A Mobile Audio Server enhanced with Semantic Personalization Capabilities

Emmanouil Skondras[1], Vasileios Triantafyllou[2], Polymnia Chersoniotaki[3]

Department of Informatics
Athens University of Economics and Business
10434 Athens, Greece
e-mail: manolisskondras@yahoo.gr[1], btriantafgr@hotmail.com[2], herson_86@windowslive.com[3]

ABSTRACT
This paper presents a mobile audio server enhanced with personalization capabilities. The server as well as the client is implemented over the Android platform. It provides mobility to the system's parts. The metadata which support personalization are separated into two categories: the metadata describing user preferences stored at each client and the resource adaptation metadata stored at the server. The multimedia models MPEG-21 and MPEG-7 are used to describe metadata information. The Web Ontology Language (OWL) is used to produce and manipulate the relative semantic descriptions about the metadata. The mobile server promotes audio tracks to the clients according to their preferences.

Key words: Android platform, Mobile Systems, Personalization, MPEG-21, MPEG-7, Semantic Descriptions

INTRODUCTION

Nowadays, the need for mobile multimedia services is increasing rapidly. However more users require multimedia services with high quality features such as mobility and personalized information. In addition, the computational capabilities of the mobile devices are evolving rapidly, enabling the ability of executing more complex processes in fully mobile environments. Tablets and smartphones with powerful specifications (such as dual core processors) are available in the market.

In this paper a prototype application that delivers personalized audio information to mobile users is described. The system is implemented over the Android platform, which means that the server as well as the client gains mobility. The server can be a powerful Android tablet or smartphone. On the other hand, the client can be a simple Android smartphone.

MPEG-7 and MPEG-21 standards are used for the audio content as well as the user preferences' description. The metadata information is managed using OWL ontologies. The information that describes the user preferences is stored and manipulated locally at each client, minimizing thus the central storage and computational requirements. The server-side information includes audio resources and resource adaptation metadata.

The remainder of the paper is organized as follows. Firstly, the related research literature as well as an overview of the standards followed in this study is described. Then the software architecture that supports the prototype application, the software elements and modules are presented. The final section concludes our work and presents possible future extensions.

MATERIALS AND METHODS

Related Work

The rapid increase in multimedia content has challenged the research communities into the development of mobile tools enhanced with personalization capabilities. An increasing number of applications use well-defined standards to enhance the personalization process.

The work described in [1] deals with personalization issues of mobile services. It presents two models for mobile services' personalization and describes advantages and disadvantages of each model. Both models are implemented over mobile services. The first one (model 1) defines the implementation of new personalized mobile services over well-defined interfaces. On the other hand, the second model (model 2) defines a way for personalizing existing mobile services. The authors also describe one case study per model. Especially, the personalization of an Internet browser according to the model 1 as well as to the model 2 is presented.

In [2] a mobile music personalization system is presented. It combines the advantages of collaborative [3] as well as item based filtering systems [4]. A multi-dimensional profile system is used for modeling system's users. It models a variety of user's characteristics such as age, gender, music preferences, shopping history, favored and unflavored items. The authors propose a wide variety of recommendation strategies including:

- "Reminder": This strategy proposes items in a periodical way.
- "More like this": It proposes items which shares similarities with a specific item.
- "Hot items": According to this strategy, currently famous items are proposed.
- "Broaden my horizon": It proposes items in respect of user's preferences.
- "Similar users like": It proposes items according to users' preferences similar with the preferences of the current user.

[5] presents a multimedia retrieval system for film heritage. The multimedia content has been indexed using an annotation tool based on the MPEG-7 standard. An OWL ontology has also been created 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 (such as WiMax and WANs) for the promotion of Film Heritage.

The work described in [6] presents an agent based multimedia broadcasting framework which uses MPEG-21/7 and Foundation for Intelligent Physical Agents (FIPA) standards [7]. A FIPA implementation is used as platform for exchanging user preferences and program information, based on the client-server architecture. The user preferences are modeled in respect of the MPEG-21/7 User Preference description scheme.

