Semantically Enhanced, Multifaceted, Collaborative Access to Cultural Heritage

Project IST-034984

D1.2 Methodology for Access to Distributed Web-based Resources

Specification

Editor:	Daniel Vladušic
Contributors:	Fabian Grzeszeyk, Maria Teresa Andrade, Nahum Korda
Dissemination Level:	Internal
Version:	1.0
Date:	30.06.2008

Executive Summary

Document Objective

This document is addressing the work being developed in task T1.4 – Definition of interfaces to multiple MOSAICA resources. The document aims to providing a broader picture on the use of the resources in the MOSAICA system: how resources are uploaded into the system, how they are stored and how they can be retrieved. The work developed in T1.4 has also focused on the selection of a uniform set of user-uploadable resources, from the technological point of view. Accordingly, this deliverable also provides details concerning types of media (video, image, audio, etc) and encoding formats. The transcript from DOW is given below.

T1.4 Definition of interfaces to multiple distributed MOSAICA resources. The access to Web resources is not a trivial problem, in particular not for the resources residing in the so-called *Deep Web*. The major difficulty is hereby the wide variety of potential access methodologies required to encompass a broad range of Web-based resources, which may include Web Services, multimedia files, database and repository interfaces, streams, etc. For each class of resources an interface allowing dynamic access to it must be specifically developed. This effort is closely related to T3.1, Distributed Multimedia Content Management. A first analysis of the stakes encountered with the selected Web-based Resources will be integrated to deliverable *D1.1- Selection of Web-based Resources in Jewish Heritage* on month 12. The methodology implemented for addressing these stakes will be documented on month 24 in the deliverable *D1.2- Methodology for Semantic Access to Distributed Web-based Resources*.

Targeted Audience
MOSAICA technological partners

Prerequisite
IST-034984 MOSAICA DoW document

© 26.11.2007 MOSAICA

This document contains proprietary information of MOSAICA and shall not be reproduced or transferred to other documents, disclosed to others, or used for any purpose other than that for which it is furnished, without the prior written consent of MOSAICA.

The trademark and service marks of MOSAICA, including the MOSAICA mark and logo, are the exclusive property of MOSAICA, and may not be used without permission. All other marks mentioned in this material are the property of their respective owners

Summary

This document describes various possibilities towards accessing the multimedia content in MOSAICA. Namely, given the distributed and web-based architecture of MOSAICA, it can be assumed that content will be presented on various devices, ranging from mobile phones to normal personal computers. This puts high pressure on guaranteeing an enjoyable user experience from the technical point of view, as we must take different bandwidth, screen resolutions, etc. into account.

The current state in the field of formats of the multimedia content shows scattered use of various encoding engines, which require different software for presentation of the content. While the sound and images have relatively stabilized with MP3, OGG and JPEG and PNG standards, the same cannot be said for video. The formats for encoding the video differ significantly, hence the unification of the video clips, uploaded to MOSAICA must be done. To this end, we have explored options towards transcoding the original user-generated content into one unified format. Here we chose the Flash Video Format, as it is the most widespread format for streaming video content over the web. We also present a modification of the YouTube model, which is one of the relevant models as it is successfully used and adopted by the end-users.

The options for execution of the transcoding step have also been explored. The simplest solution from the development and maintenance point of view is to have a specialized transcoding server, offering the transcoding service. The utilities that were explored for this step, were chosen according the criteria of source code accessibility and cost. Namely, these utilities can be optimized for the hardware that they use by compiling them and selecting appropriate target architecture. Furthermore, by using free open-source software we lower the cost of running MOSAICA.

Presentation of the third party content (i.e., not user-generated non-copyrighted content) has also been explored, but this content leaves us little choice on how to be presented. Still, we think that transcoding step can, to some extent, be performed on-the-fly.

Finally, we present conclusions which will be adopted in MOSAICA for accessing the multimedia content.

Contents

1. Introduction 5

2. Technical view on multimedia content presentation 5

3. Presentation of multimedia content on web 7

4. User-generated content in MOSAICA 10
   4.1. Transcoding execution 10
   4.2. Delivery of content towards the end user 15

5. 3rd party content in MOSAICA 16

6. Security in MOSAICA 17

7. Conclusions 17

1. INTRODUCTION

This document is addressing only the task T1.4 – Definition of interfaces to multiple MOSAICA resources. However, the Description of Work states the following: "The results [of the task T1.3 Semantic annotation of selected resources] will be documented in an internal 'Semantic Annotation of Selected Resources' document that will be maintained throughout the project life, until month 27. On month 24, examples will be extracted from this document and will be used as illustrations in the deliverable D1.2- Methodology for Semantic Access to Distributed Web-based Resources."

Nevertheless, as soon as we started preparing this deliverable, we discovered that although in the same workpackage, these two tasks handle differing aspects of the access to Web resources, and that it would be very difficult to reconcile them in a single, coherent deliverable. While the task T1.3 handles the preparation of the resources for the semantic access by providing semantic annotations to them, the task T1.4 handles merely the technical problems of various encodings and metadata schemas used for the description of the multimedia resources.

