A Personalized Audio Web Service using MPEG-7 and MPEG-21 standards

Emmanouil Skondras[1], Malamati Louta[2], Panagiotis Sarigiannidis [2]
[1]Dpt of Informatics & Computer Technology
Technological Educational Institute of Western Macedonia, 52100 Kastoria, Greece
[2]Dpt of Informatics and Telecommunications Engineering,
University of Western Macedonia, 50100, Kozani, Greece
manolisskondras@yahoo.gr, louta@uowm.gr,
psarigiannidis@ uowm.gr

Abstract—**This paper presents a web service which delivers personalized audio information. The personalization process is automated and decentralized. The metadata which support personalization are separated in two categories: the metadata describing user preferences stored at each user and the resource adaptation metadata stored at the web service host. The multimedia models MPEG-21 and MPEG-7 are used to describe metadata information and the Web Ontology Language (OWL) to produce and manipulate ontological descriptions. SPARQL is used for querying the OWL ontologies. The MPEG Query Format (MPQF) is also used, providing a well-known framework for applying queries to the metadata and to the ontologies.**

I. Introduction

Nowadays, the volume of multimedia data is increasing rapidly in many information channels. The delivery of multimedia services is a common task. However more users tend to require information retrieval services which include high quality features such as semantic description and personalization of information. In this paper a prototype application that delivers personalized audio information to users is described.

The framework uses MPEG-7 and MPEG-21 for the description of audio content as well as the users' preferences. The metadata information is managed using Web Ontology Language (OWL) ontologies, which provide semantic descriptions of the multimedia content. It also uses SPARQL for querying the metadata and the relative ontologies. The MPQF provides a well-known framework for applying the queries.

On one hand, the information that describes the user preferences is created and stored at the client. On the other hand, the web service side information contains audio resources and resource adaptation metadata, minimizing thus, the central storage and computational requirements. This significantly reduces the response time of the audio web service, handling multiple concurrent requests from users during normal operation.

The remainder of the paper is organized as follows. In section 2, the related research literature is revisited. Section 3 presents an overview of the standards followed in this study. Section 4 describes the software architecture that supports the prototype application, as well as the software elements and modules required. Section 5 presents a case study of the application prototype. Finally, section 6 concludes our work and presents possible future extensions and plans.

II. Related Work

The rapid increase in multimedia content has challenged the academic and industrial communities into the development of information retrieval tools enhanced with personalization and adaptation capabilities. An increasing number of these applications use defined standards and well-known query formats to support personalization in the web.

In [1], the design and the implementation of an MPEG-7 based Multimedia Retrieval System for Film Heritage is presented. The multimedia content has been indexed using an Annotation Tool based on MPEG-7 standard. An MPEG-7 Compliant Ontology in OWL DL has been developed to fulfill the requirements of the system. This ontology has been instantiated so that the retrieval process can be handled. This work has been assessed during the validation of the CINeSPACE project, which aims to design and implement a mobile rich media collaborative information exchange platform, accessible through a wide variety of networks (cities WiMax, WANs etc.) for the promotion of Film Heritage.

In the work described in [2], the issues associated with designing a video personalization and summarization system in heterogeneous usage environments are addressed, providing in parallel, a tutorial that introduces MPEG-7 and MPEG-21 within these contexts. The authors introduce a framework for a three-tier summarization system (server, middleware and client). The server maintains the content sources, the MPEG-7 metadata descriptions, the MPEG-21 rights expressions and content adaptability declarations. The client exploits the MPEG-7 user preferences and the MPEG-21 usage environments, in order to retrieve and display the personalized content. The middleware contains the personalization and adaptation engines, which select, adapt, and deliver the summarized rich media content to the user. The system includes MPEG-7 annotation tools, semantic summarization engines, real-time video transcoding and composition tools, application interfaces for PDA devices as well as browser portals.

In [3] a model for integrating semantic user preference descriptions within the MPEG-7/21 standard is presented. The approach preserves the hierarchical structure of the MPEG-21/7 user preference descriptions. The implementation of the model is presented, which allows descriptions of domain ontologies, semantic content descriptions and user preference descriptions in an OWL/RDF environment and also supports automatic conversion of the proposed extensions to MPEG-21/7 descriptions.

