Deliverable D2.8

Ontology Evaluation Report

Grant agreement nr	688382
Project full title	Audio Commons: An Ecosystem for Creative Reuse of Audio Content
Project acronym	AudioCommons
Project duration	36 Months (February 2016 - January 2019)
Work package	WP2
Due date	31 January 2019 (M36)
Submission date	31 January 2019 (M36)
Report availability	Public (X), Confidential ( )
Deliverable type	Report (X), Demonstrator ( ), Other ( )
Task leader	QMUL
Authors	Miguel Ceriani, Saša Rudan, György Fazekas.
Document status	Draft ( ), Final (X)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Table of contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Executive Summary

This deliverable reports on the evaluation of the Audio Commons Ontology. It shows its consistency and capability to fulfill the needs of the Audio Commons Ecosystem, both from a theoretical standpoint and practical analysis of its use.

The ontology itself is described in D2.3 and published online[1]. It has been designed as common data model for audio content in the Audio Commons Ecosystem. Is now used in its intended role in multiple tools developed in the project.

The evaluation is carried out along the following three main approaches:

- theoretical evaluation based on logical consistency, guidelines, and assessing the capability of answering a set of competency questions;
- practical evaluation based on its use in practical services and tools;
- evaluation based on future and unanticipated use case scenarios.

The evaluation results in this deliverable are presented through examples demonstrating how the ontology fulfils requirements. The reported evaluation shows the suitability of the designed ontology for its role in the project and potentially beyond it.

[1] https://w3id.org/ac-ontology/aco

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Background

The project proposes a common data model to be used to describe metadata of audio content in the Audio Commons Ecosystem. This common data model is described as an ontology, the Audio Commons Ontology, based on the survey and analysis in D2.1, drafted in D2.2, and finalised in D2.3. The ontology is developed in a public repository[2] and available for online consumption[3] (both by humans, as HTML documentation, and by machines, as OWL).

This report describes the evaluation of the Audio Commons Ontology. It relates to other deliverables reporting on tools that use the ontology (D2.7 for the mediator and D4.12 for the tool for automatic semantic description of music samples) and to D6.12 reporting on the broader evaluation of the whole Audio Content Ecosystem.

[2] https://github.com/AudioCommons/ac-ontology
[3] https://w3id.org/ac-ontology/aco

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

1 Introduction

Evaluating an ontology is a non-trivial, continuous process that does not end with its publication. Nevertheless, it is useful to evaluate it thoroughly at a point in time in which it has been published and is used by a set of tools.

Following methods considered common practice, the ontology is evaluated both formally on its own and in real usage scenarios.

The focus is on the role and usage in the Audio Commons project, but the scope of the ontology is potentially broader so the evaluation is performed also from a broader perspective.

In Section 2 the ontology is formally evaluated by analysing internal consistency, conformance to established guidelines, and ability to answer competency questions defined from scenarios in D2.1. Section 3 describes the in-use evaluation based on the actual usage in the Audio Commons Ecosystem. In Section 4 other potential usage scenarios are evaluated. Section 5 draws conclusions and Section 6 lists references to cited works.

1.1 Main objectives and goals

The objective of this deliverable is to provide an evaluation of the audio commons ontology which includes multiple aspects and perspectives.

1.2 Terminology

AudioCommons: reference to the EC H2020 funded project AudioCommons, with grant agreement nr 688382.

Audio Commons Initiative: reference to the AudioCommons project core ideas beyond the lifetime and specific scope of the funded project. The term “Audio Commons Initiative” is used to imply i) our will to continue supporting the Audio Commons Ecosystem and its ideas after the lifetime of the funded project, and ii) our will to engage new stakeholders which are not officially part of the project consortium.

Audio Commons: generic reference to the Audio Commons core ideas, without distinguishing between the concept of the initiative and the actual funded project.

Audio Commons Ecosystem (ACE): set of interconnected tools, technologies, content, users and other actors involved in publishing and consuming Audio Commons content.

Audio Commons content (ACC): audio content released under Creative Commons licenses and enhanced with meaningful contextual information (e.g., annotations, license information) that enables its publication in the ACE.

Content creator: individual users, industries or other actors that create audio content and publish in the ACE through content providers.

Content provider: services that expose content created by content creators to the ACE.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 688382

Content user: individual users, industries or other actors that use the content exposed by content providers and created by content creators in their creative workflows.

Tool developer: individual users, industries or other actors that develop tools for consuming (and also potentially publishing) Audio Commons content.

Embeddable tools: tools for consuming Audio Commons content that can be embedded in existing production workflows of creative industries.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

2 Formal Evaluation

We carried out an assessment of the ontology and the mediator by using formal methods as well as checking their fitness for our domain and purposes.

2.1 Consistency and Heuristics

The Audio Commons ontology has been checked for correctness, logical consistency, and alignment with established ontology design guidelines.

The correctness of the ontology and its serializations has been checked first by loading it in the widely used ontology editor Protégé and second through the VOWL copy[4] of the online validation service originally developed by the University of Manchester [Horridge2009]. The logical consistency has been checked by running two reasoners, HermiT (version 1.3.8.413) [Shearer2008] and FaCT++ (version 1.6.5) [Tsarkov2006]. No inconsistencies have been found.