Moreover, in parallel with this deliverable, another deliverable reporting on the progress in the workpackage WP2 was issued – D2.2 Semantic Annotation. That deliverable presents in detail the entire framework for semantic annotation, and incorporates a section introducing the MOSAICA Semantic Annotator (section 3. Semantic Annotator). That section included "examples […] extracted from this document [i.e. an internal 'Semantic Annotation of Selected Resources' document that will be maintained throughout the project life, until month 27] and […] used as illustrations" – as the DoW states. Accordingly, we decided that a more coherent deliverables will be accomplished if we leave the extracts from the internal 'Semantic Annotation of Selected Resources' document in the deliverable D2.2 Semantic Annotation, and consequently focus the deliverable D1.2 on the task T1.4 only. Consequently, we also removed "semantic" from the title of the deliverable D1.2.

The deliverables D1.1 Selection of Web-based resources in Jewish Heritage, D2.1 Conceptual Alignment of Multiple Distributed Content Sources, D3.1 Mosaica Distributed Content Management System and D5.1 System & Application Specification all use and show various uses of the various resources (heritage items). This deliverable will use the knowledge, descriptions and specifications from the above mentioned deliverables and provide a broader overview on management and use of the resources within MOSAICA.

We start with an overview of the current state-of-the-art on the end-terminal side (i.e., screen resolutions, connectivity, etc.). Then, we address the presentation of the multimedia content on the web and in the following sections describe ways of handling the multimedia content based on its origin (user-generated or copyrighted) in MOSAICA, respectively. Finally, we draw some conclusions.

2. TECHNICAL VIEW ON MULTIMEDIA CONTENT PRESENTATION

This section presents an overview of the use of multimedia content within MOSAICA. MOSAICA media types encompass images, sound and video in various formats. In MOSAICA, we aim to present multimedia content on the WWW (i.e. in browser environment), therefore the following problems need to be solved:

- **browser compatibility:** the content (HTML/CSS) can be handled differently by different browsers. Accordingly, some Web sites may be presented to the user in a way different from the originally intended.
- **client compatibility:** the way multimedia content is presented on Web browsers, greatly depends on the type of browser and local plug-ins. As there is a great number of possible computer configurations, we cannot assume that clients will have installed all the required plug-ins for the multimedia content MOSAICA is offering.
- **bandwidth optimization:** a common assumption is that the bandwidth towards the client is infinite. Usually, the more artistic website designers take the bandwidth for granted, hence the loading of the Web sites can be slow. Furthermore, as the Web services, used for example for hit counting, are included into the web page, the delays when loading the Web page can significantly rise.
- **rendering optimization:** with the rise of mobile computing a variety of screen resolutions is emerging. For example, the newer mobile phones have the screen resolutions 240x320 pixels, which correspond to most Windows Mobile[1] devices. Still, the mobile phones can have different screen resolutions, even when comparing devices from the same manufacturer (Nokia N800[2] (800x480), Nokia N95[3] (240x320), Nokia E90[4] (800x352), iPhone[5] (480x320)). Fortunately, almost all Symbian based mobile devices use Opera Mobile[6] as their primary browser, while Windows Mobile devices use Interner Explorer Mobile[7] and iPhone Safari[8]. The screen resolutions on personal computers do not vary that much, and we can safely assume the lower bound being at 800x600, or 1024x768. The majority of PC users use Firefox[9] or Internet Explorer[10].

Commentaire [mandrade1] : his text relates more with bullet 1 "browser compatibility" than with "rendering optimization". It seems to me that it should appear in the first bullet.

These topics show that, even nowadays, the presentation of multimedia content in a Web environment is not trivial. We have therefore devised an approach to overcome these problems in MOSAICA. The next paragraphs present this approach and the reasoning behind the decision to, given the above described problems, base the presentation of the MOSAICA multimedia content in browser and more importantly, using the formats that are widespread in the browser environments.

In order to present multimedia content locally, we have to install appropriate multimedia player. While this is a trivial task, the selection of appropriate software, needed for decoding of the multimedia files, is not. At this point, the software that was used to encode the multimedia content becomes extremely important, as the process is reversed when presenting the content. Hereafter, we shall refer to software that is used for encoding and decoding multimedia content as codecs[11].

