THE AUDIO EFFECTS ONTOLOGY

Thomas Wilmering, György Fazekas, Mark B. Sandler
Centre for Digital Music (C4DM)
Queen Mary University of London
thomas.wilmering@eecs.qmul.ac.uk

ABSTRACT

In this paper we present the Audio Effects Ontology for the ontological representation of audio effects in music production workflows. Designed as an extension to the Studio Ontology, its aim is to provide a framework for the detailed description and sharing of information about audio effects, their implementations, and how they are applied in real-world production scenarios. The ontology enables capturing and structuring data about the use of audio effects and thus facilitates reproducibility of audio effect application, as well as the detailed analysis of music production practices. Furthermore, the ontology may inform the creation of metadata standards for adaptive audio effects that map high-level semantic descriptors to control parameter values. The ontology is using Semantic Web technologies that enable knowledge representation and sharing, and is based on modular ontology design methodologies. It is evaluated by examining how it fulfils requirements in a number of production and retrieval use cases.

1. INTRODUCTION

The development of tools and services for the realisation of the Semantic Web has been a very active field of research in recent years, with a strong focus on linking existing data. In the field of music information management, Semantic Web technologies may facilitate searching and browsing, and help to reveal relationships with data from other domains. At the same time, many algorithms have been developed to extract low and high-level features, which enable the user to analyse music and audio in detail. The use of semantics in the process of music production however is still a relatively new field of research. With computer systems and music processing applications becoming increasingly powerful and complex in their underlying structure, semantics can help musicians and producers in decision processes, and provide more natural interactions with the systems.

Herrera and Serra [9] stressed the potential of semantic sound descriptors for the development of new audio applications in their work using MPEG-7 descriptors. They asserted that "there are [...] sound content-based processing applications waiting to be developed once we have a robust set of descriptors and structures for putting them into relation and for expressing semantic concerns about sound." We argue that Semantic Web technologies, such as Semantic Web ontologies and RDF are a superior choice for the representation of metadata in audio production, because they allow for a more flexible and extensible representation of this heterogeneous information domain. Beside, as de-facto standards of the future Web, they allow for sharing and linking structured information across different domains. Ontology-driven knowledge management in music production has also been discussed in [2,7]. Within this field, the main focus of our study is the representation of information about audio effects.

Audio effects play an integral part in modern music production. They modify an input signal and may be applied in order to enhance the perceived quality of a sound or to make more drastic changes to it in the composition process. Employing music information retrieval (MIR) and Semantic Web technologies specifically for the control of audio effects has the potential of representing a significant step in their evolution. This work therefore has a good potential to address a phenomenon described by Voorvelt [17]: "in the context of popular music production, the equipment in use generally trails the latest technological developments." Detailed descriptions of the use of audio effects in a music production project can additionally facilitate the reproduction of workflows, and add an additional layer of depth to MIR. For instance, the ontology can help answering queries such as: *Which effects have been used in a music production project and what are the parameter settings? Which effect implementations are available that suit the needs for a specific workflow?*

The Audio Effects Ontology presented in this paper is designed as an extension to the Music Ontology [14] and Studio Ontology [7], as well as our previous work detailed in [18]. First, we briefly discuss the Semantic Web Technologies underlying this work. Then we introduce the Audio Effects Ontology and provide an overview of its design and purpose. We discuss some applications in the domain of music production and music information retrieval, and finally outline directions of future work.

2. SEMANTIC WEB TECHNOLOGIES

The Semantic Web aims to bring intelligence to the Web by allowing machines to reason about Web content. With the proliferation of audio content on the Web, representing information about audio and its production is just as important as processing and linking text documents. The first step towards this goal is to represent information in a machine interpretable format.

A stack of technologies have been proposed for building the Semantic Web. The Resource Description Framework (RDF) is a data model for describing statements using *subject, predicate, object* triples. These form an RDF graph when combined. When the elements of RDF statements are identified by uniform resource identifiers (URI), we obtain an interlinked, globally distributed "database", the Web of Linked Data. However, to enable querying or reasoning over linked data, we need languages that describe the shared meaning of RDF graphs, in other words, represent knowledge about the entities described in data sets.