To validate the ontology with respect to existing good practices, we used the OntOlogy Pitfall Scanner! (OOPS!) online service [Poveda2014]. This service, based on the existing relevant literature, checks for common pitfalls in ontology design. No pitfalls have been detected in the Audio Commons ontology.

2.2 Competency Questions

In the deliverable D2.1, after analysing the results of the user survey, a set of sample scenarios were identified, described as user stories:

1. As a cafè owner I would like to search for whole songs, which are free of any licensing fees. I would like to search via a browser and search with free text search. As an example, I would like to search via a browser for “Slow funk track without vocals”. Once I found something I like, I would like to find tracks that play well together.
2. As an audio producer, I would like to have access to a 10,000+ set of high-quality audio loops from within my digital audio workstation (DAW). I want to search by instrument type, genre, key, tempo. I only want high-quality files.
3. As a game sound designer, I would like to have access to an unlimited set of high-quality audio files from within my DAW. I want to search by effect type, mood, and perceptual features like “warm”, “bright”, etc.

Based on the scenarios and the scope of the ontology, the following competency questions have been derived; they are meant to be asked with respect to a set of source repositories of audio content:

1. Which are the songs that are slow (tempo) funk (genre) tracks without vocals (instrumentation)?
2. What other tracks “play well” together with a given song in a playlist (e.g., are in the same category according to some classification)?
3. Retrieve high-quality (sample rate, bits per sample) audio loops (type of audio content) for a given instrument type, genre, key, tempo.

[4] http://visualdataweb.de/validator/validate

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 688382

4. Retrieve high-quality (sample rate, bits per sample) sound effects (type of audio content) for a given effect type, mood, and a set of perceptual features (e.g., warm and bright).

The questions can be formalised as queries from the data sources (the audio content repositories), for example using SPARQL (the standard query language for RDF). For simplicity and conciseness here the formalisation is described at a higher level, using just bits of SPARQL syntax for the graph patterns.

Represented in the vocabulary defined by the AC ontology, all the competency questions consist in queries that get a set of individuals of type `ac:AudioClip`, i.e. values of `?audioClip` where the triple `?audioClip rdf:type ac:AudioClip` exists.

The set returned is determined by some filters that are applied on all the individuals available from the data source. Most filters can be represented as belonging to a certain category of a classification (mood, genre, instrumentation, ...), hence can be formalised as the existence of a triple `?audioClip ac:audioCategory <category1>`.

Some filters (sample rate, bits per sample, ...) require to assess a numeric value. To require that the sample rate is certainly higher than 40KHz, both triples `?audioClip ac:publicationOf ?signal` and `?signal ac:sampleRate ?sampleRate` have to exist and must satisfy `?sampleRate > 40E3`.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

3 In-Use Evaluation

The Audio Commons ontology has been evaluate "in use" by analysing if it is fit for practical use in the project. We document here the aspects and contexts in which the ontology has been used and evaluated.

3.1 Utilisation as output data model for the AC API

In deliverable D2.4 the Audio Commons API has been described. That version of the API, based on the results of the survey in D2.1 but antecedent to the formalisation and implementation of the ontology in D2.3, started being used by multiple clients developed in the project.

As the main aim of the ontology was to be used as common data model of the Audio Commons Ecosystem, naturally the AC API is expected to relate to the ontology. The ontology is hence evaluated by using it as output data model of AC API. A part from representing the same information, the ontology use is expected to not introduce much more complexity in the output and enable novel usage scenarios.

We here analyse the capability of the ontology of representing the AC API output with a limited increase of complexity.

The Audio Commons Mediator is the tool that acts as middleware, integrating existing services (Freesound, Jamendo, and Europeana) and offering the standard AC API to its clients. The main endpoint of the AC API is the one offering search capability on the set of audio clips of the integrated services. A typical response of the previous version of the API (accessible at https://m.audiocommons.org/) is the following (the JSON result has been simplified for readability, while preserving the structure):

```
{ "Jamendo": {"num_results": 0, "results": [] },
  "Freesound": {
    "num_results": 1,
    "results": [
      { "id": "Freesound:385285", "name": "01 - Cars across_city.wav",
        "description": "Recorded in Moscow. Cars across on street.",
        "duration": 127.321, "author": "semenov_nick",
        "channels": 2, "bitdepth": 32, "samplerate": 48000.0,
        "bitrate": 3001, "filesize": 48915012, "format": "wav",
        "collection_url": "https://freesound.org/apiv2/packs/21409/",
        "license_deed_url": "http://creativecommons.org/publicdomain/zero/1.0/",
        "preview_url": "https://freesound.org/.../385285_3596390-hq.ogg",
        "tags": ["city", "cars"] }
    ]
  },
  "Europeana": {"num_results": 0, "results": [] } }
```

Listing 1. Example response of the first version of the API

We developed a new version of the mediator, called the Audio Commons Semantic Mediator (accessible at https://m2.audiocommons.org/ and documented in Deliverable D2.7), in which the AC ontology and semantic technologies are used in all the process of integration. The output is semantically interpretable, uses the AC ontology, and adds limited complexity in respect the original

