Personalized multimedia web services in Peer to Peer networks using MPEG-7 and MPEG-21 standards

Emmanouil Skondras[1], Angelos Michalas[1], Malamati Louta [2], George Kouzas [3]

[1]Dpt of Informatics & Computer Technology, Technological Educational Institute of Western Macedonia, 52100 Kastoria, Greece
manolisskondras@yahoo.gr, amichalas@kastoria.teikoz.gr
[2]Dpt of Informatics and Telecommunications Engineering, University of Western Macedonia, 50100, Kozani, Greece
louta@uowm.gr,
[3]Dpt of Financial and management engineering, University of the Aegean, 82100, Chios, Greece
georgekouzas@gmail.com

Abstract Multimedia information has been increased in the recent years while new content delivery services enhanced with personalization functionalities are provided to users. Several standards are proposed for the representation and retrieval of multimedia content. This paper makes an overview of the available standards and technologies. Furthermore a prototype semantic P2P architecture is presented which delivers personalized audio information. 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 P2P network's web services. 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.

1. Introduction

Nowadays, the volume of multimedia data is increasing rapidly in many information channels. Network infrastructures enable multimedia information to be easily transferred to users. The delivery of multimedia services is a common task. Several systems are emerging using various retrieval methods and algorithms to provide multimedia content. However more users tend to require information retrieval services which include high quality features such as semantic description and personalization of information.

A number of standards have been developed providing personalized multimedia content delivery services. MPEG-7 and MPEG-21 provide schemes which define efficiently multimedia metadata and user preferences. In particular user preferences include choices for specific content types, filtering and browsing modes as well as usage environment parameters. OWL provides a framework for the representation of semantic information. It offers a semantic understanding of metadata structures used for personalization. Also SPARQL and MPQF are powerful tools for querying RDF repositories and OWL ontologies.

In this paper the available standards and technologies that support multimedia retrieval services are presented. Additionally a P2P prototype application that delivers personalized audio information to users is described presenting thus an innovative approach for standards' usage. The personalization process is automated and decentralized. The information which describes the user preferences is created and stored at the client. The P2P side information includes audio resources and resource adaptation metadata, minimizing thus, the central storage and computational requirements. This significantly reduces the response time of the system, handling multiple concurrent requests from users during normal operation.

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. SPARQL is used for querying the metadata and the relative ontologies. The MPQF provides a well-known framework for applying the queries.

A client may request to listen to an audio track, upload a new audio track or retrieve a catalog of audio tracks that match specific criteria. Additionally the framework is capable to propose audio files which comply with user preferences and/or overall statistics extracted from the aggregation of all users' preferences stored as resource adaptation metadata. Moreover, a mechanism capable of combining user preferences with resource adaptation metadata is defined, which receives feedback from previous users' actions.

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. Finally, section 5 concludes our work and presents possible future extensions and plans.

2. 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 work described in [1], the authors propose a personalization process that customizes rich multimedia documents to the needs of an individual reader.

Multimedia documents, such as textbooks, reference materials and leisure materials, inherently use techniques making them accessible to people with disabilities, who are incapable of using printed materials. The authors address issues of establishing user personalization profiles, as well as adapting and customizing content, interaction and navigation. Customization of interaction and navigation leads to different user interfaces, as well as different structural content presentation. Customization of content includes insertion of a summary, synchronization of sign language video with text highlighting, selfvoicing capability, alternative support for screen readers, as well as reorganization of layout to accommodate large fonts.

The work described in [2] examines a metadata based approach, supporting the personalization process for knowledge workers who interact with 38 distributed information objects. An architecture supporting the personalization process is described, along with a prototype personalization environment. Its metadata are decentralized, in terms that the information is stored locally at client-side. The authors discuss the advantages, as well as the challenges of the suggested approach.

The authors of the approach presented in [3] introduce a wide view of personalization and user profiles, making the preferences available to a range of services and devices. Behind every instance of personalization is a profile that stores the user preferences, context of use and other information capable to deliver user experiences that describe individual users' needs and preferences. It is based upon the fact that users' needs depend on the context and current situation, (e.g. "At home", "In a Meeting", "In the Car").

In the approach proposed in [4] the user of a multimedia database returns relevance ranking to his retrieval intention for top n data of a retrieval result. Using this feedback information, the framework produces an adjustment data inherent to the user and utilizes it for personalization.

In the work described in [5] a region of interest (ROI) approach for image retrieval is presented. An "attention window" of an image is determined. Consequently, regions of interest are segmented within the attention window and relative luminance features are considered for image's decomposition. The MPEG-7 standard is used providing feature descriptions about the extracted segments. Finally, the similarity between the regions of interest is observed in respect of their MPEG-7 descriptions.

In [6], the design and the implementation of a 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 [7], 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 [8] 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.

The work described in [9] presents an agent based multimedia broadcasting framework using MPEG-21/7 and Foundation for Intelligent Physical Agents (FIPA) standards [10]. A FIPA implementation is used as platform for exchanging user preferences and program information, based on the classical client-server architecture. The user preferences are modeled in respect to the MPEG-21/7 User Preference description scheme.

[11] 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. It 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. Thus, user gets the best possible quality in respect of his terminal specifications.

[12] 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 thus 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").

The work presented in [13] 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.