[4] presents an adaptation model for content personalization by integrating MPEG-7/21 metadata. It uses web services as basic modules. A central web service is used which selects and monitors a suitable workflow in respect of user preferences, content semantics, network constrains as well as terminal capabilities. Each web service evaluates the MPEG-7/21 description and adapts the multimedia material. The user gets the best possible quality in respect of his terminal specifications.

[6] presents the MP7QL query language. It is a language for querying MPEG-7 descriptions and allows querying every aspect of an MPEG-7 multimedia description. Its design has taken into account the MPEG-7 Query Format Requirements. The queries utilize the user preferences as context, enabling personalized multimedia content retrieval. The MP7QL allows the specification of queries that refer to multimedia content satisfying specific criteria (such as “give me the multimedia objects where a goal is scored”), semantic entities (such as “give me the players affiliated to the soccer team Barcelona”) and constructs of domain ontologies expressed using MPEG-7 syntax (such as “give me the subclasses of the Player class”).

Finally, the work presented in [7] pays attention to the semantic retrieval for sports information in World Wide Web. The SPARQL query language is used. It realizes intelligent retrieval according to relations between sports such as “synonymy of”, “kind of” and “part of”. The process is as follows: Firstly, a sports-ontology is created. Then data are collected from data sources and annotated with the ontology. The search engine completes semantic matching of retrieval conditions through ontology reasoning for user's request and finds out the eligible data.

III. USED STANDARDS

This section makes an overview of the technologies and standards used for the development of the application prototype. These standards include Web Services [8], MPEG-7 [9], MPEG-21 [10], OWL [11], SPARQL [12] and MPQF [13].

A web service is a software component designed to support interoperable communication for processes over a network. Web services can be implemented using existing languages and platforms extending easily functionalities of applications. Moreover, they complement J2EE, CORBA and other standards allowing easy integration with existing distributed applications. A web service exploits a set of standards including the Simple Object Access Protocol (SOAP), the Web Services Description Language (WSDL) and the Universal Description Discovery and Integration protocol (UDDI).

MPEG-7 is a multimedia content description standard. The description is associated with the content itself, to allow fast and efficient searching for material that is of interest to the user. MPEG-7 is formally called Multimedia Content Description Interface. It does not deal with the actual encoding of moving pictures and audio, like MPEG-1, MPEG-2 and MPEG-4. It uses XML to store metadata such as timecoding of particular events, or synchronizing lyrics to a song.

The MPEG-21 standard aims at defining an open framework for multimedia applications. MPEG-21 uses the architectural concept of the Digital Item. A Digital Item is a combination of resources (such as videos, audio tracks, images), metadata (such as descriptors, identifiers), and structures describing the relationships between resources. Digital Items are declared using the Digital Item Declaration Language (DIDL). MPEG-21 Digital Item Adaptation (DIA) architecture and the MPEG-7 Multimedia Description Schemes (MDS) for content and service personalization provide a Usage Environment which models user preferences. The Usage Environment Description is part of the MPEG-21 DIA architecture and consists of the following description elements:

- The User Characteristics, which specify user features, including:
  - The User Info, where user information is stored.
  - The User Preferences, describing the user browsing, filtering and search preferences.
  - The Usage History, where the history of user interaction with digital items is presented.
  - The Presentation Preferences, which describe user preferences concerning the means of presentation of multimedia information.
  - The Accessibility Characteristics, responsible for content adaptation concerning users with auditory or visual impairments.
- The Terminal Capabilities, which describe the technical characteristics of user devices.
- The Natural Environment Characteristics, providing information about the location and time of a user in a particular environment, as well as audio-visual characteristics which may include noise levels and illumination properties of the natural environment.
- The Network Characteristics, which specify the network characteristics parameters including bandwidth utilization, packet delay and packet loss.

OWL is a family of knowledge representation languages used for composing ontologies. It is considered as an extention of the RDFS and its specifications have been authorized by the World Wide Web Consortium. Ontologies are described in owl documents by defining classes, properties and individuals. Classes are collection of concepts, attributes are properties of classes and individuals represent the objects of a particular class.

