THE STUDIO ONTOLOGY FRAMEWORK

György Fazekas and Mark B. Sandler
Queen Mary University of London, Centre for Digital Music
gyorgy.fazekas@eecs.qmul.ac.uk

ABSTRACT

This paper introduces the Studio Ontology Framework for describing and sharing detailed information about music production. The primary aim of this ontology is to capture the nuances of record production by providing an explicit, application and situation independent conceptualisation of the studio environment. We may use the ontology to describe real-world recording scenarios involving physical hardware, or (post) production on a personal computer. It builds on Semantic Web technologies and previously published ontologies for knowledge representation and knowledge sharing.

1. INTRODUCTION

Recognising that simple metadata based approaches are insufficient in complex music information management and retrieval scenarios, researchers has been focusing on using cultural information and the use of content-based features extracted from commercially released audio mixtures. Certain types of these information are rapidly becoming available on the Semantic Web and via a number of Web services. For example, events (concerts, tour dates) and artist relations can be obtained and used in intuitive ways to find connections in music [12]. However, these data remain largely editorial, and focussed on artists as opposed to music and production. We argue that another invaluable source of information exist, largely neglected to date, pertaining to the composition context, history, production and pre-release master recordings of music. Due to the lack of comprehensive open standards and methodologies for collecting production information, its use hasn't been explored yet.

While music making is an increasingly social activity, the Semantic Web could become a platform for sharing not just music, but ideas between artists and engineers. To facilitate this process, our ontologies can be utilised to denote information about music production, and propagate it through the recording workflow. They enable building better models for music information retrieval (MIR), and answering queries such as: *How was this song produced? What effects and parameters were used to achieve that particular sound of the guitar? How was the microphone array configured when recording this orchestra?*

In the rest of this paper, we first discuss why we depart from existing metadata standards, and use Semantic Web technologies and the Music Ontology [20] instead. Next, we introduce the Studio Ontology framework focussing on its foundations. Finally, we discuss some applications and conclude.

2. RELATED WORK

Numerous metadata standards are available to capture at least parts of the information we outlined previously. However, their adaptation in audio applications remains low, while a large number of concerns have been reported by researchers, developers and end-users [3], [11], [21], [22], [1]. The reasons are complex, and beyond the scope of our discussion, see [17], [6]. for thorough reviews. Instead, we summarise the main causes which makes us move away from the adaptation of existing metadata standards.

Perhaps the most important problem is the prevailing use of XML instead of logical data models. XML specifies the structure of a document, but it is insufficient in itself for defining relationships and constraints over a set of terms, hence their meaning remains ambiguous [14]. Interoperability is hindered by the lack of semantics, which also prevents automated reasoning over data sets. Essential vocabulary terms are scattered across different domains. While harmonisation is possible, it requires reverse engineering [3], [11], [1] and it remains unclear if and how these efforts can converge into a clear common conceptual model. Finally, the lack of shared unique identifiers makes publishing, linking and the use of these data difficult in anything but small MIR problems.

Notable frameworks to facilitate interoperability in research include ACE XML [16], for sharing content-based features in MIR, and the Integra Extensible Data format (IXD) [4] linking audio processing and composition environments such as PD or Max/MSP. These XML-based formats however are too specific for our use, difficult to extend, and suffer from the same drawbacks mentioned above. In the next section, we outline how Semantic Web technologies can be used to avoid these drawbacks.

3. KNOWLEDGE REPRESENTATION IN THE RECORDING STUDIO

The dual role of the sound engineer can be characterised by the aim of fulfilling artistic goals on one hand, and by the use of specific domain knowledge on the other. Capturing this knowledge, the aesthetic choices, and the use of tools in music production workflows is the primary focus of our research. It requires formalised data models and languages to represent, structure, transfer, store and query this information.

A naïve model for information management simply attaches metadata tags to audio items, but further descriptions of the entities described by tags is not possible. A relational data model resolves this issue, however its common implementation is not sufficient in itself for knowledge representation: We can not describe a hierarchy between tables or constraints over the use of terms in relational database schemata. Object orientated models resolve these limitations, but they have no sound theoretical foundations, do not support efficient query evaluation, or logical reasoning. Graph based models, such as the Resource Description Framework (RDF) [1], and expressive Description Logic (DL) [10] and Semantic Web ontology languages provide a better alternative. We briefly introduce these techniques next.