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

output model. This is an example of the output (again, the result has been simplified for readability, while preserving the structure):


```
{
  "@id": "acActions:fbf017d1-faed-4948-b58f-25c154157ea5",
  "@type": "schema:SearchAction",
  "actionStatus": "schema:CompletedActionStatus",
  "errors": [],
  "query": "cars",
  "results": [
    {
      "from": {
        "@type": "schema:SearchAction",
        "actionStatus": "schema:CompletedActionStatus",
        "object": "http://freesound.org",
      },
      "@type": "ac:AudioCollection",
      "members": [
        {
          "@type": "ac:AudioCollectionNode",
          "content": {
            "@id": "freesound-sounds:385285",
            "@type": "ac:AudioClip",
            "license": "http://creativecommons.org/publicdomain/zero/1.0/",
            "description": "Recorded in Moscow. Cars across on street.",
            "title": "01 - Cars across_city.wav",
            "audioCategories": [
              "freesound-tags:city",
              "freesound-tags:cars"
            ],
            "author": "freesound-users:semenov_nick",
            "availableAs": [
              {
                "@type": "ac:AudioFile",
                "audioChannelNumber": 2,
                "sampleRate": 48000,
                "sampleSize": 32,
                "bitRate": 192000,
                "hasAudioEncodingFormat": "ogg:",
                "locator": "http://freesound.org/.../385285_3596390-hq.ogg"
              },
              {
                "@type": "ac:AudioFile",
                "audioChannelNumber": 2,
                "sampleRate": 48000,
                "sampleSize": 32,
                "bitRate": 64000,
                "hasAudioEncodingFormat": "mp3:",
                "locator": "http://freesound.org/.../385285_3596390-lq.mp3"
              }
            ],
            "duration": 127321,
            "images": [
              {
                "@type": "ebu:Picture",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
                "frameWidth": 900,
                "frameHeight": 201,
                "locator": "http://freesound.org/.../385285_3596390_wave_L.png",
                "prefLabel": "Waveform"
              },
              {
                "@type": "ebu:Picture",
                "frameWidth": 120,
                "frameHeight": 71,
                "locator": "http://freesound.org/.../385285_3596390_wave_M.png",
                "prefLabel": "Waveform"
              }
            ],
            "originalFile": {
              "@type": "ac:AudioFile",
              "audioChannelNumber": 2,
              "sampleRate": 48000,
              "sampleSize": 32
            }
          },
          "index": 5
        }
      ]
    }
  ]
}
```


Listing 2. Example response of the new version of the API

This output can interpreted semantically through an external JSON-LD context, served on the web from the same server. The JSON-LD context maps the AC API output model to the AC ontology and other used ontologies/vocabularies.

The new AC API is available online. It has been released on the 10th and 11th of November 2018, at the first Abbey Road Hackathon that took place in Studio One of Abbey Road, challenging over a hundred pre-selected participants in creating the next technologies for music making.

During the hackathon, three groups of participants decided to experiment with the possibilities offered by the Audio Commons Ecosystem, especially the search endpoint accessible from the Mediator v2. We gathered positive feedback on the concept and implementation of the service.

For details refer to section 6.3 of the deliverable D6.12.

3.2 Ontology Utilisation for the Integration of Content Providers

The AC Semantic Mediator, apart from using the ontology in its output, uses the ontology for the process of integration of content providers (e.g. Jamendo, Freesound).

The content providers offer heterogeneous web APIs which accept different parameters and reply with differently structured outputs. The semantic mediator maps the input and the output in a

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

declarative way, using the SPARQL-Generate language: an extension to SPARQL, the standard query language for linked data, that provides ways to map JSON structures to linked data.

The AC ontology, along other used ontologies/vocabularies, fulfill effectively the role of the common output model for integrating these web APIs. These queries can be developed by providers that want to integrate with the ACE, by the maintainers of the mediator, or by third parties.

Details of the mechanism are described in Deliverable D2.7. The figure below shows schematically the way the mediator integrates the content providers and exposes its API to the clients.