SPARQL is an SQL-like language developed for issuing queries to RDF and OWL repositories. Queries are expressed in triple patterns similar to RDF whereas RDF subjects, predicates and objects could be variables. Additional language features include conjunctive or disjunctive patterns as well as value filters. SPARQL components are described in three specifications. The query language specification presents the SPARQL language structures. The query results XML specification, defines the format of the results returned from SPARQL queries as XML documents. The SARQL protocol defines the framework for sending queries from clients to remote server using HTTP or SOAP messages.

The Mpeg Query Format (MPQF) defines the interaction between clients and multimedia repositories. It specifies the message format of clients' requests and multimedia services responses. In contrast to SPARQL, the MPQF doesn't specify any transfer protocol (such as HTTP) however the SOAP message model can be used. In addition to that an extensive set of error messages is defined.

IV. Software Architecture

The architecture of our framework is being presented in this section (Figure 1). It contains a main-web-service, an MPQF parser and a SPARQL parser, also implemented as web services. Additionally, it contains a client application which interacts with the web services and retrieves personalized information. The architecture is decentralized in respect to the information required to achieve personalization. User related preferences are created and stored locally at each client. Resource adaptation metadata along with the resources are the only to be composed and stored at the main-web-service. As an effect, distribution of both computational load and personalization data is achieved improving framework's scalability. The main-web-service and the client interact with MPQF parser and SPARQL parser web services for MPQF and SPARQL queries parsing, respectively.