The work described in [8] addresses the issues associated with designing a video personalization and summarization system in heterogeneous usage environment. The authors introduce a framework for a three-tier summarization system, consisted of a server, a middleware and a client. The server maintains content resources, MPEG-7 metadata descriptions, MPEG-21 rights expressions as well as content adaptability declarations. The client exploits MPEG-7 user preferences and MPEG-21 usage environments, in order to retrieve 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 and browser portals.

[9] presents a video summarization and personalization framework for mobile devices such as Palm-OS smartphones and PDAs. The system includes the following parts: a client, a server and a video middleware. The client part consisted of a Palm-OS video application interface as well as client profiles which deal with user's preferences (user profile), device specifications (device profile) and video transmission parameters (transmission profile). The server part consisted of video resources, the relative MPEG-7 video descriptions and video annotation tools. The video middleware consisted of a semantic transcoder and a transmission server. The annotation tools enable the capability to the server to annotate the video resources and to produce the MPEG-7 video descriptions. The client shares his profiles and makes a request to the server. The middleware's semantic transcoder part matches client's request with the server's video descriptions and retrieves relative videos. Then the server transmits the personalized video to the client via the middleware's transmission server part.

Materials

This section makes an overview of the standards used for the development of the prototype architecture. These standards include MPEG-7 [10], MPEG-21 [11] and OWL [12].

The MPEG-7 is formally called Multimedia Content Description Interface. It is a multimedia content description standard. The description is associated with the content itself. It does not deal with the actual encoding of the content, like MPEG-1, MPEG-2 and MPEG-4. The metadata are related with multimedia resources and expressed in a XML structure. Descriptors (which define the syntax and the semantics of metadata elements) as well as Description Schemes (which contain Descriptions, other Description Schemes and relationships between them) are defined.

The MPEG-21 standard defines an open framework for multimedia applications. It uses the architectural concept of the Digital Item. A Digital Item is a combination of multimedia resources, metadata 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 Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies endorsed by the World Wide Web Consortium (W3C). It provides semantic description capabilities for digital items such as multimedia resources. OWL is adopted so as to create the relative ontologies and provide a common semantic understanding between the components involved in a personalization process.

Methods

This section describes the architecture of our system which is implemented over the Android platform. It is decentralized in respect to the information required to achieve personalization, minimizing thus the central computational requirements and improving framework's efficiency. User preferences' metadata are created and stored locally at each client. Resource adaptation metadata along with the resources are the only to be composed and stored centrally at the server.

The server contains the audio tracks and the respective audio metadata in an MPEG-21/7 structure. The audio tracks are divided into sixteen different categories (rock, pop, jazz, acappella etc.). Audio metadata include user defined metadata (artist, production year, category etc.), technical oriented metadata (bitrate, sample rate, track duration, audio channels, audio format, file size etc.) as well as usage history metadata (track's popularity in respect to all tracks and track's popularity in its category). Table 1 presents a sample of the audio metadata structure.