In our work, a prototype architecture delivering personalized audio information over a P2P network using a combination of available standards and different technologies is presented. The personalization process is distributed since user preferences metadata are created and stored locally at each client while the network includes audio resources and resource adaptation metadata. The audio information promoted to the user is formed taking into account both the user preferences metadata and the usage history metadata of the network. The framework achieves better performance as the computational load and personalization of data are shared among each user and the P2P network. To the best of our knowledge, this study is the first one providing a distributed personalization model over a P2P network which exploits the capabilities of Web Services for handling registration, discovery and content lookup over the network, MPEG 7/21 technologies for managing metadata as well as SPARQL and MPQF standards for the retrieval of multimedia information.

3. Used Standards

This section provides an overview of the technologies and standards employed in the development of the application prototype. These are classified in two categories. The technologies responsible for the storage and delivery of content, including P2P networks [14] and Web Services [15]. The technologies supporting the semantic description and personalization of information, including MPEG-7 [16], MPEG-21 [17], OWL [18], SPARQL [19] and MPQF [20].

3.1. P2P Networks

P2P networks are typically used for enabling peers' interaction. Each peer can act either as a client making requests to other peers or as a server responding to incoming requests. P2P networks form an overlay network at the application layer which is different from the underlying physical network. Data, including stored content in any format are transferred between P2P nodes using the underlying physical infrastructure. A P2P system supports node mobility and fault tolerance operations.

There are two main architectures for P2P networks the centralized and the decentralized. Centralized architectures such as [21] have central management servers which keep information for the content of peers and control peer activities. A central server receives queries from nodes about the location of files in other nodes. Its records are dynamically updated whenever changes occur in the P2P network.

In decentralized architectures, central servers do not exist whereas requests are directed from peers to other peers of the network. Decentralized schemes are classified into three categories, namely structured, unstructured and hybrid.

In structured P2P networks, peers are organized according to specific criteria. The topology of the overlay network and the location of files are organized according to a specific algorithm. A location protocol such as the Distributed hash table (DHT) is used to allocate files to peer nodes. DHT provides lookup functionalities employing key-value pairs to route a query to the peer containing the required resource. The task of updating the key-value mapping is distributed among nodes. DHT networks handle effectively node arrivals, departures and failures. Examples of DHT networks include Pastry [22], CAN [23], Chord [24], and Tapestry [25].

In unstructured P2P networks such as Gnutella [26] and KaZaA [27], the overlay links are randomly formed. Each new node entering the network chooses its peers according to a specific algorithm. If a peer needs to find a file, it queries its neighbors. Commonly, a query is flooded through the network to find peers that share the file. Flooding is a convenient technique for searching popular highly replicated items, however it lacks efficiency in case the requested items are rare. In the later case flooding may produce large traffic loads to the network without guaranteeing that the required files will be found due to possible restrictions on the flood range.

Hybrid architectures [28] combine structured and unstructured searching techniques to improve performance in P2P networks. Requests for popular items are performed using flooding while rare items are searched using DHT techniques. A challenging issue in hybrid P2P networks is to define mechanisms for the characterization of items as popular or rare.

3.2. Web Services

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 [29] and other standards allowing easy integration with existing distributed applications. A web service exploits a set of standards [15] including the Simple Object Access Protocol (SOAP), the Web Services Description Language (WSDL) and the Universal Description Discovery and Integration protocol (UDDI).

The Simple Object Access Protocol (SOAP) is an XML based protocol for exchanging structured data with web services. SOAP can be used in combination with a variety of other protocols such as HTTP, SMTP and FTP. The SOAP messaging framework [30] defines the SOAP message construct, guidlines for processing SOAP messages, mechanisms which provide message extensibility features as well as rules for carrying SOAP messages from an underlying protocol. A SOAP message consists of an “Envelope” element which contains a “Header” and a “Body” element. The “Envelope” is the root element of the message. The “Header” contains transmission information. The “Body” contains the message information to be transmitted and a “Fault” element which is used for error reporting. Tables 1 and 2 present a SOAP request and a SOAP response message respectively.

Table 1. A SOAP request message

```
POST /InAudioTitles HTTP/1.1
Host: www.itisanexample.gr
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body xmlns:examplens="http://www.itisanexample.gr/examplens ">
  < examplens:GetAudioTitles>
    < examplens: AudioAuthor >AuthorA </ examplens:AudioAuthor>
  </ examplens: GetAudioTitles >
</soap:Body>
</soap:Envelope>
```