3.1 Semantic Web Technologies

Semantic Web technologies include Web standards for communication and information sharing. The Uniform Resource Identifier (URI), provides a unique naming scheme for concepts and relationships (resources), while RDF allows structuring data using simple statements consisting of *subject—predicate—object* triples. A set of triples is seen as a graph of semantic relationships. Each term is identified using a URI, which enables them to quote other resources creating a Web of structured and *linked data* [2]. RDF ensures clear separation of syntax from semantics and conceptual model. There are concise human readable serialisations like N3 [3] and an efficient query language called SPARQL [4] supported by several databases and open source libraries.

Using RDF alone, one can make rather arbitrary statements however, therefore to have common ground for applications to interpret our data, we need to be able to define, and later refer to concepts such as a *Song* or an audio processing *Plugin* and its parameters, as well as their pertinent relationships. Ontology languages provide for these definitions to be declared, while *knowledge representation schema* describing a domain is what we call an *ontology*.

3.2 Knowledge Representation and Ontologies

Ontology languages such as the Ontology Web Language (OWL) [5] are formal languages to express a shared *conceptualisation* [6] of a domain. Although using a formal language facilitates syntactic interoperability in itself, making *ontological commitments* [7] pertaining to the meaning of terms require higher level constructs of a logical system. The presence or lack of this system signifies the difference between data models and knowledge representations. Most Semantic Web ontologies are based on Description Logics corresponding to fragments of First Order Logic for which practical reasoning procedures [10] can be created. The Music Ontology and the Studio Ontology are published in OWL.

4. OVERVIEW OF THE MUSIC ONTOLOGY

The Music Ontology provides a clear conceptualisation of the music domain to facilitate publishing music-related data on the Semantic Web. It was introduced in [20] and thoroughly described in [19]. We refer the reader to the literature for an introduction and its applications. Here, we outline some features which make the Music Ontology more suitable for our work than its alternatives [1], [13], [8] [11].

- **Modular and extensible design**: Published as a modular ontology library whose components may be reused or extended outside of its framework.
- **Workflow-based conceptualisation** of the music domain: It is built on the life-cycle of intellectual works — defined in the Functional Requirements for Bibliographic Records (FRBR) [18], — ranging from abstract to concrete entities: *MusicalWork*, *Expression*, *Manifestation*, *Item*.
- **Event decomposition model:** Events are modelled as first-class objects with participating agents and passive factors, and may be decomposed into sub-events.

[1] http://www.w3.org/TR/rdf- syntax/
[2] http://linkeddata.org
[3] http://www.w3.org/DesignIssues/Notation3.html
[4] http://www.w3.org/TR/rdf-sparql-query/
[5] http://www.w3.org/TR/owl-ref/
[6] Formally, a set of relations **R** over a universe of discourse D. [9]
[7] We say that an agent commits to an ontology if its observable actions are consistent with the definitions in the ontology. [9]

- **Timelines and temporal entities** can be used to localise events on different timelines: *abstract*, *discrete*, or *continuous*; *relative*, or *physical*.
- **Adaptation**: It has become a de-facto standard to publish music-related data on the web.

The above models provide the basis for content annotation as well as the decomposition of events in complex workflows, so that we can precisely say *who* did *what* and *when*. While elements of these models can also be found in other ontologies, they are not present all at once in a single unified framework. The Music Ontology provides a model to describe the production workflow from composition to delivery, including music recording, but it lacks some very basic concepts to do so in detail. The Studio Ontology fills this gap.

5. THE STUDIO ONTOLOGY FRAMEWORK

The Studio Ontology [8] is presented as a modular and extensible ontology library. It is designed to reuse existing terms and models published elsewhere that fit its requirements. The framework contains some general, domain independent elements, a set of core concepts and relationships to describe the studio domain, and some extensions covering more specific areas like microphone techniques and multitrack production tools.

