

A semantic web annotation tool for a web-based audio sequencer

Luca Restagno¹, Vincent Akkermans², Giuseppe Rizzo¹, and Antonio Servetti¹

¹ Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy,
luca.restagno@studenti.polito.it, giuseppe.rizzo@polito.it,
antonio.servetti@polito.it,

² Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain,
vincent.akkermans@upf.edu

Abstract. Music and sound have a rich semantic structure which is so clear to the composer and the listener, but that remains mostly hidden to computing machinery. Nevertheless, in recent years, the introduction of software tools for music production have enabled new opportunities for migrating this knowledge from humans to machines. A new generation of these tools may exploit sound samples and semantic information coupling for the creation not only of a musical, but also of a “semantic” composition. In this paper we describe an ontology driven content annotation framework for a web-based audio editing tool. In a supervised approach, during the editing process, the graphical web interface allows the user to annotate any part of the composition with concepts from publicly available ontologies. As a test case, we developed a collaborative web-based audio sequencer that provides users with the functionality to remix the audio samples from the Freesound website and subsequently annotate them. The annotation tool can load any ontology and thus gives users the opportunity to augment the work with annotations on the structure of the composition, the musical materials, and the creator’s reasoning and intentions. We believe this approach will provide several novel ways to make not only the final audio product, but also the creative process, first class citizens of the Semantic Web.

Keywords: Ontology driven annotation tool, Semantic audio annotation, web-based audio sequencer, Semantic Web sequencer, Semantic Web

1 Introduction

One of the key focus of the Semantic Web [1] is the process of combining data from different sources in order to easily elaborate that information programmatically. The aggregation of data extends the available information about a resource. In this context, the process of tagging or annotation, performed by human beings, plays an important role in achieving a more precise definition of the content. Even though the practice of tagging content has become widespread

among many Internet user communities, it is affected by several problems including the inconsistency of terms used to annotate contents. Then, one of the challenges in this research field is to ensure consistency across annotations in the choice of terms and their meaning. A solution is to restrict the user's annotations to a certain set of concepts defined in an ontology. Generally, an ontology defines a domain description that can be used as a controlled vocabulary and that may help in the process of context enrichment and crowdsourced content classification. On the other side, an obvious disadvantage of this approach is its possible incompleteness due to missing concepts, a problem that does not exist in the free tagging approach. However, by combining several well designed ontologies users may be guided to create high quality annotations.

The ontology driven annotation framework detailed in this paper is developed to allow users to annotate sonic material in a controlled way, using concepts extracted from different ontologies. The main feature of the framework is the capability to load from the Web any ontology specified in the Web Ontology Language (OWL) [11] and to guide users through the process of content information enrichment. When an ontology is loaded, the annotation framework allows the creation of annotations that link selected parts of the content to the concepts of that ontology. In addition, it creates a RDF graph for each annotation which is published so that annotation is available to everyone. Furthermore, the annotation framework has been completely developed using Web standard languages, in such a way that it can be easily used inside other web applications. Moreover, it provides a user-friendly web front-end to make the semantic annotation an easy task. In order to accomplish this goal, the framework loads all classes and their attributes from the ontology; then classes are easily browsable through the widgets of the Web user interface. For each attribute of a class, the tool loads the right user interface widget to let the user specify a value. Furthermore, the annotation framework has been integrated in a web-based audio sequencer in order to allow annotation of audio contents, realizing a complete environment for composing and annotating sounds. Through the Web interface, users may play sequentially or may remix sounds available on the Web. The sound files are loaded into the sequencer simply by specifying their URLs. For example our audio material has been retrieved from Freesound³, a collaborative database of Creative Commons licensed sounds. Then, loading a specific ontology, users are guided in the process of annotating the gathered sounds, enriching the description of the composition created and, finally, the meaning.

Music and sound have a rich semantic structure. They communicate a message, designed by the composer or sound maker, which ranges from nonsensical or abstract to symbolic (e.g. a piece of film music supporting a clear narrative). A majority of media production nowadays is done with software tools, which give rise to various new opportunities to monitor the production process. In this work we focused on the combination of the annotation framework and the web audio sequencer, to investigate the implications of this idea. Take for example video games, which generally have a non linear narrative that is often supported

³ <http://www.freesound.org>

by affective music. As the game world can give rise to a variety of situations the music should be able to adapt. If the composer is able to formally describe pieces of his non linear composition by using our annotation framework and the right ontology, the game would be able to generate new music by matching the formalized intentions in both domains. Another example is music education, where providing insight and understanding into the music is the primary concern. A piece of music, whose structure and different aspects have been formally annotated, could be represented in different ways and thus give students views on the work of art that match their capabilities or interests. Possibly with future work the artist can, while working, develop his own ontology. This ontology would describe the themes of the work as seen by the creator and allow himself and others to reflect and learn from this.