Furthermore, this software is different for every operating system. Let us illustrate this fact by describing the default settings for multimedia players (and codecs) for the three major personal computer operating systems:

1 http://www.microsoft.com/windowsmobile
2 http://europe.nokia.com/A4305063
3 http://europe.nokia.com/A4494366
4 http://europe.nokia.com/A4346043
5 http://www.apple.com/iphone/specs.html
6 http://www.opera.com/products/mobile/
7 http://www.microsoft.com/windowsmobile/software/iemobile.mspx
8 http://www.apple.com/iphone/features/index.html#internet
9 www.mozilla.com/firefox/
10 http://www.microsoft.com/windows/products/winfamily/ie/default.mspx
11 Codecs are used in order to save space when storing multimedia content. The uncompressed images, sound or video files consume large amounts of space (and bandwith, when transferred over the network), hence codecs are applied to compress the content. Codecs are further divided into lossless and lossy codecs, where the first group compresses the content without any loss of information and the second group achieves greater compression ratios and hence smaller file sizes but does not allow to decompress and reconstruct an exact copy of the original content, in what concerns the quality. Within MOSAICA, we shall use lossy codecs.

- **Windows (2000/XP/Vista)**: WMA[12] and MP3[13] for audio and WMV[14] for video. Furthermore, the default installation supports a huge variety of codecs, but at the same time the DivX[15], XviD[16], AC3[17], etc., which are most commonly used as network transfer formats, aren't supported.
- **Linux (Ubuntu/Debian/...):** OGG[18] for sound. Some distributions don't even support MP3 because of the licensing issues. Still, usually XviD format for video is supported.
- **MacOSX (10.4.x, 10.5):** AAC[19] and MP3 for sound and QuickTime[20] for video.

Given the described situation, where the two major operating system vendors are pursuing their own proprietary multimedia content formats and Linux is trying to avoid licensing issues, we can see that default installations of multimedia codecs are quite different. Hence, the users usually install codec packs. On Windows platform, the codec pack is usually the Combined Community Codec Pack[21], while on MacOSX and Linux, the users usually install the Mplayer package[22], which includes (albeit some of them are proprietary) almost every codec imaginable.

In MOSAICA, we cannot expect the users to install 3rd party software, as they may not have such rights on their computer and also they may not have such technical knowledge.

Let us summarize the issues encountered with respect to the multimedia content presentation:

- local machine based multimedia content presentation:
  - OS-vendors default installation supports mainly only proprietary multimedia formats,
  - downloading of files locally from web application is not elegant,
  - local installation of software is not always possible.
- browser based multimedia content presentation:
  - browser compatibility,
  - client compatibility,
  - bandwidth optimization,
  - rendering optimization.

The above points show that, given the plethora of different configurations (operating system, multimedia player, underlying codecs), presenting the multimedia content locally can be a non-trivial task (and not elegant). Therefore, the decision to base the presentation of the multimedia content in the browser was taken, despite the different set of issues, resulting in an attempt to reach the broadest possible audience.

In the following sections, we describe a similar Web-based model and its adaptation for MOSAICA.

3. PRESENTATION OF MULTIMEDIA CONTENT ON WEB

As discussed in previous sections, using multimedia resources that were created by different authors brings forward the problem of the different encoding and file formats, which were used for encoding of the multimedia content. Codecs must be installed on the client device where the

12 http://www.microsoft.com/windows/windowsmedia/forpros/codecs/audio.aspx
13 http://www.iis.fraunhofer.de/EN/bf/amm/mp3history/mp3teame.jsp
14 http://www.microsoft.com/windows/windowsmedia/forpros/codecs/video.aspx
15 http://www.divx.com/
16 http://www.xvid.org/
17 http://ac3filter.net/
18 http://www.vorbis.com/
19 http://www.apple.com/quicktime/technologies/aac/
20 http://www.apple.com/quicktime/
21 http://www.cccp-project.net/
22 http://www.mplayerhq.hu/design7/dload.html

multimedia content is to be presented (a personal computer, a PDA, a mobile phone, etc). One possible solution to enable the same content to be presented in different terminals that exhibit distinct characteristics and capabilities, is beforehand transcode the original content into a more widespread and interoperable format. One example where this approach is used is the YouTube[23] website. The YouTube was selected as it handles the most difficult type of multimedia content, i.e. video.

YouTube uses the following approach:

- the user uploads the content, encoded using one of the better known codecs[24];
- the content is transcoded into the Flash Video[25] format (FLV for short);
- the content is then publicly made available at www.youtube.com .

Schematically, the approach is shown in Figure 1.

Let us now briefly describe the advantages of using the FLV format. This format has become widespread with the advent of Web sites like YouTube, Google Video[26], Reuters.com[27], Yahoo! Video[28], MySpace[29], etc. One of the main advantages of using FLV is the widespread support that it has from various browsers and systems[30]. Accordingly, our goal of reaching the widest possible audience, can thus be satisfied[31]. In addition, users are getting more and more accustomed to FLV presentations. Furthermore, as FLV is becoming a de-facto standard for Web presentations, the support for it from the technical view point is already significant and it is expected that it will even further increase.