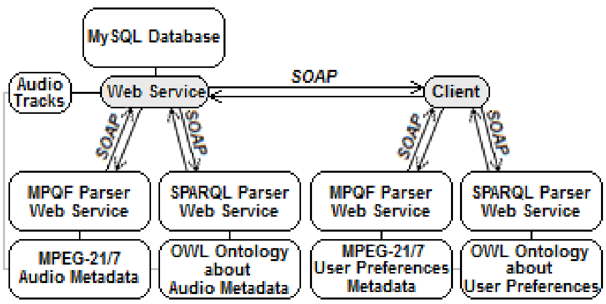

Figure 1. The basic modules of our architecture

The main-web-service contains the audio tracks and the respective audio metadata using MPEG-7 in an MPEG-21 structure. It communicates with MPQF parser and SPARQL parser web services. The audio tracks are divided in sixteen different audio categories (speech, crowd, animal, audio background effects, pop, classical, dance, electronic etc.). Audio metadata include user defined metadata (artist, producer, production year and category), technical oriented metadata (bitrate, sample rate, track duration, upload date and last download date, audio channels, audio format, file size) as well as usage history metadata (track's popularity in respect to all tracks, track's popularity in its category and recommended similar tracks). Table 1 presents a sample of the audio metadata structure.


```
<mpeg21:DIDL xmlns:mpeg21="urn:mpeg:mpeg21:2002:02-mpeg21-NS"
                    xmlns:mpeg7="http://www.mpeg.org/MPEG7/2000">
 <mpeg21:Container>
  <mpeg21:Item>
   <mpeg21:Descriptor>
    <mpeg21:Statement mpeg7:mimeType="text/plain">Metadata about  audio
     track.</mpeg21:Statement>
     </mpeg21:Descriptor>
      <mpeg21:Component>
       <mpeg21:Resource mpeg7:mimeType="application/xml">
        <mpeg7:Mpeg7>

         <mpeg7:CreationPreferences>
           <mpeg7:Title mpeg7:preferenceValue="12"
             xml:lang="en">track1.mp3</mpeg7:Title>
         </mpeg7:CreationPreferences>

         <mpeg7:CreationInformation>
           <mpeg7:Creation>
             <mpeg7:Creator>
               <mpeg7:Role
               mpeg7:href="urn:mpeg:mpeg7:cs:RoleCS:2001:AUTHOR" />
               <mpeg7:Agent xsi:type="PersonType">
                 <mpeg7:Name>
                   <mpeg7:GivenName>John</mpeg7:GivenName>
                   <mpeg7:FamilyName>Johny</mpeg7:FamilyName>
                 </mpeg7:Name>
               </mpeg7:Agent>
             </mpeg7:Creator>

             <mpeg7:Creator>
               <mpeg7:Role
               mpeg7:href="urn:mpeg:mpeg7:cs:RoleCS:2001:Publisher"/>
               <mpeg7:Agent xsi:type="PersonType">
                 <mpeg7:Name>
                   <mpeg7:GivenName>George</mpeg7:GivenName>
                   <mpeg7:FamilyName>Smith</mpeg7:FamilyName>
                 </mpeg7:Name>
               </mpeg7:Agent>
             </mpeg7:Creator>

             <mpeg7:Abstract>
               <mpeg7:FreeTextAnnotation>VeryGood
              </mpeg7:FreeTextAnnotation>
               <mpeg7:StructuredAnnotation>
                 <mpeg7:What><mpeg7:Name>Audio Track</mpeg7:Name>
                </mpeg7:What>
               </mpeg7:StructuredAnnotation>
             </mpeg7:Abstract>

             <mpeg7:CreationCoordinates>
               <mpeg7:CreationDate>
                 <mpeg7:TimePoint>2010-05-11</mpeg7:TimePoint>
                 <mpeg7:Duration>P7D</mpeg7:Duration>
               </mpeg7:CreationDate>
             </mpeg7:CreationCoordinates>
           </mpeg7:Creation>
         </mpeg7:CreationInformation>

         <mpeg7:ClassificationPreferences>
           <mpeg7:Genre mpeg7:preferenceValue="47"
             mpeg7:href="urn:mpeg:ContentCS:1">
             <mpeg7:Name xml:lang="en">Acappella</mpeg7:Name>
           </mpeg7:Genre>
         </mpeg7:ClassificationPreferences>

         <mpeg7:MediaLocator>
           <mpeg7:MediaUri>tracks/track1.mp3</mpeg7:MediaUri>
         </mpeg7:MediaLocator>
         <mpeg7:MediaTime>
           <mpeg7:MediaTimePoint>T00:00:00F100</mpeg7:MediaTimePoint>
           <mpeg7:MediaDuration>T00:13:07F100</mpeg7:MediaDuration>
         </mpeg7:MediaTime>

         <mpeg7:MediaFormat>
         <mpeg7:Content mpeg7:href="urn:mpeg:mpeg7:cs:ContentCS:2001:2">
           <mpeg7:Name xml:lang="en">audio</mpeg7:Name>
         </mpeg7:Content>
         <mpeg7:Medium
          mpeg7:href="urn:mpeg:mpeg7:cs:MediumCS:2001:2.1.1 ">
           <mpeg7:Name xml:lang="en">HD</mpeg7:Name>
         </mpeg7:Medium>
         <mpeg7:FileFormat
          mpeg7:href="urn:mpeg:mpeg7:cs:FileFormatCS:2001:3">
           <mpeg7:Name xml:lang="en">MP3</mpeg7:Name>
         </mpeg7:FileFormat>
         <mpeg7:FileSize>787082</mpeg7:FileSize>
         <mpeg7:BitRate mpeg7:minimum="N/A" mpeg7:average="8000"
          mpeg7:maximum="N/A"></mpeg7:BitRate>
         <mpeg7:AudioCoding>
```

```
            <mpeg7:Format
             mpeg7:href="urn:mpeg:mpeg7:cs:AudioCodingFormatCS:2001:1">
               <mpeg7:Name xml:lang="en">MP3</mpeg7:Name>
            </mpeg7:Format>
            <mpeg7:AudioChannels mpeg7:track="2"></mpeg7:AudioChannels>
            <mpeg7:Sample mpeg7:rate="22050" mpeg7:bitPer="0">
            </mpeg7:Sample>
           </mpeg7:AudioCoding>
          </mpeg7:MediaFormat>
         </mpeg7:Mpeg7>
        </mpeg21:Resource>
       </mpeg21:Component>
      </mpeg21:Item>
    </mpeg21:Container>
</mpeg21:DIDL>
```

