



Technology Scouting into MediaMosa support for other content types

Version 1.0
Date 7 January 2010

SURFnet/Kennisnet Innovation programme

Contents

Summary	3
Introduction	4
Media in MediaMosa	5
Information structure	5
MediaMosa workflow	6
Upload and analysis	6
Storage	7
Required extensions and adaptations	8
Separating analysis and core workflow	8
Configurable workflow and tools	8
Upload functionality	8
Analysis workflow	9
Core workflow	10
Metadata analysis and storage	11
Content analysis of files	11
Multiple storage locations	11
Transcoding and processing	12
Playing files	12
Categorisation by content type	13
Conclusions and recommendations	15

Summary

MediaMosa¹ is the open source media content management system developed by SURFnet, Kennisnet and Madcap. The VP-Core video platform was implemented using MediaMosa. This technology scouting investigates which adaptations are required in MediaMosa to also be able to store and process media other than video.

MediaMosa is a platform that can be used by end-user applications (EUA) to store and play video files. MediaMosa is 'invisible' to the end-user of these applications.

If MediaMosa were to support other types of content than video, an EUA could add new functionality for end-users. An example is making available a biography of the actor and the spoken text (in text format) for a video fragment.

This report proposes a number of adaptations and extensions to add support for content types other than video to MediaMosa. Most of these adaptations or extensions were included in the original design of MediaMosa, but have not, or only partially, been implemented.

The required adaptations and extensions are:

- Replacing the current processing by a configurable workflow;
- Dividing the processing into two separate workflows: an analysis and a core workflow;
- Allowing the EUA to use the REST interface to manage the configuration of the core workflow;
- Extending the metadata store with a record for each media file that holds the data that are generated by the analysis.

Moreover this technology scouting provides an initial sketch of the design of the analysis workflow by categorising media types into groups and identifying a toolset for each of these groups.

A first logical step would be to extend and configure MediaMosa to process images and audio files (in addition to video content). This extension is a logical functional extension of the current implementation in VP-Core. Moreover, suitable tools have been identified for analysing and processing these content types. A more far-reaching study may identify a set of tools to cover the analysis and processing in MediaMosa for the other types.

¹ <http://mediamosa.org/>

Introduction

MediaMosa is a robust and flexible and scaleable media management platform. It forms the basis of many successful video services provided by SURFnet and Stichting Kennisnet to the education community in the Netherlands now and in the future (SURFmedia for example). MediaMosa has recently been made available in an Open Source licence by both these organisations and its developer, Madcap².

VP-Core is the implementation of MediaMosa within SURFnet. VP-Core makes video content accessible that is stored on the joint SURFnet and Kennisnet video platform. In this manner VP-Core is used as a repository for SURFmedia³. Whenever the term MediaMosa is used in the remainder of this document, this will also be taken to include the SURFnet-specific VP-Core implementation.

MediaMosa also provides the possibility to distribute software and other content as learning objects. This technology scouting investigates the adaptations to VP-Core that are required for supporting storage and dissemination of other content besides video.

The first chapter describes the current processing of video files in MediaMosa. The asset concept and the current workflow are described, from uploading via analysis and transcoding to retrieval and replay.

The chapter 'Required extensions and adaptations' provides an overview of proposed adaptations of and extensions to MediaMosa. Based on an analysis of the current design and implementation a number of issues have been identified that require adaptations in order to support other content.

Moreover, types of media files have been categorised. File types have been categorised into content groups based on their (technological) characteristics. For each content type the suitable tooling for the processing of such files was investigated. Based on the results, described in the chapters 'Required extensions and adaptations' and 'Categorisation by content type', three tools are recommended for MediaMosa.

The section 'Conclusions and recommendations' provides an overview of our findings and the recommended extensions and adaptations. This chapter also advises further in-depth study by carrying out a number of spikes (in-depth explorations).

² <http://www.madcap.nl/>

³ <http://www.surfmedia.nl/>

Media in MediaMosa

MediaMosa provides an infrastructure that is used by end-user applications (EUA); end-users do not interact directly with MediaMosa. The EUAs invoke services provided by MediaMosa for retrieving, processing and storing assets. Playing media files is supported by MediaMosa through the play-proxy service. This proxy is responsible for access control; media files are to be made available to authorised users only.