The remainder of this paper is organized as follows: a review of the current state of the art is presented in section 2, key ideas of our approach are introduced in section 3 and the description of the annotation tool is showed in section 4. A contextualization of the tool, by means of a use case, is described in section 5, followed by conclusions and future work in section 6.

2 Related Work

The process of media content enrichment by means of user generated data is a challenging task that has been addressed by the Semantic Web community in several works. An often used approach is to give users the ability to annotate a content in order to more easily retrieve the information later. In particular, annotation is often required to refine and improve data descriptions when automatic feature extraction tools are employed. In this context, many efforts have been done to generate annotations in order to point to and, then, retrieve contents. Our work takes a different approach and combines the annotation tool with a web sequencer in order to provide the ability to remix sound files and to enrich the newly created composition with a controlled vocabulary. In the rest of this section we give an overview of the state of the art in the media content annotation tools.

In the LabelMe Project [14], Russell et al. produced a web-based image annotation tool to identify and define specific objects inside images. Their goal was to provide a dynamic dataset for object recognition and computer graphics. Although this research worked with tags, they did not focus on the annotation itself. Additionally, they used free textual tags, so annotations were affected by the typical folksonomy [4] problems, like polysemy and synonymy. In our approach we avoid these problems by using a controlled vocabulary, an ontology chosen by the user during the startup process, which includes a concept hierarchy from a specific knowledge domain.

The use of controlled vocabulary to improve semantic annotation of images has been explored in the M-OntoMat-Annotizer [12]. Petridis et al. implemented different ontologies based on the MPEG-7 standard to let users associate visual descriptors with content. This work permits selection of a part of an image or

frame and association of a concept retrieved from the provided ontology. Based on this work, we take advantage of multiple formalized ontology domains to extend the descriptive possibilities of an annotator.

In [16], Wang et al. addressed the annotation problem by means of a set of ontologies which are linked using a bridge ontology. This work overcame the problem of ontology reuse and prevented unnecessary ontology extension and integration. Although this idea is promising, it presents only a general idea of how to link ontologies without proposing a clear method. However, G. Kobilarov et al. [8], following [16], used a bridge ontology to categorize and link to DBpedia [2] multimedia documents located within the BBC archives. By means of this approach, they exploited object persistence in the BBC categorization system (CIS) and mapped resources according to DBpedia references: resource disambiguation is achieved and semantic information is augmented. Similarly to this approach, we provide the possibility to take advantage of multiple ontologies, but we do not try to link concepts between them. Our tool permits the use of concepts from a single online available ontology at a time. The ontology can, however, be switched whenever the annotator feels the need.

In order to facilitate annotation and sharing of annotations among many users, we implemented a web-based solution. An attempt to distribute annotations over the Web is represented by the Annotea Project [6], which aims to provide a system to share annotations on a general-purpose open RDF infrastructure. It suggests a possible set of technologies to implement a semantic web infrastructure for creating, editing, viewing and sharing annotations. In [15], Ronald Schroeter et al., proposed how to use the Annotea Schema to extend annotation links among multiple video resources. They used annotation to highlight parts of video and mapped them in the MPEG-7 standard. Pointing to different highlighted parts of video may make a new composition, which could be built automatically and could be reproduced by users: this is the idea behind the work of Rene Kaieser et al [7]. Although this problem is relevant for the research community, our work differs from it because here the user supervises the enrichment process. A similar use case was addressed by Maria Meleshkova et al. [9] where Web APIs are annotated by users with an annotation editor. Finally, we use some of the framework technologies, like the Annotea Annotation RDF Schema, and we developed a web-based framework for the ontology-driven annotation of audio contents.