2.1 Ontologies and knowledge representation

Ontology languages like the RDF Schema language and the Web Ontology Language (OWL) [1] allow for characterising entities in terms of their relationships. They describe a shared conceptualisation of a world [8] comprised of individuals, classes and relations, with formal semantics that allow automated reasoning over RDF data expressed using an ontology. RDF Schema allows for defining simple hierarchies of classes and properties with a set of constraints over their use, but without adequate logical grounding. OWL refines this model by adding tools for representing domain knowledge more precisely. For instance, we can characterise properties in terms of transitivity, symmetry or reflexivity. OWL constructs directly correspond to Description Logics (DLs), a family of logic-based knowledge representation languages which in turn are based first-order logic [10].

3. THE STUDIO ONTOLOGY

The Studio Ontology is an OWL ontology for capturing the nuances of record production by providing an explicit, application and situation independent conceptualisation of the studio environment [7]. It is presented as a modular framework of ontologies, which in turn are built on the Music Ontology framework [14] and its components. It uses its core elements that allow for the representation of time-based events (Event and Timeline ontologies), and the workflow of music production in an editorial context subsumed under broader terms defined by the Functional Requirements for Bibliographic Records (FRBR) [13].

The Music Ontology allows for describing the music production workflow from composition to delivery, however, it lacks some concepts to do so in sufficient detail. The Studio Ontology provides some of the necessary extensions that form the foundation for a comprehensive representation of audio effects, and their application in music production. Here, we outline only those of its features and components which make it suitable as basis for our work.

- **Foundational components**: The Studio Ontology allows for characterising and describing the application of technological artefacts (devices) in music production. The Device Ontology provides a fundamental device and device decomposition model, and entities for representing device states, such as variable device parameters at different levels of granularity.
- **Complex device descriptions**: The ontology provides a model for describing complex devices such as signal processing tools and their interconnections. It includes a four-layered abstract model of these devices resembling the FRBR model.
- **Core components**: The ontology provides for describing recording studios on the editorial level (e.g. personnel and available equipment). It also provides for describing signal processing workflows in the studio using a parallel event and signal flow utilising music production tools.
- **Domain specific extensions**: The Studio Ontology supports the provision of domain specific extensions. It also provides some extensions for describing audio recording (e.g. microphones), mixing, editing and a core model for describing audio effects.

The application of audio effects to signals can be described using the concept studio:Transform defined by the Studio Ontology. This concept represents an event that takes a signal as a factor, and produces a transformed signal. This concept may be subsumed in more specific effect ontologies. The Studio Ontology sets aside the problem of defining specific audio effects, their classifications, parameters and their application specific descriptions. The Audio Effects Ontology fills this gap.

4. THE AUDIO EFFECTS ONTOLOGY

The aim of the Audio Effects Ontology is the representation of knowledge concerning audio effect implementations and their application in the music production studio. For instance, music software such as digital audio workstations (DAW) and digital effects implementations, may support the audio engineer by producing and reusing knowledge that is represented using the concepts and properties defined by the ontology.

4.1 The Core Ontology

The core parts of the Audio Effects Ontology define concepts and properties for the description of audio effect implementations and how they are applied within the production process. This facilitates the incorporation of data about the application of audio effects in online catalogues or content-based music recommendation systems, and allows for using an effects database in Semantic Web applications at large. For instance, this facilitates the retrieval of songs characterised by certain effects or effect types used in a production, and the reproduction of workflows.

Rather than signal processing devices, audio effects in our proposed ontology are conceptualised as physical and acoustical phenomena, that are represented on the same conceptual layer as the abstract *Work* entity in the FRBR model [13] of intellectual works. An effect is represented by the OWL class *afx:Fx*. Furthermore, the *Fx* class can be linked to effect types (see Section 4.2), thus adding meaning to the audio effect that may not be given solely by an implementation's given name. A separate class serves the purpose of describing signal processing devices, such as software implementations or hardware effect units.