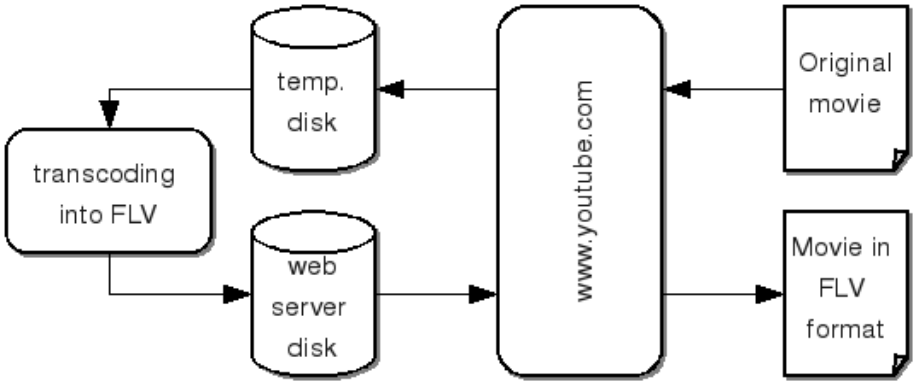

Illustration 1: The YouTube transcoding approach.

Given all that has been described above, it was then decided in MOSAICA to adopt an approach for the presentation of multimedia content, similar to the solution used by YouTube. For that

23 www.youtube.com
24 The upload formats are described at:
http://www.google.com/support/youtube/bin/answer.py?hl=en-ch&answer=55744
25 http://www.adobe.com/devnet/flash/video.html
http://en.wikipedia.org/wiki/FLV
26 http://video.google.com/
27 http://www.reuters.com/
28 http://video.yahoo.com/
29 http://www.myspace.com/
30 http://www.macromedia.com/software/flash/about/
31 If Flash Player is not installed, the installation is automated on majority of systems that are used by computer illiterate users.

reason, the next sections present the adapted YouTube model for the MOSAICA user-generated content and also the MOSAICA model for multimedia content imported from 3rd party Web sites.

4. USER-GENERATED CONTENT IN MOSAICA

This section describes the handling of the user-generated content that is uploaded to the MOSAICA system. In this situation, it can thus be safely assumed that here are no IPR issues involved – the content can be freely distributed among authorized users (defined by groups in MOSAICA and supported by the p2p-CMS system).

The MOSAICA approach to deal with this kind of content – copyright free material – is similar to that of YouTube as indicated in the previous section. However, some modifications are introduced due to different tagging (marking of the content) model, data storage model and due to various uses within MOSAICA.

The upload process into MOSAICA system consists of the following steps:
- Using one of the interfaces offered by the MOSAICA Semantic Layer, the user logs in the MOSAICA system;
- the user annotates the resource he/she wants to upload to MOSAICA, by selecting, via the Semantic Annotator, the appropriate ontology;
- the user uploads the resource into the p2p distributed content management system of MOSAICA, via the Semantic Layer;
- the p2p-CMS acknowledges the uploaded resource by providing an resource identifier (returned to the Semantic Layer as a MOSAICA URI)

The schematic overview of the upload steps is shown in Figure 2. The user-generated content is semantically annotated, uploaded[32] into p2p-CMS through the MOSAICA web site and finally, the MOSAICA URI (Content ID) is returned.

The adapted YouTube model for MOSAICA is shown in Figure 3. When compared to the schematic overview of the MOSAICA upload process, we can note the added component – the transcoding step, which will unify the formats of the user-generated content, stored in MOSAICA p2p-CMS. The transcoding step is shown in a cloud (and not as an explicit component) for the following reason: the transcoding of images/video/audio is a CPU intensive process, for which MOSAICA project will need to find an appropriate way of execution.

4.1. Transcoding execution

There are basically two different possible transcoding models or alternatives for the implementation of the transcoding step: a *client-based* solution and a *service-based* solution. The later approach can be further achieved by adopting either *a server-based* approach or a distributed, *p2p-based* approach:

- Client-based solution - client-side (local) execution of the transcoding step;
- Service-based solution:
  - MOSAICA dedicated transcoding server;
  - p2p inspired transcoding solution.

The execution of transcoding step on the client-side, i.e. on the computer of the regular MOSAICA user, puts no stress on the MOSAICA system, but at the same time requires the user to download transcoding software, install it and execute the transcoding step. While the software for transcoding may be quite easy to download and to execute for technically skilled users, we must acknowledge that this step can be overwhelming for computer illiterate ones. Also, the users may not have rights to install and execute third party software locally. Another problem that can arise with the client-based transcoding solution, can be the limited processing power of some client

32 The upload process into the p2p-CMS is described in D3.1 Mosaica Distributed Content Management System – Figure 17.

devices, which can be quite outdated (for example small-town library computers are usually quite old and slow).

Table 1 lists the possible different options for the implementation of a transcoding utility in MOSAICA, providing an indication of advantages and drawbacks of each one of them.

The adoption of the client-based transcoding solution requires that transcoding software is locally installed on the client's computer. This implies that appropriate packaging and graphical user interfaces are developed and provided for all major client platforms. Whereas this solution could be easily implemented within MOSAICA, it does not seem feasible to scale it to widespread audiences, providing and maintaining the required technical support.

