Metadata Integration and Media Transcoding in Universal-Plug-and-Play (UPnP) Enabled Networks

M. Jakab M. Kropfberger M. Ofner R. Tusch H. Hellwagner L. Böszörmenyi

Department of Information Technology
University Klagenfurt *

{Michael.Jakab, Michael.Kropfberger, Michael.Ofner, Roland.Tusch}@m3-systems.com
{Hermann.Hellwagner, Laszlo.Boeszoermenyi}@uni-klu.ac.at

Abstract

Universal Plug and Play (UPnP) is a widely accepted standard for automatically detecting devices and services in a local area network as well as for describing and controlling them. In order to deal with multimedia devices and especially content, in 2002 the UPnP-AV standard definition was released. It defines device and service descriptions for Media Servers and Renderers. Thereby, the Media Server's Content Directory Service allows an easy management and the exchange of metadata about the provided media data. Media content became browsable by semantic meta information about it.

There are still two major drawbacks of UPnP-AV, which make its usage in real world multimedia communication scenarios very difficult. First, searching for similar content on distributed Media Servers with a huge number of media files is not economically possible. Second, the media content must be consumed by Renderers as provided by the Servers, independently of their terminal capabilities and network connections.

In order to deal with these two drawbacks, this work proposes a novel approach of metadata integration and media transcoding in UPnP networks. First, the Media Server is extended by a Control Point which offers discovery of other Media Servers and fetches metadata from their Content Directories. Furthermore, it integrates the gathered information in its own Content Directory. Control Points are then able to query this Integrating Media Server for a desired content, and get a network-complete search result. Second, terminal and network capabilities of the Renderers are taken into account in order to transcode and transmit the content in a suitable way for the consuming device. These two approaches of metadata integration and media data adaptation enable searchable logical views on tailored multimedia content in UPnP-AV networks.

*This work was partially supported by the Austrian Science Fund (FWF) under project L92-N13 (CAMUS: Context-Aware Multimedia Services).

1 Introduction

Digital multimedia is becoming ubiquitous in our daily lives. This includes for example audio content such as MP3 or video content in various MPEG formats. A major problem is to easily find the desired multimedia content on devices or networked file systems accessible to the user, even if the content is correctly annotated with helpful metadata. *Universal Plug and Play Audio Visual (UPnP-AV)* [4] offers a widely accepted standard for automatically finding multimedia sources on the network. On-demand or live sources are provided by a *Media Server*. A *Control Point* queries this Media Server and initiates the playback on a *Media Renderer*, which is responsible for correct decoding and rendering of the media data. The Media Server uses the metadata for building searchable and logical views on the multimedia content, which are then browsable by the Control Points.

However, available UPnP-AV Control Points only offer access to one Media Server at a time. Thus, there is no common view on the content of all Media Servers in one unified structure. Instead, the content of each server is provided apart from each other. This implies that users have to comb the content of each server separately to get their desired content. This is not desirable for normal consumers since the user has to know on which server the desired media content resides.

This work introduces a novel approach of using UPnP-AV Servers to offer an integrated view over all available Media Servers on the network, in that the distributed multimedia content is merged into one virtual tree, which is well organized by the means of the available metadata. The user only has to select the artist or genre, without having to know where the real data resides. This requires mirroring the metadata onto and integrating it into a selected Media Server's Content Directory.

A further novelty is the extension of the UPnP-AV standard to cope with the capabilities and constraints of various end devices, including TV sets, WiFi-attached handheld PCs, and specialized UPnP-AV hardware devices. The requested media content is transcoded on the fly in order to fit the output system's characteristics.

The sequel of this work is organized into 8 sections. Following the introduction, section 2 underlines the general necessity of the proposed system in more depth by giving some example scenarios. Section 3 provides a basic introduction into UPnP, while Section 4 discusses UPnP metadata mirroring and integration. In Section 5, the main focus is on the implemented concepts and functionalities of the Integrating Mediatomb Media Server. Section 6 gives insights into the performance evaluation of the implemented extensions. Section 7 summarizes the main ideas, while Section 8 gives perspectives on future work.