Table 2. A SOAP response message

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
<?xml version="1.0"?>
<soap:Envelope  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body xmlns:mpeg7="http://www.mpeg.org/MPEG7/2000">
  <mpeg7:CreationPreferences>
    <mpeg7:Title mpeg7:preferenceValue="12" xml:lang="en">track1.mp3</mpeg7:Title>
  </mpeg7:CreationPreferences>
</soap:Body>
</soap:Envelope>
```

The Web Services Description Language (WSDL) [31] is a description language in XML which defines web services and access rules for them. WSDL defines services as a set of end points called ports which can exchange messages. Once a client initiates a connection with a web service he reads the WSDL file at the server to find which operations are available. A typical WSDL document contains the following elements:

- Types: Contains data type definitions in XML Schemas.
- Message: Includes the definitions of the data needed to perform an operation.
- Operation: Defines a function provided by the relative web service.
- PortType: Describes the web service's interface where the supported operations are defined.
- Binding: Indicates the transport protocol (SOAP) and the PortType of a service.
- Port: Specifies a connection point to a web service described as a combination of a binding with a network addresses.
- Service: Contains a collection of ports which provide access to service's operations.

The Universal Description Discovery and Integration protocol (UDDI) [15] defines a model for publishing and discovering web services. It is implemented as a central element of the service oriented approach. There are approximately forty well defined SOAP messages according to the UDDI version 2 specifications, providing publication as well as retrieval operations. Each registration consists of five data structures types assisting the management of different information. Each structure defines elements for serving business or technical purposes which are defined as follows:

- BusinessEntity structure: It provides information about the publisher of a service as well as the provided services.

▪ BusinessService structure: It represents a service classification and is a logical child of a businessEntity structure.

▪ BindingTemplate structure: It represents technical features about services and is a logical child of a businessService structure.

▪ TModel structure: It provides abstract reference descriptions of how to interact with the web service as well as service compatibility issues.

▪ PublisherAssertion structure: It extends businessEntity's functionalities by defining relationships between publishers.

In a typical scenario a service provider hosts a Web service. Additionally he defines a relative service description which publishes either directly to a service requestor or to a UDDI service registry. Subsequently, the service requestor retrieves the service description locally or from the service registry and interacts with the Web service exchanging SOAP messages.

3.3. Multimedia Description Standards

MPEG-7 [16] is a standard developed from Moving Pictures Expert Group (MPEG) for the description of multimedia information. The standard provides a framework for the description of multimedia content encoded in any existing scheme such as MPEG1, MPEG2, and MPEG4. Metadata are stored in XML allowing efficient indexing, searching and filtering of multimedia data. MPEG-7 defines the following elements:

- Description tools, which include Descriptors (D) and Description Schemes (DS). Descriptors define the syntax and the semantics of metadata elements. Description Schemes contain Descriptions, other Description Schemes as well as relationships between them.
- A Description Definition Language (DDL), which is used for defining the syntax of Description Tools and creating new or extending existing Description Schemes.
- System tools which provide mechanisms for multiplexing descriptors and synchronizing descriptions with content defining an open framework for multimedia applications

The MPEG-21 [17] standard defines a framework for effectively managing multimedia resources. 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 of the DIA framework contains the following attributes:

▪ 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.

3.4. OWL and Query Languages

The RDF Schema (RDFS) [32] provides structures for knowledge representation. It deals with the organization of ontological hierarchies such as classes, relationships and properties. However complex structures or restrictions such as the scope of properties or the cardinality of attributes can not be supported in RDFS. The need of a more powerful ontology language leads us to the Web Ontology Language (OWL).

OWL [33] 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 [13] 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 [34] presents the SPARQL language structures. The query results XML specification [35], defines the format of the results returned from SPARQL queries as XML documents. The SARQL protocol [36] defines the framework for sending queries from clients to remote server using HTTP or SOAP messages.

The Mpeg Query Format (MPQF) [20] 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. Moreover advanced multimedia retrieval operations are supported via a rich set of defined query types:

- QueryByMedia: Query using a specified image, video, audio or text.

- QueryByDescription: Query using metadata expressed in XML format.
- QueryByFreeText: Query using keywords.
- QueryByFeatureRange: Query using features such as bitrate or media duration.
- SpatialQuery: Query for spatial elements within media objects.
- TemporalQuery: Query for temporal characteristics within media objects.
- QueryByROI: Query for spatial-temporal characteristics within media objects.
- QueryByXQuery: A container which supports XQuery expressions.
- QueryByRelevanceFeedback: Query taking into consideration results from previous searches which are characterized as bad or good examples.
- QueryBySPARQL: Query which embodies a SPARQL query.

Combination of query types is also supported providing advanced multimedia requests. Table 3 presents an example which combines three query types (QueryByMedia, QueryByDescription and QueryBySPARQL) and retrieves images according to the parameters of the relevant condition blocks.

Table 3. A multipart MPQF query