On the other hand, the MOSAICA dedicated transcoding server, or a p2p-based transcoding solution, would hide the transcoding step from the normal users, thus overcoming these difficulties (as in the case of the YouTube model). The transcoding utility could be seen as a service offered by MOSAICA to its clients. Accordingly, one of these two approaches should be adopted in the MOSAICA project. The basic difference between the two solutions is in the resources that need to be made available and specifically allocated for providing the transcoding service. If we choose the dedicated server, then the MOSAICA content provider will need to make available an additional server with sufficient processing power. On the other hand, if we choose the p2p inspired solution, the dedicated server is not needed, but its implementation (which resembles the Seti@Home[33] project) is more complex, requiring additional effort from MOSAICA developers. Additionally, its successful deployment requires that the MOSAICA users agree to share their processing power with other users. Optionally, a hybrid solution could be envisaged, whereby dedicated low cost computers could be placed inside the MOSAICA p2p network, offering their full capacity for the execution of transcoding operations. This hybrid solution, although requiring dedicated machines, would lower the costs of a centralized solution, providing at the same time more reliability and availability, as well as transparency for the users.

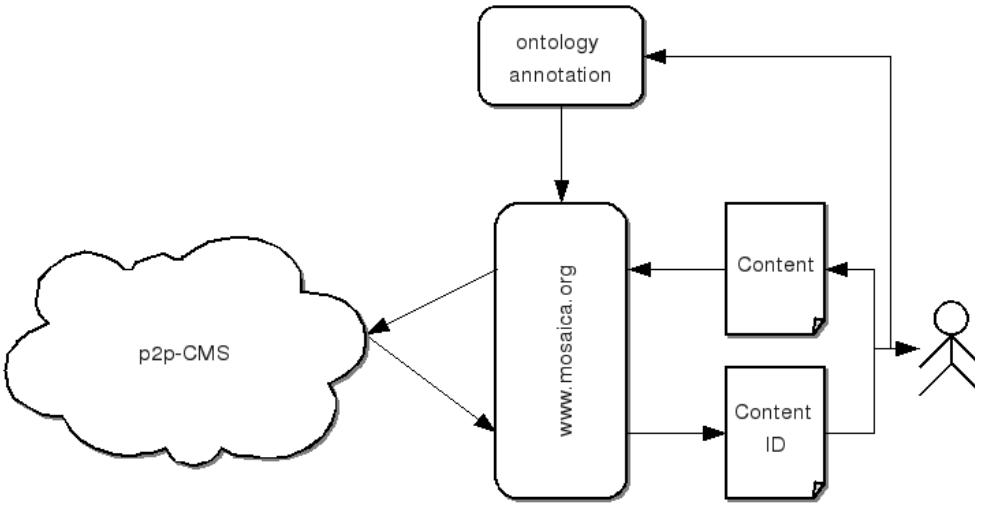

Illustration 2: Uploading content into MOSAICA system.

33 http://setiathome.berkeley.edu/

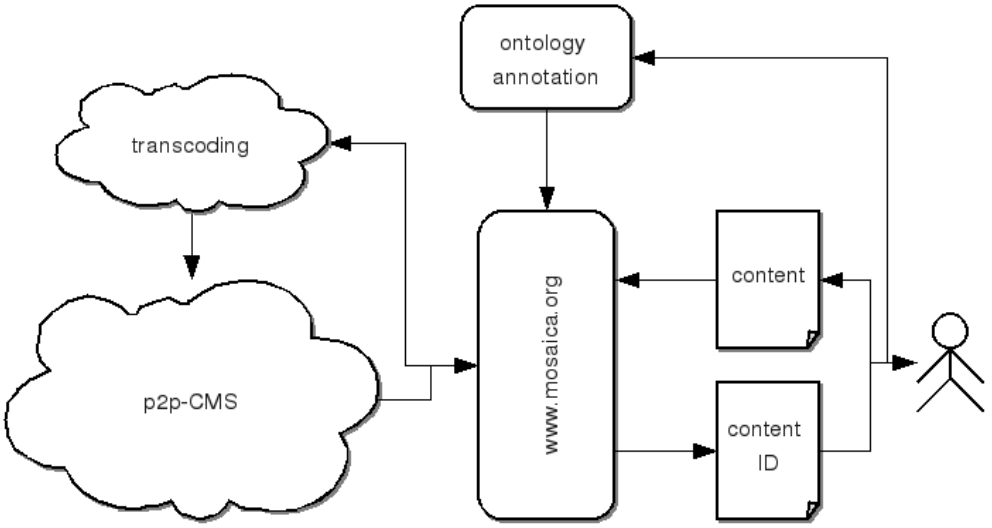

Illustration 3: Uploading content into MOSAICA system with transcoding step.