3 Rationale

Our semantic web annotation tool is a web tool for annotating any kind of resource. It can load any online available OWL ontology and guide users through the annotation process with a simple user interface. When starting the annotation process the user is allowed to select the ontology that deals with the aspect of the resource he wants to make statements about. Additionally we assess our tool with annotations of sonic material. In this context we present an open web

framework that can improve the user experience during the creative process, by means of ontology driven annotation.

Communities, that use the free text annotation method, are affected by a set of problems, like polysemy, synonymy, data scarcity, spelling errors and plurals. Polysemous tags can return undesirable results. For example, in a music collection when a user searches for the tag *love*, results could contain both love songs and songs that were tagged as such because user liked them very much. Tag synonymy is also an interesting problem. Even though it enriches the vocabulary, it also presents inconsistencies among terms used in the annotation process. According to [10], *bass drum* sounds can be annotated with the *kick drum* tag, but these sounds will not be returned when searching for *bass drum*. To avoid this problem, sometimes users tend to add redundant tags to facilitate the retrieval (e.g. using *synth*, *synthesis* and *synthetic* for a given sound).

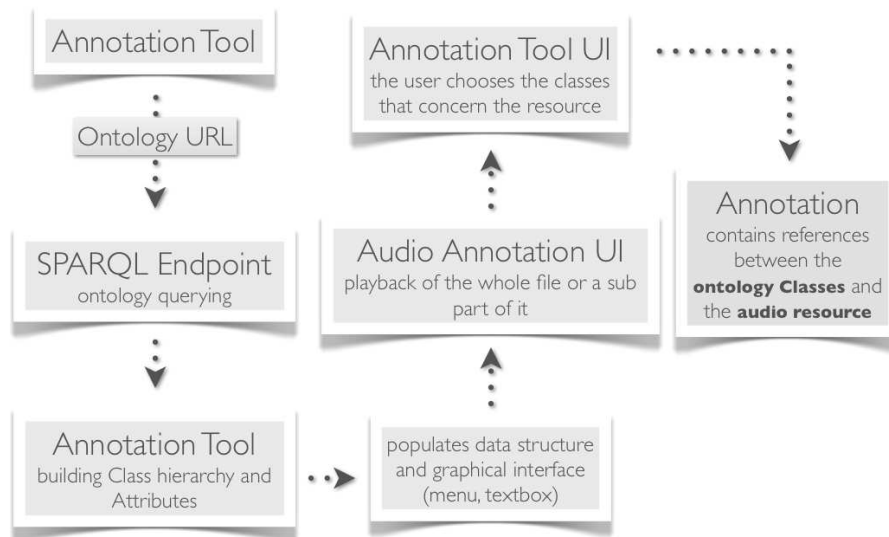


Fig. 1. The annotation tool retrieves a set of concepts from knowledge repositories (ontologies/taxonomies). Then it exposes the set of concepts to the annotator through the graphical interface. Using this front-end, the user can link semantic concepts to an audio resource whose contents were previously unknown. The result is an annotation, a document that stores the links the human annotator creates.

Figure 1 shows how our tool works. It retrieves information from knowledge repositories available on the Web. They can be formalized as ontologies, like the large and popular Music Ontology [13], an attempt to link all the information about musicians, albums and tracks together. It can exploit ontologies specifically developed for an application. Moreover, it can use Web databases that provides a query service based on the semantic web technologies, like the DBpe-

dia project that allows access to the large database of Wikipedia via semantic web resources. Pursuing this approach we wanted to conform to the Linked Data principle [5] of distributable and connected pieces of information. The user can associate with the resource any concept of the ontology, in order to extend the semantic description of the digital content.