5.1 Foundational elements

The foundational parts of the ontology deal with describing tools in audio engineering workflows.

5.1.1 Workflows, Events and Timelines

We distinguish between two types of workflows: *prescriptive* and *descriptive*. Prescriptive workflows are best understood as templates describing common data access and manipulation steps. Descriptive workflows may be seen as denotation of specific instances of the above, broadly speaking a description of *who* (or *what*) produced *what*, *when*, and *how*, *using what*. Such a description requires a workflow based conceptualisation of entities existing at various stages. The Music Ontology provides such a conceptualisation: A composition (*MusicalWork*) may be performed producing a sound, which may be recorded producing a signal (*MusicalExpressions*). We obey this model and hook into it exactly at this level. When the sound engineer manipulates a sound or a signal, new expressions are created to which additional information can be attached on how it was produced. In order to describe this process, we need to be able to talk about *events* (performance, recording, mixing, transformation), which may be spatially and temporally localised, and linked with *agents* (engineer) and *factors* (tools). We use the Event and Timeline Ontologies [20] for this purpose. The Music Ontology sets aside the problems of *how* and *using what* from the workflow above. We address this issue next.

[8] http://isophonics.net/content/studio-ontology

5.1.2 Technological Artefacts

The Device Ontology can be used to describe artefacts of technology. The *Device* concept may be subsumed by anything, a watch, a plugin, or a microphone in a more specific ontology. Our ontology generalises concepts from [7], [2], which are specific for their application domains, namely smart phones and computer networks. Similarly to the Event and Timeline Ontologies, the Device Ontology approaches a foundational domain independent status in the sense described in [15]

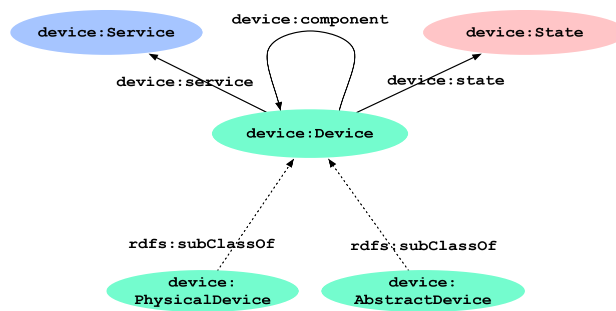

Figure 1. Overview of the Device Ontology

A device may participate in an event as a passive factor, providing a particular *service* in a particular *state*. A *state* may be useful to represent a *configuration*, such as the polar pattern or sensitivity settings of a microphone during a recording. We borrow knowledge representation elements from the OWL description of UML state machines of [5]. This resembles the paradigm of event driven finite state machines, in that it describes events related to an application of a device as reason for state changes. Events are tied together as sub-events of a main event. This has the benefit of encoding chains of state changes (in a temporal context), and the ability to assign additional information to entry and exit conditions modelled as events themselves. This may be a link to an engineer to encode details such as an option turned on by one engineer and then turned off by another, or classifications of change events, such as automatic control, fault conditions or engineering decisions.

Our ontology commits to a categorical distinction between physical and abstract devices, which worth making for the following considerations. Physical and abstract objects have different primary characteristics. For instance, physical devices have size and weight, and may be decomposed into physical or abstract components, such as an extension module or firmware. Abstract devices on the other hand may be intangible models of physical devices. Form a mereological point of view our model expresses a partial order relation on the set of components of a device, which is a reflexive, transitive and anti-symmetric property [9] .

5.1.3 Signal Processing Devices

An important class of devices in music production are tools for manipulating audio signals. We define the concept *SignalProcessingDevice* as a subclass of the more general device concept described in §5.1.2, having inputs and outputs for signal connectivity. From an ontological point of view this is sufficient to identify a signal processing device. It is interpreted broadly, and may stand for anything from a basic filter to a complex unit such as a mixing console or an audio effect. The concept is defined in a dedicated ontology called the Signal Processing Device Ontology, together with some fundamental signal processing components.