Transcoding type	Pros	Cons
local	– no stress on the MOSAICA system – no additional resources required	– development of setup packages for the transcoding software – users must download the software – users must have appropriate rights to install the software – users must know how to use the transcoding software
transcoding server	– transcoding software is installed and maintained on only one machine – transcoding software may be optimised for that particular machine	– additional resources required from MOSAICA – introduction single point of failure (if there is no replication)
p2p inspired transcoding	– additional resources required only for development efforts – no additional stress on MOSAICA resources (in terms of computers) – efficient use of already available resources	– development of seamless installations of the transcoding software on different machines – development of p2p technology for executing transcoding in distributed manner – reliability issues in terms of installed software and availability of nodes (the latter is insignificant, when number of nodes is big) – require agreement from MOSAICA users to donate their CPU time

Table 1: Pros and cons of using different transcoding types.

Assuming, that the model of transcoding has been chosen, we turn our attention towards the actual transcoding tools that will need to be deployed regardless the adopted model. We have identified a number of possible candidates for the different media types that exist in MOSAICA (images, sound and video). Table 2 lists the applications that were selected for each media. Images should be converted using the ImageMagick[34] package, audio with SoX[35] and video with FFMPEG[36] package. The major reason for this selection is the fact that all these packages are freely available and can be downloaded and used at no cost. They can also be used directly from the command line, which enables their utilization as batch conversion utilities or server-side scripting – an example of such server-side script for conversion of movie.avi into movie.flv is given below:

```
$ ffmpeg -i movie.[avi] -s 320x240 -ar 44100 -r 12 movie.flv
$ cat movie.flv | flvtool2 -U stdin movie.flv
```

34 http://www.imagemagick.org/script/index.php
35 http://sox.sourceforge.net/
36 http://ffmpeg.mplayerhq.hu/

Furthermore, as the source code for these utilities can be freely downloaded, we can utilize this fact to our advantage and gain CPU performance by compiling the source code and targeting our specific hardware.

Type of content	Package	Supported input formats	Output format
images	ImageMagick	JPG, PNG, GIF, BMP, other[37]	PNG/JPG
audio	SoX	WAV, FLAC, OGG, MP3, other[38]	MP3
video	FFMPEG	DivX, WMV, QuickTime, other[39]	FLV

Table 2: An overview of the proposed tools for multimedia content conversion. All tools have a plethora of various conversion options, hence we include also the links to the sites with all possible input formats.

Finally, we present another option for video encoding. Given a variety of possibilities regarding the resolution of the client's computer, his bandwidth, etc., we will employ the MPEG-21 standard. This standard enables use of containers in which resides the differently encoded Flash Video Format encoded content. This content will be encoded using different standard screen resolutions and different bit rates, enabling smooth playback even on mobile devices using EDGE protocol. Furthermore, MPEG-21 enables the use of Digital Rights Management (DRM) information.

MPEG-21 is an ISO/IEC standard that describes how the various elements of the multimedia content delivery chain fit together to create a truly interoperable multimedia framework, where content can be transparently consumed in heterogeneous environments with varied terminal devices and across different networks. In MPEG-21, content is conveyed as Digital Items (DIs), which can be seen as complex digital objects, composed of a number of media components (audio, images, video) and associated metadata to indicate the structure of the digital object and the relations between its components. The objective of MPEG-21 is to provide a platform to enable users to access multimedia content in a transparent way. This standard is currently composed of 17 parts: 1 - Vision, Technologies and Strategy; 2 - Digital Item Declaration (DID); 3 - Digital Item Identification and Description (DII); 4 - Intellectual Property Management and Protection Components (IPMP Components); 5 - Rights Expression Language (REL); 6 - Rights Data Dictionary (RDD); 7 - Digital Item Adaptation (DIA); 8 - Reference Software; 9 - File Format (FF); 10 - Digital Item Processing (DIP); 11 - Evaluation Tools for Persistent Association; 12 - Test Bed for MPEG-21 Resource Delivery; 14 – Conformance; 15 - Event Reporting (ER); 16 - Binary format; 17 - Fragment Identification for MPEG Media Types; 18 – Digital Item Streaming.

In MOSAICA, we are interested in using MPEG-21 to provide the means to present complex multimedia objects to the end user, adapted to the characteristics of the context of the user, in a transparent, automatic way, while preserving the intellectual property. Accordingly, the parts of the MPEG-21 standard that are more relevant to the MOSAICA project are:

– MPEG-21 Part 2 – Digital Item Declaration Language (DIDL)
  This part defines containers, which group various items. In our case, the items could be videos encoded in different “formats”, presenting the same content from the semantic point of view, but with different resolutions, bit rates, etc. An MPEG-21 DI can be a complex object encompassing a great variety of media and metadata. Part 2 of the standard specifies the Digital Item Declaration Language (DIDL). This language is used to generate the Digital Item Declaration (DID) which can be seen as the virtual container. The DID is an XML document, which describes the structure of the DI. Inside the DID, standardized MPEG-21 mechanisms are used to either convey the resources embedded directly in the DID (inline resources), or to provide information on the location of the resources that make up the DI. As the DID may contain a great diversity of components, its declaration can be quite complex.