The tool provides an intuitive Web user interface that lets users choose one of the classes in the ontology. When the user is done annotating, the annotations are converted to the RDF syntax. As example, in Figure 3 is shown their serialization by means of RDF/XML syntax. These are then sent to the server and saved in a triple store, ready to be retrieved and queried.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:an="http://www.w3.org/2000/10/annotation-ns#"
  xmlns:ext-owl="http://mtg.upf.edu/2010/02/SoundProducingEvents.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
  <an:Annotation>
    <an:annotates>
      <rdf:Description>
        <an:source rdf:resource="http://example.org/audio.mp3"/>
      </rdf:Description>
    </an:annotates>
    <an:body>
      <ext-owl:Vibrating_objects></ext-owl:Vibrating_objects>
    </an:body>
    <an:created>2010-02-22T00:04:26Z</an:created>
  </an:Annotation>
  <an:Annotation>
    <an:annotates>
      <rdf:Description>
        <an:source rdf:resource="http://example.org/audio2.mp3"/>
      </rdf:Description>
    </an:annotates>
    <an:body>
      <ext-owl:Impact>
        <ext-owl:configuration rdf:parseType="Literal"> -1 </ext-owl:configuration>
        <ext-owl:material rdf:parseType="Literal"> 2 </ext-owl:material>
        <ext-owl:size rdf:parseType="Literal"> 3 </ext-owl:size>
        <ext-owl:mallet_hardness rdf:parseType="Literal"> 4 </ext-owl:mallet_hardness>
        <ext-owl:force rdf:parseType="Literal"> 1.01034 </ext-owl:force>
      </ext-owl:Impact>
    </an:body>
    <an:created>2010-02-22T00:04:26Z</an:created>
  </an:Annotation>
</rdf:RDF>
```

Example 3 - RDF/XML annotations generated by our annotation tool.

4 Annotation tool

The annotation tool consists of a client side and a server side component.

4.1 Client-side component

The client-side component is a graphical user interface consisting of boxes, menus and input fields to let the user navigate the classes provided by the ontology. It also allows to choose one or more classes, and specifying the value for the attributes of a class, if present. According to the SoC (separation of concerns) guidelines, we developed our tool using HTML for page markup, CSS for graphical style and JavaScript to handle the program logic and user interactions. The jQuery framework⁴ was used to manipulate the Document Object Model (DOM) and the jQuery UI⁵ utilized for the GUI components like autocomplete, datepickers and complex behaviour handlers like draggable and droppable. The tool has been developed with attention to modular programming. In order to allow other developers to reuse the code, our annotation tool was divided into three reusable modules: owl.js, owl-ui.js and owl-ui.audio.js.

- owl.js: requests an interpretation of a specified ontology from the server side component and converts this to an internal data model.
- owl-ui.js: is responsible for the creation of the annotation tool panel, composed of menus and dynamic textboxes. It requires the owl.js library to populate the user interface widgets with the information retrieved from the ontology.
- owl-ui.audio.js: creates an interface to annotate audio files. It allows the user to listen to files and, using the audio waveform image, to select a sub part of a sound in order to annotate it. Then, it allows opening of the annotation tool panel generated by the owl-ui.js library in order to annotate a sound with ontology classes.

These libraries can be embedded into any web page, making it particularly easy for a developer to add the annotation feature to his own web application. Furthermore, it would be relatively easy to develop special user interface components for annotating other types of documents, like video or text.

4.2 Server-side component

The second part of our tool is the server-side component. It is a SPARQL Protocol and RDF Query Language (SPARQL) endpoint which makes queries over the ontology and retrieves all classes, properties and attributes. The response is generated in JavaScript Object Notation (JSON) format (but it is possible to request different output formats, like raw text and XML) and it is returned to the client side. Example 4.2 provides a sample response. JSON notation is used because it is a lightweight data-interchange format, it is readable by humans and can be easily converted from text to JavaScript object. The SPARQL endpoint runs on a Linux machine with the Apache 2 web server running and the PHP language available. Furthermore, the endpoint uses the Redland RDF libraries to interpret the data from the ontologies and they are a key component of our framework.

⁴ <http://jquery.com/>

⁵ <http://jqueryui.com/>

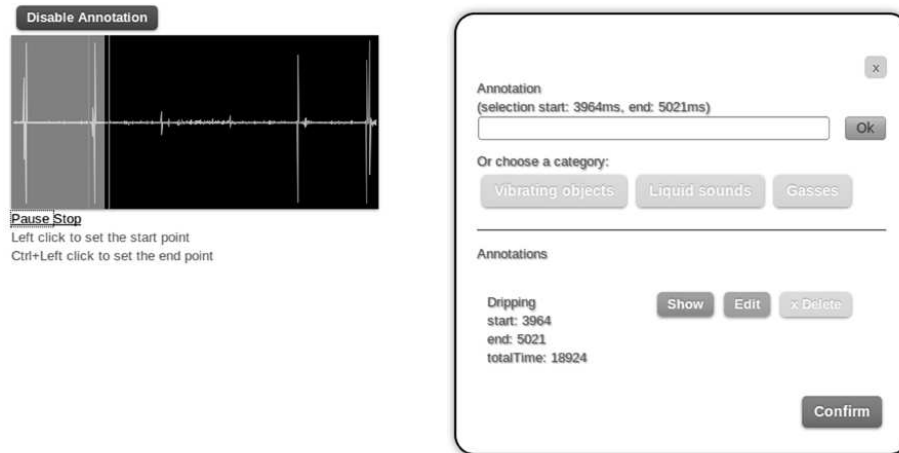


Fig. 2. An example of audio annotation using the annotation tool. On the left the waveform of a sound retrieved from Freesound, reproducing a dripping faucet. On the right the annotation tool front-end where the user is linking the Dripping category of Liquid sounds, to a temporal interval of the sound. The categories are provided by the Sound Producing Events Ontology that is loaded from the Web.