Table 1. Sample of the audio metadata structure

Each client organizes its metadata using MPEG-21/7 user preferences element of MPEG-21 usage environment. The client's metadata rely on user's preferences (favorite audio categories and top 10 audio tracks in each category). Table 2 presents a sample of the user preferences metadata structure.

```
<mpeg21:DIDL xmlns:mpeg21="urn:mpeg:mpeg21:2002:02-mpeg21-NS">
<mpeg21:Container>
  <mpeg21:Item>
    <mpeg21:Descriptor>
      <mpeg21:Statement mimeType="text/plain">This item is a metadata block
                                    about John's preferences.</mpeg21:Statement>
    </mpeg21:Descriptor>
    <mpeg21:Component>
      <mpeg21:Resource mimeType="application/xml">
        <Mpeg7 xmlns= "http://www.w3.org/2000/XMLSchema-instance"
          type="complete">
          <UserPreferences>
            <UserIdentifier protected="true">
              <UserName>John</UserName>
            </UserIdentifier>
            <UsagePreferences allowAutomaticUpdate="true">
              <FilteringAndSearchPreferences protected="true">
                <ClassificationPreference>
                  <Genre href="urn:mpeg:GenreCS" preferenceValue="32">
                    <Name> Acappella </Name>
                  </Genre>
                  <Genre href="urn:mpeg:GenreCS" preferenceValue="75">
                    <Name> Classical </Name>
                  </Genre>
                  <Genre href="urn:mpeg:GenreCS" preferenceValue="40">
                    <Name> HipHop </Name>
                  </Genre>
                  <Genre href="urn:mpeg:GenreCS" preferenceValue="18">
                    <Name> Jazz </Name>
                  </Genre>
...
                  <Genre href="urn:mpeg:GenreCS" preferenceValue="63">
                    <Name> Rock </Name>
                  </Genre>
                </ClassificationPreference>
              </FilteringAndSearchPreferences>
            </UsagePreferences>
          </UserPreferences>
        </Mpeg7>
      </mpeg21:Resource>
    </mpeg21:Component>
  </mpeg21:Item>
</mpeg21:Container>
</mpeg21:DIDL>
```

Table 2. Sample of the user preferences metadata structure

Suitable OWL ontologies for metadata manipulation have been created. The main-web-service extends the OWL ontology presented in [16] for managing the audio metadata. The client uses its personal metadata based on the OWL ontology presented in Figure 2.

A client requests to listen to an audio track, upload a new audio track or retrieve a catalog of audio tracks that match specific criteria (e.g. belong in a specific audio category, comply with user preferences etc) using SPARQL queries.

Furthermore the main-web-service can propose audio tracks to the clients, based on their choices and preferences. The list of audio tracks promoted to the user is formed according to an adaptive weighting method of the user preferences metadata stored at each client and the usage history metadata stored at the server.

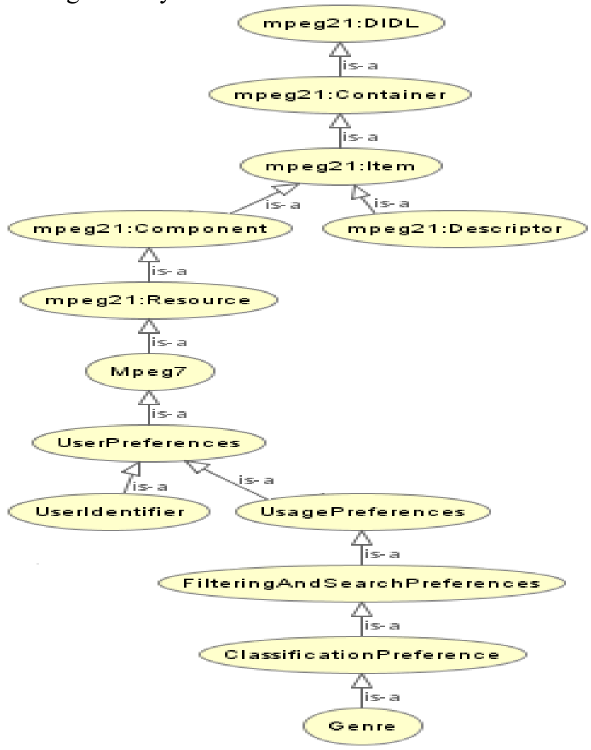

Figure 2. The OWL ontology about user preferences

At each client the weights of the user preferences metadata w_p and the usage history metadata w_h are updated according to the user actions. When the user requests to listen to an audio file that has been promoted due to the user preferences values, w_p increases while w_h decreases. Adversely, the opposite operation is performed when the audio file has been promoted due to the usage history values. Weight values are updated according to the formula:

$$w_{p/h} = w_{p/h} \pm \frac{pref\_value_i}{\sum_{j=1}^{N} pref\_value_j}$$

where $pref\_value_i$ stands for the preference value of the requested audio file, while N represents the number of all audio files and $w_p + w_h = 1$.