5.1.4 Device Connectivity

The Connectivity Ontology allows for describing how signal processing devices, or other tools, such as microphones, in a recording and processing workflow are interconnected. Its paramount concept *Terminal* represents inputs and outputs in an abstract way, encompassing electrical or software interfaces and may be linked with a particular physical connector and communication protocol. In figure 2 we illustrate its basic structure. The exemplified instances of *Connector* and *Protocol* can be thought to represent the output of a digital microphone having a 3 pin male XLR connector, and using the AES42 digital microphone interface protocol. The ontology defines some individuals of connectors and protocols common in audio production. An interesting feature of the ontology is that we can use it to match signal characteristics to interface characteristics, for instance the number of accepted channels.

5.2 Core components

The core Studio Ontology parallels the three levels of expressiveness of the Music Ontology and provides studio specific extensions. On the first level it provides for describing recording studios and facilities. For example, we can differentiate between commercial, project and home studios, different audio engineering roles such as mixing or mastering engineer, describe various recording rooms and the equipment in them. This includes a large vocabulary of tools with top level concepts such as *Amplifier*, *Analyser*, *MixerDevice*, *MonitoringSystem*, *EffectUnit*, *DigitalAudioWorkstation* or *Plugin*.

[9] Note that it requires OWL2 to express all constraints.

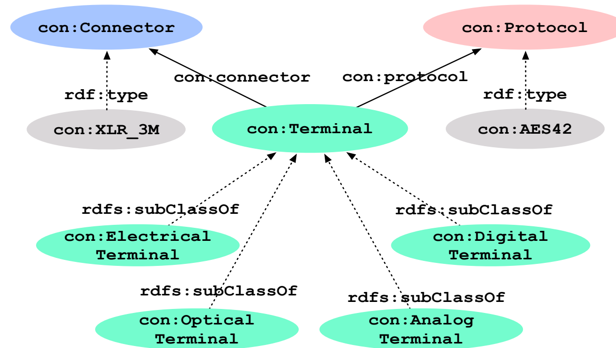

Figure 2. Overview of the Connectivity Ontology (with simplified examples)

The second level includes complex events such as different types of recording and post production sessions, and provides for describing the production workflow on the level of audio transformations and signal processing as described in §5.2.1

The third level provides some extension points to describe specific tools, such as multitrack audio production software (see §5.3.4); the audio editing workflow and project structure.

5.2.1 Signal Processing Workflows

To describe how a piece of music is processed in the studio, it is insufficient in itself to describe a signal flow (i.e. flow chart) or a set of transformations. We need to consider a random set of mixing or transformation events, as in non-linear editing, as well as real-rime, quasi-simultaneous [10] transformations, such as a signal routed through several processing units for recording. To fulfil both requirements, we consider parallel signal and event flows linked using signal entities that are instances of the `mo:Signal` concept. This is illustrated in figure 3. The concepts *Recording, Mixing,* and *Transform* are subclasses of *Event* defined in the Event Ontology (see §5.1.1) while *MixerDevice* and *EffectUnit* subsume *SignalProcessingDevice* defined in §5.1.3. Several signals (not shown for brevity) can be attached to a mixing event and corresponding device. This set up signifies our ontological commitment to changing identities, a problem thoroughly discussed in philosophy [23]. Once transformed, a signal receives new identity which alleviates difficult transaction management problems in our system regarding the changing attributes of signals.

5.3 Extensions

Ontology extensions are useful to allow the user to choose a desired level of granularity, given some domain spe-

[10] Apart from the small latency of signal processing units, these have the same duration as the recording event itself.

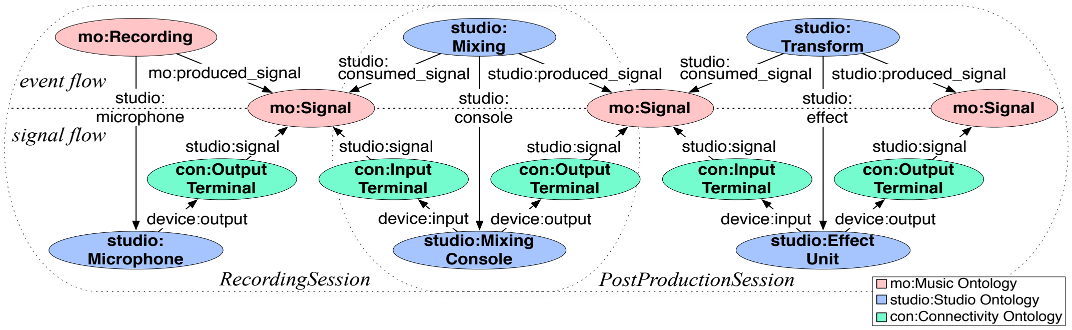

Figure 3. Recording, mixing and transformation events with an associated signal flow

cific details provided by the modeller. In this section we describe some extensions of the Studio Ontology.