37 http://www.imagemagick.org/script/formats.php
38 http://sox.sourceforge.net/
39 http://en.wikipedia.org/wiki/FFmpeg#Codecs_and_formats_supported

Moreover, in addition to media components and multiple versions of them, the DI may include descriptive information in different formats (for example, semantic annotations in Dublin Core and/or in MPEG-7, or contextual metadata such as terminal capabilities or user preferences) or rights expressions related data, among others.

- MPEG-21 Part 4 – Intellectual Property Management and Protection (IPMP)
  This part consists of the available tools for DRM management. Still, as the content that is encoded in the MPEG-21 standard is user-generated and hence has no limitations on use, we deem this part not very important.
- MPEG-21 Part 5 – Rights Expression Language (REL)
  Rights expression language is similar to policy languages, i.e., it enables the content provider to describe who can view or modify the content or, more generically, the type of operations that each user is entitled to perform upon the protected content.
- MPEG-21 Part 7 - Digital Item Adaptation (DIA)
  Using this part of the standard we can easily describe the capabilities of, for example, the user-terminal or of the network characteristics and conditions (as for example, bandwidth utilization). By describing our items using this part of the MPEG-21 standard, we can easily identify which items can be displayed on each type of end-user terminals; or we can obtain the information on how to adapt the content (by transcoding for example) to make it suitable to be presented in a specific device.. MPEG-21 DIA specifies a set of tools to assist the adaptation of multimedia content, when the content is in the form of Digital Items (DIs). It provides the means to characterize the context of usage and to relate the quality of audiovisual signals to application parameters and to the required resources, such as bandwidth, network losses or screen dimensions For example, it allows to relate a *good* or *very good* video quality with a specific set of values of the encoding parameters of available transcoders, together with the capabilities/characteristics of the terminal and conditions of the network. From the available MPEG-21 DIA tools, the more relevant to implement the adaptation of content to different context conditions (different terminal capabilities, different network conditions and different user preferences) are the Usage Environment Descriptor (UED), the Adaptation Quality of Service (AQoS) and the Universal Constraints Descriptor (UCD). UED provides the mechanisms to describe the characteristics of the environment in which the content is to be consumed, notably terminal capabilities, characteristics of the network and information regarding the user and his/her surrounding natural environment. UCD can be used to express the characteristics of the environment in terms of constraints. It enables the transformation of the current conditions of the usage context (conveyed as UED), into the form of constraints. AQoS provides the means to express the relation between resources (or constraints), adaptation operations and quality. The combined use of these tools, allows a system to decide which is the best adaptation operation to perform to meet the constraints imposed by the context (terminal, network and user) while providing the best possible quality (under the indicated constraints).

4.2. Delivery of content towards the end user

The MOSAICA solution for the distribution and download of the MOSAICA resources is based on the Bit Torrent suite of protocols, which can be seen as the lower layer of the p2p-CMS (details of the p2p-CMS architecture are provided in D3.1). Additionally, the upper layer of the p2p-CMS, based on the JXTA project, provides the access to a distributed database that maintains the metadata on the resources, enabling the execution of efficient search requests. Accordingly, by using the p2p infrastructure as a distributed content and metadata database, MOSAICA can provide fast search times with meaningful results and can exploit the use of replication mechanisms, through the use of multiple “seeders” for the download of the resources. We can thus expect the delivery of the content to be performed in reasonably fast times.

Furthermore, we can setup actual caching policies on super-nodes, if it is acceptable by the storage providers (i.e., the users running as super-nodes). The delivery times should significantly improve with the use of such technique as the content would no longer need to be reassembled from various seeders.

5. 3RD PARTY CONTENT IN MOSAICA

The approach towards using 3d party content (i.e., content offered by various museums, etc.), which is copyright protected, differs from the previously described approach for the user-generated content. The difference is in the fact that MOSAICA cannot store the actual content into the p2p-CMS system. Basically, p2p-CMS is used only as storage for MOSAICA semantic annotation and the actual link to the content on the original web-site. The MOSAICA users are therefore using the MOSAICA URIs, which can be viewed only as handles for retrieving the actual content from the original site. The slightly modified diagram of uploading such content is shown in Figure 4. Note that there is no transcoding step in the uploading process.