Initially, the client interacts with the main-web-service and sends the user's preferences along with the respective credentials. Next the service checks user credentials, establishes a session with the client and promotes audio tracks to the client according to its preferences.

The main-web-service interacts with the SPARQL parser to obtain the metadata structure expressed in OWL in order to satisfy a client request. Subsequently, the MPQF parser receives the query together with the metadata structure and returns the requested multimedia info. The client retrieves the audio information and updates its preferences.

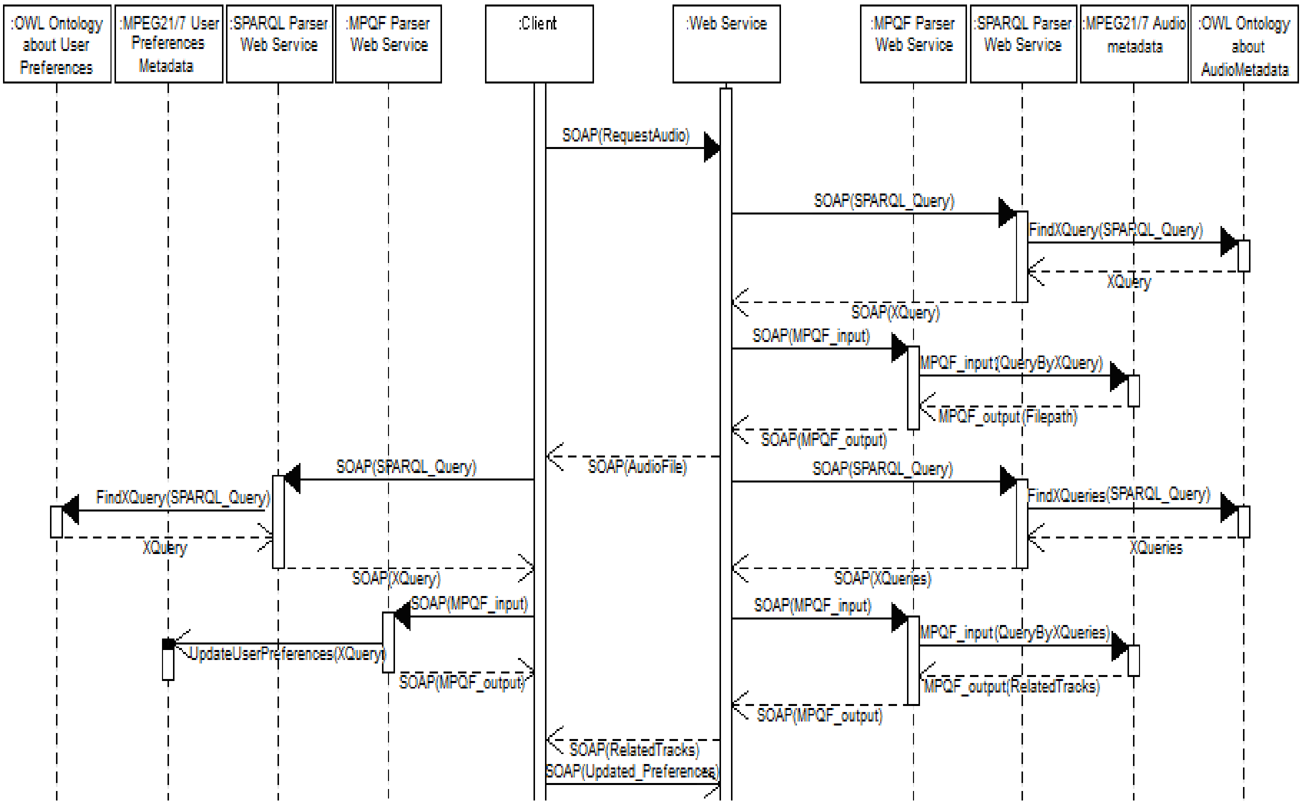

Figure 3. Web service proposes audio track

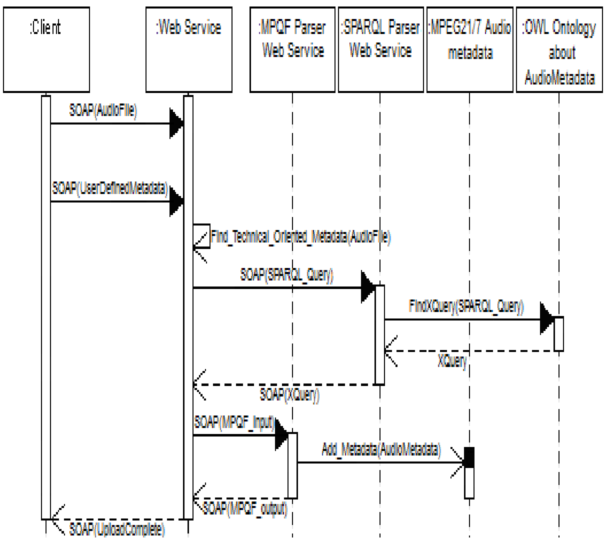

Figure 4. Updating the audio metadata

Thereafter, it sends its updated preferences to the web service as feedback information, to enhance future retrievals. The system operation is graphically illustrated in the sequence diagram of Figure 3.

User defined metadata of a specific resource are created from the client when a new audio track is uploaded to the web service's host. The main-web-service uses the Java Media Framework to analyze the uploaded audio track and extracts technical oriented metadata. Then, the service formats and inserts all the audio metadata into the metadata file according to the relative standards and to the OWL ontology. The file upload operation is presented in Figure 4. MySQL is used for handling the user's credentials at the main-web-service. The web services and client modules are developed using Java and Java Media Framework. The MPEG-21/7, SPARQL and MPQF parsers are mapped to Java classes.

V. CASE STUDY

This section presents an example of our framework's functionality. The client contacts the main-web-service and sends its preferences through the relative OWL ontology. As a next step, the client may retrieve an audio catalog using the SPARQL query presented in Table 3. The query retrieves a catalog consisted of the audio files contained to the main-web-service according to the arguments of the "FILTER" statement. The results are ordered in descending sequence according to their popularity. When the catalog is received, the client requests a specific audio track. The web service uses the relative OWL ontology to manipulate the audio metadata and sends the requested track to the client.