5.3.1 Audio Recording

The Microphone Ontology includes a small taxonomy of microphones organised by their transducer principle (i.e. *CondenserMicrophone, RibbonMicrophone* etc...). It also allows for describing most properties one may find in a microphone data sheet, for instance *diaphragm type* and *size* or *polar pattern*. The *Configuration* concept (subclass of `device:State`) can be used to describe variable parameters of microphones such as sensitivity, or variable polar pattern setting, in a particular recording event. The ontology includes the concept *MicrophoneArrangement* and allows for describing stereo and spatial recording techniques, such as a *BlumleinPair* or *DeccaTree*, with their constituent microphones, and their distances, angles and configurations.

5.3.2 Audio Mixing

The Audio Mixer Ontology allows detailed description of mixing consoles both in terms of static characteristics and particular settings (such as channel strip configuration) in an event. The ontology is modelled after a generalised blueprint of mixing consoles obtained from studying several commercial hardware designs, however software implementations were also taken into account. It defines concepts such as *Channel, Bus* or *InsertTerminal* and properties to describe fader levels, panning, equalisation (linked to an Audio Effect Ontology) and routing in a particular event, including automation.

5.3.3 Audio Effects

The core ontology includes concepts to refer to audio effect units and plugins that are particular hardware or software devices, and a small taxonomy of audio effects based on their typical applications in audio engineering. However, audio effects are best conceptualised as physical phenomena, separated from implementation (circuit designs or algorithms), concrete devices, and their applications to signals. Therefore, we have four conceptual layers which include the concepts: *AudioEffect*, *Model*, *Implementation*, *EffectDevice*, *Transform*. The Studio Ontology sets the problem of implementation details aside. Creating an Audio Effects Ontology based on multidisciplinary classification [24] is ongoing work in our lab.

5.3.4 Audio Editing

Modern digital audio workstations organise recording projects into a set of tracks — which may correspond to input channels or created in an ad hoc way — and potentially overlapping clips contained in them corresponding to various takes during a recording session. The Multitrack Ontology relates the the hierarchy of *Clip*s and *Track*s to other concepts in the Music and Studio ontologies. It defines terms such as *MultitrackProject*, *MediaTrack*, *AudioTrack*, and *AudioClip* [6].

A small Edit Ontology provides for describing a succession of edit decisions modelled as events linked to the universal timeline using `event:time` and the audio signal timeline using `edit:media_time`. These ontologies may be subsumed to describe operations in a specific tool such as a multitrack audio editor.

6. APPLICATIONS AND IMPLEMENTATION

The ability to provide machine-processable representations of the information one may find on web pages of recording studios is a contribution to the Semantic Web in itself. It facilitates finding studios with specific equipment or personnel using complex queries. However, a more significant benefit comes with the ability to denote how a piece of music was produced. We can argue that contributions form the producer or the sound engineer are just as important in modern music as composition, but we had no way to record his/her actions and choices with the transparency music is denoted using scores. Collecting these data in production is a significant effort, however a lot can be done automatically if ontology based models are available in digital mixing consoles and post production tools. The Meta Object Facility Specification [11] enables source code generation from conceptual models. To take the continuously evolving nature of ontologies into account, we provide an alternative using run-time model generation [6].

7. CONCLUSIONS AND FUTURE WORK

We presented a novel conceptualisation of the recording studio environment and its implementation as a Semantic Web ontology. Our framework is unique in itself, therefore we have no grounds for direct comparison, but we evaluated it against a music production text corpus, and found that it has good lexical coverage, and represents approximately 75% of commonly occurring production situations. Further extensions remain future work, as well as audio editor prototypes which enable automatic collection of production information, and provide easy to use data entry facilities for capturing data external to a computer system. However, to achieve its full potential, our system should be included in digital music production tools.