Table 1. 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="27"
            xml:lang="en"> rockConcert.mp3</mpeg7:Title>
         </mpeg7:CreationPreferences>
         <mpeg7:CreationInformation>
          <mpeg7:Creation>
            <mpeg7:Creator>
             <mpeg7:Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:AUTHOR"/>
             <mpeg7:Agent xsi:type="PersonType">
              <mpeg7:Name>
               <mpeg7:GivenName>Nick</mpeg7:GivenName>
               <mpeg7:FamilyName>Smith</mpeg7:FamilyName>
              </mpeg7:Name>
             </mpeg7:Agent>
            </mpeg7:Creator>
            <mpeg7:Creator>
             <mpeg7:Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:Publisher"/>
             <mpeg7:Agent xsi:type="PersonType">
              <mpeg7:Name>
               <mpeg7:GivenName>John</mpeg7:GivenName>
               <mpeg7:FamilyName>Smith</mpeg7:FamilyName>
              </mpeg7:Name>
             </mpeg7:Agent>
            </mpeg7:Creator>
            <mpeg7:Abstract>
             <mpeg7:FreeTextAnnotation>Excelent!!!
             </mpeg7:FreeTextAnnotation>
             <mpeg7:StructuredAnnotation>
              <mpeg7:What><mpeg7:Name>Music Track</mpeg7:Name>
              </mpeg7:What>
             </mpeg7:StructuredAnnotation>
            </mpeg7:Abstract>
            <mpeg7:CreationCoordinates>
             <mpeg7:CreationDate>
              <mpeg7:TimePoint>2011-04-17</mpeg7:TimePoint>
              <mpeg7:Duration>P7D</mpeg7:Duration>
             </mpeg7:CreationDate>
            </mpeg7:CreationCoordinates>
           </mpeg7:Creation>
         </mpeg7:CreationInformation>
         <mpeg7:ClassificationPreferences>
        <mpeg7:Genre mpeg7:preferenceValue="92"  href="urn:mpeg:ContentCS:1">
           <mpeg7:Name xml:lang="en">Rock</mpeg7:Name>
          </mpeg7:Genre>
         </mpeg7:ClassificationPreferences>
         <mpeg7:MediaLocator>
          <mpeg7:MediaUri>tracks/rockConcert.mp3</mpeg7:MediaUri>
         </mpeg7:MediaLocator>
         <mpeg7:MediaTime>
          <mpeg7:MediaTimePoint>T00:00:00F100</mpeg7:MediaTimePoint>
          <mpeg7:MediaDuration>T00:04:27F100</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>3700747</mpeg7:FileSize>
          <mpeg7:BitRate mpeg7:minimum="N/A" mpeg7:average="101000"
           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>
```


The client's metadata describe user's preferences in respect of the MPEG-21 and MPEG-7 standards. Table 2 presents a sample of the user preferences metadata structure.

Table 2. 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 user preferences.</mpeg21:Statement>
     </mpeg21:Descriptor>
     <mpeg21:Component>
      <mpeg21:Resource mimeType="application/xml">
        <Mpeg7 xmlns= "http://www.w3.org/2000/XMLSchema-instance"
          type="complete">
         <UserPreferences>
          <UsagePreferences allowAutomaticUpdate="true">
           <FilteringAndSearchPreferences protected="true">
            <ClassificationPreference>
             <Genre href="urn:mpeg:GenreCS" preferenceValue="42">
               <Name>Acappella</Name>
             </Genre>
             <Genre href="urn:mpeg:GenreCS" preferenceValue="0">
               <Name>Acoustic</Name>
             </Genre>
             <Genre href="urn:mpeg:GenreCS" preferenceValue="4">
               <Name>Bass</Name>
             </Genre>
             <Genre href="urn:mpeg:GenreCS" preferenceValue="75">
               <Name>Classical</Name>
             </Genre>
. . .
             <Genre href="urn:mpeg:GenreCS" preferenceValue="63">
               <Name>Rock</Name>
             </Genre>
             <Genre href="urn:mpeg:GenreCS" preferenceValue="0">
               <Name>Techno</Name>
             </Genre>
             <Genre href="urn:mpeg:GenreCS" preferenceValue="0">
               <Name>Traditional</Name>
             </Genre>
             <Genre href="urn:mpeg:GenreCS" preferenceValue="0">
               <Name>Other</Name>
             </Genre>
            </ClassificationPreference>
           </FilteringAndSearchPreferences>
          </UsagePreferences>
         </UserPreferences>
        </Mpeg7>
      </mpeg21:Resource>
    </mpeg21:Component>
  </mpeg21:Item>
</mpeg21:Container>
</mpeg21:DIDL>
```

