International Conference on Music Information Retrieval (ISMIR)*, 2014.

[3] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia: an audio analysis library for music information retrieval. In *proc. 14th International Society for Music Information Retrieval Conference (ISMIR'13)*, pages 493–498, Curitiba, Brazil, 2013.

[4] G. Fazekas, Y. Raimond, K. Jakobson, and M. Sandler. An overview of semantic web activities in the OMRAS2 project. *Journal of New Music Research (JNMR)*, 39(4), 2010.

[5] G. Fazekas and M. Sandler. The studio ontology framework. In *proc. 12th International Society for Music Information Retrieval (ISMIR'11) conference, Miami, USA, 24-28 Oct.*, pages 24–28, 2011.

[6] R. Fiebrink, G. Wang, and Perry R. Cook. Support for MIR prototyping and real-time applications in the chuck programming language. In *proc. 9th International Conference on Music Information Retrieval, Philadelphia, PA, USA, September 14-18*, 2008.

[7] B. Fields, K. Page, D. De Roure, and T. Crawford. The segment ontology: Bridging music-generic and domain-specific. In *proc. IEEE International Conference on Multimedia and Expo, Barcelona, Spain, 11-15 July*, 2011.

[8] E. J. Humphrey, J. Salamon, O. Nieto, J. Forsyth, R. Bittner, and J. P. Bello. JAMS: A JSON annotated music specification for reproducible MIR research. In *15th Int. Soc. for Music Info. Retrieval Conf.*, pages 591–596, Taipei, Taiwan, Oct. 2014.

[9] K. Jacobson, Y. Raimond, and M. Sandler. An ecosystem for transparent music similarity in an open world. In *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe, Japan, October 26-30*, 2009.

[10] M. Lanthaler and C. Gütl. On using JSON-LD to create evolvable restful services. In *proc. 3rd International Workshop on RESTful Design at WWW'12*, 2012.

[11] B. Mathieu, B. Essid, T. Fillon, J. Prado, and G. Richard. YAAFE, an easy to use and efficient audio feature extraction software. In *proc. 11th International Conference on Music Information Retrieval (ISMIR), Utrecht, Netherlands, August 9-13*, 2010.

[12] R. Mayer and A. Rauber. Towards time-resilient mir processes. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR), Porto, Portugal, October 8-12*, 2012.

[13] C. McKay and I. Fujinaga. Improving automatic music classification performance by extracting features from different types of data. In *proc. of the International Conference on Multimedia Information Retrieval*, pages 257–266, New York, USA, 2010.

[14] D. Mitrovic, M. Zeppelzauer, and C. Breiteneder. Features for content-based audio retrieval. *Advances in Computers*, 78:71–150, 2010.

[15] D Moffat, D Ronan, and J. D. Reiss. An evaluation of audio feature extraction toolboxes. In *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, 2015.

[16] Meinard Müller and Sebastian Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *proc. 12th International Society for Music Information Retrieval (ISMIR'11) conference, Miami, USA*, 2011.

[17] K. Page, B. Fields, B. Nagel, G. O'Neill, D. De Roure, and T. Crawford. Semantics for music analysis through linked data: How country is my country? In *Proceedings of the 6th International Conference on e-Science, Brisbane, Australia, 7-10 Dec.*, 2010.

[18] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. rep., IRCAM, 2004.

[19] A. Porter, D. Bogdanov, R. Kaye, R. Tsukanov, and X. Serra. Acousticbrainz: a community platform for gathering music information obtained from audio. In *proc. 16th International Society for Music Information Retrieval (ISMIR) Conference*, 2015.

[20] Y Raimond, S. Abdallah, M. Sandler, and F. Giasson. The music ontology. In *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007, Vienna, Austria, September 23-27*, 2007.

[21] Markus Schedl. The CoMIRVA Toolkit for Visualizing Music-Related Data. Technical report, Dept. of Computational Perception, Johannes Kepler Univ., 2006.

[22] Florian Thalmann, Alfonso Perez Carillo, György Fazekas, Geraint A. Wiggins, and Mark Sandler. The mobile audio ontology: Experiencing dynamic music objects on mobile devices. In *proc. 10th IEEE International Conference on Semantic Computing, Laguna Hills, CA, USA*, 2016.

[23] M. Vacura, V. Svátek, C. Saathoff, T. Ranz, and R. Troncy. Describing low-level image features using the COMM ontology. In *proc. 15th International Conference on Image Processing (ICIP), San Diego, California, USA*, pages 49–52, 2008.

[24] K. West, A. Kumar, A. Shirk, G. Zhu, J. S. Downie, A. F. Ehmann, and M. Bay. The networked environment for music analysis (NEMA). In *proc. 6th World Congress on Services, Miami, USA, July 5-10*, 2010.