Information structure

The basic elements in the MediaMosa data structure design (see Figure 1 Information structure Asset) are assets. One asset comprises a set of metadata, both descriptive and technical, and linked media files. Before a media file is uploaded to MediaMosa an asset is created and the uploaded file is linked to this asset.

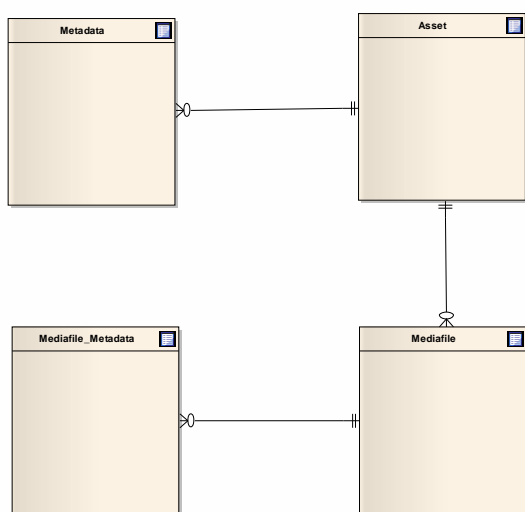


Figure 1 Information structure Asset

An asset can comprise multiple media files; in the case of video these can be various versions of the same video fragment that are encoded in different formats. Media files for an asset can also differ in type and content, such as video stills from a video file.

A media file has its own set of metadata. It comprises mainly technological data, such as the used codec, bitrate, fps, etc. This data is automatically derived from the media files themselves during the analysis phase.

Depending on the configured workflow MediaMosa can generate multiple media files (transcoding into a different format, creating 'stills' with video fragments) on the basis of a single uploaded media file, and link it to the asset.

MediaMosa workflow

The workflow of the processing of a media file in MediaMosa comprises the following steps:

- **Uploading** the original file. This step consists of four sub-steps: creating the asset, creating a media file for this asset, requesting an upload ticket, and uploading the file.
- **Analysis:** the file is analysed to determine the (technical) format. On the basis of the result of this analysis a file may be transcoded into a different format. The underlying assumption for the analysis is that the file extension is not reliable; MediaMosa determines (and overrules) the format and type assigned (implicitly) by the EUA or the user.
- **Storage:** the media file is stored on the underlying storage medium.
- **Transcoding:** A command to transcode the media file into a number of different formats may be included when it is uploaded.
- **Replay:** This last step is separate from the workflow for creating the asset and uploading media files. An end-user submits a request to replay a media file through an EUA. The EUA passes this request to MediaMosa.

Replaying a media file in the context of MediaMosa involves a choice from three types of replay actions:

- Returning a URL to a replay proxy;
- Returning a segment of HTML code (which contains a link to the replay proxy URL) to be used to embed the video in an HTML page;
- Returning a link to download the file.

MediaMosa executes this workflow in part by a job mechanism. Jobs are created and planned for tasks that require a substantial processing time and capacity, such as transcoding. Besides these tasks MediaMosa also supports jobs for checking the integrity between the information on assets in the database and the files stored on the file system, as well as a job for cleaning up empty assets and media files.

Upload and analysis

In the current workflow the analysis of the media files is limited to determining the (video) format and determining a number of (technical) metadata fields. There is no check or analysis on the content. This workflow starts automatically when a file is uploaded.

The analysis of uploaded media files is now carried out by means of the file command and the FFmpeg⁴ tool, specially suited for determining video formats (and codecs). The file command compares the headers of files with a set of so-called 'magic files' of known formats, and determines the type in this way. In the next step an FFmpeg is used for a more in-depth analysis of video files.

⁴ <http://ffmpeg.org/>

As the name suggests, the current ftp batch upload only works with ftp; media files and metadata can be uploaded to MediaMosa using ftp. A Video Upload File (VUF) is uploaded along with these file collections. The VUF contains the instructions on how to import them in MediaMosa.