8. REFERENCES

[1] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura. Comm: Designing a well-founded multimedia ontology for the web. In *In Proceedings of the 6th International Semantic Web Conference (ISWC'2007), Busan, Korea*, pages 11–15, 2007.

[2] A. Bandara, T. R. Payne, D. de Roure, and G. Clemo. An ontological framework for semantic description of devices. *in Proc. International Semantic Web Conference (ISWC), Hiroshima, Japan*, 2004.

[3] D. Beenham, P. Schmidt, and G. Sylvester-Bradley. XML based dictionaries for MXF/AAF applications. Technical report, Sony Broadcast and Professional Research Laboratories, UK, 2000.

[4] J. Bullock and H. Frisk. libintegra: A system for software-independent multimedia module description and storage. *in Proceedings of the International Computer Music Conference, Copenhagen, Denmark*, 2007.

[5] P. Dolog. *Model-Driven Navigation Design For Semantic Web Applications with the UML-Guide*. in Maristella Matera and Sara Comai (Eds.) Engineering Advanced Web Applications, Rinton Press., 2005.

[6] G. Fazekas and M. Sandler. Novel methods in information management for advanced audio workflows. *In proceedings of 12th International Conference on Digital Audio Effects, Como, Italy*, 2009.

[7] FIPA. Device ontology specification. Foundation for Intelligent Physical Agents, Working Spec., 2002.

[8] R. García and O. Celma. Semantic integration and retrieval of multimedia metadata. *Proceedings of the 5th International Workshop on Knowledge Markup and Semantic Annotation*, 2005.

[11] http://www.omg.org/mof/

[9] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43:907–928, 1993.

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

[11] J. Hunter. Enhancing the semantic interoperability of multimedia through a core ontology. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):49–58, 2003.

[12] K. Jacobson, Y. Raimond, and M. Sandler. An ecosystem for transparent music similarity in an open world. *in Proc. 10th International Society for Music Information Retrieval Conference (ISMIR 2009), Kobe, Japan*, 2009.

[13] M. Kanzaki. Music Vocabulary. Available Online: `http://www.kanzaki.com/ns/music`, 2007.

[14] M. Klein, D. Fensel, F. van Harmelen, and I. Horrocks. The relation between ontologies and XML schemas. *Linköping Electronic Articles in Compiter and Information Science*, 2001.

[15] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. Wonderweb deliverable d18: Ontology library. Technical report, Laboratory For Applied Ontology - ISTC-CNR, 2003.

[16] C. McKay, J. A. Burgoyne, J. Thompson, and I. Fujinaga. Using ACE XML 2.0 to store and share feature, instance and class data for musical classification. *Proceedings of the International Society for Music Information Retrieval Conference*, (303-8.), 2009.

[17] F. Nack, J. van Ossenbruggen, and L. Hardman. That obscure object of desire: multimedia metadata on the web, part 2. *IEEE Multimedia*, 12(1):54–63, 2005.

[18] M.-F. Plassard, editor. *Functional Requirements for Bibliographic Records, Final Report*. International Federation of Library Associations and Institutions, 1998.

[19] Y. Raimond. *A Distributed Music Information System*. PhD Thesis, Queen Mary University, School of Electronic Engineering and Computer Science, 2008.

[20] Y. Raimond, S. Abdallah, M. Sandler, and G. Frederick. The music ontology. *in Proc. 7th International Conference on Music Information Retrieval (ISMIR 2007), Vienna, Austria*, 2007.

[21] J. R. Smith and P. Schirling. Metadata standards roundup. *IEEE Multimedia, April-June 2006*, Vol. 13.(No. 2.):pp. 84–88, 2006.

[22] B. Smithers. Going DAW to DAW. *Electronic Musician*, October 2007.

[23] P. F. Strawson. *Individuals. An Essay in Descriptive Metaphysics*. Routledge, London and New York, 1959.

[24] 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*, 2006.