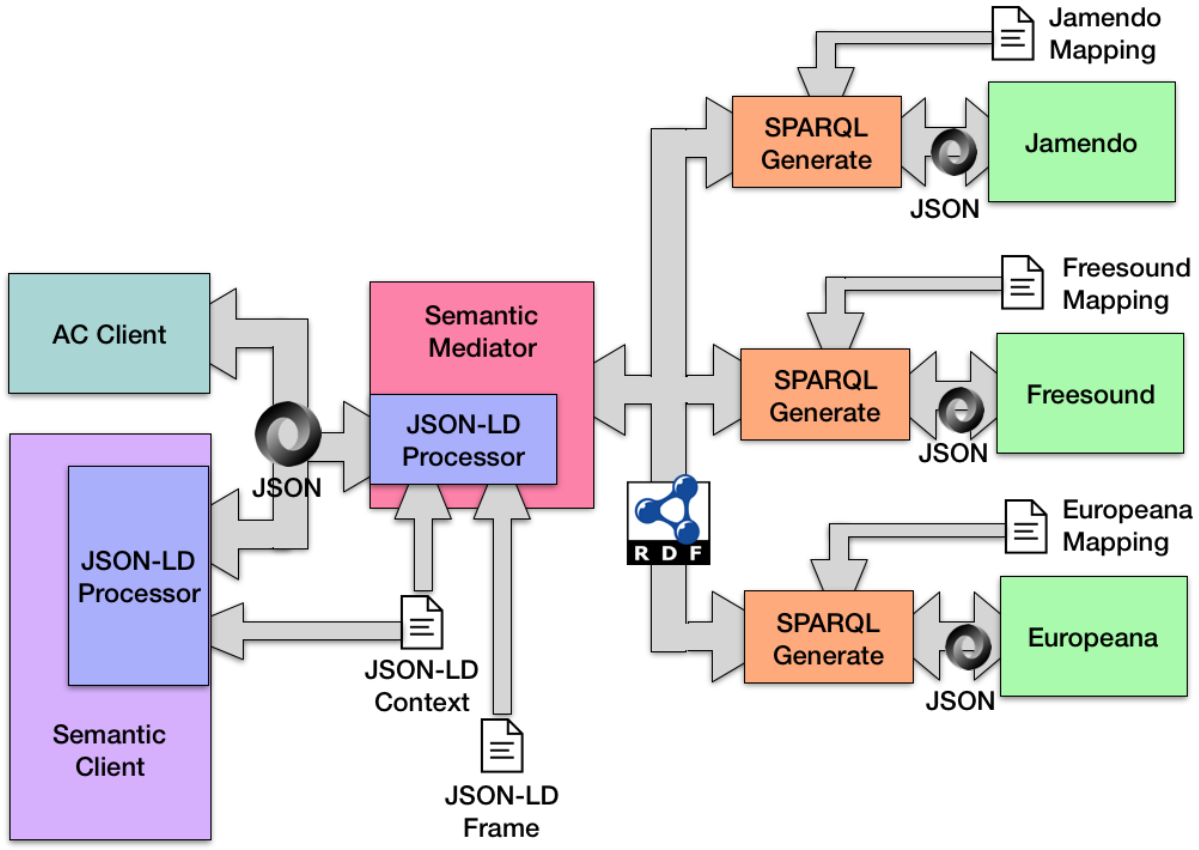

Fig. 1. Data flow in the semantic mediator

The following SPARQL-Generate query gives an example of what a service mapping look like. This the mapping of the freesound API to the search action as defined by the AC API (the query is a bit simplified for presentation purposes).

```
[prefix declarations]

# Adapter from Freesound search service to Audio Commons Ecosystem
# (http://audiocommons.org/).
#
# Input parameters:
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
#    - $token (required): freesound API key
#    - $pattern (required): text used for search
#    - $limit: max number of results (default: 15)
#    - $page: results page number (default: 1)


GENERATE {

  GENERATE {

    # Generate Freesound description for provenance
    <http://freesound.org>
        rdf:type foaf:Organization ; # prov:Agent
        foaf:name "Freesound" .

    # Generate search action description
    ?searchAction
        a schema:SearchAction ;
        schema:object <http://freesound.org> ;
        schema:query $pattern ;
        schema:startTime $startTime ;
        schema:endTime ?endTime ;
        schema:actionStatus ?actionStatus ;
        schema:result ?audioCollection ;
        schema:error ?error .

    # If service call to Freesound successful, generate audio collection resource
    ?audioCollection
        rdf:type ac:AudioCollection ;
        ac:nodeCount ?nodeCount .

    GENERATE {
      # Generate node for the audio collection
       ?audioCollection ac:memberNode ?audioCollectionNode .
      ?audioCollectionNode
          a ac:AudioCollectionNode ;
          ac:nodeIndex ?index ;
          ac:nodeContent ?audioClip .

      # Generate audio clip as content of the audio collection node
      ?audioClip
          a ac:AudioClip; #, prov:Entity ;
          dc:title ?title ;
          dc:description ?description ;
          ac:author ?author ;
          ac:duration ?duration ;
          cc:license ?license ;
          ac:availableAs _:audioFileHqOgg, _:audioFileLqMp3 .

      _:audioFileHqOgg
          a ac:AudioFile; #, schema:AudioObject ;
          ebu:locator ?audioFileHqOggUrl ;
          ebu:bitRate 192000; # 192 kbps
          ebu:hasAudioEncodingFormat encoding_ogg: ;
          ebu:audioChannelNumber ?signalChannels ;
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
        ebu:sampleSize ?signalBitsPerSample ;
        ebu:sampleRate ?signalSampleRate .

    _:audioFileLqMp3
        a ac:AudioFile; #, schema:AudioObject ;
        ebu:locator ?audioFileLqMp3Url ;
        ebu:bitRate 64000; # 64 kbps
        ebu:hasAudioEncodingFormat encoding_mp3: ;
        ebu:audioChannelNumber ?signalChannels ;
        ebu:sampleSize ?signalBitsPerSample ;
        ebu:sampleRate ?signalSampleRate .

    [generation of further related entities]

    # Bind a ?category for each used tag
    GENERATE {?audioClip ac:audioCategory ?category}
    ITERATOR iter:JSONPath(?res, "tags[*]") AS ?tag
    WHERE {
      BIND(STR(?tag) AS ?tagStr)
      BIND(
          IRI(CONCAT(
            STR(freesound-tags:),
            SUBSTR(?tagStr, 2, STRLEN(?tagStr) - 2)))
          AS ?category)
    } .
  }
  # Iterate over result set
  ITERATOR iter:JSONElement(?source,"results[*]") AS ?resIterator
  WHERE {
    # Bind blank node to ?audioCollectionNode
    BIND(BNODE() AS ?audioCollectionNode)
    BIND(fn:JSONPath(?resIterator, "element") AS ?res)
    BIND(fn:JSONPath(?resIterator, "position") AS ?indexFromZero)
    BIND(?indexFromZero + 1 AS ?index)

    # From Freesound ID, mint URI for ?audioClip
    BIND(
        IRI(CONCAT(
          STR(freesound-sounds:),
          STR(fn:JSONPath(?res, "id"))))
        AS ?audioClip)

    # Bind relevant metadata
    BIND(IRI(fn:JSONPath(?res, "url")) AS ?homepage)
    BIND(IRI(fn:JSONPath(?res, "download")) AS ?originalAudioFileUrl)
    BIND(IRI(fn:JSONPath(?res, "bitrate")) AS ?originalAudioFileBitrate)
    BIND(COALESCE(fn:JSONPath(?res, "type"), "unknown")
        AS ?originalAudioFileEncodingType)

    OPTIONAL {
      VALUES (?originalAudioFileEncodingType ?originalAudioFileEncoding) {
        ("wav" encoding_wav:)
        ("mp3" encoding_mp3:)
        ("flac" encoding_flac:)
      }
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
      }

      BIND(fn:JSONPath(?res, "name") AS ?title)
      BIND(fn:JSONPath(?res, "description") AS ?description)

      BIND(1000 * fn:JSONPath(?res, "duration") AS ?duration)
      BIND(IRI(fn:JSONPath(?res, "license")) AS ?license)
      BIND(fn:JSONPath(?res, "tags") AS ?tagSet)

      BIND(fn:JSONPath(?res, "channels") AS ?signalChannels)
      BIND(fn:JSONPath(?res, "bitdepth") AS ?signalBitsPerSample)
      BIND(fn:JSONPath(?res, "samplerate") AS ?signalSampleRate)

      [bind further metadata]

    } .
  }

  # Call service and bind the response to ?source
  SOURCE ?serviceCall AS ?source
  WHERE {
    # Bind blank node to ?searchAction
    # If successful :
    #   Bind ?resultSet to JSON result set
    #   Bind ?actionStatus to schema:CompletedActionStatus
    #   Bind blank node to ?audioCollection
    # Else :
    #   Bind blank node to ?error
    BIND(BNODE() AS ?searchAction)
    BIND(fn:JSONPath(?source, "results") AS ?resultSet)
    BIND(
       IF(BOUND(?resultSet),
         schema:CompletedActionStatus,
         schema:FailedActionStatus)
       AS ?actionStatus)
    OPTIONAL {
       BIND(BNODE() AS ?audioCollection).
      FILTER(BOUND(?resultSet))
    }
    OPTIONAL {
       BIND(BNODE() AS ?error)
      FILTER(!BOUND(?resultSet))
    }
  } .
}
WHERE {
  # Generate URL for service call, based on input parameters
  BIND(IRI(CONCAT(
      STR(freesound-api:), "search/text/",
      "?token=", ENCODE_FOR_URI($token),
      "&fields=",
          "id,name,description,images,pack,username,created,duration,license,",
          "tags,channels,bitdepth,samplerate,previews",
      "&query=", ENCODE_FOR_URI($pattern),
      IF(BOUND($limit),CONCAT("&page_size=", ENCODE_FOR_URI(STR($limit))),""),
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
      IF(BOUND($page),CONCAT("&page=", ENCODE_FOR_URI(STR($page))),"")
  )) AS ?serviceCall)
}
```