```
<MpeqQuery><Query><Input>
  <QueryCondition>
    <TargetMediaType>image/*</TargetMediaType>
      <Condition xsi:type="AND">
        <Condition xsi:type="QueryByMedia">
          <MediaResource xsi:type="MediaResourceType"><MediaResource>
            <InlineMedia type="image/jpeg"><MediaData64>PeQYmmW4ML8m2iQ3AzMBTbmodr</MediaData64></InlineMedia>
          </MediaResource></MediaResource>
        </Condition>
        <Condition xsi:type="QueryByDescription" matchType="exact">
          <DescriptionResource resourceID="des01"> <AnyDescription xmlns:mpeg7=" http://www.mpeg.org/MPEG7/2000">
             <mpeg7:Mpeg7>
              <mpeg7:MediaFormat><mpeg7:FileFormat mpeg7:href="urn:mpeg:mpeg7:cs:FileFormatCS:2001:3">
                <mpeg7:Name xml:lang="en">JPEG</mpeg7:Name></mpeg7:FileFormat>
              </mpeg7:MediaFormat>
           </mpeg7:Mpeg7>
         </AnyDescription> </DescriptionResource>
       </Condition>
       <Condition xsi:type="QueryBySPARQL">
         <SPARQL>
           PREFIX mpeg7: < http://www.mpeg.org/MPEG7/2000>
           SELECT ?Title WHERE { ?x mpeg7:title ?title . FILTER (?Genre=Natural || ?Genre=Animal ) }
         </SPARQL>
       </Condition>
     </Condition>
   </QueryCondition>
</Input></Query></MpegQuery>
```

Furthermore MPQF supports query management tools which define inquiries for multimedia services that satisfy specific requirements. In this way, a management request searches for multimedia services that satisfy specific requirements. Following, a management response returns the appropriate service descriptions. The standard includes procedures for service discovery, query formats for service capabilities and descriptions of service capabilities.

4. A Prototype Architecture

In this section a P2P prototype application that delivers personalized information is presented (Figure 1). Each peer interacts with a main-web-service, an MPQF parser and a SPARQL parser, also implemented as web services. A Network Access Server (NAS) is used providing the necessary login functionalities to the users and a UDDI service enhances web services' interaction capabilities. UDDI describes each peer's content as well as web services' capabilities.

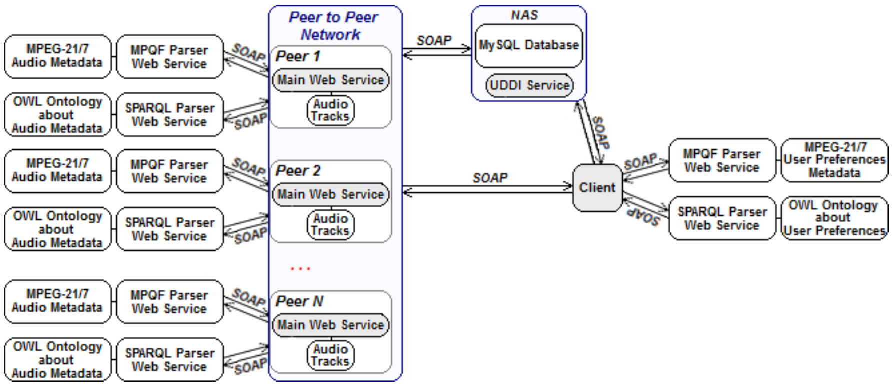

Fig.1. The basic modules of our architecture

The information contained in the UDDI registry service can be classified in two main categories. The first, provides abstract reference descriptions about the web services and according to the UDDI terminology is referred as tModel (Table 4). The second, defines services implementations as well as detailed technical information about services functionalities. Each peer interacts with a set of web services (main-web-service, MPQF parser and SPARQL parser) and publishes its capabilities to the UDDI service. The client communicates with the UDDI service and retrieves the appropriate information needed to interact effectively with the peers and receive personalized information.

Table 4. An example of tModel entry


```
<businessEntity xmlns="urn:uddi-org:api" businessKey="00000000-0000-0000-0000-000000000000">
 <name>Main Web Service</name>
 <description xml:lang="en">The main web service</description>
 <businessServices>
   <businessService  businessKey="00000000-0000-0000-0000-000000000000" serviceKey="11111111-1111-1111-1111-111111111111">
   <name>Promote Audio</name>
   <bindingTemplates>
   <bindingTemplate serviceKey="11111111-1111-1111-1111-111111111111"
     bindingKey="AAAAAAAA-AAAA-AAAA-AAAA-AAAAAAAAAAAA">
   <accessPoint URLType="http">http://peerhost.com/mainwebservice.jws</accessPoint>
    <tModelInstanceDetails>
     <tModelInstanceInfo
     tModelKey="UUID:BBBBBBBB-BBBB-BBBB-BBBB-BBBBBBBBBBBB"/>
    </tModelInstanceDetails>
   </bindingTemplate>
  </bindingTemplates>
  </businessService>
 </businessServices>
 <categoryBag>
  <keyedReference
  tModelKey="UUID:AB12C1DA-AD34-12A1-DACD-A1DAC23C12AC"
  keyName="Multimedia personalization services"
  keyValue="70.24.12.11.08"/>
 </categoryBag>
</businessEntity>
```

The architecture is decentralized in respect to the information required to achieve personalization. User related preferences are created and stored at each client. Resource adaptation metadata along with the resources are the only to be composed and stored centrally at the P2P network. 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. 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.

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 thirty 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 5 presents a sample of the audio metadata structure.

Table 5. 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>Music Track</mpeg7:Name>
                </mpeg7:What>
               </mpeg7:StructuredAnnotation>
             </mpeg7:Abstract>

            <mpeg7:CreationCoordinates>
```

```xml
                    <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>
```


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 6 presents a sample of the user preferences metadata structure.