```

{
  "head": {
    "vars": [ "parentClass", "subClass" ]
  },
  "results": {
    "ordered" : false,
    "distinct" : false,
    "bindings" : [
      {
        "parentClass" : { "type": "uri", "value": "http://www.example.org/ChordTaxonomy.owl#Triad" },
        "subClass" : { "type": "uri", "value": "http://www.example.org/ChordTaxonomy.owl#Minor" }
      },
      ...
    ]
  }
}

```

Example 4.2 - JSON serialization of a SPARQL query result. This query returns each class of the ontology and a sub class of it.

4.3 Annotation process details

Figure 1 shows the annotation tool flow chart. When our annotation tool is initialized, it makes a synchronous call to the SPARQL Endpoint hosted by a server machine and it sends three main parameters: the URL of the ontology to query, the SPARQL query to execute and the format of the response.

The annotation tool receives a response, by default a JSON object, containing each class and subclass of the ontology. Processing this data our tool creates

a JavaScript structure of objects containing the complete ontology hierarchy of classes. At this point the annotation tool creates the user interface populated with data retrieved from the ontology. The resulting GUI widget includes a textbox, in which dynamic suggestions are provided by means of the autocomplete feature. The user can traverse the class hierarchy through a tree menu to choose a concept related to the resource he is annotating. When the user selects a class from the menu or from the textbox he is presented with a new widget where the user can assign a value to each attribute of the class. The tool chooses the right widget for each possible attribute type.

The annotations are collected into a stack and when the annotation process is completed, the user confirms the annotation. The tool generates an RDF representation of the annotations and subsequently sends it to a server where it could be stored in a triple store and retrieved later. Thanks to namespaces and URIs that identify uniquely a resource, the generated RDF/XML annotation holds the complete semantic description and the information the user has associated with the resource.

4.4 An example of audio annotation

In this section we illustrate the widgets developed to allow users to annotate audio resources. In this specific case, we retrieve a dripping faucet sound and its waveform image from the Freesound repository.

As shown in Figure 2, a user may playback the entire sound and select a part of it by clicking on the audio’s waveform. This way the annotator can link multiple different concepts to the same sound, or even to particular events that occur within the sound recording. In this case we loaded the Sound Producing Events Ontology, based on the work of W. W. Gaver [3]. In his book “What in the world do we hear? an ecological approach to auditory event perception” he proposed a framework for describing sound in terms of audible source attributes. We formalized a possible ontology based on the work of Gaver, using the Web Ontology Language (OWL) and published it on the Web in the form of an RDF graph. The ontology used can be easily substituted by the developer, specifying the URL of another ontology, so that classes coming from the new repository can be available to the widgets, ready to be linked to a digital resource.

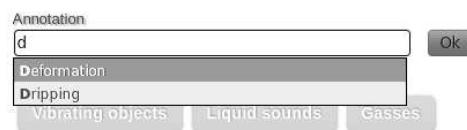


Fig. 3. The annotation tool offers intuitive widgets to find the right ontology concepts. The image illustrates a text field with an autocomplete feature.

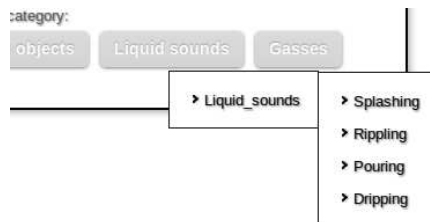


Fig. 4. The annotation tool allows navigation through the class hierarchy with a dynamic interactive menu.

