RDF in the Semantic Hifi European project

Max Jacob
Institut de Recherche et Coordination Acoustique Musique
Place Igor Stravinsky 1, 75004 Paris, France
Max.Jacob@ircam.fr

Abstract

Meta-data related to musical content have become a very popular research issue over the past years. The affirmation of ID3 embedded meta-data in mp3 files and the release of the Mpeg7 standard are only the surface of a phenomenon that is attracting more and more attention in the research labs of many business oriented companies and universities.

The main reason for this interest is that music has just recently, but very rapidly, become a real electronic resource, that is managed through a computer and can be downloaded over the Internet. The consequences of this fact are enormous. First of all, there is a clear need to manage large sets of music files on a hard disk, which is of course quite different from managing them on a bookcase. This issue has been addressed quite successfully by several popular softwares, which allow you to index music files by author, title and genre. Another need is to find your way through the very large offerings of musical content that you can access over the Internet and that is getting more and more indistinct, as it has already happened for other kinds of media: it may not be hard to find something you are looking for, but the discovery of interesting content of which you are not yet aware, might really be difficult.

Besides these needs, there are also new possibilities disclosed by this scenario. First of all, the simple title, author and genre paradigm can be enhanced in order to provide more interesting heuristics and services relying on them. Secondly, music can easily be manipulated and listened to in a much more interactive way than on a CD player.

All these aspects are targeted by the Semantic Hifi European project, that tries to coordinate scientific research, social investigation, musical knowledge and information science in order to propose new concepts for browsing, searching and listening to musical content. This agenda is really challenging, also because things get really complicated once you start investigating what actually are meta-data related to musical content. Soon you will realize that, besides a very small set of well defined concepts, most of the important parameters that people use to classify music strongly depend on the cultural context or even on subjective matters. And this is not just due to a lack in standardization. There really does not seem to exist a right way to describe music, nor can such a way be invented and imposed on people, if we want it to be useful for their needs.

On the other hand, it would make little sense if everybody used his own criteria, because there are for sure many superpositions among different people that can be exploited.

For these reasons, we need a description logic that is as flexible as possible, but that still guarantees the maximum amount of interoperability at any degree of customization. This is where RDF/OWL comes in to play.

In this discussion we will give an overview on the Semantic Hifi project and examine a set of use cases that outline the role of the Semantic Web standards in our implementation plans.

1. Overview of Semantic HiFi

1.1 The Semantic HiFi network

Semantic Hifi aims to be a system providing following services to its users:

- find information about the music they own
- find interesting music
- browse among music pieces following interesting relations
- interact with the music
- produce music and make it available to others

To provide all this we plan to build a peer to peer system allowing users to share music and meta data. Users might be common people at home, music labels, amateur musicians that want to share their music, research projects that exploit and/or contribute to the information on the netwiork, and probably many others. This variety of users has to access to

the Semantic HiFi network with different specialized applications. The Semantic Hifi project will provide a framework that makes all these applications interoperable and plans to implement a couple them.

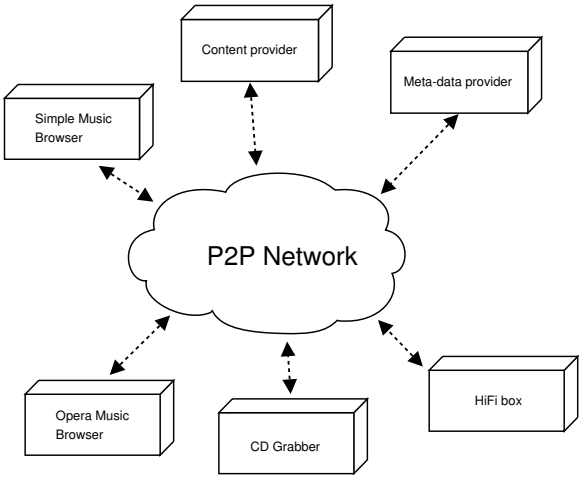

Figure 1. The Semantic HiFi network

1.2 The Semantic HiFi framework

The Semantic HiFi framework is a set of software libraries, RDF/OWL based description schemes, specifications and guidelines. Figure 2 shows the architecture of a typical Semantic HiFi application.

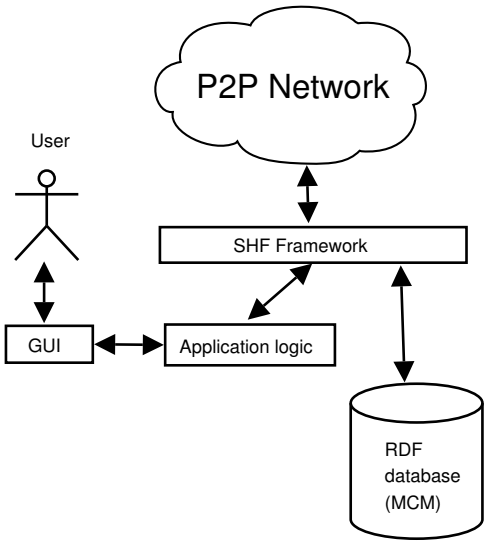

Figure 2. Semantic HiFi application

1.2.1 The database

Many SemanticHiFi applications will have a meta-data repository. Applications are encouraged to share these meta-data, exposing them as RDF graphs that can be queried against (a maybe extended version of) SquishQL [5]. We will provide an implementation of such a database, probably relying on the Sony MCM [11] library. This database should allow to:

- Execute SquishQL queries providing some OWL inference
- Import OWL ontologies
- Import RDF graphs
- Manage the data through an easy API

1.2.2 The query language

The SquishQL [5] query language (maybe with some extension) is the shared language of the Semantic HiFi applications. Semantic HiFi will provide a software layer based on the free JXTA [12] peer to peer library that allows to send queries all over the network or to a single peer and collect the results.

1.2.3 Ontology definition extensions

One more feature we are studying about is a standard way to embed function definitions in an OWL ontology. These functions should then be visible to the query language. Also this feature should be available in our database implementation. In any case we will provide some built-in function in order to perform XPaths [6] on `rdf:XMLLiterals` [3] and to support XMLSchema types [1].

Another possible feature is a way be to embed *inference logics* in an OWL ontology.

1.2.4 File identification

Semantic HiFi defines a standard way to identify files through a hash function. This function provides a possibly unique identifier based on the byte content of a file. In RDF the relation between a resource and its byte hash is represented with the `shf:hash` property.

We also provide a standard way to identify musical content, adopting Fraunhofer Institut's *audio fingerprint* [7]. An audio fingerprint is based on the musical content rather than on the bytes in a file, and this makes it more robust identifying a music piece independently on the file type, audio encoding, amplitude and silence header. The relation between a musical resource and its audio finger print is expressed by the `shf:afp` attribute.

1.2.5 File sharing

We provide also a way applications can share files without having too much trouble with IPR. An encrypted licence produced by a Semantic HiFi software module can be applied to a file, and this will make the file shareable over the peer to peer network using its binary hash. This allows to share things like playing lists, mix files, SDIF files and so on. The solution is not very satisfying, since it requires applications to be *trusted*, but for the moment we have no better one.

Another option we plan to provide is to assign an HTTP location to a resource by a standard RDF property. This implies that the resource owner has (probably) a static IP address and can more or less be tracked if he shares some content he is not allowed to. By this mean, it is relatively simple for a user to share his own music freely. The HTTP address might point directly to the resource or to an HTML page that makes it somehow available. This second option allows content sellers, for example, to set up an HTML based paying system.

A Semantic HiFi application can download the resource if the HTTP address points to it directly, else open a web browser in order to display the HTML content of the addressed page.

1.2.6 Specifications

We also will define some description schemes for music meta-data. We will use existing standards, like Dublic Core, vCard and Mpeg7, or parts of them as much as possible. Besides the standards, we will probably add several ontology elements modeling the core concepts of the musical world that are not defined by those standards. We will propose RDF bindings for the standards we want to use that do not already define such a binding.

2. Uses of RDF

All features targeted by Semantic HiFi rely, at least partially, on RDF. In this section we will see some pertinent use case for each.

2.1 Find information about music

An application that includes CD grabbing, might want to find information about the grabbed tracks. The application can extract the *audio fingerprint* of the track and use it to query the Semantic HiFi network for meta-data. If for example the application wants to know the title of a resource, it can send out the following query:

```
select
  ?title
where
  ( ?track shf:afp "..." ),
  ( ?track dc:title ?title )
```

This example also shows the importance of using the same description logic where ever possible. If all applications use the generic Dublin Core *title* element to store the relation between a music track and its title, they will be able to interoperate with our CD grabber (in both ways).

If our application is thought for an opera fan, it might ask for many more meta-data, like the title and the author of the opera, the singers and so on. This will for sure involve some exotic description elements that were thought for opera fans, and only applications managing these elements will be able to answer.

If the application gets contraddictory information, the user might choose among the most frequently answered alternatives. If an information could not be retrieved, the user might fill it up manually. In any case, all the retrieved, chosen and manually written meta-data, together with the audio fingerprint will be reflected to the local database. This allows the user to access them later. If the application shares the meta-data, the whole network will benefit on the newly entered information. Note that there is another benefit coming from the chosen alternatives among mismatching results: if most users choose the correct alternative, this alternative will become predominant.

Another way an application has to retrieve information about a music track, is to have built-in algorithms that extract automatically information from the audio file. In some way the *audio fingerprint* is an example of an automatically extracted descriptor. In Semantic HiFi we will develop also other descriptor extraction algorithms and integrate them in our applications. These should include tempo, meter, key, energy, sung / instrumental, rythm complexity, beat segmentation and others.

Since some of these extractors are quite heavy, the sharing of their results is in some sense a way to share computing power.