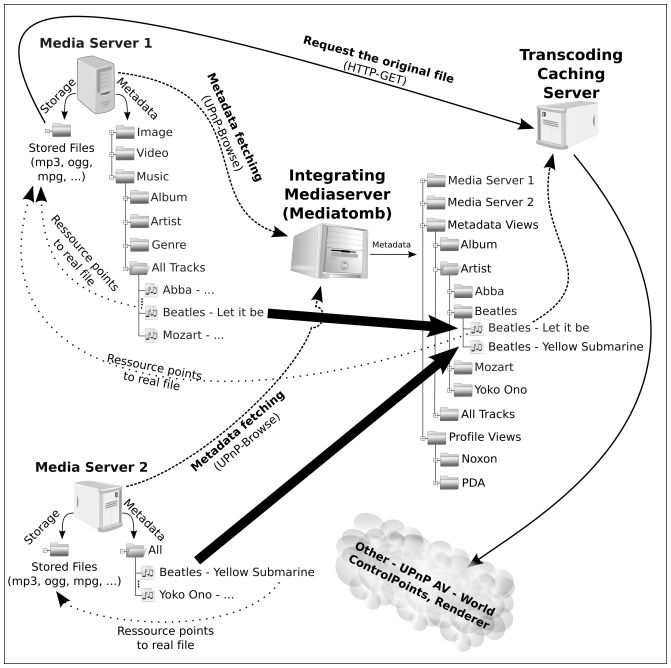

Figure 1. Example for Metadata Integration and Transcoding

2 Scenarios for Metadata Integration

The main idea of metadata integration is to offer a single source of media information for Control Points and to provide the best possible view on multimedia content. This feature is equally useful in mobile and in stationary scenarios.

2.1 Home Entertainment Systems

The first scenario covers home entertainment systems. In this scenario many different Media Servers are present within the same local area network. For instance, this could be the case in a campus network where students share their audio and video files via the network. With the help of UPnP-AV [4] it is easy to offer media distribution, without the need of significant configuration effort. UPnP-AV offers means to store, manage and exchange metadata as well as to control the actual audio/video delivery. Each Media Server provides its own *Content Directory* structure, and it is the responsibility of Control Points to display a view on the available Media Servers and their distinct content.

In order to illustrate this (see Figure 1), imagine a Media Server providing the song entitled "Let it be" and another Media Server providing the song "Yellow submarine", both composed by the artists "Beatles". A standard Control Point would browse both Media Servers and display each of their hierarchies separately. In order to find all songs of the "Beatles", all delivered Content Directory structures have to be examined independently.

Based on our concept of metadata integration, a new Content Directory structure, which, for example, uses the "artist" metadata element to create a container named "Beatles", is built. Within this container, all songs composed by the "Beatles" are collected from all available Media Servers. There is no need to search for songs created by this group elsewhere. With the help of UPnP, Media Servers broadcast their existance on start up and are then integrated into the metadata structure of an Integrating Media Server. If a contributing Media Server disconnects – for example when a Media Server residing on a mobile computer leaves the campus network – the metadata of this server is automatically detached from the Integrating Media Server.

2.2 Touristic Hotspots

Another field of application for metadata integration is mobile devices moving between different WiFi hotspots. For example, such a mobile device runs an Integrating Media Server and offers its own locally stored content. In addition, a special Control Point and a Media Renderer run on the mobile device. The Control Point is configured to coop-

erate with the Integrating Media Server running on the mobile device and only shows the local Media Server view (see Figure 2(a)). After entering a WiFi hotspot, the Integrating Media Server finds all other Media Servers and starts its integration process. The foreign metadata is included alongside with the local one so that an integrated view of both, local and foreign metadata, emerges (see Figure 2(b)). When the mobile device leaves the WiFi coverage, the previously integrated Media Servers and their metadata content are removed again. The locally stored media data remains accessible. For example, this is a realistic situation for a mobile tourist guide, which is used in a tourist region that cannot be covered by a single WiFi network. Rather, separate hotspots are located on different touristic objects of interest, and the content offered by the local servers are provided in a distributed manner.