The annotation tool front-end is composed of a text field with autocompletion (as shown in Figure 3), so that by typing a user receives suggestions on the available concepts. Alternatively, a user can traverse the complete concept hierarchy through a tree menu (as shown in Figure 4) which illustrates the concept relationships and that appears by clicking on the root concepts (in this case Vibrating objects, Liquid sounds and Gasses).

On the lower part of the user interface there is a stack of concepts the user already linked to the resource. It allows editing of attributes of the concept, showing annotated sound parts and in addition permits to delete an annotation previously created. Clicking on the Confirm button, the annotation tool generates an RDF/XML GRAPH which stores all the links and the information between the OWL classes and resources. This graph can be easily stored and retrieved later.

5 Use case: a web-based audio sequencer

In order to test the capabilities of our annotation tool in terms of usability and reliability, we developed a web-based audio sequencer for the Web, where users can work with sounds, mixing and annotating them in a production environment. The tool is available at the test project web page⁶. This site implements the annotation tool technology described above and it works as a hub, because it refers to sound files hosted by the Freesound website.

We chose to realize a web-based audio sequencer completely developed with standard web languages. We used the standard mark-up language designed for the Web, HTML, graphical customization allowed by CSS stylesheets and we handled the business logic and user interactions with JavaScript. We also tried to exploit the multimedia capabilities of the new version of the HTML standard, but our project required advanced audio synchronization features that HTML5 Audio does not yet provide. We had to fall back to Adobe Flash technology that is responsible for handling audio playback. Figure 6 shows three different layers of our application:

⁶ <http://pittore.polito.it:8080/wcs/>

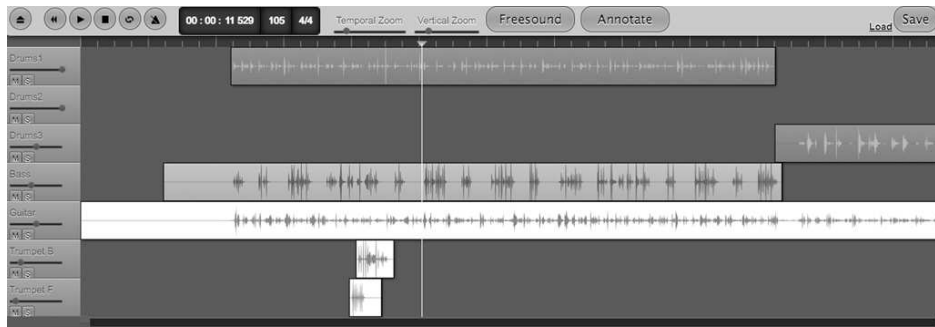


Fig. 5. The web sequencer user interface is similar to a professional audio editing program. It permits synchronization of tracks by dragging them on the grid, controlling the audio playback of the composition and zooming the view. It also implements searching of the Freesound large database of sounds. Thanks to the integration of the annotation tool, it is possible to describe each sound event of the composition with accuracy.

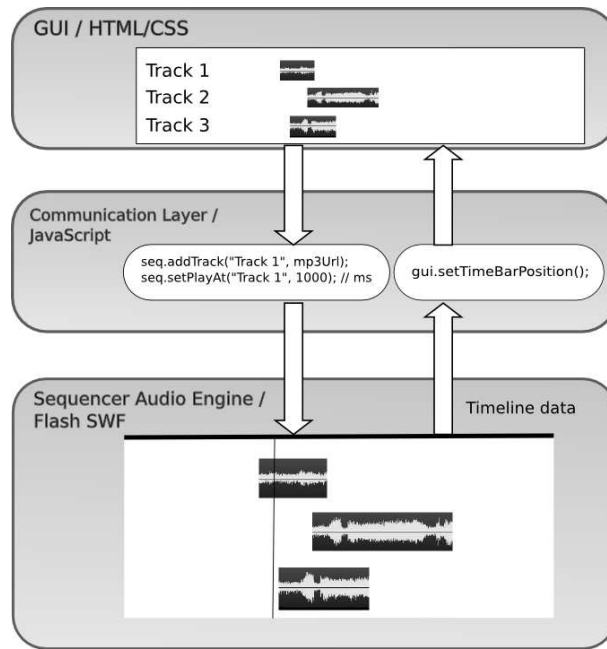


Fig. 6. The architecture of the web sequencer is composed of three layers: the Audio Engine, the Communication Layer and the Graphical Interface.

- The Audio Engine Layer is responsible for the playback of the audio. It retrieves audio files from the Web, then it synchronizes tracks and handles the virtual timeline. In addition, it has features to mute and solo a track and change its volume. It also permits looping a section of the composition and includes a metronome. It communicates with the Communication Layer described below.
- The Communication layer controls bidirectionally the Audio Engine Layer and the Graphical Interface of the application. When a user performs an action, it is handled by the Communication Layer that transmits the instructions to the Audio Engine Layer. It also receives events from the Audio Engine Layer and updates the User Interface.
- The Graphical User Interface handles all interactions with the user through drag functionalities, buttons, sliders and editable text fields. When a user interacts with the GUI, the Communication Layer propagates the action to the Audio Engine.