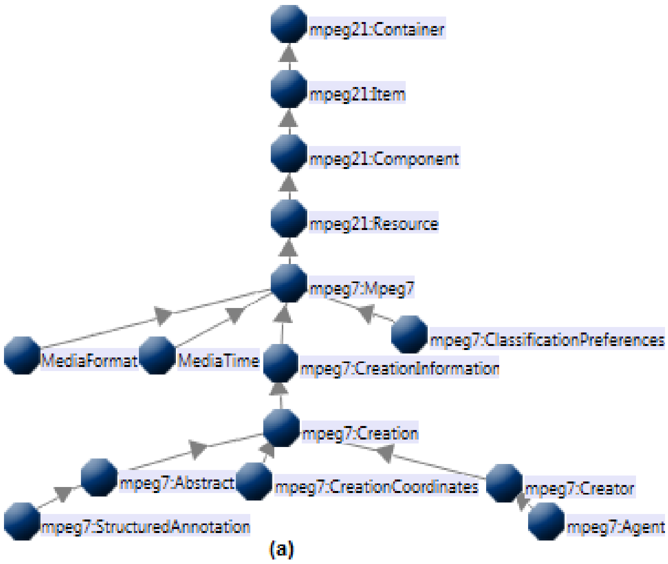

Figure 1. OWL ontologies about (a) audio metadata and (b) user preferences metadata

The suitable OWL ontologies which provide semantic descriptions about the metadata have also been created. Figure 1a presents the OWL ontology, which is used by the server for the audio metadata manipulation. On the other hand, the client uses its user preference metadata according to the OWL ontology that presented in Figure 1b.

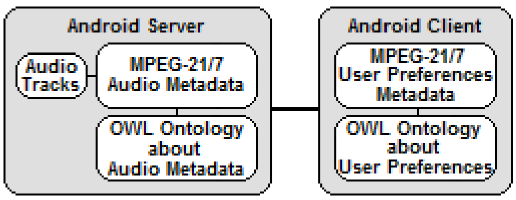

Figure 2. The basic modules of our architecture

Figure 2 presents the basic modules of our architecture. The client interacts with the server and sends a request which encapsulates a criterion. A criterion can be one of the following:

- An audio category, such as Pop, Rock, Jazz etc. (The client says "Promote me audio tracks that belongs to a specific audio category").
- The user preferences (The client says "Promote me audio tracks according to my preferences").
- The audio resources' usage history (The client says "Promote me the most famous audio tracks").

The server promotes audio tracks to the client according to its request. The client receives the response and appears the relative list which contains the promoted audio tracks. Then the client requests an audio track using this list. The server sends the audio track and updates the relative usage history metadata. On the other hand, the client receives the desired audio track and updates its user preferences metadata. The system operation is graphically illustrated in the sequence diagram of Figure 3.

When the client uploads a new audio track to the server, it also creates and uploads the relative user defined metadata. The server analyzes the uploaded audio track and extracts technical oriented metadata. Then, the server formats and inserts all the audio metadata into its metadata repository according to the relative MPEG-21/7 standards and to the OWL ontology.