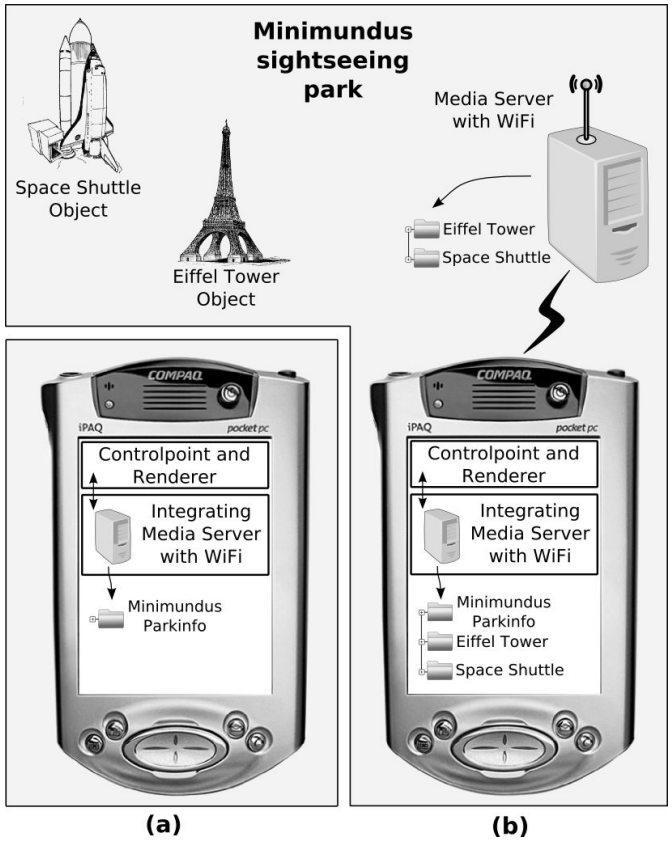

Figure 2. Touristic Hotspot: (a) Integrating Media Server with its locally stored content running on a mobile device; (b) Integrating Media Server on a mobile device integrating the content of the Minimundus Media Server within the local ones

Our research lab has developed a non-UPnP tourist guide, which is available in Minimundus, a themepark in Klagenfurt, Austria. The exhibition comprises about 150 small scale (1:25) models of famous buildings in the world. This tourist guide is implemented by using web programming languages and Internet protocols, and offers about 750 multimedia items, resulting from multi-lingual audio and video content covering 40 miniatures. The setup process of the system is complex and undesirable in terms of flexibility and portability. The UPnP-AV architecture takes out unnecessary complexity. The relatively small amount of 750 multimedia items can be integrated in terms of metadata with little effort in time and computational power.

The tourist guide system offers a *transcoding* facility which is used to adapt video and audio content to specific needs such as network bandwidth or device capabilities. In the case of high network load and resulting collisions, the ability to generate bandwidth-saving versions of the media content can be used to handle the network limitations.

Nevertheless, there are scenarios in which metadata mirroring and integration (see Section 4) do not scale properly. Metadata mirroring is problematic whenever a very large amount of items is going to be mirrored, and the devices which have to be mirrored are appearing and vanishing rapidly. In such a case, it is not possible for an Integrating Media Server to fetch all the foreign metadata in time, resulting in an incomplete global view. However, mirroring is a good choice for a network with many changing Media Servers which are carrying a small amount of files (touristic hotspots scenario) up to networks with Media Servers carrying a large amount of files lingering for a long period (home entertainment scenario).

3 Technological Background

Universal Plug and Play (*UPnP*) [4] enables for a platform and programming language independent discovery, description, subscription, eventing, and control of devices and services within local area networks. The UPnP technology makes use of existing Internet standards and protocols. In general, UPnP defines Devices and Control Points as components which communicate with each other.

In 2002, a new standard for audio and visual for UPnP (*UPnP-AV* [4]) was published. This standard defines device and service descriptions for Media Servers (i.e., devices carrying media files) and Media Renderers (i. e., devices playing media files). One important service for this approach is the *Content Directory Service*, which is located on Media Servers and stores fetched metadata from audio, image, and video files. The Content Directory organizes its metadata content in hierarchical form, similarly to a file system with directory structure, and makes this content available for Control Points. With a Control Point, it is possible to get a list of multimedia items from a Media Server and to

control playback on a Media Renderer.