```
PREFIX mpeg7: < http://www.mpeg.org/MPEG7/2000>
SELECT ?Title
WHERE { ?x mpeg7:title ?title .
 FILTER (?Genre=Classical || ?Genre=Jazz
        && ?Publisher=Sony
        && ?CreationDate>=2005-01-01
        && ?MediaDuration>60
        && ?Format=MP3
        && ?FileSize>10000
        && ?BitRate>=42000)
 }ORDER BY DESC(?preferenceValue)
```

Table 3. The client retrieves an audio catalog using SPARQL

The client receives a form which plays the requested audio track and presents a list of similar tracks. This list contains promoted audio tracks according to user's preferences as well as to the usage history stored at the web service side. The user can request any track from the list. The selection of an audio track results to an update of user preferences metadata at the client and the usage history resource adaptation metadata at the web service. Figure 5 presents the relative user preference metadata block before and after the client's request. Accordingly, the 'preferenceValue' concerning audio track's genre has been increased. Similarly the CreationPreferences and the ClassificationPreferences metadata stored at the web service are increased.

```
before:
...<Genre href="urn:mpeg:GenreCS" preferenceValue="18">
    <Name>Jazz</Name>
  </Genre>...
after:
...<Genre href="urn:mpeg:GenreCS" preferenceValue="19">
    <Name>Jazz</Name>
  </Genre>...
```

Figure 5. The relative user preference block before and after the client's request

Consequently, Figure 6 presents the response times for audio proposals in respect to a SPARQL request or an audio track request. The SPARQL response times are marginally greater than the relative response times from audio track requests. This is an effect of the higher computational complexity of SPARQL requests. Additionally, as the tracks number increases the response time increases as well due to the information load processed by the main-web-service.