Listing 3. SPARQL-Generate query that maps Freesound service to the common model

2.2.3 The Ontology as an Output Model for an Automatic Annotation Tool

As part of the project, a tool for automatic semantic description of music samples has been released. The tool offers both a simple JSON output and a JSON-LD output using the AC ontology, along with the Audio Features Ontology and other relevant vocabularies.

The tool is described in detail in Deliverable D4.12. Here we compare the two output formats, showing that also in this case that the ontology is capable of representing effectively the output data.

JSON output format:


```
{
    "duration": 9.241541862487793,
    "lossless": 1.0,
    "codec": "pcm_s16le",
    "bitrate": 705600.0,
    "samplerate": 44100.0,
    "channels": 1.0,
    "audio_md5": "2722ac23a142ce727e0642b0a63c7347",
    "loudness": -28.64586639404297,
    "dynamic_range": 3.432065963745117,
    "temporal_centroid": 0.5782503485679626,
    "log_attack_time": 0.6950863599777222,
    "filesize": 815294,
    "single_event": false,
    "tonality": "G# major",
    "tonality_confidence": 0.5119080543518066,
    "loop": false,
    "tempo": 84,
    "tempo_confidence": 0.42026047706604003,
    "note_midi": 74,
    "note_name": "D5",
    "note_frequency": 608.390625,
    "note_confidence": 0.0,
    "brightness": 60.313207479409286,
    "depth": 16.728879931862544,
    "hardness": 82.90738501480826,
    "roughness": 6.646583836789146
}
```


Listing 4. JSON output of the automatic annotation tool

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

These data represent mainly audio features extracted by the tool from the given audio file. To represent the audio features, the Audio Features Ontology (AFO)[5] is used.

The used terms are took form Audio Commons Ontology, Audio Features Ontology (for the audio features and their confidence), EBU Core Ontology[6] (for the metadata of the audio resource), and NEPOMUK File Ontology[7] (just to describe the compression type).