The metadata communicated between the Media Server and the Control Point is encapsulated within a *DIDL-Lite* XML format [4, 2]. Listing 1 shows a DIDL-Lite response to a Browse Action sent from a Control Point to a Media Server.

```
<DIDL-Lite>
  <container id="100" parentID="10"
     childCount="7" restricted="1">
  <dc:title>Mozart</dc:title>
  <upnp:class>
     object.container.musicContainer
  </upnp:class>
</container>
<item id="101" parentID="10" restricted="1">
  <dc:title>Carolina In My Mind</dc:title>
  <upnp:artist>James Taylor</upnp:artist>
  <upnp:album>Greatest Hits</upnp:album>
  <upnp:genre>SoundClip</upnp:genre>
  <res size="3787266" duration="0:03:56.416"
      protocolInfo="http-get:*:audio/mpeg:*" >
      http://192.168.1.5:9001/disk/101.mp3</res>
  <upnp:class>
    object.item.audioItem.musicTrack
  </upnp:class>
</item>
</DIDL-Lite>
```

Listing 1. Example DIDL-Lite Fragment

Looking at the DIDL-Lite fragment of Listing 1, it is apparent that a container (e. g., directory) entitled "Mozart" and one item (e. g., file) entitled "Carolina In My Mind" is returned from the Content Directory service. Besides that, the item contains tags which describe the artist, the album, and the genre of the audio item. The resource tag (`res`) contains information about the MIME-type, the used transmission protocol and the URL of the raw media file. This meta-information is extracted from the original MP3 file. The user first chooses this certain item from the Control Point's user interface, and then the original file (provided via the `res` tag) is streamed and played on the Media Renderer.

The DIDL-Lite schema has a restricted number of possible metadata fields, which are imported from the UPnP and Dublin Core[1] namespaces. Usable descriptive information are stored in many digital video, audio and image files in addition to the multimedia contents. For instance, not only MP3/ID3 tags can be used, but also EXIF[2] extensions for JPEG files may be utilized. This descriptive information is extracted from media files and mapped to DIDL-Lite tags.

[1] Dublin Core Website: http://dublincore.org/documents/dces/

[2] Exchangeable Image File Format Website: http://www.exif.org

4 Metadata Mirroring and Integration

The UPnP standard explicitly allows the embedding of a Control Point into a device, which could be a Media Server or Media Renderer [4, 2, 1]. Most hardware UPnP-AV compliant Media Renderers are combined with a Control Point in order to offer a built in user interface. This work will show the advantage of embedding a Control Point into a Media Server in order to enhance its functionality. This server is able to discover other Media Servers and fetches metadata from remote Content Directories. This feature is referred to as *metadata mirroring*, defined by Intel [1]. We make a distinction between (1) metadata mirroring as just importing the exact remote metadata view and (2) metadata *integration* which is metadata mirroring with the enhancement of organizing the mirrored metadata into one unified view (see Section 5.1). In both cases the video, audio, and image files themselves are never copied and remain on the original server, until they are requested directly by the Media Renderer when it starts playing the content.

As shown in the scenario "Home Entertainment Systems" (see Section 2.1), metadata mirroring provides a Control Point with a single point of access to all items. This may offer an easier navigation for badly designed user interfaces where the switching between different Media Servers is cumbersome like on the hardware UPnP-AV Media Renderer DSM-320 from D-Link[3]. Besides, it is possible to provide UPnP-AV actions which the original Media Servers did not implement. Not every Media Server provides the *Search Action* which allows to easily search for content that matches some search criteria. Furthermore, if not only mirroring but also integration is used, a unified metadata view on e. g. all available songs from a certain artist is provided.

As shown in Figure 1 an example of integration in a UPnP network is given. Three Media Servers offer different multimedia content. Every multimedia metadata item of the Content Directory points to a real media file. The Integrating Media Server fetches the metadata from the other Media Servers. The resource metadata entries which point to the real file are not changed by the integrating server and therefore still point to the original location of the media file.

5 Integrating and Transcoding Media Server

This new approach of the Integrating Media Server provides arbitrary Control Points with a single point of information by integrating other Media Servers with many prebuilt views based on metadata which is included in the items (e. g.: artist or genre) and predefined profile views for special device capabilities. These offer on-the-fly content transcoding to meet the needs of specific Media Renderers

[3] D-Link Website: http://www.dlink.com

so that they do not need to transcode the media data on their own.

