

# AUFX-O: Novel Methods for the Representation of Audio Processing Workflows

Thomas Wilmering<sup>(✉)</sup>, György Fazekas, and Mark B. Sandler

Centre for Digital Music (C4DM),  
Queen Mary University of London, London E1 4NS, UK  
{t.wilmering,g.fazekas,mark.sandler}@qmul.ac.uk

**Abstract.** This paper introduces the Audio Effect Ontology (AUFX-O) building on previous theoretical models describing audio processing units and workflows in the context of music production. We discuss important conceptualisations of different abstraction layers, their necessity to successfully model audio effects, and their application method. We present use cases concerning the use of effects in music production projects and the creation of audio effect metadata facilitating a linked data service exposing information about effect implementations. By doing so, we show how our model facilitates knowledge sharing, reproducibility and analysis of audio production workflows.

## 1 Introduction

Audio effect devices are essential tools in contemporary music production and composition and are used to enhance the perceived quality of an audio signal, as well as for creative sound design. They can be defined as signal processing techniques used to modify an audio signal in a controlled way. Audio effect devices can take the form of hardware units or software applications, both of which find widespread use in professional and home studios. Effects usually exhibit an interface that allows setting parameters of the implementation, thereby adjusting the behaviour of the signal processing mechanism. Detailed descriptions of effect types and their implementation can be found in [11]. In light of the popularity of Web services for music production and consumption, significant added value may be provided by a thorough description of production workflows using well defined concepts and properties. Furthermore, multimedia and audio specifically constitutes a significant part of the Web. This should be made better accessible and represented on the Semantic Web. Recommendation systems and cultural archives consuming Linked Data about music production will require the transparent, interoperable provenance metadata this ontology enables.

This paper introduces the Audio Effect Ontology (AUFX-O)<sup>1</sup> for the description of audio signal processing units and workflows in the context of music production [8]. The ontology builds on previous work, such as the Music Ontology framework [6] and borrows ideas from the Vamp Plugins Ontology for

---

<sup>1</sup> Specification at <https://w3id.org/aufx/ontology/1.0#>.

the description of feature extraction plugins and transformations<sup>2</sup>. It extends the Studio Ontology which is designed to capture the work of the audio engineer and producer in the studio [2]. It enables a detailed description of audio effects, audio effects implementations and their application in music production projects, including domain-specific provenance information based on the generalised model of the Provenance Ontology (PROV-O) [4]. This present work extends and refines previous theoretical models [10] and describes the first published version of the ontology. The need for layered conceptualisation of audio effects to support the description and analysis audio production workflows is discussed, followed by use cases concerning the application of effects in music production and their description in linked data services. Finally we conclude and present future directions.

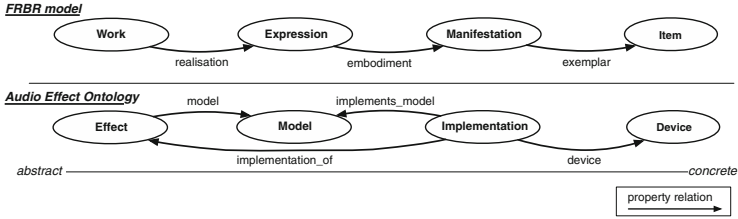
## 2 The Audio Effect Ontology

### 2.1 Conceptual Model of Audio Effects

The representation of effects and effect implementations in the Audio Effect Ontology closely follows the conceptualisation of devices in the Device Ontology within the Studio Ontology framework [2]. An audio effect is conceptualised as some perceived effect of a physical process. For example, the reflection of sound waves (echo) produces an audio effect. This physical phenomenon is the most abstract element of our model. Its level of abstraction similar to the abstract concept of intellectual works in the Functional Requirements for Bibliographic Records (FRBR) [5]. An algorithm that reproduces the effect can be described as a model that approximates the perceived effect. An echo effect, for instance, can be modelled by various delay line arrangements, comparable with how a work may be realised through several expressions. We define the level of abstraction of a model's implementation as being analogous with that of manifestations. Different implementations of a model in our domain may be an algorithm implemented in different programming languages, or a circuit design implemented using discrete components. On the most concrete level, an instance of an implementation represents a concrete device, for example an audio effect plugin running on a specific computer. Using this model we can describe concrete instances of audio effect devices, linked to an actual implementation of a model or algorithm that represents a physical phenomenon. Figure 1 shows these main concepts and the four levels of abstractions.