The description of audio transformations in music production is another purpose of the ontology. To enable this functionality, the Audio Effects Ontology defines concepts for describing the application of effects to a signal. These concepts integrate seamlessly with the Studio Ontology that already provides a class for transformations (see Figure 1).

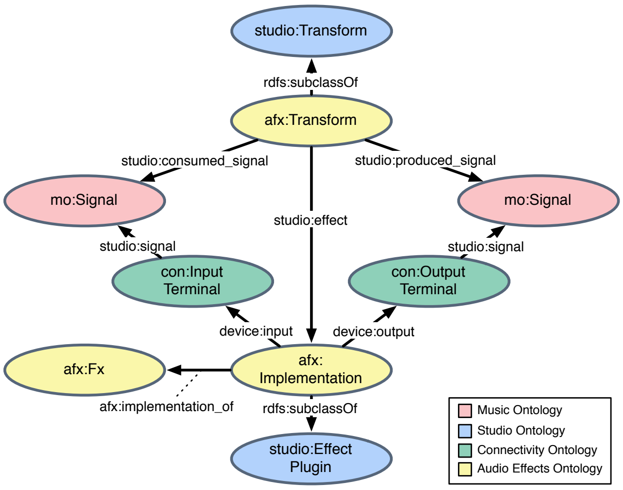

Figure 1. Transformation of an audio signal described using the Studio Ontology and Audio Effects Ontology.

A *transform* can be linked to an effect implementation using the *studio:effect* property.Using this mechanism we can express details about the effect implementation involved in a transformation and the device state at the time of the transformation. The ontology provides additional concepts for the description of implementation attributes, such as the plugin format, operating system, parameters and parameter settings (see Table 1). Instead of being conceptualised as static individuals of the respective classes, operating systems and plugin formats are conceptualised as subclasses, which enables a more detailed description of the execution context, for instance, by specifying a particular version number of an effect plugin. This implementation class may also act as the connection to the *afx:Fx* class specifying the effect type with the property *afx:implementation_of*. Associating events on an audio signal timeline — using appropriate Music Ontology terms — to a particular transform and its parameters, it is possible to state where a certain effect has been applied during the course of a track.

Finally, it is a common practice to automate effect parameters in music production, i.e. the parameters of a given effect may change over time. Modern DAWs are able to store the automation data for this purpose. In order to represent changing effect parameter values the Audio Effects Ontology provides the class *afx:State* which represents a similar concept as the *device:State* class, proposed in the context of *consolidated reification* in [6]. It conceptualises variable attributes and relationships of an audio effect. The *afx:State* class is a subclass of *event:Event* in the Event Ontology [1]. This allows describing a region on a signal timeline, during which a certain parameter setting is true. The region is defined by an *entry* and *exit* point. These are subclasses of *tl:Instant* of the Timeline Ontology [2].

4.2 Audio Effects Classification

In order to produce metadata describing the workflow and the elements of an audio production, it is beneficial to describe which specific audio effect implementation has been used for a given transformation, and also to specify the effect type. This facilitates the comparison of workflows independently from the tools that have been used in the studio. The conceptualisation of effect types for instance enables the search for similarities in the use of audio effects in a database of music production data. While the classification of audio effects has several applications in music production, the heterogeneity of possible taxonomies, as well as the many viable points of view for organising effects present research problems that are not easily solved. Creating extensible ontologies provide a possible solution to this problem. Musicians and music producers have a large number of digital audio effects at their disposal, while over 70 types of effects have been identified in academic research [21]. There are different approaches to the task of audio effects classification depending on a variety of factors. For instance, we may group effects by their perceptual attributes or classify them by their underlying signal processing implementations. The best classification depends on the intended use. A developer for example would probably want to emphasise signal processing techniques, whereas a musician would prefer to classify effects by their perceptual qualities. An example of inter-disciplinary effect classification has been proposed in [15], as part of an effort to facilitate communication and collaborations between DSP programmers, sound engineers, composers, performers and musicologists. To address this issue, we incorporated several linked classification systems subsumed under the concept *afx:Fx* in our ontology. These are based on different criteria, including technical aspects as well as perceptual attributes. As a result, an MIR system using the ontology may answer questions such as: *Which audio effects affect the timing of the audio material? Which productions used delay-based audio effects?*