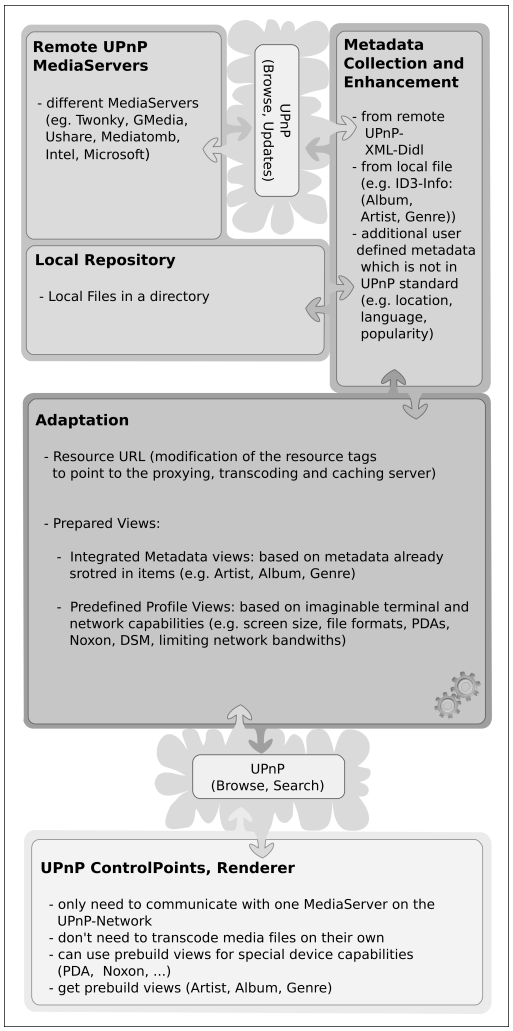

Figure 3. Conceptual View

The proposed system is able to access metadata from other Media Servers via UPnP and from its local data storage (see Figure 3). The collected metadata of both input channels is managed in the Content Directory and can be enriched with additional metadata which is usable e. g. for location coordinates for the Touristic Hotspot scenario (see Section 2.2).

5.1 Integrated Metadata Views

All local and remote metadata items are grouped by a number of suitable metadata categories like actor, album, author, artist, class (music, video, audio or text), date, genre, language and region, which are selected from the DIDL-Lite schema [1]. The selection is nearly congruent with the one of the TwonkyVision[4] Media Server, which also uses metadata views to improve the usability for Control Points.

Figure 1 illustrates an example for metadata integration in which a distinct set of music items is integrated. All of these items (e.g. different songs of a certain artist) are inserted into the same container side by side. In the case of an item where exactly the same metadata reside on two different servers there are two ways to resolve this conflict. The first idea is that all items are treated as if they were distinct items and are inserted alongside. Second, the same items residing on different Media Servers can be merged into one item on the Integrating Media Server. Each variation of the same item gets its own resource-tag, where multiple ressources for one item are within the standard of the Content Directory. In the second solution the responsibility of choosing among the different entries is given to the Control Point, while, in the first solution this is the user's decision. The implementation of the Integrating Media Server considers this first way of integration.

5.2 Predefined Profile Views

Since in a UPnP-AV standard implementation there is no way to obtain terminal capabilites from Media Renderers which could be used to transcode multimedia files to fit their needs optimally, the idea of predefined profile views is proposed. Terminal capabilities which enable dynamic adaption could be the screen size, network bandwidth, color capability, sampling rate or their known audio/video formats.

Basic terminal capabilities like supported video codecs could be acquired by binding the Control Point to a specific Media Renderer beforehand. Then the Control Point receives its terminal capabilities and proceed with all Browse Actions and Search Actions afterwards. In this case, metadata not fitting a Media Renderer could be hidden by the Control Point. Such a behavior is conceivable, but would stress the UPnP standard.

In fact, nearly every hardware UPnP-AV Media Renderer implements its own Control Point which hides undecodable items. If there is a request for a certain content which is not supported, the Media Renderer just does not react to the request or replies with an error message. For instance, the Noxon 2 UPnP-AV audio player[5] and the D-Link UPnP-AV audio/video player DSM-320 behave in this way.