Storage

MediaMosa uses a single mode for storing uploaded files. In the current processing all files are written to the same location. It is not possible to distinguish between specific locations because MediaMosa uses a single mount point to an underlying file system. It is not possible to make a specific choice on the basis of characteristics of the file, EUA or collection.

Required extensions and adaptations

This chapter discusses the extensions to and adaptations of MediaMosa, required to be able to process other content types. The number of changes may seem substantial, but for the most part these are an implementation of facilities that were already envisaged in the original design.

Separating analysis and core workflow

Support of more content types in MediaMosa requires a separation in the workflow. The current steps in the workflow are more or less hard-coded and aimed at video files. A more flexible and future-proof platform results if the processing were to consist of two separate steps: an analysis and metadata extraction step followed by the actual processing.

Configurable workflow and tools

A configurable workflow creates a step-based scheme, with each step corresponding to performing a predefined action on a file. Examples of such actions are invoking an (external) tool for the analysis or the transcoding of a file.

Both the analysis workflow and the core workflow make use of the same rule module, in which the two workflows are created and managed. The Drupal Rules module appears to be the best candidate here.

The current MediaMosa setup can be applied when adding external tools for analysing (and/or processing) files. This means that a tool is invoked using the command line of the operating system. The output of the tool (text or XML-based) is directed to MediaMosa and is further processed there. This mechanism offers a very flexible method of integrating tools with MediaMosa.

Upload functionality

The analysis workflow starts when the file is uploaded. The current upload mechanism suffices for other content types as well. No adaptations are required here.

However, there are two scenarios that do require an extension of the upload function:

- uploading batches of assets bundled in archives;
- uploading rich media formats which require that the internal integrity is safeguarded.

In both scenarios the MediaMosa upload function needs to be extended with the possibility to process zip files comprising all the constituent parts of the asset together with processing instructions. This could be implemented as an extension of the current ftp upload mechanism, by placing the VUF file within the zip file.

Analysis workflow

The basic principle for the analysis workflow remains that MediaMosa determines the format and type of a media file, even when working with multiple content types. The analysis needs to be extended with additional steps, and with tools that can sufficiently analyse the media files for each additional step.

A configurable workflow allows this analysis to be configured within MediaMosa. A uploaded file can be tested by multiple tools. Based on the test result a next analysis step can be initiated or skipped.

The result of the analysis workflow is the aggregated metadata, which is stored with the uploaded file. The results of each analysis step are combined into a single set of metadata for the file. The MediaMosa analyser component returns this to the Core in RFC 822⁵ format.