Table 6. Sample of the user preferences metadata structure


```xml
<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 first'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="42">
                    <Name>Acappella</Name>
                  </Genre>
                 <Genre href="urn:mpeg:GenreCS" preferenceValue="30">
                    <Name>Animal</Name>
                  </Genre>
                  <Genre href="urn:mpeg:GenreCS" preferenceValue="2">
                    <Name>Acoustic</Name>
                  </Genre>
```

```
              <Genre href="urn:mpeg:GenreCS" preferenceValue="34">
                <Name>Bass</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="91">
                <Name>Jazz</Name>
              </Genre>
              <Genre href="urn:mpeg:GenreCS" preferenceValue="38">
                <Name>Pop</Name>
              </Genre>
              <Genre href="urn:mpeg:GenreCS" preferenceValue="24">
                <Name>Speech</Name>
              </Genre>
…
              <Genre href="urn:mpeg:GenreCS" preferenceValue="17">
                <Name>Traditional</Name>
              </Genre>
              <Genre href="urn:mpeg:GenreCS" preferenceValue="52">
                <Name>Other</Name>
              </Genre>
            </ClassificationPreference>
          </FilteringAndSearchPreferences>
        </UsagePreferences>
      </UserPreferences>
    </Mpeg7>
   </mpeg21:Resource>
  </mpeg21:Component>
 </mpeg21:Item>
</mpeg21:Container>
</mpeg21:DIDL>
```


Moreover, suitable OWL ontologies for metadata manipulation have been created. The main-web-service extends the OWL ontology presented in [37] for managing the audio metadata. The client uses its personal metadata which contain the semantic description of user preferences, based on the OWL ontology presented in the class diagram of Figure 2.

The client interacts with the P2P network and sends the user's preferences along with the respective credentials. Next, the NAS checks user credentials, establishes a session between the client and the P2P network. Thereafter, the P2P network promotes music tracks to the clients, according to their choices and preferences. A client can upload a new music track, request to listen to a specific music track as well as retrieve a catalog of music tracks that match specific criteria (e.g. belong in a specific music category, comply with user preferences) using SPARQL queries. The aforementioned functionalities are discussed below.

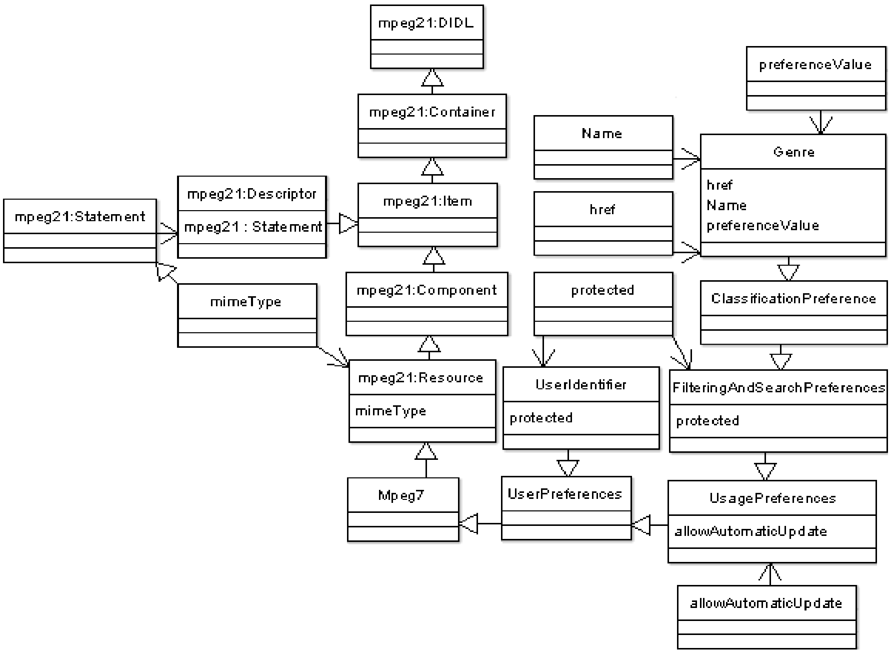

Fig.2. OWL ontology about user preferences metadata

4.1 Audio Track Upload

User defined metadata of a specific resource are created from the client when a new audio track is uploaded to the P2P network. The client interacts with the UDDI service and finds which peers can receive the new audio file.

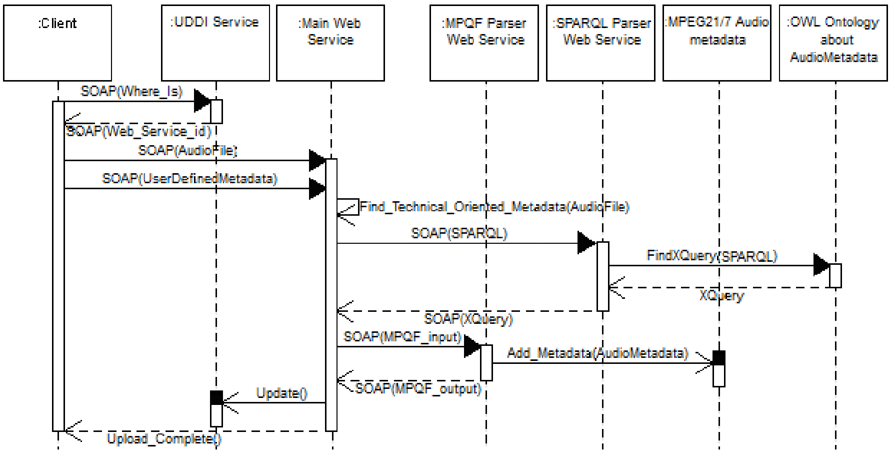

Fig.3. Adding a new audio file

The main-web-service uses the Java Media Framework to analyze the uploaded audio track and extracts technical oriented metadata. After these actions, the service interacts with the SPARQL/MPQF parsers 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 3.