The solution to this problem is to create views for known terminal and network capabilities (*predefined profiles*) in the Content Directory, where all media are filtered and the requested multimedia data are adapted to those capabilities. Useful input data for these predefined profiles would be the screen size, network bandwidth, color capability, sampling

[4]The TwonkyVision Website: http://www.twonkyvision.de/
[5]Noxon 2 Website: http://entertainde.terratec.net/

rate or known audio/video formats or any combination of those parameters.

For instance, to fit the needs of the Noxon UPnP-AV audio player which is not able to play any video content, a Noxon profile could be created. In this profile no videos would be listed, or only the audio stream could be provided after having been extracted from the video. Another example is a special PDA profile view in which the screen size, the sampling rate and the network bandwidth requirements are limited to fit the needs of these devices (see Figure 1).

5.3 Transcoding Capability

In order to fulfill an arbitrary Media Renderer's device capabilities, the adaptation of a full size video or audio is performed in a separate transcoding and caching server [3], which expects the location of the file encapsulated within an URL. Apart from the location, all transcoding parameters to fit the necessary device capabilities are added. On a request, the transcoding server fetches the original file from the original Media Server, performs a transcoding step and keeps the media content variation in the cache. On another request of the same file, the demand can be served out of the cache.

```
<DIDL-Lite>
  <item id="11" parentID="33" restricted="1">
    <dc:title>Sample Video</dc:title>
    <res size="4528266" duration="0:01:44.125"
      protocolInfo="http-get:*:video/mp4:*" >
      http://www.mytranscodingserver.com/?
        res=http://192.168.1.5:9001/disk/video.mp4&
        screensize=640x480&samplerate=10
    </res>
    <upnp:class>
      object.item.videoItem.musicVideoClip
    </upnp:class>
  </item>
</DIDL-Lite>
```

Listing 2. Modified DIDL-Lite Fragment

The URLs themselves are generated during the preparation of the Content Directory on the Integrating and Transcoding Media Server and are stored in the resource tags of each multimedia item. For a Media Renderer this step does not change anything in its usual standardized behaviour. It requests a transcoded media item like any other item, except that this one is streamed from the transcoding server since the resource tag of the media item points to it.

An example for an modified DIDL-Lite fragment is given in Listing 2. The previously explained modification takes place in the resource tag (`res`) of the item and indicates that the original file resides on the server with the IP address 192.168.1.5, the expected screen size is 640x480 pixel, and the sample rate amounts to 10 samples per second. Each Media Server expects the file on the transcoding server, without the need of any background knowledge about the behavior. A graphical illustration is shown in Figure 1. Microsoft's Media Connect [5] utilizes nearly the same idea of URL modification for transcoding.

5.4 Implementation

The implementation makes use of the open source Media Server *MediaTomb*[6] and uses the open source UPnP stack *Libupnp*[7]. The server is implemented in C++ and runs on the Linux operating system on Intel and StrongARM platforms. It is used as a code basis for all enhancements and is enriched with the functionality of a Control Point.

The mirroring and integrating functionality is implemented. Metadata from foreign Media Servers are mirrored and inserted into a separate container with the name of the origin Media Server under the Content Directory's root container. Furthermore, items are arranged in the integrated metadata and predefined profile views which allow to transcode audios and videos using the transcoding server.

6 Performance Evaluation

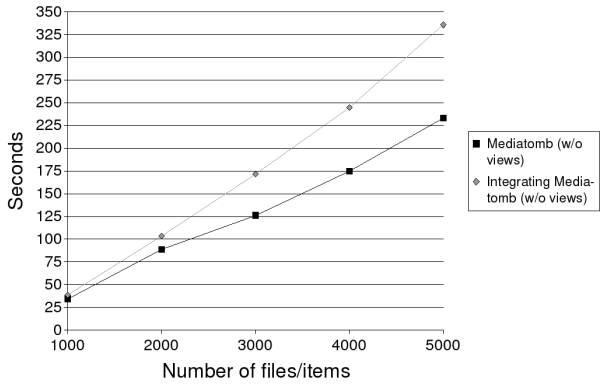

Figure 4. Initial building vs. UPnP integration

One of the most important aspects in examining networked item integration is its performance, compared with the time needed to build up the Content Directories from locally available files. In order to compare these two different sources of items, the number of files used for building the Content Directory in the local case and the number of items

[6] http://mediatomb.org/

[7] http://upnp.sourceforge.net/

which are integrated from another Media Server in the integrating case have to be the same.