JSON-LD output format:


```
{
    "@context": {
        "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
        "ac": "https://w3id.org/ac-ontology/aco#",
        "afo": "https://w3id.org/afo/onto/1.1#",
        "afv": "https://w3id.org/afo/vocab/1.1#",
        "ebucore": "http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#",
        "nfo": "http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#"
    },
    "@type": "ac:AudioFile",
    "ebucore:bitrate": 705600.0,
    "ebucore:filesize": 529278,
    "ebucore:hasCodec": {
        "@type": "ebucore:AudioCodec",
        "ebucore:codecId": "pcm_s16le"
    },
    "nfo:compressionType": "nfo:losslessCompressionType",
    "ac:audioMd5": "8da67c9c2acbd13998c9002aa0f60466",
    "ac:availableItemOf": {
        "@type": "ac:AudioClip"
    },
    "ac:signalAudioFeature": [
        {
            "@type": "afv:Loop",
            "afo:value": true
        },
        {
            "@type": "afv:Tempo",
            "afo:confidence": 1.0,
            "afo:value": 120
        },
        {
            "@type": "afv:Key",
            "afo:confidence": 0.2868785858154297,
            "afo:value": "G# minor"
        },
        {
            "@type": "afv:TemporalCentroid",
            "afo:value": 0.5078766345977783
        },
        {
            "@type": "afv:MIDINote",
```


[5] https://w3id.org/afo/onto/
[6] https://www.ebu.ch/metadata/ontologies/ebucore/
[7] http://www.semanticdesktop.org/ontologies/2007/03/22/nfo

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

```
            "afo:confidence": 0.0,
            "afo:value": 74
        },
        {
            "@type": "afv:Pitch",
            "afo:confidence": 0.0,
            "afo:value": 592.681884765625
        },
        {
            "@type": "afv:Loudness",
            "afo:value": -28.207069396972656
        },
        {
            "@type": "afv:Note",
            "afo:confidence": 0.0,
            "afo:value": "D5"
        },
        {
            "@type": "afv:LogAttackTime",
            "afo:value": 0.30115795135498047
        }
    ],
    "ac:signalChannels": 1,
    "ac:signalDuration": 6.0,
    "ac:singalSamplerate": 44100.0
}
```


Listing 5. JSON output of the automatic annotation tool

This tool can be used on its own, but being compatible with the common AC data model facilitates its integration in a workflow including usage of multiple services.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

4 Potential Future Use Case Scenarios

The purpose of an ontology is not only to serve as data model for a specific application or set of applications. The purpose of an ontology is to be used also in unpredicted ways, as the data is accessed, integrated, manipulated in ways not necessarily planned in advance.

While such a serendipitous usage is not evaluable in general, we envision some specific interesting scenarios that the ontology and the mediator may support even if they had not yet been practically implemented.

4.1 Exploiting the Graph Structure of the API Output

When the output of the AC API is interpreted as semantic data, the graph structure of linked data is exposed. The data is composed of resources (e.g., audio clips, audio collections, people, audio categories) connected by the so called object properties (e.g., authorship, membership). The so called data properties are used to describe the resources (e.g., title, bit rate).

A client may use this graph model to query, browse, or organise the results in ways that would be harder to achieve with tree-structured data (simple JSON).

As a simple example, let us consider just the relationship between audio clips and audio categories. Having access to the graph it is possible (with linked data libraries and tools) to query it for all the categories involved or even just a specific type of categories (e.g., musical instruments). The audio clips can then be for example visualised clustered by these categories, rather than simply showing them all together by the order they are returned.

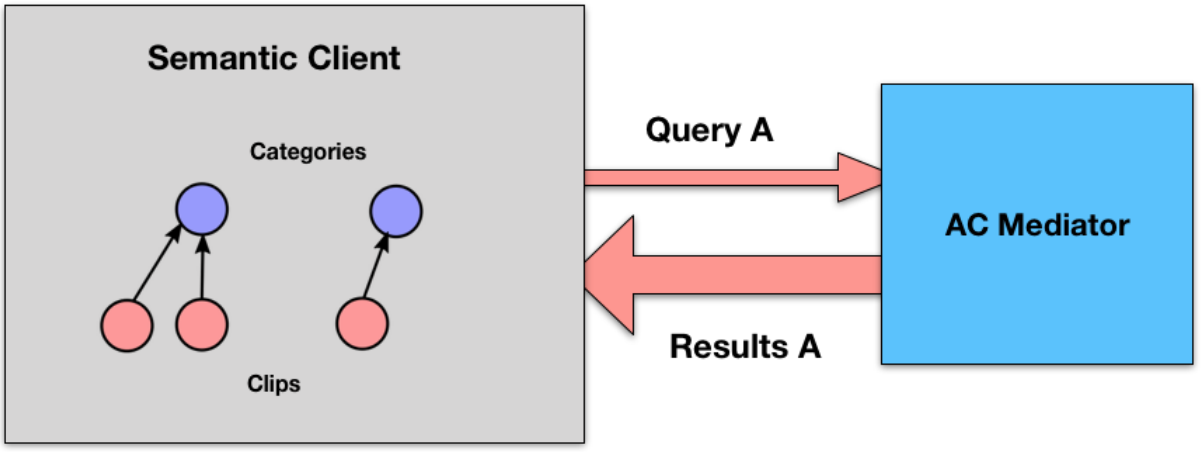

Fig. 2. Graph model of the results from the execution of a query through the mediator

The resources in the linked data model have universal identifiers (URIs) that are meant to be stable. That means that if a client executes multiple queries the graph results can be trivially merged and then analysed together.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Extending the previous example, the same client could allow the user to execute a sequence of queries using different keywords and incrementally aggregate the results. These aggregated results can still be easily organised in categories as before as the data model did not change.