4.3 Effect Parameters

Recognising the fact that not all audio effect implementations adhere to parameter naming conventions, we extend our ontology with the Parameter Ontology module. We conceptualise effect parameters in such a way that we can assign a parameter type to a parameter that is linked to an audio effect implementation. We distinguish between two types of parameters: numerical parameters and indexed pa-

[1] http://motools.sf.net/event/event.html
[2] http://motools.sf.net/timeline/timeline.html

concept	property	range	some subclasses/individuals
Product name	dc:title	*literal* (xsd:string)	-
FX format	available_as	Format	Vst, Au, Lv2, Rtas
Operating system	os	Os	Windows, MacOS, Linux
FX type	implementation_of	Fx	class1:Chorus, class2:Bandpass
FX technique	technique	Technique	SchroederMoorer, PhaseVocoder
FX parameters	parameter	Parameter	NumParameter, IndexedParameter
FX preset	preset	Preset	-
Audio inputs	audio_inputs	*literal* (xsd:int)	-

Table 1. Some of the concepts and properties for the description of a digital audio effect implementation.

rameters. The former is set by numerical values, while the latter consists of a list of string values. For instance, a parameter may be used to specify a filter type or a waveshape for an oscillator. We may want to query for effects that have a specific type of parameter (e.g. the delay time or attack time). Consistent parameter names, which are not necessarily given, are a prerequisite for efficient comparison. The Parameter Ontology solves this by providing concepts for parameter types. Instead of simply labelling a parameter with a literal stating its given name, linking parameters to conceptualised parameter types facilitates retrieving this information independently from the actual given parameter names. Furthermore, by specifying the unit (for the delay time this may be milliseconds or seconds) we can compare and transfer settings across different implementations. To achieve this we use concepts of the Quantities, Units, Dimensions and Data Types (QUDT) ontology [3] . Figure 2 shows a description for a delay time parameter.

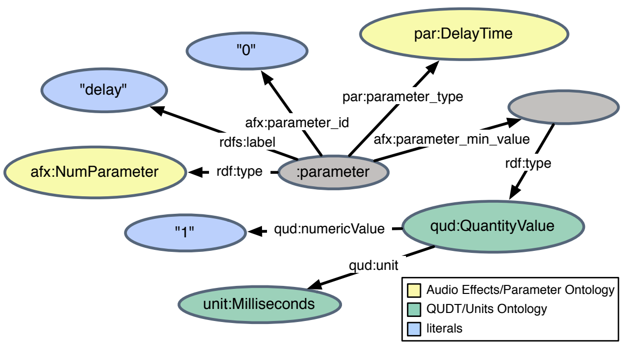

Figure 2. Partial description for a delay time parameter as it appears in an effect implementation.

4.4 Provenance

Provenance information describes entities, activities, and people involved in producing data. It enables software agents to track changes to data, thus ensuring a level of transparency and trust by providing information about the sources of data items. For instance, provenance information can consist of a statement about who created a particular resource. A detailed review of provenance ontologies, both general and discipline-specific, is provided in [5].