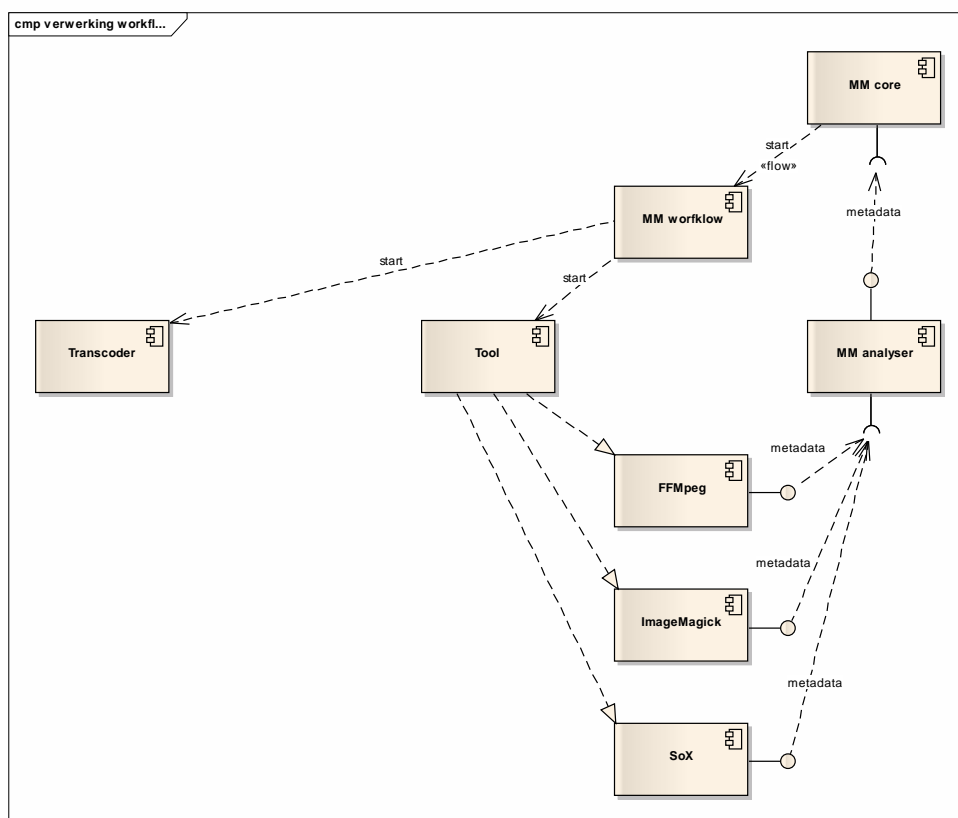


Figure 2 Analysis workflow

⁵ <http://tools.ietf.org/html/rfc822>

Figure 2 shows the structure of the analysis workflow. It is initiated by the MediaMosa Core when a file is uploaded. The workflow component then starts the analysis workflow. The analysis workflow configuration determines which tools are invoked and in which order. The result of this analysis or these analyses is then aggregated by the MediaMosa analyser component.

Core workflow

The Core is assigned its own workflow, which decides on the steps required for the actual processing on the basis of the result of the analysis. These steps can be actions such as transcoding to a different format, or adding a file to a specific collection.

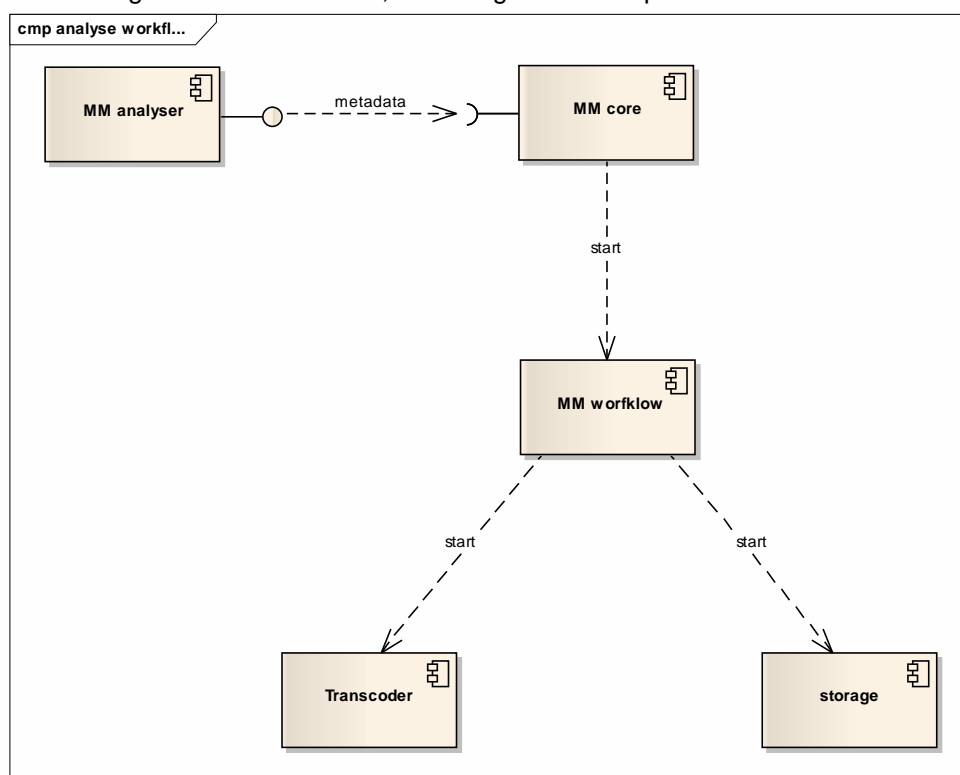


Figure 3 Core workflow

This workflow is shown in figure 3. Metadata is submitted by the analysis to the MediaMosa Core. This is where the workflow for the Core processing starts. The Core workflow operates on assets. The workflow is not restricted to using the metadata of the newly uploaded file to make decisions; it can also use the metadata of the asset and of the other media files in the asset.