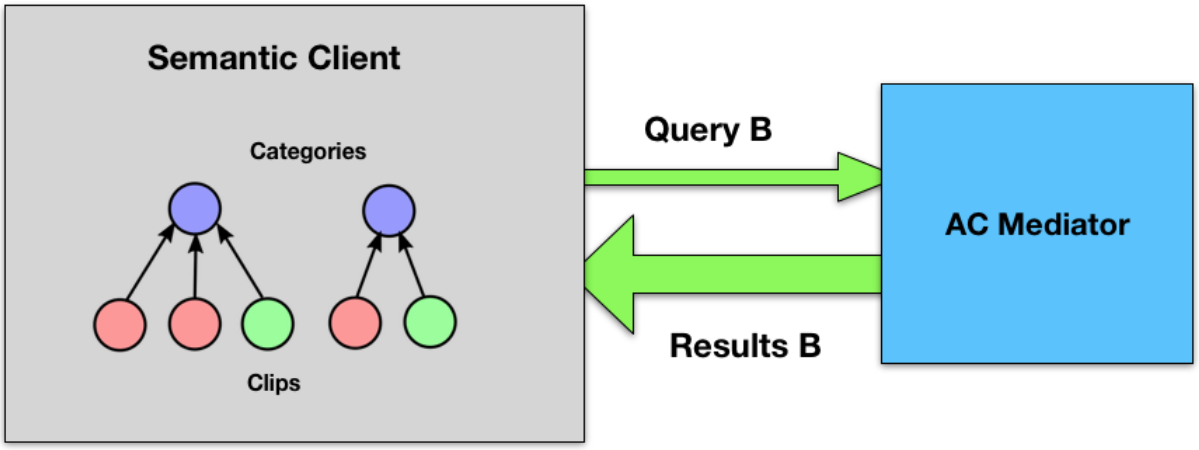

Fig. 3. Graph model merging the results of two queries

4.2 Semantic Integration with External Sources

As the linked data model is designed for data integration on web scale, it enables scenarios where the data from the API is integrated with data from other sources.

Extending our previous example and considering that we are interested in organising the audio clips by musical instrument, we may find a taxonomy of musical instruments on an external data source.

This external taxonomy could be already represented as linked data or adapted as linked data with a number of existing methods. We thus assume that we can access the taxonomy as linked data.

The resources used in the taxonomy to represent could be or not the ones used by the API to represent the corresponding categories. If they are different they can be mapped, even by a third party, by established linked data techniques. We thus for simplicity assume that we end up with a taxonomy using the same resources as the AC API.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

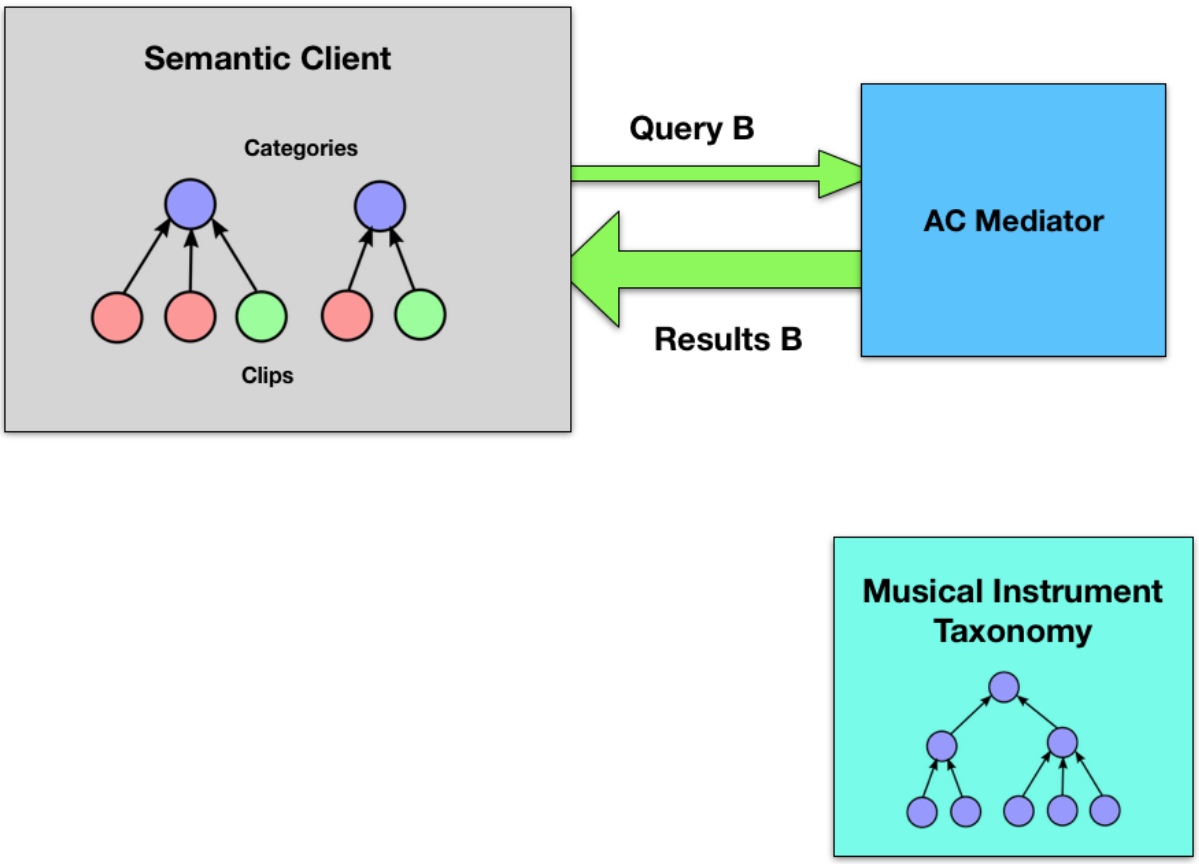

Fig. 4. Scenario including an external data source

We can now merge the taxonomy in our graph on the client, to include the information on the classification of the musical instruments.