4.2 Audio Track Request

The client interacts with the UDDI service, finds which peers contain a specific audio track and makes a request to them. Each peer main-web-service, interacts with the SPARQL parser and obtains 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. As a next step, each peer sends a part of the requested audio track. The client retrieves the audio information and updates its preferences. Thereafter, it sends its updated preferences to the main-web-services as feedback information, to enhance future retrievals. The P2P network’s web services interact with each other to synchronize the usage history metadata.

Furthermore each peer involved in the track request process proposes 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 main-web-service.

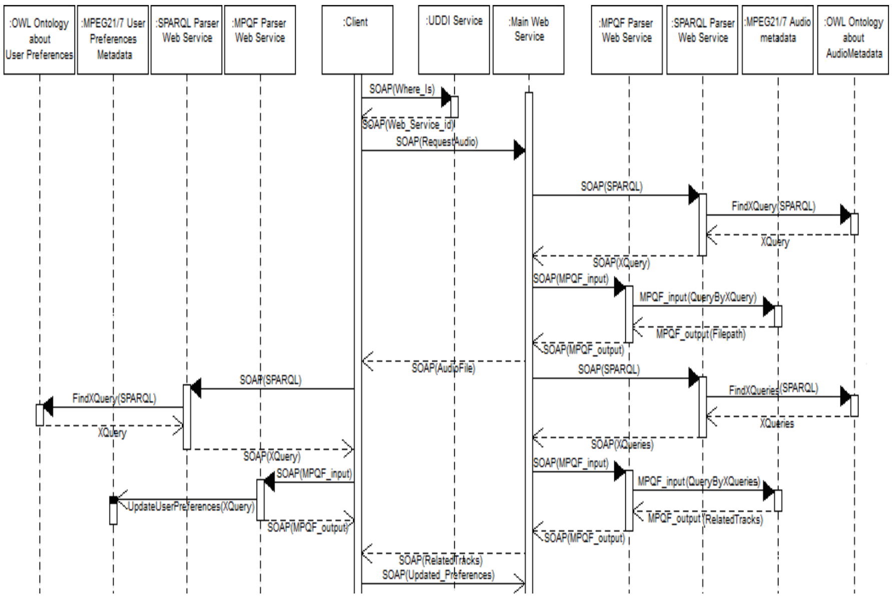

Fig.4. Web service proposes audio tracks

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 $w_p + w_h = 1$. The system operation is graphically illustrated in the sequence diagram of Figure 4.

4.3 SPARQL Request

The client retrieves audio catalogues matching specific criteria using SPARQL queries. The client interacts with the UDDI service and retrieves a list of the available peers. Then it makes a respective request containing the SPARQL query to the relevant peers. Each peer main-web-service receives the SPARQL query, interacts with the SPARQL parser and obtains the metadata structure expressed in OWL in order to satisfy the client's request. Consequently the MPQF parser receives the query together with the metadata structure and returns the requested audio list. Finally, the main-web-service sends the extracted list to the client. Figure 5 presents the catalogue retrieval process.

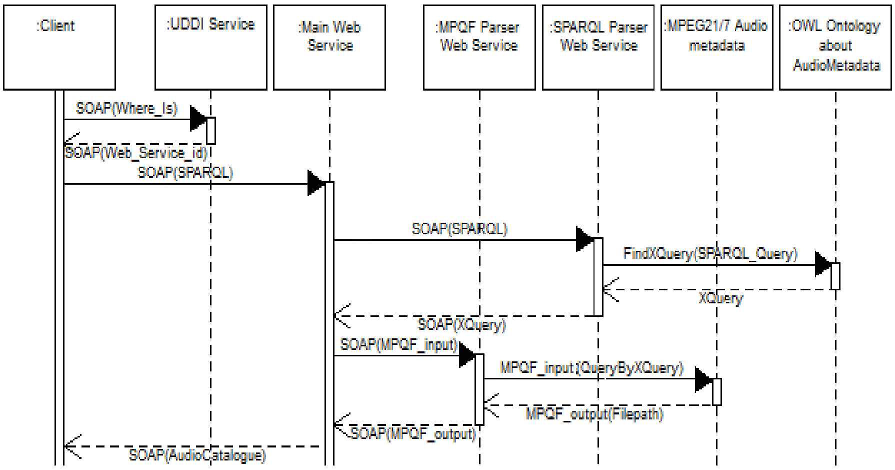

Fig.5. Audio catalogue retrieval

5. Case Study

This section presents an example of our framework's functionality. At first, the client contacts the P2P network and sends his preferences according to the relative OWL ontology using a SOAP message. As a next step the client may retrieve an audio catalog using the SPARQL query presented in Table 7. The query retrieves a catalog of audio files according to the arguments of the "FILTER" statement. The results are ordered by descendance sequence according to their popularity.