As the desired workflow at this level is determined by the EUA, it will need to be managed and controlled by the EUA. MediaMosa's REST interface will have to be extended with the relevant operations to support this. The EUA can indicate a specific workflow profile when uploading a file. The EUA must define these profiles in advance in MediaMosa using (new) REST operations.

Metadata analysis and storage

Not only will the number of analysis steps be increased to support multiple content types, the support of metadata within the analysis will be extended as well.

The analysis of the media file will result in a different and or larger set of metadata compared to the current one, depending on the tools, the file format and the properties of the media type.

Metadata can be divided into technical and descriptive data. The technical data provides information on the technical properties of the media file, such as the codec used or the number of pixels. The descriptive data comprises information at the end-user level, including title, description and tags.

MediaMosa considered all metadata returned by FFmpeg after analysing a media file to be technical data. This is no longer a valid assumption when other content is also supported. Formats like jpeg contain both technical data and metadata defined by the user, such as tags.

The way to resolve this in MediaMosa is to ensure that the metadata resulting from the analysis of the file is always available without modification. This metadata is stored with an asset in its own record, to ensure that it remains available even after the metadata is updated (for instance by the EUA). The metadata resulting from all analysis tools that are applied to a media file is combined and stored as a single metadata set.

Content analysis of files

A check of the content of media files is also required, even if just from a security perspective, in addition to retrieving the file type and format and acquiring the metadata. A check for viruses and other risks can be enabled by the systems management provider (as a storage service). Explicit support in MediaMosa does not seem required.

MediaMosa will have to implement a notification or synchronisation facility that logs events in the MediaMosa administration when contaminated files are found and removed. Kennisnet Groepen (2.0) has been provided with a similar facility; this could also be applied to MediaMosa.

Multiple storage locations

The workflow can be extended with a configurable option for the location where files are to be stored based on the determined content type and file format. This option offers possibilities for managing performance (specialised high-performance storage) and costs (more affordable hosting).

Transcoding and processing

The current setup, where transcoding only takes place after an upload with an explicit transcoding instruction, will be extended. MediaMosa will initiate other processing tasks, depending on the file type as determined during the analysis. Examples of such processing tasks are generating stills for videos, or converting (PowerPoint) presentations to flash.

Similar to the analysis stage, a workflow can then be configured for each (group of) file types, allowing specific tools to be invoked for the processing and/or conversion of files.

In addition to uploading single files and VUFs, it is a desirable feature to support the processing of a zip (or other format) archive based on an included VUF. This provides the option of using protocols other than ftp to upload files in batches.

This extension could also be applied to rich content. In that case an asset that comprises multiple related media types may be uploaded and created as a whole. The VUF instruction set should be extended depending on the format of the rich content.

Playing files

The options for replay of a file type strongly depend on the properties of that type and on the workflow defined for that type in MediaMosa. The three basic actions as they are currently defined are: return a replay URL, return embedded code, and return a download link. These will have to be defined for each distinct (group of) media type.

Categorisation by content type

We categorised media types to be able to analyse the possibilities and impossibilities of supporting other content types. This categorisation is based on the interactive and management properties of the types and especially on the available (open source) analysis tools. The categories and the assignment of certain files to these categories is arbitrary to a certain extent, but it provides the necessary footholds to undertake the analysis.