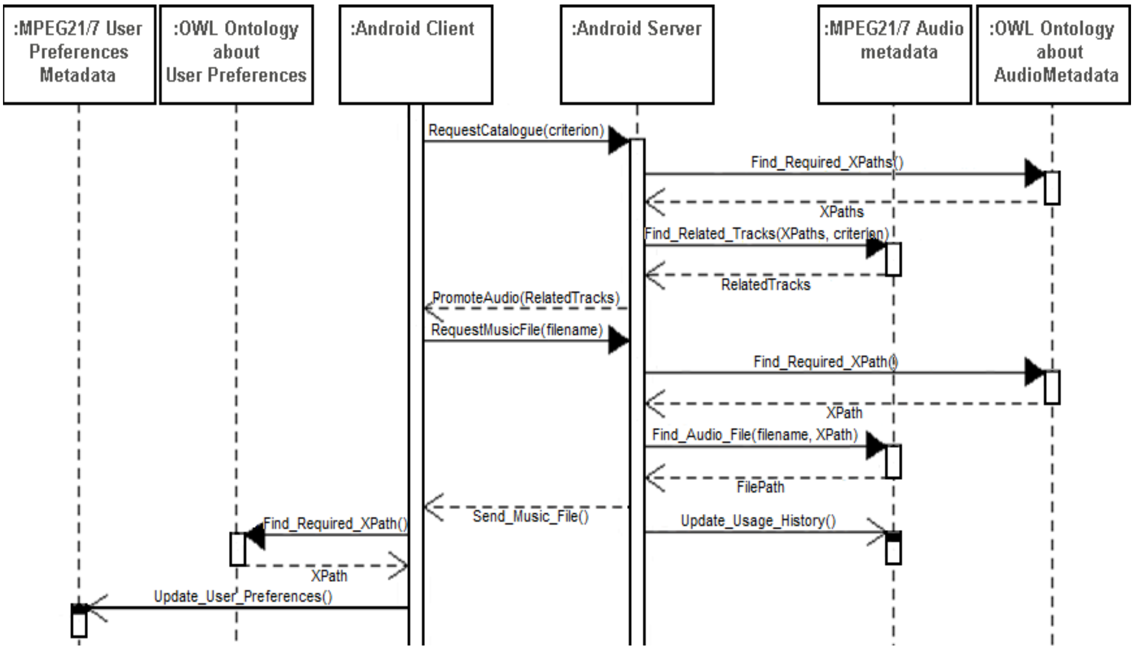

Figure 3. Personalized audio catalog retrieval and audio track request

The file upload operation is presented in Figure 4. The server's as well as the client's modules are developed using the Java Android API [13].

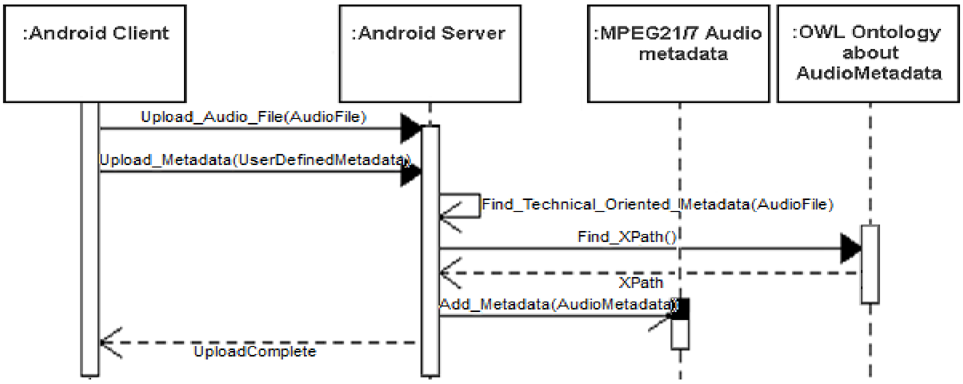

Figure 4. Audio track upload and audio metadata creation

RESULT AND DISCUSSION

This section presents an example of our system's functionality. Two Android virtual devices are emulated. One emulator instance runs as server and the other as client. The client interacts with the server and requests an audio track list according to a specific criterion (e.g. Pop). The server manipulates the audio metadata according to the relative OWL ontology and sends the desired list (Figure 5). Then, the client receives the list, selects and listens to an audio track.

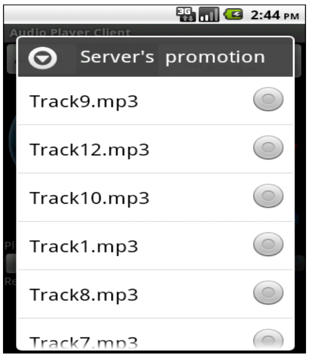

Figure 5. The client appears the proposed audio tracks' list

Consequently, the client's user preferences as well as the usage history in server's audio metadata are updated, according to the relative OWL ontologies. Figure 6 presents the respective user preference metadata block before and after the client's request. In this block, the 'preferenceValue' about the Pop genre becomes from 27 to 28.

```
before:
<Genre href="urn:mpeg:GenreCS" preferenceValue="27">
  <Name>Pop</Name>
</Genre>. . .
------------------------------------------------
after:
<Genre href="urn:mpeg:GenreCS" preferenceValue="28">
  <Name>Pop</Name>
</Genre>. . .
```