Table 7. The client retrieves an audio catalog using SPARQL

```
PREFIX mpeg7: < http://www.mpeg.org/MPEG7/2000>
SELECT ?Title
WHERE { ?x mpeg7:title ?Title .
FILTER (?Genre=Speech
    && ?Author=Yakub
    && ?Subject=Blood Pressure || ?Subject=Cholesterol
    && ?CreationDate>=2001-01-01 && ?CreationDate<=2004-03-15
    && ?Language=English
    && ?MediaDuration>=3600
    && ?Format=MP3 || ?Format=WAV
    && ?Filesize<=20000000
    )
 }ORDER BY DESC(?preferenceValue)
```

Following, when the catalog is received, the client requests a specific audio track. The P2P network's web services use the relative OWL ontology to manipulate the audio metadata and send the requested track to the client. 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 stored locally at the client as well as to the usage history stored at the P2P network. 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 P2P network.

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

Fig.6. The relative user preference block before and after the client's request

Figure 6 presents the user preference metadata block before and after the client's request. Accordingly, the 'preferenceValue' concerning audio track's genre is increased.

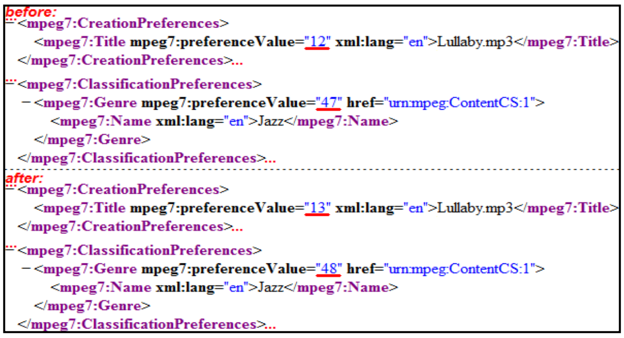

Fig.7. The relative audio metadata blocks before and after the client's request

Similarly, figure 7 presents the CreationPreferences and the ClassificationPreferences metadata blocks stored at the relative main-web-services of the P2P network before and after client's request. The 'preferenceValue' of the former block shows the number of times the relative audio track has been requested from all users. Respectively, the 'preferenceValue' of the later shows the times the relative genre of tracks has been requested from all users. Both values are increased after the client request.

Subsequently the client selects to listen to audio files promoted by the user preferences values. Figure 8 illustrates the percentage of the promoted audio files based on the user preferences values for 5 consecutive requests. The proposed adaptive weighting method improves the retrieval process by increasing the percentage of the promoted audio files according to user choices.

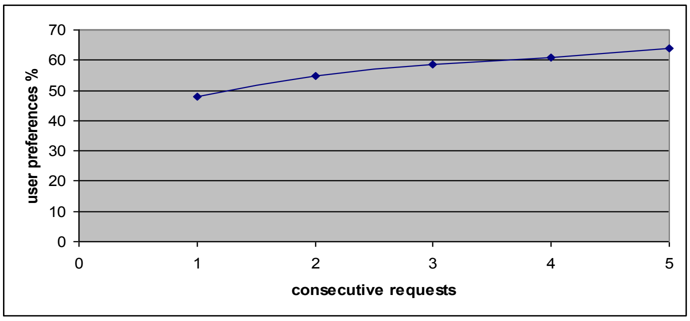

Fig.8. Audio files promoted to the user based on the user preferences values

Figure 9 presents the response times for audio proposals in respect to a SPARQL or an audio track request. In this scenario each peer contains 20 audio tracks and the relative audio catalogue is extracted according to the mechanisms described in the previous sections. As the peers number increases, the response time increases proportionally as an effect of the load of information process. Additionally, the SPARQL response times are marginally greater compared to the relative response times from audio requests. This is an effect of the complexity of SPARQL requests resulting to higher computational requirements.

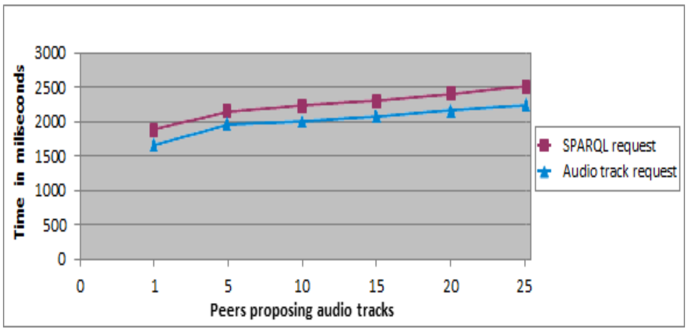

Fig.9. Response times for audio track and SPARQL request in respect of peers' number

Consequently, Figure 10 presents the response times for audio proposals in respect to the tracks contained in each peer. Similarly the SPARQL response times are greater than the relative audio request times. Respectively, as the tracks number increases the response time increases as well due to the information load processed per peer.

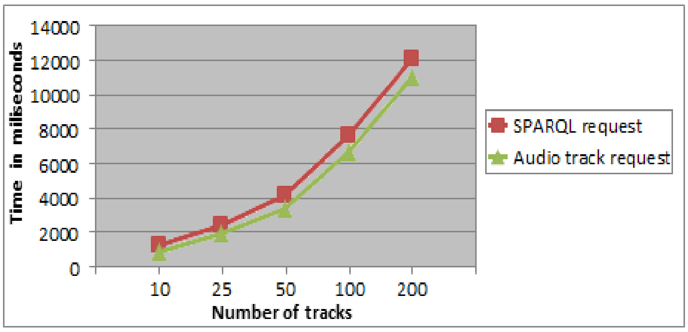

Fig.10. Response times for audio track and SPARQL request in respect of tracks' number

6. Conclusions

Our approach 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 is implemented over web services in a P2P network. It uses SOAP messages for services communication and applies queries to the metadata and to the ontologies using MPQF and SPARQL models. A UDDI service is also used describing each peer's content as well as web services' capabilities and enhancing thus framework's functionalities. The architecture is decentralized improving framework's scalability. Each client organizes its own metadata locally. The P2P network hosts the resource adaptation metadata along with the resources, proposing audio tracks to the clients based on their choices and preferences. A client can also retrieve audio catalogues using SPARQL queries.