The categories range from 'simple' text types to 'rich media', a category in which multiple media files of multiple types can be delivered and played in combination. The categories are the following:

- Text: comprises text documents, as well as files with limited formatting and (non-XML) wiki formats;
- Schema: comprises all XML and SGML-like types;
- Office/doc: the Microsoft and Open Office document types (.doc, .docx, .odt) and pdf;
- Office/spreadsheet: the Microsoft and Open Office spreadsheet types;
- Office/presentation: the Microsoft and Open Office presentation types;
- Archive: Compound files such as zip, gzip and jar;
- Image: well-known image formats;
- Audio: well-known audio formats such as .wav, .mp3;
- Video: the current set of video formats;
- Interactive applications: Applications that may be executed in a browser, such as flex, flash, json, fx;
- Rich media: Specific formats of related data, for instance SCORM⁶.

Table 1 provides an overview of the results of this analysis.

⁶ http://en.wikipedia.org/wiki/Sharable_Content_Object_Reference_Model

Content group	Types	Tools	Metadata
general	all	file	owner, file name, file date
text	txt, rich text, (media)wiki		difficult to determine
schema	xml, xsd, html	parser, schema check	depending on schema
office	OOXML, ODF, PDF	office to PDF converter, metadata extractor	extended schemas especially aimed at 'user data'
archive	zip, rar, arj, jar	various	depending on type, ranging from technical (compression method) to descriptive (manifest files in the archive)
image	gif, jpeg, png, svg/xml, tiff	ImageMagick	depending on type and use. The focus usually ranges from technical (colour depth) to descriptive (tags, geodata)
audio	wav, wma, vorbis, mp3	SoX	usually technical, but descriptive is also supported by formats (title, artist, album)
video	wmv, flash, mpeg4/h264, Quicktime	FFmpeg	usually technical
application	flex, flash, silverlight, fx, json	usually restricted to support for replay	usually technical
rich	scorm	within ELOs, little tooling available as yet	depending on format (and support for the format)

Table 1 Categorisation of file types into content groups

This analysis points to three open source tools as candidates for use in MediaMosa. These are ImageMagick⁷ for analysing images and SoX⁸ for processing audio files, in addition to FFmpeg that is already being used.

Tools for other categories were identified, but a choice could not yet be made based on this first analysis.

⁷ <http://www.imagemagick.org/>

⁸ <http://sox.sourceforge.net/>

Conclusions and recommendations

The current implementation of MediaMosa is aimed at processing video files. Extending the functionality to support other content can build upon the already existing job mechanism. The fact that MediaMosa will analyse the uploaded media files will remain the basic principle.

A second principle is that EUAs remain responsible for the internal integrity of assets. MediaMosa has no type system for assets, as types are assigned to media files (based on the analysis). For rich media a hierarchy of assets can be used to store the related collection of media files as one main asset in the repository.

To support other content effectively the current hard-coded workflow will need to be replaced with a configurable workflow, using external tools for analysing and processing media files. The metadata set that is generated by the analysis needs to be stored 'permanently' with the media file.

Also, the processing will need to be separated into two independent workflows for the analysis and for the Core. The analysis of multiple content types will require multiple steps using several tools, requiring a specific workflow in each case. MediaMosa Core will require a workflow that directs further processing and storage based on the results of the analysis.

The Core workflow is determined by the required behaviour of the EUA. For this reason it needs to be managed and controlled by the EUA through the REST interface. An EUA can assign a workflow profile when files are uploaded. This profile must be defined in advance by the EUA in MediaMosa (through the REST interface).

MediaMosa provides three replay modes for media files (play via a replay proxy, HTML code with a link to the replay proxy for embedding and a download link) that will also be supported with other media types. The configurable workflow needs to provide support for specific (groups of) content types to prepare a version of the file for each replay mode. For instance, a PowerPoint presentation may be submitted to the replay proxy in Flash format.

The processing using separate configurable workflows will reinforce the flexibility and the future-proofness of MediaMosa. The described categories of media types and the recommended tools can serve as a foundation for designing the workflow. The extent to which the (standard) MediaMosa features must be detailed will eventually be determined by the aims and the requirements of the EUA users.

A first logical step would be to extend and configure MediaMosa to support processing of images and audio files (besides current support for videos). Good tools are available for analysing and processing these content types. For the remaining types an in-depth study (spike) can identify a set of tools to cover analysis and processing in MediaMosa.