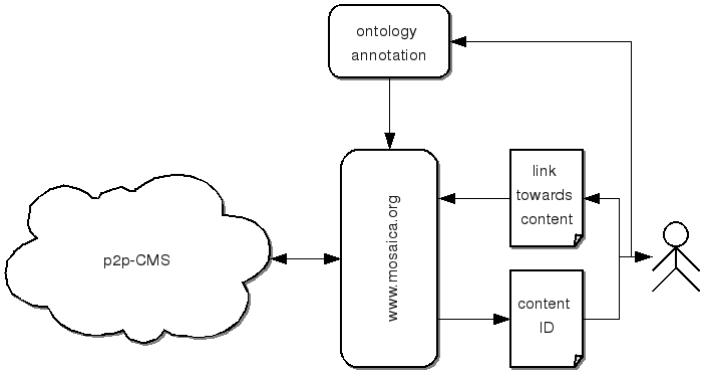

Illustration 4: Uploading of copyrighted content.

On the other hand, when using such content, we search for the contentID in the p2p-CMS, which then returns the link to the actual resource on the 3rd party website. Please note, that we can use transcoding on-the-fly in order to unify the format of the resources. Still, such use of the transcoding is time consuming (e.g. for video), hence we acknowledge the fact that we may omit it from the final system, hence we are placing a question mark on that functionality.

Transcoding when retrieving the content from third party websites may also be used to apply graphical filters or effects to images, which need to uniformly blend into, for example, virtual expeditions. These filters or effects should be simple as the original content must be clearly recognisable. Such operations must be defined in the Virtual Expedition Editor and must be stored along the link towards the resource. When retrieving the content, these filters must be applied.

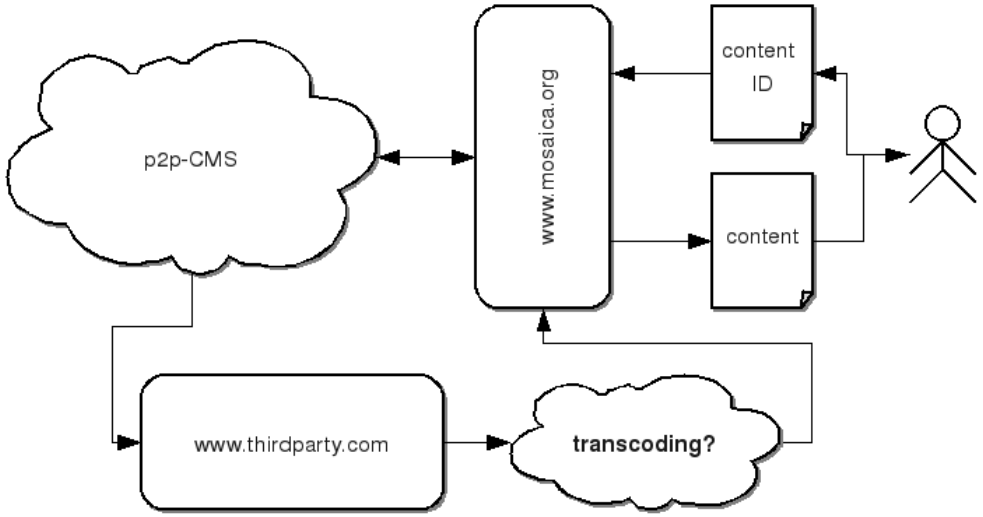

Illustration 5: Use of copyrighted content in MOSAICA

6. SECURITY IN MOSAICA

MOSAICA will rely on grouping and user settings provided by the p2p-CMS. The reasoning behind this choice is in the fact that all data can be accessed only through the p2p-CMS system. As the MOSAICA will have moderation groups, filtering possibly offending content, we cannot justify implementation of fully blown Security, Trust and Privacy subsystem. Still, the group and rights system implemented in the p2p-CMS can be later used to implement a more strict security subsystem, using for example Liberty Alliance (with XACML as provider of the policy-driven trust management, while SAML (the backbone of Liberty Alliance) will provide the basis for authentication and secure connection. In addition, as referred above, MOSAICA will have a moderation mechanism, by which all content submitted to the MOSAICA p22-CMS, will become available for download only after a MOSAICA administrator has validated that content. It is only at that point that the torrent of the uploaded content becomes visible and accessible to the MOSAICA peers for download.

7. CONCLUSIONS

Having explored the options for presenting the multimedia content on various devices, we can conclude the following:

- In order to preserve the MOSAICA experience solely as web based, we will use the Flash Video Format for presenting video content, with improvements gained by use of MPEG-21 multimedia framework, which will enable use of different bitrates, resolutions, based on end-devices.
- Sound and images will be encoded in MP3 or OGG and PNG or JPEG, respectively.
- The transcoding model that will be adopted in MOSAICA will be a service-based model. It will either be performed by a dedicated server offering the transcoding utility as a Web Service; or by a number of low-cost computers distributed along the MOSAICA p2p network, being nevertheless seen by the MOSAICA users, also as a Web Service accessible via the JXTA layer of the MOSAICA p2p-CMS (i.e., the JXTA will expose the transcoding service to the end users via a Web Services interface);
- The user-generated content will be transcoded upon storing in the p2p-CMS;

- The third party content will be transcoded when retrieved, but to smaller extent;
- Security will use the users and groups functionality from the p2p-CMS system.