Through this application users can mix sounds available over the web simply using the URL of the audio resources. It implements the basic functionalities of every sequencer, like audio playback, visual tracks synchronization and looping, so that users can create their own audio composition. We also integrated searching of the Creative Commons licensed sound repository Freesound, so that users can retrieve sounds from a large repository.

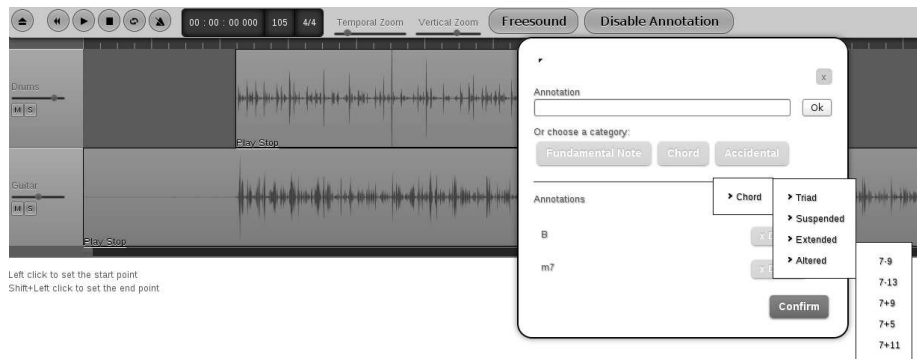


Fig. 7. Integration between the Web audio sequencer and the annotation tool. In this case the user loaded the Chord Taxonomy, in order to describe the harmony of the musical samples included in the composition.

As shown in Figure 7 the annotation tool has been integrated into this sequencer. A user can use it to give the semantic description of an audio file content. By clicking on the Annotation button the user has the opportunity to choose an argument: every argument corresponds to an existing ontology. The current implementation proposes the Sound Producing Events ontology recommended for natural sounds and the taxonomy on Chords, useful to describe the

harmony of a music track. Adding new ontologies is really simple for a developer, due to the fact that every ontology is an OWL file located over the Web. After this choice, the GUI is enriched by some new buttons and text. At this point the user can select a portion of an audio sample from the sequencer composition and choose to annotate it. Then, from the annotation tool panel that appears on the screen, he or she can listen to portions of the samples and select concepts that better describe the content (Figure 8). In this way we improve the granularity of the annotation, enriching the semantic description of an audio resource.

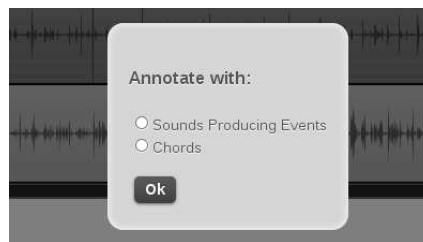


Fig. 8. In our use case, we propose two ontologies. The first is useful in annotating natural sounds, while the second is useful in describing the harmony aspect of a music track.

The technologies used permit easy integration of the annotation tool on every website. What is needed is to include the Javascript libraries into the site code. Furthermore, the possibility to plug in any ontology available on the web makes the tool a possibly useful instrument for web sites that want to include annotation functionality.

6 Conclusions and future work

In recent years, the introduction of software tools for music production have enabled new opportunities for migrating this knowledge from humans to machines. A new generation of these tools may exploit sound samples and semantic information aggregation for the creation not only of a musical, but also of a “semantic” composition. In this paper we have presented a semantic annotation tool integrated with a web-based audio sequencer. A novel approach is used to manipulate multiple audio contents in a dynamic way through a web front-end by simply pointing to their semantic references. The value of the annotation tool lies in the fact that it can load any OWL ontology and guide the end user in the annotation process. Using an easy-to-use graphical interface, it permits the annotator to make links among formalized concepts and resources not described yet, in order to increase the global knowledge of the contents on the web. Furthermore, we realized a collaborative web-based audio sequencer to test our tool. With it, users can remix sounds from the Freesound website and annotate them.

The project has followed web standards and principles with the goal of making music compositions part of the Semantic Web.