Digital audio effects alter audio data and consequentially the audio features associated with it. The studio ontology already provides mechanisms for describing the author, for instance an audio engineer, that was involved in the creation of a music production. Moreover, relating a *transform* with an effect implementation documents which software device has been used in the process. We introduce additional provenance properties in the Audio Effects Ontology for describing timed audio features that have been produced or altered by the application of an audio effect. Since the Studio and Audio Effects ontologies are developed in the context of future intelligent audio workstations that produce detailed metadata about the audio material and workflows in music production, the inclusion of the provenance properties facilitates adding effect-specific metadata to annotations of audio signals. For describing provenance information in the DAFX ontology, we introduce subproperties subsuming properties of the The Open Provenance Model Vocabulary (OPMV), that is based on the Open Provenance Model (OPM) [12]. In OPM, an *artefact* is defined as an ”immutable piece of state” which may refer to an actual physical object or a digital representation. A *process* is the action that creates artefacts, be it by acting on an existing artefact or by being caused by one. *Agent* describes an entity involved in a process by enabling or influencing its execution. Edges denote causal dependencies between its source (the effect) and its destination (the cause). OPM can be used in combination with terms from the Dublin Core specification which on its own we found to be insufficient for our requirements.

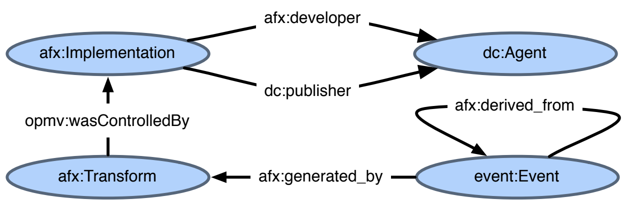

Figure 3. Provenance properties in the Audio Effects Ontology.

Using these properties of the Audio Effects Ontology (Figure 3), we can represent provenance information about audio features. For instance, an echo effect produces additional note onsets, since it adds the delayed signal to the original. We can express that such an event (artefact) has been *generated by* a given transformaion (process), and that the transformation *was controlled by* an audio effect implementation, the echo effect (agent). We can express that the existence of an audio feature produced by the transformation is dependent on a previously existing feature. For instance, a delayed note onset event may be *derived from* an onset in the original audio material prior to trans-

[3] http://www.qudt.org/

formation.

5. APPLICATIONS

5.1 Creation of Music Production Studio Databases for Information Retrieval

Designed as an extension of the Studio Ontology, knowledge represented with the Audio Effects Ontology can be seamlessly integrated into a database using the Studio Ontology framework. This information may include details about audio effects, such as the type of effects, their characteristics and parameter configuration of specific instances of effects applied in a music production project. Including detailed information about the application of audio effects may facilitate retrieval for various purposes for music production. This also facilitates reproducibility of workflows concerning the application of audio effects. Furthermore, by employing techniques similar to those applied in music recommendation systems based on Semantic Web technologies [4] [11], the Audio Effects Ontology provides a framework for the retrieval of audio effects given a set of specified criteria. These criteria can be technical aspects, as well editorial information such as the developers or vendors involved in audio effect implementations.

5.2 Publishing effect data on the Semantic Web

The Audio Effects Ontology is capable of refining the retrieval of songs based on production procedures as proposed in the context of the Studio Ontology framework. Data about audio effects may be published as Linked Data resources. This allows for the creation of an audio effects database on the Semantic Web, and facilitates the incorporation of data about the application of audio effects in music productions in online catalogues and content-based music recommendation systems. For example, this enables the retrieval of songs characterised by certain types of effects, or the actual effect used in a track. Assuming a large enough database of production data, this also allows for musicological research with regard to trends in the application of audio effects. Furthermore, plugin presets may be shared on the Semantic Web and retrieved by users and agents helping students or the work of professional engineers.

5.3 Adaptive Audio Effects

Previous implementations of adaptive audio effects [16] that map high-level features stored in a database to control parameters either use proprietary non-standardised formats, or MPEG-7 descriptors for the representation of audio features (e.g. [3]). However, the majority of metadata standards only specify the syntax of documents, while the semantics remain implicit and hardcoded in procedural software. Using Semantic Web technologies on the other hand, provide a uniform way of encoding and linking information. The Audio Effects Ontology provides the means for integrating adaptive audio effects seamlessly into a music production system that supports RDF based knowledge representation and retrieval. Examples of this new class of audio effects are given in [19, 20].