AUFX-O introduces additional concepts to describe a product for the distribution of a given audio effect implementation published by an individual or a software company. As opposed to the solution previously employed in the Device Ontology where product names are expressed as string literals linked to individual devices, in AUFX-O specific concepts for products and product families can be linked to implementations. This offers increased expressivity and allows the description of audio effects in a database that assigns URIs to products and

<sup>2</sup> <http://vamp-plugins.org/rdf/>.



**Fig. 1.** Entities and relationships in the model of audio effect devices in the Audio Effect Ontology. The levels of abstraction are comparable to those of the FRBR model.

different versions of a product unified in a product family, independently from instantiations of devices in a music production scenario. Implementations and products need to be clearly distinguished – different versions of a *product*, for instance, updates of audio effect software or versions targeting different operating systems may exhibit different characteristics and functionality, but are nevertheless marketed under the same name.

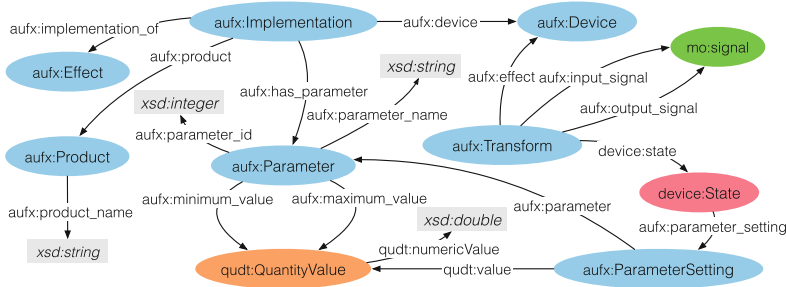
## 2.2 Modelling Details

AUFX-O is designed with two main areas of application in mind: (1) the description of audio effects on the implementation level and (2) the description of audio signal transformations on the device level. Modelling decisions draw from the authors’ expertise in the domain and specific use cases, including collecting, sharing and querying production metadata in digital audio workstations (DAWs) and web-based tools for linking and retrieving such data. Data-driven evaluation [1] of the ontology was presented in our earlier work [10] showing good lexical fit using a text corpus consisting of product descriptions of over 3000 effect implementations.

Figure 2 shows an overview of how implementations and transformations are modelled in AUFX-O. The **Implementation** is linked directly to the **Effect**, skipping the **Model** layer with the term **implementation\_of**. This property has been introduced to enable efficient querying. In many cases, especially for commercial effect implementations, the model or algorithm employed is not made public. An important factor in the description of an effect implementations is the precise representation of the variable parameters exposed to the user. These parameters allow the user to alter the behaviour of a signal processing device, i.e. to adjust the sound output within the constraints imposed by the parameter range. In AUFX-O, an **Implementation** can be linked to a **Parameter** using the property **has\_parameter**. To describe a parameter’s value range, we reuse terms from the Quantities, Units, Dimensions and Data Types Ontology (QUDT) [3]. The terms **minimum\_value** and **maximum\_value** are subsumed by the property **qudt:value**, enabling the explicit, unambiguous description of parameter values.

The **Transform** class represents a signal transformation performed by a **Device** which stands for a concrete instance of an **Implementation**. To describe

the device state at the time of transformation, we reuse the `device:State` concept of the Device ontology. The settings of variable parameters are represented by a `ParameterSetting` linked to a given `Parameter`. The value of the setting is specified using the `qudt:value` property of QUDT. An audio effect performs a signal transformation, i.e., it takes an input signal and produces a new signal, the output signal, by applying the effect in a controlled way. The terms `input_signal` and `output_signal` link the `Transform` to instances of `mo:Signal` of the Music Ontology.



**Fig. 2.** Selected concepts for the description of implementation characteristics and signal transformations, reusing concepts of the Music Ontology (mo), Device Ontology (device) and QUDT (qudt).