Every test is performed against five testing repositories containing 1000, 2000, 3000, 4000 and 5000 items and is repeated ten times. In a home multimedia scenario, the number of items could be compared with 100 music albums with 10 songs each, followed by 200, 300, 400 and 500 respectively. The tests were performed on an insulated 100 Mbit ethernet network.

The first test measures the time it takes to build the Content Directory from local files. The second test measures the time it takes to integrate the same amount of items from an already set up foreign Media Server via the network. The measurement results are illustrated in Figure 4.

The metadata integration via the network with its network/UPnP overhead and its continuous UPnP-AV Browse Actions on average takes longer for these five repositories compared with the tests on local files. However, in a usual home entertainment scenario this overhead is in an acceptable range.

7 Conclusion

The ideas of mirroring and integrating address the shortcomings of the Universal Plug and Play (UPnP) standard and its extension for audio and visual (UPnP-AV) regarding an efficient multimedia communication in real-world usage scenarios like *home entertainment systems* and *touristic hotspots*. The two major drawbacks are: (1) Integrated views on the Content Directories of all UPnP Media Servers on a network are missing. This drawback does not allow for an efficient search of all occurrences of a desired media content especially in a server environment with a large number of media sources. (2) The terminal and network capabilities of Media Renderers are not taken into account on the Media Server side. Thus, a renderer consumes a server's media content in a quality which may not be suitable for its terminal and network capabilities (e.g., a high bit-rate video stream for a resource-constrained device like a PDA).

These two major problems are solved by using the Integrating and Transcoding Media Server. On the one hand, this server acts as a normal Media Server with its own Content Directory. On the other hand, it embedds a Control Point in order to collect metadata from the Content Directories of all the other Media Servers on the network. After the mirroring step, the metadata is integrated into the Content Directory of the Integrating Media Server. A unified view on all media content which is available in the UPnP-AV network is generated. This represents a central "database" for browse/search queries of Control Points. Browse/Search queries may target the topics of the integrated views, which resulted from metadata integration, such as artist or genre in the case of music content. These queries embrace all of the available metadata on the network.

A further novelty is the possibility to adapt content to the capabilites of various predefined end devices, including TV sets, WiFi-attached handheld PCs, and specialized UPnP-AV hardware. By offering predefined profiles for these devices, adaptions can be carried out without breaking the UPnP-AV standard. The live transcoding itself takes place in a separated transcoding and caching server.

8 Future Work

In future it is suggested that the modification of the resource tags, needed for transcoding is moved to a specialized Control Point which gets the capabilities of Media Renderers and therefore enables dynamic transcoding to their special needs.

Furthermore, the optional Search-Action of the UPnP-AV standard in the Integrating Media Server should be integrated in order to make it easier for Control Points to find specific items. In addition, it was found that metadata mirroring would work much faster if there was a special vendor specific action which delivered a flat item hierarchy (one container with all items in it) without any categorization on metadata. Currently, the same item is listed in different views, e.g., one specific song of the Beatles will be inserted in the artist view and also in the album view. This makes mirroring difficult and causes long and redundant runs for the mirroring process. The proposed vendor extension would lead to great improvements.

References

[1] Intel Corporation. Designing a UPnP-AV MediaServer. Technical report, Intel Corporation, 2003. http://www.intel.com/cd/ids/developer/asmo-na/eng/downloads/upnp/documents/index.htm.

[2] Intel Universal Plug and Play Technology, May 2006. http://www.intel.com/cd/ids/developer/asmo-na/eng/downloads/upnp/overview/index.htm.

[3] P. Schojer, L. Böszörmenyi, H. Hellwagner, B. Penz, and S. Podlipnig. Architecture of a Quality Based Intelligent Proxy (QBIX) for MPEG-4 Videos. In *Proc. Twelfth International World Wide Web Conference*, pages 394–402. ACM, 2003.

[4] Universal Plug and Play Forum, May 2006. http://www.upnp.org/.

[5] D. Wrede. How to Build a Network Device Compatible with Windows Media Connect. Technical report, Microsoft Corporation, July 2004.