6. CASE STUDY: RETRIEVING AUDIO EFFECT SETTINGS BY QUERYING METADATA

Audio effect play a crucial role in creating the right "sound" of a track in most contemporary musical genres. Reproducing the application of effects including their exact parameter settings is therefore a very important use case.

The Audio Effects Ontology covers the necessary concepts to represent the information about an effect transformation in such a way that it is possible to query for a given effect and its parameter settings at a given temporal location relative to the audio signal. In the following we show an example of how we can query data represented with the Audio Effects Ontology to retrieve information about the application of audio effects in a music production project, where production metadata is available using our ontology framework. We may want to investigate an audio effect found in an annotated music production where one minute into the song an echo effect has been applied to the guitar. Using the appropriate SPARQL [4] queries we are able to retrieve the necessary information in order to identify comparable effect plugins present in our studio setup. First, we query the project database for the parameter types, settings and units of an echo effect applied on the specified track at the specified time instant (Listing 1). We assume the timeline for the guitar track as *:guitarTimeline* starting at the beginning of the project.

```
SELECT ?parameter_type ?value ?unit
WHERE {
 ?transform a afx:Transform ;
   studio:effect [ a afx:Implementation ;
     implementation_of ?class1:Echo ;
     afx:state ?state ] .
 ?event a event:Event ;
   event:time [ a tl; instant ;
     tl:timeline :guitarTimeline ;
     tl:at "60.0s"^^xsd:duration ] .
 ?state afx:entry ?event ;
   afx:parameter [ a afx:Parameter ;
     par:parameter_type ?parameter_type ;
     qud:value [ qudt:numericValue ?value ]
         ;
     afx:unit ?unit ] . }
```

Listing 1. Query retrieving effect settings.

The parameter unit specifications can form the basis of the conversion of settings between implementations having parameters of the same type with different units. This may be useful in case the implementations used originally are not available, and we wish to approximate the transformations with effects at our disposal in our studio. In a second step we query the database describing our studio facility for existing echo effect implementations having parameters of the same type (Listing 2). The query retrieves all the echo effect implementations available in our studio setup that have parameters of the same type as the one used in the production project in question. Since we also know the respective units for the parameter values it is possible to transfer the settings for the retrieved effect implementations. The information can act as a starting point for the

[4] SPARQL is a recursive acronym for SPARQL Protocol and RDF Query Language.

```
SELECT ?publisher ?fx_name ?time_name
 ?time_unit ?parameter_name ?parameter_unit
WHERE {
 :ourStudio studio:equipment ?device ;
 ?device a afx:Implementation ;
  dc:publisher [ fc:name ?publisher ] ;
  dc:title ?fx_name ;
  implementation_of [ class1:Echo ] ;
  has_parameter [ a afx:Parameter ;
   afx:parameter_type par:DelayTime ;
   rdfs:label ?time_name ;
   afx:unit ?time_unit ] ,
   [ a afx:Parameter ;
   afx:parameter_type par:LowpassFilter ;
   rdfs:label ?lowpass_name ;
   afx:unit ?lowpass_unit ] . }
```

Listing 2. Query retrieving effect implementations.

approximation of transformations performed by one implementation of a given effect type with another implementation of the same effect family.

7. CONCLUSIONS AND FUTURE WORK

We presented a Semantic Web Ontology covering the domain of audio effects and their implementations that is designed as an extension to the Studio Ontology framework. Using the ontology it is possible to create detailed metadata about the application of effects in music production projects and to classify and describe audio effect implementations. The applications of the Audio Effects Ontology range from adaptive audio effects using high-level semantic metadata, content-aware music production tools, and searchable audio effect databases. We have shown that querying RDF databases storing information about projects, studio equipment and available effect implementations enables access to detailed information about workflows, and facilitates their reproduction. Moreover, it allows for the analysis and comparison of musical works with regards to the use of audio effects.