In addition to the properties specifying parameter values, multiple other terms of AUFX-O are subsumed by concepts of existing ontologies. The four abstraction layers for the conceptualisation of audio effects from physical phenomenon to concrete device is based on the Device ontology (see Sect. 2.1). The AUFX-O classes **Effect**, **Model**, **Implementation** and **Device** are subsumed by respective terms of the Device Ontology. To increase flexibility of AUFX-O, it

**Table 1.** Core classes and properties of AUFEX-O subsumed by concepts of the Studio/Device Ontology (studio/device), QUDT (qudt) and PROV-O (prov).

AUFX class	subclass_of	AUFX property	subproperty_of
Effect	device:Phenomenon	device	device:exemplar
Model	device:Model	effect	studio:effect, prov:wasAssociatedWith
Implementation	device:Implementation	input_signal	device:consumed_signal, prov:used
Device	device:Device, prov:Agent	output_signal	device:generated_signal, prov:generated
Transform	prov:Activity	minimum_value	qudt:value
		maximum_value	qudt:value

defines inverse property relations to link instances of these classes. Several concepts are linked to PROV-O to enable interoperable provenance metadata. For instance, a **Device** is defined as a **prov:Agent** while **Transform** is a subclass of **prov:Activity**. Table 1 provides an overview of some core terms and their relations to existing ontologies. Data-driven evaluation of AUFX-O measuring “fit” to a corpus of audio effect related documents with 246544 stemmed words and 8023 unique stems is given in [10].

### 3 Use Cases

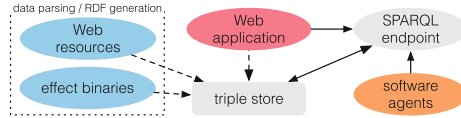
#### 3.1 Audio Effects Product Database on the Semantic Web

Linked data about audio effects may benefit music production professionals and amateurs alike. For instance, linking effect implementations by their available parameters or sonic properties can help finding the right effect for a production scenario. Linking effects and parameter settings to released audio has several applications too, for example in music education. Albeit there are already existing online databases of audio effects, these do not provide standard query end points and do not rely on a common, interoperable conceptualisation and machine readable representation of effect data.

A linked data service using AUFX-O exposes metadata about audio effect implementations and a Web application for data entry and retrieval [9]. An overview of the database and its use is shown in Fig. 3. The data is exposed via a triple store that can be accessed by software agents. Content of the database is initially obtained by parsing Web resources and audio effect binaries that expose information about their structure. This is translated to RDF. For instance, unstructured data retrieved from the KVR audio effect database<sup>3</sup> is mapped onto AUFX-O. The KVR database is accessible through a Website and contains descriptions of over 3500 digital audio effect products, including developer and product name, plugin APIs, operating systems, pricing, and product information in text form. Effects are classified by user-generated tags indicating the effect type, which can be mapped to AUFX-O using the `kvr_tag` property. Listing 1 shows the partial description of an effect implementation. It includes details such as product name and versioning information, as well as the plugin API and parameter description. AUFX-O is capable of representing the information in KVR, however, it also allows for more fine-grained semantic descriptions using the terms discussed in Sect. 2.2.

This linked data service enables novel queries that automate manually performed tasks. An audio engineer may want run simple queries finding audio effects with a given plugin API, in order to identify effect implementations compatible with a given studio setup. Furthermore, users and software agents can find similar audio effects based on their parameter setup and specific tags for the automatic classification of effects following specified implementation characteristics. The detailed description of effect parameters enables more complex

<sup>3</sup> <http://www.kvraudio.com/plugins/effects/>.



**Fig. 3.** High level overview of the audio effects database system

---

```

:ex_implementation a :Implementation ; :plugin_api [ a :AudioUnit ] ;
:ex_product [ :product_name "exDelayFx"^^xsd:string ;
:kvr_tag "Delay / Echo"^^xsd:string ; version "1.0"^^xsd:string ] ;
:has_parameter [ a :NumericParameter ;
:rdfs:label "delay time"^^xsd:string ;
:maximum_value [ qdt:numericValue "18.0" ; qdt:unit unit:Decibel ] ;
:minimum_value [ qdt:numericValue "-18.0" ; qdt:unit unit:Decibel ] ;
:parameter_id "12"^^xsd:integer ] .