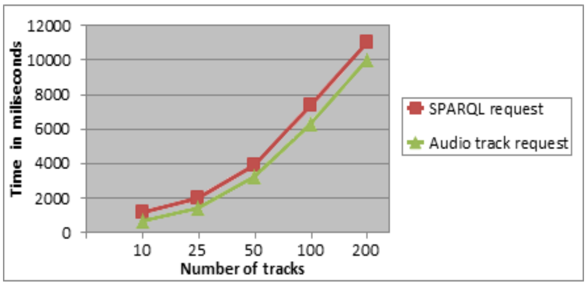

Figure 6. Response times for audio track and SPARQL request in respect of tracks' number

VI. Conclusions

Our approach is implemented through web services and relies on MPEG-21 and MPEG-7 standards to achieve personalization. MPEG-21 DIDL and DIA are used handling Digital Items declaration and user preferences, respectively. Moreover, the appropriate OWL ontologies are used for managing the metadata. The framework applies queries to the metadata and to the ontologies using MPQF and SPARQL models. The architecture is decentralized, since each client organizes its own metadata locally. The web service hosts the resource adaptation metadata along with the resources, proposing audio tracks to the clients based on their choices and preferences.

Applications conforming to MPEG-21 and MPEG-7 may use the metadata produced by our framework. Additionally the model presented in this paper can be applied to any type of multimedia resources.

Future work includes the extension of the MPQF parser functionalities to support the QueryByMedia query type. Thus, it will give the capability to the web service to propose audio tracks according to other sample audio tracks. For instance, the client will send along with its request an audio track that contains violin and the web service will promote audio tracks that also contain this musical instrument.

References

[1] Yolanda Cobos, Cristina Sarasua, Maria Teresa Linaza, Ivan Jimerez, Ander Garcia, "Retrieving Film Heritage content using an MPEG-7 Compliant Ontology", SMAP '08: Third International Workshop on Semantic Media Adaptation and Personalization, pp. 63-68, Prague, Czech Republic, December 15-16, 2008

[2] Belle Tseng, Ching Yung Lin, John Smith, "Using MPEG7 and MPEG-21 for Personilizing Video", IEEE Multimedia Journal, Vol. 11, No. 1, pp. 42-53, IEEE Computer Society, January 2004

[3] Chrisa Tsinaraki, Stavros Christodoulakis, "Semantic User Preference Description in MPEG-7/21", Fourth Hellenic Data Management Symposium, pid 6, Athens, Greece, 25-26 August, 2005

[4] Benjamin Kohncke, Wolf Tilo Balke, "Preference-driven personalization for flexible digital item adaptation", Multimedia Systems, Vol. 13, No. 2, pp. 119-130, Springer, 2007

[6] Chrisa Tsinaraki, Stavros Christodoulakis, "A User Preference Model and a Query Language that allow Semantic Retrieval and Filtering of Multimedia Content", SMAP '06: First International Workshop on Semantic Media Adaptation and Personalization, pp. 121-128, Athens, Greece, December 4-5, 2006

[7] Jun Zhai, Kaitao Zhou, "Semantic Retrieval for Sports Information Based on Ontology and SPARQL", ISME 2010: International Conference of Information Science and Management Engineering, Vol. 2, pp. 395-398, Beijing, China, August 7-8, 2010

[8] Heather Kreger, "Web Services Conceptual Architecture", IBM, May 2001

[9] MPEG-7, ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm

[10] MPEG-21, ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm

[11] Web Ontology Language, World Wide Web Consortium, http://www.w3.org/TR/owl-features/

[12] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/

[13] Mario Döller, Ruben Tous, Matthias Gruhne, Kyoungro Yoon, Masanori Sano, Ian S. Burnett, "The MPEG Query Format: Unifying Access to Multimedia Retrieval Systems", IEEE Multimedia, Vol. 5, No 4, pp. 82-95, October-December, 2008

[14] Resource Description Framework – RDF, http://www.w3.org/RDF/

[15] Ruben Tous, Jaime Delgado, "Semantic-driven multimedia retrieval with the MPEG Query Format", Multimedia Tools and Applications, Springer, Vol. 49, No 1, pp.213-233, 2010

[16] MPEG-7 OWL Ontology: http://metadata.net/mpeg7/mpeg7.owl