2.2 Find interesting music

The most simple way to find music is querying the system about music pieces having some property set to some value. For example a DJ might want to look for instrumental pieces being of genre *disco*, having a high energy and a tempo around 120 bpm. This can be asked with a query similar to the following:

```
select
  ?title, ?loc
where
  ( ?track ex:tempo ?t ),
  ( ?track ex:energy ?e ),
```

```
  ( ?track ex:genre ex:disco ),
  ( ?track shf:httpLocation ?loc ),
  ( ?track dc:title ?title )
and
  ?t > 118 and ?t < 122 and ?e > 0.8
```

Another way to find new music is looking for track that are similar to a given one. An application can perform this, searching for tracks that have similar values on some descriptors. The process can be entirely automatic if the descriptors of the target track are extracted automatically or retrieved over the network. In the same DJ example, the DJ might just ask for some tracks that are similar to a given one, and the system can do the rest by its own.

There might also be some applications detecting similarity between music pieces on the basis of some statistical analysis and sharing the results in RDF. For example we thought about an application collecting all the available play-lists on the network and looking for pieces that often occur together. This application might output directly similarity factors and share them on the network. We should provide a standard way to do this, as for example by sub-classing a standard Semantic HiFi property that means roughly *similar to*. This would allow applications to find just *similar* pieces on the network, without any further specification. Other applications might look for all available similarity criteria, and reflect them dynamically on the user interface.

But the most common way to find interesting music in currently available peer to peer systems is to browse the repository of other users that seem to have similar taste or interests. This feature is also included in Semantic HiFi, since an application can address a peer directly.

2.3 Browse among music pieces following interesting relations

The similarity seen in the preceeding paragraph, is already a kind of *browsing* feature. More specific browsing features can be implemented by specific applications.

For example an application conceived for an opera fan, might implement a function that retrieves all existing editions of a given opera. The query to do this, might look like the following:

```
select
  ?director, ?date
where
  ( ?edition ex:operaIs ?opera ),
  ( ?opera dc:title "Rigoletto" ),
  ( ?edition ex:directedBy ?director ),
  ( ?edition dc:date ?date )
```

After this, the application could provide a way to retrieve the titles and HTTP locations of all tracks of a particular edition, or list all singers and roles and so on.

With these kind of functionalities, a user can browse through his local database and on virtually all the information available on the Semantic HiFi network.

2.4 Interaction with the music

For this example we will consider a score following module that is developped by IRCAM real-time team [10]. This module takes a prepared score file as input and can then follow a singer playing the music in time with his singing.

First we can think on an application that hepls editing these score files. Such an application might use the Semantic HiFi network in order to find a MIDI file and lyrics of a given song. Once the score file is prepared, it is stored and a relation is created in the database telling this is a score following file for that song.

This information can be used by another application that can perform the score following and play together with the user.

Another example of interaction is the creation of mix files and play lists. Mix files are files that point to some original music tracks and specifies how to mix them together. An application that wants to produce mix files, can use the Semantic HiFi network in order to find music. Play lists are files that specify a list of music pieces and some transaction algorithm between one piece and the following. As mix files, play lists compilation can be helped by the Seamantic HiFi network. Furthermore a transaction algorithm can search for the tempo or the beat segmentation in order to help rendering the transaction.

2.5 Music production and sharing

Semantic HiFi applications might have some simple music production features. For example the mix files are some kind of music production. If an application allows you to record while you sing on a followed score, this also produces a music file. Generally these kind of productions should be sharable in Semantic HiFi, since they are not supposed to be dangerous for IPR.

It is also possible to share music produced outside the Semantic HiFi applications, by setting up a web site, putting the music files on it and informing some database connected to the network. Of course Semantic HiFi applications can be built that help this process, mirroring an ftp site with some local folders, providing graphical user interfaces to describe your music and so on.

3 Final note

The Semantic HiFi project started in 2004 and lasts three years. Though the global concepts should be defined, some of the details exposed here are still object of discussion.

References

[1] *XML Schema*.
http://www.w3c.org/XML/Schema

[2] *Extensible Markup Language (XML) 1.0*.
http://www.w3.org/TR/REC-xml

[3] *Resource Definition Framework (RDF)*.
http://www.w3.org/TR/rdf-mt

[4] *OWL Web Ontology Language*.
http://www.w3.org/TR/owl-ref/

[5] *The SquishQL RDF query language*.
http://swordfish.rdfweb.org/rdfquery/squish-bnf.html

[6] *XML Path Language (XPath)*.
http://www.w3.org/TR/xpath

[7] J. Herre, *Robust Matching of Audio Material*.
http://www.tnt.uni-hannover.de/project/mpeg/audio/general/aes110_6_Matching.pdf

[8] *Dublin Core*.
http://dublincore.org/

[9] *Mpeg 7 Documents*.
http://www.tnt.uni-hannover.de/project/mpeg/audio/public/#mpeg7

[10] *IRCAM real-time team*.
http://www.ircam.fr/58.html?&L=1

[11] *Sony CSL MCM*.
http://music.csl.sony.fr/MCM/

[12] *The JXTA peer to peer library*.
http://www.jxta.org/