```

---

**Listing 1.** Partial Description of an audio effect implementation.

queries with regards to the capabilities of effect implementations. For instance, the SPARQL query in Listing 2 retrieves Echo effect implementations that are compatible with Audio Unit hosts and allow effect settings delaying the incoming signal by at least 1000 ms. In this particular example, the returned data contains the publisher and product name. Integrated in a DAW, this database can be the basis for an effect recommender system. This may return implementations that are available to the user, i.e., plugins that are actually installed on the system, or may refer the user to online sources, e.g., companies offering similar effects.

---

```

SELECT ?publisher ?name WHERE {
  ?x a :Implementation ; :plugin_api [ a :AudioUnit ] ;
  :has_parameter [ rdfs:label "delay time" ;
  :maximum_value [ qdt:numericValue ?max ; qdt:unit unit:Millisecond ] ;
  :product [ dc:publisher [ fc:name ?publisher ] ;
  :product_name ?name ; :kvr_tag "Delay / Echo"^^xsd:string ] .
FILTER ( ?max >= 1000 ) }

```

---

**Listing 2.** SPARQL query retrieving echo effect implementations with a delay time parameter that ranges above 1000 ms.

### 3.2 Semantic Metadata in Digital Audio Workstations

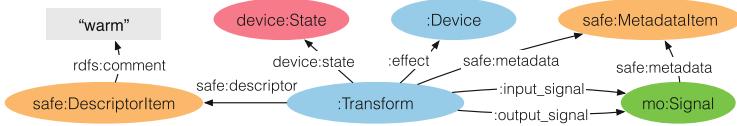
Audio effects used in music production are usually controlled by parameter settings linked to variables in an effect algorithm that alter low-level properties of the audio signal. Configuring an audio effect to achieve a desired outcome often requires in-depth technical knowledge about the algorithms. Music producers and musicians however often describe sound transformations using semantic descriptors such as *warm*, *bright* or *harsh*. These descriptors cannot always be mapped to effect parameters and may have a nontrivial relationship to the settings. The Semantic Audio Feature Extraction (SAFE) project<sup>4</sup> aims to address

<sup>4</sup> <http://semanticaudio.co.uk>.

this by interpreting the relationship between parameter settings, audio signal features and semantic descriptors to provide a queryable database of signal transformations associating descriptors with effect settings [7].

Specialised audio effect plugins have been developed for the project to gather data including effect parameters, changes in audio features, details about the music as well as the user including the intended sonic outcome (e.g. warm). A triple store is used on the server side to collect this information using ontologies including AUFX-O (see Fig. 4). Here, the **Transform** becomes a core concept to link crucial metadata elements. Querying the SAFE data facilitates novel ways for finding effect settings using semantic descriptors. A producer for instance can find an equaliser setting for processing a guitar track with the intention of giving it a warmer sound. The SPARQL query in Listing 3 retrieves equaliser settings associated with this term, as well as the instrument and music genre, enabling effect parameters to be set solely on the basis of high-level descriptions. Effect settings may also be retrieved by specifying expected change in the audio features.

At the time of writing, the publicly available effect plugins have been downloaded over 4500 times. A triple store collecting data from effect application enables sharing data among a community of users, both music producers and researchers. A dataset of over 240 million triples describing several thousand effect transforms and audio analysis results will be published in the near future.



**Fig. 4.** Data model for the description of signal transformations in the SAFE project.

```
SELECT ?descriptor ?paramName ?paramValue WHERE {
  ?transform :effect [ :device_of
    aufxdb:implementation_SFEQ_VST_1.32 ] ;
    :input_signal ?inputSignal ; device:state ?state ;
    safe:Descriptor [ rdfs:comment ?descriptor ] .
  FILTER REGEX(?descriptor, "warm", "i").
  ?inputSignal safe:Metadata [ rdfs:label "genre" ; rdfs:comment ?genre ] ;
    safe:Metadata [ rdfs:label "instrument" ; rdfs:comment ?instrument ] .
  FILTER REGEX(?genre, "reggae", "i") .
  FILTER REGEX(?instrument, "guitar", "i") .
  ?state :parameter_setting [ :parameter [ qdqt:value ?paramValueNode ;
    :parameter_id ?paramId ] ] .
  ?paramValueNode qdqt:numericValue ?paramValue .
  aufxdb:implementation_SFEQ_VST_1.32 :has_parameter [
    :parameter_id ?paramId ; rdfs:label ?paramName ] .
} ORDER BY ?transform ?paramId
```