Future work includes the development of software applications that support the Studio Ontology framework, such as content-aware audio production tools that automatically retrieve information and annotate multitrack projects automatically.

8. REFERENCES

[1] G. Antoniou and F. van Harmelen. Web ontology language: OWL. *in S. Staab and R. Studer (eds.), Handbook on Ontologies, International Handbooks on Information Systems*, Springer-Verlag Berlin Heidelberg, pages 91–110, 2009.

[2] K. Barkati, A. Bonardi, A. Vincent, and F. Rousseaux. GAMELAN: A knowledge management approach for digital audio production workflow. In E. Mercier-Laurent, editor, *Proceedings of Artificial intelligence for Knowledge Management Workshop (AI4KM) of ECAI*, 2012.

[3] O. Celma, E. Gómez, J. Janer, F. Gouyon, P. Herrera, and D. Garcia. Tools for content-based retrieval and transformation of audio using mpeg-7: the spoffline and the mdtools. *presented at the AES 25th International Conference, London, UK*, June 2004.

[4] Ò. Celma and X. Serra. FOAFing the music: Bridging the semantic gap in music recommendation. *Lecture Notes in Computer Science*, 4273, 2006.

[5] L. Ding, J. Bao, J. Michaelis, J. Zhao, and D. L. McGuinness. Reflections on provenance ontology encodings. *Proceedings of the 3rd International Provenance and Annotation Workshop (IPAW), Troy, New York, USA*, June 2010.

[6] G. Fazekas. *Semantic Audio Analysis - Utilities and Applications*. PhD thesis, Queen Mary University of London, 2012.

[7] G. Fazekas and M. B. Sandler. The studio ontology framework. *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011.

[8] M. R. Genesereth and N. J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, San Mateo, CA, USA., 1987.

[9] P. Herrera and X. Serra. Audio descriptors and descriptor schemes in the context of mpeg-7. *Proceedings of the International Computer Music Conference, Beijing, China*, 1999.

[10] I. Horrocks. Ontologies and the Semantic Web. *Communications of the ACM*, Vol. 51(12):pp. 58–67., 2008.

[11] K. Jacobson, G. Fazekas, Y. Raimond, and M. Smethurst. Music and the web of linked data. *Tutorial presented at the 10th International Society for Music Information Retrieval Conference (ISMIR), Kobe, Japan*, 2009.

[12] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. den Bussche. The open provenance model core specification (v1.1). *Future Generation Computer Systems*, Preprint, July 2010.

[13] M.-F. Plassard, editor. *Functional Requirements For Bibliographic Records : final report / IFLA Study Group on the Functional Requirements for Bibliographic Records*, volume 19. K.G. Saur, 1998.

[14] Y. Raimond, S. Abdallah, M. Sandler, and F. Giasson. The music ontology. *Proceedings of the International Conference on Music Information Retrieval*, 2007.

[15] V. Verfaille, C. Guastavino, and C. Traube. An interdisciplinary approach to audio effect classification. *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06), Montreal, Canada*, September 2006.

[16] V. Verfaille, U. Zölzer, and D. Arfib. Adaptive digital audio effects (A-DAFx): A new class of sound transformations. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5), 2006.

[17] M. Voorvelt. New sounds, old technology. *Organised Sound*, 5(2):67–73, 2000.

[18] T. Wilmering, G. Fazekas, and M. B. Sandler. Towards ontological representations of digital audio effects. *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11), Paris, France*, 2011.

[19] T. Wilmering, G. Fazekas, and M. B. Sandler. High level semantic metadata for the control of multitrack adaptive audio effects. *presented at the 133rd Convention of the AES, San Francisco, USA*, 2012.

[20] T. Wilmering and M. Sandler. RDFx: Audio effects utilising musical metadata. *presented at the 4th IEEE International Conference on Semantic Computing (IEEE ICSC 2010), Pittsburgh, PA, USA*, 2010.

[21] U. Zölzer. *DAFX - Digital Audio Effects*. J. Wiley & Sons, 2nd edition, 2011.