Future plans consist in extending the tool by developing new user interfaces to annotate other types of resources. We focused on the annotation of sound and music, but it is possible to easily implement visual interfaces for the annotation of video, images and text documents. The tool has been developed to be modular, so that it is not necessary to modify the core libraries. Furthermore we would like to improve the implementation of the web front-end in order to make it completely cross-browser and cross-platform. An evaluation of this tool in a user community may help to draw user trends, performing a classification of interactions. In addition, some work will be devoted to integrate an automatic tagging process, by means of a text mining algorithm with the annotation supervised by a human.

Acknowledgment

This project was conducted at the Music Technology Group of the Universitat Pompeu Fabra in Barcelona, and in collaboration with the Dipartimento di Automatica e Informatica of the Politecnico di Torino.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* pp. 34–43 (May 2001)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - a crystallization point for the web of data. *Web Semant.* 7, 154–165 (September 2009)
3. Gaver, W.W.: What in the world do we hear? an ecological approach to auditory event perception. *Ecological Psychology* 5, 1–29 (1993)
4. Halpin, H., Robu, V., Shepherd, H.: The complex dynamics of collaborative tagging. In: *Proceedings of the 16th international conference on World Wide Web*. pp. 211–220. WWW '07, ACM, New York, NY, USA (2007)
5. Hausenblas, M.: Exploiting linked data to build web applications. *IEEE Internet Computing* 13, 68–73 (2009)
6. Kahan, J., Koivunen, M.R.: Annotea: an open rdf infrastructure for shared web annotations. In: *Proceedings of the 10th international conference on World Wide Web*. pp. 623–632. WWW '01, ACM, New York, NY, USA (2001)
7. Kaiser, R., Hausenblas, M., Umgeher, M.: Metadata-driven interactive web video assembly. *Multimedia Tools and Applications* 41, 437–467 (2009)
8. Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., Lee, R.: Media meets semantic web — how the bbc uses dbpedia and linked data to make connections. In: *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*. pp. 723–737. ESWC 2009 Heraklion, Springer-Verlag, Berlin, Heidelberg (2009)
9. Maleshkova, M., Pedrinaci, C., Domingue, J.: Semantic annotation of web apis with sweet. In: *6th Workshop on Scripting and Development for the Semantic Web, colocated with ESWC 2010* (2010)

10. Martnez, E., Celma, O., Sordo, M., De Jong, B., Serra, X.: Extending the folksonomies of freesound.org using content-based audio analysis. In: Sound and Music Computing Conference. Porto, Portugal (23/07/2009 2009)
11. Motik, B., Patel-Schneider, P.F., Parsia, B.: Owl 2 web ontology language: Structural specification and functional-style syntax (2009), <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/>
12. Petridis, K., Anastasopoulos, D., Saathoff, C., Timmermann, N., Kompatsiaris, Y., Staab, S.: M-ontomat-annotizer: Image annotation linking ontologies and multimedia low-level features. In: Gabrys, B., Howlett, R., Jain, L. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, vol. 4253, pp. 633–640. Springer Berlin / Heidelberg (2006)
13. Raimond, Y., Sutton, C., Sandler, M.: Interlinking music-related data on the web. *IEEE Multimedia* 16, 52–63 (2009)
14. Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision* 77, 157–173 (2008)
15. Schroeter, R., Hunter, J., Newman, A.: Annotating relationships between multiple mixed-media digital objects by extending annotea. In: Franconi, E., Kifer, M., May, W. (eds.) *The Semantic Web: Research and Applications*, Lecture Notes in Computer Science, vol. 4519, pp. 533–548. Springer Berlin / Heidelberg (2007)
16. Wang, P., Xu, B.W., Lu, J.J., Kang, D.Z., Li, Y.H.: A novel approach to semantic annotation based on multi-ontologies. In: *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*. vol. 3, pp. 1452 – 1457 (August 2004)