**Listing 3.** SPARQL query to retrieve equaliser parameter settings being used to alter a guitar sound in the reggae genre, semantically described as *warm*.

## 4 Conclusions and Future Work

This paper discusses the newly published Audio Effect Ontology for describing of audio effects, their implementations and their use in music production. Through several use cases we have shown that detailed metadata about audio effects provided as Linked Data facilitates novel applications for the description and analysis music production workflows, including the search and retrieval of audio effect implementations. Educational tools focusing on audio production and engineering may also benefit from AUFEX-O. The modelling principles for effect parameters and settings are applicable in other domains where device settings for specific processes are of interest, e.g. for signal processing devices in other media domains or laboratory devices in scientific experiments.

Future work includes the publication of effect classification vocabularies extending AUFEX-O, both for audio effect types and parameter types. These vocabularies combine low-level descriptors for technical classification targeted to audio engineers, and high-level descriptors and auditory perceptual attributes which may be preferred by musicians and composers. Well-defined conceptualisations of effect and parameter types can further improve audio effect related semantic descriptions and enable tasks such as querying for similar audio effects based on their types or parameters. The Studio Ontology and AUFEX-O may also be used to support intelligent music production tools, such as Web-based audio workstations capable of storing and interpreting semantic metadata to inform the production process.

**Acknowledgments.** This paper is supported by EPSRC Grant EP/ L019981/1, Fusing Audio and Semantic Technologies for Intelligent Music Production and Consumption and the European Commission H2020 research and innovation grant AudioCommons (688382). Mark B. Sandler acknowledges the support of the Royal Society as a recipient of a Wolfson Research Merit Award.

## References

1. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data driven ontology evaluation. In: Proceedings of the International Conference on Language Resources and Evaluation, Lisbon, Portugal (2004)
2. Fazekas, G., Sandler, M.B.: The studio ontology framework. In: Proceeding of the 12th International Society for Music Information Retrieval Conference (ISMIR) (2011)
3. Hodgson, R., Keller, P.J., Hodges, J., Spivak, J.: QUDT - quantities, units, dimensions and data types ontologies (2014). <http://www.qudt.org/>
4. Lebo, T., Sahoo, S., McGuinness, D.: PROV-O: the PROV ontology. W3C recommendation, 30 April 2013. World Wide Web Consortium (2013)
5. Plassard, M.F. (ed.): Functional Requirements For Bibliographic Records : final report/IFLA Study Group on FRBR, vol. 19. K.G. Saur (1998)
6. Raimond, Y., Abdallah, S., Sandler, M., Giasson, F.: The music ontology. In: Proceeding of the International Conference on Music Information Retrieval (ISMIR), Vienna, Austria (2007)



7. Stables, R., Enderby, S., Man, B.D., Fazekas, G., Reiss, J.G.: SAFE: a system for the extraction and retrieval of semantic audio descriptors. In: 15th International Society for Music Information Retrieval Conference (ISMIR), Taipei, Taiwan (2014)
8. Wilmering, T., Fazekas, G.: AUFX-O: the audio effect ontology (2016). <http://isophonics.net/content/aufx>
9. Wilmering, T., Fazekas, G., Allik, A., Sandler, M.B.: Audio effects data on the semantic web. In: 139th Audio Engineering Society Convention, New York, USA (2015)
10. Wilmering, T., Fazekas, G., Sandler, M.B.: Semantic metadata for music production projects. In: Proceeding of the 1st International Workshop on Semantic Music and Media (SMAM), Sydney, Australia (2013)
11. Zölzer, U. (ed.): DAFX - Digital Audio Effects, 2nd edn. Wiley, New York (2011)