Now we can organise the audio clips by broad categories of musical instruments, even if there were used in the data from the provider. As an example, we may group together the clips containing brass instruments, even if they were labelled on the provider just for having specific instruments of the category (e.g., trumpet, tuba).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

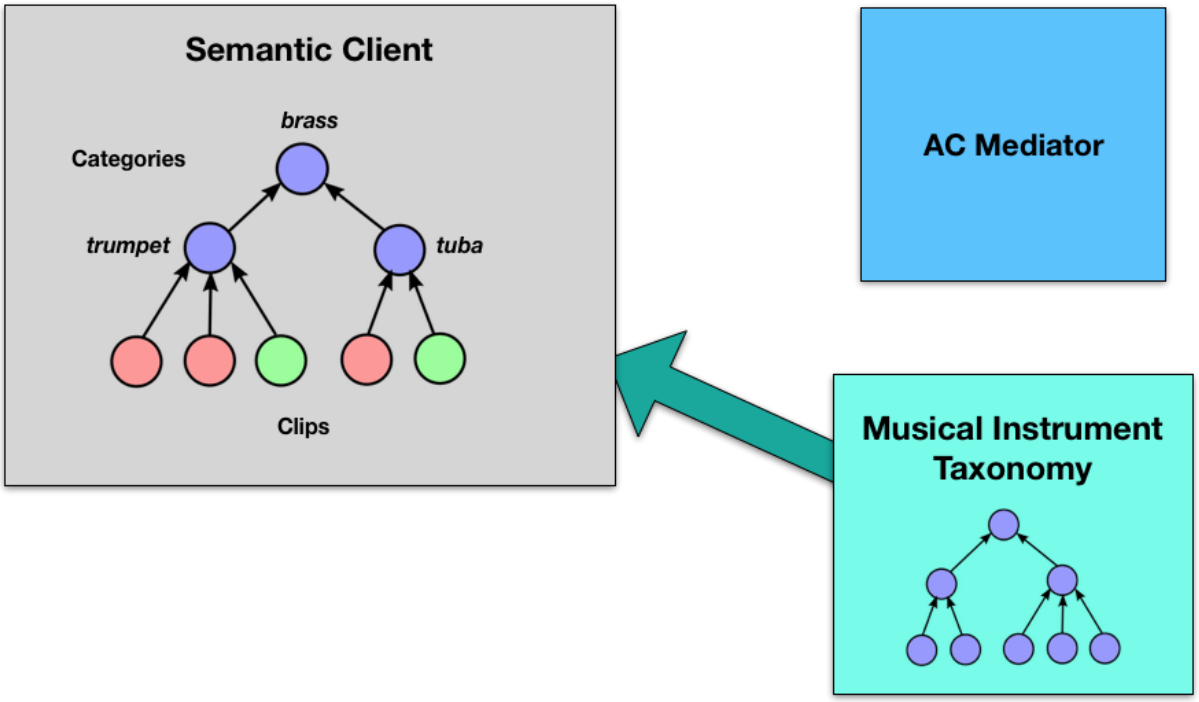

Fig. 5. Graph model merging the results of two queries and data from an external source

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

5 Conclusion

We reported on multiple activities of evaluation of the Audio Commons Ontology.

The evaluation shows that the ontology is consistent, respect existing guidelines, and most importantly that has been successfully used as data model for multiple aspects of the ACE. We offer also a more speculative evaluation based on scenarios that were not directly studied in the project but for which the ontology provides support.

The ontology is publicly available[8] and is used in the last version of the AC API and other AC tools. This means that the ontology is meant to be used beyond the end of the Audio Commons project. This usage will invariably lead to future rounds of formal or informal evaluation, which may in turn lead to evolve the ontology as needed.

[8] https://w3id.org/ac-ontology/aco

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

6 References

[Horridge2009] Horridge, M., Bechhofer, S.: The OWL API: a Java API forworking with OWL 2 ontologies. In: Proc. OWLED 2009. vol.529, pp. 49–58. CEUR-WS.org (2009)

[Poveda2014] Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa,M.C.: Oops!(ontology pitfall scanner!): An on-line tool for on-tology evaluation. International Journal on Semantic Web andInformation Systems (IJSWIS) 10(2), 7–34 (2014)

[Shearer2008] Shearer, R., Motik, B., Horrocks, I.: HermiT: A Highly-Efficient OWL Reasoner. In: Proc. OWLED 2008. vol. 432,p. 91 (2008)

[Tsarkov2006] Tsarkov, D., Horrocks, I.: Fact++ description logic reasoner:System description. Automated reasoning pp. 292–297 (2006)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382