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

Future work includes the extension of the MPQF parser's functionalities to support the QueryByMedia query type. Thus, it will give the capability to the web services 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 services will promote audio tracks that also contain this musical instrument.

References

[1] H.L. Petrie, G. Weber, W. Fisher, "Personalization, interaction, and navigation in rich multimedia documents for print-disabled users", IBM Systems Journal, Vol 44, No 3, pp. 629-635, IBM, 2005

[2] David L. Hicks, Klaus Tochtermann, "Personalizing Information Spaces: A Metadata Based Approach", International Conference on Dublin Core and Metadata Applications, pp.213-220, Tokyo, Japan, October 24-26, 2001

[3] Tatiana Kovacikova, Francoise Petersen, Mike Pluke, Valentin Alonso Alvarez, Giovanni Bartolomeo, Antonella Frisiello, Erik Zetterstrom, Scott Cadzow, "Personalization and User Profile Standardization", Seventh IASTED International Conference on Communication Systems and Networks, Track 629-043, Palma de Mallorca, Spain, September 1-3, 2008

[4] Tsuyoshi Takayama, Hirotaka Sasaki, Shigeyuki Kuroda, "Personalization by Relevance Ranking Feedback in Impression-based Retrieval for Multimedia Database", Journal of Systemics, Cybernetics and Informatics, Vol. 3, No 2, pp.85-89, IIISCI, 2005

[5] Mi Suk Seo, Byoung Chul Ko, Hong Chung, Jae Yeal Nam, "ROI-Based Medical Image Retrieval Using Human-Perception and MPEG-7 Visual Descriptors", Book: Lectures in Computer Science, Vol. 4071, pp.231-240, Springer, 2006

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

[7] Belle Tseng, Ching Yung Lin, John Smith, "Using MPEG7 and MPEG-21 for Personilizing Video", IEEE Multimedia Journal, Vol. 11, Num. 1, pp. 42-53, IEEE Computer Society, January 2004

[8] 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

[9] Munchurl Kim, Jeongyeon Lim, Kyeongok Kang, Jinwoong Kim, "Agent-Based Intelligent Multimedia Broadcasting within MPEG-21 Multimedia Framework", ETRI Journal, Vol 26, Num 2, Daejeon, South Korea, April 2004

[10] The Foundation for Intelligent Physical Agents, http://www.fipa.org/

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

[12] 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

[13] 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

[14] Andy Oram, "Peer to Peer: Harnessing the Power of Disruptive Technologies", O'Reilly & Associates, 2001

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

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

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

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

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

[20] 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

[21] Ivan Lee, Yifeng He, Ling Guan, "Centralized P2P Streaming with MDC", Seventh IEEE Workshop on Multimedia Signal Processing, pp. 1-4, October 30-November 2, 2005

[22] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001.

[23] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-addressable Network. In Proceedings of ACM SIGCOMM, pages 161-172, San Diego, CA, August, 2001.

[24] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In Proceedings of ACM SIGCOMM, pages pp. 149–160, San Diego, CA, August, 2001.

[25] K. Hildrum, J. D. Kubiatowicz, S. Rao, and B. Y. Zhao, "Distributed object location in a dynamic network," Theory of Computing Systems, vol. 37, no. 3, pp. 405-440, May 2004.

[26] Gnutella. http://rfc-gnutella.sourceforge.net/

[27] KazaA. KaZaA. http://www.kazaa.com.

[28] B. T. Loo, R. Huebsch, I. Stoica, and J. M. Hellerstein, "The case for a hybrid P2P search infrastructure,", G. M. Voelker and S. Shenker, Eds., Lecture Notes in Computer Science, Vol. 3279, Berlin, Germany: Springer-Verlag, 2005, pp. 141-150.

[29] Mahmoud Qusay, "Overview of CORBA", Distributed Programming with Java, Manning Publications, Chapter 11, pp. 153-161, January 2001

[30] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation, 27 April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[31] Web Services Description Language (WSDL) 1.1, W3C Note, 15 March 2001, http://www.w3.org/TR/wsdl

[32] RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/rdf-schema/

[33] OWL 2 Web Ontology Language Document Overview, W3C Recommendation, 27 October 2009, http://www.w3.org/TR/owl2-overview/

[34] SPARQL Query Language for RDF, W3C Recommendation, 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/

[35] SPARQL Query Results XML Format, W3C Recommendation, 15 January 2008, http://www.w3.org/TR/rdf-sparql-XMLres/

[36] SPARQL Protocol for RDF, W3C Working Draft, 14 January 2005, ftp://www.w3.org/TR/2005/WD-rdf-sparql-protocol-20050114/

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