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

Figure 7 presents the respective audio metadata blocks before and after the client's request. The 'preferenceValue' of the 'CreationPreferences' block shows how much times has been requested the relative audio track -from all users- and becomes from 51 to 52. In addition, the 'preferenceValue' of the 'ClassificationPreferences' block shows how much times have been requested the tracks of the relative genre -from all users- and becomes from 84 to 85.

```
before:
<mpeg7:CreationPreferences>
  <mpeg7:Title mpeg7:preferenceValue="51" xml:lang="en">audiotrack.mp3</mpeg7:Title>
</mpeg7:CreationPreferences>
. . .
<mpeg7:ClassificationPreferences>
  <mpeg7:Genre mpeg7:preferenceValue="84" href="urn:mpeg:ContentCS:1">
    <mpeg7:Name xml:lang="en">Pop</mpeg7:Name>
  </mpeg7:Genre>
</mpeg7:ClassificationPreferences>. . .
------------------------------------------------
after:
<mpeg7:CreationPreferences>
  <mpeg7:Title mpeg7:preferenceValue="52" xml:lang="en">audiotrack.mp3</mpeg7:Title>
</mpeg7:CreationPreferences>
. . .
<mpeg7:ClassificationPreferences>
  <mpeg7:Genre mpeg7:preferenceValue="85" href="urn:mpeg:ContentCS:1">
    <mpeg7:Name xml:lang="en">Pop</mpeg7:Name>
  </mpeg7:Genre>
</mpeg7:ClassificationPreferences>. . .
```

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

Figure 8 presents the average response times for audio list proposal in respect of the tracks contained in the server. As the tracks number increases the response time increases as well, due to the information load processed by the server.

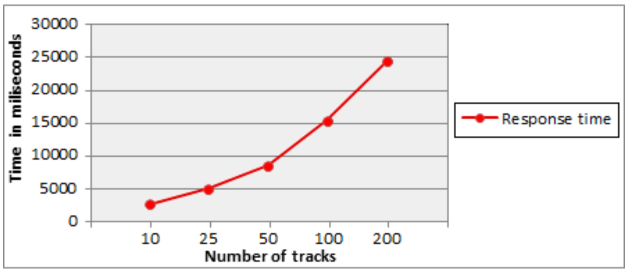

Figure 8. Average response times for audio track list proposal in respect of tracks' number

CONCLUSION

Our system consisted of a mobile audio server as well as a client implemented over the Android platform. The MPEG-21 and MPEG-7 standards are used to enhance the personalization process. In addition, the suitable OWL ontologies for metadata's manipulation have been created. The architecture is decentralized, in respect of each client stores and manipulates its own metadata locally. The server hosts the resource adaptation metadata along with the resources and proposes audio tracks to the clients according to specific criteria.

Future work includes the system's extension with SPARQL queries' execution capabilities as well as with the MPEG Query Format (MPQF). It will improve the personalization results and the user's experience.

References:

[1] Ivar Jorstad, Do van Thanh, Schahram Dustdar, "The Personalization of Mobile Services", WiMob 2005: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Vol. 4, pp. 59-65, IEEE, August 22-24, 2005

[2] Erich Gstein, Florian Kleedorfer, Robert Mayer, Christoph Schmotzer, Gerhard Widmer, Oliver Holle, Silvia Miksch, "Adaptive Personalization: A Multi-Dimensional Approach to Boosting a Large Scale Mobile Music Portal", Fifth Open Workshop on MUSICNETWORK: Integration of Music in Multimedia Applications, pp. 1-8, Vienna, Austria, 2005

[3] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl, "Explaining Collaborative Filtering Recommendations", ACM 2000: Conference on Computer Supported Collaborative Work, ACM, New York, USA, 2000

[4] Manolis Vozalis, Konstantinos G. Margaritis, "Enhancing Collaborative Filtering with Demographic Data: The case of Item-based Filtering", ISDA04: Fourth International Conference of Intelligent Systems Design and Applications, Budapest, Hungary, August 26-28, 2004

[5] 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

[6] 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

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

[8] 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

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

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

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

[13] Java Android API, http://developer.android.com/reference/packages.html